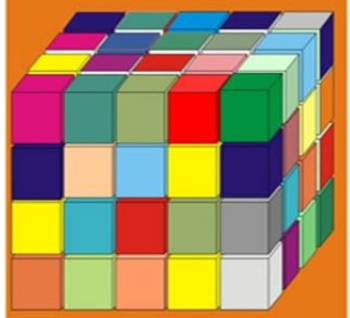
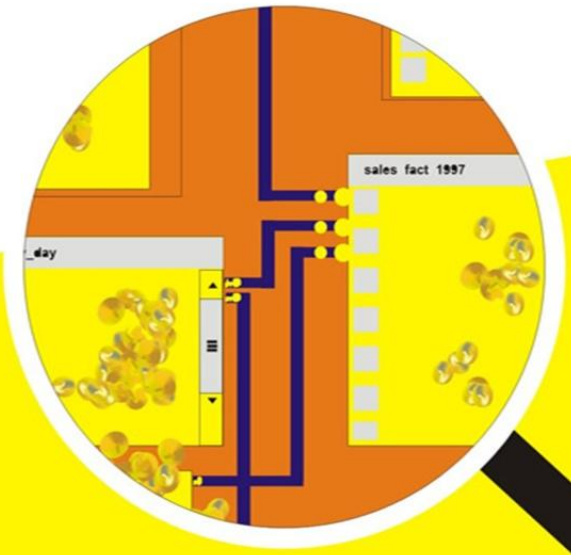
Βάσεις, Αποθήκες και Εξόρυξη Δεδομένων με τον SQL Server

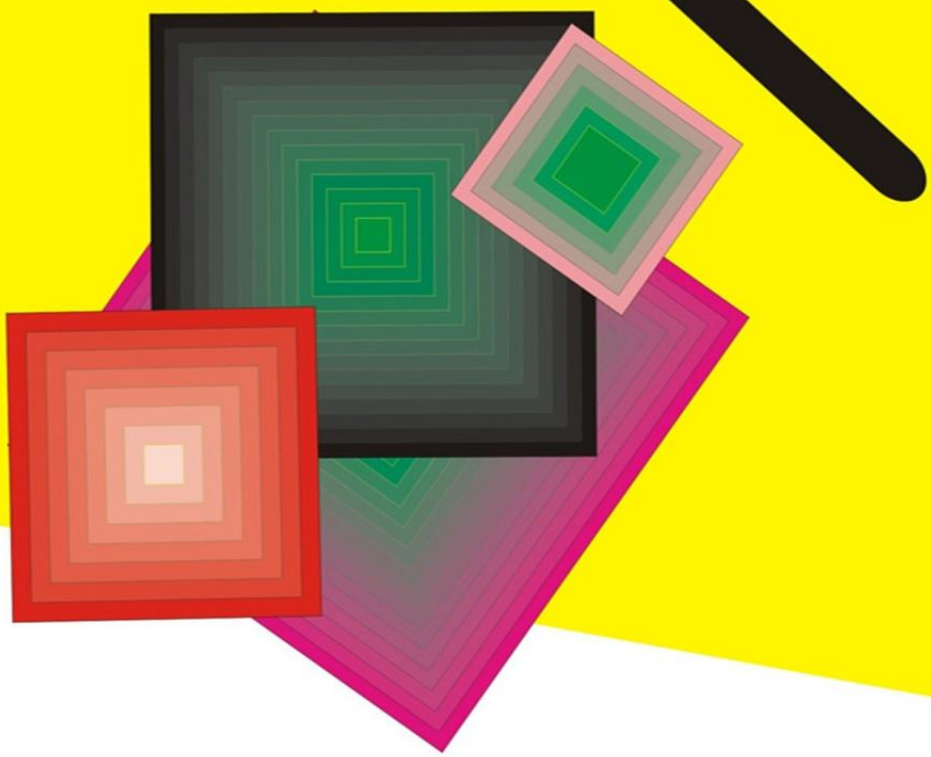
Εργαστηριακός Οδηγός

Παναγιώτης Συμεωνίδης
Αναστάσιος Γούναρης

Τμήμα Πληροφορικής
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης







ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
HEALLINK
Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών
Ελληνικά Ακαδημαϊκά Ηλεκτρονικά Συγγράμματα και Βοηθήματα
www.kallipos.gr

ΠΑΝΑΓΙΩΤΗΣ ΣΥΜΕΩΝΙΔΗΣ
Διδάκτωρ Τμήματος Πληροφορικής
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

ΑΝΑΣΤΑΣΙΟΣ ΓΟΥΝΑΡΗΣ
Επ. Καθηγητής Τμήματος Πληροφορικής
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Βάσεις, Αποθήκες και Εξόρυξη Δεδομένων με τον SQL Server

Εργαστηριακός Οδηγός

Τίτλος Ηλεκτρονικού Συγγράμματος

Βάσεις, Αποθήκες και Εξόρυξη Δεδομένων με τον SQL Server
Εργαστηριακός Οδηγός

Κύριος Συγγραφέας

Παναγιώτης Συμεωνίδης

Συν-συγγραφέας

Αναστάστιος Γούναρης

Κριτικός αναγνώστης

Ιωάννης Θεοδωρίδης

Συντελεστές έκδοσης

ΓΛΩΣΣΙΚΗ ΕΠΙΜΕΛΕΙΑ: Φώτης Συμεωνίδης

ΓΡΑΦΙΣΤΙΚΗ ΕΠΙΜΕΛΕΙΑ: Ανδρέας Πάσχος

ΤΕΧΝΙΚΗ ΕΠΕΞΕΡΓΑΣΙΑ: Χρήστος Άνδρας

ISBN: 978-960-603-021-5

ΣΥΝΔΕΣΜΟΣ ΕΛΛΗΝΙΚΩΝ ΑΚΑΔΗΜΑΪΚΩΝ ΒΙΒΛΙΟΘΗΚΩΝ

Εθνικό Μετσόβιο Πολυτεχνείο

Ηρώων Πολυτεχνείου 9, 15780 Ζωγράφου

www.kallipos.gr

Πίνακας περιεχομένων

Πίνακας συντομεύσεων-ακρωνύμια

ARIMA	AutoRegressive Integrated Moving Average
ARTXP	AutoRegressive Tree XP model
BDE	Bayesian Dirichlet Equivalent with Uniform Prior
BK2	Bayesian with K2 Prior
DBO	Data Base Owner
DDL	Data Definition Language
DML	Data Manipulation Language
EM	Expectation Maximization
KDD	Knowledge Discovery in Databases

Εισαγωγή

Αυτό το ηλεκτρονικό βιβλίο αποτελεί έναν εργαστηριακό οδηγό σε θέματα βάσεων, αποθηκών και εξόρυξης δεδομένων με τον MS SQL Server. Το ebook αξιοποιεί όλα τα σύγχρονα χαρακτηριστικά των tablets και του διαδικτύου, προσφέροντας στον αναγνώστη μοναδικές δυνατότητες σε σχέση με τα συμβατικά βιβλία και τον μέχρι πρότινος τρόπο διδασκαλίας στο εργαστήριο πληροφορικής.

Μερικές από τις κύριες δυνατότητες που προσφέρονται από το ebook είναι οι ακόλουθες:

1. Μπορεί να αναγνωστεί σε όλες τις προσφερόμενες πλατφόρμες των tablets (Android, iOS, κλπ.).
2. Περιέχει έναν κεντρικό πίνακα περιεχομένων μέσα από τον οποίο, ο αναγνώστης μπορεί να περιηγηθεί στα 11 κεφάλαια του βιβλίου. Επίσης, είναι εμπλουτισμένο με εκατοντάδες υπερσυνδέσμους που παρέχουν στον αναγνώστη επιπρόσθετες επεξηγήσεις όρων και λέξεων κλειδιών.
3. Το ebook έχει περισσότερες από 450 εικόνες/σχήματα που καθοδηγούν τον αναγνώστη βήμα-βήμα στην εκτέλεση των διαδικασιών που περιγράφονται κάθε φορά. Επίσης, εμπεριέχει 25 πίνακες με συγκεντρωτικές πληροφορίες.
4. Στο τέλος κάθε κεφαλαίου υπάρχουν ασκήσεις για την καλύτερη εμπέδωση της ύλης του. Συνολικά, στο ebook διατίθενται 70 ασκήσεις, εκ των οποίων οι 45 είναι λυμένες.
5. Το πρώτο μέρος του βιβλίου αφορά μια βάση δεδομένων ενός DVDclub, η οποία είναι διαθέσιμη συνεχώς στο διαδίκτυο για να προσπελαστεί οποιαδήποτε χρονική στιγμή από τον αναγνώστη. Επίσης, διατίθενται 75 εκφωνήσεις ερωτημάτων SQL με τις λύσεις τους και τα αποτελέσματα του κάθε ερωτήματος.
6. Το ebook διαθέτει έναν Web-based SQL Editor όπου ο αναγνώστης μπορεί να συντάσσει, να τρέχει και να βλέπει τα αποτελέσματα των ερωτημάτων του, χωρίς να υπάρχει ανάγκη για εγκατάσταση κανενός λογισμικού, π.χ. Sql Server 2015.

Συνοψίζοντας, το ebook επιτρέπει στον αναγνώστη να κατανοήσει και να εκτελέσει ερωτήματα και αλγορίθμους χωρίς την ανάγκη ύπαρξης κάποιου εργαστηρίου πληροφορικής ή κάποιου λογισμικού. Αναλυτική περιγραφή των δυνατοτήτων και οδηγίες χρήσης του ebook δίνονται στην ακόλουθη τρίλεπτη βιντεοπαρουσίαση (https://www.youtube.com/watch?v=ldM-GcBD_yA). Το ebook αποτελείται από δύο μέρη.

Πρώτο μέρος: Θέματα Βάσεων Δεδομένων (Κεφάλαια 1-5)

Το πρώτο μέρος του βιβλίου (Κεφάλαια 1- 5) περιγράφει τις βασικές λειτουργίες και δυνατότητες που προσφέρονται από ένα **σύστημα διαχείρισης βάσεων δεδομένων (data base management system)**. Συγκεκριμένα, στο βιβλίο αυτό περιγράφονται οι δυνατότητες του SQL Server 2014. Ο SQL Server είναι ένα ισχυρό εργαλείο που περιέχει ένα πλήθος δυνατοτήτων, όπως είναι οι εντολές που αφορούν τη **γλώσσα ορισμού δεδομένων (Data Definition Language)**, οι εντολές που αφορούν τη **γλώσσα χειρισμού δεδομένων (Data Manipulation Language)**, και η γλώσσα προγραμματισμού Transact-SQL για την δημιουργία αποθηκευμένων διαδικασιών, εναυσμάτων και συναλλαγών.

Πιο αναλυτικά, στο πρώτο κεφάλαιο, που είναι εισαγωγικό, ο αναγνώστης θα πληροφορηθεί πώς μπορεί να εγκαταστήσει τον SQL Server 2014. Συγκεκριμένα, περιγράφονται τα βασικά βήματα εγκατάστασης του SQL Server και, στη συνέχεια, ο έλεγχος που απαιτείται να γίνει, προκειμένου να βεβαιωθεί η καλή λειτουργία της εγκατάστασης. Τέλος, γίνεται μια σύντομη ξενάγηση στο βασικό γραφικό περιβάλλον του SQL Server Management Studio.

Το δεύτερο κεφάλαιο περιγράφει τη δημιουργία μιας βάσης δεδομένων που αφορά ένα DVDclub. Θα εργαστούμε κυρίως με εντολές από τη γλώσσα ορισμού δεδομένων (DDL). Θα συζητήσουμε για τους διαφορετικούς τύπους δεδομένων και για τον τρόπο δημιουργίας πινάκων με γραφικό τρόπο και με εντολές της SQL. Τέλος, θα δημιουργήσουμε συσχετίσεις μεταξύ των πινάκων στο Database Diagram και θα εισαγάγουμε τιμές στους πίνακές μας.

Στο τρίτο κεφάλαιο θα παρουσιάσουμε βασικά και σύνθετα ερωτήματα της SQL. Τα ερωτήματα θα υποβληθούν στη βάση δεδομένων DVDclub που δημιουργήθηκε στο προηγούμενο κεφάλαιο. Συγκεκριμένα,

θα μελετηθούν εντολές της SQL που αφορούν τη διαχείριση δεδομένων (DML). Ενδεικτικά αναφέρεται ότι θα παρουσιαστούν ερωτήματα σύνδεσης πινάκων, ομαδοποίησης, πράξεων συνόλων, καθώς και η δημιουργία ερωτημάτων με γραφικό τρόπο (Query by Example) μέσα από το περιβάλλον του Query Designer.

Στο τέταρτο κεφάλαιο θα παρουσιάσουμε προχωρημένες λειτουργίες που γίνονται στο περιβάλλον του SQL Server. Συγκεκριμένα, θα μελετήσουμε εντολές της SQL που αφορούν τη γλώσσα DDL, όπως τη μεταβολή της δομής των πινάκων, τη διαγραφή τους κτλ. Άλλα προχωρημένα θέματα που θα παρουσιαστούν είναι η βελτιστοποίηση ερωτημάτων με τη χρήση indices, οι αποθηκευμένες διαδικασίες, τα εναύσματα και οι συναλλαγές με τη βοήθεια της γλώσσας προγραμματισμού Transact-SQL.

Στο πέμπτο κεφάλαιο θα περιγράψουμε τη δημιουργία φορμών, προκειμένου να εισάγουμε δεδομένα και να εμφανίζουμε στοιχεία από τους πίνακες της βάσης DVDclub με έναν τυποποιημένο τρόπο. Συγκεκριμένα, θα μελετήσουμε πώς μπορούμε να φτιάχνουμε απλές κύριες φόρμες, καθώς και κύριες με δευτερεύουσες φόρμες στο περιβάλλον της MS Access. Επιπροσθέτως, θα περιγράψουμε τη δημιουργία λιστών αναζήτησης για τη γρήγορη εύρεση στοιχείων σε μια φόρμα αναζήτησης και τη δημιουργία υπολογιζόμενων πεδίων φορμών (derived attributes).

Δεύτερο μέρος: Θέματα Αποθηκών και Εξόρυξης Δεδομένων (Κεφάλαια 6-11)

Το δεύτερο μέρος του βιβλίου (Κεφάλαια 6 -11) περιγράφει τις βασικές τεχνικές εξόρυξης δεδομένων και τη διαδικασία δημιουργίας κύβων δεδομένων (data cubes). Η **εξόρυξη δεδομένων (Data Mining)** ή, στη διεθνή ορολογία, Knowledge Discovery in Databases (KDD) είναι μια προηγμένη διαδικασία ανάλυσης μεγάλου όγκου δεδομένων. Αυτή η ανάλυση των δεδομένων μπορεί να γίνει με τη χρήση τεχνικών όπως decision trees, clustering, association rules, time series κτλ. Οι **κύβοι δεδομένων (data cubes)** είναι ένας τρόπος οργάνωσης των δεδομένων σε συγκεντρωτικούς πίνακες (Pivot tables) για την πραγματοποίηση γρήγορης ανάλυσης των δεδομένων και την λήψη στρατηγικών αποφάσεων. Οι τεχνικές αποθηκών και εξόρυξης δεδομένων ξεπερνούν κατά πολύ σε δυνατότητες ανάλυσης την DML. Στο δεύτερο μέρος, λοιπόν, περιγράφονται οι παραπάνω τεχνικές μέσα από τη χρήση του MS SQL Server 2014 Business Intelligence. Ο Business Intelligence του Visual Studio είναι ένα ισχυρό εργαλείο που περιέχει ένα πλήθος από αλγορίθμους για την υλοποίηση των τεχνικών εξόρυξης δεδομένων που έχουν ήδη αναφερθεί.

Κάθε κεφάλαιο του δεύτερου μέρους περιγράφει μια διαφορετική τεχνική εξόρυξης δεδομένων, με εξαίρεση το τελευταίο που περιγράφει τη δημιουργία ενός κύβου δεδομένων. Πιο αναλυτικά, στο έκτο κεφάλαιο, που είναι εισαγωγικό, ο αναγνώστης αρχικά θα πληροφορηθεί πώς μπορεί να δημιουργήσει μία βάση δεδομένων με τη χρήση του SQL Server Management Studio. Στη συνέχεια, θα εισαγάγει αυτή τη βάση σε ένα νέο project που θα δημιουργήσει στο Data Tools του Visual Studio. Η εισαγωγή και η προεπεξεργασία δεδομένων θα γίνει σε τρεις διαφορετικές βάσεις δεδομένων (MovieClick, FoodMart, AdventureWorks), προκειμένου, μέσω αυτών, να εφαρμόσουμε τεχνικές εξόρυξης δεδομένων σε επόμενα κεφάλαια. .

Το έβδομο κεφάλαιο περιγράφει την κατηγοριοποίηση με τη χρήση δέντρων απόφασης. Γίνεται περιγραφή του τρόπου επιλογής των δεδομένων της βάσης δεδομένων που χρησιμοποιούμε και , ακολούθως, δίνεται μια αναλυτική περιγραφή των παραμέτρων του αλγορίθμου. Στο παράδειγμα που περιγράφεται, καθώς και στις ασκήσεις που συμπληρώνουν την ενότητα, γίνεται προβολή του δέντρου και αξιολόγηση της ικανότητας πρόβλεψης του μοντέλου μας για διαφορετικές τιμές των παραμέτρων του αλγορίθμου.

Το όγδοο κεφάλαιο ασχολείται με το clustering μέσω των αλγορίθμων ομαδοποίησης k-means και EM clustering. Όπως και στο προηγούμενο κεφάλαιο, γίνεται μια αναλυτική περιγραφή των παραμέτρων του κάθε αλγορίθμου και βλέπουμε τα clusters που δημιουργούνται μεταβάλλοντας τις τιμές των παραμέτρων αυτών. Η αξιολόγηση των clusters γίνεται με δύο τρόπους: με τη χρήση διαγραμμάτων και με την αναλυτική εξερεύνηση του κάθε cluster, ώστε να διερευνήσουμε από ποια στοιχεία-μέλη συνίσταται το καθένα.

Στο ένατο κεφάλαιο περιγράφεται ο αλγόριθμος εξαγωγής κανόνων συσχέτισης. Ο συγκεκριμένος αλγόριθμος παράγει συσχετίσεις μεταξύ συνόλων αντικειμένων και ανήκει στην οικογένεια των Apriori αλγορίθμων. Οι ομάδες αντικειμένων που εξάγονται ονομάζονται itemsets (σύνολα αντικειμένων). Με βάση τα itemssets που έχουν παραχθεί, εξάγονται οι κανόνες συσχέτισης μεταξύ των αντικειμένων. Ένας κανόνας συσχέτισης σηματοδοτεί τη συσχέτιση ενός συνόλου αντικειμένων από ένα άλλο σύνολο αντικειμένων. Μέσα από τις ασκήσεις περιγράφονται αναλυτικά τα itemsets και οι association rules που δημιουργεί ο αλγόριθμος αλλάζοντας τις τιμές των παραμέτρων του.

Στο δέκατο κεφάλαιο περιγράφεται η δημιουργία ενός μοντέλου χρονοσειρών (time series). Συγκεκριμένα, θα μάθουμε τον τρόπο με τον οποίο δημιουργείται και χρησιμοποιείται ένα μοντέλο χρονοσειρών για την βάση δεδομένων AdventureWorks. Η Adventure Works είναι μια βάση δεδομένων που

αφορά μια υποθετική πολυεθνική εταιρία που εμπορεύεται ποδήλατα σε διάφορες ηπείρους/χώρες. Το τμήμα πωλήσεων αυτής υποθέτουμε ότι επιθυμεί να προβλέψει τις μελλοντικές πωλήσεις ανά μοντέλο ποδηλάτου βάσει των πωλήσεων που σημειώθηκαν στο παρελθόν. Ένα μοντέλο χρονοσειρών είναι πολύ χρήσιμο σε τέτοιες περιπτώσεις.

Το ενδέκατο κεφάλαιο αποτελεί μία εισαγωγή στις αποθήκες και τους κύβους δεδομένων. Ο κύβος είναι μια πολυδιάστατη δομή δεδομένων που εμπεριέχει συναθροιστικές πληροφορίες για μια ή περισσότερες βάσεις δεδομένων. Η συνάθροιση της πληροφορίας οδηγεί σε γρήγορους χρόνους απόκρισης ερωτημάτων που τίθενται από υψηλόβαθμα στελέχη επιχειρήσεων, προκειμένου αυτά να λάβουν συνήθως στρατηγικές αποφάσεις για την επιχείρηση.

Συμπερασματικά, ένας χρήστης του SQL Server μπορεί με τους αλγορίθμους εξόρυξης δεδομένων που διατίθενται από το περιβάλλον να προβλέψει τις τιμές των χαρακτηριστικών που τον ενδιαφέρουν, ενώ με τη δημιουργία κύβων δεδομένων μπορεί να διαχειριστεί μεγαλύτερο όγκο δεδομένων ευκολότερα και αποτελεσματικότερα, προκειμένου να λάβει στρατηγικές αποφάσεις.

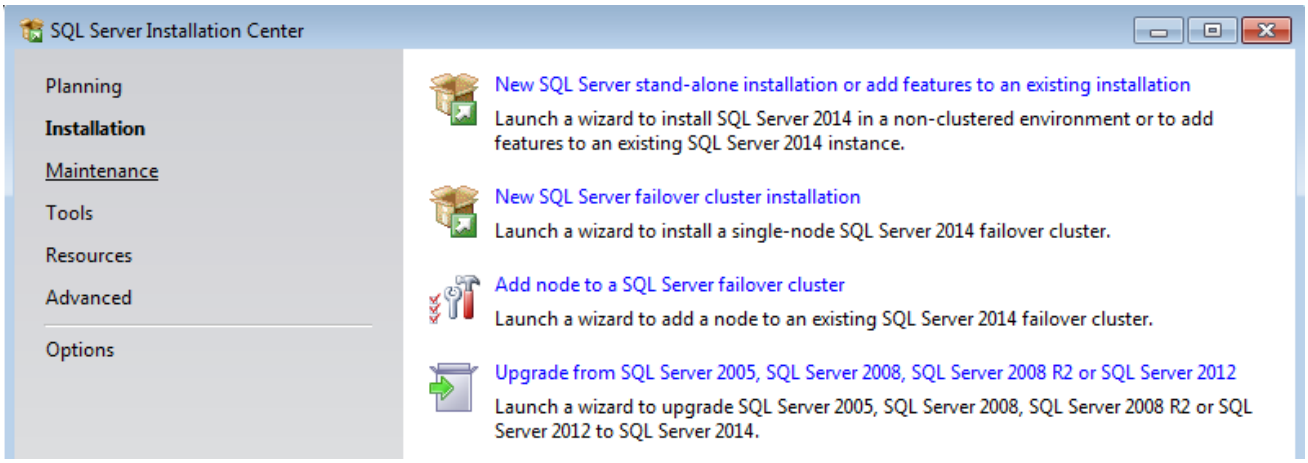
Σε αυτό το σημείο, θα ήταν παράλειψή μου να μην ευχαριστήσω τον κ. Α. Νανόπουλο (για την προ δεκαετίας αρχική έκδοση του εργαστηριακού οδηγού) και, ασφαλώς, όλα τα μέλη του Εργαστηρίου Τεχνολογίας και Επεξεργασίας Δεδομένων (Delab) του Τμήματος Πληροφορικής του Α.Π.Θ. που κατά καιρούς ανέλαβαν τη διδασκαλία του εργαστηριακού μέρους των μαθημάτων Βάσεων, Αποθηκών και Εξόρυξης Δεδομένων, συνεισφέροντας σε υλικό και ασκήσεις. Επιπροσθέτως, να ευχαριστήσω τον Ανδρέα Πάσχο για την τεχνική επεξεργασία του ηλεκτρονικού βιβλίου, καθώς και τον Φώτη Συμεωνίδη για τη φιλολογική επιμέλεια του βιβλίου, το εξώφυλλο και το βίντεο με οδηγίες χρήσης για το ebook. Επίσης, θέλω να ευχαριστήσω ιδιαίτερα τον συνάδελφο και φίλο μου Χρήστο Άνδρα που αφενός βοήθησε στην επικαιροποίηση των κεφαλαίων του πρώτου μέρους του βιβλίου, προκειμένου αυτά να προσαρμοστούν στο νέο περιβάλλον του SQL Server 2014, και αφετέρου παρείχε χρήσιμες παρατηρήσεις για την περαιτέρω βελτίωση του παρόντος πονήματος. Τέλος, επιθυμώ να ευχαριστήσω τους μεταπτυχιακούς φοιτητές Φοίβο Κολιόπουλο, Στέλλα Γκουτζιούρη, Στέλλα Μαυρομάτη και Ναούμ Τσιόπτσια που βοήθησαν στην επικαιροποίηση (από την προηγούμενη έκδοση) των κεφαλαίων του δεύτερου μέρους του βιβλίου, προκειμένου αυτά να προσαρμοστούν στο νέο περιβάλλον του SQL Server 2014.

Κεφάλαιο 1. Εγκατάσταση και Περιβάλλον του SQL Server 2014

Σύνοψη
Σ' αυτό το κεφάλαιο περιγράφονται τα βασικά βήματα εγκατάστασης του SQL Server. Επιπλέον, περιγράφεται ο έλεγχος που απαιτείται να γίνει, προκειμένου να βεβαιωθεί η καλή λειτουργία της εγκατάστασης. Τέλος, γίνεται μια σύντομη ξενάγηση στο βασικό γραφικό περιβάλλον του SQL Server Management Studio.

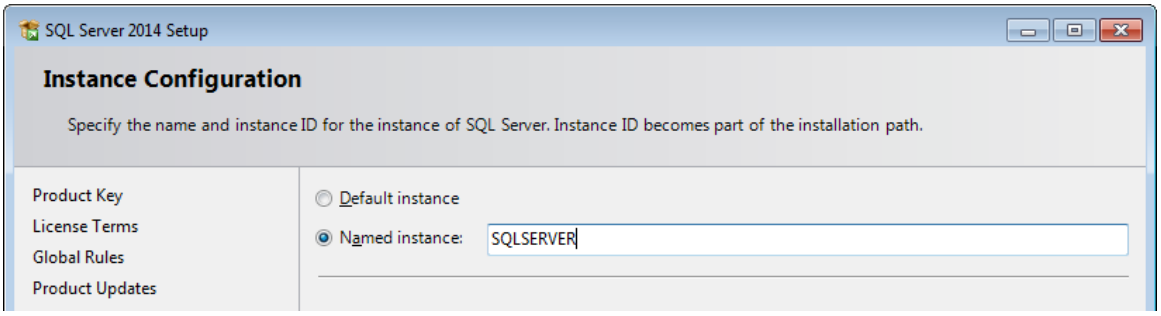
1.1. Εγκατάσταση του SQL Server

Ο SQL Server 2014 διατίθεται σε διαφορετικές εκδόσεις (Express, Standard, Enterprise), οι οποίες καλύπτουν είτε βασικές είτε πιο εξειδικευμένες ανάγκες μιας επιχείρησης αναφορικά με τη δημιουργία και διαχείριση βάσεων δεδομένων. Εμείς θα εγκαταστήσουμε τη Standard έκδοση. Εκτελώντας το αρχείο Setup.exe, εμφανίζεται το παράθυρο της Εικόνας 1.1, όπου εμφανίζονται διάφορες επιλογές. Εμείς επιλέγουμε New SQL Server stand-alone installation… για να εγκαταστήσουμε μία καινούρια εγκατάσταση.



Εικόνα 1.1

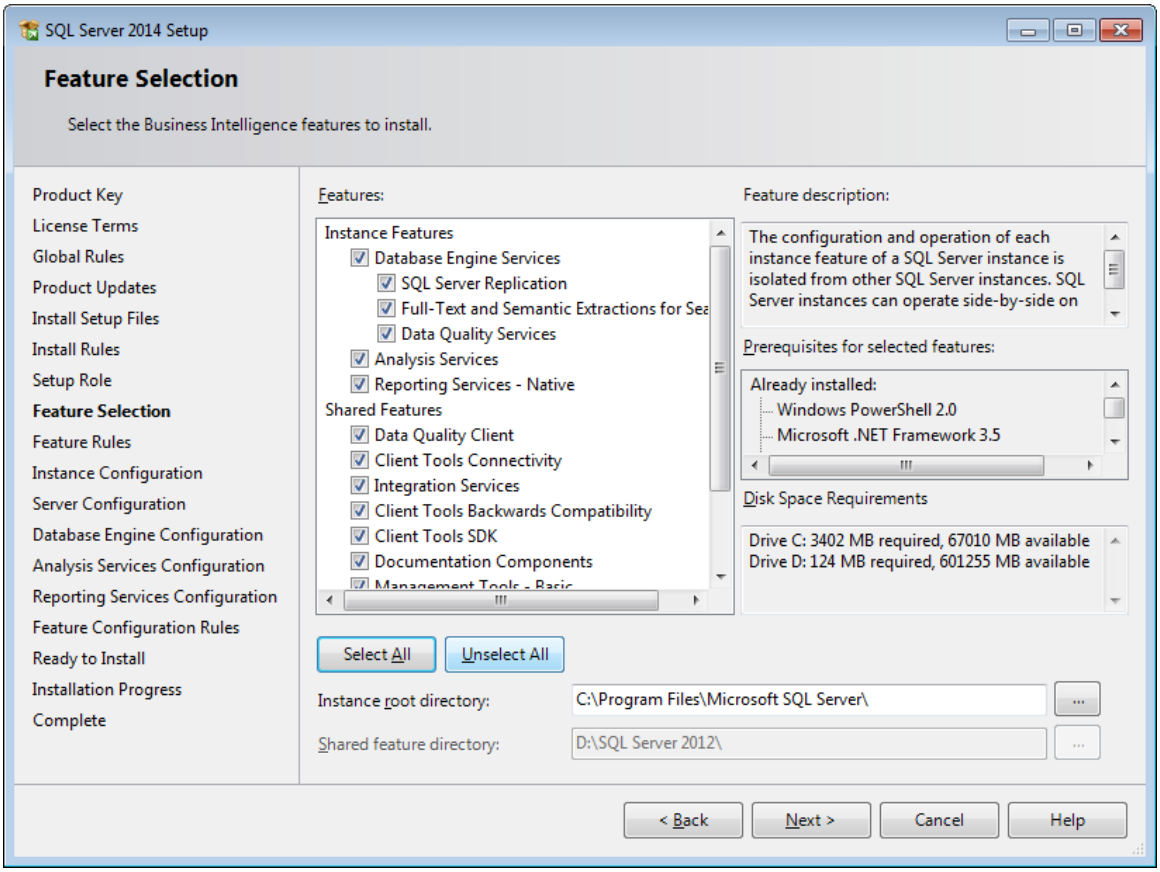
Καταρχήν εκτελούνται κάποιοι έλεγχοι για την ύπαρξη ή μη των ελάχιστων υπολογιστικών προδιαγραφών που πρέπει να έχει ο υπολογιστής στον οποίο γίνεται η εγκατάσταση. Σε επόμενο στάδιο μάς ζητείται το όνομα του instance της εγκατάστασης, όπως φαίνεται στην Εικόνα 1.2. Εφόσον επιλέξουμε «Named instance», θα πρέπει να δώσουμε ένα όνομα στην εγκατάσταση, το οποίο θα προστεθεί στο όνομα του υπολογιστή μας. Στην περίπτωσή μας, έστω ότι δίνουμε το όνομα «SQLSERVER». Συνεπώς, αν ο υπολογιστής μας ονομάζεται Chris, τότε το όνομα του SQL Server, στον οποίο μπορούμε να συνδεόμαστε, θα είναι «Chris\SQLSERVER». Εφόσον επιλέξουμε Default instance, τότε ο SQL Server χρησιμοποιεί ως όνομα μόνο αυτό του υπολογιστή, δηλαδή «Chris».



Εικόνα 1.2

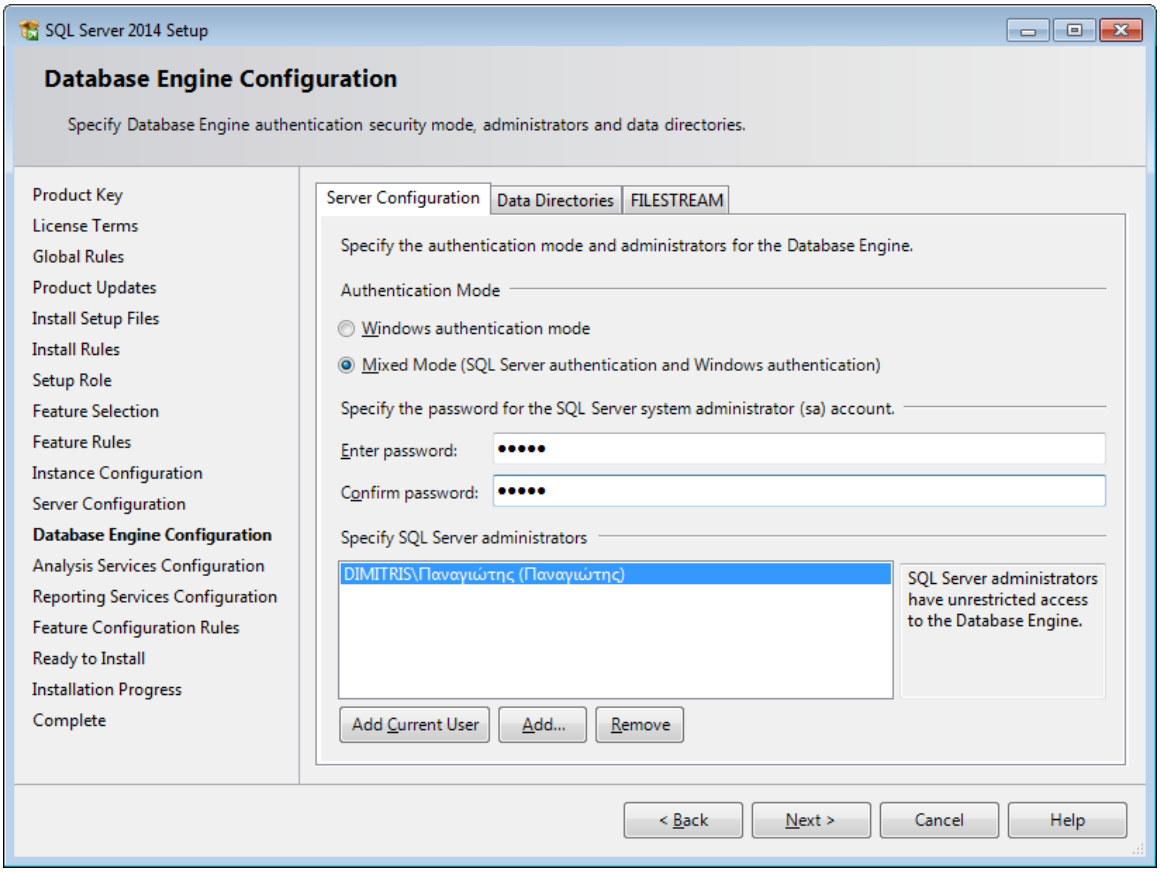
ΠΡΟΣΟΧΗ! Σε περίπτωση που αλλάξουμε το όνομα του υπολογιστή μας, θα πρέπει να αλλάξουμε και το όνομα του instance για τη σύνδεσή μας στον SQL Server.

Στο παράθυρο της Εικόνας 1.3 μπορούμε να επιλέξουμε ποια χαρακτηριστικά θέλουμε να εγκατασταθούν. Στο παράδειγμά μας, εμείς θα επιλέξουμε να εγκατασταθούν όλα τα χαρακτηριστικά του SQL Server.



Εικόνα 1.3

Στο παράθυρο της Εικόνας 1.4 επιλέγουμε τον τρόπο πρόσβασης στον SQL Server. Συγκεκριμένα, επιλέγουμε αν θα επιτρέπεται η πρόσβαση στον SQL Server με τους ίδιους κωδικούς που έχουμε για τα Windows (Windows authentication mode) ή αν θα προσδιορίσουμε και ένα ανεξάρτητο σύστημα πιστοποίησης του SQL Server (Mixed Mode authentication). Στη δεύτερη περίπτωση (η οποία προτείνεται , ως απαραίτητη για να συνδεόμαστε μέσω δικτύου στον SQL Server) υπάρχει ένας predefined default user με όνομα «sa», μέσω του οποίου προσδιορίζουμε σε αυτό το σημείο της εγκατάστασης το password που επιθυμούμε. Αργότερα μπορούμε να φτιάξουμε και άλλους χρήστες ή Logins ανάλογα με τις ανάγκες μας (Hoffer, Venkatarama, & Topi, 2013· Μανωλόπουλος, & Παπαδόπουλος, 2006). Όπως επίσης φαίνεται στην Εικόνα 1.4, ορίζεται από την εγκατάσταση ένας διαχειριστής με δικαιώματα πλήρους πρόσβασης στον SQL Server. Συγκεκριμένα, ο χρήστης των Windows που έχει ενεργοποιήσει την εγκατάσταση ονομάζεται Παναγιώτης, αλλά η εγκατάσταση γίνεται στον υπολογιστή DIMITRIS.



Εικόνα 1.4

Μετά από λίγο θα τερματιστεί η εγκατάσταση.

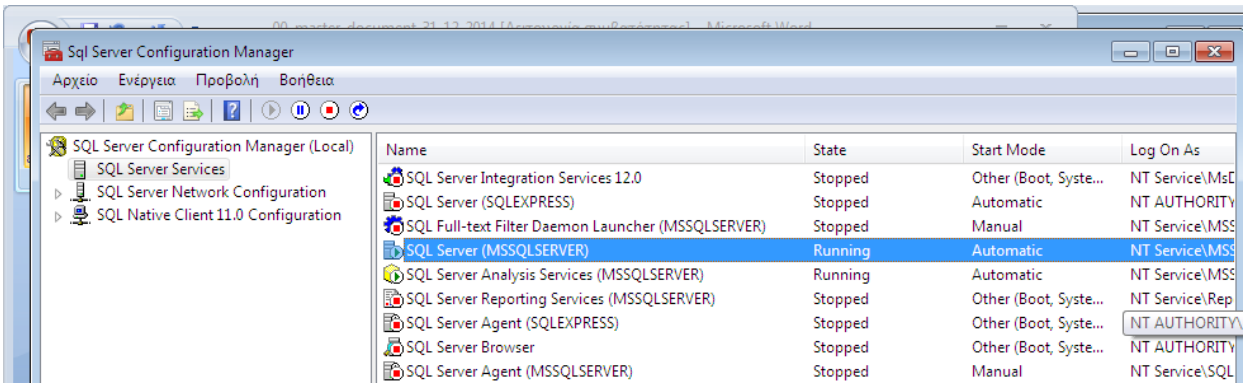
ΠΡΟΣΟΧΗ! Σε περίπτωση προβλήματος κατά την διάρκεια της εγκατάστασης, δεν επιχειρούμε να εκτελέσουμε νέες εγκαταστάσεις τη μία πάνω στην άλλη. Αποφεύγουμε το χειρωνακτικό σβήσιμο φακέλων. Ο ενδεδειγμένος τρόπος είναι η απεγκατάσταση οποιουδήποτε προηγούμενου ίχνους μιας προβληματικής εγκατάστασης του SQL Server μέσω της επιλογής προσθαφαίρεσης προγραμμάτων .

1.2. Έλεγχος καλής λειτουργίας της εγκατάστασης

Το γεγονός ότι εγκαταστάθηκε με επιτυχία ο SQL Server δεν σημαίνει ότι θα συνδεόμαστε σ' αυτόν πάντα χωρίς προβλήματα. Για παράδειγμα, στην περίπτωση που το service του SQL Server δεν τρέχει στο background του υπολογιστή μας, θα προσπαθούμε μάταια να συνδεθούμε σε αυτόν. Μπορούμε να ελέγχουμε τις παραμέτρους λειτουργίας των υπηρεσιών (services) του SQL Server, επιλέγοντας SQL Server Configuration Manager από το μενού, όπως φαίνεται στην Εικόνα 1.5.

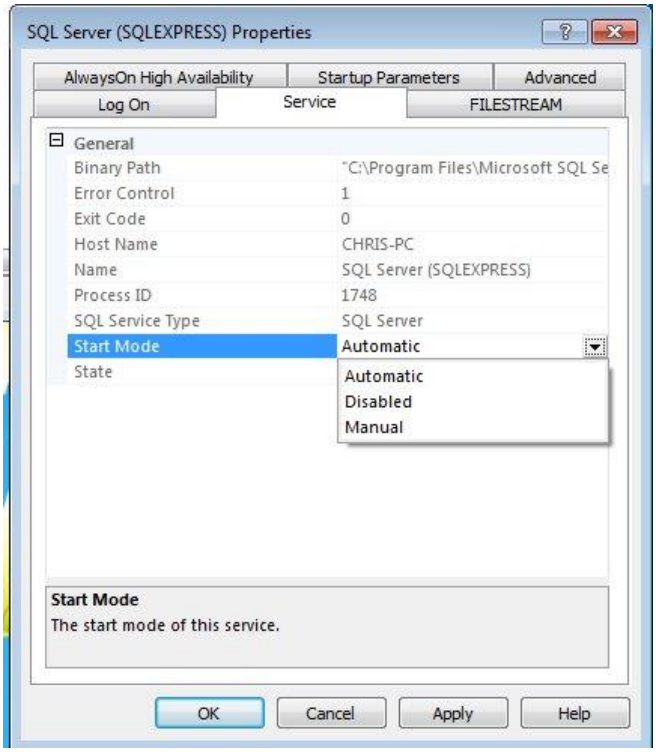
Εικόνα 1.5

Το πρώτο και βασικό service που πρέπει να έχει κατάσταση (state) running είναι αυτό του SQL Server. Διαφορετικά, δεν θα μπορούμε να συνδεθούμε στη βάση δεδομένων μας ή να αποθηκεύουμε δεδομένα ή να εκτελούμε ερωτήματα σε αυτήν. Εξάλλου, είναι το πρώτο που ελέγχουμε, όπως φαίνεται στην Εικόνα 1.6. Το δεύτερο σημαντικότερο service είναι αυτό του SQL Server Analysis Services, το οποίο είναι υπεύθυνο για την λειτουργία του Data Mining Tools και Business Intelligence που αναφέρονται στα κεφάλαια 6-11.



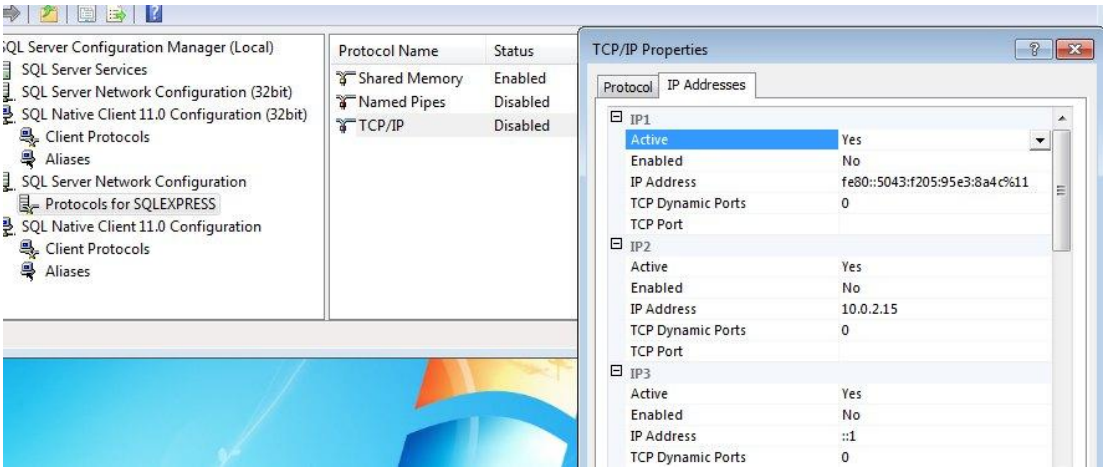
Εικόνα 1.6

Μπορούμε, ανάλογα με τις ανάγκες μας, να ρυθμίσουμε, ώστε κάθε υπηρεσία να ξεκινά (σε σχέση με τον υπολογιστή) αυτόματα ή χειρωνακτικά. Επιπρόσθετα, υπάρχει η επιλογή της πλήρους απενεργοποίησης, η οποία είναι χρήσιμη, εφόσον δεν σκοπεύουμε να χρησιμοποιήσουμε τον SQL Server για μεγάλο χρονικό διάστημα και, επομένως, δεν θέλουμε να δεσμεύουμε τη μνήμη του υπολογιστή μας με ένα service του. Όπως φαίνεται στην Εικόνα 1.7, με δεξί κλικ πάνω σε μια υπηρεσία και μετά κλικ στο properties μπορούμε να αλλάξουμε τις ρυθμίσεις της.



Εικόνα 1.7

Τέλος, υπάρχουν πιο σύνθετες ρυθμίσεις για τα πρωτόκολλα, μέσω των οποίων μπορούν να συνδέονται εφαρμογές στα δεδομένα του SQL Server. Αυτές μπορούν να ενεργοποιηθούν ή όχι, ορίζοντας τις ανάλογες πόρτες TCP, όπως στο παράδειγμα της Εικόνας 1.8.

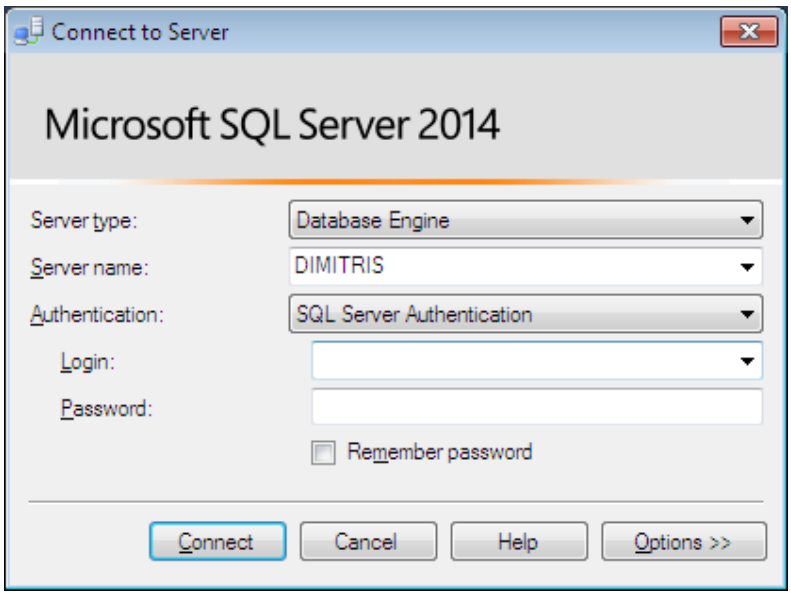


Εικόνα 1.8

1.3. Το περιβάλλον του SQL Server Management Studio

Το πρόγραμμα αυτό αποτελεί ένα γραφικό περιβάλλον σύνδεσης και διαχείρισης του SQL Server. Επισημαίνεται ότι δεν αφορά μόνο τη σύνδεση με τον SQL Server που εγκαταστάθηκε στον υπολογιστή σας αλλά και οποιονδήποτε άλλον απομακρυσμένο SQL Server, είτε μέσω τοπικού δικτύου είτε μέσω διαδικτύου, εφόσον έχουμε τις πληροφορίες σύνδεσης (π.χ. connection string, username, password).

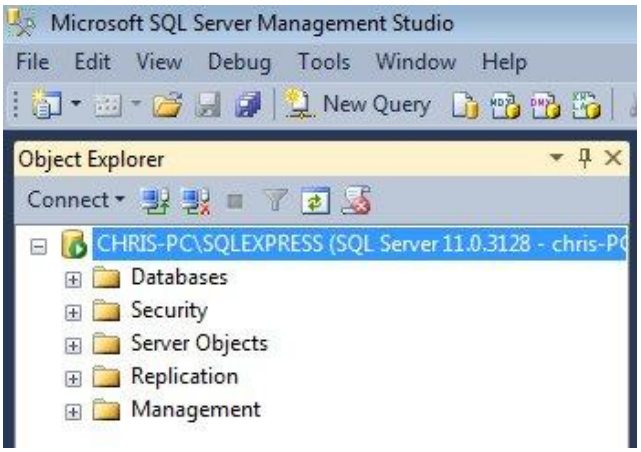
Από το μενού του SQLServer, στα προγράμματα θα βρούμε την επιλογή Start → Programs → Microsoft SQL Server 2014 → SQL Server Management Studio. Κάνοντας κλικ θα ξεκινήσει η εφαρμογή. Η πρώτη οθόνη ζητά να συνδεθούμε σε κάποιο instance, όπως φαίνεται στην Εικόνα 1.9.



Εικόνα 1.9

Προκειμένου να συνδεθούμε, αρκεί να επιλέξουμε το Servername και το Authentication mode. Στην περίπτωση που επιλέξουμε ως τρόπο πρόσβασης το SQL Server authentication mode, θα πρέπει να εισάγουμε username και password, όπως φαίνεται στην Εικόνα 1.9. Διαφορετικά, θα μπορούσαμε να επιλέξουμε την επιλογή Windows Authentication, στην οποία χρησιμοποιούνται αυτόματα τα στοιχεία που έχει δηλώσει ο χρήστης ως username και password στο λειτουργικό των Windows.

Εφόσον συνδεθούμε επιτυχώς, βλέπουμε σε δενδροειδή μορφή επιλογές διαχείρισης του SQLServer, όπως φαίνεται στην Εικόνα 1.10. Η βασική επιλογή με την οποία θα ασχοληθούμε στο βιβλίο μας είναι η επιλογή **Databases**. Σ' αυτόν τον φάκελο μπορούμε να διαχειριστούμε μία ή περισσότερες βάσεις δεδομένων και τα εμφανιζόμενα αντικείμενά τους, κάνοντας κλικ πάνω σε μία βάση δεδομένων στον φάκελο databases.



Εικόνα 1.1

1.4. Ασκήσεις

1. Να αναζητήσετε από το διαδίκτυο πληροφορίες για τις διαφορετικές εκδόσεις του SQL Server 2014 (Express, Standard, Enterprise). Να συγκρίνετε τις διαφορετικές εκδόσεις του SQL Server 2014, δημιουργώντας έναν πίνακα σύγκρισης με τα χαρακτηριστικά που υποστηρίζει (ή δεν υποστηρίζει) η κάθε έκδοση.

2. Να κατεβάσετε από το διαδίκτυο την έκδοση Sql Server 2014 Express και να την εγκαταστήσετε στον υπολογιστή σας, προκειμένου να έχετε πρόσβαση στο περιβάλλον του Management Studio.

3. Κατά την εγκατάσταση του SQL Server να επλέξετε τρόπο πρόσβασης στο περιβάλλον του είτε μέσω του windows authentication mode είτε μέσω του mixed mode. Να περιγράψετε πότε και γιατί θα πρέπει να επιλέγεται η κάθε κατάσταση.

4. Να κατεβάσετε από το διαδίκτυο την έκδοση Data Tools 2013 του Visual Studιo και να την εγκαταστήσετε στον υπολογιστή σας, προκειμένου να έχετε πρόσβαση στο περιβάλλον του Business Intelligence και Analysis Services.

5. Να εφαρμόσετε τα βήματα για τον τρόπο ελέγχου της καλής εγκατάστασης και λειτουργίας του SQL Server, προκειμένου να μπορείτε να έχετε πρόσβαση στις υπηρεσίες που προσφέρονται στο περιβάλλον του Management Studio και του Business Intelligence.

1.5. Βιβλιογραφία/Αναφορές

Hoffer, J. A., Venkatarama, R., & Topi, H. (2013). *Modern Database Management*, Prentice Hall.

Μανωλόπουλος, Ι., & Παπαδόπουλος, Α. Ν. (2006). *Συστήματα Βάσεων Δεδομένων: Θεωρία & Πρακτική Εφαρμογή*, Αθήνα, Εκδόσεις Νέων Τεχνολογιών.

Κεφάλαιο 2. Δημιουργία Βάσης Δεδομένων και Πινάκων

Σύνοψη
Σ' αυτό το κεφάλαιο θα δημιουργήσουμε μια βάση δεδομένων που αφορά ένα κατάστημα ενοικίασης ψηφιακών δίσκων με το όνομα DVDClub. Θα εργαστούμε, κυρίως, με εντολές από την γλώσσα ορισμού δεδομένων (Data Definition Language). Θα συζητήσουμε για τους διαφορετικούς τύπους δεδομένων και τον τρόπο δημιουργίας πινάκων, με γραφικό τρόπο και με κώδικα. Τέλος, θα δημιουργήσουμε συσχετίσεις μεταξύ των πινάκων στο Database diagram και θα εισάγουμε τιμές στους πίνακες.

2.1. Ορισμός και Δημιουργία μιας Βάσης Δεδομένων

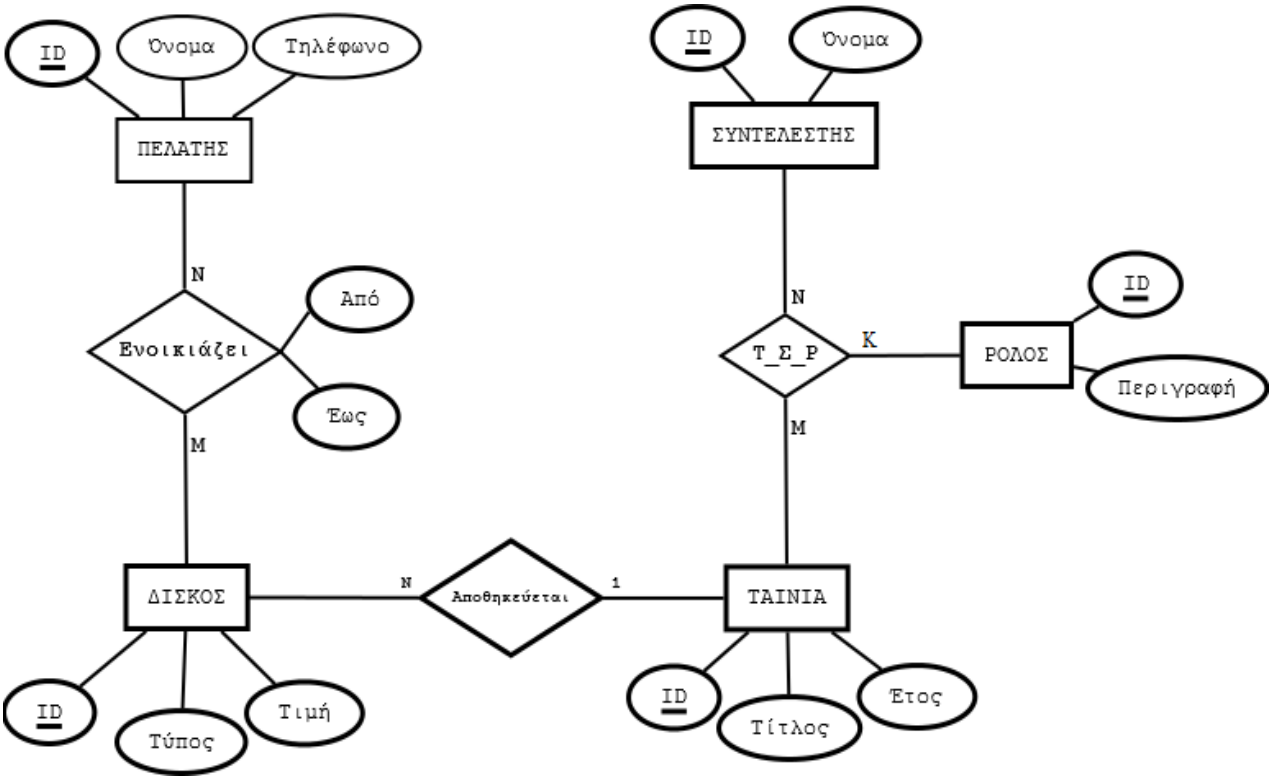
Ο SQL Server είναι ένα σύστημα διαχείρισης βάσεων δεδομένων (database management system). Ως εκ τούτου, υποστηρίζει όλες τις λειτουργίες που πρέπει να προσφέρει μια βάση δεδομένων, όπως είναι η αναζήτηση, η εισαγωγή, η διαγραφή, η ενημέρωση εγγραφών κλπ. Η βάση δεδομένων είναι μια συλλογή στοιχείων που σχετίζονται μεταξύ τους και είναι καταχωρημένα με κατάλληλα δομημένο τρόπο. Συγκεκριμένα, μια βάση δεδομένων χαρακτηρίζεται ως σχεσιακή, όταν το βασικό δομικό στοιχείο της είναι η σχέση ή, αλλιώς, ο πίνακας, ο οποίος διέπεται από συγκεκριμένες ιδιότητες (μοναδικότητα κάθε εγγραφής, ατομικότητα τιμών κτλ.). Ο SQL Server παρέχει δύο μεθόδους για να δημιουργήσετε μια βάση δεδομένων:

- το SQL Server Management Studio (Γραφικό περιβάλλον),
- εντολές της SQL.

Στη συνέχεια, χρησιμοποιώντας και τις δύο μεθόδους, θα δημιουργήσουμε τη βάση δεδομένων και τους πίνακες ενός DVDclub που θα διέπεται από τους παρακάτω κανόνες της ανάλυσης απαιτήσεων:

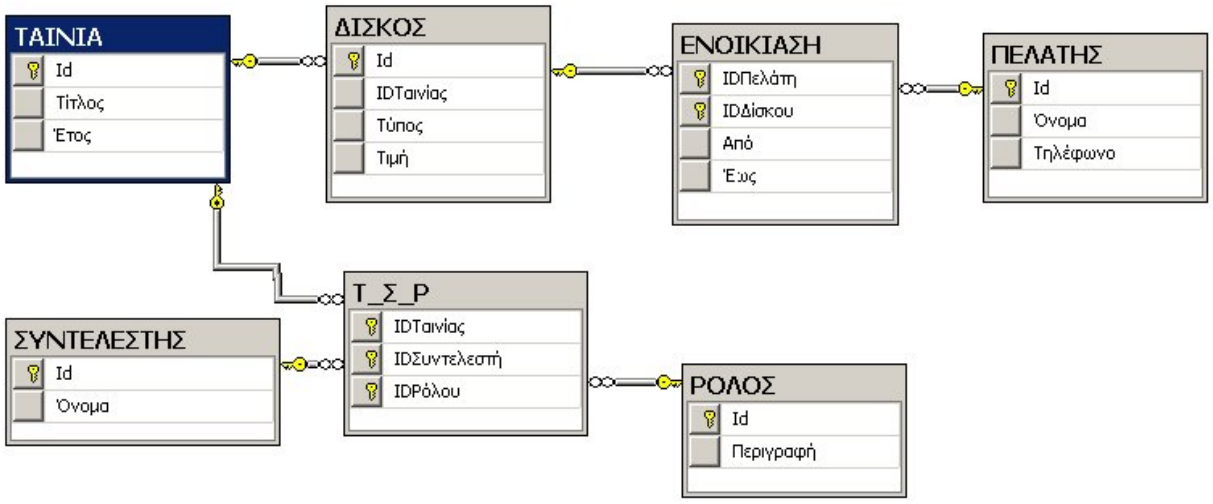
- Το DVDclub έχει ένα σύνολο από πελάτες για τους οποίους καταχωρεί ένα μοναδικό κωδικό, το όνομά τους και το τηλέφωνό τους.
- Ο κάθε πελάτης μπορεί να νοικιάσει ταινίες που αποθηκεύονται σε δίσκους dvd από μία ημερομηνία μέχρι μία άλλη ημερομηνία.
- Η κάθε ταινία χαρακτηρίζεται από ένα μοναδικό κωδικό, τον τίτλο της ταινίας που αντιστοιχεί και το έτος το οποίο γυρίστηκε αυτή η ταινία.
- Ο κάθε δίσκος dvd χαρακτηρίζεται από ένα μοναδικό κωδικό, τον τύπο του (π.χ. Blu-ray) και την τιμή ενοικίασής του. Μπορούν να υπάρχουν πολλά αντίγραφα δίσκου dvd για την ίδια ταινία, τα οποία διακρίνονται από τον κωδικό τους. Για απλότητα, θεωρούμε ότι ένα αντίγραφο δίσκου dvd μπορεί να ενοικιαστεί από κάποιον πελάτη μία μόνο φορά.
- Σε κάθε ταινία μετέχει ένα σύνολο από συντελεστές, οι οποίοι χαρακτηρίζονται από ένα μοναδικό κωδικό και το όνομά τους.
- Ο κάθε συντελεστής μπορεί να έχει παραπάνω από έναν ρόλους σε κάθε ταινία, π.χ. σκηνοθέτης, ηθοποιός κ.ο.κ.
- Για κάθε ρόλο ενός συντελεστή υπάρχει ένας μοναδικός κωδικός, καθώς και η περιγραφή του.

Βάσει της παραπάνω ανάλυσης απαιτήσεων, ακολουθεί το **διάγραμμα οντοτήτων-συσχετίσεων (E-R diagram)** που δίνεται στην Εικόνα 2.1.



Εικόνα 2.1 Διάγραμμα Οντοτήτων – Συσχετίσεων του DVD club

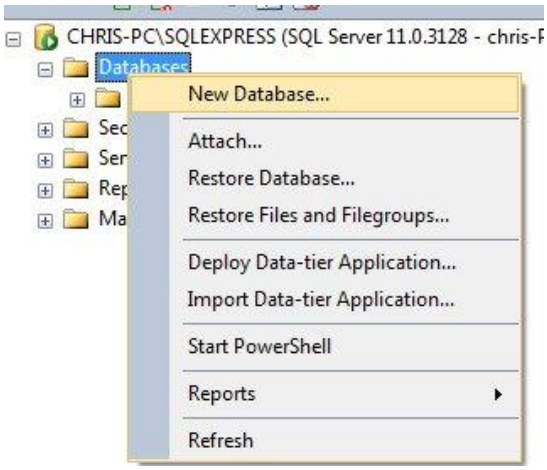
Συνολικά, το διάγραμμα Οντοτήτων-Συσχετίσεων (Ο-Σ) αποτελείται από πέντε οντότητες (ΠΕΛΑΤΗΣ, ΔΙΣΚΟΣ, ΤΑΙΝΙΑ, ΣΥΝΤΕΛΕΣΤΗΣ, ΡΟΛΟΣ) και τρεις συσχετίσεις (Ενοικιάζει, Αποθηκεύεται, Τ_Σ_Ρ). Από αυτές τις συσχετίσεις, οι δύο (Ενοικιάζει και Τ_Σ_Ρ) συσχετίζουν οντότητες με συσχέτιση πολλά προς πολλά και, ως εκ τούτου, πρέπει να αναχθούν σε συσχετιστικές οντότητες στο σχεσιακό σχήμα, με αποτέλεσμα να προκύπτουν συνολικά επτά πίνακες δεδομένων, όπως φαίνεται στην Εικόνα 2.2. Σύμφωνα με τη θεωρία των Βάσεων Δεδομένων, το E-R διάγραμμα της Εικόνας 2.1 αντιστοιχίζεται με το σχεσιακό σχήμα της Εικόνας 2.2, που αποτελείται από επτά σχέσεις: μία για κάθε οντότητα και από μία για τις συσχετίσεις M:N και M:N:K. Η συσχέτιση 1:N ενσωματώνεται στη σχέση που προκύπτει από την οντότητα ΔΙΣΚΟΣ. Τέλος, τονίζουμε ότι στις επόμενες ενότητες θα παρουσιάσουμε βήμα-βήμα την δημιουργία της βάσης δεδομένων DVDclub βάσει του σχεσιακού σχήματος της Εικόνας 2.2.



Εικόνα 2.2 Σχεσιακό Σχήμα του DVDclub

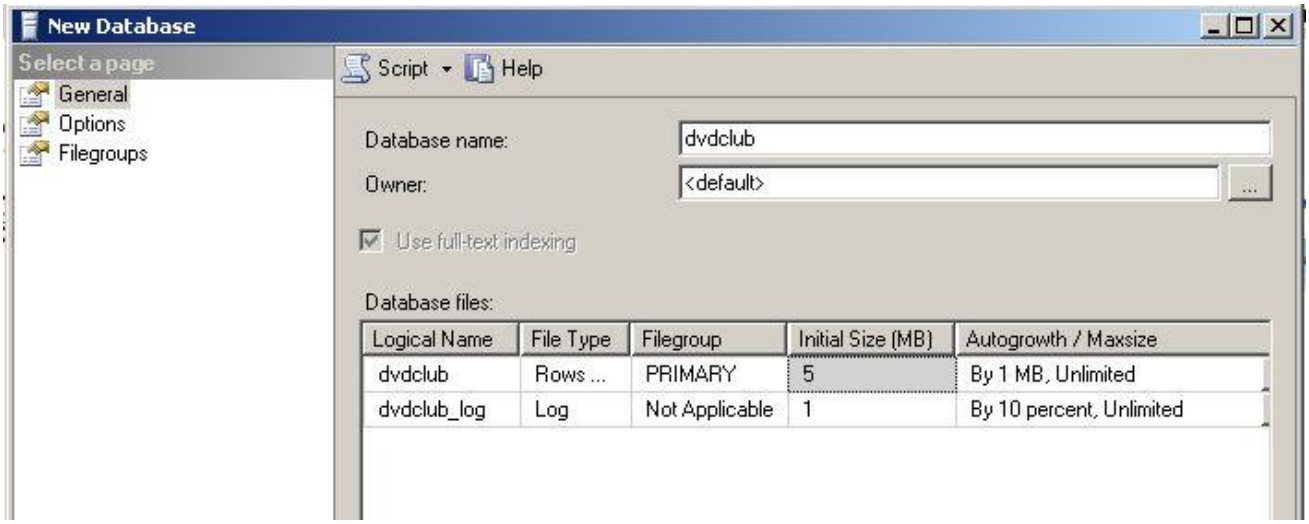
2.1.1. Δημιουργία νέας βάσης σε γραφικό περιβάλλον

Με δεξί κλικ πάνω στο φάκελο Databases επιλέγουμε “New Database…”.



Εικόνα 2.3

Δίνουμε το όνομα της βάσης δεδομένων μας (DVDclub), όπως φαίνεται στην Εικόνα 2.4.

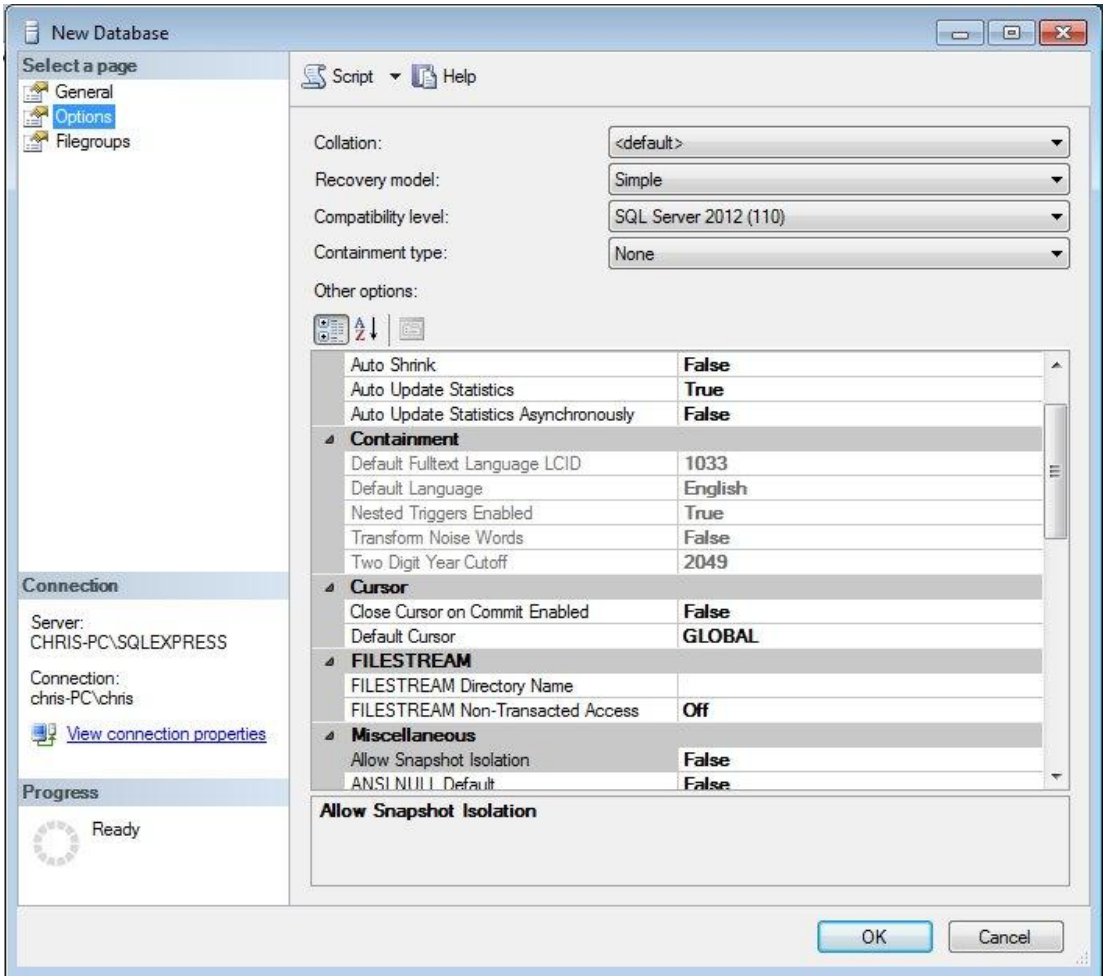


Εικόνα 2.4

Δημιουργούνται δύο αρχεία: το primary και το transaction log. Το κύριο αρχείο δεδομένων έχει επέκταση .mdf ενώ το transaction log έχει την επέκταση .ldf. Τονίζεται ότι και τα δύο αρχεία δημιουργούνται αυτόματα και παίρνουν το όνομα της βάσης δεδομένων ως πρόθεμα. Μπορούμε να αποδεχθούμε το όνομα ή να πληκτρολογήσουμε ένα διαφορετικό. Στο κύριο αρχείο αποθηκεύονται τα δεδομένα της βάσης δεδομένων, ενώ στο transaction log file τηρούνται οι τελευταίες μεταβολές που έγιναν στην βάση δεδομένων, προκειμένου να δοθεί η δυνατότητα επαναφοράς της σε περίπτωση βλάβης του συστήματος. Κάνοντας κλικ στην επιλογή «Options» εμφανίζεται η Εικόνα 2.4, οπότε μπορούμε να καθορίσουμε τις πιο εξειδικευμένες ρυθμίσεις που εμφανίζονται στην Εικόνα 2.5.

Οι πιο σημαντικές είναι οι παρακάτω:

Compatibility level: Αν θέλουμε να μεταφέρουμε την βάση δεδομένων που θα φτιάξουμε ή αν θέλουμε να είναι προσβάσιμη από άλλες εφαρμογές που δεν έχουν την δυνατότητα να συνδεθούν στο SQL Server 2014, τότε μπορούμε να επιλέξουμε συμβατότητα με προηγούμενες εκδόσεις (Compatibility level).

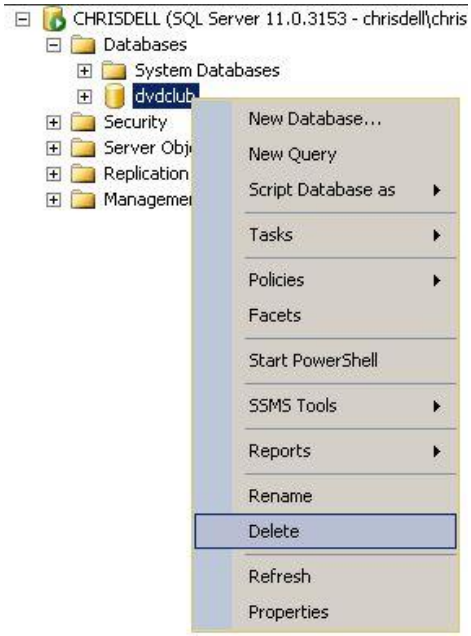


Εικόνα 2.5

Recovery Model: Η ρύθμιση αφορά τον τύπο των αντιγράφων ασφαλείας. Όπως φαίνεται στην Εικόνα 2.6, το μοντέλο Simple ελαχιστοποιεί το transactions log file και αποθηκεύει τα δεδομένα απευθείας στο primary file. Έτσι, δεν δίνεται η δυνατότητα να επιστρέψουμε σε μια προηγούμενη χρονική στιγμή της βάσης δεδομένων, παρά μόνο στην τελευταία. Αντιθέτως, το μοντέλο Full δίνει την δυνατότητα να επιστρέψουμε σε οποιαδήποτε χρονική στιγμή της βάσης δεδομένων μας. Βέβαια, το μοντέλο αυτό απαιτεί να τηρούνται και τα ανάλογα back-ups του log file και καταλαμβάνει περισσότερο χώρο στο δίσκο.

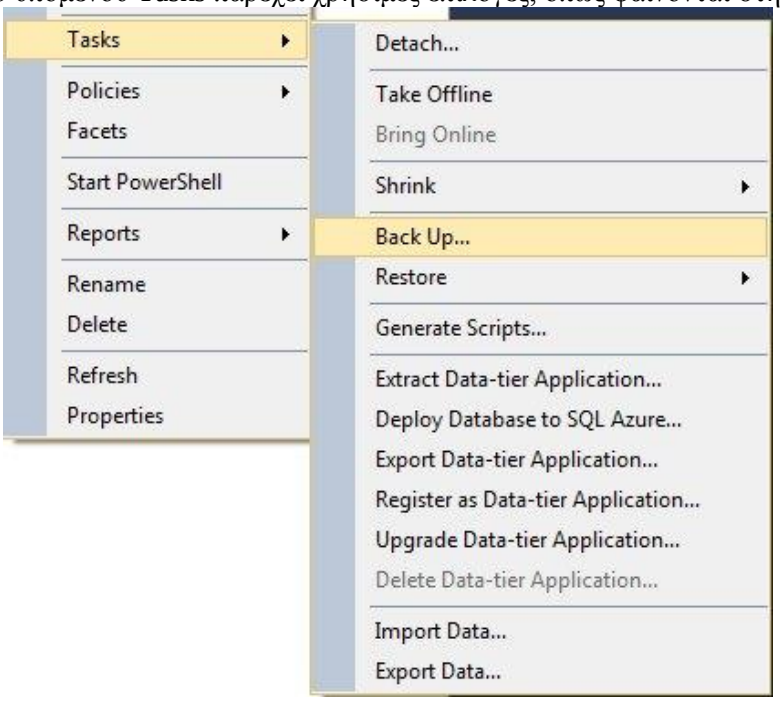
Εικόνα 2.6

Για να αποθηκευτεί η βάση δεδομένων μας (DVDclub), κάνουμε κλικ στο κουμπί OK (βλέπε Εικόνα 2.4). Τώρα, κάνοντας δεξί κλικ πάνω στη βάση δεδομένων μας, βλέπουμε τις διαθέσιμες ενέργειες που εμφανίζονται στην Εικόνα 2.7, όπως π.χ. η διαγραφή της ΒΔ με την επιλογή Delete.



Εικόνα 2.7

Στην επιλογή «Properties» μπορούμε να αλλάξουμε τις βασικές ρυθμίσεις που ορίσαμε κατά τη δημιουργία της. Ακόμη, το υπομενού Tasks παρέχει χρήσιμες επιλογές, όπως φαίνονται στην Εικόνα 2.8.



Εικόνα 2.8

Πιο σημαντική είναι η διαδικασία του «Back Up», κατά την οποία θα επιλέξουμε το όνομα του αρχείου, προκειμένου να αποθηκεύεται κάθε φορά ένα αρχείο με κατάληξη «bak» που θα περιέχει όλα τα δεδομένα της βάσης μας. Μια άλλη χρήσιμη επιλογή είναι το "Restore", όταν θέλουμε να επαναφέρουμε μια βάση δεδομένων από ένα αρχείο .bak (backup).

Η επιλογή "detach" ουσιαστικά κάνει κατ' αναλογία την επιλογή cut όταν εργαζόμαστε με αρχεία. Δηλαδή το αρχείο mdf θα αποκοπεί και θα σβηστεί από το περιβάλλον εργασίας μας, με σκοπό να μεταφερθεί με την επιλογή Databases – Attach σε κάποιον άλλο υπολογιστή. Τέλος, η επιλογή Shrink επιχειρεί να μειώσει τον αποθηκευτικό χώρο που καταλαμβάνει μία βάση δεδομένων, χωρίς απώλεια δεδομένων.

2.1.2. Δημιουργία νέας βάσης με κώδικα SQL

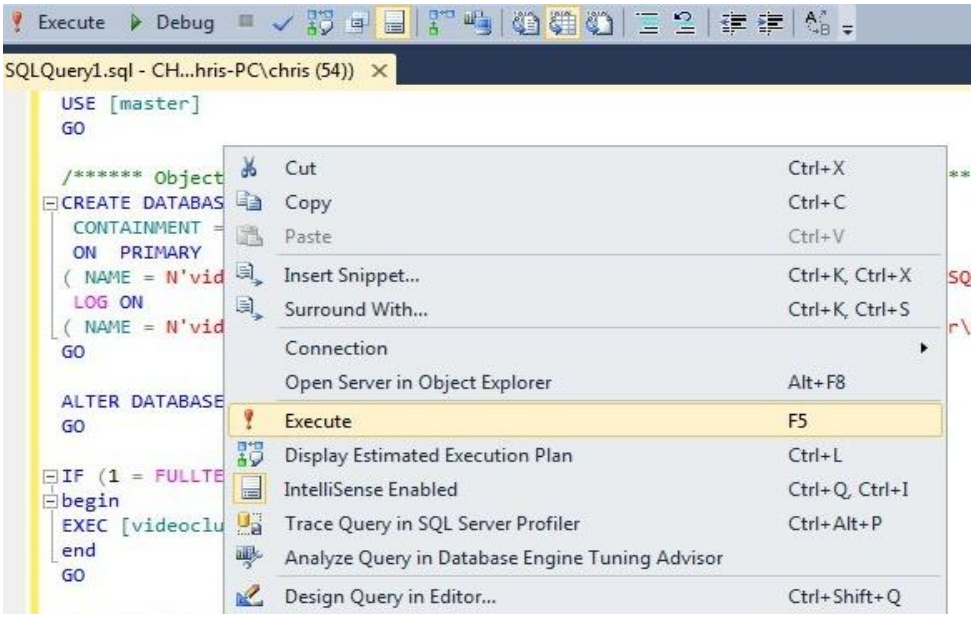
Εναλλακτικά, η δημιουργία της βάσης δεδομένων μπορεί να επιτευχθεί με τη χρήση της γλώσσας SQL. Κάνοντας κλικ στην εργαλειοθήκη στο κουμπί «New Query» επιλέγουμε να δημιουργήσουμε ένα νέο ερώτημα SQL, το οποίο θα εκτελεστεί στη βάση δεδομένων master του συστήματος. Στη συνέχεια, θα πληκτρολογήσουμε στον Query Editor την εντολή: **create database dvdclub**. Αφού ολοκληρώσουμε τη σύνταξη του ερωτήματος δημιουργίας βάσης, θα εκτελέσουμε τον κώδικα SQL είτε κάνοντας κλικ στο κουμπί "Execute" είτε πατώντας F5 στο πληκτρολόγιο, όπως φαίνεται στην Εικόνα 2.10.

Αφού δημιουργήσουμε τη βάση δεδομένων, επιλέγουμε Generate Scripts από το αναδυόμενο μενού της Εικόνας 2.8 και, έτσι, φανερώνεται ένας οδηγός όπου αυτόματα παράγονται οι ανάλογες εντολές SQL, οι οποίες μπορούν να δημιουργήσουν τα υπόλοιπα στοιχεία της βάσης που φτιάξαμε με γραφικό τρόπο, όπως φαίνεται στην Εικόνα 2.9.

SQLQuery1.sql - CHRIS...chrisdell\chris (54))

```
USE [master]
GO
/****** Object:  Database [dvdclub]    Script Date: 10/12/2014 17:33:22 ******/
CREATE DATABASE [dvdclub]
 CONTAINMENT = NONE
 ON  PRIMARY
( NAME = N'dvdclub', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLS
 LOG ON
( NAME = N'dvdclub_log', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL11.MS
GO
ALTER DATABASE [dvdclub] SET COMPATIBILITY_LEVEL = 110
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [dvdclub].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO
ALTER DATABASE [dvdclub] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [dvdclub] SET ANSI NULLS OFF
```

Εικόνα 2.9



Εικόνα 2.10

ΠΡΟΣΟΧΗ! Το Management studio πρέπει να αναγνωρίζει σε ποια βάση δεδομένων θα εκτελέσει το κάθε ερώτημα. Γι αυτό, πρέπει, μετά τη δημιουργία της βάσης, είτε να δηλώσουμε στην αρχή του ερωτήματος ότι χρησιμοποιούμε/ εργαζόμαστε στην βάση δεδομένων DVDclub (με τη χρήση της δήλωσης «use DVDclub») είτε να δηλώνουμε κάθε φορά την πλήρη διαδρομή ονόματος του κάθε πίνακα (π.χ DVDclub.dbo.ΔΙΣΚΟΣ). Σημειώνεται ότι η δήλωση dbo είναι μία προκαθορισμένη συντόμευση του Management Studio για τον ιδιοκτήτη της βάσης δεδομένων (DataBase Owner), ο οποίος έχει και τα πλήρη δικαιώματα διαχείρισης της υπό εξέτασης βάσης δεδομένων.

2.1.3. Διαγραφή μίας βάσης με κώδικα SQL

Στην περίπτωση που θέλουμε να διαγράψουμε μία βάση δεδομένων, μπορούμε να εκτελέσουμε το ερώτημα: **drop database dvdclub** ή να επιλέξουμε την αντίστοιχη εντολή από το menu, κάνοντας δεξί κλικ στο όνομα της βάσης δεδομένων.

2.2. Βασικές Έννοιες και Δημιουργία Πινάκων

Οι σχεσιακές βάσεις δεδομένων χρησιμοποιούν τις σχέσεις ή, αλλιώς, τους πίνακες για την αναπαράσταση των δεδομένων τους (Hoffer, Venkatarama, & Topi, 2013· Μανωλόπουλος, & Παπαδόπουλος, 2006). Ο κάθε πίνακας έχει ένα μοναδικό όνομα και προσδιορίζεται από ένα σύνολο γραμμών και στηλών. Κάθε γραμμή ενός πίνακα αναπαριστά μια εγγραφή (record) δεδομένων. Οι στήλες του πίνακα ορίζουν τα χαρακτηριστικά της κάθε εγγραφής. Για κάθε χαρακτηριστικό υπάρχει ένα σύνολο επιτρεπτών τιμών, το οποίο καλείται «πεδίο ορισμού του χαρακτηριστικού». Για τον πλήρη προσδιορισμό του πεδίου ορισμού ενός χαρακτηριστικού είναι απαραίτητο να γνωρίζουμε τον τύπο δεδομένων του (data type) και τη μορφοποίηση του. Οι βασικοί τύποι δεδομένων του SQL Server αναλύονται στη επόμενη υποενότητα.

2.2.1. Τύποι Δεδομένων

Στον SQL Server κάθε στήλη ενός πίνακα σχετίζεται με ένα τύπο δεδομένων, ο οποίος αποτελεί ένα χαρακτηριστικό που προσδιορίζει το είδος των δεδομένων (integer, character, date κτλ.). Ο Πίνακας 2.1 παρέχει τις περιγραφές των βασικών κατηγοριών των τύπων δεδομένων που υποστηρίζει ο SQL Server και τις περιγραφές των βασικών τύπων δεδομένων που περιέχει κάθε κατηγορία:

Κατηγορία	Περιγραφή	Τύπος δεδομένων	Περιγραφή
Character	Τα δεδομένα χαρακτήρων αποτελούνται από οποιοδήποτε συνδυασμό γραμμάτων, συμβόλων και αριθμητικών δεδομένων. Για παράδειγμα, έγκυρα δεδομένα χαρακτήρων είναι οι συνδυασμοί "John123" και "(0$%b99"	***Char***	Τα δεδομένα έχουν το ίδιο σταθερό μήκος (μέχρι 8KB)
		varchar	Τα δεδομένα μπορούν να ποικίλουν στον αριθμό των χαρακτήρων, αλλά το μήκος δεν μπορεί να υπερβαίνει τα 8KB
Unicode	Περιλαμβάνει όλους τους χαρακτήρες που ορίζονται στα διάφορα σύνολα χαρακτήρων. Οι Unicode τύποι δεδομένων καταλαμβάνουν διπλάσιο χώρο αποθήκευσης συγκριτικά με τους μη Unicode τύπους δεδομένων.	***nchar***	Τα δεδομένα πρέπει να έχουν το ίδιο σταθερό μήκος (μέχρι 4000 Unicode χαρακτήρες)
		nvarchar	Τα δεδομένα μπορούν να ποικίλουν στον αριθμό των Unicode χαρακτήρων τους (μέχρι 4000 Unicode χαρακτήρες)
Date and time	Τα δεδομένα ημερομηνίας και ώρας αποτελούνται από συνδυασμούς ημερομηνίας και ώρας.	***datetime***	Τα δεδομένα κυμαίνονται από 1η Ιανουαρίου 1753 μέχρι την 31η Δεκεμβρίου 9999. Συμπεριλαμβάνεται και η ώρα. Π.χ.: 2007-05-08 12:35:29
		smalldatetime	Τα δεδομένα κυμαίνονται από 1η Ιανουαρίου 1900 μέχρι την 6η Ιουνίου 2079 μαζί συμπεριλαμβάνεται και η ώρα.
		Date	Αποθηκεύεται μόνο η ημερομηνία χωρίς την ώρα. Π.χ.: 2007-05-08
Integer	Αποτελούνται από αρνητικούς και θετικούς αριθμούς, όπως είναι οι -15, 0, 5 και 2.507.	***bigint***	Τα δεδομένα είναι αριθμοί από -2^63 μέχρι 2^63-1 (8 bytes)
		int	Τα δεδομένα είναι αριθμοί από -2.147.483.648 μέχρι 2.147.483.647 (4 bytes)
		smallint	Τα δεδομένα είναι αριθμοί από -32.768 μέχρι 32.767 (2 bytes)
		tinyint	Τα δεδομένα είναι αριθμοί από 0 μέχρι 255 (1 byte)
Decimal	Δεκαδικός αριθμός με απόλυτη ακρίβεια.	***decimal - numeric***	Παράμετροί του είναι πόσα ψηφία θέλουμε πριν και μετά την υποδιαστολή. Π.χ. Decimal (9,2)
Floating point	Τα προσεγγιστικά αριθμητικά δεδομένα βασίζονται στην ακρίβεια που προσφέρει το δυαδικό αριθμητικό σύστημα	***float***	Τα δεδομένα είναι floating-point αριθμοί από -1,79E + 308 μέχρι 1,79E + 308
		real	Τα δεδομένα είναι floating-point αριθμοί από -3,40E + 308 μέχρι 3,40E + 308.

Πίνακας 2.1

2.2.2. Χρήσιμες συμβουλές για τους τύπους δεδομένων

Υπάρχουν πολλοί τύποι δεδομένων που παρουσιάζουν ελάχιστες διαφορές αλλά κάνουν την ίδια δουλειά. Ενδέχεται, σε πολύ εξειδικευμένο στάδιο, να προκαλέσουν προβλήματα ή ζητήματα απόδοσης. Δίνονται, λοιπόν, κάποιες χρήσιμες και πρακτικές συμβουλές που αφορούν τους τύπους δεδομένων:

- Εφόσον το μέγεθος ενός πεδίου συμβολοσειράς είναι μεταβλητό και όχι σταθερό (π.χ όλες οι καταχωρήσεις αποτελούνται από ένα κωδικό σταθερού μήκους 5 χαρακτήρων) προτιμήστε τους τύπους varchar, nvarchar και όχι char, nchar.

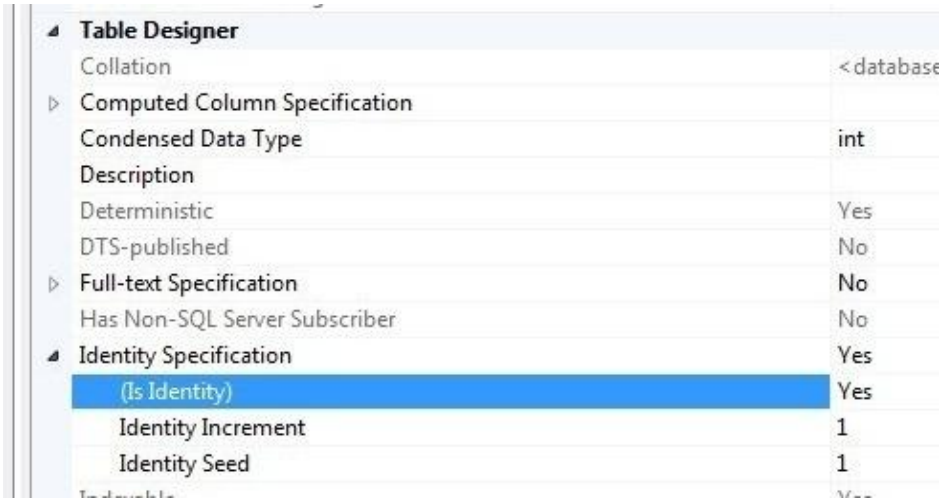
- Εφόσον θέλουμε η εφαρμογή μας να υποστηρίζει τη δυνατότητα του χρήστη να καταχωρεί Unicode χαρακτήρες (π.χ. γαλλικά ή ρωσικά ονόματα), πρέπει να χρησιμοποιήσουμε τον τύπο nvarchar και όχι varchar.

- Εφόσον θέλουμε να αποθηκεύσουμε μόνο την ημερομηνία μιας συναλλαγής, χρησιμοποιούμε τον τύπο Date και όχι τον τύπο Datetime, ο οποίος αποθηκεύει και την ώρα της συναλλαγής. Ο δεύτερος τύπος δεδομένων όχι μόνο σπαταλά αποθηκευτικό χώρο αλλά μπορεί και να δημιουργήσει προβλήματα σε ερωτήματα SQL. Αν για παράδειγμα, αποθηκεύσουμε στη βάση δεδομένων την ημερομηνία και την ακριβή ώρα στην οποία αγοράζεται ένα προϊόν με τη χρήση του τύπου δεδομένων Datetime (π.χ. '01/31/2012 13:47'), τότε, στην περίπτωση που θα θέλαμε να ανακτήσουμε όλες τις αγορές προϊόντων που έγιναν μέχρι και τις 31 Ιανουρίου του 2012, θα πρέπει να διαμορφώσουμε ένα ερώτημα με τέτοιο τρόπο ώστε να συμπεριλαμβάνεται ολόκληρη η υπό εξέταση μέρα, δηλαδή (<= '01/31/2012 23:59'). Διαφορετικά, αν το ερώτημα μας είχε απλά την μορφή (<= '01/31/2012'), τότε δεν θα εμφανίζονταν η αγορά που έγινε στις 31 Ιανουρίου του 2012, διότι το σύστημα εσφαλμένα θα εκλάμβανε το ερώτημα μας ως (<= '01/31/2012 00:00').

- Εφόσον ένα πεδίο χρησιμοποιείται σαν κύριο κλειδί ή σαν ξένο κλειδί (τμήμα συσχέτισης), καλό είναι να χρησιμοποιούμε τον αριθμητικό τύπο δεδομένων int και όχι smallint, tinyint.

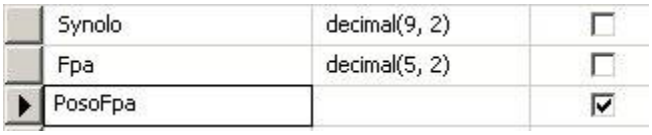
- Γενικότερα οι αριθμητικοί τύποι δεδομένων χωρίζονται σε προσεγγιστικούς (Real, Float) και ακριβείς (decimal,numeric). Στους τελευταίους δηλώνουμε πόσα ακριβώς ψηφία επιθυμούμε πριν και μετά την υποδιαστολή. Το σημείο αυτό χρήζει προσοχής σε σχέση με την διαχείριση αθροισμάτων και ποσών. Στους προσεγγιστικούς αριθμούς, ανάλογα με τον τρόπο υπολογισμού αθροισμάτων, μπορούμε να έχουμε προβλήματα απώλειας – ακρίβειας δεκαδικών ψηφίων. Για την κλασική ανάγκη αποθήκευσης χρηματικών ποσών προτείνεται ο, απόλυτα ακριβής, τύπος decimal (9,2) ή decimal (18,2) για μεγαλύτερα ποσά. .

- Πολύ πρακτική και χρήσιμη είναι η λειτουργία ενός αυτόματου κλειδιού. Αυτό, όπως φαίνεται στην Εικόνα 2.11, υλοποιείται με ένα πεδίο τύπου int, στις ιδιότητες του οποίου ρυθμίζουμε σε yes την επιλογή Identity Specification. Με τον τρόπο αυτό είμαστε σίγουροι ότι θα έχουμε, χωρίς να ασχολούμαστε στα ερωτήματα εισαγωγής, ένα σωστό μοναδικό κλειδί αυτόματα.



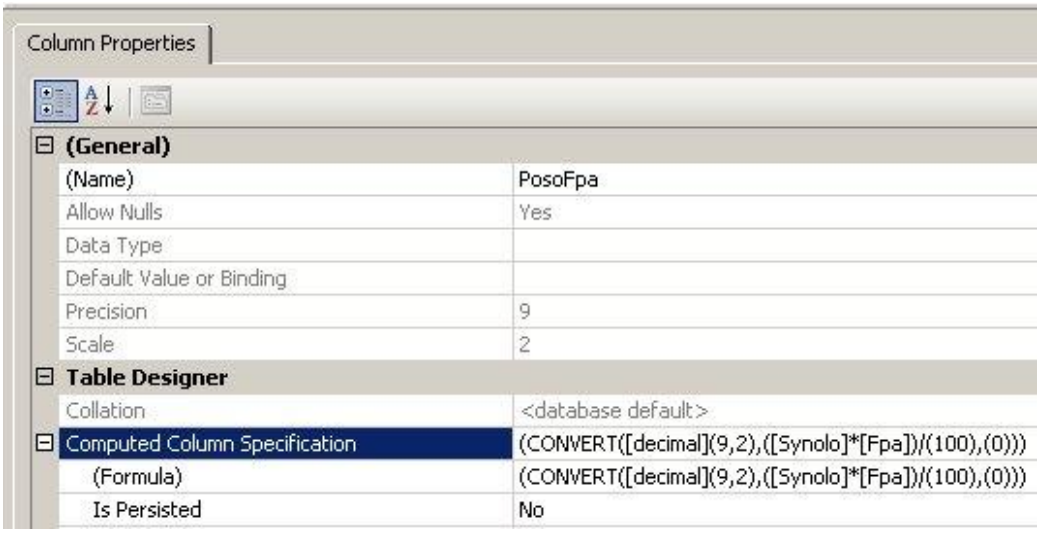
Εικόνα 2.11

- Πολύ χρήσιμος είναι ο τύπος υπολογιζόμενου πεδίου (Computed column specification), όπως εμφανίζεται στην Εικόνα 2.11. Πρόκειται για ένα πεδίο το οποίο ο χρήστης δεν μπορεί να καταχωρεί ή τροποποιεί, γιατί απλά αποτελεί το αποτέλεσμα μίας πράξης μεταξύ άλλων πεδίων. Ας υποθέσουμε, για παράδειγμα, ότι έχουμε 2 αριθμητικά πεδία: το σύνολο καθαρής αξίας (Synolo) και τον συντελεστή ΦΠΑ (Fpa), όπως φαίνεται παρακάτω. Τότε, το πεδίο PosoFpa μπορεί να οριστεί ως ένα υπολογιζόμενο πεδίο, το περιεχόμενο του οποίου θα προκύπτει από τον πολλαπλασιασμό των άλλων δύο πεδίων.



Εικόνα 2.12

- Τέλος, μπορούμε στο επίπεδο της βάσης δεδομένων (για απόλυτη αξιοπιστία) και όχι στην εφαρμογή που θα φτιάξουμε να έχουμε αυτόματα σαν πεδίο το ποσό ΦΠΑ. Όπως φαίνεται στην Εικόνα 2.13, στο τμήμα formula του πεδίου Computed Column Specification γράφουμε τον τύπο υπολογισμού [Synolo]*[Fpa]. Αν θέλουμε να στρογγυλοποιούμε στα δύο δεκαδικά, τότε γράφουμε: CONVERT([decimal](9,2),([Synolo]*[Fpa])/(100),(0)).



Εικόνα 2.13

2.2.3. Δημιουργία πινάκων με τον Management Studio

Ο SQL Server υποστηρίζει την δημιουργία σχεσιακών βάσεων δεδομένων, των οποίων το κύριο δομικό συστατικό είναι η σχέση ή, αλλιώς, πίνακας. Μερικές από τις βασικότερες ιδιότητες των πινάκων είναι οι παρακάτω:

- Ο κάθε πίνακας της βάσης δεδομένων έχει ένα δικό του μοναδικό όνομα.
- Η τιμή ενός χαρακτηριστικού σε μια γραμμή ενός πίνακα είναι ατομική.
- Δυο εγγραφές ενός πίνακα δεν επιτρέπεται να ταυτίζονται σε όλα τα χαρακτηριστικά τους. Πρέπει να διαφοροποιούνται τουλάχιστον ως προς ένα χαρακτηριστικό τους.

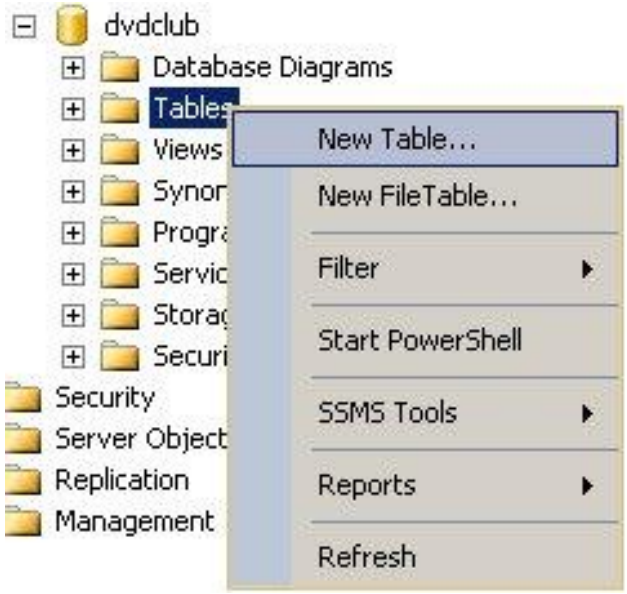
Ένα χαρακτηριστικό (ή ένα σύνολο χαρακτηριστικών) ενός πίνακα ονομάζεται όταν ταυτοποιεί μοναδικά τις εγγραφές του. Όταν απαιτούνται περισσότερα χαρακτηριστικά ενός πίνακα για να συνθέσουν ένα πρωτεύον κλειδί, κάτι που συμβαίνει συχνά, τότε ονομάζεται **σύνθετο κλειδί (composite key).**

Τώρα, θα επιχειρήσουμε να δημιουργήσουμε το νέο πίνακα ΤΑΙΝΙΑ με τα ακόλουθα πεδία:

Field	Type	Null	Key
ID	int		PRI
Τίτλος	varchar (100)		
Έτος	int	YES	

Πίνακας 2.2 ΤΑΙΝΙΑ

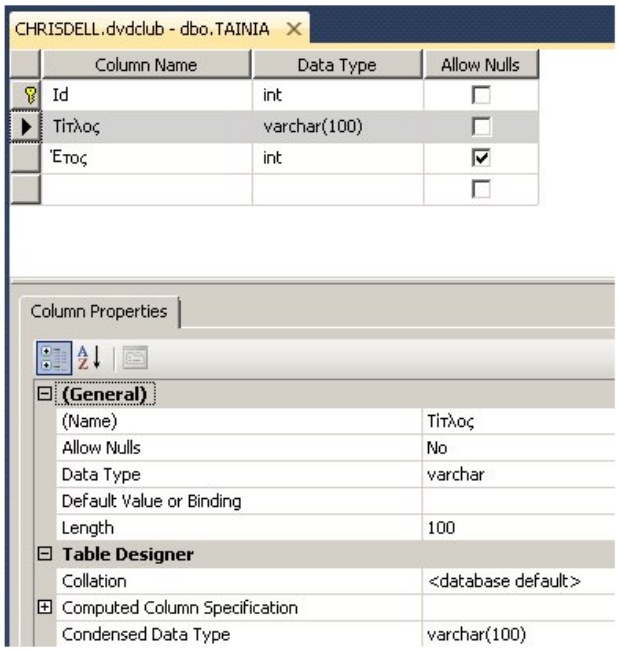
Προκειμένου να δημιουργήσουμε αυτόν το νέο πίνακα στην βάση μας με γραφικό τρόπο, ανοίγουμε τη δενδρική δομή. Στο αριστερό pane του Management Studio επεκτείνουμε τη βάση δεδομένων DVDclub, όπως φαίνεται στην Εικόνα 2.14. Στη συνέχεια, επεκτείνοντας το φάκελο Tables, βλέπουμε τους πίνακες που δημιουργεί και αποθηκεύει ο SQL Server για κάθε νέα βάση. Κάνουμε δεξί κλικ στο δεξιό τμήμα και επιλέγουμε "New Table…"



Εικόνα 2.14

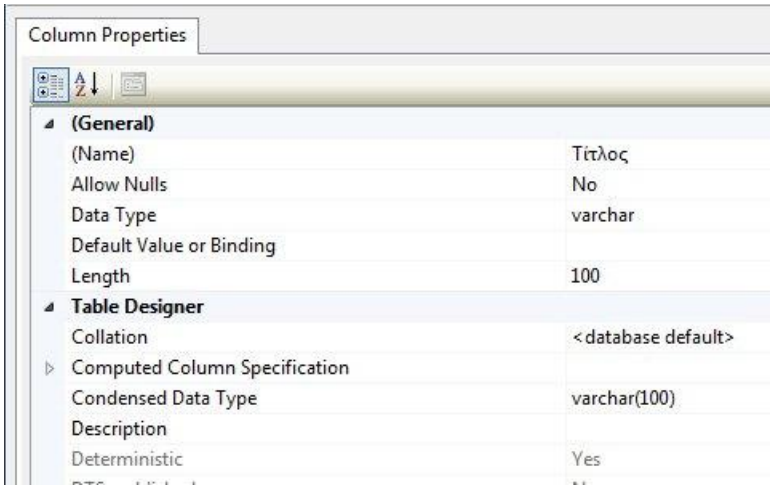
Ανοίγει ο Table Designer, όπως φαίνεται στην Εικόνα 2.15, ο οποίος μας επιτρέπει να σχεδιάσουμε το νέο πίνακα με γραφικό τρόπο. Συμπληρώνουμε τα στοιχεία του Πίνακα 2.2, όπως φαίνεται στην Εικόνα 2.15. Για

την επιλογή ενός τύπου δεδομένων, είτε πληκτρολογούμε το όνομά του είτε διαλέγουμε κάποιον από την drop-down λίστα (δεύτερη στήλη). Στη συνέχεια, ορίζουμε το κύριο κλειδί κάνοντας δεξί κλικ στη γραμμή του πεδίου ID και επιλέγουμε "Set Primary Key". Δίπλα στη γραμμή εμφανίζεται ένα εικονίδιο που αναπαριστά ένα κίτρινο κλειδί. Προσέχουμε σε κάθε πεδίο του πίνακα αν θα επιτρέπουμε NULL τιμές (π.χ. το τικ στην τελευταία στήλη της Εικόνας 2.15 δείχνει ότι το επιτρέπουμε).



Εικόνα 2.15

Παρατηρήστε τις ιδιότητες του υπό εξέταση κάθε φορά πεδίου του πίνακα ΤΑΙΝΙΑ. Για παράδειγμα, όπως φαίνεται στην Εικόνα 2.16, στον ορισμό του μήκους για το πεδίο Τίτλος είχαμε ορίσει μέγεθος 100 (η τιμή αυτή αφορά bytes). Τέλος, από τη γραμμή εργαλείων σώζουμε τον πίνακα κάνοντας κλικ στο εικονίδιο της δισκέτας. Δίνουμε το όνομα ΤΑΙΝΙΑ στον πίνακα και κλείνουμε το παράθυρο δημιουργίας πίνακα. Στη λίστα με τους πίνακες της βάσης DVDclub μπορούμε να δούμε πλέον τον πίνακα που δημιουργήσαμε.



Εικόνα 2.16

Μπορούμε να επαναλάβουμε το ίδιο για τη δημιουργία του πίνακα ΠΕΛΑΤΗΣ, με την ακόλουθη μορφή:

Field	Type	Null	Key
ID	Int		PRI
Όνομα	Varchar(30)		
Τηλέφωνο	Varchar(10)	YES	

Πίνακας 2.3 ΠΕΛΑΤΗΣ

2.2.4. Δημιουργία πινάκων με κώδικα της SQL

Εναλλακτικά, χρησιμοποιώντας τον κώδικα της SQL, θα δημιουργήσουμε τον πίνακα ΣΥΝΤΕΛΕΣΤΗΣ με την ακόλουθη μορφή:

Field	Type	Null	Key
ID	Int		PRI
Όνομα	Varchar(50)		

Πίνακας 2.4 ΣΥΝΤΕΛΕΣΤΗΣ

Ανοίγουμε τον Query Editor και προσέχουμε η ενεργή/ επιλεγμένη βάση να είναι η DVDclub. Συνεπώς, μπορούμε να πληκτρολογήσουμε τον παρακάτω κώδικα για να δημιουργήσουμε τον πίνακα ΣΥΝΤΕΛΕΣΤΗΣ με τους δύο ακόλουθους εναλλακτικούς τρόπους:

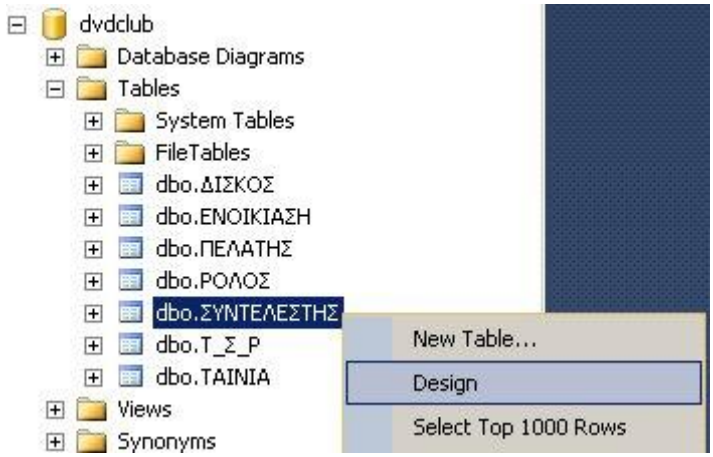
1ος τρόπος:

```
CREATE TABLE ΣΥΝΤΕΛΕΣΤΗΣ
(
        ID int NOT NULL,
        Όνομα varchar (50) NOT NULL,
        PRIMARY KEY (ID)
)
```

2ος τρόπος:

```
CREATE TABLE ΣΥΝΤΕΛΕΣΤΗΣ
(
        ID int PRIMARY KEY
        Όνομα varchar (50) NOT NULL
)
```

Σχετικά με τον πρώτο τρόπο σύνταξης, θα πρέπει να τονίσουμε ότι το NOT NULL στο χαρακτηριστικό ID είναι προαιρετικό, γιατί δηλώνεται, στη συνέχεια, ως πρωτεύον κλειδί και δεν θα είναι ποτέ NULL. Για να εκτελέσουμε τον παραπάνω κώδικα, πατάμε είτε στην εργαλειοθήκη το κουμπί Execute είτε στο πληκτρολόγιο το F5, προσέχοντας ότι έχουμε επιλέξει τη βάση DVDclub και όχι τη master. Κάνοντας δεξί κλικ πάνω από τον φάκελο Tables και στη συνέχεια επιλέγοντας Refresh, βλέπουμε τους τρεις πίνακες που έχουμε ήδη δημιουργήσει. Αν θέλουμε να αλλάξουμε οτιδήποτε στη σχεδίαση ενός πίνακα, τον επιλέγουμε με δεξί κλικ και κάνουμε κλικ στην επιλογή Design, όπως φαίνεται στην Εικόνα 2.17.



Εικόνα 2.17

Στη συνέχεια, θα δημιουργήσουμε τον πίνακα ΔΙΣΚΟΣ. Τονίζεται ότι για πρώτη φορά θα ορίσουμε και μια συσχέτιση μεταξύ δύο πινάκων. Η συσχέτιση αφορά τον πίνακα ΔΙΣΚΟΣ με τον πίνακα ΤΑΙΝΙΑ. Όπως αναφέραμε και στο μοντέλο Οντοτήτων-Συσχετίσεων, μπορούν να υπάρχουν πολλά αντίγραφα ενός δίσκου dvd για την ίδια ταινία. Συνεπώς, το πρωτεύον κλειδί (ID) του πίνακα ΤΑΙΝΙΑ θα πρέπει να μπορεί να εμφανίζεται πολλές φορές ως χαρακτηριστικό (IDΤαινίας) του πίνακα ΔΙΣΚΟΣ. Στη περίπτωση αυτή, το πρωτεύον κλειδί του πίνακα ΤΑΙΝΙΑ αποτελεί **ξένο κλειδί (foreign key)** για τον πίνακα ΔΙΣΚΟΣ.

Ο πίνακας ΔΙΣΚΟΣ έχει την ακόλουθη μορφή:

Field	Type	Null	Key
ID	Int		PRI
IDΤαινίας	Int		
Τύπος	Varchar(4)		
Τιμή	decimal (9,2)		

Πίνακας 2.5 ΔΙΣΚΟΣ

Επιστρέφουμε στον Query Editor. Προσέχουμε η επιλεγμένη βάση να είναι η DVDclub. Σβήνουμε από τον editor όλα τα περιεχόμενα. Για να δημιουργήσουμε τον πίνακα ΔΙΣΚΟΣ επιλέγουμε έναν από τους δύο τρόπους που παρουσιάζουμε παρακάτω:

1ος τρόπος:

```
CREATE TABLE ΔΙΣΚΟΣ
(
        ID int,
        IDΤαινίας int NOT NULL,
        Τύπος varchar (4) NOT NULL,
        Τιμή decimal(9,2) NOT NULL,
        PRIMARY KEY (ID),
        FOREIGN KEY (IDΤαινίας) REFERENCES TAINIA(ID) on delete cascade
)
```

2ος τρόπος:

```
CREATE TABLE ΔΙΣΚΟΣ
(
        ID int PRIMARY KEY,
        IDΤαινίας int REFERENCES TAINIA(ID) ON DELETE CASCADE,
        Τύπος varchar (4) NOT NULL,
        Τιμή decimal(9,2) NOT NULL
)
```

Ο περιορισμός ξένου κλειδιού (FOREIGN KEY constraint) ορίζει μια συσχέτιση μεταξύ δύο πινάκων (Μανωλόπουλος, & Παπαδόπουλος, 2006· Ramakrishnan, & Gehrke, 2003). Το κύριο κλειδί ενός πίνακα γίνεται ξένο κλειδί σε έναν άλλο πίνακα. Συνεπώς, δημιουργείται μία συσχέτιση ένα προς πολλά. Ο περιορισμός ξένου κλειδιού αποτρέπει ενέργειες που αφήνουν «ορφανές» εγγραφές σε ένα ξένο κλειδί/ πεδίο ενός πίνακα όταν αυτό το πεδίο αναφέρεται σε ένα κύριο κλειδί ενός άλλου πίνακα (Αναφορική Ακεραιότητα).

Στον πίνακα ΔΙΣΚΟΣ ορίζουμε ότι το πεδίο IDΤαινίας αναφέρεται (είναι ξένο κλειδί) στο πεδίο ID του πίνακα ΤΑΙΝΙΑ. Με αυτόν τον τρόπο δεν θα μπορεί να υπάρξει μια συγκεκριμένη τιμή στο πεδίο IDΤαινίας του πίνακα ΔΙΣΚΟΣ, αν αυτή η τιμή δεν έχει καταχωρηθεί προηγουμένως στο πεδίο ID του πίνακα ΤΑΙΝΙΑ.

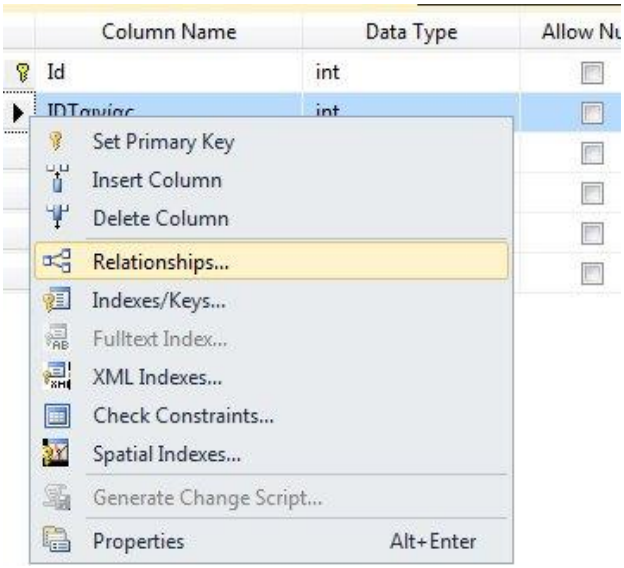
Η χρήση του ON DELETE ελέγχει την περίπτωση διαγραφής μιας εγγραφής του πίνακα ΤΑΙΝΙΑ. Υπάρχουν οι εξής επιλογές:

- **NO ACTION:** Αποτρέπει τη διαγραφή και προβάλλει μήνυμα λάθους.
- **CASCADE:** Διαγράφει την εγγραφή και προκαλεί διαγραφή όλων των εγγραφών με την ίδια τιμή στον πίνακα ΔΙΣΚΟΣ.
- **Set Null**: Διαγράφει την εγγραφή και εισάγει την τιμή Null στις εξαρτώμενες εγγραφές.
- **Set Default**: Διαγράφει την εγγραφή και εισάγει στις εξαρτώμενες εγγραφές την default τιμή που ορίσαμε κατά την δημιουργία του πεδίου.

Σημειώνεται ότι αντίστοιχοι περιορισμοί μπορούν να ορισθούν, μέσω του όρου ON UPDATE, και για την περίπτωση μεταβολής του κύριου κλειδιού.

2.2.5. Συσχετίσεις/Relationships Πινάκων

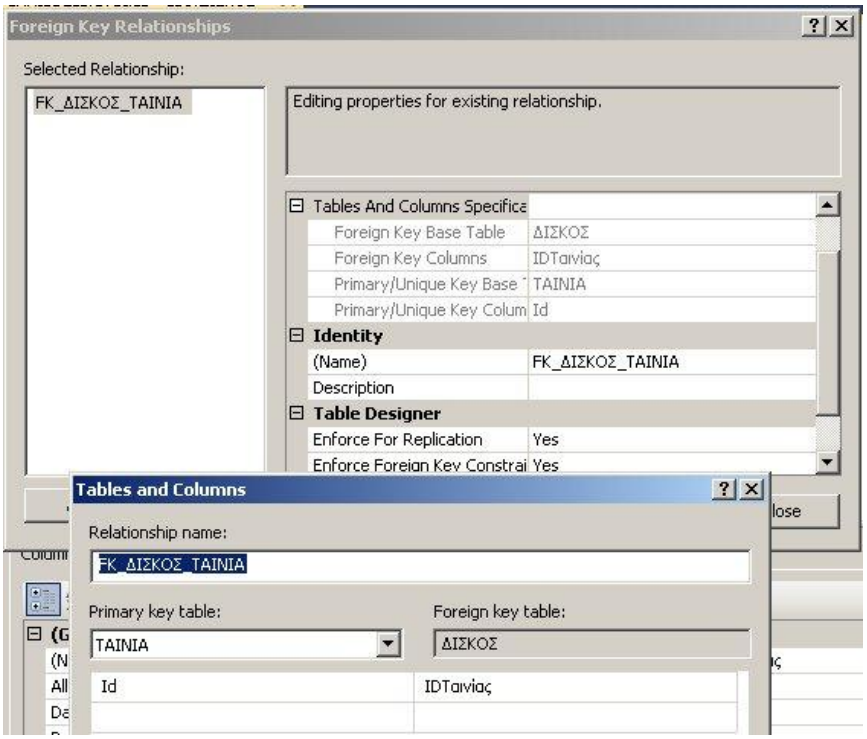
Στην Ενότητα 2.2.4 ο πίνακας ΔΙΣΚΟΣ συσχετίστηκε με τον πίνακα ΤΑΙΝΙΑ μέσω του πεδίου IDΤαινίας. Αν δεν είχαμε κάνει τη συσχέτιση των δύο πινάκων (της προηγούμενης ενότητας) αλλά είχαμε δημιουργήσει μόνο τους πίνακες, τότε, προκειμένου να δημιουργήσουμε την ίδια συσχέτιση με γραφικό τρόπο, πρέπει να επεκτείνουμε το φάκελο Tables, να κάνουμε δεξί κλικ πάνω στον πίνακα ΔΙΣΚΟΣ και, τέλος, κλικ στην επιλογή Design. Αφού ανοίξει ο Table Designer, με δεξί κλικ οπουδήποτε πάνω στην σχεδίαση επιλέγουμε Relationships, όπως φαίνεται στην Εικόνα 2.18.



Εικόνα 2.18

Στη συνέχεια εμφανίζεται το αρχικό παράθυρο της Εικόνας 2.19. Πατώντας στο κουμπί ADD μπορούμε να δημιουργήσουμε τη συσχέτιση με γραφικό τρόπο. Αρχικά μετονομάζουμε τη συσχέτιση σε FK_

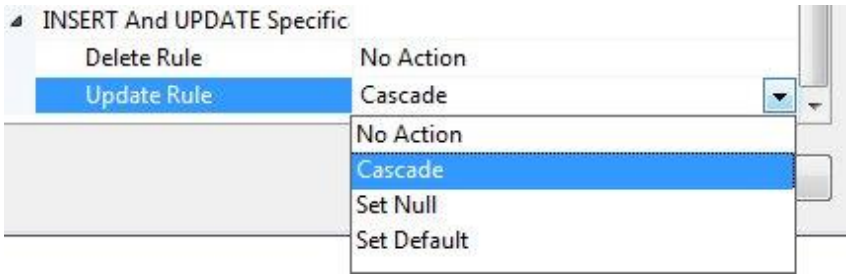
ΔΙΣΚΟΣ_ΤΑΙΝΙΑ. Στη συνέχεια κάνουμε κλικ στην επιλογή Tables and Columns Specification. Εμφανίζεται το αναδυόμενο παράθυρο της Εικόνας 2.19, στο οποίο δηλώνουμε τον κύριο πίνακα με το κύριο κλειδί του (ΤΑΙΝΙΑ.ID), καθώς και τον εξαρτώμενο πίνακα με το ξένο κλειδί (ΔΙΣΚΟΣ.IDΤαινίας).



Εικόνα 2.19

Μ' αυτόν τον τρόπο, στον πίνακα ΔΙΣΚΟΣ ορίζουμε το πεδίο IDΤαινίας να είναι ξένο κλειδί του πεδίου ID στον πίνακα ΤΑΙΝΙΑ. Έτσι, δεν θα μπορεί να εισαχθεί μια τιμή στο πεδίο IDΤαινίας του πίνακα ΔΙΣΚΟΣ, αν αυτή προηγουμένως δεν έχει εισαχθεί στο πεδίο ID του πίνακα ΤΑΙΝΙΑ.

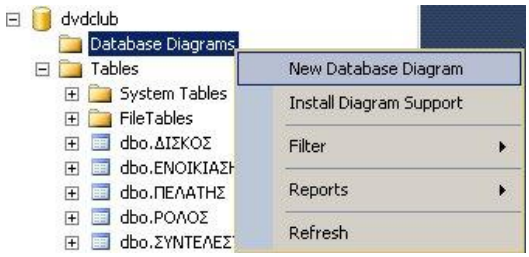
Τέλος, όπως φαίνεται στην Εικόνα 2.20, μπορούμε και με γραφικό τρόπο να κάνουμε χρήση του DELETE Rule, με επιλογές παρόμοιες με αυτές που αναφέρθηκαν στην Ενότητα 2.2.4 (π.χ. No Action, Cascade, Set Null, Set Default). Αντίστοιχα, η χρήση της ρύθμισης Update Rule αφορά την περίπτωση που γίνεται update στο κύριο κλειδί του κύριου πίνακα.



Εικόνα 2.20

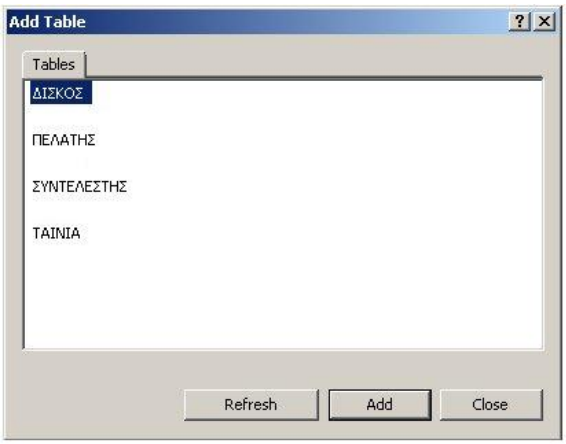
2.2.6. Δημιουργία πινάκων με τον Database Diagrams

Κάνοντας δεξί κλικ πάνω στον φάκελο Database Diagrams, όπως φαίνεται στην Εικόνα 2.21, δημιουργούμε ένα νέο διάγραμμα, στο οποίο μπορούμε με οπτικό τρόπο να ορίσουμε τη δομή και τις συσχετίσεις των πινάκων της βάσης δεδομένων μας.



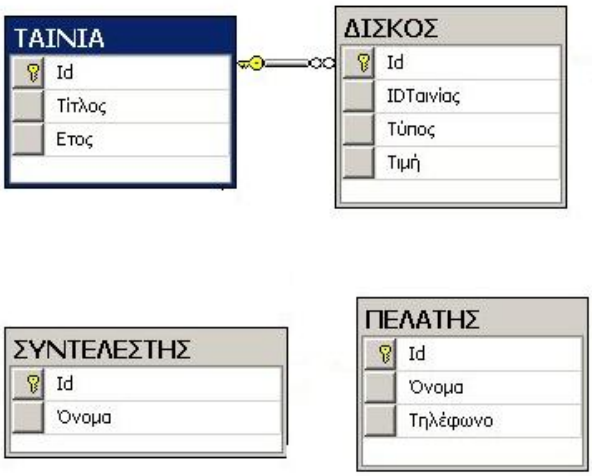
Εικόνα 2.21

Στο παράθυρο που εμφανίζει όλους τους διαθέσιμους πίνακες, όπως φαίνεται στην Εικόνα 2.22, επιλέγουμε αυτούς που θέλουμε να εμφανιστούν και τους εισάγουμε με add στο χώρο του διαγράμματός μας.



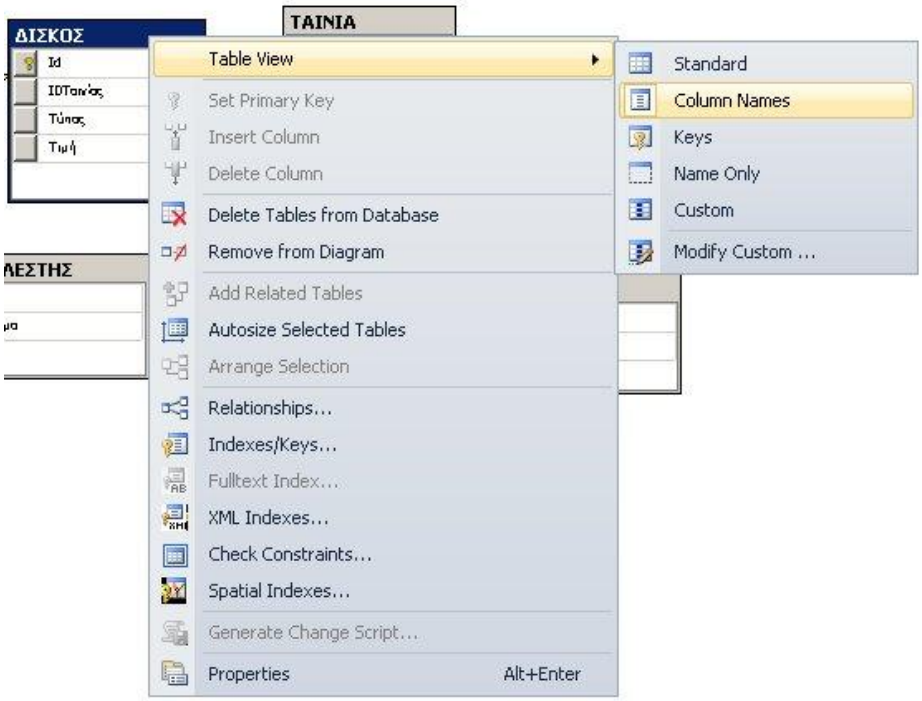
Εικόνα 2.22

Με τη χρήση του ποντικιού μπορούμε εύκολα να αλλάξουμε το μέγεθος και τη διάταξη των πινάκων, όπως φαίνεται στην Εικόνα 2.23.



Εικόνα 2.23

Με δεξί κλικ πάνω σε κάθε γραμμή συσχέτισης μπορούμε να χειριστούμε τις ιδιότητες μιας συσχέτισης. Επίσης με δεξί κλικ πάνω σε πίνακα μπορούμε να χειριστούμε το τι βλέπουμε στο διάγραμμα καθώς και όλες τις αλλαγές που μπορούμε να επιβάλλουμε στην σχεδίαση του, όπως φαίνεται στην Εικόνα 2.24.



Εικόνα 2.24

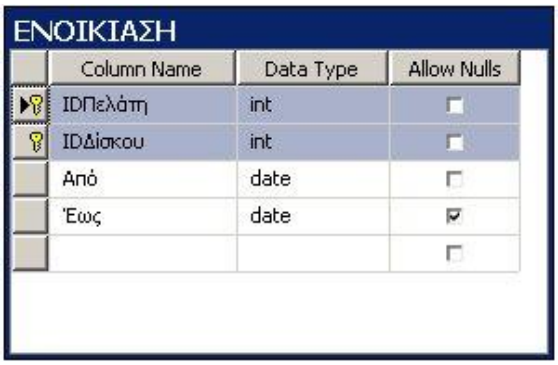
Μπορούμε τώρα να προχωρήσουμε στη δημιουργία του πίνακα ΕΝΟΙΚΙΑΣΗ, ο οποίος θα έχει την ακόλουθη μορφή:

Field	Type	Null	Key
IDΠελάτη	int		PRI
IDΔίσκου	int		PRI
Από	date		
Έως	date	YES	

Πίνακας 2.6 ΕΝΟΙΚΙΑΣΗ

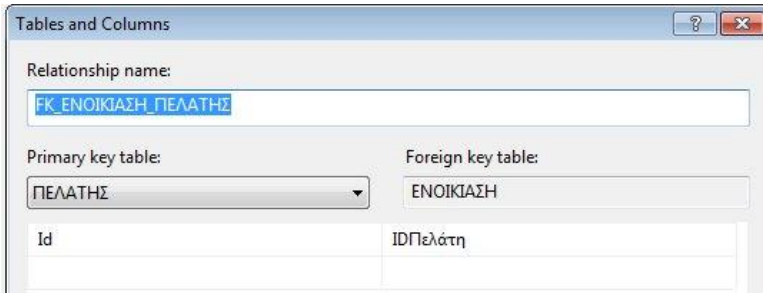
Στον πίνακα ΕΝΟΙΚΙΑΣΗ το κύριο κλειδί αποτελείται από περισσότερα του ενός πεδία (IDΠελάτη, IDΔίσκου). Σε αυτήν την περίπτωση, επιτρέπεται να υπάρχουν διπλές τιμές στο ίδιο πεδίο. Όμως, κάθε συνδυασμός των τιμών των δύο πεδίων που αποτελούν το κύριο κλειδί πρέπει να είναι μοναδικός. Μ' αυτόν τον τρόπο, ο σχεδιασμός δεν επιτρέπει ο ίδιος πελάτης να ενοικιάσει δεύτερη φορά μια ταινία. Πρέπει να επισημάνουμε ότι η στήλη IDΠελάτη είναι ξένο κλειδί προς τη στήλη ID του πίνακα ΠΕΛΑΤΗΣ, ενώ η στήλη IDΔίσκου είναι ξένο κλειδί προς τη στήλη ID του πίνακα ΔΙΣΚΟΣ.

Προκειμένου να φτιάξουμε τον πίνακα ΕΝΟΙΚΙΑΣΗ μέσα από το περιβάλλον του Database Diagrams, είτε κάνουμε κλικ στο κουμπί **New Table** στην γραμμή εργαλείων είτε κάνουμε δεξί κλικ στο διάγραμμα και επιλέγουμε **New Table**. Στο παράθυρο διαλόγου Choose Name επιλέγουμε το όνομα ΕΝΟΙΚΙΑΣΗ και πατάμε OK. Στο αναδυόμενο παράθυρο (που μοιάζει με αυτό του Table Designer) προσθέτουμε τα στοιχεία για τα πεδία και ορίζουμε το κύριο κλειδί. Η επιλογή των δύο πεδίων γίνεται κρατώντας πατημένο το πλήκτρο Ctrl και ταυτόχρονα κάνοντας δεξί κλικ πάνω τους και κλικ στο Set Primary Key. Τότε, τα δύο πεδία εμφανίζονται με σκίαση, όπως φαίνεται στην Εικόνα 2.25:



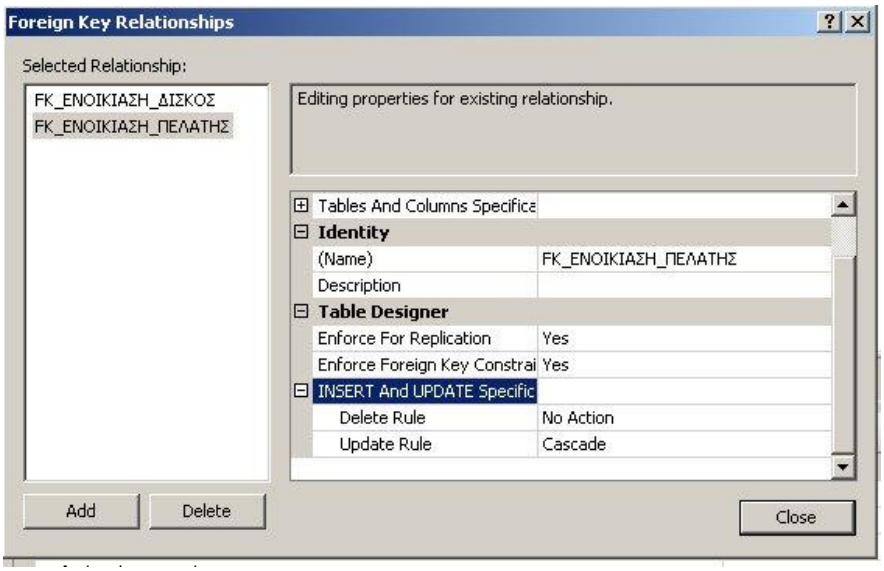
Εικόνα 2.25

Για να συσχετίσουμε τον πίνακα ΠΕΛΑΤΗΣ με τον πίνακα ΕΝΟΙΚΙΑΣΗ, επιλέγουμε το πεδίο ID του πίνακα ΠΕΛΑΤΗΣ (κρατώντας πατημένο το ποντίκι στο εικονίδιο με το κίτρινο κλειδί) και το μεταφέρουμε (με drag & drop) πάνω από το πεδίο IDΠελάτη του πίνακα ΕΝΟΙΚΙΑΣΗ. Εμφανίζεται, έτσι, το παράθυρο της Εικόνας 2.26, με τα πεδία που συμμετέχουν στη συσχέτιση να είναι προεπιλεγμένα.



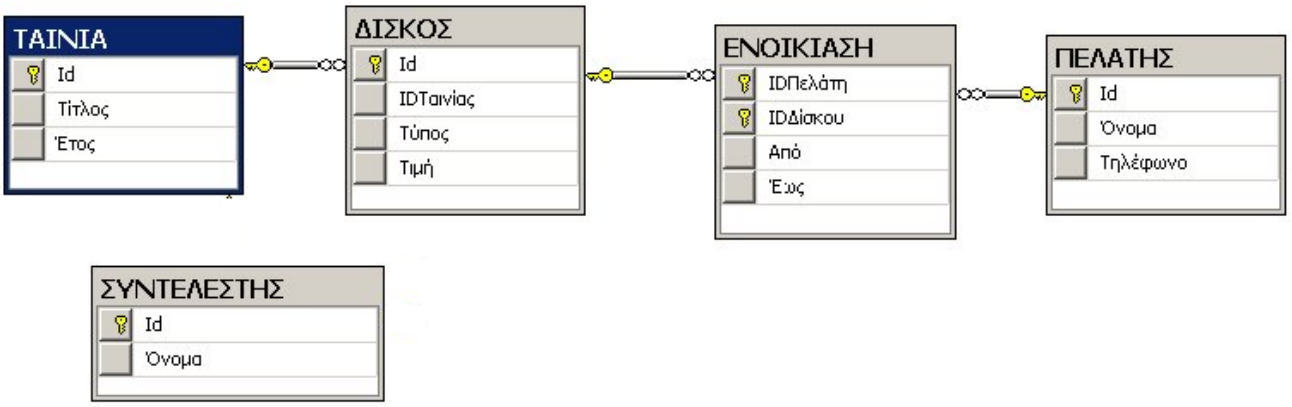
Εικόνα 2.26

Επιλέγουμε OK και, στη συνέχεια, επεκτείνουμε την επιλογή Insert and Update Specification. Επιλέγουμε Cascade, το οποίο αφορά στο τι θα γίνει σε περίπτωση αλλαγής (update) της τιμής κύριου κλειδιού, όπως φαίνεται στην Εικόνα 2.27.



Εικόνα 2.27

Κατά τον ίδιο τρόπο συσχετίζουμε τον πίνακα ΔΙΣΚΟΣ με τον πίνακα ΕΝΟΙΚΙΑΣΗ με βάση τα πεδία (ΔΙΣΚΟΣ.Id) και (ΕΝΟΙΚΙΑΣΗ.ΙΔΔίσκου). Η τρέχουσα κατάσταση του Database Diagram φαίνεται στην Εικόνα 2.28.



Εικόνα 2.28

Στη συνέχεια θα δημιουργήσουμε τον πίνακα ΡΟΛΟΣ, όπου θα καταγράφονται οι διαφορετικοί τύποι συμμετοχής ενός συντελεστή σε μια ταινία (π.χ. ηθοποιός, σκηνοθέτης, ηχολήπτης, κτλ.). Προκειμένου να δημιουργήσουμε τον πίνακα ΡΟΛΟΣ (έτσι όπως φαίνεται στον Πίνακα 2.8), κάνουμε δεξί κλικ μέσα στο Database Diagram, επιλέγουμε New Table και δίνουμε το όνομα ΡΟΛΟΣ. Στην φόρμα που θα εμφανιστεί συμπληρώνουμε τα στοιχεία του Πίνακα 2.8. Για την επιλογή ενός τύπου δεδομένων, είτε πληκτρολογούμε το όνομά του είτε διαλέγουμε κάποιον από την drop-down λίστα της δεύτερης στήλης. Στη συνέχεια, ορίζουμε το κύριο κλειδί κάνοντας δεξί κλικ στη γραμμή του πεδίου ID και επιλέγουμε "Set Primary Key". Επίσης, σε κάθε πεδίο του πίνακα προσέχουμε αν θα επιτρέπουμε NULL τιμές.

Field	Type	Null	Key
ID	Int		PRI
Περιγραφή	Varchar(25)		

Πίνακας 2.8 ΡΟΛΟΣ

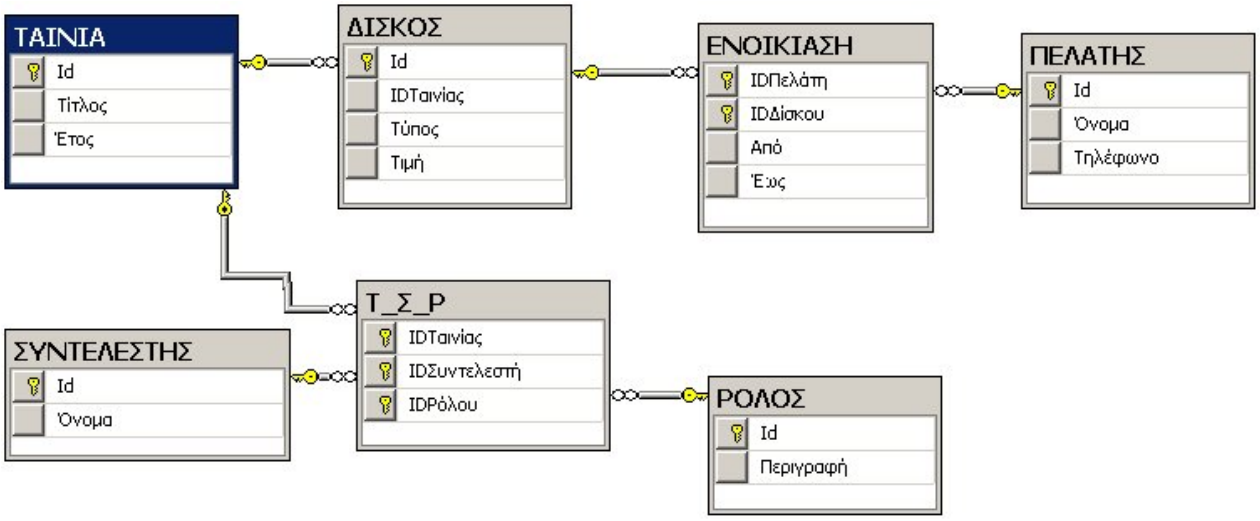
Τέλος, θα δημιουργήσουμε τον πίνακα Τ_Σ_Ρ, που συσχετίζει τον πίνακα ΤΑΙΝΙΑ με τον πίνακα ΣΥΝΤΕΛΕΣΤΗΣ και τον πίνακα ΡΟΛΟΣ, έχοντας ένα τριπλό σύνθετο κύριο κλειδί, όπως φαίνεται στον Πίνακα 2.9. Σημειώνεται ότι η στήλη IDΤαινίας είναι ξένο κλειδί προς τη στήλη ID του πίνακα ΤΑΙΝΙΑ, η στήλη ΙΔΣυντελεστή είναι ξένο κλειδί προς τη στήλη ID του πίνακα ΣΥΝΤΕΛΕΣΤΗΣ και η στήλη IDΡόλου είναι ξένο κλειδί προς την στήλη ID του πίνακα ΡΟΛΟΣ.

Field	Type	Null	Key
IDΤαινίας	Int		PRI
ΙΔΣυντελεστή	Int		PRI
IDΡόλου	Int		PRI

Πίνακας 2.9 Τ_Σ_Ρ

Μέσω της παραπάνω τριαδικής συσχέτισης του πίνακα Τ_Σ_Ρ, ο ίδιος συντελεστής θα μπορεί να συμμετέχει σε μία ταινία με περισσότερους από έναν ρόλους (π.χ. ηθοποιός και σκηνοθέτης ταυτόχρονα στην ίδια ταινία). Αυτό υλοποιείται με τη σύνδεση του πίνακα Τ_Σ_Ρ με τους πίνακες ΣΥΝΤΕΛΕΣΤΗΣ, ΡΟΛΟΣ και ΤΑΙΝΙΑ.

Μετά από τις τελευταίες δύο προσθήκες πινάκων (Τ_Σ_Ρ και ΡΟΛΟΣ) το τελικό σχήμα της βάσης δεδομένων μας είναι αυτό που φαίνεται στην Εικόνα 2.2, η οποία παρουσιάζεται ξανά αμέσως μετά (όπως φαίνεται στην Εικόνα 2.29) για λόγους εύκολης και άμεσης αναγνωσιμότητας.



Εικόνα 2.29

Μετά τη δημιουργία πινάκων, ο χρήστης είναι σε θέση να εφαρμόσει οποιαδήποτε αλλαγή επιθυμεί στους πίνακες της βάσης δεδομένων που θα επιλέξει. Στη συνέχεια, παρουσιάζουμε τη σύνταξη του αντίστοιχου ερωτήματος με ένα παράδειγμα εφαρμογής αλλαγών στον πίνακα ΠΕΛΑΤΗΣ.

ALTER TABLE ΠΕΛΑΤΗΣ
ADD COLUMN Ημερομηνία_Γέννησης **DATE NOT NULL**

ALTER TABLE ΠΕΛΑΤΗΣ
DROP COLUMN Ημερομηνία_Γέννησης

Σε αυτό το παράδειγμα εφαρμόζουμε αλλαγές στον πίνακα ΠΕΛΑΤΗΣ, προσθέτοντας τη στήλη Ημερομηνία_Γέννησης και, στη συνέχεια, διαγράφουμε τη στήλη.

Επίσης, μπορούμε να διαγράψουμε πίνακες από τη βάση δεδομένων που έχουμε δημιουργήσει με τη σύνταξη του παρακάτω ερωτήματος, με το οποίο διαγράφουμε τον πίνακα ΕΝΟΙΚΙΑΣΗ:

DROP Table ΕΝΟΙΚΙΑΣΗ (**RESTRICT** ή **CASCADE**)

Η επιλογή restrict δηλώνει ότι στην περίπτωση που ο πίνακας χρησιμοποιείται σε περιορισμούς στους ορισμούς άλλων πινάκων, δε θα διαγραφεί. Αντίθετα, με τον ορισμό cascade ο πίνακας που έχουμε δηλώσει διαγράφεται και, μαζί με αυτόν, διαγράφονται και οι περιορισμοί που τον χρησιμοποιούν.

2.3. Εισαγωγή εγγραφών στους πίνακες

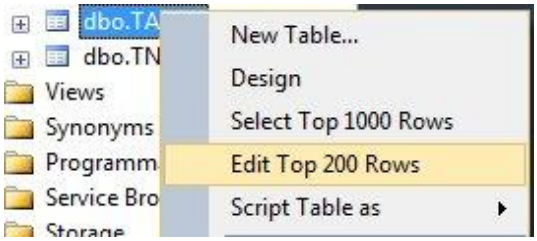
2.3.1. Εισαγωγή εγγραφών στους πίνακες με γραφικό τρόπο

Θα επιχειρήσουμε τώρα να εισάγουμε τις ακόλουθες εγγραφές στον πίνακα ΤΑΙΝΙΑ:

IDΤαινίας	Τίτλος	Χρονιά
1	Rear Window	1954
2	Psycho	1960
3	Ben-Hur	1959

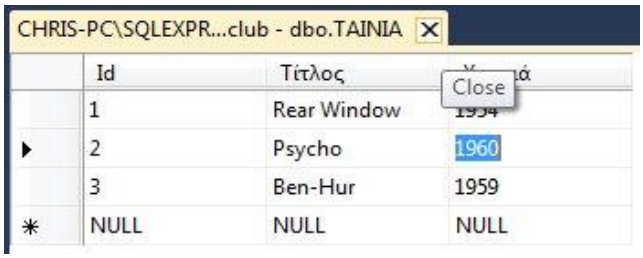
Πίνακας 2.8 ΤΑΙΝΙΑ

Είμαστε στην κονσόλα του Management Studio και βλέπουμε τους πίνακες της βάσης DVDclub. Κάνουμε δεξί κλικ στον πίνακα ΤΑΙΝΙΑ. Επιλέγουμε Edit Top 200 Rows, όπως φαίνεται στην Εικόνα 2.30.



Εικόνα 2.3

Εισάγουμε τα δεδομένα με τον ίδιο τρόπο που εισάγουμε τιμές σε ένα φύλλο του Excel, όπως φαίνεται στην Εικόνα 2.31. Κάθε φορά που καταχωρούμε μία εγγραφή, ο δείκτης πηγαίνει στην επόμενη γραμμή, κάτι που σημαίνει ότι η εγγραφή μας έχει αποθηκευτεί.



Εικόνα 2.31

Επαναλαμβάνουμε τα προηγούμενα βήματα για να εισάγουμε τις ακόλουθες εγγραφές στον πίνακα ΣΥΝΤΕΛΕΣΤΗΣ:

ID	Όνομα
1	Alfred Hitchcock
2	Grace Kelly
3	Anthony Perkins

Πίνακας 2.9 ΣΥΝΤΕΛΕΣΤΗΣ

Επίσης, για τον πίνακα ΡΟΛΟΣ εισάγουμε τις ακόλουθες εγγραφές:

ID	Περιγραφή
1	Σκηνοθέτης
2	Ηθοποιός

Πίνακας 2.9 ΡΟΛΟΣ

Επαναλαμβάνουμε τα προηγούμενα βήματα για να εισάγουμε τις ακόλουθες εγγραφές στον πίνακα ΠΕΛΑΤΗΣ:

ID	Όνομα	Τηλέφωνο
1	Perkins	246801
2	Καντακουζηνός	246801
3	Παλαιολόγος	987654

Πίνακας 2.10 ΠΕΛΑΤΗΣ

Τέλος, επαναλαμβάνουμε τα προηγούμενα βήματα για να εισάγουμε τις ακόλουθες εγγραφές στον πίνακα Τ_Σ_Ρ:

IDΤαινίας	IDΣυντελεστή	IDΡόλου
1	1	1
2	1	2
1	2	1
2	3	2

Πίνακας 2.11 Τ_Σ_Ρ

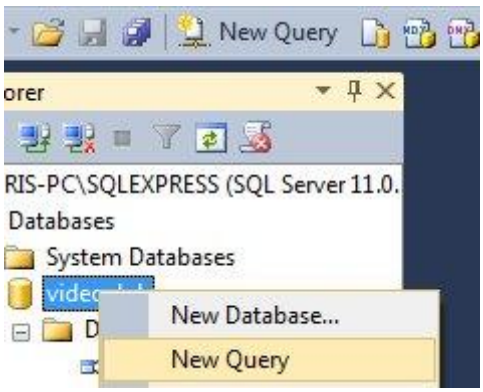
2.3.2. Εισαγωγή εγγραφών στους πίνακες με εντολές SQL

Η δήλωση INSERT χρησιμοποιείται για να προσθέσει μια νέα εγγραφή (ή εγγραφές) σε έναν πίνακα. Θα εισάγουμε τις ακόλουθες εγγραφές στον πίνακα ΔΙΣΚΟΣ:

ID	IDΤαινίας	Τύπος	Τιμή
1	1	BLU-RAY	2
2	1	DVD	3
3	2	BLU-RAY	2

Πίνακας 2.12 ΔΙΣΚΟΣ

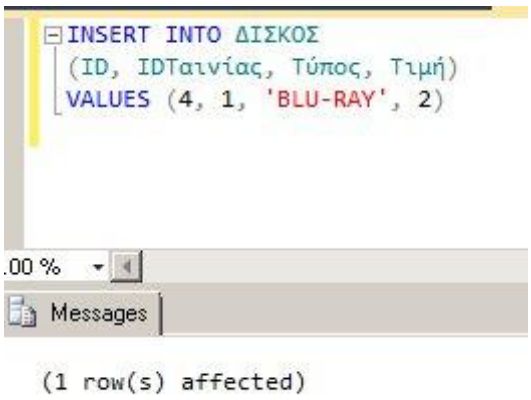
Μεταβαίνουμε στο Management Studio και επιλέγουμε από την μπάρα εργαλείων New Query. Η ίδια επιλογή μπορεί να γίνει με δεξί κλικ στην database DVDclub και κλικ στο New Query, όπως φαίνεται στην Εικόνα 2.32.



Εικόνα 2.32

Δίνουμε την παρακάτω εντολή, η οποία φαίνεται και στην Εικόνα 2.33:

INSERT INTO ΔΙΣΚΟΣ
(ID, IDΤαινίας, Τύπος, Τιμή)
VALUES (4, 1, 'BLU-RAY', 2)
GO



Εικόνα 2.33

Επιλέγουμε EXECUTE/F5 για εκτέλεση. Η εγγραφή θα εισαχθεί (εκτός απροόποτου) στο παράθυρο Messages, οπότε θα δούμε το μήνυμα (1 row(s) affected), όπως φαίνεται στην Εικόνα 2.33.

Η χρήση των ονομάτων των στηλών είναι προαιρετική. Πρέπει, όμως, να βάλουμε όλα τα ορίσματα με τη σωστή σειρά. Για παράδειγμα, η δεύτερη εγγραφή μπορεί να εισαχθεί ως εξής:

INSERT INTO ΔΙΣΚΟΣ
VALUES (2, 1, 'DVD', 3)
GO

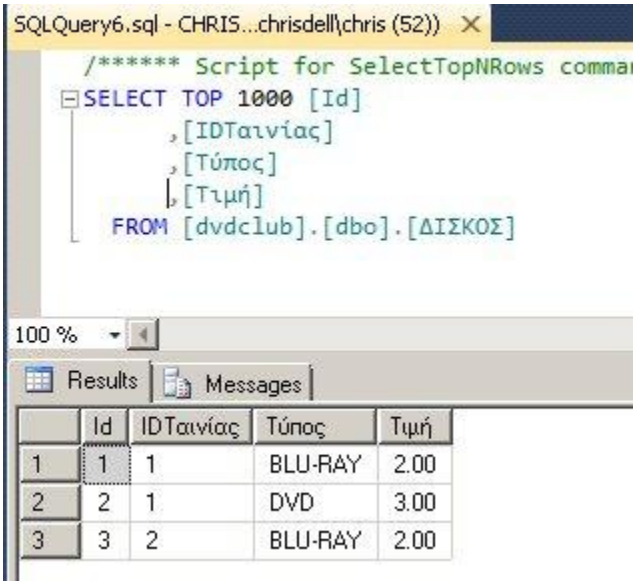
Επίσης, μπορούμε να αλλάξουμε τη σειρά με την οποία δηλώνουμε τα ορίσματα. Για παράδειγμα:

INSERT INTO ΔΙΣΚΟΣ
(ID, IDΤαινίας, Τιμή, Τύπος)
VALUES (3, 2, 2, 'BLU-RAY')
GO

Για να ελέγξουμε αν εισήχθησαν σωστά οι εγγραφές στον πίνακα ΔΙΣΚΟΣ, κάνουμε δεξί κλικ στον πίνακα και επιλέγουμε την εντολή «Select Top 1000 Rows», όπως φαίνεται στην Εικόνα 2.34.

Εικόνα 2.34

Στην Εικόνα 2.35 βλέπουμε τις τιμές των πεδίων του πίνακα ΔΙΣΚΟΣ, καθώς και το ερώτημα SQL, το οποίο συντάσσεται αυτόματα από το Management Studio.



Εικόνα 2.35

Επαναλαμβάνουμε τα προηγούμενα βήματα για την εισαγωγή των ακόλουθων εγγραφών στον πίνακα ΕΝΟΙΚΙΑΣΗ, προσέχοντας ιδιαίτερα να γράφουμε πρώτα το μήνα και μετά την ημέρα:

ΙΔΠελάτη	**ΙΔΔίσκου**	**Από**	**Έως**
1	1	07/10/2006	09/10/2006
1	2	09/20/2006	11/20/2006
2	1	09/10/2006	Null
2	2	09/30/2006	09/30/2006

Πίνακας 2.13 ΕΝΟΙΚΙΑΣΗ

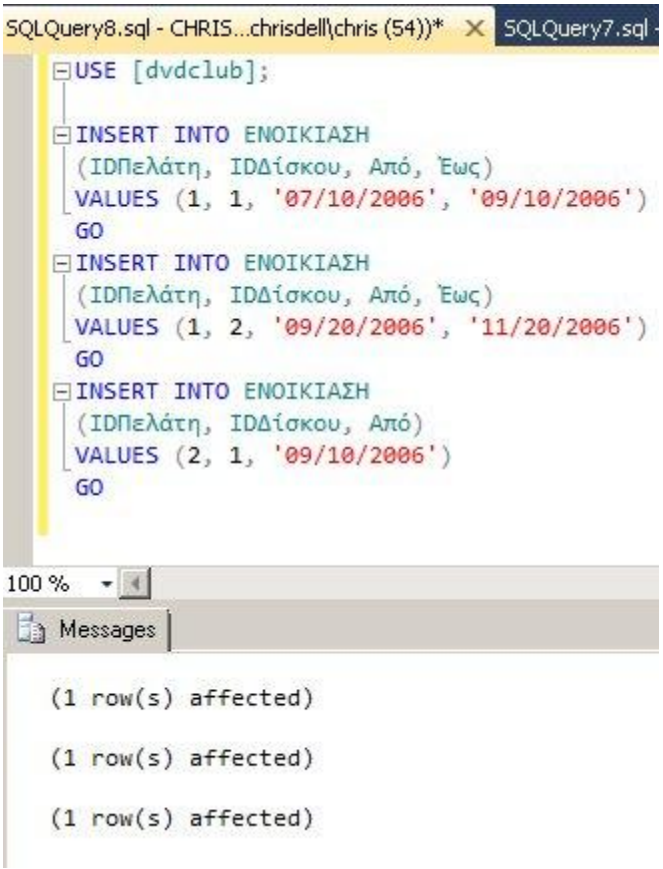
```
INSERT INTO ΕΝΟΙΚΙΑΣΗ
(ΙΔΠελάτη, ΙΔΔίσκου, Από, Έως)
VALUES (1, 1, '07/10/2006', '09/10/2006')
GO

INSERT INTO ΕΝΟΙΚΙΑΣΗ
(ΙΔΠελάτη, ΙΔΔίσκου, Από, Έως)
VALUES (1, 2, '09/20/2006', '11/20/2006')
GO
```

Αν παραλείψουμε ένα όρισμα, τότε στο αντίστοιχο πεδίο εισάγεται η τιμή NULL, εφόσον, βέβαια, δεν υπάρχει περιορισμός NOT NULL για αυτό το πεδίο. Για παράδειγμα:

```
INSERT INTO ΕΝΟΙΚΙΑΣΗ
(ΙΔΠελάτη, ΙΔΔίσκου, Από)
VALUES (2, 1, '09/10/2006')
GO
```

Αν δοκιμάσουμε να εκτελέσουμε μαζί τις εντολές και όχι ξεχωριστά, τότε το αποτέλεσμα θα φαίνεται όπως αυτό της Εικόνας 2.36:



Εικόνα 2.36

Τέλος, προκειμένου να εισάγουμε την τέταρτη εγγραφή του πίνακα 2.13, πρέπει να εκτελέσουμε την παρακάτω εντολή:

INSERT INTO ΕΝΟΙΚΙΑΣΗ
(ΙΔΠελάτη, ΙΔΔίσκου, Από, Έως)
VALUES (2, 2, '09/30/2006', '9/30/2006')
GO

2.4. Αλλαγή σε δεδομένα πινάκων

Τα δεδομένα στους πίνακες μπορούν να αλλάζουν ή να διαγράφονται. Παραθέτουμε εδώ τις δύο βασικές εντολές. Το συγκεκριμένο θέμα θα εξεταστεί πιο αναλυτικά στο Κεφάλαιο 4.

2.4.1. Ενημέρωση δεδομένων

Στην περίπτωση που απαιτείται η αλλαγή των τιμών των στηλών ενός πίνακα, χρησιμοποιούμε την εντολή UPDATE, η σύνταξη της οποίας περιγράφεται στη συνέχεια:

UPDATE όνομα_πίνακα
SET πεδίο=νέα_τιμή
WHERE κριτήρια

Η ενημέρωση των δεδομένων αποτυγχάνει στην περίπτωση που μετά από έλεγχο δεν ικανοποιούνται οι περιορισμοί ακεραιότητας. Επίσης, θα πρέπει να δοθεί προσοχή στον τρόπο που θα ορίσουμε τη συνθήκη WHERE, ώστε να προσδιορίσουμε αν θα ενημερωθούν τα δεδομένα σε μία μόνο εγγραφή του πίνακα, προσθέτοντας στη συνθήκη κάποιο κλειδί του πίνακα. Στην περίπτωση που το κλειδί παραλειφθεί, υπάρχει κίνδυνος να αλλοιωθούν τα δεδομένα του πίνακα. Ας υποθέσουμε, λοιπόν, ότι μας ζητείται να αλλάξουμε την ημερομηνία έναρξης ενοικίασης από 09/30/2006 σε 09/29/2006 στην εγγραφή του πίνακα ΕΝΟΙΚΙΑΣΗ που αφορά τον οπτικό δίσκο με κωδικό (ΙΔΔίσκου = 2), ο οποίος ενοικιάστηκε από τον πελάτη με κωδικό (ΙΔΠελάτη=2). Η σύνταξη της εντολής DELETE είναι:

UPDATE ΕΝΟΙΚΙΑΣΗ
SET Έως = ’09/30/2006’
WHERE ΙΔΔίσκου = 2 and ΙΔΠελάτη = 2

2.4.2. Διαγραφή δεδομένων

Εκτός από την ανανέωση των δεδομένων ενός πίνακα, μπορούμε να διαγράψουμε δεδομένα μέσα σε πίνακες. Με την εντολή διαγραφής DELETE μπορούμε να καταργήσουμε τις γραμμές που επιθυμούμε από επιλεγμένους πίνακες. Η σύνταξη της εντολής DELETE είναι:

DELETE FROM όνομα_πίνακα
WHERE κριτήρια

Σ’αυτήν την περίπτωση, γίνεται έλεγχος των περιορισμών ακεραιότητας για την επιτυχή εκτέλεση της διαγραφής. Ας υποθέσουμε, λοιπόν, ότι μας ζητείται να διαγράψουμε την εγγραφή του πίνακα ΕΝΟΙΚΙΑΣΗ που αφορά τον οπτικό δίσκο με κωδικό (ΙΔΔίσκου = 2), ο οποίος ενοικιάστηκε από τον πελάτη με κωδικό (ΙΔΠελάτη=2). Η αντίστοιχη εντολή σε SQL είναι:

DELETE FROM ΕΝΟΙΚΙΑΣΗ
WHERE ΙΔΔίσκου = 2 and ΙΔΠελάτη = 2

2.5. Κώδικας SQL για τη δημιουργία της βάσης δεδομένων DVDclub

Παρακάτω δίνεται ολόκληρος ο κώδικας SQL για τη δημιουργία της βάσης δεδομένων DVDclub, προκειμένου να αποφευχθούν πιθανά προβλήματα λόγω λαθών από την τμηματική εκτέλεση των ερωτημάτων, όπως αυτά αναπτύχθηκαν στις προηγούμενες ενοτήτες. Η πρότασή μας προς τον χρήστη είναι να διαγράψει τη βάση δεδομένων DVDclub από το σύστημα και να την δημιουργήσει ξανά. Στη συνέχεια, επιλέγοντας ως ενεργή βάση δεδομένων τη DVDclub (όχι τη master) και ανοίγοντας τον Query Editor, να τρέξει τον παρακάτω κώδικα SQL:

```
USE [DVDclub]
GO
/****** Object:  Table [dbo].[ΔΙΣΚΟΣ]    Script Date: 11/7/2015 5:23:16 μμ
******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[ΔΙΣΚΟΣ](
      [Id] [int] NOT NULL,
      [IDΤαινίας] [int] NOT NULL,
      [Τύπος] [varchar](7) NOT NULL,
      [Τιμή] [decimal](9, 2) NOT NULL,
 CONSTRAINT [PK_ΔΙΣΚΟΣ] PRIMARY KEY CLUSTERED
(
      [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/****** Object:  Table [dbo].[ΕΝΟΙΚΙΑΣΗ]    Script Date: 11/7/2015 5:23:16 μμ
******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[ΕΝΟΙΚΙΑΣΗ](
      [IDΠελάτη] [int] NOT NULL,
      [IDΔίσκου] [int] NOT NULL,
      [Από] [date] NOT NULL,
      [Έως] [date] NULL,
 CONSTRAINT [PK_ΕΝΟΙΚΙΑΣΗ] PRIMARY KEY CLUSTERED
(
      [IDΠελάτη] ASC,
      [IDΔίσκου] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
/****** Object:  Table [dbo].[ΠΕΛΑΤΗΣ]    Script Date: 11/7/2015 5:23:16 μμ
******/
SET ANSI_NULLS ON
```

```
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[ΠΕΛΑΤΗΣ](
      [Id] [int] NOT NULL,
      [Όνομα] [varchar](30) NOT NULL,
      [Τηλέφωνο] [varchar](10) NULL,
 CONSTRAINT [PK_ΠΕΛΑΤΗΣ] PRIMARY KEY CLUSTERED
(
      [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/****** Object:  Table [dbo].[ΡΟΛΟΣ]    Script Date: 11/7/2015 5:23:16 μμ
******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[ΡΟΛΟΣ](
      [Id] [int] NOT NULL,
      [Περιγραφή] [varchar](25) NOT NULL,
 CONSTRAINT [PK_ΡΟΛΟΣ] PRIMARY KEY CLUSTERED
(
      [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/****** Object:  Table [dbo].[ΣΥΝΤΕΛΕΣΤΗΣ]    Script Date: 11/7/2015 5:23:16
μμ ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[ΣΥΝΤΕΛΕΣΤΗΣ](
      [Id] [int] NOT NULL,
      [Όνομα] [varchar](50) NULL,
 CONSTRAINT [PK_ΣΥΝΤΕΛΕΣΤΗΣ] PRIMARY KEY CLUSTERED
(
      [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

```
GO
SET ANSI_PADDING OFF
GO
/****** Object:  Table [dbo].[T_Σ_P]    Script Date: 11/7/2015 5:23:16 μμ
******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[T_Σ_P](
      [IDΤαινίας] [int] NOT NULL,
      [IDΣυντελεστή] [int] NOT NULL,
      [IDΡόλου] [int] NOT NULL,
 CONSTRAINT [PK_ΤΣ2] PRIMARY KEY CLUSTERED
(
      [IDΤαινίας] ASC,
      [IDΣυντελεστή] ASC,
      [IDΡόλου] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
/****** Object:  Table [dbo].[TAINIA]    Script Date: 11/7/2015 5:23:16 μμ
******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[TAINIA](
      [Id] [int] NOT NULL,
      [Τίτλος] [varchar](100) NOT NULL,
      [Έτος] [int] NULL,
 CONSTRAINT [PK_TAINIA] PRIMARY KEY CLUSTERED
(
      [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
INSERT [dbo].[ΔΙΣΚΟΣ] ([Id], [IDΤαινίας], [Τύπος], [Τιμή]) VALUES (1, 1,
N'BLU-RAY', CAST(2.00 AS Decimal(9, 2)))
GO
INSERT [dbo].[ΔΙΣΚΟΣ] ([Id], [IDΤαινίας], [Τύπος], [Τιμή]) VALUES (2, 1,
N'DVD ', CAST(3.00 AS Decimal(9, 2)))
GO
INSERT [dbo].[ΔΙΣΚΟΣ] ([Id], [IDΤαινίας], [Τύπος], [Τιμή]) VALUES (3, 2,
N'BLU-RAY', CAST(2.00 AS Decimal(9, 2)))
GO

INSERT [dbo].[ΕΝΟΙΚΙΑΣΗ] ([IDΠελάτη], [IDΔίσκου], [Από], [Έως]) VALUES (1, 1,
CAST(N'2006-07-10' AS Date), CAST(N'2006-09-10' AS Date))
GO
```

```
INSERT [dbo].[ΕΝΟΙΚΙΑΣΗ] ([IDΠελάτη], [IDΔίσκου], [Από], [Έως]) (1, 2,
CAST(N'2006-09-20' AS Date), CAST(N'2006-11-20' AS Date))
GO
INSERT [dbo].[ΕΝΟΙΚΙΑΣΗ] ([IDΠελάτη], [IDΔίσκου], [Από], [Έως]) VALUES (2, 1,
CAST(N'2006-09-10' AS Date), NULL)
GO

INSERT [dbo].[ΠΕΛΑΤΗΣ] ([Id], [Όνομα], [Τηλέφωνο]) VALUES (1, N'Perkins',
N'246801')
GO
INSERT [dbo].[ΠΕΛΑΤΗΣ] ([Id], [Όνομα], [Τηλέφωνο]) VALUES (2,
N'Καντακουζηνός', N'246801')
GO
INSERT [dbo].[ΠΕΛΑΤΗΣ] ([Id], [Όνομα], [Τηλέφωνο]) VALUES (3, N'Παλαιολόγος',
N'987654')
GO


INSERT [dbo].[ΡΟΛΟΣ] ([Id], [Περιγραφή]) VALUES (1, N'Σκηνοθέτης')
GO
INSERT [dbo].[ΡΟΛΟΣ] ([Id], [Περιγραφή]) VALUES (2, N'Ηθοποιός')
GO

INSERT [dbo].[ΣΥΝΤΕΛΕΣΤΗΣ] ([Id], [Όνομα]) VALUES (1, N'Alfred Hitchcock')
GO
INSERT [dbo].[ΣΥΝΤΕΛΕΣΤΗΣ] ([Id], [Όνομα]) VALUES (2, N'Grace Kelly')
GO
INSERT [dbo].[ΣΥΝΤΕΛΕΣΤΗΣ] ([Id], [Όνομα]) VALUES (3, N'Anthony Perkins')
GO

INSERT [dbo].[T_Σ_P] ([IDΤαινίας], [IDΣυντελεστή], [IDΡόλου]) VALUES (1, 1,
1)
GO
INSERT [dbo].[T_Σ_P] ([IDΤαινίας], [IDΣυντελεστή], [IDΡόλου]) VALUES (1, 2,
2)
GO
INSERT [dbo].[T_Σ_P] ([IDΤαινίας], [IDΣυντελεστή], [IDΡόλου]) VALUES (2, 1,
1)
GO
INSERT [dbo].[T_Σ_P] ([IDΤαινίας], [IDΣυντελεστή], [IDΡόλου]) VALUES (2, 3,
2)
GO

INSERT [dbo].[TAINIA] ([Id], [Τίτλος], [Έτος]) VALUES (1, N'Rear Window',
1954)
GO
INSERT [dbo].[TAINIA] ([Id], [Τίτλος], [Έτος]) VALUES (2, N'Psycho', 1960)
GO
INSERT [dbo].[TAINIA] ([Id], [Τίτλος], [Έτος]) VALUES (3, N'Ben-Hur', 1959)
GO

ALTER TABLE [dbo].[ΔΙΣΚΟΣ]  WITH CHECK ADD  CONSTRAINT [FK_ΔΙΣΚΟΣ_TAINIA]
FOREIGN KEY ([IDΤαινίας])
REFERENCES [dbo].[TAINIA] ([Id])
ON UPDATE CASCADE
GO
ALTER TABLE [dbo].[ΔΙΣΚΟΣ] CHECK CONSTRAINT [FK_ΔΙΣΚΟΣ_TAINIA]
GO
```

```
ALTER TABLE [dbo].[ΕΝΟΙΚΙΑΣΗ]  WITH CHECK ADD  CONSTRAINT
[FK_ΕΝΟΙΚΙΑΣΗ_ΔΙΣΚΟΣ] FOREIGN KEY ([IDΔίσκου])
REFERENCES [dbo].[ΔΙΣΚΟΣ] ([Id])
ON UPDATE CASCADE
GO
ALTER TABLE [dbo].[ΕΝΟΙΚΙΑΣΗ] CHECK CONSTRAINT [FK_ΕΝΟΙΚΙΑΣΗ_ΔΙΣΚΟΣ]
GO
ALTER TABLE [dbo].[ΕΝΟΙΚΙΑΣΗ]  WITH CHECK ADD  CONSTRAINT
[FK_ΕΝΟΙΚΙΑΣΗ_ΠΕΛΑΤΗΣ] FOREIGN KEY ([IDΠελάτη])
REFERENCES [dbo].[ΠΕΛΑΤΗΣ] ([Id])
ON UPDATE CASCADE
GO
ALTER TABLE [dbo].[ΕΝΟΙΚΙΑΣΗ] CHECK CONSTRAINT [FK_ΕΝΟΙΚΙΑΣΗ_ΠΕΛΑΤΗΣ]
GO
ALTER TABLE [dbo].[Τ_Σ_Ρ]  WITH CHECK ADD  CONSTRAINT [FK_Τ_Σ_Ρ_ΡΟΛΟΣ]
FOREIGN KEY ([IDΤαινίας])
REFERENCES [dbo].[ΡΟΛΟΣ] ([Id])
GO
ALTER TABLE [dbo].[Τ_Σ_Ρ] CHECK CONSTRAINT [FK_Τ_Σ_Ρ_ΡΟΛΟΣ]
GO0
ALTER TABLE [dbo].[Τ_Σ_Ρ]  WITH CHECK ADD  CONSTRAINT [FK_Τ_Σ_Ρ_ΣΥΝΤΕΛΕΣΤΗΣ]
FOREIGN KEY ([IDΣυντελεστή])
REFERENCES [dbo].[ΣΥΝΤΕΛΕΣΤΗΣ] ([Id])
ON UPDATE CASCADE
GO
ALTER TABLE [dbo].[Τ_Σ_Ρ] CHECK CONSTRAINT [FK_Τ_Σ_Ρ_ΣΥΝΤΕΛΕΣΤΗΣ]
GO
ALTER TABLE [dbo].[Τ_Σ_Ρ]  WITH CHECK ADD  CONSTRAINT [FK_Τ_Σ_Ρ_TAINIA]
FOREIGN KEY ([IDΤαινίας])
REFERENCES [dbo].[TAINIA] ([Id])
ON UPDATE CASCADE
GO
ALTER TABLE [dbo].[Τ_Σ_Ρ] CHECK CONSTRAINT [FK_Τ_Σ_Ρ_TAINIA]
GO
```

2.6. Ασκήσεις

1. Να δημιουργήσετε στο περιβάλλον του SQL Server τους παρακάτω πίνακες που αφορούν μια βάση δεδομένων με το όνομα REAL ESTATE για ένα μεσιτικό γραφείο. Λάβετε υπόψη σας ότι με την ενιαία υπογράμμιση ορίζονται τα πρωτεύονται κλειδιά των πινάκων, ενώ με τη διακεκομμένη υπογράμμιση ορίζονται τα ξένα κλειδιά.

   ΠΕΛΑΤΕΣ (Πελάτης_Id, Όνομα, Περιοχή_Id)

   ΠΕΡΙΟΧΕΣ (Περιοχή_Id, Περιγραφή)

   ΑΚΙΝΗΤΑ(Ακίνητο_Id, Περιοχή_Id, Τύπος, Τετραγωνικά, τιμή_εκκίνησης)

   ΑΓΟΡΑΠΩΛΗΣΙΕΣ(Πελάτης_id, Ακίνητο_id, Ημερομηνία, Ποσό_αγοράς)

2. Να εισάγετε τις παρακάτω εγγραφές στους αντίστοιχους πίνακες της βάσης δεδομένων REAL ESTATE (που δημιουργήσατε στην άσκηση 1) με τη χρήση εντολών της SQL.

Πελάτης_id	Όνομα	Περιοχή_id
1	MIXALIS	1
2	BASILIS	2
3	KOSTAS	3

Πίνακας 2.14 ΠΕΛΑΤΕΣ

Περιοχή _id	Περιγραφή
1	KALAMARIA
2	KRINI
3	PILEA

Πίνακας 2.15 ΠΕΡΙΟΧΕΣ

3. Να εισάγετε τις παρακάτω εγγραφές στους αντίστοιχους πίνακες της βάσης δεδομένων REAL ESTATE (που δημιουργήσατε στην άσκηση 1) με γραφικό τρόπο μέσα από το Management Studio.

Ακίνητο_id	Περιοχή_id	Τύπος	Τετραγωνικά	Τιμή_εκκίνησης
1	1	DIAMERISMA	100	9000
2	2	KATASTIMA	150	14000
3	3	MEZONETA	200	19000

Πίνακας 2.16 ΑΚΙΝΗΤΑ

Πελάτης_id	Ακίνητο_id	Ημερομηνία	Ποσό_αγοράς
2	1	1/17/1990	10000
2	2	2/18/1995	15000

Πίνακας 2.16 ΑΓΟΡΑΠΩΛΗΣΙΕΣ

4. Χρησιμοποιώντας την εντολή UPDATE της SQL να αλλάξετε την ημερομηνία αγοράς από 1/17/1990 σε 7/11/2015 στην εγγραφή ου πίνακα ΑΓΟΡΑΠΩΛΗΣΙΕΣ που αφορά το Ακίνητο με κωδικό (Ακίνητο_id = 1), το οποίο αγοράστηκε από τον πελάτη με κωδικό (Πελάτης_id = 2).

5. Χρησιμοποιώντας της εντολή DELETE της SQL να διαγράψετε την εγγραφή του πίνακα ΑΓΟΡΑΠΩΛΗΣΙΕΣ που αφορά το ακίνητο με κωδικό (Ακίνητο_id = 2), το οποίο αγοράστηκε από τον πελάτη με κωδικό (ΙΔΠελάτη=2).

2.7. Βιβλιογραφία/Αναφορές

Hoffer, J. A., Venkatarama, R., & Topi, H. (2013). *Modern Database Management*, Prentice Hall.

Μανωλόπουλος, Ι., & Παπαδόπουλος, Α. Ν. (2006). *Συστήματα Βάσεων Δεδομένων: Θεωρία & Πρακτική Εφαρμογή*, Αθήνα, Εκδόσεις Νέων Τεχνολογιών.

Ramakrishnan, R., & Gehrke, J. (2003). Database Management Systems, McGraw-Hill.

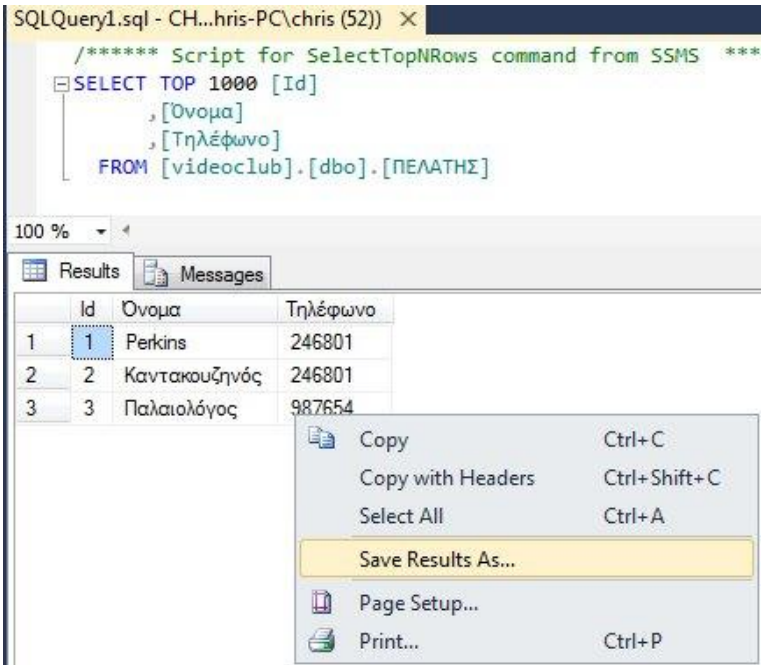
Κεφάλαιο 3. Ερωτήματα SQL

Σύνοψη
Σ' αυτό το κεφάλαιο θα παρουσιάσουμε βασικά και σύνθετα ερωτήματα της SQL. Τα ερωτήματα θα υποβληθούν στην βάση δεδομένων DVDclub που δημιουργήθηκε στο προηγούμενο κεφάλαιο. Πιο συγκεκριμένα, θα μελετηθούν εντολές της SQL που αφορούν τη διαχείριση δεδομένων (Data Manipulation Language). Ενδεικτικά αναφέρεται ότι θα παρουσιαστούν ερωτήματα σύνδεσης πινάκων, ομαδοποίησης, πράξεων συνόλων. Επίσης, θα παρουσιαστεί η δημιουργία ερωτημάτων με γραφικό τρόπο (Query by Example) μέσα από το περιβάλλον του Query Designer. Οι πίνακες της παραπάνω βάσης δεδομένων βρίσκονται και σε ιστοσελίδα στο διαδίκτυο, στη διεύθυνση ***http://www.donotwait.gr/formslib/runsql.aspx.*** *Αν, λοιπόν, γράψουμε τα ερωτήματα SQL στο text editor που υπάρχει στη σελίδα, τότε τα αποτελέσματα του κάθε ερωτήματος θα είναι διαθέσιμα απευθείας στην ίδια σελίδα.*

3.1. Βασικά Ερωτήματα

3.1.1. Διαχείριση του Results Pane

Όταν εκτελείται ένα ερώτημα SQL, τα αποτελέσματα του ερωτήματος εμφανίζονται στο Results Pane. Στην καρτέλα Result μπορεί να εμφανιστεί το σύνολο των αποτελεσμάτων, είτε σε μορφή κειμένου/text είτε σε μορφή πλέγματος/grid, όπως αυτό που φαίνεται στην Εικόνα 3.1. Το πλέγμα μάς επιτρέπει να επιλέγουμε ξεχωριστά κελιά, στήλες ή γραμμές από το σύνολο των αποτελεσμάτων.



Εικόνα 3.1

Με δεξί κλικ πάνω στο πλέγμα/grid μπορούμε είτε να αντιγράψουμε τα αποτελέσματα, είτε να τα αποθηκεύσουμε σε κάποια μορφή εξωτερικού αρχείου, είτε να τα εκτυπώσουμε, όπως φαίνεται στο αναδυόμενο παράθυρο της Εικόνας 3.1.

3.1.2. Ερωτήματα επιλογής εγγραφών από έναν πίνακα.

Η βασική λειτουργία της αναζήτησης πληροφοριών στη βάση δεδομένων γίνεται με ερωτήματα της ακόλουθης μορφής:

Select A1, A2, …, An
From R1, R2, …, Rm
Where συνθήκη;

Η σύνταξη της εντολής περιλαμβάνει τρεις *όρους*:

- Ο όρος **from** δηλώνει το σύνολο των *m* σχέσεων (πινάκων) R1, R2, …, Rm στους οποίους γίνεται η αναζήτηση (αντιστοιχεί στην πράξη του Καρτεσιανού γινομένου).
- Ο όρος **select** χρησιμοποιείται για τη δήλωση των *n* πεδίων A1, A2, …, An που θα περιέχονται στο αποτέλεσμα (αντιστοιχεί στην πράξη της προβολής/ projection χωρίς απαλοιφή όμοιων εγγραφών).
- Ο όρος **where** περιέχει την *συνθήκη* που πρέπει να πληρούν οι εγγραφές του αποτελέσματος (αντιστοιχεί στην πράξη της επιλογής/selection). Η χρήση του είναι προαιρετική. Αν δεν υπάρχει ο όρος **where,** τότε στο αποτέλεσμα περιλαμβάνονται όλες οι εγγραφές, ανεξαρτήτως συνθήκης.

Έστω ότι αναφερόμαστε στον πίνακα ΠΕΛΑΤΗΣ, ο οποίος περιέχει πληροφορίες για τους πελάτες του DVDclub. Ένα παράδειγμα ερωτήματος είναι:

"Να δοθούν τα ονόματα όλων των πελατών ". Η αντίστοιχη εντολή σε SQL είναι:

Q1

Select Όνομα
From ΠΕΛΑΤΗΣ

Το αποτέλεσμα του ερωτήματος είναι το παρακάτω:

```
Όνομα
------------------------------
Perkins
Καντακουζηνός
Παλαιολόγος
```

Ο όρος **select** περιέχει το πεδίο Όνομα, το οποίο θα περιέχεται στο αποτέλεσμα. Τα υπόλοιπα πεδία της σχέσης ΠΕΛΑΤΗΣ, όπως για παράδειγμα το πεδίο Τηλέφωνο, δεν θα συμπεριληφθούν στο αποτέλεσμα. Ο όρος **from** περιέχει τη σχέση ΠΕΛΑΤΗΣ από την οποία θα γίνει η επιλογή των εγγραφών του αποτελέσματος. Στην συγκεκριμένη εντολή γίνεται επιλογή από μία μόνο σχέση. Η εντολή του παραδείγματος δεν έχει τον όρο **where**, με αποτέλεσμα να επιλεχθούν όλοι οι πελάτες ανεξαρτήτως συνθήκης, όπως απαιτούσε η ερώτηση.

Τα αποτελέσματα που προκύπτουν από μια εντολή της SQL μπορεί να περιέχουν δύο ίδιες εγγραφές. Για παράδειγμα, έστω το ερώτημα:

"Να δοθούν τα τηλέφωνα όλων των πελατών ". Η αντίστοιχη εντολή σε SQL είναι:

Q2

Select Τηλέφωνο
From ΠΕΛΑΤΗΣ

Το αποτέλεσμα του ερωτήματος είναι το παρακάτω:

Τηλέφωνο

246801
246801
987654

Αν δύο πελάτες έχουν το ίδιο τηλέφωνο (π.χ., είναι συγκάτοικοι), τότε αυτό εμφανίζεται δύο φορές στο αποτέλεσμα. Αν θέλουμε να πάρουμε έναν κατάλογο με όλα τα τηλέφωνα των πελατών μας, όπου επιθυμούμε κάθε τηλέφωνο να εμφανίζεται μόνο μία φορά, τότε απαιτείται η χρήση ενός επιπλέον όρου που ονομάζεται **distinct**. Το αποτέλεσμα δίνεται μετά από διαγραφή όλων των όμοιων εγγραφών, όπως δηλαδή συμβαίνει με την εφαρμογή της πράξης της προβολής στη σχεσιακή άλγεβρα. Για παράδειγμα, έστω το ερώτημα:

"Να δοθούν τα τηλέφωνα όλων των πελατών, όπου το κάθε τηλέφωνο να εμφανίζεται μία φορά μόνο": Η αντίστοιχη εντολή σε SQL είναι:

Q3

Select distinct Τηλέφωνο
From ΠΕΛΑΤΗΣ

Τηλέφωνο

246801
987654

Συχνά είναι χρήσιμη η επιλογή όλων των πεδίων μίας σχέσης. Αυτό μπορεί να γίνει με την παράθεση όλων των πεδίων μετά τον όρο **select**, ή, πιο απλά, με τη χρήση του συμβόλου συντόμευσης * (αστερίσκος). Για παράδειγμα, για την ερώτηση:

"Να δοθούν όλα τα στοιχεία των πελατών". Η αντίστοιχη εντολή σε SQL είναι:

Q4

Select *
From ΠΕΛΑΤΗΣ

ID	Όνομα	Τηλέφωνο
1	Perkins	246801
2	Καντακουζηνός	246801
3	Παλαιολόγος	987654

Με τον όρο **Select** μπορεί να χρησιμοποιηθεί οποιαδήποτε από τις πράξεις της πρόσθεσης (+), αφαίρεσης (-) ή άλλων συναρτήσεων, στις τιμές των πεδίων που εμφανίζονται στο αποτέλεσμα.

Έστω, για παράδειγμα, ότι στο πεδίο Τηλέφωνο θέλουμε να εμφανίζεται το πρόθεμα 2310. Θα χρησιμοποιήσουμε τη συνένωση συμβολοσειρών.

Q5

```
Select Όνομα, '2310' + Τηλέφωνο
From ΠΕΛΑΤΗΣ
```

Όνομα	Τηλέφωνο
Perkins	2310246801
Καντακουζηνός	2310246801
Παλαιολόγος	2310987654

Ο όρος **where** περιέχει μια συνθήκη που πρέπει να ικανοποιούν οι εγγραφές του αποτελέσματος. Έστω, για παράδειγμα, η ερώτηση:

"Να βρεθούν οι κωδικοί των δίσκων που είναι τύπου BLU-RAY". Η αντίστοιχη εντολή της SQL είναι:

Q6

```
Select ID
From ΔΙΣΚΟΣ
Where Τύπος = 'BLU-RAY'
```

```
ID
-----------
1
3
```

Ως συνθήκη μπορεί να ληφθεί οποιαδήποτε λογική έκφραση που αποτελείται από μία ή περισσότερες προτάσεις συνδυασμένες με λογικό *και* (and) ή λογικό *ή* (or) και παρενθέσεις. Χρειάζεται προσοχή στην προτεραιότητα μεταξύ των τελεστών, η οποία επιτυγχάνεται με τη χρήση παρενθέσεων. Για παράδειγμα, για την ερώτηση:

"Να βρεθούν οι κωδικοί από τους δίσκους που είναι τύπου BLU-RAY ή αλλιώς η τιμή τους είναι μεγαλύτερη του 2". Η αντίστοιχη εντολή SQL είναι:

Q7

```
Select ID
From ΔΙΣΚΟΣ
Where (Τύπος = 'BLU-RAY') or (Τιμή > 2)
```

```
ID
-----------
1
2
3
```

Για αλφαριθμητικά δεδομένα, είναι χρήσιμος ο τελεστής like, για την ταύτιση μιας συμβολοσειράς εντός μιας άλλης συμβολοσειράς. Με τον τελεστή like χρησιμοποιούνται δύο ειδικοί χαρακτήρες (μπαλαντέρ):

- % για ταύτιση οποιασδήποτε συμβολοσειράς
- _ για ταύτιση οποιουδήποτε χαρακτήρα

Για παράδειγμα, για την ερώτηση: **"Να βρεθούν ποια ονόματα πελατών αρχίζουν από Κ", η** αντίστοιχη εντολή SQL είναι:

Q8

Select Όνομα
From ΠΕΛΑΤΗΣ
Where Όνομα like 'Κ%'

```
Όνομα
------------------------------
Καντακουζηνός
```

Για την εύρεση πεδίων που έχουν ή όχι τιμή NULL ορίζονται οι τελεστές IS NULL και IS NOT NULL.

Για παράδειγμα, για την ερώτηση: **"Να βρεθούν τα στοιχεία των ενοικιάσεων που δεν έχει ορισθεί ημερομηνία επιστροφής", η** αντίστοιχη εντολή SQL είναι:

Q9

Select *
From ΕΝΟΙΚΙΑΣΗ
Where Έως **IS NULL**

IDΠελάτη	IDΔίσκου	Από	Έως
2	1	2006-09-10	NULL

3.1.3. Ταξινόμηση αποτελεσμάτων

Στην γλώσσα SQL η διάταξη των εγγραφών δεν έχει σημασία, αφού οι εγγραφές ενός πίνακα αντιστοιχούν στα στοιχεία ενός συνόλου που, ως γνωστόν, δεν είναι διατεταγμένα. Πολλές φορές, όμως, προκύπτει η ανάγκη ταξινόμησης των αποτελεσμάτων με βάση την τιμή ενός πεδίου. Η SQL επιτρέπει την ταξινόμηση με τη χρήση του όρου **Order by**. Έστω, για παράδειγμα, η ερώτηση:

"Να δοθούν οι κωδικοί και η τιμή των δίσκων ταξινομημένοι ως προς τη τιμή τους, κατά αύξοντα τρόπο":

Q10

Select ID, Τιμή
From ΔΙΣΚΟΣ
Order by Τιμή

ID	Τιμή
1	2.00
3	2.00
2	3.00

Εξ ορισμού η ταξινόμηση γίνεται κατά αύξουσα σειρά. Ο προσδιορισμός αύξουσας ή φθίνουσας ταξινόμησης γίνεται με τους όρους **asc** και **desc** αντίστοιχα. Επίσης, μπορεί να γίνει ταξινόμηση με βάση περισσότερα πεδία. Έστω, για παράδειγμα, η ερώτηση:

"Να δοθούν οι κωδικοί των δίσκων ταξινομημένοι κατά φθίνουσα σειρά ως προς την τιμή ενοικίασης. Στην περίπτωση ίσων τιμών ενοικίασης, η ταξινόμηση να γίνει κατά αύξουσα σειρά ως προς το ID τους":

Q11

```
Select Τιμή, ID
From ΔΙΣΚΟΣ
Order by Τιμή desc, ID asc
```

Τιμή	ID
3.00	2
2.00	1
2.00	3

Εκτός των πεδίων με αριθμητικές τιμές, η ταξινόμηση μπορεί να γίνει και σε διαφορετικούς τύπους δεδομένων. Επίσης, ας σημειωθεί ότι η τιμή NULL θεωρείται μικρότερη από κάθε άλλη. Έστω, για παράδειγμα, η ερώτηση:

"Να δοθούν οι κωδικοί των δίσκων που έχουν ενοικιαστεί, καθώς και οι ημερομηνίες επιστροφής τους, ταξινομημένες ως προς τις ημερομηνίες επιστροφής":

Q12

```
Select IDΔίσκου, DAY(Έως) as Ημέρα, MONTH(Έως) as Μήνας, YEAR(Έως) as Έτος
From ΕΝΟΙΚΙΑΣΗ
Order by Έως
```

IDΔίσκου	Ημέρα	Μήνας	Έτος
1	NULL	NULL	NULL
1	10	9	2006
2	20	11	2006

Τονίζεται ότι οι συναρτήσεις DAY, MONTH, YEAR επιστρέφουν την ημερομηνία, τον μήνα και τον χρόνο, αντιστοίχως, ενός τύπου δεδομένων date. Όταν στα αποτελέσματά μας θέλουμε μόνο τις πρώτες *k* πρώτες εγγραφές, τότε χρησιμοποιούμε τον τελεστή **top** *k*. Έστω, για παράδειγμα, η ερώτηση:

"Να δοθούν οι κωδικοί των 2 δίσκων με τη μεγαλύτερη τιμή".

Q13

```
Select top 2 ID
From ΔΙΣΚΟΣ
Order by Τιμή
```

ID
1
3

3.2. Ερωτήματα επιλογής εγγραφών από πολλούς πίνακες

3.2.1. Εσωτερική και εξωτερική σύνδεση πινάκων

Με τον όρο **from** δηλώνεται μια λίστα ονομάτων πινάκων, από την οποία μπορούν να αντληθούν τα δεδομένα. Στα προηγούμενα παραδείγματα οι αναζητήσεις δεδομένων αφορούσαν έναν μόνο πίνακα. Η αναζήτηση δεδομένων σε περισσότερους πίνακες γίνεται με τη βοήθεια της πράξης της σύνδεσης (join). Η πράξη της σύνδεσης είναι ουσιαστικά μια επιλογή πάνω στο καρτεσιανό γινόμενο (Hoffer, Venkatarama, & Topi, 2013· Μανωλόπουλος, & Παπαδόπουλος, 2006), που μπορεί να οριστεί τόσο στο πεδίο **where** όσο και στο πεδίο **from**. Για παράδειγμα, έστω η ερώτηση:

"Να δοθεί για κάθε συντελεστή το όνομά του και οι ρόλοι με τους οποίους αυτός έχει συμμετάσχει σε ταινίες". Η αντίστοιχη εντολή SQL είναι:

Q14

```
Select Όνομα, Περιγραφή
From Τ_Σ_Ρ, ΣΥΝΤΕΛΕΣΤΗΣ, ΡΟΛΟΣ
Where Τ_Σ_Ρ.IDΣυντελεστή = ΣΥΝΤΕΛΕΣΤΗΣ.ID
and Τ_Σ_Ρ.IDΡόλου = ΡΟΛΟΣ.Id
```

Όνομα	Περιγραφή
Alfred Hitchcock	Σκηνοθέτης
Grace Kelly	Ηθοποιός
Alfred Hitchcock	Σκηνοθέτης
Anthony Perkins	Ηθοποιός

Στο προηγούμενο παράδειγμα σύνδεσης ο όρος **where** περιείχε τις συνθήκες που ήταν απαραίτητες για τη δημιουργία των κατάλληλων συνδέσεων μεταξύ των πινάκων. Η βασική λογική συνίσταται στο γεγονός ότι η τιμή του ξένου κλειδιού ενός πίνακα πρέπει να ισούται με την τιμή του κύριου κλειδιού του άλλου πίνακα, στον οποίο αναφέρεται το ξένο κλειδί. Εκτός αυτής της συνθήκης είναι δυνατό να περιέχονται και άλλες, οι οποίες πρέπει να πληρούνται από τις εγγραφές του αποτελέσματος. Για παράδειγμα, έστω το ερώτημα:

"Να δοθούν οι κωδικοί των ταινιών στις οποίες έχει συμμετάσχει ο Alfred Hitchcock". Η αντίστοιχη εντολή SQL είναι:

Q15

```
Select IDΤαινίας
From Τ_Σ_Ρ, ΣΥΝΤΕΛΕΣΤΗΣ
Where Τ_Σ_Ρ.IDΣυντελεστή = ΣΥΝΤΕΛΕΣΤΗΣ.ID and
ΣΥΝΤΕΛΕΣΤΗΣ.Όνομα = 'Alfred Hitchcock'
```

IDΤαινίας
1
2

Στο παραπάνω παράδειγμα, εκτός από την συνθήκη σύνδεσης υπάρχει και ο απαιτούμενος περιορισμός για το όνομα του συντελεστή. Η πράξη της σύνδεσης μπορεί να δηλωθεί και εκτός του where. Αυτό γίνεται με τη χρήση του **inner join** στο **from**. Έστω ξανά το ερώτημα του προηγούμενου παραδείγματος:

"Να δοθούν οι κωδικοί των ταινιών στις οποίες έχει συμμετάσχει ο Alfred Hitchcock". Η αντίστοιχη εντολή SQL με τη χρήση του όρου inner join είναι:

Q16

```
Select IDΤαινίας
From Τ_Σ_Ρ inner join ΣΥΝΤΕΛΕΣΤΗΣ on
Τ_Σ_Ρ.IDΣυντελεστή = ΣΥΝΤΕΛΕΣΤΗΣ.ID
Where ΣΥΝΤΕΛΕΣΤΗΣ.Όνομα = 'Alfred Hitchcock'
```

```
IDΤαινίας
-----------
1
2
```

ΠΡΟΣΟΧΗ! Στο ερώτημα ποιον από τους δύο τρόπους πρέπει να χρησιμοποιούμε, θα μπορούσαμε να πούμε ότι στην περίπτωση όπου η αναζήτηση αφορά λίγους πίνακες, η σύνδεση με τη βοήθεια του όρου where είναι εξίσου βολική με αυτή που ορίζεται στο πεδίο from. Όμως, στην περίπτωση που η αναζήτηση αφορά πολλούς πίνακες, τότε η χρήση του inner join είναι προτιμότερη. Αυτό συμβαίνει γιατί μπορούμε να δούμε με ποια πεδία συνδέονται οι υπό σύνδεση πίνακες. Συνεπώς, γίνεται ευκολότερα η ανάγνωση ενός πολύπλοκου ερωτήματος σύνδεσης πινάκων και επίσης, η σύνδεση μπορεί να γίνει και σε περισσότερους από 2 πίνακες. Για παράδειγμα, έστω το ερώτημα:

"Να βρεθούν για κάθε πελάτη το όνομα του, ο κωδικός και η τιμή των δίσκων που έχει ενοικιάσει". Η αντίστοιχη εντολή SQL είναι:

Q17

```
Select ΠΕΛΑΤΗΣ.Όνομα, ΔΙΣΚΟΣ.ID, ΔΙΣΚΟΣ.Τιμή
From ΠΕΛΑΤΗΣ inner join ΕΝΟΙΚΙΑΣΗ on
ΠΕΛΑΤΗΣ.ID = ΕΝΟΙΚΙΑΣΗ.IDΠελάτη inner join ΔΙΣΚΟΣ on
ΕΝΟΙΚΙΑΣΗ.IDΔίσκου = ΔΙΣΚΟΣ.ID
```

Όνομα	ID	Τιμή
Perkins	1	2,00
Perkins	2	3,00
Καντακουζηνός	1	2,00

Στο αποτέλεσμα του **inner join** συμμετέχουν μόνο οι εγγραφές των πινάκων, για τις οποίες υπάρχει μία τουλάχιστον ταύτιση. Όπως φαίνεται, στο αποτέλεσμα δεν συμμετέχουν τα υπόλοιπα ονόματα των πελατών, επειδή για αυτά δεν βρέθηκε ταύτιση.

Υπάρχει και μια ειδικότερη πράξη σύνδεσης (join) μεταξύ δυο πινάκων, που ονομάζεται ***left outer join***. Με την πράξη του *left outer join* το αποτέλεσμα περιέχει όλες τις εγγραφές για τις οποίες υπάρχει ταύτιση, και επιπλέον όλες τις εγγραφές του αριστερού πίνακα για τις οποίες δεν έγινε ταύτιση με καμία από τις εγγραφές του δεξιού πίνακα. Για τις εγγραφές αυτού του είδους, οι στήλες που αντιστοιχούν στο δεύτερο πίνακα έχουν τιμή ίση με null. Τελικά, κάθε εγγραφή του αριστερού πίνακα συμμετέχει στο αποτέλεσμα. Για παράδειγμα, έστω το ερώτημα:

"Να βρεθούν για κάθε πελάτη το όνομά του, ο κωδικός και η τιμή των δίσκων που έχει ενοικιάσει. Να εμφανίζονται και οι πελάτες που δεν έχουν ενοικιάσει κάποιο δίσκο". Η αντίστοιχη εντολή SQL είναι:

Q18

Select ΠΕΛΑΤΗΣ.Όνομα, ΔΙΣΚΟΣ.ID, ΔΙΣΚΟΣ.Τιμή
From ΠΕΛΑΤΗΣ **left outer join** ΕΝΟΙΚΙΑΣΗ **on**
ΠΕΛΑΤΗΣ.ID = ΕΝΟΙΚΙΑΣΗ.IDΠελάτη **left outer join** ΔΙΣΚΟΣ **on**
ΕΝΟΙΚΙΑΣΗ.IDΔίσκου =ΔΙΣΚΟΣ.ID

Όνομα	ID	Τιμή
Perkins	1	2,00
Perkins	2	3,00
Καντακουζηνός	1	2,00
Παλαιολόγος	NULL	NULL

Όπως φαίνεται στα αποτελέσματα του παραδείγματός μας, για τον πελάτη με όνομα Παλαιολόγος, δεν έχει γίνει καμία ταύτιση, οπότε οι αντίστοιχες στήλες ID και Τιμή είναι ίσες με NULL.

Αντίστοιχα, με την πράξη *right outer join* το αποτέλεσμα περιέχει όλες αυτές τις εγγραφές για τις οποίες υπάρχει ταύτιση, και επιπλέον όλες τις εγγραφές του δεξιού πίνακα για τις οποίες δεν έγινε ταύτιση με καμία εγγραφή του αριστερού πίνακα. Γι΄ αυτές τις εγγραφές (για τις οποίες δεν υπήρξε ταύτιση), οι στήλες που αντιστοιχούν στο δεύτερο πίνακα έχουν τιμή ίση με null. Συνεπώς, κάθε εγγραφή του δεξιού πίνακα συμμετέχει στο αποτέλεσμα. Για παράδειγμα, έστω το ερώτημα:

"Να βρεθούν για κάθε πελάτη το όνομά του, ο κωδικός και η τιμή των δίσκων που έχει ενοικιάσει. Επίσης, να εμφανίζονται οι κωδικοί και οι τιμές των δίσκων που δεν έχουν ενοικιαστεί από κάποιον πελάτη". Η αντίστοιχη εντολή SQL είναι:

Q19

Select ΠΕΛΑΤΗΣ.Όνομα, ΔΙΣΚΟΣ.ID, ΔΙΣΚΟΣ.Τιμή
From ΠΕΛΑΤΗΣ **right outer join** ΕΝΟΙΚΙΑΣΗ **on** ΠΕΛΑΤΗΣ.ID = ΕΝΟΙΚΙΑΣΗ.IDΠελάτη **right outer join ΔΙΣΚΟΣ on** ΕΝΟΙΚΙΑΣΗ.IDΔίσκου = ΔΙΣΚΟΣ.ID

Όνομα	ID	Τιμή
Perkins	1	2,00
Καντακουζηνός	1	2,00
Perkins	2	3,00
NULL	3	2,00

Τέλος, υπάρχει και η πράξη ***full outer join***, το αποτέλεσμα της οποίας περιέχει επιπλέον όλες τις εγγραφές της δεξιάς σχέσης και της αριστερής σχέσης για τις οποίες δεν έγινε ταύτιση. Στις εγγραφές αυτού του είδους, οι στήλες που αντιστοιχούν στη δεύτερη σχέση έχουν τιμή ίση με null. Συνεπώς, κάθε εγγραφή είτε της αριστερής είτε της δεξιάς σχέσης συμμετέχει στο αποτέλεσμα. Για παράδειγμα, έστω το ερώτημα:

"Να βρεθούν για κάθε πελάτη το όνομά του, ο κωδικός και η τιμή των δίσκων που έχει ενοικιάσει. Να εμφανίζονται και οι πελάτες που δεν έχουν ενοικιάσει κάποιο δίσκο, αλλά και οι κωδικοί και τιμές των δίσκων που δεν έχουν ενοικιαστεί από κάποιον πελάτη ". Η αντίστοιχη εντολή SQL είναι:

Q20

```
Select ΠΕΛΑΤΗΣ.Όνομα, ΔΙΣΚΟΣ.ID, ΔΙΣΚΟΣ.Τιμή
From ΠΕΛΑΤΗΣ full outer join ΕΝΟΙΚΙΑΣΗ on ΠΕΛΑΤΗΣ.ID = ΕΝΟΙΚΙΑΣΗ.IDΠελάτη
full outer join ΔΙΣΚΟΣ on ΕΝΟΙΚΙΑΣΗ.IDΔίσκου = ΔΙΣΚΟΣ.ID
```

Όνομα	ID	Τιμή
Perkins	1	2,00
Perkins	2	3,00
Καντακουζηνός	1	2,00
Παλαιολόγος	NULL	NULL
NULL	3	2,00

3.2.2. Μετονομασία και αυτό-σύνδεση

Σε ορισμένα ερωτήματα προκύπτει η ανάγκη μετονομασίας πεδίων ή πινάκων. Αυτό γίνεται με τη χρήση ενός επιπλέον όρου που ονομάζεται **as**. Έστω το παράδειγμα του ερωτήματος Q5 στην Ενότητα 3.1.2,, στην οποία έγινε προσθήκη του προθέματος "2310" στα τηλέφωνα των πελατών. Το αποτέλεσμα είναι μια σχέση με δύο πεδία. Το δεύτερο πεδίο είναι αποτέλεσμα πράξης μεταξύ συμβολοσειρών, οπότε δεν έχει όνομα. Σε αυτή την περίπτωση χρησιμοποιείται ο όρος **as** ως εξής:

Θέλουμε στο πεδίο Τηλέφωνο να εμφανίζεται το πρόθεμα 2310 και στο αποτέλεσμα η στήλη να έχει όνομα Τηλέφωνο.

Q21

```
Select Όνομα, ('2310'+ Τηλέφωνο) as Τηλέφωνο
From ΠΕΛΑΤΗΣ
```

Όνομα	Τηλέφωνο
Perkins	2310246801
Καντακουζηνός	2310246801
Παλαιολόγος	2310987654

Σημειώνουμε ότι ο όρος **as** χρησιμοποιείται και με τον όρο **from** για τη μετονομασία πινάκων. Έστω, για παράδειγμα, η ερώτηση:

"Να βρεθούν τα ονόματα των πελατών που έχουν το ίδιο τηλέφωνο με αυτό του κ. Perkins (εκτός του ίδιου του Perkins)". Η αντίστοιχη εντολή της SQL με χρήση του όρου **as** είναι:

Q22

```
Select Β.Όνομα
From ΠΕΛΑΤΗΣ as Α, ΠΕΛΑΤΗΣ as Β
Where Α.Τηλέφωνο = Β.Τηλέφωνο and Α.Όνομα = 'Perkins' and Β.Όνομα <> 'Perkins'
```

```
Όνομα
------------------------------
Καντακουζηνός
```

Επισημαίνουμε ότι χωρίς μετονομασία δεν θα μπορούσε να γραφεί στη συνθήκη ότι, για παράδειγμα, ΠΕΛΑΤΗΣ.Τηλέφωνο = ΠΕΛΑΤΗΣ.Τηλέφωνο, γιατί έτσι δεν θα γινόταν η διάκριση. Η πράξη αυτή ονομάζεται και **self join**. Η πράξη self join χρησιμοποιείται όταν πρέπει, εννοιολογικά, να ελεγχθεί κάθε γραμμή ενός πίνακα με όλες τις υπόλοιπες. Έστω, για παράδειγμα, η ερώτηση:

"Να βρεθεί ο κωδικός κάθε ταινίας για την οποία ο δίσκος τύπου BLU-RAY είναι σε μικρότερη τιμή από τον αντίστοιχο δίσκο τύπου DVD". Η αντίστοιχη εντολή SQL είναι:

Q23

```
Select Α.IDΤαινίας
From ΔΙΣΚΟΣ as Α, ΔΙΣΚΟΣ as Β
Where Α.IDΤαινίας = Β.IDΤαινίας and Α.Τύπος = 'BLU-RAY' and Β.Τύπος = 'DVD' and
Α.Τιμή < Β.Τιμή
```

```
IDΤαινίας
-----------
1
```

3.3. Ερωτήματα συνάθροισης και ομαδοποίησης

3.3.1. Ερωτήματα με συναρτήσεις συνάθροισης

Όπως φαίνεται στον Πίνακα 3.14, η SQL περιέχει τις εξής κυριότερες συναρτήσεις συνάθροισης (aggregate functions),:

Συνάρτηση	**Όρος SQL**
Μέσος όρος - Average	**Avg**
Ελάχιστο - Minimum	**Min**
Μέγιστο - Maximum	**Max**
Άθροισμα – Summation	**Sum**
Απαρίθμηση – Count	**Count**

Πίνακας 3.14

Οι συναρτήσεις *αθροίσματος* και *μέσου όρου* δέχονται σαν είσοδο μόνο αριθμητικές τιμές, ενώ οι υπόλοιπες μπορούν να δεχθούν τιμές και άλλων τύπων, όπως αλφαριθμητικά. Για παράδειγμα, έστω το παρακάτω ερώτημα:

"Να βρεθεί η μεγαλύτερη τιμή ενοικίασης ενός δίσκου". Η εντολή της SQL είναι:

Q24

```
Select max(Τιμή) as 'Μέγιστη Τιμή Δίσκου'
From ΔΙΣΚΟΣ
```

```
Μέγιστη Τιμή Δίσκου
--------------------
3,00
```

Ένα δεύτερο παράδειγμα ερωτήματος είναι το παρακάτω :

"Να βρεθεί ο συνολικός αριθμός των δίσκων". Η αντίστοιχη εντολή SQL είναι:

Q25

```
Select count(*)
From ΔΙΣΚΟΣ
```

```
-----------
3
```

Μπορούν να εφαρμοσθούν και αλγεβρικές πράξεις μεταξύ των συναρτήσεων. Για παράδειγμα, έστω το παρακάτω ερώτημα:

"Να βρεθεί ο η διαφορά μεταξύ της ακριβότερης και της φθηνότερης τιμής ενοικίασης ενός δίσκου". Η εντολή της SQL είναι:

Q26

```
Select max(Τιμή) - min(Τιμή)
From ΔΙΣΚΟΣ
```

```
-----------
1,00
```

Η ύπαρξη ορισμάτων εντός μιας συνάρτησης ομαδοποίησης δηλώνει ότι αυτή εφαρμόζεται στο αποτέλεσμα προβολής ως προς αυτά τα ορίσματα. Απαιτείται προσοχή στη χρήση του όρου distinct εντός της συνάρτησης count. Για παράδειγμα, έστω το παρακάτω ερώτημα:

"Να βρεθεί ο αριθμός των πελατών που έχουν κάνει τουλάχιστον μία ενοικίαση ενός δίσκου". Η αντίστοιχη εντολή SQL είναι:

Q27

```
Select count(distinct ΙΔΠελάτη)
From ΕΝΟΙΚΙΑΣΗ
```

```
-----------
2
```

Σε περίπτωση που στο παραπάνω ερώτημα δεν χρησιμοποιούσαμε τον όρο distinct, τότε το ερώτημα θα επέστρεφε ως αποτέλεσμα τον αριθμό 3, που είναι ο συνολικός αριθμός ενοικιάσεων που έχουν γίνει από το DVDclub μας. Συνεπώς, μπορούμε εύκολα να συμπεράνουμε ότι δύο από τις τρεις συνολικά ενοικιάσεις δίσκων DVD έχουν γίνει από τον ίδιο πελάτη.

3.3.2 Ομαδοποίηση των δεδομένων - Ο όρος Group by

Σε αρκετές περιπτώσεις είναι αναγκαίες ενέργειες ο διαμερισμός των εγγραφών μιας σχέσης σε τμήματα και η εφαρμογή μιας συνάρτησης ομαδοποίησης σε κάθε τμήμα. Για τον διαμερισμό χρησιμοποιείται ο όρος **group by**. Ένα παράδειγμα ερώτησης είναι:

"Να βρεθεί ο μέσος όρος τιμής ενοικίασης ανά τύπο δίσκου (BLU-RAY ή DVD)". Η εντολή της SQL είναι:

Q28

```
Select Τύπος, avg(Τιμή) as 'Μέση Τιμή'
From ΔΙΣΚΟΣ
Group by Τύπος
```

Τύπος	Μέση Τιμή
DVD	3,000000
BLU-RAY	2,000000

Μπορούμε να ορίζουμε περισσότερες από μία στήλες στον όρο **group by**. Η τοποθέτηση περισσοτέρων από μία στήλες σημαίνει ότι το σύνολο αποτελέσματος θα ομαδοποιηθεί σύμφωνα με τις στήλες ομαδοποίησης με την σειρά στην οποία εμφανίζονται οι στήλες. Τονίζεται ότι οι στήλες που έχουν χρησιμοποιηθεί ως ορίσματα στο group by, θα πρέπει να χρησιμοποιούνται και στον όρο **Select** αντίστοιχα. Για παράδειγμα, έστω το ερώτημα:

"Για κάθε πελάτη (κωδικός) να βρεθεί ο αριθμός των φορών που ενοικίασε κάθε δίσκο (κωδικός)". Η εντολή SQL είναι:

Q29

```
Select ΙΔΠελάτη, ΙΔΔίσκου, count(ΙΔΔίσκου) as 'Αριθμός Ενοικιάσεων'
From ΕΝΟΙΚΙΑΣΗ
Group by ΙΔΠελάτη, ΙΔΔίσκου
```

ΙΔΠελάτη	ΙΔΔίσκου	Αριθμός Ενοικιάσεων
1	1	1
1	2	1
2	1	1

Παρατηρούμε ότι ο κωδικός πελάτη με τιμή 1 εμφανίζεται περισσότερες από μία φορές, επειδή ομαδοποιείται κάτω από διαφορετικό ΙΔΔίσκου. Για παράδειγμα, έστω το ερώτημα:

"Να βρεθεί η μέση τιμή ενοικίασης ανά τύπο δίσκου και τα αποτελέσματα να είναι ταξινομημένα κατά αύξουσα μέση τιμή". Η εντολή SQL είναι:

Q30

```
Select Τύπος, avg(Τιμή) as 'Μέση Τιμή'
From ΔΙΣΚΟΣ
Group by Τύπος
Order by 'Μέση Τιμή'
```

Τύπος	Μέση Τιμή
BLU-RAY	2,000000
DVD	3,000000

Αν θέλουμε να έχουμε ομαδοποίηση με τον τελεστή count και να εμφανίζονται (με count ίσο με 0) οι εγγραφές που δεν συμμετέχουν, τότε χρησιμοποιούμε μια περίπλοκη σύνταξη, με χρήση του left outer join. Για παράδειγμα, έστω το ερώτημα:

"Να βρεθεί ο αριθμός ενοικιάσεων ανά πελάτη (κωδικός). Στα αποτελέσματα να εμφανίζονται και οι πελάτες που δεν έχουν κάνει κάποια ενοικίαση. Ο αριθμός ενοικιάσεων για αυτούς τους πελάτες να είναι ίσος με 0". Η εντολή SQL είναι:

Q31

```
Select ΠΕΛΑΤΗΣ.ID,count(ΙΔΔίσκου) as 'Αριθμός Ενοικιάσεων'
From ΠΕΛΑΤΗΣ left outer join ΕΝΟΙΚΙΑΣΗ on
ΠΕΛΑΤΗΣ.ID = ΕΝΟΙΚΙΑΣΗ.ΙΔΠελάτη
Group by ΠΕΛΑΤΗΣ.ID
```

ID	Αριθμός Ενοικιάσεων
1	2
2	1
3	0

Η χρήση της προβολής ως προς ΙΔΔίσκου στη συνάρτηση count είναι απαραίτητη, επειδή ο τελεστής count(*) προσμετρά και τις NULL τιμές.

3.3.3. Ο όρος Having

Είναι δυνατό ο διαχωρισμός ενός πίνακα σε τμήματα με τον όρο group by να συνδυαστεί και με κάποια συνθήκη που πρέπει να ικανοποιεί το κάθε ξεχωριστό τμήμα εφόσον ομαδοποιηθεί και μετά. Σε αυτήν την περίπτωση χρησιμοποιείται ο όρος **having** μετά από τον όρο group by, ώστε μια συνθήκη να εξεταστεί αν πληρείται αφότου ομαδοποιηθούν τα αποτελέσματα. Συγκεκριμένα, ο όρος **having** χρησιμοποείται για να ορίσει περιορισμούς που σχετίζονται με την ομαδοποίηση που έχει πραγματοποιηθεί. Έστω, για παράδειγμα, η ερώτηση:

“Να βρεθεί ο τύπος δίσκου για τον οποίο η μέση τιμή ενοικίασης είναι μεγαλύτερη από 2”. Η εντολή SQL είναι:

Q32

```
Select Τύπος, avg(Τιμή) as 'Μέση Τιμή Ενοικίασης'
From ΔΙΣΚΟΣ
Group by Τύπος
Having avg(Τιμή) > 2
```

Τύπος	Μέση Τιμή Ενοικίασης
DVD	3,000000

Τέλος, ένα παράδειγμα ερωτήματος, στο οποίο εμφανίζεται τόσο ο όρος where όσο και ο όρος having, είναι το παρακάτω:

“Να βρεθούν οι κωδικοί των συντελεστών που είναι σκηνοθέτες και που έχουν σκηνοθετήσει περισσότερες από μία ταινίες”. Η εντολή SQL είναι:

Q33

```
Select T_Σ_P.IDΣυντελεστή
From T_Σ_P, ΡΟΛΟΣ
where T_Σ_P.IDΡόλου = ΡΟΛΟΣ.ID and ΡΟΛΟΣ.Περιγραφή = 'Σκηνοθέτης'
Group by T_Σ_P.IDΣυντελεστή
Having count(distinct T_Σ_P.IDΤαινίας) > 1
```

IDΣυντελεστή
1

3.4. Ερωτήματα με πράξεις συνόλων και εμφωλευμένα ερωτήματα

3.4.1. Βασικές πράξεις

Ο όρος **Union** πραγματοποιεί την πράξη της *ένωσης* σχέσεων της σχεσιακής άλγεβρας. Για την εφαρμογή αυτού του όρου μεταξύ δύο σχέσεων πρέπει αυτές να έχουν τον ίδιο αριθμό χαρακτηριστικών και τα πεδία ορισμού των αντίστοιχων χαρακτηριστικών τους να είναι ίδιου τύπου δεδομένων. Για παράδειγμα, έστω η ερώτηση:

"Να βρεθούν οι τίτλοι των ταινιών που γυρίστηκαν το 1959 ή των ταινιών με τύπο δίσκου 'BLU-RAY'". Η αντίστοιχη εντολή SQL είναι:

Q34

```
(Select Τίτλος
From TAINIA
Where Έτος = 1959)

union

(Select Τίτλος
From TAINIA, ΔΙΣΚΟΣ
Where TAINIA.ID = ΔΙΣΚΟΣ.IDΤαινίας and Τύπος = 'BLU-RAY')
```

```
Τίτλος
------------------------
Ben-Hur
Psycho
Rear Window
```

Ο όρος **Intersect** πραγματοποιεί την πράξη της *τομής* σχέσεων της σχεσιακής άλγεβρας. Έστω, για παράδειγμα, η ερώτηση:

"**Να βρεθούν οι τίτλοι των ταινιών που το δεύτερο γράμμα είναι το "e" και που γυρίστηκαν το 1954.**"

Q35

```
Select Τίτλος
From TAINIA
Where Τίτλος like '_e%'

intersect

Select Τίτλος
From TAINIA
Where Χρονιά = '1954'
```

```
Τίτλος
------------------------
Rear Window
```

Ο όρος **Except** πραγματοποιεί την πράξη της *διαφοράς* σχέσεων. Έστω, για παράδειγμα, η ερώτηση:

"**Να βρεθούν οι τίτλοι των ταινιών που το δεύτερο γράμμα τους είναι το "e", εκτός από αυτές που γυρίστηκαν το 1954.**"

Q36

```
Select Τίτλος
From ΤΑΙΝΙΑ
Where Τίτλος like '_e%'

except

Select Τίτλος
From ΤΑΙΝΙΑ
Where Χρονιά = '1954'
```

```
Τίτλος
------------------------
Ben-Hur
```

Να σημειώσουμε ότι οι πράξεις **union**, **intersect** και **except** απαλείφουν αυτόματα τα διπλότυπα. Επίσης, να τονίσουμε ότι θα πρέπει να υπάρχει συμβατότητα μεταξύ των δύο μελών των παραπάνω πράξεων. Επομένως, πρέπει αφενός να έχουμε ίδιο αριθμό χαρακτηριστικών και αφετέρου τα πεδία ορισμού των αντίστοιχων χαρακτηριστικών να είναι ίδιου τύπου δεδομένων, προκειμένου να είναι συντακτικά σωστό ένα ερώτημα SQL.

3.4.2. Εμφωλευμένα ερωτήματα

Οι εντολές της SQL έχουν την ιδιότητα της *κλειστότητας,* δηλαδή το αποτέλεσμα οποιαδήποτε πράξης μεταξύ δυο ή περισσοτέρων πινάκων οδηγεί σε έναν νέο πίνακα. Μ' αυτόν τον τρόπο είναι δυνατό να εμφωλιαστούν εντολές, ώστε το αποτέλεσμα μιας πράξης να είναι είσοδος σε μια άλλη πράξη. Η πιο συχνή χρήση αυτής της ιδιότητας γίνεται για έλεγχο συνθηκών μεταξύ συνόλων. Η SQL περιέχει τον όρο **in,** ο οποίος ελέγχει αν μια γραμμή ανήκει σε έναν πίνακα που είναι αποτέλεσμα μιας φωλιασμένης εντολής. Αντιστοιχεί στον μαθηματικό τελεστή συνόλων «ανήκει» (σύμβολο $\in$). Έστω, για παράδειγμα, η ερώτηση:

"Να βρεθούν οι πελάτες που έχουν κάνει ενοικίαση τουλάχιστον ενός δίσκου". Αυτή η ερώτηση μπορεί να απαντηθεί με την πράξη της σύνδεσης (join), όπως έχει ήδη αναφερθεί. Παρόλα αυτά, θέλουμε να εξεταστεί η απάντηση με χρήση του όρου **in**.

Q37

```
Select Όνομα
From ΠΕΛΑΤΗΣ
Where ID in (Select ΙΔΠελάτη From ΕΝΟΙΚΙΑΣΗ)
```

```
Όνομα
------------------------------
Perkins
Καντακουζηνός
```

Εκτός από τον όρο **in**, μπορεί να χρησιμοποιηθεί και ο όρος **not in,** ο οποίος ελέγχει αν μια εγγραφή δεν ανήκει σε μια σχέση. Ο όρος **in** μπορεί να χρησιμοποιηθεί και για σύνολα απαρίθμησης (enumerated sets). Έστω, για παράδειγμα, η ερώτηση:

"Να βρεθούν όλοι οι συντελεστές που δεν ονομάζονται Alfred Hitchcock ή Grace Kelly". Η εντολή SQL είναι :

Q38

```
Select Όνομα
From ΣΥΝΤΕΛΕΣΤΗΣ
Where Όνομα not in ('Alfred Hitchcock', 'Grace Kelly')
```

```
Όνομα
----------------------
Anthony Perkins
```

Η χρήση του όρου **in** βοηθά στον επιμερισμό της σύνταξης μίας SQL επερώτησης σε τμήματα, επομένως στην απλούστευσή της. Έστω, για παράδειγμα, η ερώτηση:

"Να βρεθούν οι κωδικοί των ταινιών στις οποίες έχει συμμετάσχει ο Alfred Hitchcock και έχουν ενοικιασθεί περισσότερες από δύο φορές". Η εντολή SQL είναι :

Q39

```
Select ΔΙΣΚΟΣ.IDΤαινίας
From ΔΙΣΚΟΣ inner join ΕΝΟΙΚΙΑΣΗ on
ΕΝΟΙΚΙΑΣΗ.IDΔίσκου = ΔΙΣΚΟΣ.ID Where ΔΙΣΚΟΣ.IDΤαινίας IN
    (Select IDΤαινίας From Τ_Σ_Ρ inner join ΣΥΝΤΕΛΕΣΤΗΣ on
    Τ_Σ_Ρ.IDΣυντελεστή = ΣΥΝΤΕΛΕΣΤΗΣ.ID
    where ΣΥΝΤΕΛΕΣΤΗΣ.Όνομα = 'Alfred Hitchcock')
    Group by ΔΙΣΚΟΣ.IDΤαινίας
    Having count(*) > 2
```

```
IDΤαινίας
-----------
1
```

3.4.3. Σύγκριση μεταξύ συνόλων

Σε κάποιες περιπτώσεις προκύπτουν ερωτήματα στα οποία ένα πεδίο πρέπει να συγκριθεί με την τιμή του ίδιου πεδίου σε τουλάχιστον μία άλλη εγγραφή ενός πίνακα. Σ' αυτήν την περίπτωση χρησιμοποιείται ο όρος **some** (ισοδύναμος είναι ο όρος **any**). Αν πρέπει να συγκριθεί με την τιμή του πεδίου *όλων* των γραμμών της σχέσης, τότε χρησιμοποιείται ο όρος **all**. Έστω, για παράδειγμα, το παρακάτω ερώτημα:

"Να βρεθούν οι πελάτες (Όνομα) για τους οποίους υπάρχει κάποιος ενοικιασμένος δίσκος χωρίς ορισμένη ημερομηνία επιστροφής". Η εντολή SQL είναι :

Q40

```
Select Όνομα
From ΠΕΛΑΤΗΣ
Where ID = some
    (Select IDΠελάτη
    From ΕΝΟΙΚΙΑΣΗ
    where Έως IS NULL)
```

```
Όνομα
------------------------------
Καντακουζηνός
```

Οι όροι **all** και **some** μπορούν να χρησιμοποιηθούν και με συναρτήσεις ομαδοποίησης. Ειδικότερα, επειδή οι όροι **count** και **max** δεν μπορούν να χρησιμοποιηθούν με φωλιασμένο τρόπο, δηλαδή **max**(**count**(*)), είναι δυνατό να χρησιμοποιηθεί ο όρος **all** για να δηλώσει ότι μέγιστη τιμή είναι αυτή που είναι μεγαλύτερη από όλες. Έστω, για παράδειγμα, το παρακάτω ερώτημα:

"Να βρεθεί ο κωδικός πελάτη με το μεγαλύτερο αριθμό ενοικιάσεων". Η εντολή SQL είναι :

Q41

```
Select ΙΔΠελάτη
from ΕΝΟΙΚΙΑΣΗ
group by ΙΔΠελάτη
having count(*) >= all
     ( Select count(*)
     from ΕΝΟΙΚΙΑΣΗ
     group by ΙΔΠελάτη )
```

```
ΙΔΠελάτη
-------------
1
```

3.4.4. Έλεγχος κενότητας

Η SQL περιέχει τους όρους **exists** και **not exists,** οι οποίοι ελέγχουν, αντίστοιχα, αν μια άλλη εντολή SQL παράγει σαν αποτέλεσμα μία σχέση που έχει εγγραφές ή αν είναι άδεια,. Έστω, για παράδειγμα, το παρακάτω ερώτημα:

"Να βρεθούν τα ονόματα των πελατών που έχουν κάνει μία τουλάχιστον ενοικίαση". Η εντολή SQL με τον όρο exists είναι:

Q42

```
Select Όνομα
from ΠΕΛΑΤΗΣ
where exists
    (select ΙΔΠελάτη
    from ΕΝΟΙΚΙΑΣΗ
    where ΙΔΠελάτη = ΠΕΛΑΤΗΣ.ID)
```

```
Όνομα
------------------------
Perkins
Καντακουζηνός
```

Προσοχή: Εναλλακτικά το παραπάνω ερώτημα θα μπορούσε να γίνει και με την χρήση του όρου **inner join**.

Ο όρος **not exists** μπορεί να χρησιμοποιηθεί για να ελεγχθεί αν μια σχέση *Y* περιέχει μια άλλη σχέση *X*, δηλαδή είναι υπερσύνολό της. Βασισμένοι στη συνολοθεωρία, γνωρίζουμε ότι: $X \subseteq Y \Leftrightarrow X - Y = \emptyset$. Σε αυτή την περίπτωση, δεν υπάρχει εγγραφή που να ανήκει στην *X* που να μην ανήκει στην *Y*. Έστω, για παράδειγμα, το παρακάτω ερώτημα:

"Να δοθούν οι κωδικοί των πελατών που έχουν ενοικιάσει τουλάχιστον όλες τις ταινίες που έχει ενοικιάσει ο πελάτης με κωδικό 2 (αυτός να μην εμφανίζεται στο αποτέλεσμα)". Η εντολή SQL είναι :

Q43

```
Select distinct E1.IDΠελάτη
from ΕΝΟΙΚΙΑΣΗ as E1
where not exists
    (select E2.IDΔίσκου
    from ΕΝΟΙΚΙΑΣΗ as E2
    where E2.IDΠελάτη = 2 and
    E2.IDΔίσκου not in
        (select E3.IDΔίσκου
        from ΕΝΟΙΚΙΑΣΗ as E3
        where E3.IDΠελάτη = E1.IDΠελάτη))
        and E1.IDΠελάτη <> 2
```

```
IDΠελάτη
----------
1
```

3.5. Ερωτήματα SQL για όψεις

Η όψη (view) είναι ένα αφηρημένο υποσύνολο της βάσης δεδομένων (υπό μορφή πίνακα) που αντιστοιχεί σε ένα τμήμα ενός πίνακα της βάσης δεδομένων ή σε αποτέλεσμα ενός ερωτήματος που αφορά πολλούς πίνακες. Ο ορισμός όψης στην SQL γίνεται με τη δήλωση **create view**. Για παράδειγμα, έστω το ερώτημα:

"Να ορισθεί όψη με όνομα MyView, που περιέχει όλους τους κωδικούς των ταινιών που συμμετείχε ο Alfred Hitchcock". Η εντολή SQL είναι :

Q44

```
Create View MyView as
(Select T_Σ_P.IDΤαινίας
 from ΣΥΝΤΕΛΕΣΤΗΣ inner join T_Σ_P ON
ΣΥΝΤΕΛΕΣΤΗΣ.Id = T_Σ_P.IDΣυντελεστή
where ΣΥΝΤΕΛΕΣΤΗΣ.Όνομα = 'Alfred Hitchcock')
```

Η διαγραφή μιας όψης γίνεται με τη δήλωση **drop view**. Έστω η παρακάτω εντολή:

"Διαγράψτε την όψη MyView". Η αντίστοιχη εντολή SQL είναι:

Q45

```
drop view MyView
```

Χάρη στις όψεις έχουμε αποθηκευμένο ένα ερώτημα που μπορούμε να το χρησιμοποιήσουμε παραπέρα σαν τμήμα/ κομμάτι, για να χτίσουμε πιο σύνθετα ερωτήματα. Για παράδειγμα, μέσω της χρήσης της όψης MyView, μπορούμε να απλοποιήσουμε το παρακάτω ερώτημα που έχει ήδη εμφανιστεί ως ερώτημα Q4 στην Ενότητα 3.4.2:

"Να βρεθούν οι κωδικοί των ταινιών στις οποίες έχει συμμετάσχει ο Alfred Hitchcock και έχουν ενοικιασθεί περισσότερες από δύο φορές ". Η αντίστοιχη εντολή SQL είναι:

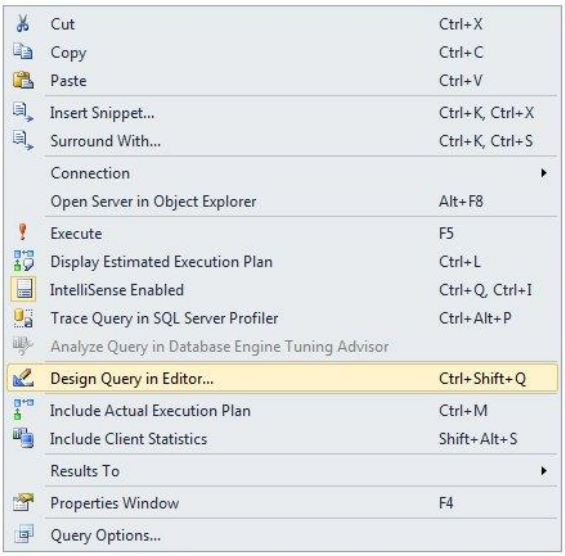
Q46

```
Select ΔΙΣΚΟΣ.IDΤαινίας
From ΔΙΣΚΟΣ inner join ΕΝΟΙΚΙΑΣΗ on ΕΝΟΙΚΙΑΣΗ.IDΔίσκου = ΔΙΣΚΟΣ.ID Where
ΔΙΣΚΟΣ.IDΤαινίας IN (select * from MyView)
 Group by ΔΙΣΚΟΣ.IDΤαινίας
 Having count(*) > 2
```

```
IDΤαινίας
-----------
1
```

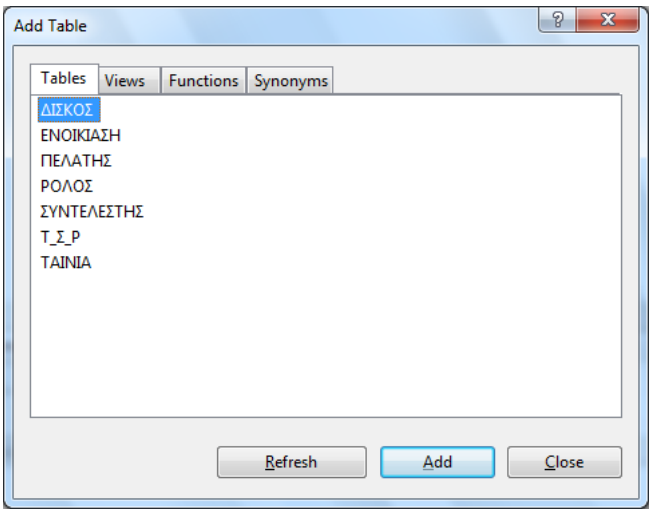
3.6. Το εργαλείο Query Designer για δημιουργία ερωτημάτων Query by Example

Η σύνταξη όλων των ερωτημάτων μέχρι τώρα έγινε στον Query Editor. Όμως, αυτός ο τρόπος σύνταξης SQL ερωτημάτων θεωρείται αργός και δύσκολος, διότι πρέπει να πληκτρολογούμε όλο το ερώτημα και, ταυτόχρονα, να θυμόμαστε και τα ονόματα των πεδίων/ πινάκων που θα συμμετάσχουν σ' αυτό. Προκειμένου να αντιμετωπιστούν τα παραπάνω προβλήματα, υπάρχει η δυνατότητα να συντάξουμε τα ερωτήματά μας με οπτικό (visual) τρόπο. Συγκεκριμένα, από την γραμμή εργαλείων επιλέγουμε την βάση δεδομένων DVDclub ως την ενεργή βάση δεδομένων μας και κάνουμε κλικ στο New Query. Στη συνέχεια, με δεξί κλικ μέσα στον χώρο σύνταξης του ερωτήματος εμφανίζεται το μενού της Εικόνας 3.2, στο οποίο επιλέγουμε Design Query in Editor.



Εικόνα 3.2

Στη συνέχεια, εμφανίζεται το αναδυόμενο παράθυρο επιλογής των πινάκων που σχετίζονται με την σύνταξη του ερωτήματος μας, όπως φαίνεται στην Εικόνα 3.3.

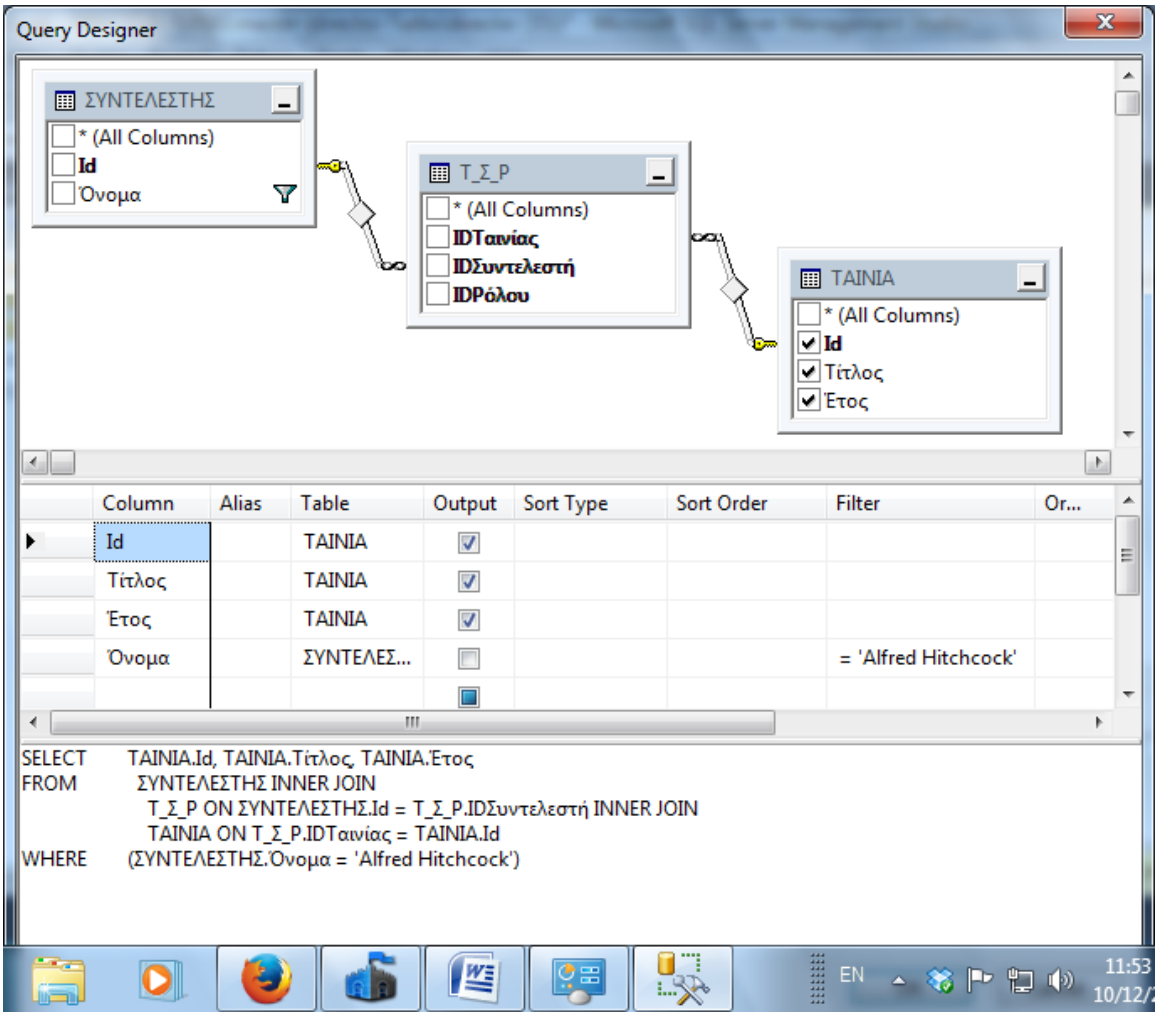


Εικόνα 3.3

Στο παράδειγμά μας θα υλοποιήσουμε το παρακάτω ερώτημα:

Q47	**«Να εμφανίσετε τα στοιχεία των ταινιών στα οποία συμμετείχε ως συντελεστής ο Alfred Hitchcock».**

Για την απάντηση του παραπάνω ερωτήματος πρέπει να αντλήσουμε δεδομένα από τρεις συσχετιζόμενους πίνακες. Συνεπώς, από τους διαθέσιμους πίνακες της Εικόνας 3.3 επιλέγουμε τους τρεις πίνακες που θα συμμετάσχουν στο ερώτημα μας (ΤΑΙΝΙΑ, Τ_Σ_Ρ, ΣΥΝΤΕΛΕΣΤΗΣ). Όπως φαίνεται στην Εικόνα 3.4, οι συσχετίσεις τους εμφανίζονται αυτόματα. Με Drag & Drop μπορούμε να προσαρμόζουμε την θέση των πινάκων όπως επιθυμούμε. Ο Query Designer ενσωματώνει ένα grid τύπου excel, το οποίο έχει μια σειρά από στήλες (Column, Alias, Table, Output, Sort Type, Sort Order, Filter, Or, κτλ.). Σ' αυτό το grid θα δημιουργήσουμε το ερώτημά μας, τσεκάροντας τα αντίστοιχα πεδία από τους συσχετιζόμενους πίνακες. Επιλέγοντας, λοιπόν, τα πεδία (ID, Τίτλος, Έτος) από τον πίνακα ΤΑΙΝΙΑ, αυτά εμφανίζονται αυτόματα στις στήλες column και output του grid. Στη συνέχεια, αφού επιλέξουμε το πεδίο Όνομα από τον πίνακα ΣΥΝΤΕΛΕΣΤΗΣ, πληκτρολογούμε το όνομα του συντελεστή στην στήλη Filter, όπως φαίνεται στην Εικόνα 3.4. Το ερώτημα έχει πλέον δημιουργηθεί και εμφανίζεται σε εντολές SQL στο κάτω μέρος της Εικόνας 3.4.



Εικόνα 3.4

Παρακάτω περιγράφουμε τις βασικές στήλες του grid:

- Η στήλη Alias ορίζει προαιρετικά το τελικό όνομα του πεδίου όπως αυτό θα εμφανιστεί στα αποτελέσματα, εφόσον θέλουμε να αλλάξουμε το όνομα του με κάποιο άλλο.
- Η στήλη Output ορίζει αν το πεδίο θα εμφανιστεί στα τελικά αποτελέσματα ή αν θα συμμετάσχει μόνο ως φίλτρο.
- Οι στήλες Sort Type (Asc, Desc – Αύξουσα, Φθίνουσα) και Sort Order διαμορφώνουν τον όρο Order By της SQL.
- Τέλος, οι στήλες Filter και Or μάς επιτρέπουν να ορίσουμε τα φίλτρα ενός ή περισσοτέρων πεδίων. Σε κάθε επόμενη στήλη μπορούμε να ορίσουμε μια νέα τιμή ενός πεδίου μέσω του τελεστή Or.

Για να εκτελέσουμε, λοιπόν, το προαναφερθέν ερώτημά μας, πατάμε OK και, στη συνέχεια, επιλέγουμε F5. Παρακάτω εμφανίζονται τα αποτελέσματα, σύμφωνα με τα οποία ο Alfred Hitchcock έχει συμμετάσχει ως συντελεστής σε δύο ταινίες.

```
ID  Τίτλος          Έτος
----  ---------------     ------------
1   Rear Window     1954
2   Psycho          1960

(2 row(s) affected)
```

Επισημαίνουμε ότι για να διορθώσουμε κάτι που μας διέφυγε στο ερώτημά μας, πρέπει να επιλέξουμε με το ποντίκι όλες τις εντολές (ή τμήμα των εντολών) του ερωτήματος SQL και, με δεξί κλικ πάνω από όλη την επιλεγμένη περιοχή, να επιλέξουμε ξανά Design Query in Editor… Μ΄αυτόν τον τρόπο, ο Editor του Management Studio επανασχεδιάζει οπτικά το ερώτημά μας, ώστε να συνεχίσουμε εκ νέου.

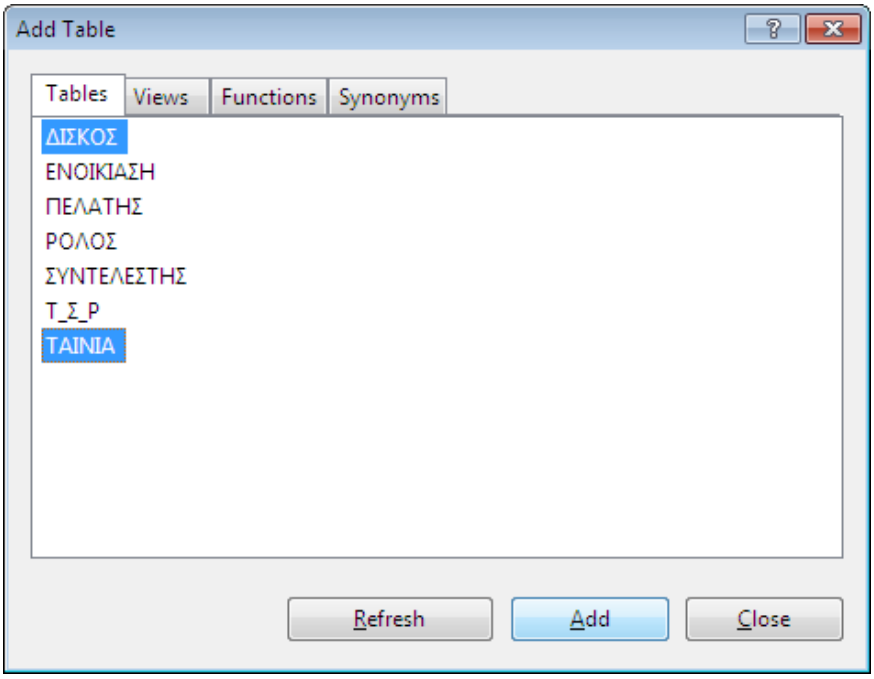
Συμπερασματικά, με τον οπτικό (visual) τρόπο μπορούμε να κερδίσουμε πολύ χρόνο στην σύνταξη ερωτημάτων, αποφεύγοντας την πληκτρολόγηση και διατηρώντας την μόνο εκεί που είναι απαραίτητη. Δεν πρέπει, όμως, να πιστέψουμε ότι, επειδή υπάρχει ένα εύκολο οπτικό εργαλείο, δεν χρειάζεται να γνωρίζουμε τους κανόνες σύνταξης ερωτημάτων SQL.

Τέλος, παρακάτω παρουσιάζουμε ένα παράδειγμα δημιουργίας ερωτήματος ομαδοποίησης με QBE:

Q48 **«Για κάθε ταινία (τίτλος) και τύπο δίσκου (είτε BLU-RAY είτε DVD) να βρεθεί ο αριθμός αντιτύπων που περιέχουν την ταινία.».**

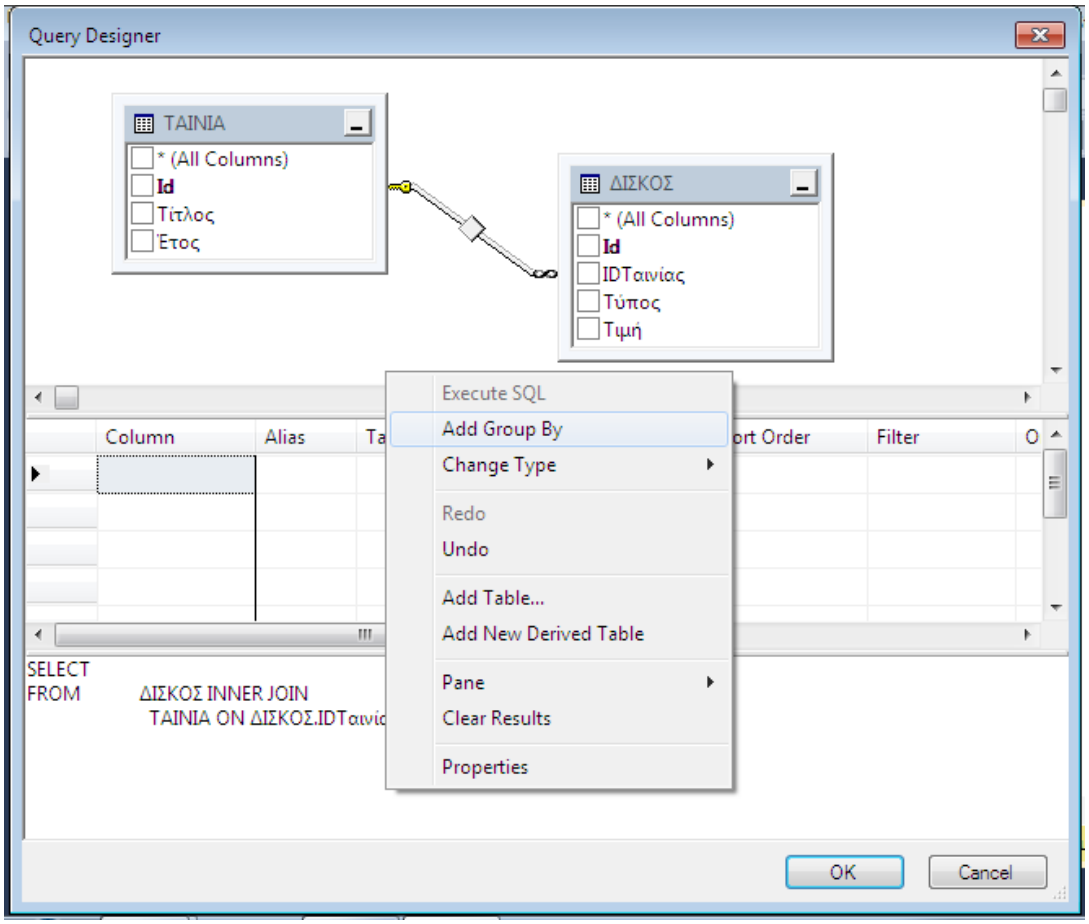
Τίτλος	Τύπος	Αριθμός
---	---	---
Psycho	BLU-RAY	1
Rear Window	BLU-RAY	1
Rear Window	DVD	1

Για να πάρουμε το παραπάνω αποτέλεσμα, κάνουμε δεξί κλικ μέσα στον επεξεργαστή ερωτημάτων. Εκεί, αφού επιλέξουμε Design Query in Editor, εμφανίζεται ένα αναδυόμενο παράθυρο επιλογής των πινάκων (ΔΙΣΚΟΣ και ΤΑΙΝΙΑ) που σχετίζονται με την σύνταξη του ερωτήματός μας, όπως φαίνεται στην Εικόνα 3.5.



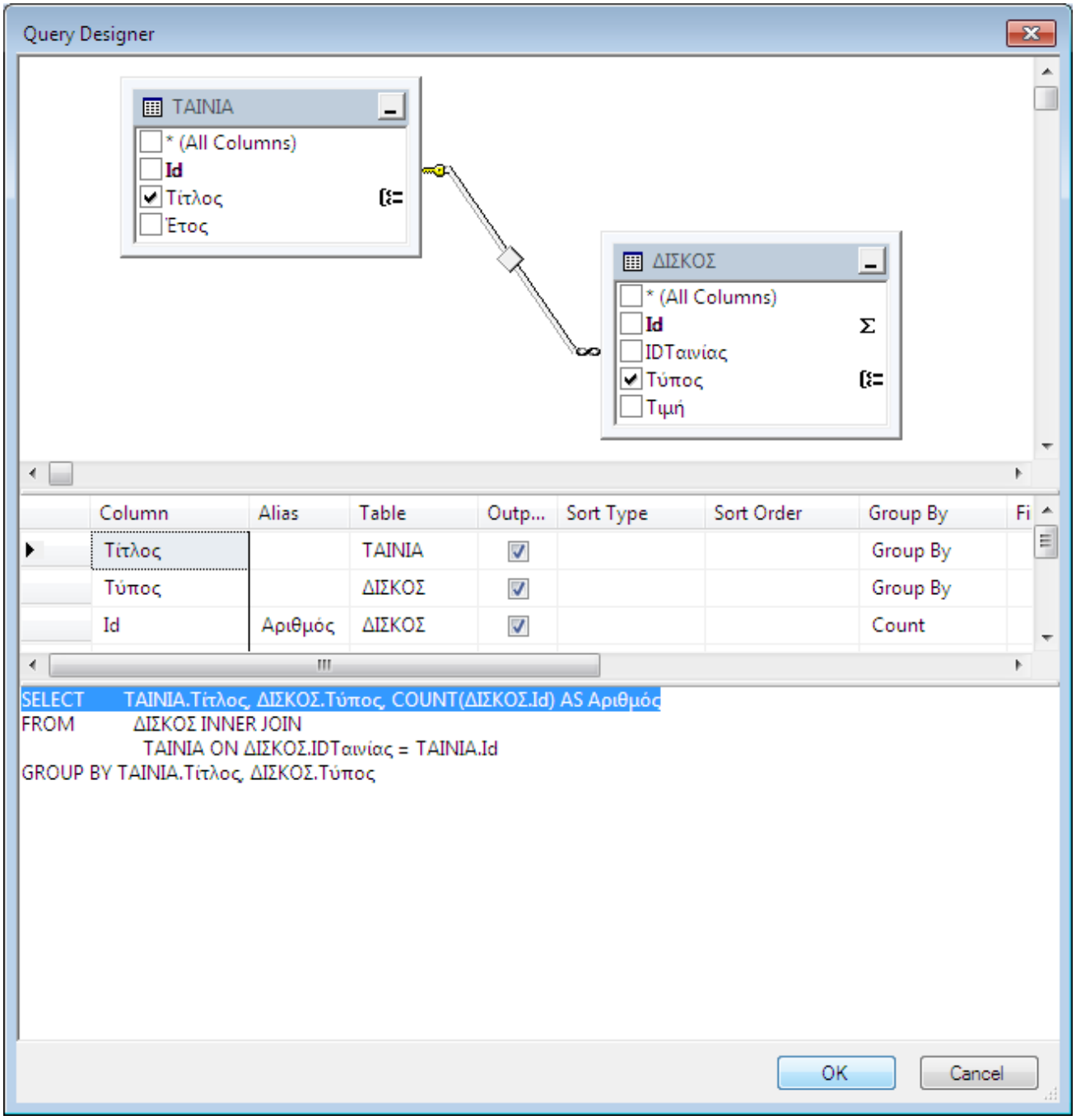
Εικόνα 3.5

Στη συνέχεια, όπως φαίνεται στην Εικόνα 3.6, εμφανίζεται το παράθυρο με την συσχέτιση των δύο πινάκων. Εκεί, αφού κάνουμε δεξί κλικ, επιλέγουμε Add Group by, επειδή θέλουμε να ομαδοποιήσουμε τα αποτελέσματα μας βάσει τίτλου ταινίας και τύπου δίσκου.



Εικόνα 3.6

Τέλος, αφού ομαδοποιήσουμε τα αποτελέσματά μας βάσει των πεδίων Τίτλος και Τύπος (όπως φαίνεται στην Εικόνα 3.7), εφαρμόζουμε Count στο πεδίο Id του πίνακα ΔΙΣΚΟΣ.



Εικόνα 3.7

3.7. Ασκήσεις με ερωτήματα SQL

3.7.1. Ασκήσεις με ερωτήματα επιλογής γραμμών από ένα πίνακα

1. Για κάθε δίσκο, να προβληθεί ο κωδικός και η τιμή. Η τιμή να εμφανίζεται χωρίς το ΦΠΑ, δηλαδή μειωμένη κατά 0.23*Τιμή.
2. Να προβληθούν οι κωδικοί των συντελεστών που έχουν συμμετάσχει σε τουλάχιστον μία ταινία. Κάθε κωδικός να εμφανίζεται μία φορά (όχι διπλοεγγραφές).
3. Να προβληθούν οι ταινίες (όλα τα στοιχεία) που ο τίτλος τους περιέχει το χαρακτήρα "-" ή έχουν γυριστεί πριν το 1955.
4. Να βρεθούν όλα τα στοιχεία των ενοικιάσεων που έχουν γίνει (πεδίο Από) μεταξύ 15 Σεπ 2006 και 30 Σεπ 2006.
5. Να βρεθούν οι κωδικοί των δίσκων που έχουν ενοικιασθεί και έχουν επιστραφεί, θεωρώντας ότι το πεδίο Έως του πίνακα ΕΝΟΙΚΙΑΣΗ ενημερώνεται με την επιστροφή του δίσκου.

3.7.2. Ασκήσεις με ερωτήματα επιλογής γραμμών από πολλούς πίνακες

6. Να βρεθούν τα ονόματα των πελατών που έχουν ενοικιάσει τουλάχιστον έναν δίσκο.
7. Να βρεθούν τα ονόματα των πελατών που δεν έχουν ενοικιάσει ούτε ένα ψηφιακό δίσκο (Σημείωση: Να μην χρησιμοποιηθεί ο τελεστής NOT IN).
8. Να βρεθούν οι κωδικοί των συντελεστών που έχουν συμμετάσχει σε τουλάχιστον 2 ταινίες (Σημείωση: Να μην χρησιμοποιηθεί ο τελεστής COUNT).
9. Να βρεθούν οι τίτλοι των ταινιών για τις οποίες είτε δεν υπάρχει δίσκος είτε υπάρχει σχετικός δίσκος που δεν έχει ενοικιαστεί ποτέ (Σημείωση: Να μην χρησιμοποιηθεί ο τελεστής NOT IN).
10. Να βρεθούν οι πελάτες με επίθετο ίδιο με αυτό κάποιου συντελεστή ταινίας.

3.7.3. Ασκήσεις με Ερωτήματα ομαδοποίησης/συνάθροισης δεδομένων

11. Να βρεθεί ο αριθμός των ταινιών που έχει συμμετάσχει ο Alfred Hitchcock. (Σημείωση: Οι ταινίες στις οποίες έχει συμμετάσχει με περισσότερους από έναν ρόλους να προσμετρούνται μία μόνο φορά).
12. Για κάθε ταινία (τίτλος), να βρεθεί ο συνολικός αριθμός δίσκων (BLU-RAY και DVD) που περιέχουν την ταινία. Στο αποτέλεσμα να εμφανίζονται και οι ταινίες για τις οποίες δεν υπάρχει κανένας δίσκος.
13. Να βρεθούν οι κωδικοί των δίσκων που είναι τύπου BLU-RAY και έχουν ενοικιασθεί περισσότερες από μία φορές.

3.7.4. Ασκήσεις με ερωτήματα με φωλιασμένες εντολές SQL

14. Να βρεθούν οι τίτλοι των ταινιών που δεν έχουν ενοικιασθεί ποτέ. (Σημείωση: Να μην χρησιμοποιηθεί outer join).
15. Να βρεθεί το όνομα του Συντελεστή που έχει συμμετάσχει στις περισσότερες ταινίες.

3.8. Λύσεις ασκήσεων με ερωτήματα SQL

3.8.1. Λύσεις ασκήσεων με ερωτήματα επιλογής γραμμών από ένα πίνακα

1. Για κάθε δίσκο, να προβληθεί ο κωδικός και η τιμή. Η τιμή να εμφανίζεται χωρίς το ΦΠΑ, δηλαδή μειωμένη κατά 0.23*Τιμή.

```
Select  ID, Τιμή/1.23 as 'Αποφορολογημένη Τιμή'
from    ΔΙΣΚΟΣ
```

ID	Αποφορολογημένη Τιμή
1	1.626016
2	2.439024
3	1.626016

2. Να προβληθούν οι κωδικοί των συντελεστών που έχουν συμμετάσχει σε τουλάχιστον μία ταινία. Κάθε κωδικός να εμφανίζεται μία φορά (όχι διπλοεγγραφές).

```
Select  distinct IDΣυντελεστή
from    T_Σ_P
```

IDΣυντελεστή
1
2
3

3. Να προβληθούν οι ταινίες (όλα τα στοιχεία) που ο τίτλος τους περιέχει το χαρακτήρα '-' ή έχουν γυριστεί πριν το 1955.

```
select  *
from    ΤΑΙΝΙΑ
where   Τίτλος like '%-%' or Έτος < 1955
```

Id	Τίτλος	Έτος
1	Rear Window	1954
3	Ben-Hur	1959

4. Να βρεθούν όλα τα στοιχεία των ενοικιάσεων που έχουν γίνει (πεδίο Από) μεταξύ 15 Σεπ 2006 και 30 Σεπ 2006.

```
Select  *
from    ΕΝΟΙΚΙΑΣΗ
where   Από between '2006-09-15' and '2006-09-30'
```

IDΠελάτη	IDΔίσκου	Από	Έως
1	2	2006-09-20	2006-11-20

5. Να βρεθούν οι κωδικοί των δίσκων που έχουν ενοικιασθεί και έχουν επιστραφεί, θεωρώντας ότι το πεδίο Έως του πίνακα ΕΝΟΙΚΙΑΣΗ ενημερώνεται με την επιστροφή του δίσκου.

```
Select IDΔίσκου
from ΕΝΟΙΚΙΑΣΗ
where Έως IS NOT NULL

IDΔίσκου
-----------
1
2
```

3.8.2. Λύσεις ασκήσεων με ερωτήματα επιλογής γραμμών από πολλούς πίνακες

6. Να βρεθούν τα ονόματα των πελατών που έχουν ενοικιάσει τουλάχιστον ένα δίσκο.

```
Select   distinct ΠΕΛΑΤΗΣ.Όνομα
From    ΠΕΛΑΤΗΣ inner join ΕΝΟΙΚΙΑΣΗ on
ΕΝΟΙΚΙΑΣΗ.IDΠελάτη = ΠΕΛΑΤΗΣ.ID

Όνομα
------------------------------
Perkins
Καντακουζηνός
```

7. Να βρεθούν τα ονόματα των πελατών που δεν έχουν ενοικιάσει ούτε ένα δίσκο (Σημείωση: Να μην χρησιμοποιηθεί ο τελεστής NOT IN).

```
Select   ΠΕΛΑΤΗΣ.Όνομα
From    ΠΕΛΑΤΗΣ left outer join ΕΝΟΙΚΙΑΣΗ on
ΕΝΟΙΚΙΑΣΗ.IDΠελάτη = ΠΕΛΑΤΗΣ.ID
Where  ΕΝΟΙΚΙΑΣΗ.IDΠελάτη IS NULL

Όνομα
------------------------------
Παλαιολόγος
```

8. Να βρεθούν οι κωδικοί των συντελεστών που έχουν συμμετάσχει σε τουλάχιστον 2 ταινίες (Σημείωση: Να μην χρησιμοποιηθεί ο τελεστής COUNT).

```
Select   distinct ΤΣΡ1.IDΣυντελεστή
From    Τ_Σ_Ρ as ΤΣΡ1, Τ_Σ_Ρ as ΤΣΡ2
Where  ΤΣΡ1.IDΣυντελεστή = ΤΣΡ2.IDΣυντελεστή and
ΤΣΡ1.IDΤαινίας <> ΤΣΡ2.IDΤαινίας

IDΣυντελεστή
------------
1
```

9. Να βρεθούν οι τίτλοι των ταινιών για τις οποίες είτε δεν έχουν αποθηκευθεί σε δίσκο είτε υπάρχει δίσκος που δεν έχει ενοικιαστεί ποτέ (Σημείωση: Να μην χρησιμοποιηθεί ο τελεστής NOT IN).

```
Select  Τίτλος
From    TAINIA left outer join ΔΙΣΚΟΣ on ΔΙΣΚΟΣ.IDΤαινίας = TAINIA.ID
left outer join ΕΝΟΙΚΙΑΣΗ on ΕΝΟΙΚΙΑΣΗ.IDΔίσκου = ΔΙΣΚΟΣ.ID
Where   ΕΝΟΙΚΙΑΣΗ.IDΔίσκου IS NULL
```

```
Τίτλος
-----------------
Psycho
Ben-Hur
```

10. Να βρεθούν οι πελάτες με όνομα παρόμοιο με αυτό κάποιου συντελεστή ταινίας.

```
Select  ΠΕΛΑΤΗΣ.Όνομα
From    ΠΕΛΑΤΗΣ, ΣΥΝΤΕΛΕΣΤΗΣ
Where   ΣΥΝΤΕΛΕΣΤΗΣ.Όνομα like ('%' + ΠΕΛΑΤΗΣ.Όνομα + '%')
```

```
Όνομα
--------------
Perkins
```

3.8.3. Λύσεις ασκήσεων με ερωτήματα ομαδοποίησης/συνάθροισης δεδομένων

11. Να βρεθεί ο αριθμός των ταινιών που έχει συμμετάσχει ο Alfred Hitchcock. (Σημείωση: Οι ταινίες στις οποίες έχει συμμετάσχει με περισσότερους από έναν ρόλους να προσμετρούνται μία μόνο φορά).

```
Select count(distinct Τ_Σ_Ρ.IDΤαινίας) as 'Αριθμός Ταινιών'
From Τ_Σ_Ρ inner join ΣΥΝΤΕΛΕΣΤΗΣ on Τ_Σ_Ρ.IDΣυντελεστή = ΣΥΝΤΕΛΕΣΤΗΣ.ID
Where ΣΥΝΤΕΛΕΣΤΗΣ.Όνομα = 'Alfred Hitchcock'
```

```
Αριθμός Ταινιών
---------------------
2
```

12. Για κάθε ταινία (τίτλος), να βρεθεί ο συνολικός αριθμός δίσκων (BLU-RAY και DVD) που περιέχει την ταινία. Στο αποτέλεσμα να εμφανίζονται και οι ταινίες για τις οποίες δεν δίσκος.

```
Select TAINIA.Τίτλος, count(ΔΙΣΚΟΣ.Τύπος) as 'Αριθμός'
From TAINIA left outer join ΔΙΣΚΟΣ on ΔΙΣΚΟΣ.IDΤαινίας = TAINIA.ID
Group By TAINIA.ID, TAINIA.Τίτλος
```

```
Τίτλος              Αριθμός
--------------------  -----------
Rear Window         2
Psycho              1
Ben-Hur             0
```

13. Να βρεθούν οι κωδικοί των δίσκων που είναι τύπου BLU-RAY και έχουν ενοικιασθεί περισσότερες από μία φορές.

```
Select ΔΙΣΚΟΣ.ID
From ΔΙΣΚΟΣ inner join ΕΝΟΙΚΙΑΣΗ on ΔΙΣΚΟΣ.ID = ΕΝΟΙΚΙΑΣΗ.IDΔίσκου
Where ΔΙΣΚΟΣ.Τύπος = 'BLU-RAY'
Group by ΔΙΣΚΟΣ.ID
Having count(*) > 1

ID
-----------
1
```

3.8.4. Λύσεις ασκήσεων με ερωτήματα με φωλιασμένες εντολές SQL

14. Να βρεθούν οι τίτλοι των ταινιών που δεν έχουν ενοικιασθεί ποτέ. (Σημείωση: Να μην χρησιμοποιηθεί outer join).

```
Select   Τίτλος
From   TAINIA
where not exists (select ΔΙΣΚΟΣ.IDΤαινίας
from ΕΝΟΙΚΙΑΣΗ inner join ΔΙΣΚΟΣ on ΕΝΟΙΚΙΑΣΗ.IDΔίσκου = ΔΙΣΚΟΣ.ID
where IDΤαινίας = TAINIA.ID)

Τίτλος
----------------
Psycho
Ben-Hur
```

15. Να βρεθεί το όνομα του Συντελεστή που έχει συμμετάσχει στις περισσότερες ταινίες.

```
Select Όνομα
From   ΣΥΝΤΕΛΕΣΤΗΣ
where ID in
        (Select IDΣυντελεστή
        from T_Σ_P
        Group by IDΣυντελεστή
        Having count(IDΤαινίας) >= all
                (Select  count(*)
                From T_Σ_P
                Group by IDΣυντελεστή
                )
        )

Όνομα
--------------------------------------------------
Alfred Hitchcock
```

3.9. Βιβλιογραφία/Αναφορές

Hoffer, J. A., Venkatarama, R., & Topi, H. (2013). *Modern Database Management*, Prentice Hall.

Μανωλόπουλος, Ι., & Παπαδόπουλος, Α. Ν. (2006). *Συστήματα Βάσεων Δεδομένων: Θεωρία & Πρακτική Εφαρμογή*, Αθήνα, Εκδόσεις Νέων Τεχνολογιών.

Κεφάλαιο 4. Προχωρημένες λειτουργίες στον SQL Server

Σύνοψη
Σ' αυτό το κεφάλαιο θα παρουσιάσουμε προχωρημένες λειτουργίες που γίνονται στο περιβάλλον του SQL Server. Πιο συγκεκριμένα, αφού μελετηθούν εντολές της SQL που αφορούν τον ορισμό δεδομένων (Data Definition Language), θα παρουσιαστούν κι άλλα προχωρημένα θέματα όπως είναι η βελτιστοποίηση ερωτημάτων, οι αποθηκευμένες διαδικασίες, τα εναύσματα και οι συναλλαγές με την βοήθεια της γλώσσας προγραμματισμού Transact-SQL.

4.1. Ερωτήματα ορισμού δεδομένων

Οι μέχρι τώρα εντολές της SQL αφορούσαν την αναζήτηση πληροφορίας σε μια βάση δεδομένων. Όμως, η SQL περιέχει εντολές και για την εισαγωγή, τη διαγραφή και τη μεταβολή της πληροφορίας που βρίσκεται στη βάση δεδομένων. Αυτές θα παρουσιαστούν στη συνέχεια.

4.1.1. Εισαγωγή πολλών γραμμών σε πίνακα

Στην SQL είναι δυνατή η μαζική εισαγωγή πολλών εγγραφών σε έναν πίνακα με την βοήθεια ενός ερωτήματος επιλογής από έναν άλλο πίνακα (Hoffer, Venkatarama, & Topi, 2013· Μανωλόπουλος, & Παπαδόπουλος, 2006). Για παράδειγμα, έστω τα παρακάτω δύο ερωτήματα:

"Να δημιουργηθεί ένας νέος πίνακας για ταινίες πριν από το 1960, με όνομα ΠΑΛΙΑ_ΤΑΙΝΙΑ". Η εντολή SQL είναι:

Q49

```
CREATE TABLE ΠΑΛΙΑ_ΤΑΙΝΙΑ(
ID int NOT NULL ,
Τίτλος varchar (100) NOT NULL,
Έτος int NULL,
PRIMARY KEY (ID))
```

```
-----------
Command(s) completed successfully.
```

"Στον πίνακα ΠΑΛΙΑ_ΤΑΙΝΙΑ να προστεθούν οι γραμμές του πίνακα ΤΑΙΝΙΑ, οι οποίες αντιστοιχούν σε ταινίες με έτος παραγωγής μικρότερο του 1960". Η εντολή SQL είναι:

Q50

```
Insert into ΠΑΛΙΑ_ΤΑΙΝΙΑ
Select ID, Τίτλος, Έτος
From ΤΑΙΝΙΑ
Where Έτος < 1960
```

```
-----------
(2 row(s) affected)
```

4.1.2. Ενημέρωση τιμής των πεδίων ενός πίνακα

Με την SQL είναι δυνατή η μεταβολή της τιμής των πεδίων ορισμένων ή όλων των γραμμών ενός πίνακα. Η διαδικασία αυτή γίνεται με τους όρους update και set της SQL. Για παράδειγμα, έστω το ερώτημα:

"Να μειωθεί κατά 1 ευρώ η τιμή του δίσκου με κωδικό 1.". Η εντολή SQL είναι:

Q51

```
Update ΔΙΣΚΟΣ
Set Τιμή = Τιμή - 1
Where ID = 1
```

```
-----------
(1 row(s) affected)
```

Ο όρος Set περιέχει τα πεδία που θα ενημερωθούν, τα οποία μπορεί να είναι και περισσότερα από ένα. Τονίζεται, επίσης, ότι αν θέλαμε να μη μειώσουμε απλώς την τιμή ενός δίσκου αλλά να τη διαγράψουμε πλήρως, τότε θα αρκούσε να θέσουμε στο όρισμα SET Τιμή = NULL. Μ' αυτόν τον τρόπο θα πετυχαίναμε την διαγραφή τιμής για ένα ορισμένο πεδίο και όχι τη διαγραφή ολόκληρων εγγραφών, η οποία θα επιδειχθεί στην επόμενη ενότητα. Τέλος, ο όρος where είναι προαιρετικός. Αυτό σημαίνει ότι αν δεν υπάρχει ο όρος WHERE, τότε ενημερώνονται ανάλογα οι τιμές όλων των εγγραφών για το συγκεκριμένο πεδίο. Για παράδειγμα, έστω το ερώτημα:

"Έγινε ανακαίνιση στο DVDclub και πλέον όλοι οι δίσκοι είναι τύπου BLU-RAY. Να ενημερωθεί ο πίνακας ΔΙΣΚΟΣ". Η εντολή SQL είναι:

Q52

```
Update ΔΙΣΚΟΣ
Set Τύπος = 'BLU-RAY'
```

```
-----------
(3 row(s) affected)
```

4.1.3. Διαγραφή των γραμμών ενός πίνακα

Στην SQL επιτρέπεται η διαγραφή ολόκληρων εγγραφών και *όχι* η μερική διαγραφή για ορισμένα μόνο πεδία. Η διαγραφή γίνεται με την εντολή **delete from.** Για παράδειγμα, έστω το αίτημα:

"Να διαγραφεί ο πελάτης με κωδικό 4". Η αντίστοιχη εντολή είναι:

Q53

```
Delete from ΠΕΛΑΤΗΣ
Where ID = 4
```

Ο όρος where είναι προαιρετικός. Στην περίπτωση που δεν δίνεται συνθήκη, τότε διαγράφονται όλες οι εγγραφές της σχέσης. Για παράδειγμα, έστω το αίτημα:

"Διαγράψτε όλα τα περιεχόμενα του πίνακα ΠΕΛΑΤΗΣ". Η εντολή SQL είναι:

Q54

```
Delete from ΠΕΛΑΤΗΣ
```

4.1.4. Μεταβολή της δομής ενός πίνακα

Για την προσθήκη νέων πεδίων σε έναν πίνακα, υπάρχει η εντολή SQL alter table σε συνδυασμό με τον όρο add. Έστω, για παράδειγμα, το αίτημα:

"Να γίνει προσθήκη ενός πεδίου με όνομα Γλώσσα, τύπου char(30), στον πίνακα ΤΑΙΝΙΑ".

Q55
```
Alter table TAINIA
add Γλώσσα char(30)
```

ΠΡΟΣΟΧΗ! Οι τιμές του πεδίου Γλώσσα είναι αρχικά null για όλες τις γραμμές. Επιπλέον, θα πρέπει να τονιστεί ότι σε κάποιες περιπτώσεις ο SQL Server ενδέχεται για λόγους ασφαλείας να μην εφαρμόσει την αλλαγή σχεδίασης ενός πίνακα. Στη συγκεκριμένη περίπτωση, θα πρέπει, μέσω του μενού, να ακολουθήσουμε τη διαδρομή Tools → Options → Designers και, στην καρτέλα που θα εμφανιστεί, να αποεπιλέξουμε την επιλογή "Prevent saving changes that require table re-creation".

Για την διαγραφή πεδίων σε έναν πίνακα, υπάρχει η εντολή SQL alter table σε συνδυασμό με τον όρο drop column. Έστω, για παράδειγμα, το παρακάτω αίτημα:

"Να διαγραφεί το πεδίο Γλώσσα από τον πίνακα ΤΑΙΝΙΑ":

Q56
```
Alter table TAINIA drop column Γλώσσα
```

Επιπλέον, είναι δυνατή και η αλλαγή τύπου δεδομένων για ένα πεδίο ενός πίνακα. Έστω, για παράδειγμα, το παρακάτω αίτημα:

"Στον πίνακα ΡΟΛΟΣ, ο τύπος του πεδίου Περιγραφή να γίνει varchar (100)":

Q57
```
Alter table ΡΟΛΟΣ
Alter column Περιγραφή varchar (100)
```

4.1.5. Μετονομασία πίνακα και πεδίου πίνακα

Η μετονομασία πίνακα στον SQL Server γίνεται με τη χρήση της ενσωματωμένης συνάρτησης **sp_rename** σε συνδυασμό με τον όρο **exec**. Για παράδειγμα, έστω το αίτημα:

"Να μετονομαστεί ο πίνακας ΔΙΣΚΟΣ σε DVD". Η εντολή SQL είναι:

Q58
```
Exec sp_rename 'ΔΙΣΚΟΣ', 'DVD'
```

Η μετονομασία πεδίου πίνακα στην Transact-SQL γίνεται, επίσης, με τη χρήση της ενσωματωμένης συνάρτησης **sp_rename.** Για παράδειγμα, έστω το αίτημα:

"Να μετονομαστεί το πεδίο Έτος του πίνακα ΤΑΙΝΙΑ σε Χρονιά". Η εντολή SQL είναι:

Q59
```
Exec sp_rename 'ΤΑΙΝΙΑ.Έτος ', 'Χρονιά', 'COLUMN'
```

4.1.6. Διαγραφή πίνακα και βάσης δεδομένων

Για την κατάργηση ενός πίνακα από τη βάση δεδομένων υπάρχει η εντολή drop table. Έστω, για παράδειγμα, το αίτημα:

"Να διαγραφεί ο πίνακας ΠΑΛΙΑ_ΤΑΙΝΙΑ". Η εντολή SQL είναι:

Q60	Drop table ΠΑΛΙΑ_ΤΑΙΝΙΑ

Πρέπει απαραίτητα να σημειωθεί ότι η παραπάνω εντολή: (α) διαγράφει *όλες* τις γραμμές και (β) καταργεί τον ίδιο τον πίνακα. Επομένως, για παράδειγμα, δεν είναι δυνατή η αναφορά για εισαγωγή γραμμών στον πίνακα που διαγράφηκε. Αντιθέτως, η εντολή delete from, που αναφέρθηκε νωρίτερα, θα διαγράψει όλες τις γραμμές αλλά όχι τον πίνακα. Τέλος, σημειώνουμε ότι η διαγραφή ολόκληρης της βάσης δεδομένων γίνεται επίσης με την εντολή **drop**. Για παράδειγμα, έστω το παρακάτω αίτημα:

Να διαγραφεί η βάση δεδομένων database_name". Η εντολή SQL είναι:

Q61	Drop database database_name

4.2. Αποθηκευμένες Διαδικασίες, Εναύσματα, Συναλλαγές

4.2.1. Αποθηκευμένες διαδικασίες/ Stored Procedures

Οι αποθηκευμένες διαδικασίες/ stored procedures είναι τμήματα κώδικα προγράμματος συνδυασμένα με SQL ερωτήματα που αποθηκεύονται στην βάση δεδομένων και ενεργοποιούνται κάθε φορά που εμείς θέλουμε να τα χρησιμοποιήσουμε. Συνήθως αφορούν εργασίες που γίνονται πολύ συχνά και δεν υπάρχει λόγος κάθε φορά να τις φτιάχνουμε από την αρχή. Ο SQL Server έχει ενσωματωμένη την γλώσσα προγραμματισμού Transact-SQL (T-SQL), με την οποία μπορούμε να δημιουργούμε αποθηκευμένες διαδικασίες. Για παράδειγμα, έστω το παρακάτω ερώτημα:

"Να δημιουργηθεί μια διαδικασία (Stored Procedure) που θα εισάγει *N* εγγραφές πελατών κάθε φορά που την καλούμε.". Ο κώδικας της αποθηκευμένης διαδικασίας είναι:

Q62

```
use DVDclub
go
Create procedure InsertRandomPelates @CustomersTotalNumber as int
 as
begin
   SET NOCOUNT ON;
   Declare @CustomersName as varchar (30)
   Declare @counter as int
   Set @counter=4

   while @counter <= @CustomersTotalNumber
      begin
      Set @CustomersName = cast(@counter as varchar) + '-Name'

        INSERT INTO ΠΕΛΑΤΗΣ (Id, Όνομα) VALUES (@counter, @CustomersName)

      Set @counter= @counter +1
 end
end
```

Ακολούθως αναλύουμε τις πιο σημαντικές εντολές από το παραπάνω ερώτημα:

- *Create procedure InsertRandomPelates @CustomersTotalNumber as int*: Η εντολή αυτή δημιουργεί μια stored procedure με όνομα InsertRandomPelates όπου δηλώνεται μια παράμετρος εισόδου με όνομα @CustomersTotalNumber τύπου integer, μέσω της οποίας θα ορίζουμε πόσους πελάτες θα εισάγουμε στον πίνακα ΠΕΛΑΤΗΣ.

- *as begin… end*: Μέσα στα όρια του begin..end είναι το κυρίως πρόγραμμα που εκτελείται.

- *Declare @CustomersName as varchar (30):* Δηλώνουμε μια μεταβλητή με μήκος 30 χαρακτήρες, η οποία θα κρατά το όνομα πελάτη που δημιουργείται κάθε φορά με σειριακό/αύξοντα τρόπο.

- *Declare @counter as int:* Δηλώνουμε μια μεταβλητή τύπου integer που θα λειτουργεί ως μετρητής, για να μετράμε πόσες εντολές εισαγωγής γραμμών εκτελέσαμε στον πίνακα ΠΕΛΑΤΗΣ.

- *Set @counter=0:* Μηδενίζουμε την μεταβλητή counter.

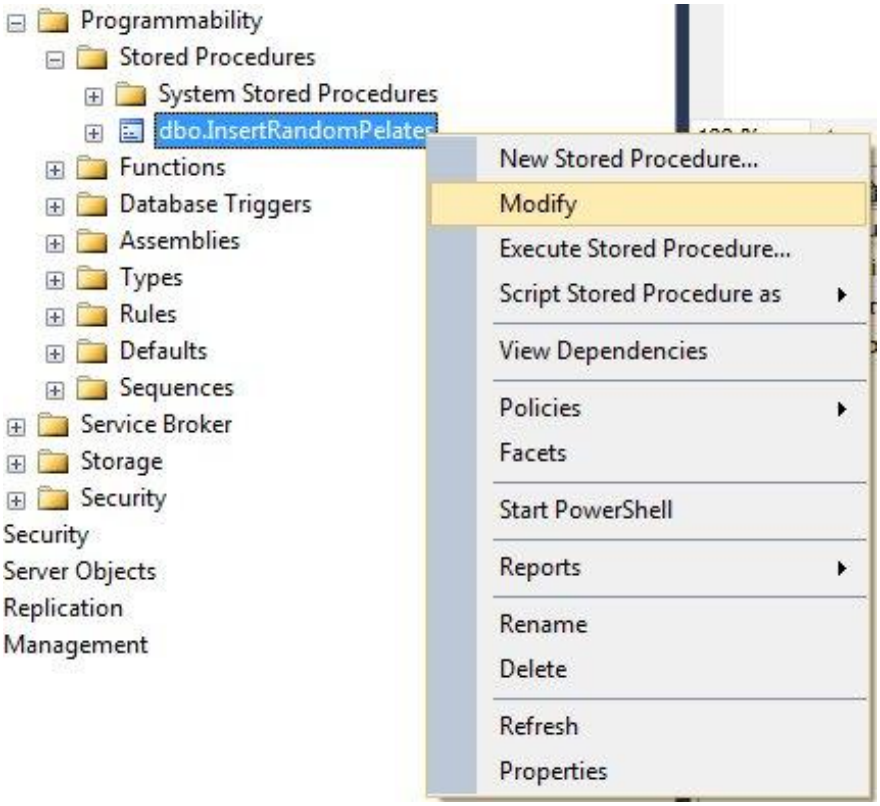
- *while @counter<= @CustomersTotalNumber:* Ξεκινά η επαναληπτική διαδικασία εισαγωγής γραμμών. Διαρκεί όσο ο μετρητής μας είναι μικρότερος ή ίσος από τον συνολικό αριθμό εισαγωγής γραμμών που έχουμε δηλώσει με την μεταβλητή @CustomersTotalNumber.

- *Set @CustomersName=cast(@counter as varchar) + ' -Name':* Δημιουργούμε το όνομα του πελάτη που αποτελείται από 2 κομμάτια: τον τρέχοντα αύξοντα αριθμό και το σταθερό όνομα -Name. Συνεπώς, το πρώτο όνομα πελάτη που θα εισάγουμε είναι το 4-Name, 5-Name κτλ. Τονίζεται ότι η εντολή cast μετατρέπει τη μεταβλητή @counter τύπου int σε μεταβλητή τύπου varchar (χαρακτήρας).

- *INSERT INTO ΠΕΛΑΤΗΣ (ID, Όνομα) VALUES (@counter, @CustomersName):* Είναι η εντολή που εισάγει μια νέα εγγραφή στον πίνακα ΠΕΛΑΤΗΣ.

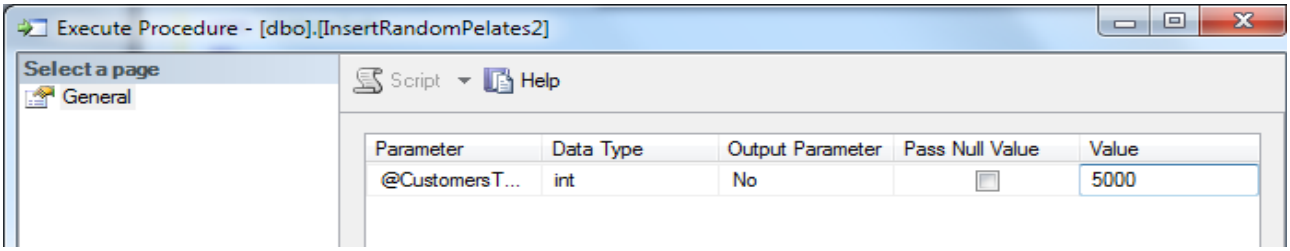
- *Set @m= @m +1:* Είναι η εντολή που αυξάνει τον μετρητή counter κατά 1.

Μετά την επιτυχή εκτέλεση της αποθηκευμένης διαδικασίας, έχουμε τη δυνατότητα να την εντοπίσουμε στο φάκελο Programmability – Stored Procedures που βρίσκεται μέσα στο φάκελο της βάσης δεδομένων dvdclub, όπως φαίνεται στην Εικόνα 4.1. Αν θέλουμε να αλλάξουμε τον κώδικα της διαδικασίας, επιλέγουμε δεξί κλικ πάνω της και, στη συνέχεια, την εντολή Modify.



Εικόνα 4.1

Για την εκτέλεση του κώδικα της διαδικασίας επιλέγουμε δεξί κλικ και Execute Stored Procedure. Ακολούθως, εμφανίζεται ένα παράθυρο εκτέλεσης, όπως αυτό της Εικόνας 4.2, όπου μπορούμε να εισάγουμε τις παραμέτρους εισόδου. Στο παράδειγμά μας θα εισάγουμε τον αριθμό 5000, γιατί θέλουμε να δημιουργηθούν 5000 νέοι πελάτες. Επιλέγοντας OK παράγεται αυτόματα ο αντίστοιχος κώδικας που εκτελεί την διαδικασία και, έτσι, ξεκινά η εκτέλεσή της.



Εικόνα 4.2

Ανάλογα με τον όγκο της εργασίας της, η αποθηκευμένη διαδικασία εκτελείται στο χρονικό διάστημα δευτερολέπτων ή λεπτών της ώρας. Πρέπει να περιμένουμε μέχρι να δούμε το αντίστοιχο μήνυμα επιτυχούς εκτέλεσης ή όχι. Προκειμένου να επιβεβαιώσουμε ότι τα δεδομένα μας έχουν εισαχθεί στον πίνακα ΠΕΛΑΤΗΣ, αρκεί να τον ανοίξουμε με την επιλογή «Select top 1000 rows». Ένα δεύτερο παράδειγμα μιας stored procedure δίνεται με το παρακάτω ερώτημα:

“Να δημιουργηθεί μια διαδικασία (Stored Procedure) που θα εμφανίζει τα τηλέφωνα όλων των πελατών κάθε φορά που την καλούμε.”. Ο κώδικας της αποθηκευμένης διαδικασίας είναι:

Q63

```
use DVDclub
go
Create Procedure myphone1
as
Select Τηλέφωνο
From ΠΕΛΑΤΗΣ
```

Προκειμένου να εκτελέσουμε την παραπάνω αποθηκευμένη διαδικασία, μπορούμε σε ένα Query Editor να δώσουμε την εντολή EXEC myphone1. Τέλος, η παραπάνω αποθηκευμένη διαδικασία μπορεί να τροποποιηθεί κατάλληλα, προκειμένου να εμφανίζει μόνο το τηλέφωνο ενός συγκεκριμένου πελάτη. Για παράδειγμα, έστω το παρακάτω ερώτημα:

“Να δημιουργηθεί μια διαδικασία (Stored Procedure) που θα εμφανίζει το τηλέφωνο ενός υπό εξέταση πελάτη”. Ο κώδικας της αποθηκευμένης διαδικασίας είναι:

Q64

```
use DVDclub
go
Create Procedure myphone2 @lastname varchar (40)
as
Select Τηλέφωνο
From ΠΕΛΑΤΗΣ
Where Όνομα = @lastname
```

Προκειμένου να εκτελέσουμε την παραπάνω αποθηκευμένη διαδικασία, μπορούμε σε ένα Query Editor να δώσουμε την εντολή EXEC myphone2 ‘Καντακουζηνός’. Τέλος, ως άσκηση, θα μπορούσαμε να προχωρήσουμε στη δημιουργία μιας stored procedure που θα εμφανίζει όλους τους ηθοποιούς.

4.2.2. Εναύσματα/Triggers

Το εύναυσμα/trigger είναι μια διαδικασία την οποία ορίζει ο διαχειριστής της βάσης δεδομένων. Ενεργοποιείται αυτόματα από τον SQL Server κάθε φορά που συμβαίνουν μεταβολές εισαγωγής, διαγραφής ή ενημέρωσης στα δεδομένα ενός πίνακα. Η βασική (χωρίς τα προαιρετικά ορίσματα) σύνταξη ενός trigger δίνεται παρακάτω:

```
CREATE TRIGGER trigger_name
ON {table | view}
{FOR | AFTER } { [DELETE] [,] [INSERT] [,] [UPDATE] }
        AS
            sql_statement
```

όπου

- **trigger_name** είναι το όνομα του trigger.
- **{ table | view }** είναι ο πίνακας ή η όψη για τον/την οποίο/ οποία θα ενεργοποιηθεί το trigger.
- **{ FOR | AFTER }** προσδιορίζουν πότε θα ενεργοποιηθεί ένα trigger. Πιο συγκεκριμένα, το FOR ενεργοποιεί το trigger όταν γίνει κάποιο deletion, insertion ή update στον πίνακα ή την όψη. Το AFTER ενεργοποιεί το trigger αφού έχει ολοκληρωθεί επιτυχώς το sql statement του trigger.
- **{ [DELETE] [,] [INSERT] [,] [UPDATE] }** προσδιορίζουν την πράξη που θα γίνει σε έναν πίνακα ή σε μια όψη και θα ενεργοποιήσει το trigger.
- **sql_statement** είναι οι ενέργειες που θα εκτελεστούν από το trigger.

Υποθέτουμε, λοιπόν, ότι θέλουμε να φτιάξουμε ένα trigger το οποίο θα ενημερώνει ένα πεδίο του πίνακα ΔΙΣΚΟΣ κάθε φορά που ένας δίσκος DVD ενοικιάζεται στο πίνακα ΕΝΟΙΚΙΑΣΗ. Για παράδειγμα, έστω τα παρακάτω ερωτήματα:

"Δημιουργήστε ένα νέο πεδίο στον πίνακα ΔΙΣΚΟΣ με όνομα 'loaned' και τύπο δεδομένων char(1). Το πεδίο θα έχει default αρχική τιμή ('n') , που σημαίνει ότι ο δίσκος δεν είναι ενοικιασμένος."

Q65

```
use DVDclub
go
Alter Table ΔΙΣΚΟΣ
add loaned char(1) default 'n'
```

Προσοχή! Σημειώνεται ότι οι ήδη υπάρχουσες εγγραφές του πίνακα ΔΙΣΚΟΣ θα έχουν τιμή NULL στο πεδίο loaned. Μόνο οι νέες εγγραφές θα παίρνουν εξαρχής αυτόματα την τιμή `'n'`. Συνεπώς, θα πρέπει να ενημερώσουμε τις παλιές εγγραφές με το παρακάτω ερώτημα:

"Ενημερώστε τον πίνακα ΔΙΣΚΟΣ, ώστε το πεδίο loaned στις παλιές εγγραφές να έχει την τιμή ('n') .

Q66

```
use DVDclub
go
UPDATE ΔΙΣΚΟΣ
SET loaned='n'
```

“Το πεδίο loaned θα ενημερώνεται με την τιμή true (‘y’) κάθε φορά που θα ενοικιάζεται ένας ψηφιακός δίσκος στον πίνακα ΕΝΟΙΚΙΑΣΗ”. Ο κώδικας της αποθηκευμένης διαδικασίας είναι:

Q67

```
use DVDclub
go
Create Trigger loaned_updater ON ΕΝΟΙΚΙΑΣΗ For Insert
as
        Update ΔΙΣΚΟΣ Set loaned = 'y'
         From ΔΙΣΚΟΣ, INSERTED i
        Where ΔΙΣΚΟΣ.ID = i.IDΔίσκου
```

Εφόσον εκτελέσουμε επιτυχώς τον παραπάνω κώδικα στον Query Editor, μπορούμε να επιβεβαιώσουμε την δημιουργία του trigger στον πίνακα ΕΝΟΙΚΙΑΣΗ μέσα στον φάκελο triggers. Το trigger αυτό θα ενεργοποιείται κάθε φορά που θα γίνεται η εισαγωγή μιας νέας εγγραφής στον πίνακα ΕΝΟΙΚΙΑΣΗ. Πιο συγκεκριμένα, μετά την εισαγωγή μιας εγγραφής στο πίνακα ΕΝΟΙΚΙΑΣΗ, η συγκεκριμένη εγγραφή εισάγεται μέσα στο trigger με την βοήθεια ενός virtual πίνακα, ο οποίος έχει το δεσμευμένο (by default) όνομα INSERTED. Ο πίνακας INSERTED περιλαμβάνει την εγγραφή ή τις εγγραφές που κάθε φορά εισάγονται στον πίνακα ΕΝΟΙΚΙΑΣΗ και ενεργοποιούν το trigger μας. Εμείς, για συντομία, μετονομάζουμε το όνομα του πίνακα INSERTED σε i και εφαρμόζουμε την πράξη τη φυσικής σύνδεσης, προκειμένου να γίνει update στην τιμή εκείνης της εγγραφής του πίνακα ΔΙΣΚΟΣ που έχει τιμή ίση με το i.IDΔίσκου.

Στη συνέχεια θα περιγράψουμε μια ακόμη περίπτωση, στην οποία είναι αναγκαία η χρήση εναυσμάτων. Ας υποθέσουμε ότι στη βάση δεδομένων μας υπάρχει ένα θέμα ασφαλείας, καθώς ένας υπάλληλος σκόπιμα διαγράφει εγγραφές από τον πίνακα ΕΝΟΙΚΙΑΣΗ, ώστε να μπορεί να ενοικιάζει σε φίλους του ταινίες που εσκεμμένα δεν καταγράφονται και δεν χρεώνονται ποτέ. Για να αποφύγουμε την παραπάνω περίπτωση, φτιάχνουμε τον πίνακα ΕΝΟΙΚΙΑΣΗ_DELETED, ο οποίος θα αποθηκεύει τις εγγραφές που διαγράφονται από τον πίνακα ΕΝΟΙΚΙΑΣΗ. Επιπρόσθετα, θα καταγράφει με ακρίβεια την ημερομηνία και την ώρα που πραγματοποιήθηκε η διαγραφή μιας εγγραφής από τον πίνακα ΕΝΟΙΚΙΑΣΗ, ώστε να μπορούμε να ερευνήσουμε από ποιον υπάλληλο έγινε η διαγραφή. Η δομή του πίνακα ΕΝΟΙΚΙΑΣΗ_DELETED φαίνεται στην Πίνακα 4.1.

Field	Type	Null	Key
ID	Int		PRI
IDΠελάτη	int		
IDΔίσκου	int		
Από	date		
Έως	date	YES	
DateTimeOfDeleted	datetime		

Πίνακας 4.1 ΕΝΟΙΚΙΑΣΗ_DELETED

Ο κώδικας για την δημιουργία του trigger είναι ο παρακάτω:

Q68

```
Create trigger ΕΝΟΙΚΙΑΣΗ_After_Delete
ON ΕΝΟΙΚΙΑΣΗ After Delete
as
    Begin
        Set nocount on

        Insert into ΕΝΟΙΚΙΑΣΗ_DELETED
        ([IDΠελάτη],[IDΔίσκου] ,[Από] ,[Έως] ,[DateTimeOfDelete])

        Select IDΠελάτη,IDΔίσκου, Από, Έως, getdate() from Deleted
    End
```

Ο παραπάνω κώδικας φτιάχνει ένα trigger με το όνομα «ΕΝΟΙΚΙΑΣΗ_After_Delete» που θα εφαρμοστεί στον πίνακα ΕΝΟΙΚΙΑΣΗ. Συγκεκριμένα, θα εκτελείται μετά από κάθε γεγονός διαγραφής *(AFTER DELETE)* μιας ή περισσοτέρων εγγραφών από τον πίνακα ΕΝΟΙΚΙΑΣΗ. Μέσα στο trigger προϋπάρχει εξάλλου, ένας virtual πίνακας με όνομα **Deleted**, που περιέχει τα πεδία και τις εγγραφές του πίνακα ΕΝΟΙΚΙΑΣΗ που διαγράφηκαν την τελευταία φορά.

Με το ερώτημα εισαγωγής «*Insert into ΕΝΟΙΚΙΑΣΗ_DELETED (ΙΔΠελάτη,ΙΔΔίσκου,Από ,Έως,DateTimeOfDelete)»* εισάγουμε τα διαγραφέντα στοιχεία στον νέο πίνακα ΕΝΟΙΚΙΑΣΗ_DELETED.

Με το ερώτημα επιλογής «*Select ΙΔΠελάτη,ΙΔΔίσκου, Από, Έως, getdate() from Deleted*» συλλέγουμε όλα τα διαγραφέντα πεδία με την τρέχουσα ημερομηνία και ώρα διαγραφής.

Προκειμένου να επιβεβαιώσουμε την ορθή λειτουργία του εναύσματος, ανοίγουμε τον πίνακα ΕΝΟΙΚΙΑΣΗ με δεξί κλικ και επιλέγουμε Edit, για να εμφανιστούν οι τρεις εμπεριεχόμενες εγγραφές. Επιλέγοντας μια εγγραφή και πατώντας το πλήκτρο Delete, θα την διαγράψουμε. Αν τώρα πάμε στον πίνακα ΕΝΟΙΚΙΑΣΗ_DELETED και επιλέξουμε να δούμε τις εγγραφές του, θα βρούμε την εγγραφή που μόλις εισήχθηκε.

4.2.3. Συναλλαγές/ Transactions

Συναλλαγή ή δοσοληψία (transaction) καλείται ένα σύνολο λειτουργιών/ ενεργειών ενημέρωσης, διαγραφής ή εισαγωγής γραμμών, το οποίο αποτελεί μια ενιαία λογική λειτουργική μονάδα. Συγκεκριμένα, εντάσσουμε σε μια συναλλαγή όλες εκείνες τις ενέργειες διαγραφής, ενημέρωσης ή εισαγωγής γραμμών που πρέπει να εκτελεστούν μαζί επιτυχώς. Αν τουλάχιστον μία ενέργεια αποτύχει, τότε καμία από τις άλλες επιτυχημένες ενέργειες δεν θα γίνει αποδεκτή από τον SQL Server (δεν θα γίνει Commit). Αντίθετα, ο SQL Server θα επιστρέψει στην αρχική του κατάσταση, σαν να μην είχε συμβεί καμία ενέργεια (Rollback). Ας μελετήσουμε, για παράδειγμα, τα παρακάτω δύο ερωτήματα, εκ των οποίων το πρώτο σκοπίμως είναι εσφαλμένο και δεν θα εκτελεστεί:

"Εισάγετε στον πίνακα ΕΝΟΙΚΙΑΣΗ την ενοικίαση ενός δίσκου με ΙDΔίσκου 2 και στην συνέχεια ενημερώστε τον πίνακα ΔΙΣΚΟΣ, ώστε το πεδίο loaned για τον δίσκο με κωδικό Id 2 να είναι 'y'".

Q69

```
use DVDclub
go

Insert into ΕΝΟΙΚΙΑΣΗ (ΙDΠελάτη,IDΔίσκου,Από,Έως)
Values (2,2,'07/24/20133',Null)
Go

Update ΔΙΣΚΟΣ SET loaned='y' where Id= 2
Go
```

```
-----------------------------------
Msg 241, Level 16, State 1, Line 1
Conversion failed when converting date and/or time from character string.

(1 row(s) affected)
```

Προσοχή! Όπως γίνεται αντιληπτό από τα παραπάνω αποτελέσματα, το ερώτημα εισαγωγής νέας εγγραφής δεν εκτελέστηκε λόγω λάθους ημερομηνίας ('07/24/20133'), ενώ το ερώτημα ενημέρωσης εκτελέστηκε επιτυχώς. Αποτέλεσμα της πρώτης μη επιτυχημένης εκτέλεσης είναι να έχουμε ασυνέπεια δεδομένων στην βάση μας. Δηλαδή, ενώ ένας δίσκος δεν φαίνεται να έχει ενοικιαστεί στο πίνακα ΕΝΟΙΚΙΑΣΗ, φαίνεται νοικιασμένος στον πίνακα ΔΙΣΚΟΣ.

Το πρόβλημα που προέκυψε θα λυνόταν εύκολα με τη χρήση ενός transaction που θα εξασφάλιζε ότι οι δύο εντολές, εφόσον εκτελεστούν ταυτόχρονα επιτυχώς, θα γίνουν COMMIT από τον SQL Server. Το παραπάνω ερώτημα γίνεται με τη βοήθεια της χρήσης της παρακάτω συναλλαγής:

Q70

```
use DVDclub
go

Declare @myerror as int
Select @myerror = 0

Begin Transaction

Insert into ΕΝΟΙΚΙΑΣΗ (ΙΔΠελάτη,ΙΔΔίσκου,Από,Έως) Values (2,2,'07/24/20133',Null)

Update ΔΙΣΚΟΣ SET loaned='y' where Id= 2

Select @myerror = @@error
IF @myerror != 0 GOTO handle_error

Commit Transaction

handle_error:
IF @myerror != 0
Begin
        print 'Error in Script. Rollback is applied'
        Rollback Transaction
End
```

Ακολούθως αναλύουμε τις πιο σημαντικές εντολές από το παραπάνω ερώτημα:

Declare @myerror as int: Ορίζουμε μια μεταβλητή, η οποία θα πάρει τιμή διάφορη του μηδενός στην περίπτωση που υπάρχει σφάλμα σε μία από της δύο εντολές SQL που περιλαμβάνει η συναλλαγή μας.

Select @myerror = @@error: Η μεταβλητή μας παίρνει τιμή διάφορη του μηδενός στην περίπτωση που υπάρξει σφάλμα εκτέλεσης σε ένα από τα δύο ερωτήματα της συναλλαγής.

Commit Transaction: Στην περίπτωση που δεν υπάρξουν σφάλματα, αποδεχόμαστε τα αποτελέσματα των δύο ερωτημάτων μας και τα αποθηκεύουμε μόνιμα στην βάση δεδομένων μας.

Rollback Transaction: Στην περίπτωση που υπάρξουν σφάλματα, δεν αποδεχόμαστε τα αποτελέσματα των δύο ερωτημάτων μας και επιστρέφουμε στην προηγούμενη κατάσταση της βάσης δεδομένων μας.

4.3. Βελτιστοποίηση Ερωτημάτων

Πριν από την εκτέλεση ενός ερωτήματος από τον SQL Server υπάρχει το στάδιο της βελτιστοποίησης (optimization). Το αρχικό ερώτημα που υποβάλλει ο χρήστης μπορεί να βελτιστοποιηθεί από τον SQL Server με ισοδύναμες πράξεις της SQL, έτσι ώστε να μειωθεί ο χρόνος επεξεργασίας του. Στο στάδιο της βελτιστοποίησης προσδιορίζεται, επίσης, το πλάνο εκτέλεσης ερωτήματος (query execution plan), το οποίο περιέχει την σειρά εκτέλεσης των πράξεων SQL, τις μεθόδους που θα χρησιμοποιηθούν για την εκτέλεση κάθε πράξης κτλ.

4.3.1. Παρακολούθηση του πλάνου εκτέλεσης ερωτήματος SQL

Η βελτιστοποίηση ερωτήματος του SQL Server προσπαθεί να βρει το βέλτιστο πλάνο εκτέλεσης για κάθε ερώτημα SQL. Συγκεκριμένα, ο SQL Server αναλύει κάθε ερώτημα, υπολογίζοντας τον αριθμό των διαφορετικών πλάνων εκτέλεσης και το κόστος του κάθε πλάνου, σε σχέση με τους πόρους που είναι απαραίτητοι και τον χρόνο επεξεργασίας που απαιτείται. Τελικά, επιλέγεται το λιγότερο ακριβό πλάνο εκτέλεσης. Αν θέλουμε να δούμε το πλάνο εκτέλεσης που ο SQL Server επέλεξε για το SQL ερώτημά μας , πρέπει να πληκτρολογήσουμε το ερώτημα και να πατήσουμε Ctrl+L ή, διαφορετικά, να επιλέξουμε με δεξί κλικ Display Estimated Execution Plan, όπως φαίνεται στην Εικόνα 4.3.

Εικόνα 4.3

Για παράδειγμα, έστω ότι θέλουμε να απεικονισθεί το πλάνο εκτέλεσης για το παρακάτω ερώτημα:

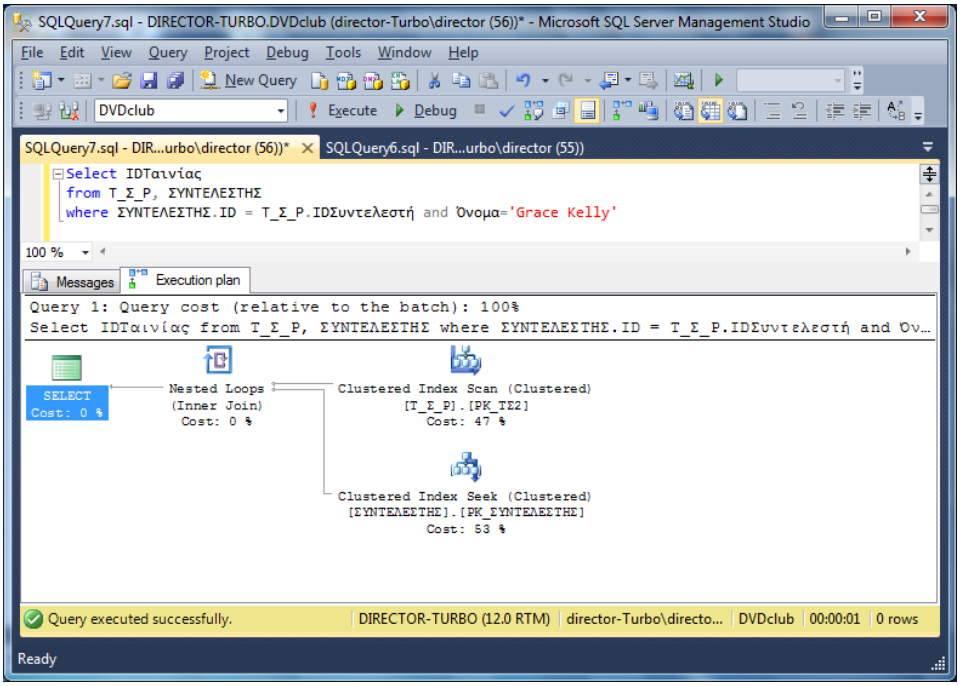
“Προβάλετε τους κωδικούς των ταινιών στις οποίες συμμετείχε η Grace Kelly ”. Η εντολή SQL είναι:

Q71

```
Select IDΤαινίας
from Τ_Σ_Ρ, ΣΥΝΤΕΛΕΣΤΗΣ
where ΣΥΝΤΕΛΕΣΤΗΣ.ID = Τ_Σ_Ρ.IDΣυντελεστή and Όνομα='Grace Kelly'
```

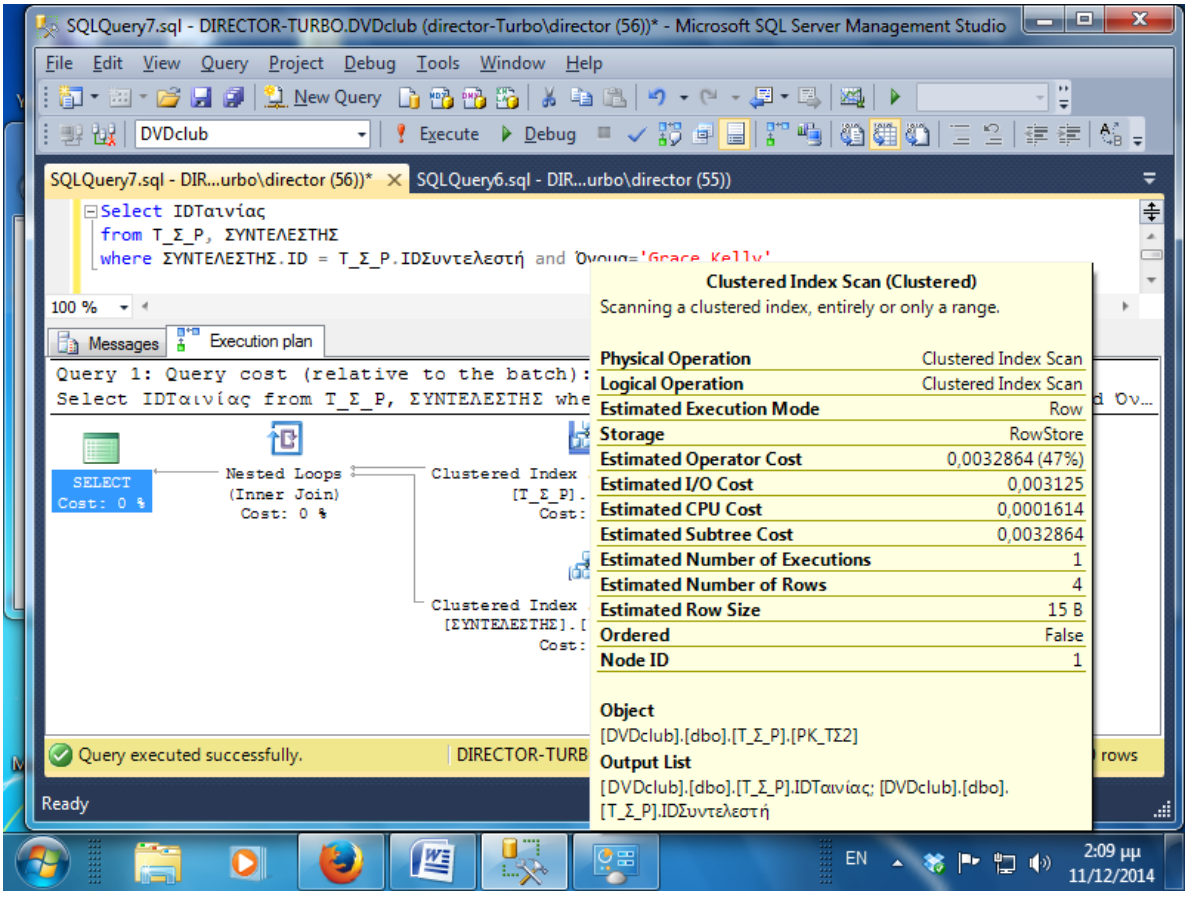
```
IDΤανίας
-----------
1
```

Έχοντας πληκτρολογήσει το παραπάνω ερώτημα και πατώντας Ctrl+L, εμφανίζεται το περιεχόμενο της Εικόνας 4.4.



Εικόνα 4.4

Το εκτιμώμενο πλάνο εκτέλεσης παρέχει πρόσβαση σε επιπρόσθετες πληροφορίες. Αν θέλουμε να τις δούμε, πρέπει να κρατήσουμε το ποντίκι επάνω στο εικονίδιο της λειτουργίας, οπότε εμφανίζεται ένα αναδυόμενο παράθυρο που τις περιέχει, όπως φαίνεται στην Εικόνα 4.5.



Εικόνα 4.5

Οι πιο σημαντικές από τις πληροφορίες που εμπεριέχονται στο αναδυόμενο παράθυρο της Εικόνας 4.5 είναι οι παρακάτω:

- **Estimated Number of Rows:** Ο εκτιμώμενος αριθμός των γραμμών που θα ανακτηθούν.
- **Estimated Row Size:** Το εκτιμώμενο μέγεθος των ανακτώμενων γραμμών σε bytes.
- **Estimated I/O Cost:** Ο εκτιμώμενος χρόνος I/O του ερωτήματος.
- **Estimated CPU Cost:** Ο εκτιμώμενος χρόνος CPU του ερωτήματος.

4.3.2. Ευρετήρια/ Indices

Το ευρετήριο είναι μια βοηθητική δομή που μας επιτρέπει να βελτιώσουμε την απόδοση των ερωτημάτων μας, μειώνοντας το μέγεθος της απαιτούμενης I/O δραστηριότητας για την ανάκτηση των ζητούμενων δεδομένων. Πράγματι, είναι χρήσιμο να ορίζουμε ευρετήρια για τα πεδία ενός πίνακα που χρησιμοποιούνται συχνά μέσα στο πεδίο Where των ερωτημάτων. Μ' αυτόν τον τρόπο επιταχύνουμε την διαδιακασία επεξεργασίας των ερωτημάτων. Για παράδειγμα, έστω το παρακάτω ερώτημα:

"Να δημιουργηθεί ευρετήριο για το όνομα του συντελεστή". Η εντολή SQL είναι:

Q72

Create index IND_ΣΥΝΤΕΛΕΣΤΗΣ **on** ΣΥΝΤΕΛΕΣΤΗΣ**(** Όνομα **)**

Η ύπαρξη ενός ευρετηρίου, π.χ. στο πεδίο Όνομα του πίνακα ΣΥΝΤΕΛΕΣΤΗΣ, βοηθά το πλάνο εκτέλεσης, ώστε να γίνει ταχύτερα πρώτα η επιλογή των γραμμών βάσει της συνθήκης για το όνομα και, στη συνέχεια, η σύνδεση των πινάκων. Αν, για παράδειγμα, θέλουμε να βελτιώσουμε το πλάνο εκτέλεσης του ερωτήματος Q71 της Ενότητας 4.3.1., μπορούμε να εφαρμόσουμε το παρακάτω ερώτημα που κάνει χρήση ευρετηρίου για το όνομα του συντελεστή:

"Να προβάλετε τους κωδικούς των ταινιών στις οποίες συμμετείχε η Grace Kelly ". Η εντολή SQL είναι:

Q73

```
select IDΤαινίας
from Τ_Σ_Ρ, ΣΥΝΤΕΛΕΣΤΗΣ with(INDEX(IND_ΣΥΝΤΕΛΕΣΤΗΣ))
where ΣΥΝΤΕΛΕΣΤΗΣ.ID = Τ_Σ_Ρ.IDΣυντελεστή and Όνομα='Grace Kelly'
```

Τονίζεται ότι στο παραπάνω ερώτημα η χρήση του όρου WITH είναι προαιρετική. Η εφαρμογή του αξιοποιεί τη χρήση του ευρετηρίου (Q72) και, συνεπώς, επιτυγχάνει την επιτάχυνση ανάκτησης της πληροφορίας. Στο νέο πλάνο εκτέλεσης φαίνεται ότι η επιλογή των εγγραφών βάσει του ονόματος συντελεστή γίνεται πλέον μέσω του ευρετηρίου που δημιουργήσαμε. Επισημαίνουμε ότι σ' αυτό το απλοποιημένο παράδειγμα, με τις λίγες εγγραφές που ανακτώνται από την βάση δεδομένων, δεν υπάρχει εμφανής διαφορά στους χρόνους εκτέλεσης μεταξύ των δύο ερωτημάτων Q71 και Q73. Γι' αυτόν τον λόγο, στην επόμενη ενότητα θα υλοποιήσουμε ένα παράδειγμα όπου θα γίνεται αισθητή η χρονική μείωση στη εκτέλεση ενός ερωτήματος που επιτυγχάνεται με τη χρήση ευρετηρίων.

Το ευρετήριο που κατασκευάσαμε στα προηγούμενα ερωτήματα στηρίχθηκε σε ένα μόνο πεδίο ενός πίνακα. Βέβαια, η δημιουργία σύνθετων ευρετηρίων που στηρίζονται σε περισσότερα του ενός πεδία ενός πίνακα κρίνεται αναγκαία σε πολλές περιπτώσεις, όπως, για παράδειγμα, όταν αυτά εμφανίζονται συχνά μέσα στο Order by ερωτημάτων. Για παράδειγμα, έστω το παρακάτω ερώτημα:

"Να δημιουργηθεί ευρετήριο για τον τύπο και την τιμή ενός δίσκου". Η εντολή SQL είναι:

Q74

Create index IND_ΤΥΠΟΣ_ΤΙΜΗ **on** ΔΙΣΚΟΣ**(Τύπος, Τιμή)**

Τέλος, στην SQL επιτρέπεται η διαγραφή ενός ευρετηρίου, όπως φαίνεται στο παρακάτω ερώτημα:

"Να διαγραφεί το ευρετήριο IND_ΣΥΝΤΕΛΕΣΤΗΣ". Η εντολή SQL είναι:

Q75

Drop index ΣΥΝΤΕΛΕΣΤΗΣ.IND_ΣΥΝΤΕΛΕΣΤΗΣ

4.3.3. Παράδειγμα βελτίωσης απόδοσης ερωτήματος με τη χρήση ευρετηρίου

Τα Ευρετήρια/ Indexes επιτρέπουν τη σημαντική βελτίωση του χρόνου εκτέλεσης ερωτημάτων. Αυτό, βέβαια, είναι κάτι που γίνεται αντιληπτό όταν έχουμε μεγάλο όγκο δεδομένων και τα ερωτήματα απαιτούν αρκετό χρόνο για να εκτελεστούν. Για παράδειγμα, στην Ενότητα 4.2.1. εισαγάγαμε, με την βοήθεια του ερωτήματος Q62, 5000 εγγραφές στον πίνακα ΠΕΛΑΤΗΣ. Έστω, λοιπόν, ότι θέλουμε να εκτελέσουμε το παρακάτω ερώτημα:

"Να βρεθούν τα ονόματα των πελατών που έχουν ως πρώτο χαρακτήρα το '1' και να ταξινομηθούν σε φθίνουσα σειρά". Η εντολή SQL είναι:

Q76

```
Select Όνομα
From ΠΕΛΑΤΗΣ
Where Όνομα like '1%'
Order by Όνομα desc
```

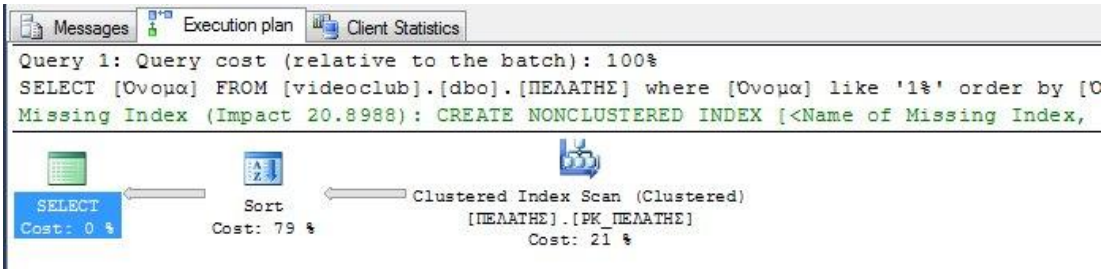
Αν κάνουμε δεξί κλικ στον χώρο του ερωτήματος και επιλέξουμε Estimated Execution Plan, ο SQL Server θα μας δείξει ένα διάγραμμα στο οποίο φαίνονται τα στάδια εκτέλεσης του ερωτήματος και το κόστος σε πόρους (CPU & I/O disks) που το καθένα απαιτεί. Πράγματι, στην Εικόνα 4.4 φαίνεται πως το συνολικό κόστος του ερωτήματός μας αφορά τρία στάδια:

- Η αναζήτηση και επιλογή των γραμμών από τον πίνακα ΠΕΛΑΤΗΣ βάσει της συνθήκης WHERE και με τη χρήση του κύριου κλειδιού του πίνακα έχει κόστος το 21% του συνολικού χρόνου. Σημειώνουμε ότι το ευρετήριο τύπου clustered με όνομα PK_ΠΕΛΑΤΗΣ δημιουργήθηκε εξ ορισμού από τον SQL Server στο κύριο κλειδί (στο παράδειγμα μας, στο ΠΕΛΑΤΗΣ.ID).

- Η φθίνουσα ταξινόμηση των ονομάτων κοστίζει το 79% του συνολικού χρόνου.

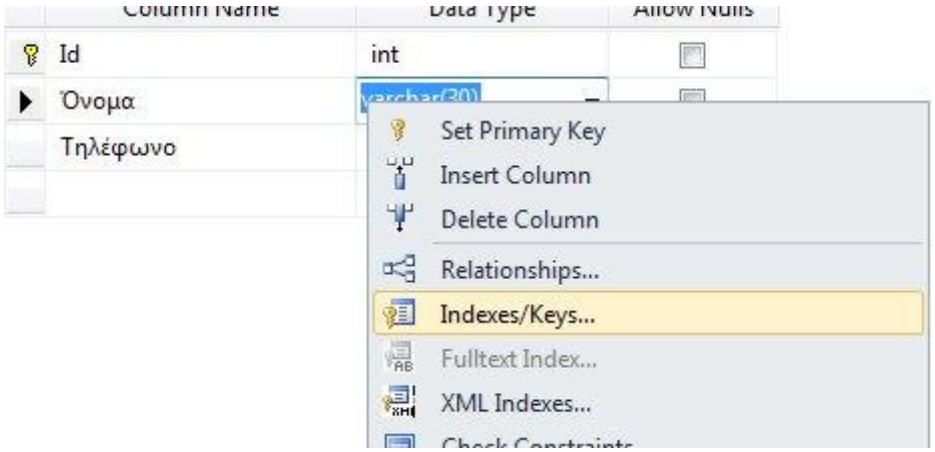
- Η πράξη της προβολής (Select) και το ποια πεδία του πίνακα θα εμφανιστούν δεν μας κοστίζει τίποτα.

Στην Εικόνα 4.6 παρατηρούμε την οδηγία που δίνεται γραμμένη με πράσινα γράμματα. Το σύστημα έχει εντοπίσει την απουσία ενός index για το πεδίο Όνομα του πίνακα ΠΕΛΑΤΗΣ, που αποτελεί και κριτήριο αναζήτησης αλλά και πεδίο ταξινόμησης. Μάλιστα, γίνεται εκτίμηση ότι αν υπήρχε ένας τέτοιος index, το impact στην βελτίωση της απόδοσης του ερωτήματός μας θα ήταν 20.8988 επί του συνολικού χρόνου.



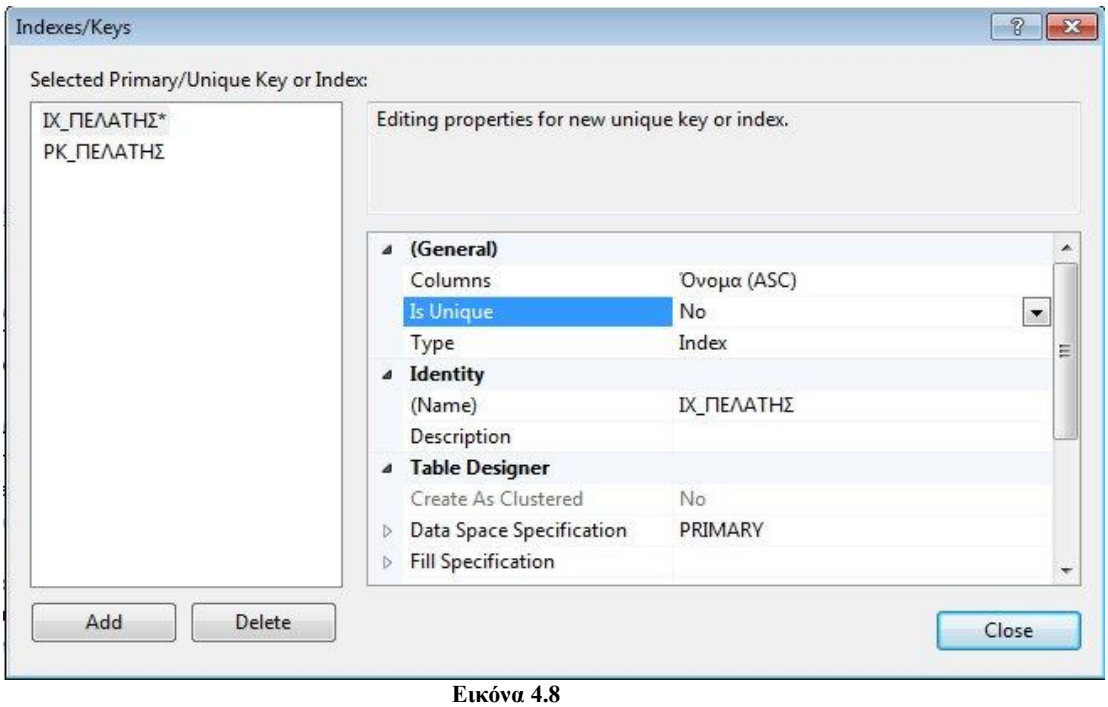
Εικόνα 4.6

Προκειμένου, λοιπόν, να βελτιστοποιήσουμε το αρχικό μας ερώτημα, επιλέγουμε τον πίνακα ΠΕΛΑΤΗΣ και κάνουμε δεξί κλικ στο Design, σκοπεύοντας να φτιάξουμε έναν Index στο πεδίο Όνομα. Κάνουμε δεξί κλικ οπουδήποτε και επιλέγουμε Indexes/Keys, προκειμένου να εμφανιστεί το παράθυρο της Εικόνας 4.7.



Εικόνα 4.7

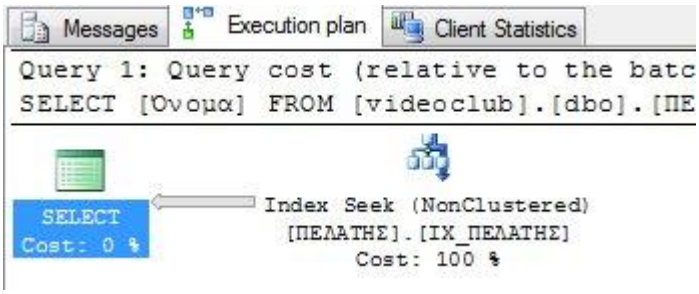
Βλέπουμε ότι υπάρχει ήδη ένα ευρετήριο με όνομα PK_ΠΕΛΑΤΗΣ (του κύριου κλειδιού) που δημιουργείται αυτόματα όταν δηλώνουμε μια στήλη ως κύριο κλειδί. Αφού επιλέξουμε Add, στο πεδίο Columns επιλέγουμε το πεδίο Όνομα (με ASC ταξινόμηση), όπως φαίνεται στην Εικόνα 4.8. Αν θέλουμε το πεδίο να είναι μοναδικό στην ρύθμιση 'Is Unique', επιλέγουμε 'Yes'.



Εικόνα 4.8

Μετά τη δημιουργία του ευρετηρίου στο πεδίο Όνομα, μπορούμε να κάνουμε δεξί κλικ στον Query Editor και να επιλέξουμε το Estimated Execution Plan. Βλέπουμε τότε, όπως φαίνεται στην Εικόνα 4.9, ότι το πλάνο εκτέλεσης άλλαξε και πλέον ο SQL Server χρησιμοποιεί το index που ορίσαμε για την αναζήτηση και την ταξινόμηση του ερωτήματός μας χωρίς να σπαταλά χρόνο. Βέβαια, δεν μπορούμε να διακρίνουμε εύκολα τις διαφορές στους χρόνους εκτέλεσης ερωτημάτων σε έναν πίνακα με 5000 μόνο εγγραφές. Αντίθετα, οι διαφορές θα ήταν εμφανείς αν πειραματιζόμασταν με τον χρόνο εκτέλεσης του ερωτήματός μας, έχοντας

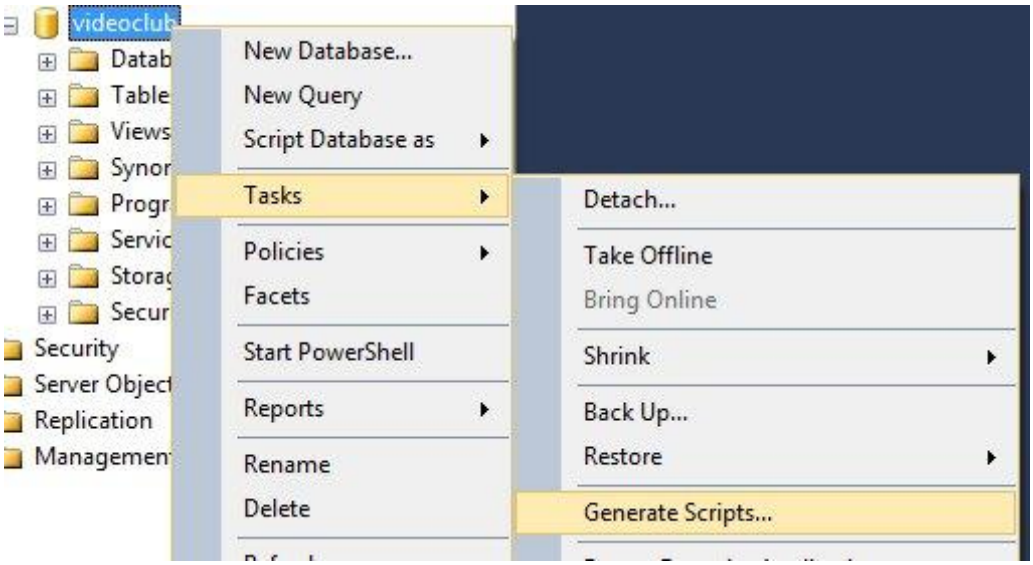
πρώτα εισάγει δύο ή τρία εκατομμύρια πελάτες με την αποθηκευμένη διαδικασία της Ενότητας 4.2.1. Σημειώνουμε ότι στα επόμενα κεφάλαια, όπου διαπραγματευόμαστε τεχνικές αποθηκών και εξόρυξης δεδομένων, θα έχουμε πρόσβαση σε μεγαλύτερες βάσεις δεδομένων.



Εικόνα 4.9

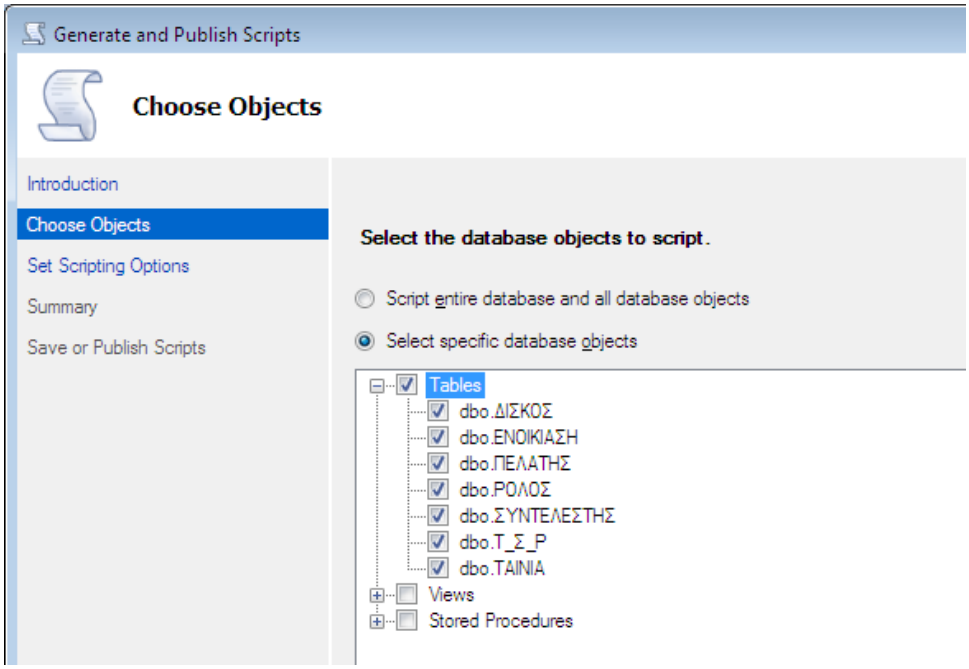
4.4. Εξαγωγή του κώδικα της βάσης δεδομένων

Με τη λειτουργία του Generate Scripts μπορούμε να δημιουργήσουμε ένα αρχείο με εντολές SQL, οι οποίες θα κατασκευάζουν από την αρχή τη βάση δεδομένων μας. Το αρχείο αυτό μπορεί να λειτουργήσει και ως αντίγραφο ασφαλείας για τη βάση δεδομένων, καθώς με την εκτέλεσή του ανακτούμε το σχήμα της βάσης δεδομένων μας. Για να εξάγουμε τον κώδικα SQL, κάνουμε δεξί κλικ στη βάση δεδομένων DVDclub και, όπως φαίνεται στην Εικόνα 4.10, επιλέγουμε τη διαδρομή Tasks και, στη συνέχεια, Generate SQL Script.



Εικόνα 4.10

Στο νέο παράθυρο, που φαίνεται στην Εικόνα 4.11, επιλέγουμε είτε «Script entire database and all database objects» είτε «select specific database objects». Για το παράδειγμά μας ζητάμε την δεύτερη επιλογή, επειδή θέλουμε κάποιο υποσύνολο των αντικειμένων της βάσης δεδομένων, και, στη συνέχεια, επιλέγουμε τα αντικείμενα που θέλουμε να εξαχθούν σε εντολές SQL.

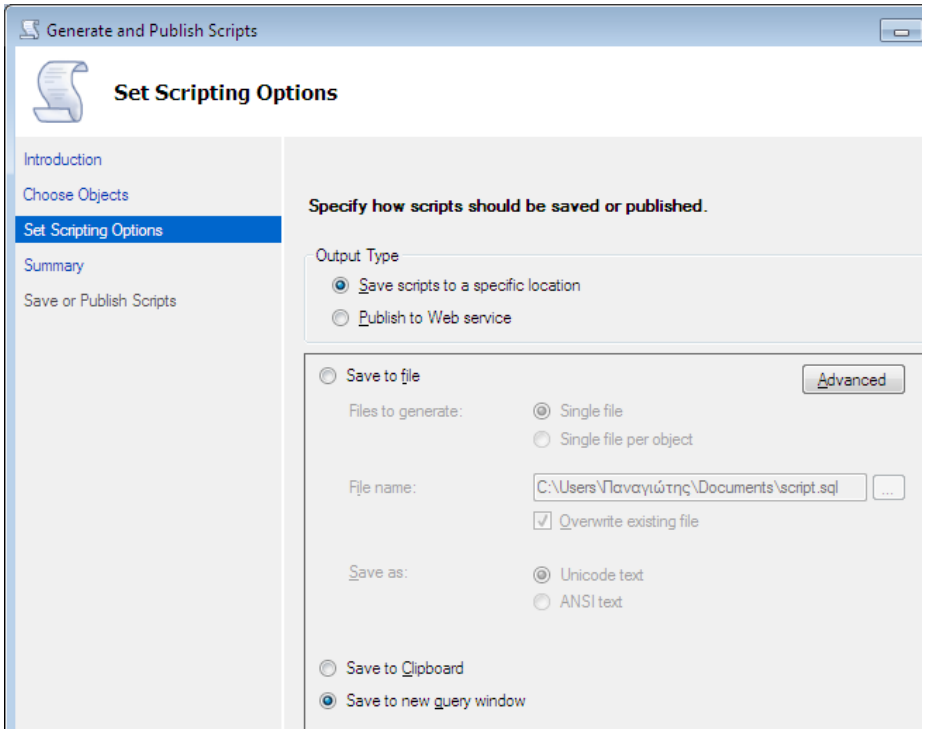


Εικόνα 4.11

Στη συνέχεια, πρέπει να επιλέξουμε τον τρόπο που θα αποθηκεύσουμε τον κώδικα που θα παραχθεί. Έχουμε τις παρακάτω εναλλακτικές επιλογές:

- Να τον αποθηκεύσουμε σε αρχείο (save to file).
- Να το αντιγράψουμε στο πρόχειρο (save to clipboard).
- Να το εμφανίσουμε σε ένα νέο παράθυρο ερωτήματος (save to new query window).

Εμείς επιλέγουμε, για το παράδειγμά μας, την τρίτη επιλογή, όπως φαίνεται στην Εικόνα 4.12:



Εικόνα 4.12

Τέλος, αφού πατήσουμε Next, σε ένα νέο παράθυρο ερωτήματος (αυτό που φαίνεται στην Εικόνα 4.13) προβάλλεται ο κώδικας που δημιουργεί όλα τα αντικείμενα της βάσης δεδομένων.

```
/****** Object:  Table [dbo].[ΕΝΟΙΚΙΑΣΗ]    Script Date: 4/7/2013 9:10:10 πμ ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[ΕΝΟΙΚΙΑΣΗ](
    [IDΠελάτη] [int] NOT NULL,
    [IDΚασέτας] [int] NOT NULL,
    [Από] [date] NOT NULL,
    [Έως] [date] NULL,
 CONSTRAINT [PK_ΕΝΟΙΚΙΑΣΗ] PRIMARY KEY CLUSTERED
(
    [IDΠελάτη] ASC,
    [IDΚασέτας] ASC,
    [Από] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCK
) ON [PRIMARY]

GO
/****** Object:  Table [dbo].[ΚΑΣΕΤΕΣ]    Script Date: 4/7/2013 9:10:10 πμ ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[ΚΑΣΕΤΕΣ](
    [Id] [int] NOT NULL,
    [IDΤαινίας] [int] NOT NULL,
    [Τύπος] [varchar](4) NOT NULL,
    [Ποσότητα] [tinyint] NOT NULL,
    [Τιμή] [decimal](9, 2) NOT NULL,
 CONSTRAINT [PK_ΚΑΣΕΤΕΣ] PRIMARY KEY CLUSTERED
```

Εικόνα 4.13

Προσοχή! Πρέπει πρώτα να έχει δημιουργηθεί η βάση δεδομένων (create), ώστε να μπορούν να φτιαχτούν όλα τα αντικείμενά της με τον κώδικα που μόλις παράχθηκε. Επιπλέον, θα πρέπει να τονιστεί ότι αυτή η διαδικασία δεν εξάγει τα δεδομένα της βάσης, παρά μόνο το σχήμα της. Προκειμένου να εξάγουμε και αυτά, θα πρέπει να πατήσουμε το κουμπί Advanced της Εικόνας 4.12 και, στη συνέχεια, στο option 'Types of data to script' να επιλέξουμε Schema and Data.

Θυμίζουμε, επιπροσθέτως, ότι μπορούμε ανά πάσα στιγμή να παράξουμε τον κώδικα δημιουργίας της βάσης δεδομένων και από την επιλογή Script Database as, ακολουθώντας τη διαδρομή Create To και, στη συνέχεια, New Query Editor Window, όπως φαίνεται στην Εικόνα 4.14.

Εικόνα 4.14

4.5. Εκχώρηση δικαιωμάτων πρόσβασης χρηστών στη βάση δεδομένων

Στόχος της παρούσας ενότητας είναι η δημιουργία τριών διαβαθμισμένων χρηστών (manager, employee, customer), οι οποίοι θα έχουν διαφορετικά προνόμια πρόσβασης στη βάση DVDclub, σύμφωνα με τον παρακάτω πίνακα:

Ενέργειες / Πίνακας	Select	Insert	Update	Delete
ΠΕΛΑΤΗΣ	Manager	Manager	Manager	Manager
	Employee	Employee	Employee	
ΕΝΟΙΚΙΑΣΗ	Manager	Manager	Manager	Manager
	Employee	Employee	Employee	
		Customer		
ΣΥΝΤΕΛΕΣΤΗΣ	Manager	Manager	Manager	Manager
	Employee	Employee	Employee	
	Customer			
ΔΙΣΚΟΣ	Manager	Manager	Manager	Manager
	Employee	Employee	Employee	
	Customer			
Τ_Σ_Ρ	Manager	Manager	Manager	Manager
	Employee	Employee	Employee	
	Customer			
ΤΑΙΝΙΑ	Manager	Manager	Manager	Manager
	Employee	Employee	Employee	
	Customer			
ΡΟΛΟΣ	Manager	Manager	Manager	Manager
	Employee	Employee	Employee	
	Customer			

Πίνακας 4.10

Παρακάτω θα δημιουργήσουμε ένα login με το οποίο θα έχουμε πρόσβαση στον sql server. Στη συνέχεια, θα δημιουργήσουμε έναν χρήστη - employee, ο οποίος θα αναφέρεται σ' αυτό το login. Τονίζεται ότι με τον ίδιο τρόπο μπορούμε να δημιουργήσουμε και άλλους χρήστες. Το login αφορά όλο τον sql server, ενώ ο χρήστης – employee αφορά μόνο τη βάση DVDclub.

```
Create login employee with password = '123'

go

use DVDclub
Create user employee for login employee with default_schema=[dbo]
```

4.5.1. Εκχώρηση δικαιωμάτων χρήστη

Για την εκχώρηση ενός προνομίου ο διαχειριστής της βάσης δεδομένων πρέπει να χρησιμοποιεί την εντολή grant.

Η γενική σύνταξη της εντολής είναι:

GRANT privilege_name
ON object_name
TO {user_name |PUBLIC |role_name}
[WITH GRANT OPTION];

Οι όροι της παραπάνω εντολής επεξηγούνται ως εξής:

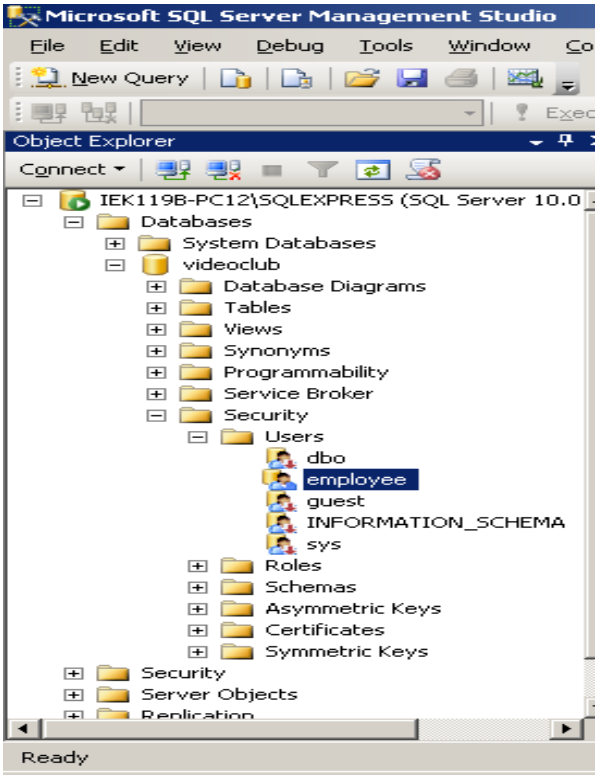
- **Privilege_name** είναι το προνόμιο ή δικαίωμα πρόσβασης το οποίο εκχωρείται στο χρήστη. Μερικά από τα δικαιώματα πρόσβασης είναι ALL, EXECUTE και SELECT.
- **Object_name** είναι το όνομα ενός αντικειμένου της βάσης (πίνακας, όψη).
- **User_name** είναι το όνομα του χρήστη στον οποίο εκχωρείται ένα δικαίωμα πρόσβασης.
- **PUBLIC** αφορά την εκχώρηση δικαιωμάτων πρόσβασης σε όλους τους χρήστες.
- **ROLES** είναι ένα σύνολο δικαιωμάτων που ομαδοποιούνται.
- **WITH GRANT OPTION** είναι προαιρετικό και επιτρέπει σ' έναν χρήστη να εκχωρήσει δικαιώματα πρόσβασης σε άλλους χρήστες.

4.5.2. Εκχώρηση δικαιωμάτων στο χρήστη Employee με κώδικα SQL.

Σ' αυτήν την ενότητα θα περιγράψουμε την εκχώρηση δικαιωμάτων στον χρήστη employee. Θέλουμε ο employee να έχει πλήρη πρόσβαση στη βάση δεδομένων, ώστε να μπορεί να δημιουργεί καινούριες εγγραφές και να τροποποιεί τις υπάρχουσες.

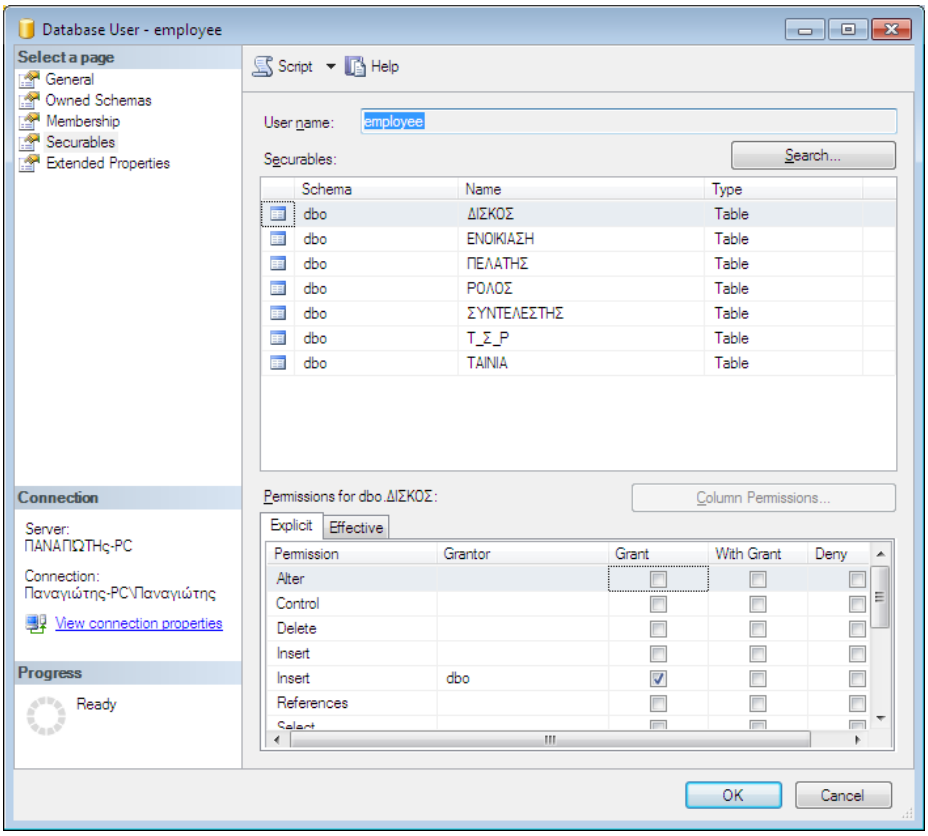
grant select, insert, update on ΠΕΛΑΤΗΣ to employee
grant select, insert, update on ΕΝΟΙΚΙΑΣΗ to employee
grant select, insert, update on ΣΥΝΤΕΛΕΣΤΗΣ to employee
grant select, insert, update on ΔΙΣΚΟΣ to employee
grant select, insert, update on Τ_Σ_Ρ to employee
grant select, insert, update on ΤΑΙΝΙΑ to employee
grant select, insert, update on ΡΟΛΟΣ to employee

Πατάμε το **execute** και δημιουργείται ο χρήστης employee με τα επιθυμητά δικαιώματα. Για να επαληθεύσουμε τη δημιουργία του χρήστη, κάνουμε κλικ στο Security και, στη συνέχεια, κάνουμε κλικ στο φάκελο Logins, όπου πρέπει να υπάρχει ο νέος χρήστης, όπως φαίνεται στην Εικόνα 4.15.



Εικόνα 4.15

Αφού επαληθεύσουμε τη δημιουργία του χρήστη employee, κάνουμε διπλό κλικ στον χρήστη. Στο παράθυρο που εμφανίζεται, επιλέγουμε το Securables, για να επαληθεύσουμε ότι έχει πάρει τα δικαιώματα που θέλουμε, όπως φαίνεται στην Εικόνα 4.16.



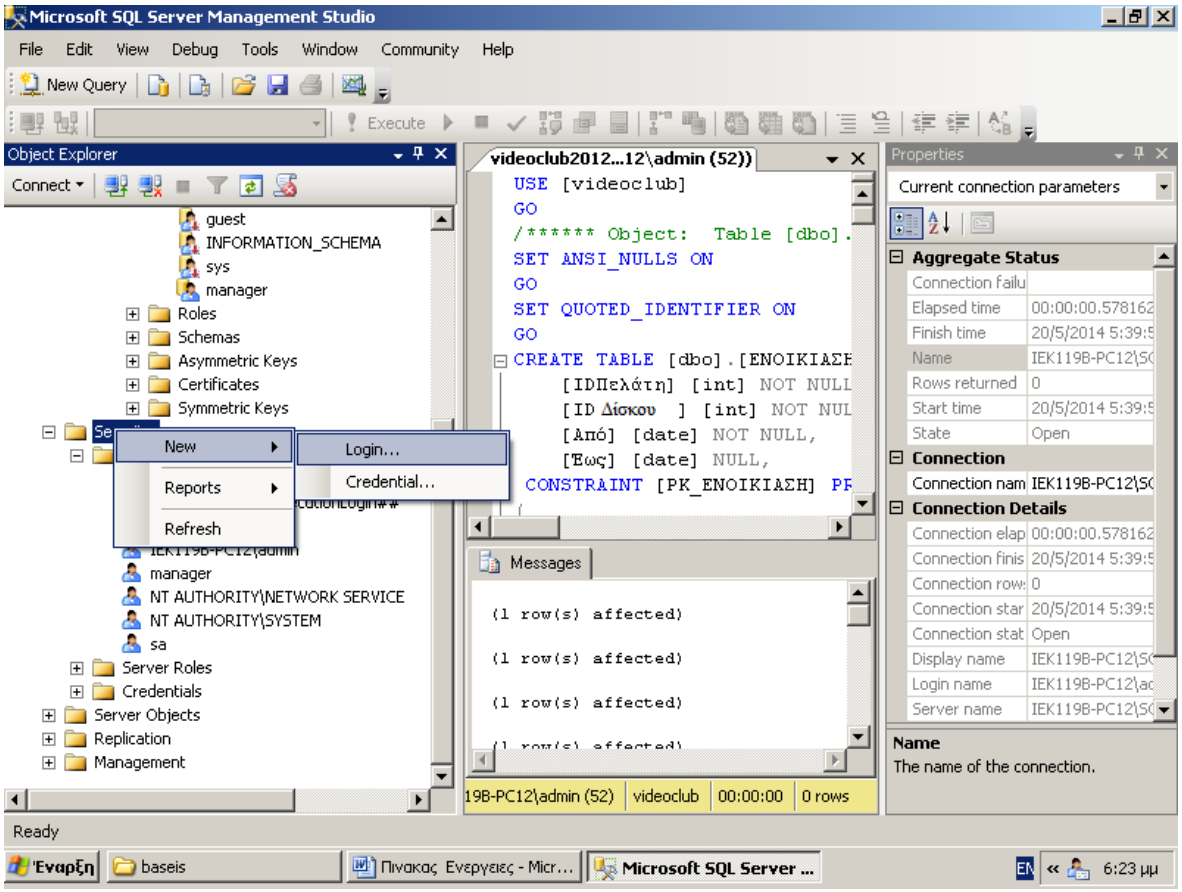
Εικόνα 4.16

4.5.3. Εκχώρηση δικαιωμάτων στο χρήστη Manager με γραφικό τρόπο

Σε αυτή την ενότητα θα δείξουμε πώς γίνεται η εκχώρηση δικαιωμάτων στο χρήστη Manager με γραφικό τρόπο. Η διαδικασία είναι ακριβώς ίδια μ' αυτήν που περιγράψαμε στην προηγουμένη ενότητα, μόνο που τώρα γίνεται με γραφικό τρόπο.

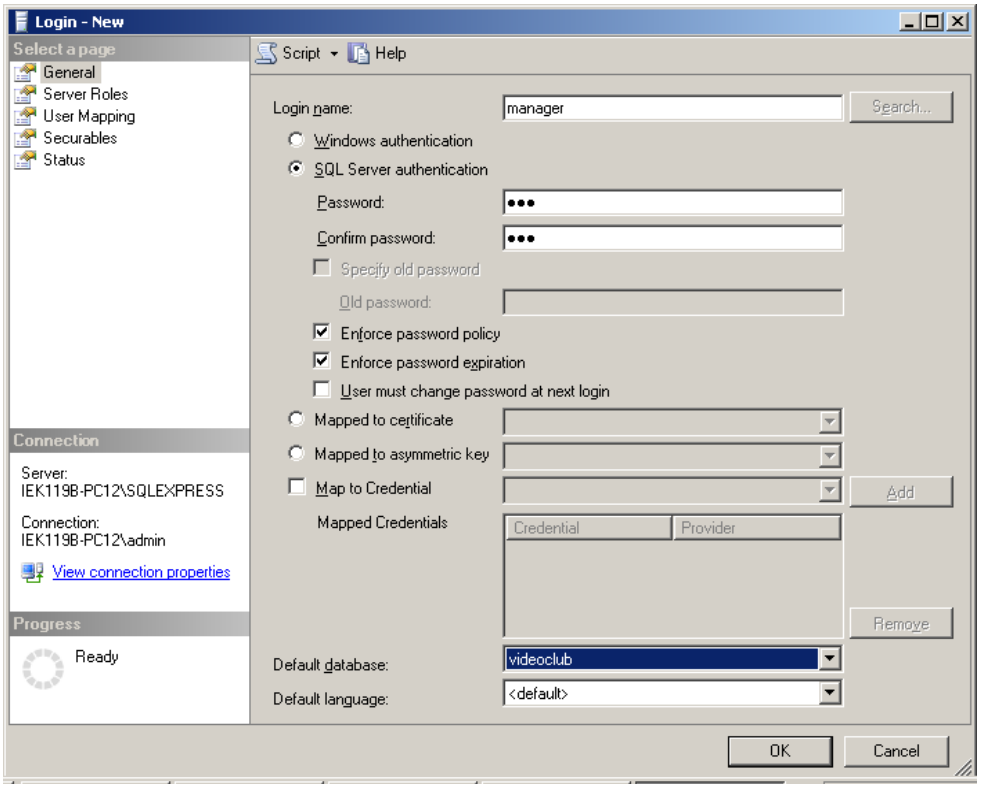
Για να δημιουργήσουμε τον χρήστη Manager, ακολουθούμε τα εξής βήματα:

Βήμα 1: Καθώς βρισκόμαστε στο κεντρικό περιβάλλον του αριστερού pane στον sql server, επιλέγουμε τη διαδρομή Security -> New -> Login, όπως φαίνεται στην Εικόνα 4.17.



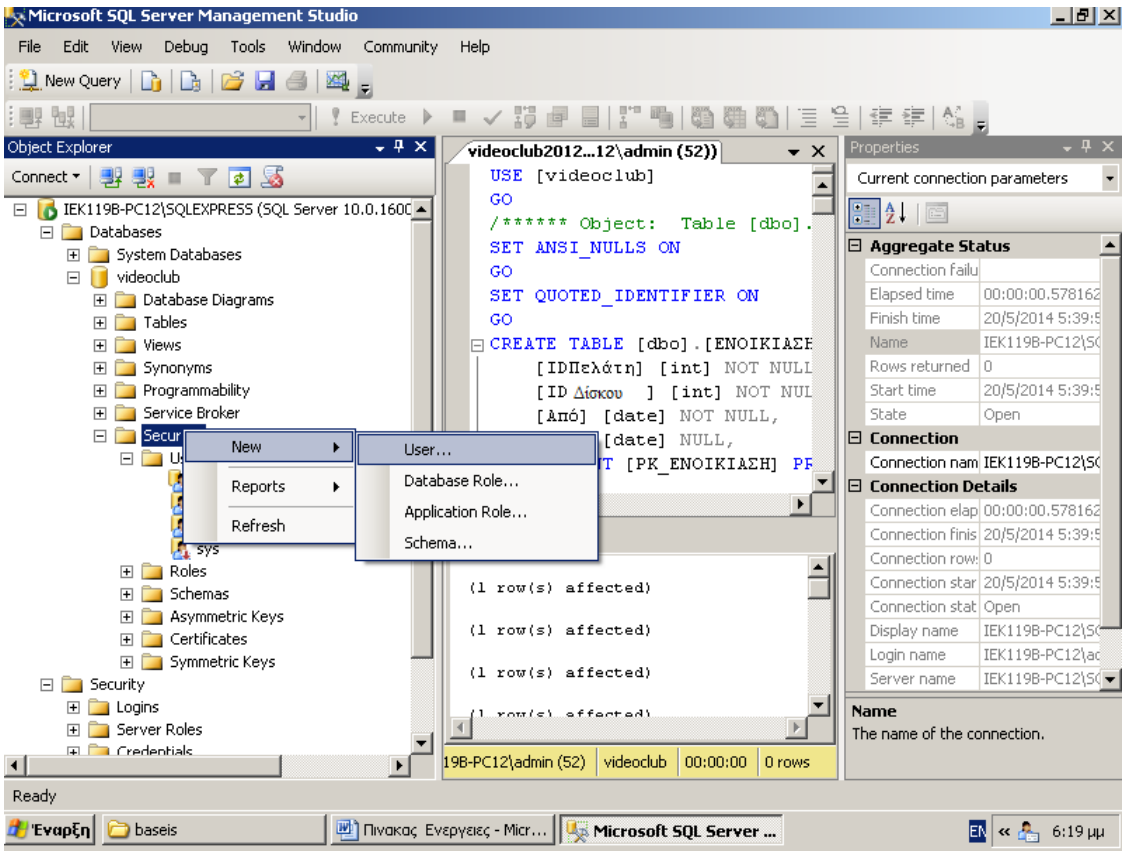
Εικόνα 4.17

Βήμα 2: Εμφανίζεται ένα νέο παράθυρο, το οποίο συμπληρώνουμε όπως φαίνεται στην Εικόνα 4.18. Πιο συγκεκριμένα, στο Login name πληκτρολογούμε manager. Επιλέγουμε με κλικ το Sql Server Authentication και συμπληρώνουμε (πληκτρολογώντας: 123) τα πεδία Password και Confirm password. Αποεπιλέγουμε το User must change password at next login. Συμπληρώνουμε το πεδίο Default database πληκτρολογώντας (επιλέγοντας) το όνομα της βάσης μας, δηλαδή: DVDclub. Πατάμε OK.



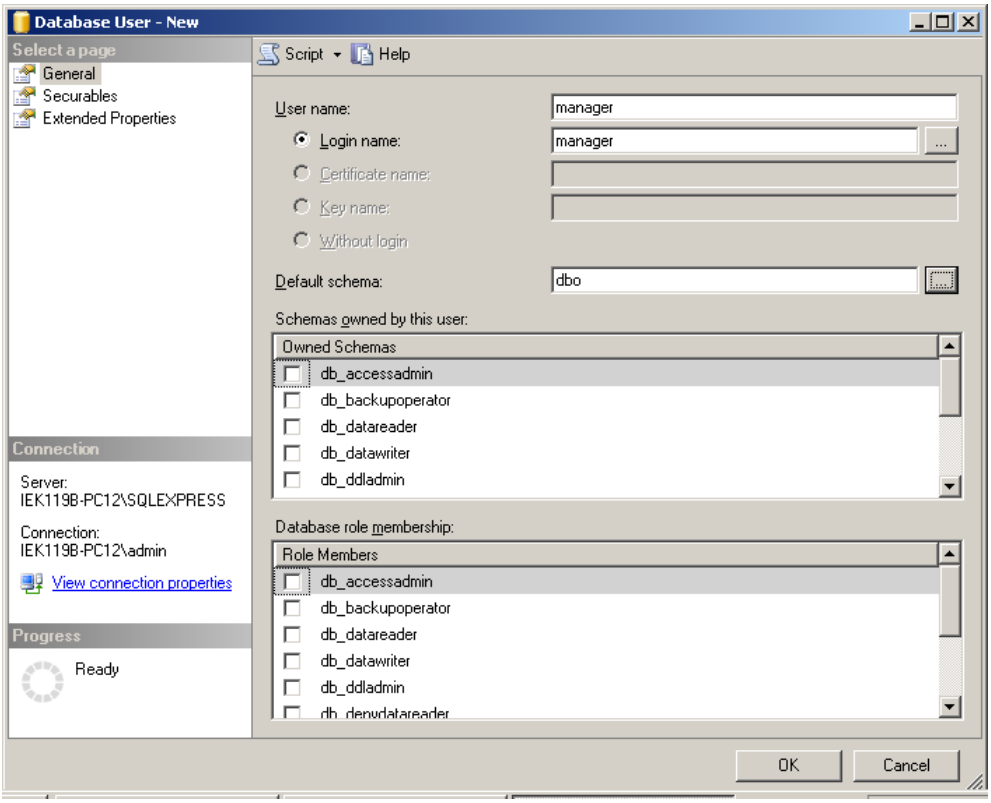
Εικόνα 4.18

Βήμα 3: Κάνουμε δεξί κλικ στο Security και επιλέγουμε τη διαδρομή New → User στο περιβάλλον της βάσης δεδομένων DVDclub, όπως φαίνεται στην Εικόνα 4.19.



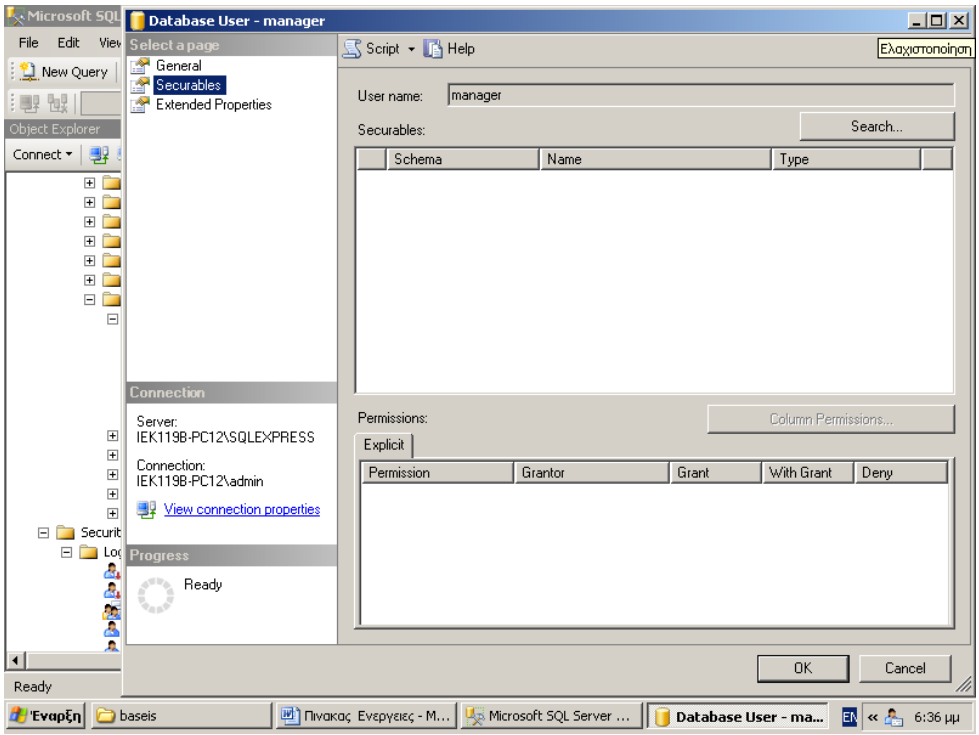
Εικόνα 4.19

Βήμα 4: Πληκτρολογούμε την λέξη manager στα πεδία User name και Login name, όπως φαίνεται στην Εικόνα 4.20, και πατάμε OK.



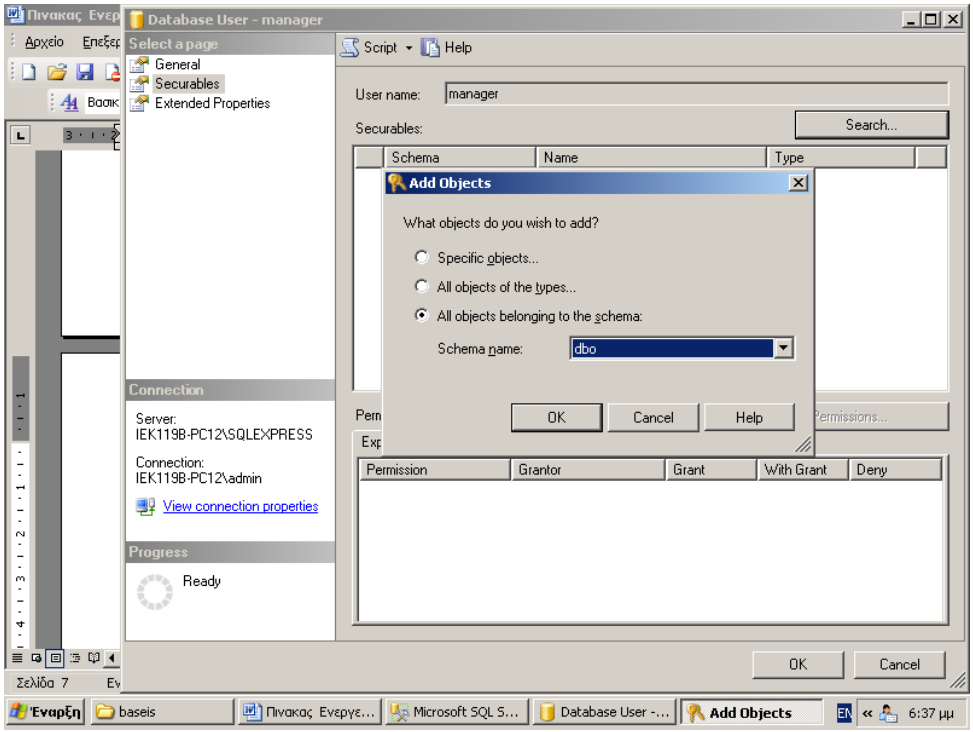
Εικόνα 4.20

Βήμα 5: Στη συνέχεια κάνουμε δεξί κλικ στην επιλογή Security, επιλέγουμε το Users και, τέλος, στο χρήστη manager που δημιουργήσαμε, κάνουμε διπλό κλικ και επιλέγουμε το Securables (βλέπε Εικόνα 4.21).



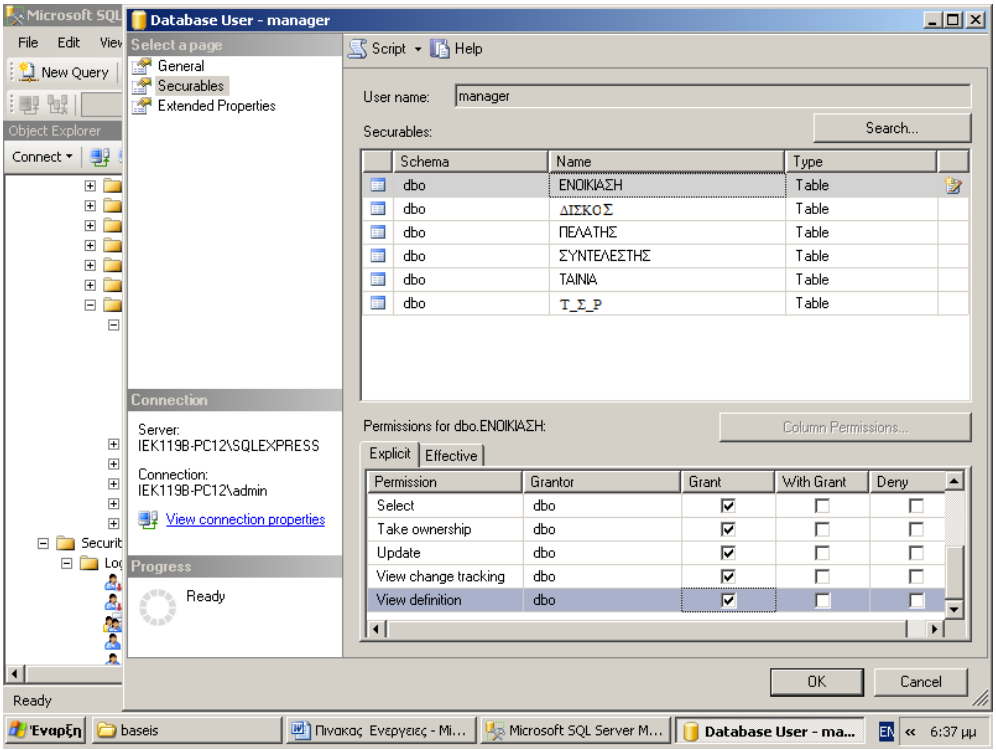
Εικόνα 4.21

Βήμα 6: Πατάμε το Search και, στο παράθυρο που εμφανίζεται, επιλέγουμε All objects belonging to the schema. Στο schema name επιλέγουμε dbo. Η Εικόνα 4.22 εμφανίζει τις επιλογές αυτού του βήματος.



Εικόνα 4.22

Μ' αυτόν τον τρόπο εμφανίζονται οι πίνακες της βάσης, ώστε να εκχωρήσουμε τα δικαιώματα στον καθένα χωριστά. Για παράδειγμα, όπως φαίνεται στην Εικόνα 4.23), επιλέγουμε τον πίνακα Ενοικίαση και τσεκάρουμε όλα τα κουτάκια στη στήλη Grant. Η ίδια διαδικασία πρέπει να γίνει σε όλους τους πίνακες.



Εικόνα 4.23

Προσοχή!: Πρέπει, πριν πατήσουμε OK, να έχουμε κάνει τη διαδικασία του Βήματος 6 για όλους τους πίνακες. Διαφορετικά, θα πρέπει να ξαναμπούμε στο Securables και να επαναλάβουμε τη διαδικασία από το Βήμα 5 και μετά.

4.5.4. Αφαίρεση δικαιωμάτων από τον χρήστη

Στην ενότητα αυτή θα αναλύσουμε τον τρόπο λειτουργίας της εντολής revoke, με την οποία αφαιρούμε τα δικαιώματα που μπορεί να έχει κάποιος χρήστης σε έναν ή περισσότερους πίνακες. Η σύνταξη της εντολής είναι η ακόλουθη:

Ανάλυση της εντολής **Revoke**.

REVOKE privilege_name
ON object_name
FROM {user_name |PUBLIC |role_name}

Για παράδειγμα: Revoke select on TAINIA FROM customer.

Αυτή η εντολή θα ανακαλέσει το δικαίωμα εμφάνισης που έχει ο χρήστης customer στον πίνακα TAINIA. Όταν γίνει η αναίρεση του δικαιώματος εμφάνισης σε έναν πίνακα για έναν χρήστη, ο χρήστης δεν θα μπορεί να εμφανίζει δεδομένα από αυτόν τον πίνακα. Ωστόσο, αν ο χρήστης έχει λάβει δικαιώματα εμφάνισης για τον συγκεκριμένο πίνακα από περισσοτέρους από έναν χρήστες, τότε θα μπορεί να εμφανίζει δεδομένα από αυτόν τον πίνακα μέχρι αυτοί που τα έχουν εκχωρήσει να κάνουν αναίρεση.

4.5.5. Άρνηση δικαιωμάτων σε χρήστη

Σ' αυτήν την ενότητα θα εξηγήσουμε τον τρόπο λειτουργίας της εντολής Deny, η οποία κάνει άρνηση δικαιωμάτων σε χρήστη. Η σύνταξη της Deny είναι η εξής:

Deny ALL | permission_name
On object_name
TO user_name

Παράδειγμα: **Deny update On TAINIA To customer**

Αυτή η εντολή απαγορεύει στον χρήστη customer να ενημερώσει τα περιεχόμενα του πίνακα TAINIA.

4.6. Ασκήσεις

1. **Σ**τον πίνακα ΤΑΙΝΙΑ της βάσης δεδομένων DVDclub να δημιουργήσετε ένα νέο πεδίο με όνομα 'Language' και τύπο δεδομένων char(2). Το πεδίο θα έχει default αρχική τιμή ('En'), που σημαίνει ότι η γλώσσα της ταινίας είναι Αγγλική.

2. Να δημιουργηθεί μια διαδικασία (Stored Procedure) που θα εισάγει (κάθε φορά που την καλούμε) Ν εγγραφές ταινιών (με τυχαία αλφαριθμητικά ή αριθμητικά δεδομένα αντίστοιχα) στον πίνακα ΤΑΙΝΙΑ.

3. Να δημιουργήσετε ένα trigger για την περίπτωση της επιστροφής ενός ψηφιακού δίσκου που είχε προηγουμένως ενοικιαστεί.

4. Με γραφικό τρόπο μέσα από το περιβάλλον του SQL Server, να εκχωρήσετε δικαιώματα για τον χρήστη Customer στους πίνακες ΣΥΝΤΕΛΕΣΤΗΣ και ΔΙΣΚΟΣ, όπως αυτά τα δικαιώματα προβλέπονται στον πίνακα 4.10.

5. Να γίνει η εκχώρηση δικαιωμάτων για το χρήστη Customer στους πίνακες ΡΟΛΟΣ και ΤΑΙΝΙΑ, όπως αυτά τα δικαιώματα προβλέπονται στον πίνακα 4.10. Η εκχώρηση των δικαιωμάτων να γίνει με εντολές της SQL.

4.7. Βιβλιογραφία/Αναφορές

Hoffer, J. A., Venkatarama, R., & Topi, H. (2013). *Modern Database Management*, Prentice Hall.

Μανωλόπουλος, Ι., & Παπαδόπουλος, Α. Ν. (2006). *Συστήματα Βάσεων Δεδομένων: Θεωρία & Πρακτική Εφαρμογή*, Αθήνα, Εκδόσεις Νέων Τεχνολογιών.

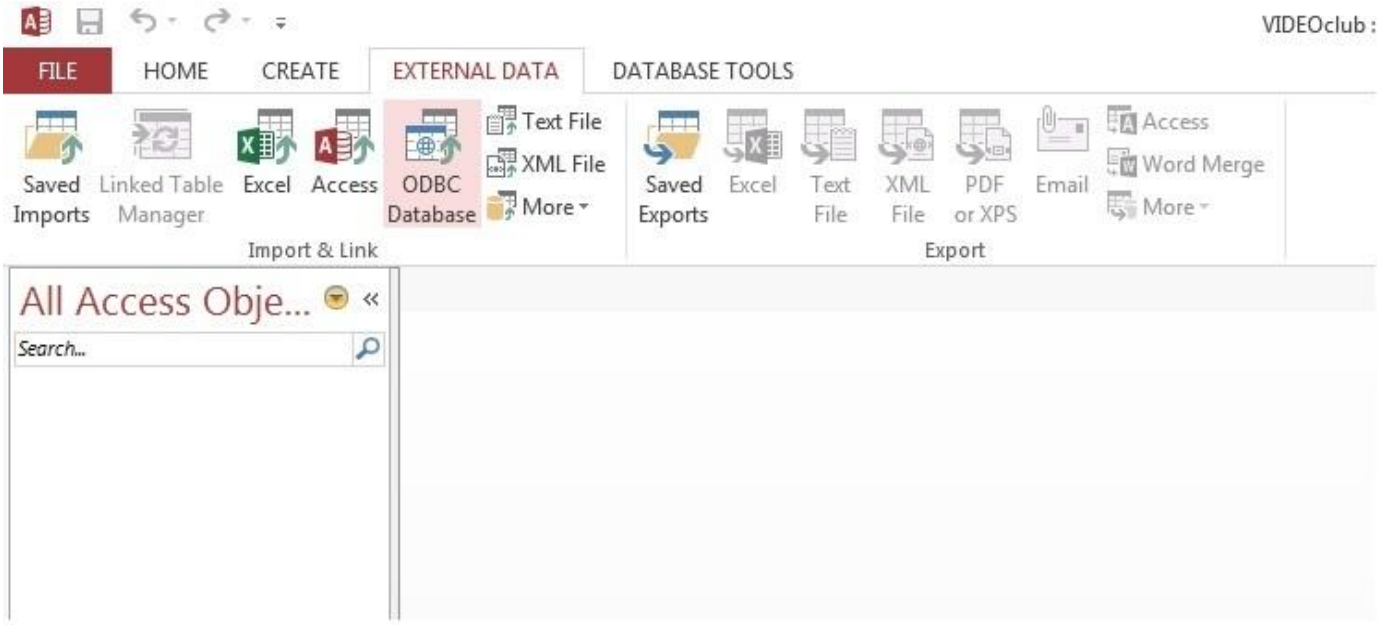
Κεφάλαιο 5. Δημιουργία φορμών για τη βάση δεδομένων DVDclub

Σύνοψη
Σ' αυτό το κεφάλαιο θα περιγράψουμε τη δημιουργία φορμών, προκειμένου να εισάγουμε δεδομένα και να εμφανίζουμε στοιχεία από τους πίνακες της βάσης DVDclub με τυποποιημένο τρόπο και σε φιλικότερο περιβάλλον διεπαφής με τον χρήστη. Συγκεκριμένα, θα δούμε πώς μπορούμε να φτιάχνουμε απλές κύριες φόρμες, καθώς και κύριες με δευτερεύουσες φόρμες στο περιβάλλον της MS Access. Επιπροσθέτως, θα μελετήσουμε τη δημιουργία λιστών αναζήτησης, στοχεύοντας στην γρήγορη εύρεση στοιχείων σε μια βάση δεδομένων και στη δημιουργία υπολογιζόμενων πεδίων φορμών (derived attributes).

5.1. Δημιουργία συνδεδεμένων πινάκων από τον SQL Server στην Access 2013 του Microsoft Office

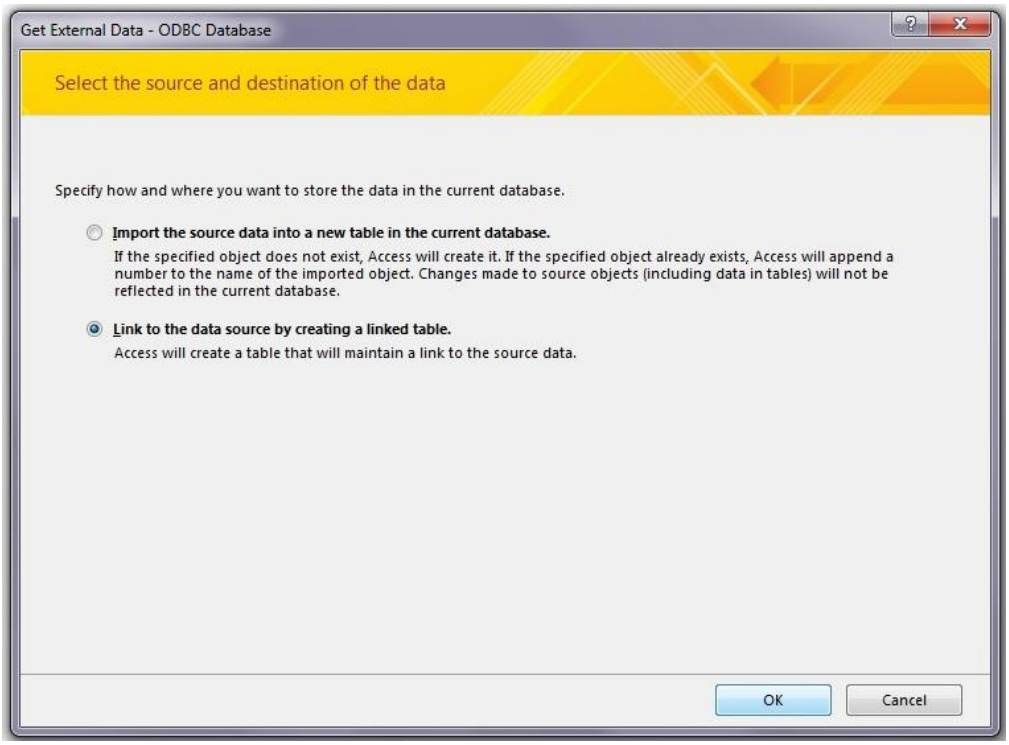
Βήμα 1: Προκειμένου να εισάγουμε τους πίνακες της βάσης δεδομένων DVDclub στο περιβάλλον της Access, πρέπει προηγουμένως να μεταβούμε στον Πίνακα Ελέγχου και στα Εργαλεία Διαχείρισης, για να δημιουργήσουμε ένα αρχείο ODBC (Open Data Base Connectivity) με όνομα DVDclubODBC. Με τη βοήθεια αυτού του αρχείου μπορούμε να εισάγουμε τα δεδομένα του MS SQL Server σε βάσεις δεδομένων άλλων εταιρειών (π.χ. ORACLE, MYSQL, κτλ.).

Βήμα 2: Ανοίγουμε την Access και δημιουργούμε μια κενή βάση δεδομένων. Στη συνέχεια, από την καρτέλα EXTERNAL DATA επιλέγουμε ODBC Database, όπως φαίνεται στην Εικόνα 5.1.



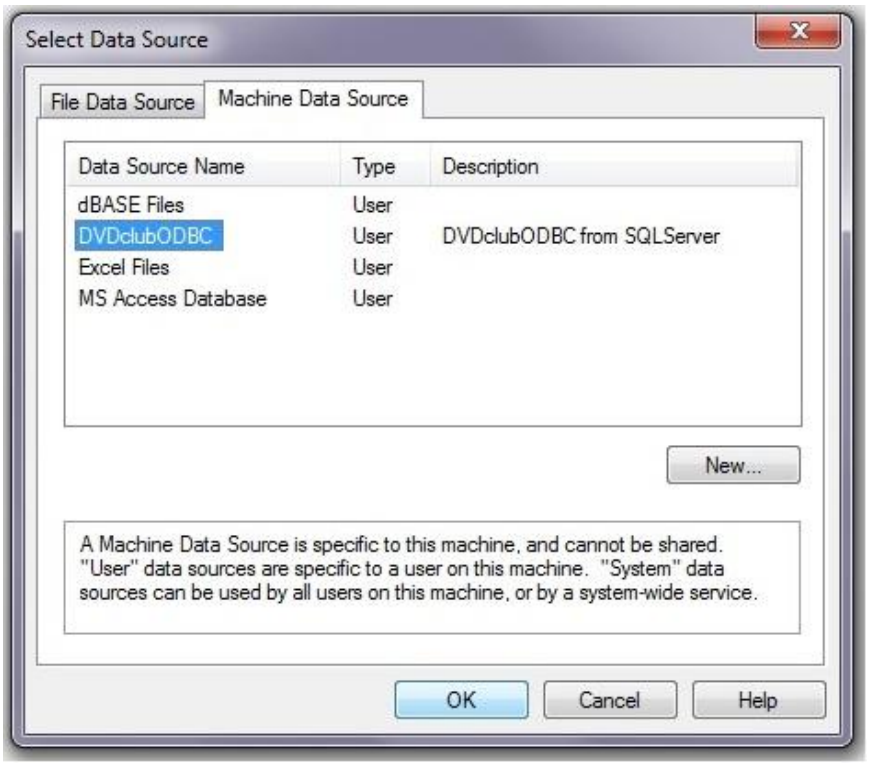
Εικόνα 5.1

Βήμα 3: Επιλέγουμε Link to the data source by creating a link table, όπως φαίνεται στην Εικόνα 5.2, και πατάμε ok.



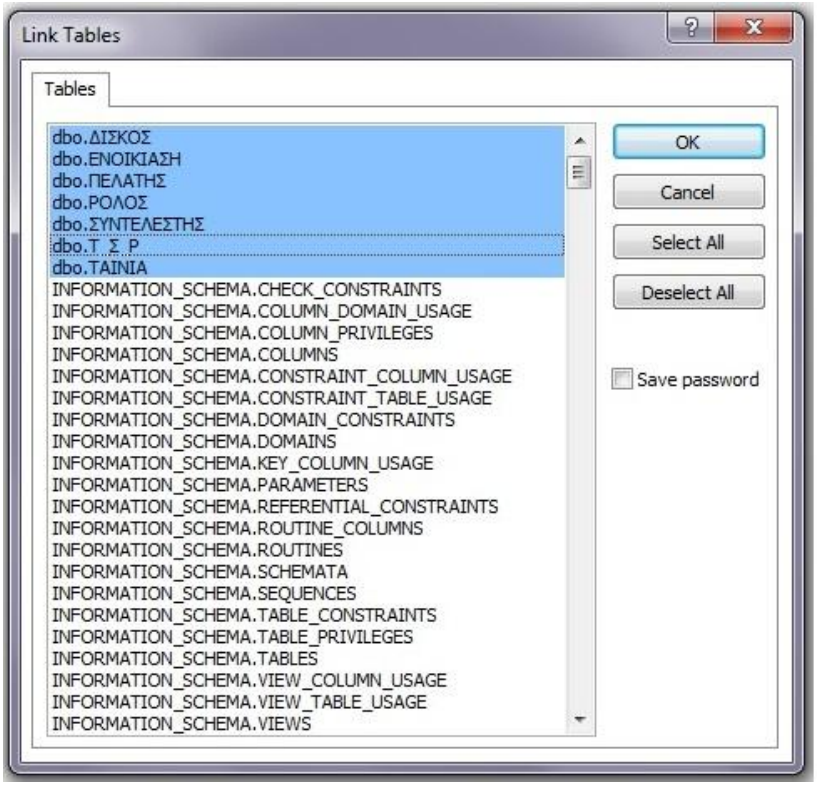
Εικόνα 5.2

Βήμα 4: Επιλέγουμε την καρτέλα Machine Data Source και επιλέγουμε το DVDclubOBDC που έχουμε δημιουργήσει, όπως φαίνεται στην Εικόνα 5.3.



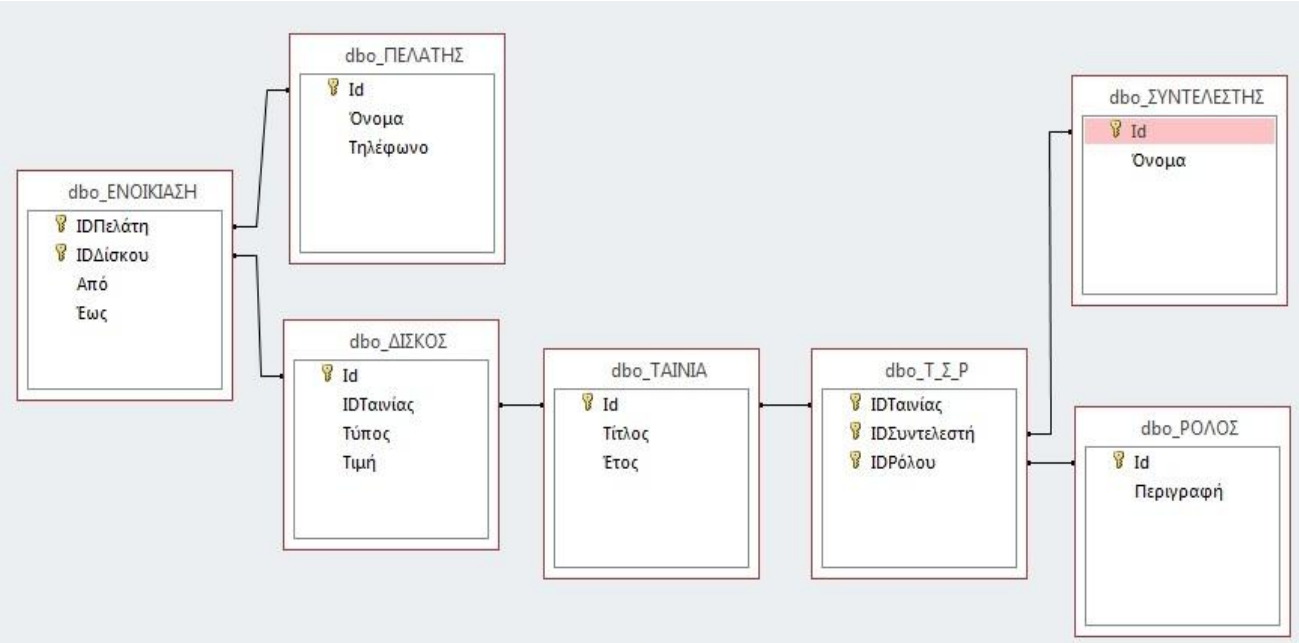
Εικόνα 5.3

Βήμα 5: Επιλέγουμε όλα τα TABLES, όπως φαίνεται στην Εικόνα 5.4, και πατάμε οκ.



Εικόνα 5.4

Βήμα 6: **Προσοχή!** Από την επιλογή DATABASE TOOLS -> Relationships θα πρέπει να συνδέσουμε ξανά τους πίνακες μεταξύ τους από την αρχή, όπως φαίνεται στην Εικόνα 5.5.

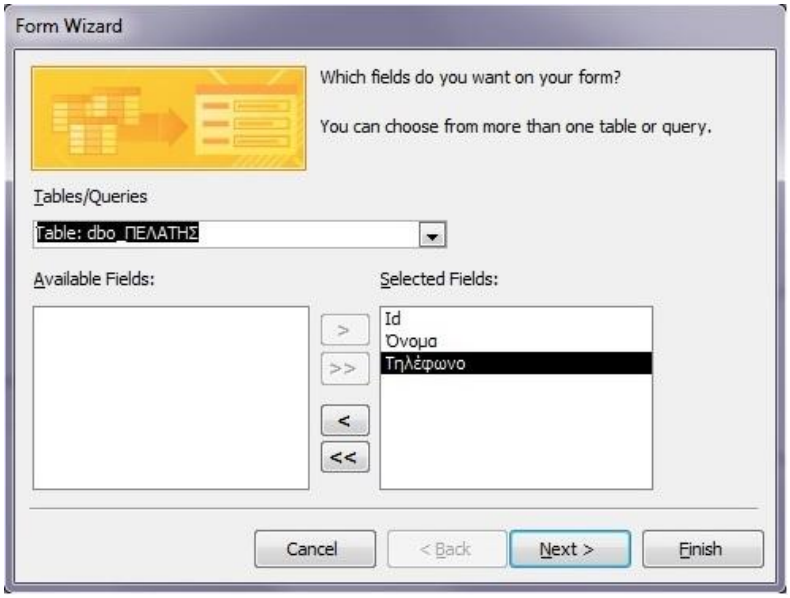


Εικόνα 5.5

5.2. Δημιουργία απλής φόρμας εισαγωγής στοιχείων και σύνθετης κύριας/ δευτερεύουσας φόρμας

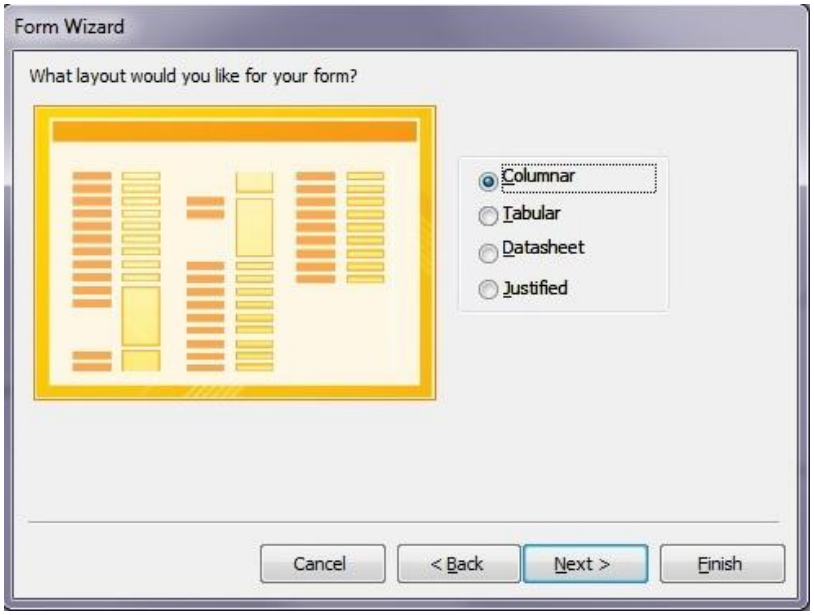
5.2.1. Δημιουργία μιας απλής φόρμας εισαγωγής στοιχείων Πελατών

Επιθυμούμε να δημιουργήσουμε μια φόρμα εισαγωγής για την καταχώριση και την προβολή των Πελατών. Επιλέγουμε από την καρτέλα CREATE -> Form Wizard, όπως φαίνεται στην Εικόνα 5.6. Στο πεδίο Tables/Queries επιλέγουμε τον πίνακα που μας ενδιαφέρει και, πατώντας το εικονίδιο >> , επιλέγουμε όλα τα πεδία του πίνακα, για να περάσουν στη φόρμα.



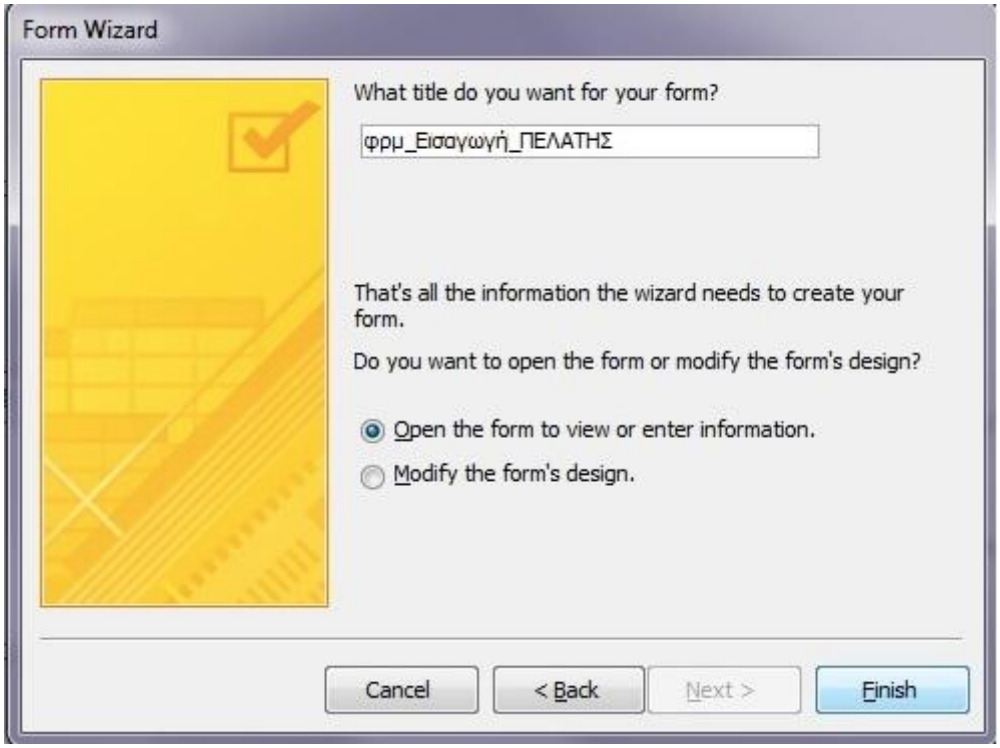
Εικόνα 5.6

Επιλέγουμε Columnar, όπως φαίνεται στην Εικόνα 5.7. Η συγεκριμένη επιλογή σημαίνει ότι θα βλέπουμε στη φόρμα μας μια εγγραφή κάθε φορά με όλα τα πεδία της. Η επιλογή tabular σημαίνει ότι θα βλέπουμε όλες τις εγγραφές με την μορφή ενός πίνακα. Η επιλογή Datasheet ισοδυναμεί με εμφάνιση όπως στο excel, όπου επεξεργαζόμαστε αμορφοποίητα τα πεδία-στήλες σε πολλές γραμμές-εγγραφές. Πατάμε Next.



Εικόνα 5.7

Προκειμένου να θυμόμαστε ότι πρόκειται για φόρμα, καλό είναι να μετονομάσουμε τη φόρμα που μόλις φτιάξαμε σε **φρμ_Εισαγωγή_ΠΕΛΑΤΗΣ**, όπως φαίνεται στην Εικόνα 5.8. Τέλος, πατάμε Finish.

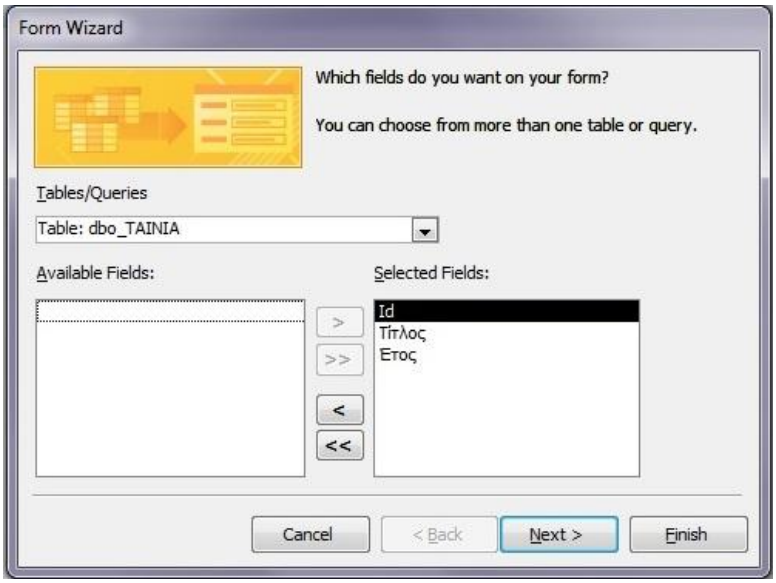


Εικόνα 5.8

5.2.2. Δημιουργία Κύριας και Δευτερεύουσας φόρμας

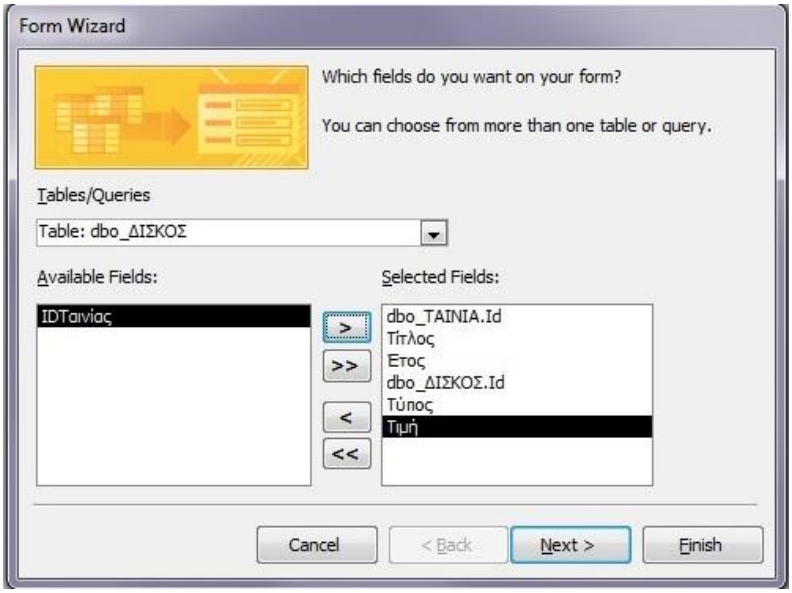
5.2.2.1. Δημιουργία φόρμας ταινίας- δίσκων με τη χρήση μόνο πινάκων

Η ενότητα αυτή αφορά τις οντότητες που σχετίζονται στο E-R διάγραμμά μας (Εικόνα 2.1) με σχέσεις ένα προς πολλά. Στόχος μας είναι η δημιουργία κύριας και δευτερεύουσας φόρμας;, προκειμένου να καταχωρούνται τα αντίτυπα δίσκων της ίδιας ταινίας. Επιλέγουμε από την καρτέλα CREATE -> Form Wizard. Στο πεδίο Tables/Queries επιλέγουμε τον πίνακα dbo_TAINIA και, πατώντας το >> , επιλέγουμε όλα τα πεδία του πίνακα, όπως φαίνεται στην Εικόνα 5.9, για να περάσουν στη φόρμα.



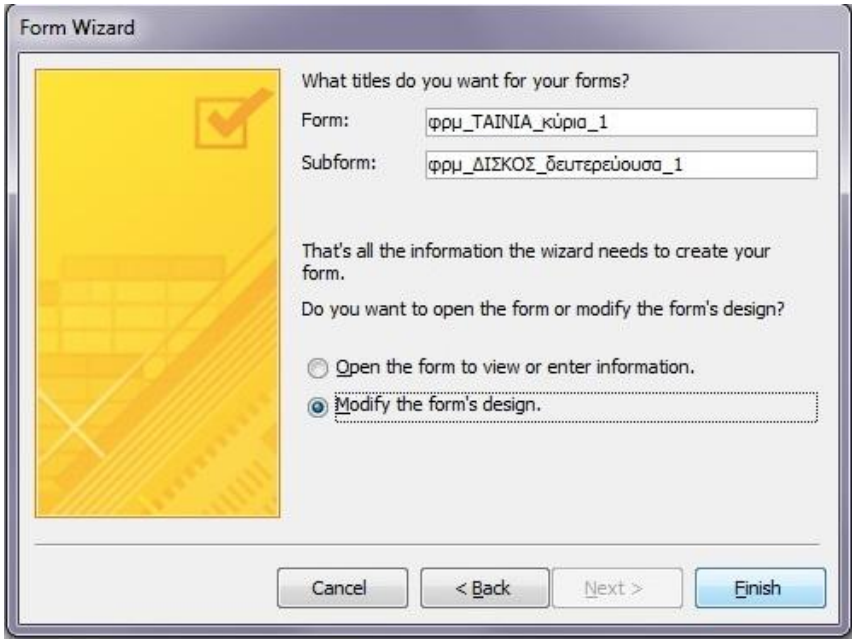
Εικόνα 5.9

Στη συνέχεια, χωρίς να πατήσουμε ακόμη το next, επιλέγουμε τον πίνακα dbo_ΔΙΣΚΟΣ και παίρνουμε όλα τα πεδία του, εκτός από το IDΤαινίας, όπως φαίνεται στην Εικόνα 5.10.



Εικόνα 5.10

Επειδή στο τέλος θα έχουμε πολλές κύριες – δευτερεύουσες φόρμες και δεν θα ξέρουμε ποια αντιστοιχεί σε ποια, στο σημείο αυτό μετονομάζουμε την κύρια σε φρμ_ΤΑΙΝΙΑ_κύρια_1 και τη δευτερεύουσα σε φρμ_ΔΙΣΚΟΣ _Δευτερεύουσα_1, όπως φαίνεται στην Εικόνα 5.11. Τέλος, πατάμε finish.

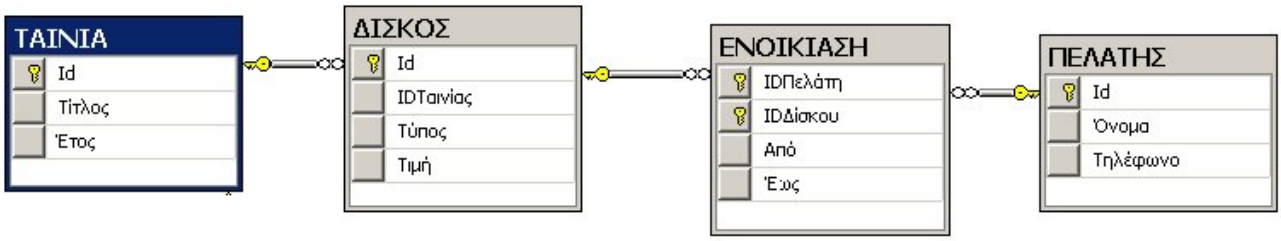


Εικόνα 5.11

5.2.2.2. Δημιουργία φόρμας πελάτη-ενοικιαζόμενων δίσκων με τη χρήση ερωτήματος

(Αφορά πίνακες που σχετίζονταν στο E-R με σχέσεις πολλά προς πολλά)

Έστω λοιπόν ο πίνακας dbo_ΠΕΛΑΤΗΣ και οι πίνακες dbo_ΔΙΣΚΟΣ και dbo_ΤΑΙΝΙΑ. Οι συσχετίσεις μεταξύ των προαναφερθέντων πινάκων είναι οι παρακάτω:



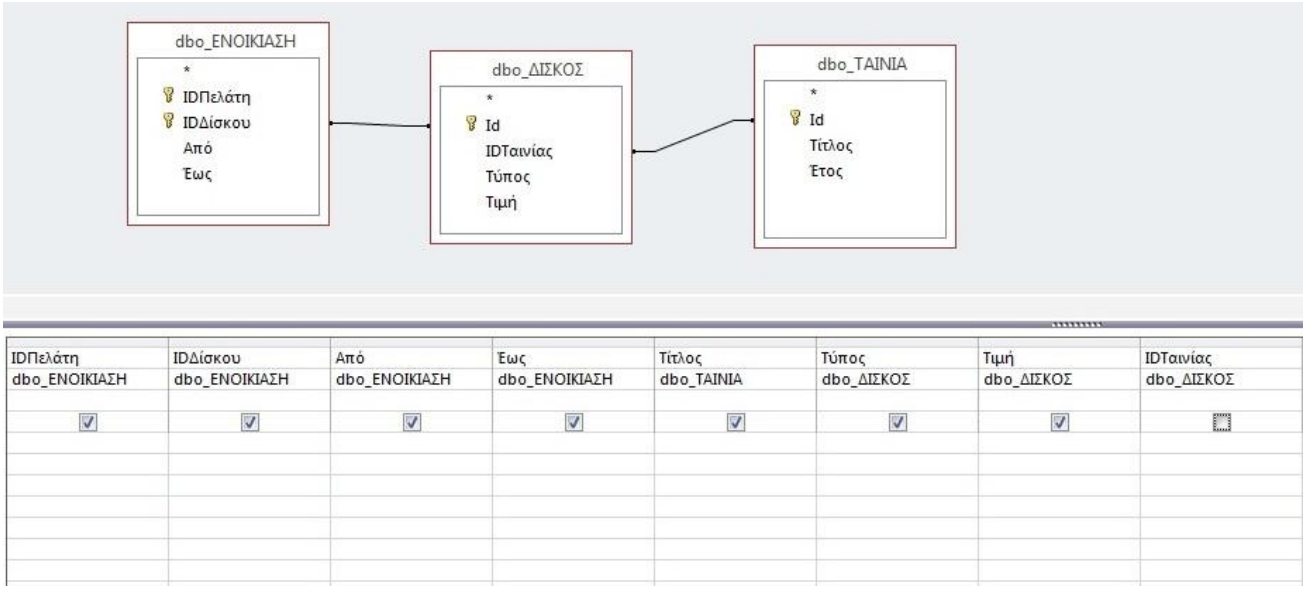
Εικόνα 5.12

Για να φτιάξουμε μια κύρια μαζί με μια δευτερεύουσα φόρμα, όπου η κύρια να παίρνει τιμές από τον πίνακα dbo_ΠΕΛΑΤΗΣ, ενώ στη δευτερεύουσα να καταχωρούμε τους δίσκους που ενοικίασε ένας πελάτης εργαζόμαστε ως εξής:

1. Χρησιμοποιούμε τον ενδιάμεσο πίνακα dbo_ΕΝΟΙΚΙΑΣΗ που συνδέει τον πίνακα dbo_ΔΙΣΚΟΣ με τον πίνακα dbo_ΠΕΛΑΤΗΣ. Ο πίνακας dbo_ΕΝΟΙΚΙΑΣΗ έχει ως πεδία τα πρωτεύοντα κλειδιά των δύο πινάκων που συνδέει και όσα άλλα πεδία είναι απαραίτητα (π.χ. πεδίο Από, πεδίο Έως).

2. Δημιουργούμε ένα ερώτημα (CREATE-> Query design) που παίρνει τιμές από τον ενδιάμεσο πίνακα (dbo_ΕΝΟΙΚΙΑΣΗ), τον πίνακα (dbo_ΤΑΙΝΙΑ) και τον πίνακα (dbo_ΔΙΣΚΟΣ). Στο ερώτημα συμπεριλαμβάνουμε
   (i) όλα τα πεδία του ενδιάμεσου πίνακα dbo_ΕΝΟΙΚΙΑΣΗ,
   (ii) από τον πίνακα dbo_ΔΙΣΚΟΣ εκείνα τα πεδία που θέλουμε να εμφανίζονται στην δευτερεύουσα φόρμα (εκτός από το πεδίο κλειδί) και
   (iii) από τον πίνακα dbo_ΤΑΙΝΙΑ το πεδίο Τίτλος.

   Τέλος, ονομάζουμε το ερώτημα με το όνομα **ΕρΕνοικίαση_Δίσκοι**.



Εικόνα 5.13

Παρακάτω βλέπουμε τα αποτελέσματα όταν τρέχουμε το ερώτημα μας:

ΕρΕνοικίαση_Κασέτες

ΙΔΠελάτη	ΙΔΔίσκου	Από	Έως	Τίτλος	Τύπος	Τιμή
1	1	10/7/2006	10/9/2006	Rear Window	BLU-RAY	2
1	2	20/9/2006	20/11/2006	Rear Window	DVD	3
2	1	10/9/2006		Rear Window	BLU-RAY	2

Εικόνα 5.14

Στόχος μας είναι να δημιουργήσουμε μια κύρια φόρμα πελάτη μαζί με μια δευτερεύουσα φόρμα, για να εισάγουμε τις ταινίες που αυτός ενοικιάζει κάθε φορά. Συγκεκριμένα, δημιουργούμε μια κύρια μαζί με μια δευτερεύουσα φόρμα, όπου η κύρια φόρμα παίρνει τιμές από τον πίνακα (dbo_ΠΕΛΑΤΗΣ), ενώ η δευτερεύουσα φόρμα παίρνει τιμές από το ερώτημα ΕρΕνοικίαση_Δίσκοι. Η κύρια μαζί με τη δευτερεύουσα φόρμα εμφανίζονται στην Εικόνα 5.15.

Id 1
Όνομα Perkins
Τηλέφωνο 246801

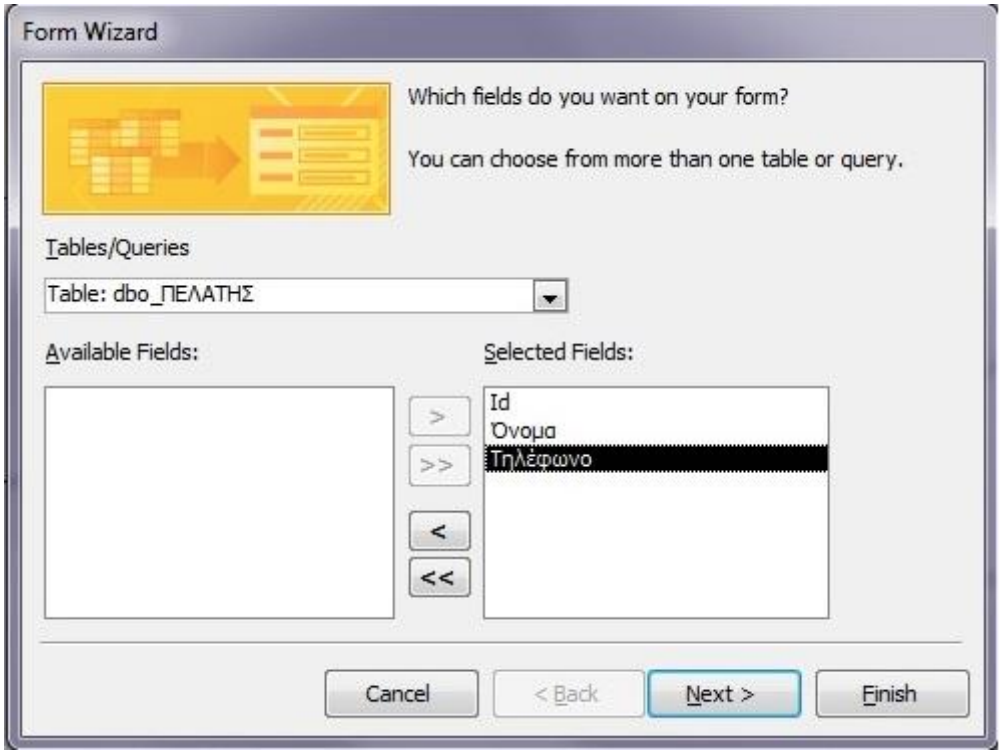
φρμ_ΕρΕνοικίαση _Δευτερ

ΙΔΔίσκου	Από	Έως	Τίτλος
1	10/7/2006	10/9/2006	Rear Window
2	20/9/2006	20/11/2006	Rear Window

Record: 1 of 2 No Filter Search

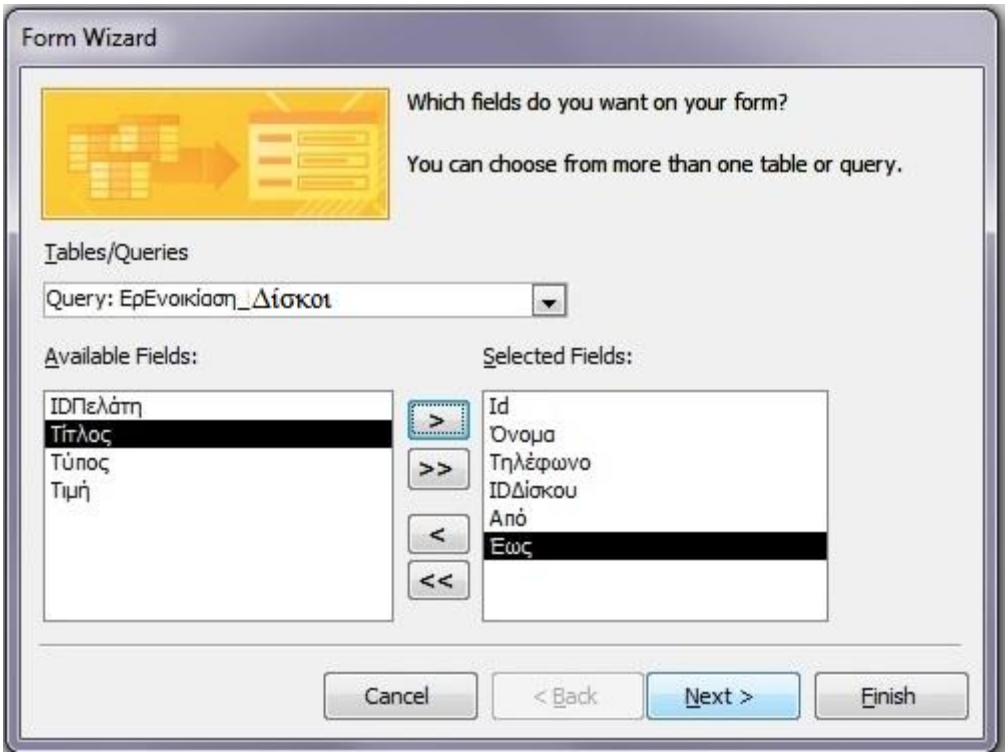
Εικόνα 5.15

Επιλέγουμε από την καρτέλα CREATE -> Form Wizard. Στη συνέχεια, επιλέγουμε τον πίνακα dbo_ΠΕΛΑΤΗΣ και παίρνουμε όλα τα πεδία του, όπως φαίνεται στην Εικόνα 5.16.



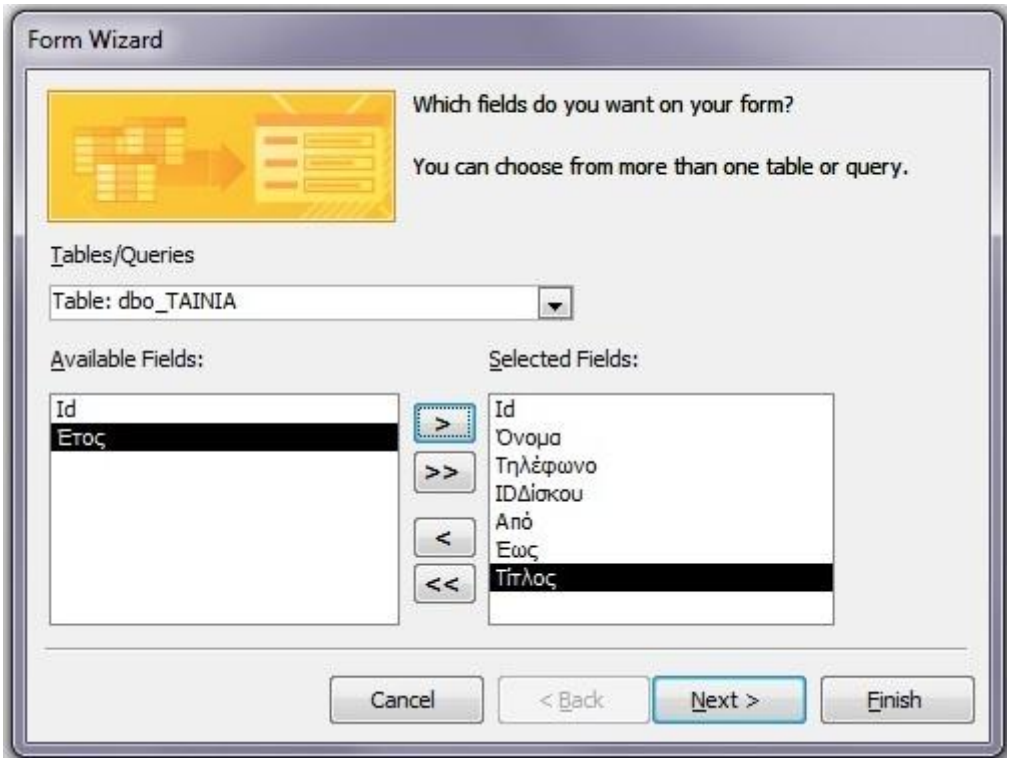
Εικόνα 5.16

Στη συνέχεια, χωρίς να πατήσουμε ακόμη Next, επιλέγουμε το ερώτημα ΕρΕνοικίαση_Δίσκοι και παίρνουμε τα παρακάτω πεδία: ΙΔΔίσκου, Από, Έως. Η Εικόνα 5.17 αποτυπώνει τη φάση στην οποία βρισκόμαστε.



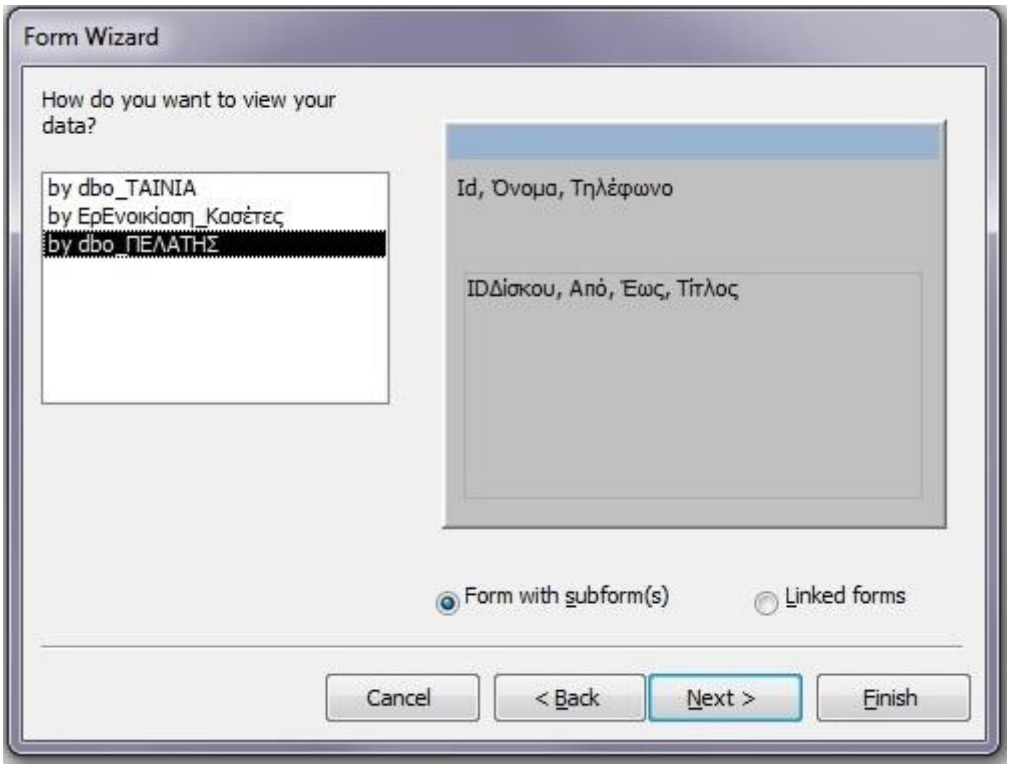
Εικόνα 5.17

Τέλος, στο ίδιο παράθυρο επιλέγουμε τον πίνακα dbo_TAINIA και παίρνουμε το πεδίο Τίτλος, όπως φαίνεται στην Εικόνα 5.18.



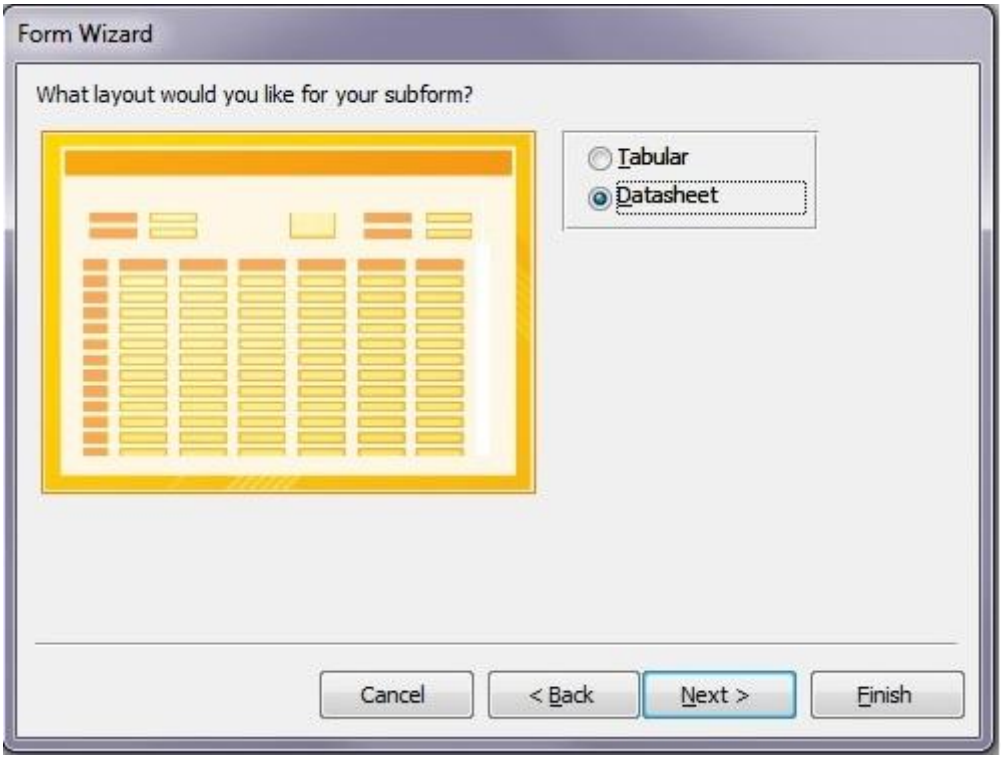
Εικόνα 5.18

Πατάμε Next και εμφανίζεται η Εικόνα 5.19, στην οποία επιλέγουμε ως φαίνεται.



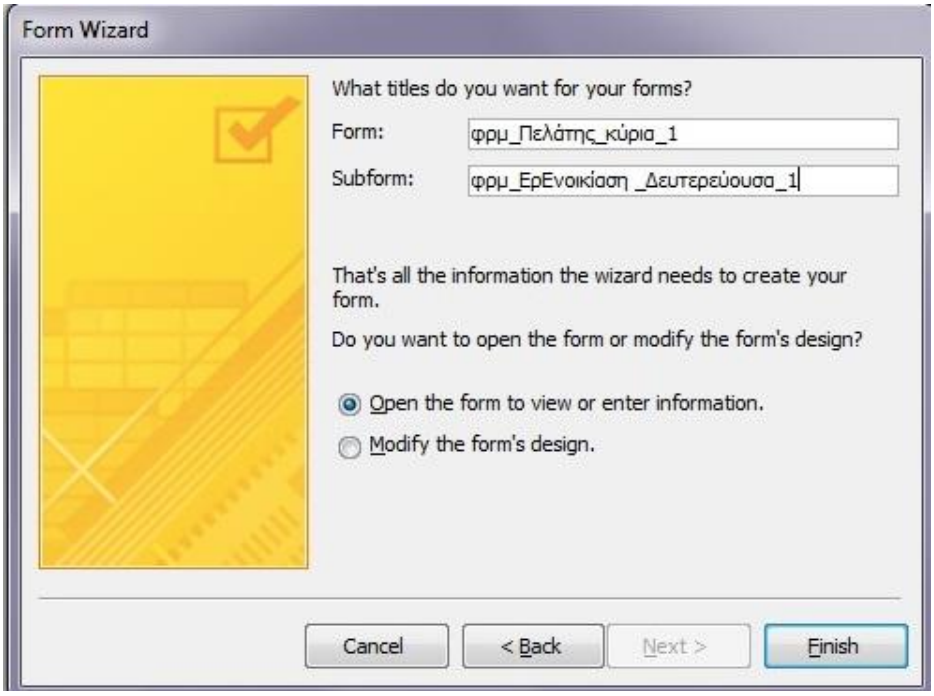
Εικόνα 5.19

Πατάμε Next και εμφανίζεται η Εικόνα 5.20, στην οποία επιλέγουμε ως φαίνεται.



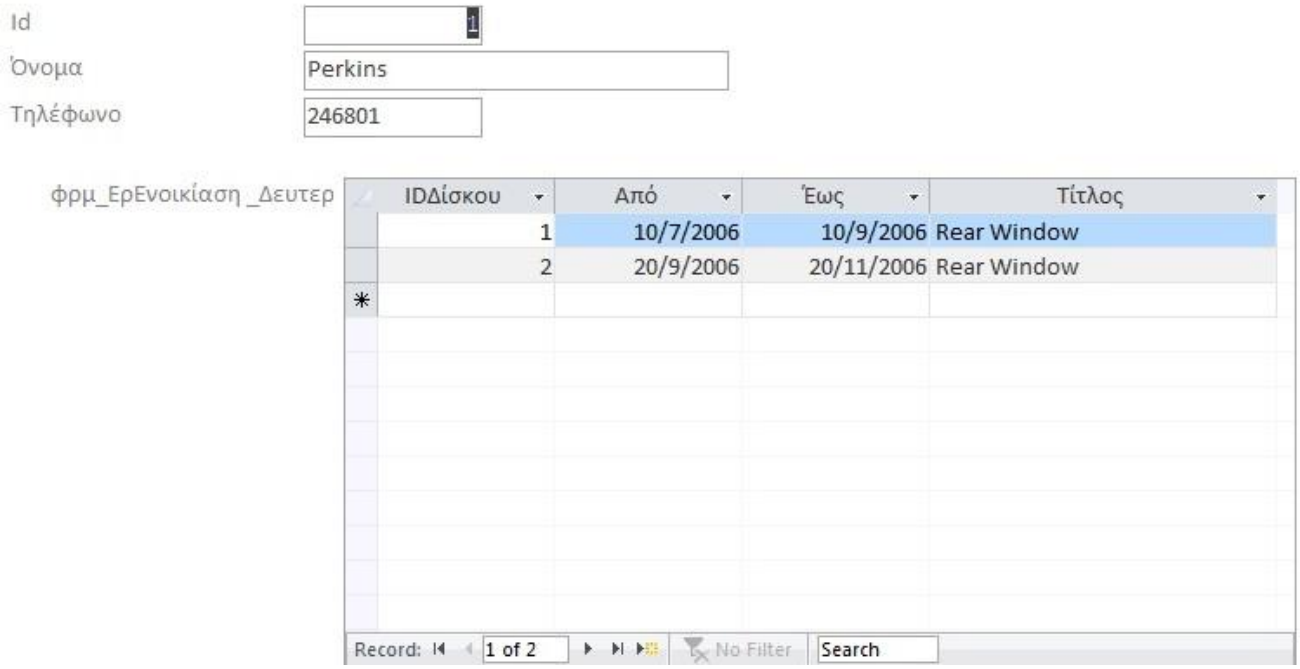
Εικόνα 5.20

Πατάμε Next και εμφανίζεται η Εικόνα 5.21. Επειδή στο τέλος θα έχουμε πολλές κύριες – δευτερεύουσες φόρμες και δεν θα ξέρουμε ποια αντιστοιχεί σε ποια, στο σημείο αυτό μετονομάζουμε την κύρια σε φρμ_Πελάτης_κύρια_1. Αντίστοιχα μετονομάζουμε την δευτερεύουσα φόρμα σε φρμ_ΕρΕνοικίαση _Δευτερεύουσα_1. Τέλος, πατάμε Finish.



Εικόνα 5.21

Η τελική μορφή της σύνθετης φόρμας που δημιουργήσαμε εμφανίζεται στην Εικόνα 5.22. Όπως μπορούμε να παρατηρήσουμε, η σύνθετη φόρμα επιτρέπει την ταυτόχρονη εμφάνιση τόσο των στοιχείων του πελάτη (στη συγκεκριμένη Εικόνα εμφανίζεται ο κ. Perkins) όσο και των ενοικιάσεων DVD που έχει κάνει.

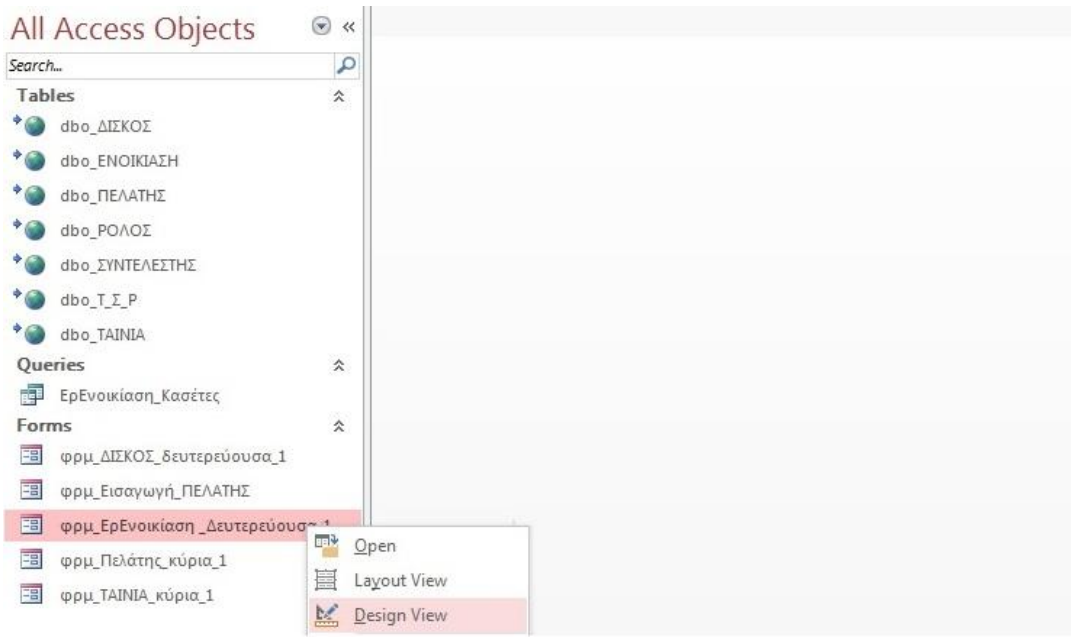


Εικόνα 5.22

5.3. Δημιουργία λίστας αναζήτησης σε φόρμα

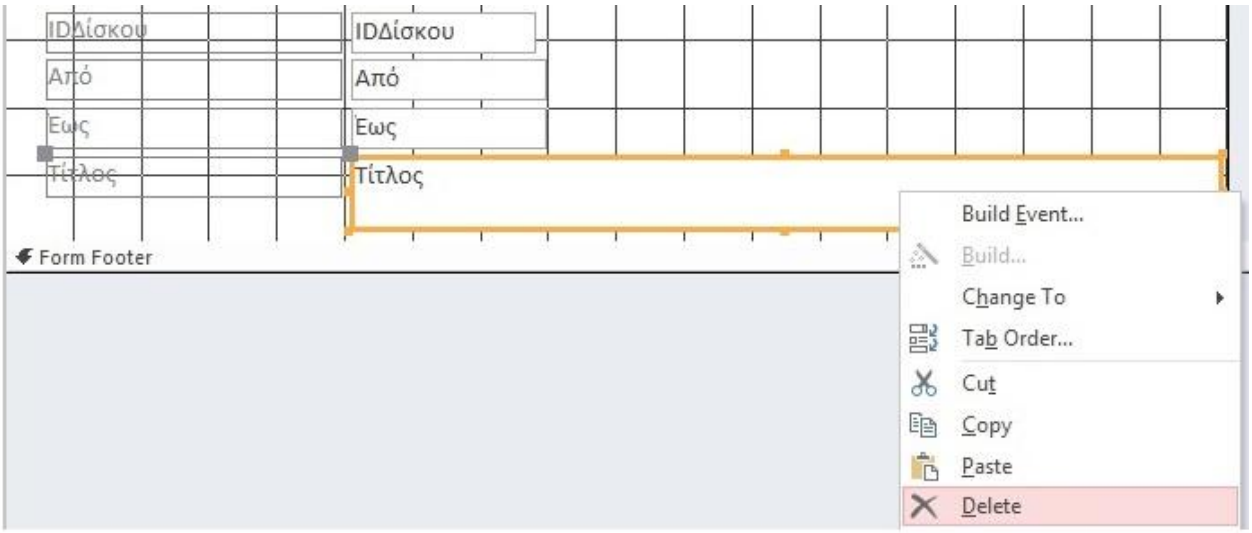
Σ' αυτήν την ενότητα θα παρουσιάσουμε τα απαιτούμενα βήματα, προκειμένου να μπορεί κάποιος να κάνει αναζήτηση στις εγγραφές ενός πεδίου σε μια δευτερεύουσα φόρμα. Για παράδειγμα, θα δείξουμε πώς μπορούμε να βάλουμε λίστα αναζήτησης στο πεδίο Τίτλος του πίνακα ΤΑΙΝΙΑ, προκειμένου να βρίσκουμε τις ταινίες βάσει του ονόματός τους και όχι τού ID τους.

Βήμα 1: Κάνουμε δεξί click στην φόρμα με το όνομα φρμ_ΕρΕνοικίαση _Δευτερεύουσα_1. Στη συνέχεια, επιλέγουμε Design View, όπως φαίνεται στην Εικόνα 5.23.



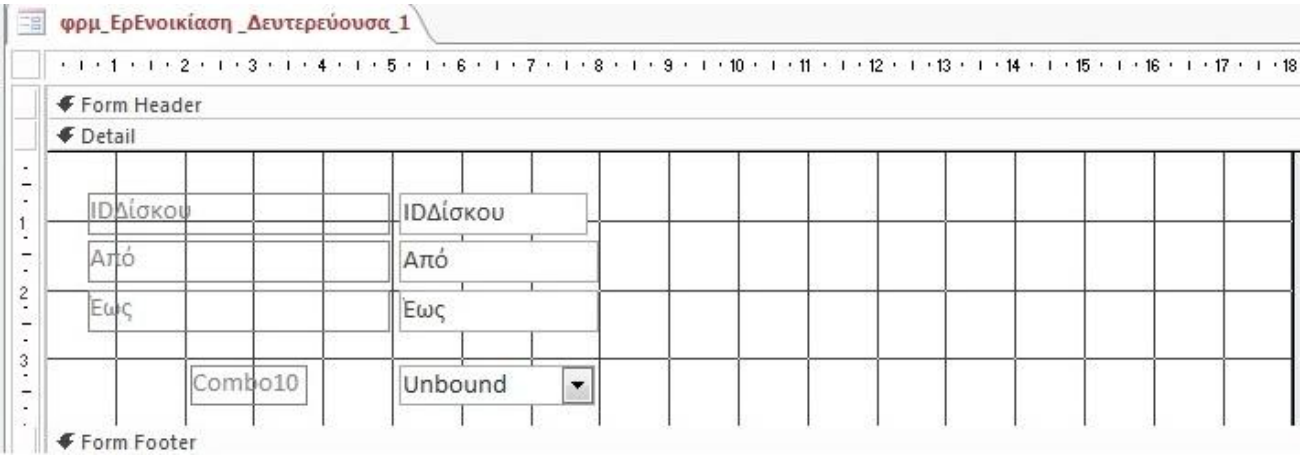
Εικόνα 5.23

Βήμα 2: Κάνουμε δεξί click στο πεδίο Τίτλος και το διαγράφουμε, όπως φαίνεται στην Εικόνα 5.24.



Εικόνα 5.24

Βήμα 3: Αφού διαγραφεί ο Τίτλος, προσθέτουμε το Combo Box (λίστα αναζήτησης) από την καρτέλα Design. Τέλος, το σύρουμε και το αφήνουμε στη θέση του διαγραφέντος, όπως φαίνεται στην Εικόνα 5.25.



Εικόνα 5.25

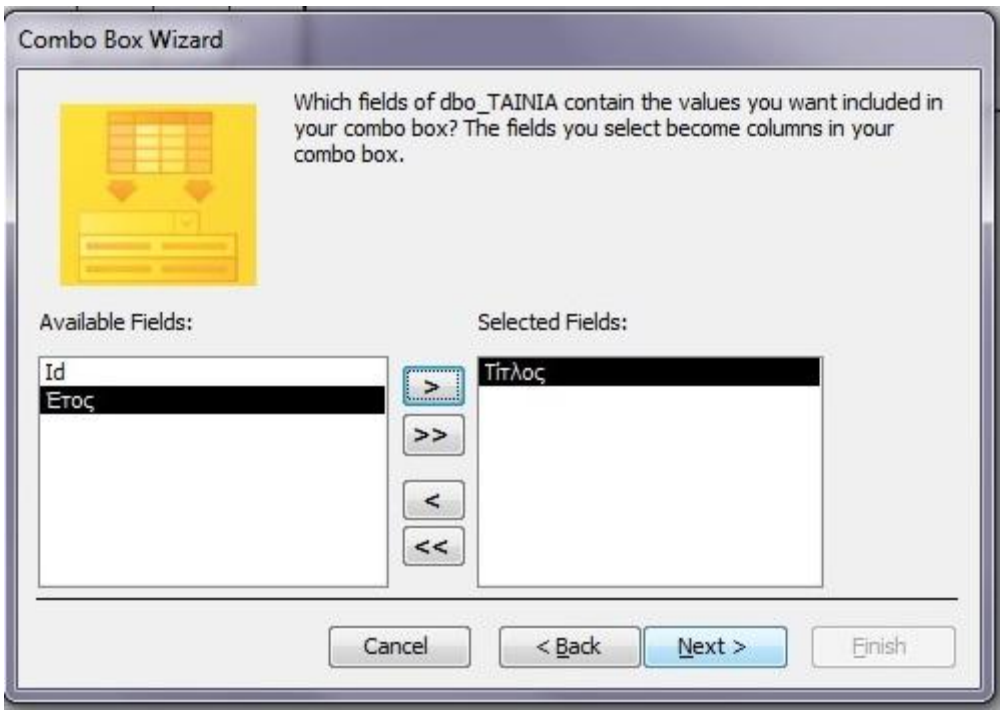
Βήμα 4: Πατάμε Next και εμφανίζεται η Εικόνα 5.26.. Κάνουμε την επιλογή που φαίνεται και ζητάμε Next.

Εικόνα 5.26

Βήμα 5: Εμφανίζεται ένα νέο παράθυρο, στο οποίο αφενός επιλέγουμε τον πίνακα ΤΑΙΝΙΑ και αφετέρου τσεκάρουμε το Tables, όπως φαίνεται στην Εικόνα 5.27. Επιλέγουμε Next.

Εικόνα 5.27

Βήμα 6: Από τα διαθέσιμα πεδία επιλεγούμε Τίτλος και το μεταφέρουμε στα επιλεγμένα πεδία με το σύμβολο > , όπως φαίνεται στην Εικόνα 5.28. Ζητάμε Next..



Εικόνα 5.28

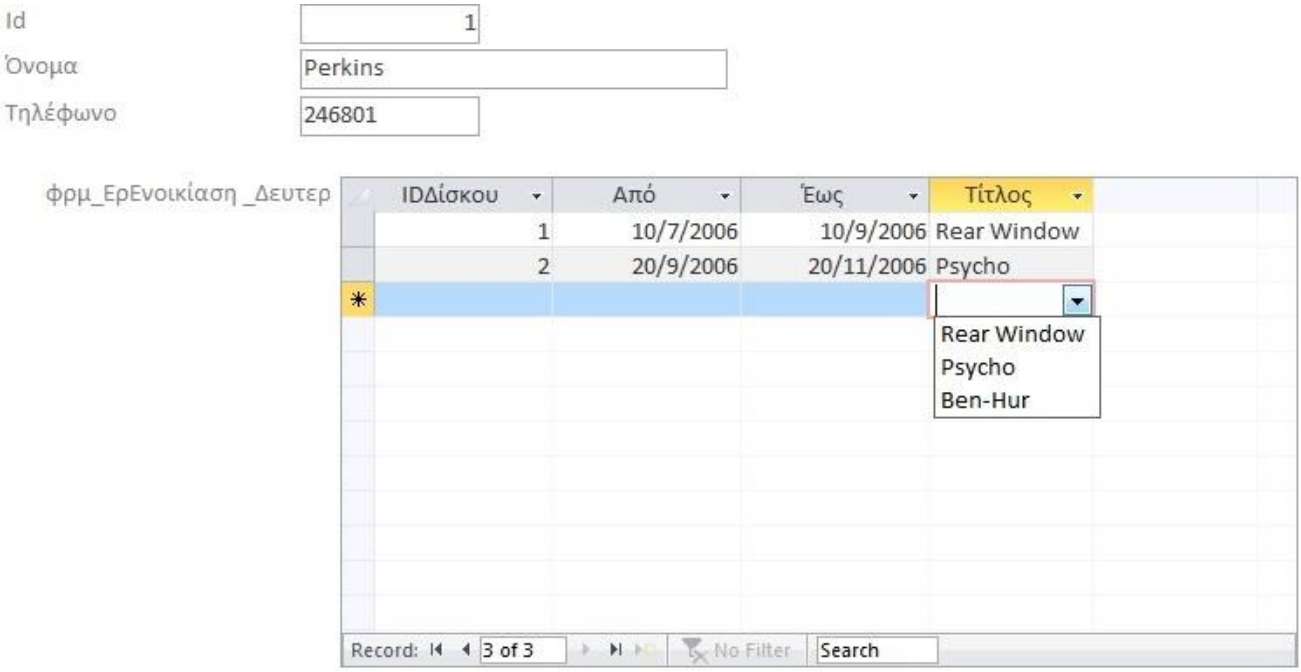
Βήμα 7: Πατάμε Next μέχρι να φτάσουμε στο παράθυρο της Εικόνας 5.29. Σ' αυτό το σημείο ακολουθούμε τις επιλογές της Εικόνας 5.29.

Εικόνα 5.29

Βήμα 8: Πατάμε Next και, στο παράθυρο της Εικόνας 5.30, ζητάμε Finish

Εικόνα 5.30

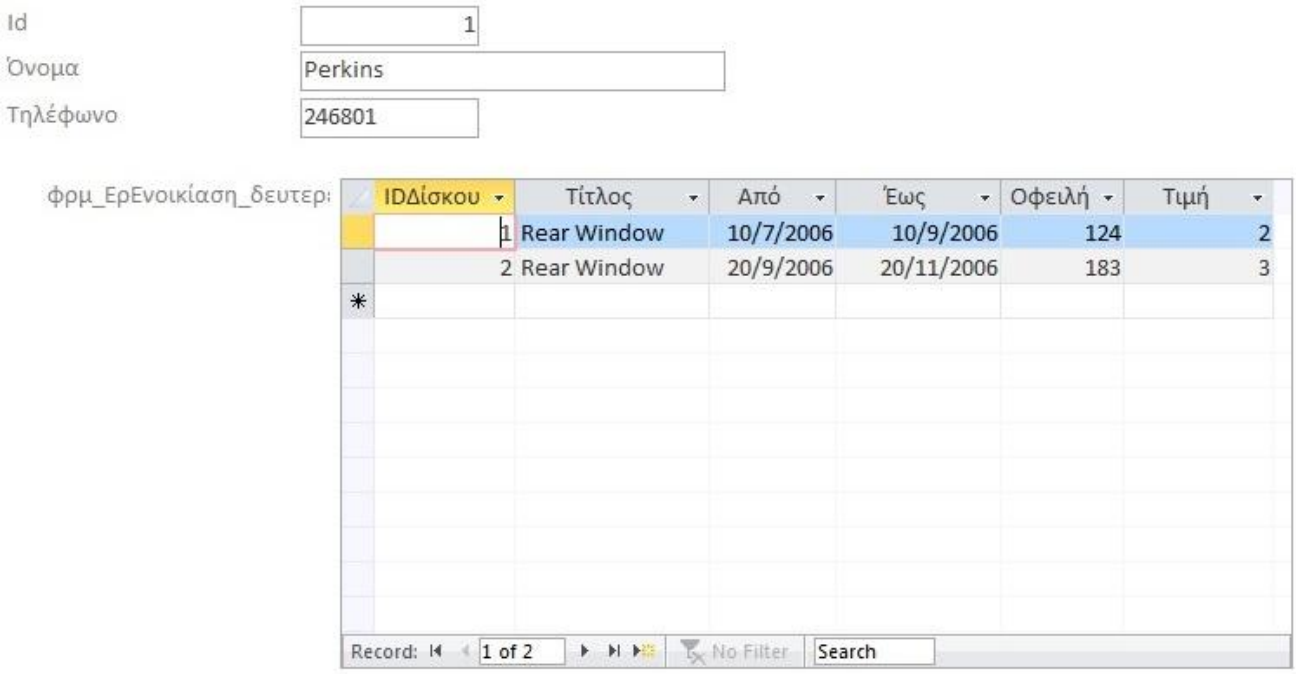
Βήμα 9: Μεταβαίνουμε στην κύρια φόρμα και εμφανίζεται η Εικόνα 5.31, στην οποία πλέον μπορούμε να αναζητήσουμε μια ταινία με το όνομα της με την χρήση της λίστας αναζήτησης και χωρίς να πρέπει να γνωρίζουμε εξ αρχής τον κωδικό της.



Εικόνα 5.31

5.4. Δημιουργία υπολογιζόμενου πεδίου σε δευτερεύουσα φόρμα.

Στόχος της παρούσας ενότητας είναι η δημιουργία ενός υπολογιζόμενου πεδίου με τίτλο «Οφειλή», στο οποίο θα υπολογίζεται σε ευρώ το ποσό με το οποίο θα χρεώνεται ο πελάτης, όταν επιστρέφει έναν δίσκο. Το ποσό θα εμφανίζεται στη στήλη Οφειλή, όπως φαίνεται στην Εικόνα 5.32.



Εικόνα 5.32

Το υπολογιζόμενο πεδίο Οφειλή προκύπτει από τον παρακάτω τύπο:

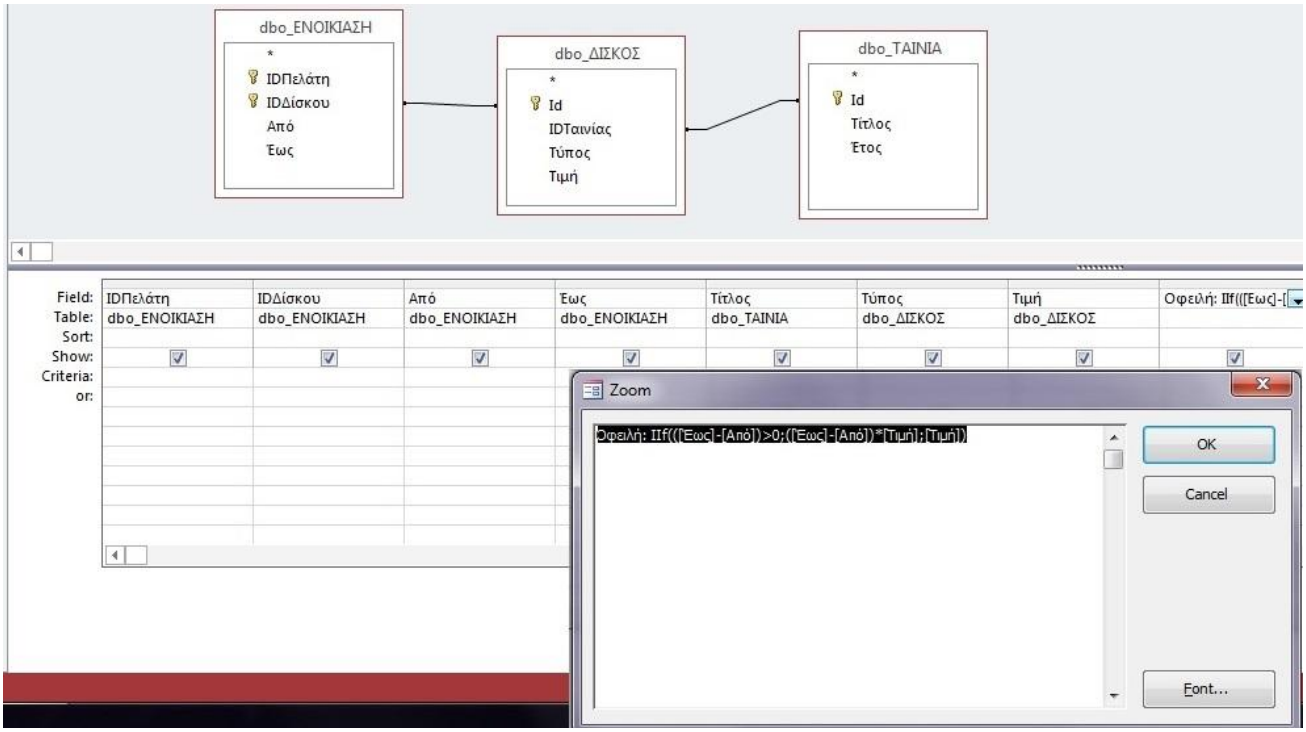
Οφειλή: ([Έως]-[Από])*[Τιμή]

Σύμφωνα με τον παραπάνω τύπο, προκειμένου να βρούμε την χρέωση ενός πελάτη για ένα δίσκο που ενοικίασε πρέπει να βρούμε τις ημέρες κράτησης αφαιρώντας τα πεδία Έως και Από και να πολλαπλασιάσουμε επί το πεδίο Τιμή που αφορά την τιμή ενοικίασης του δίσκου.

Για την περίπτωση που ένας πελάτης επιστρέψει τον ψηφιακό δίσκο αυθημερόν θα πρέπει να διαμορφώσουμε τον παραπάνω τύπο ως εξής:

Οφειλή: IIf(([Έως]-[Από])>0;([Έως]-[Από])*[Τιμή];[Τιμή])

Ο παραπάνω τύπος πρέπει να προστεθεί ως επιπλέον πεδίο «Οφειλή» στο ερώτημα ΕρΕνοικίαση_Δίσκοι, όπως φαίνεται στην Εικόνα 5.33.



Εικόνα 5.33

5.5. Ασκήσεις

1. Τι είναι κύρια και τι δευτερεύουσα φόρμα; Δώστε ένα παράδειγμα για τη βάση δεδομένων DVDclub.

2. Δημιουργήστε μια κύρια/δευτερεύουσα φόρμα, όπου στην κύρια φόρμα θα φαίνεται ο δίσκος DVD και στη δευτερεύουσα θα μπορεί κάποιος να καταχωρεί τον πελάτη που ενοικιάζει ή επιστρέφει τον δίσκο DVD.

3. Δημιουργήστε μια λίστα αναζήτησης στην παραπάνω δευτερεύουσα φόρμα στο πεδίο όνομα πελάτη.

4. Δημιουργήστε μια κύρια/δευτερεύσουσα φόρμα, όπου στη κύρια φόρμα θα φαίνεται η Ταινία και στη δευτερεύουσα θα μπορεί κάποιος να καταχωρεί τους συντελεστές που συμμετέχουν σ' αυτήν.

5. Δημιουργήστε μια λίστα αναζήτησης στην παραπάνω δευτερεύουσα φόρμα στο πεδίο όνομα συντελεστή.

6. Στην κύρια/δευτερεύουσα φόρμα της Εικόνας 5.32 να δημιουργήσετε ένα ακόμη υπολογιζόμενο πεδίο στο οποίο θα εμφανίζεται η τελική οφειλή ενός πελάτη μαζί με τον φόρο ΦΠΑ που είναι ίσος με 23%. Το νέο υπολογιζόμενο πεδίο θα ονομάζεται «Οφειλή με ΦΠΑ μαζί».

Κεφάλαιο 6. Προετοιμασία Δεδομένων ενόψει της Διαδικασίας Εξόρυξης

Σύνοψη

Το έκτο κεφάλαιο είναι εισαγωγικό. Αρχικά θα δημιουργήσουμε μια βάση δεδομένων με τη χρήση του SQL Server Management Studio. Στη συνέχεια, θα εισάγουμε αυτήν τη βάση σ' ένα νέο project που θα δημιουργήσουμε στο SQL Server Business Intelligence. Συγκεκριμένα, θα εισάγουμε και θα προεπεξεργαστούμε τρεις διαφορετικές βάσεις δεδομένων (MovieClick, FoodMart, AdventureWorks) στο περιβάλλον του SQL Server, ώστε να είμαστε σε θέση, σε επόμενα κεφάλαια. να εφαρμόσουμε σε αυτές τεχνικές εξόρυξης δεδομένων.

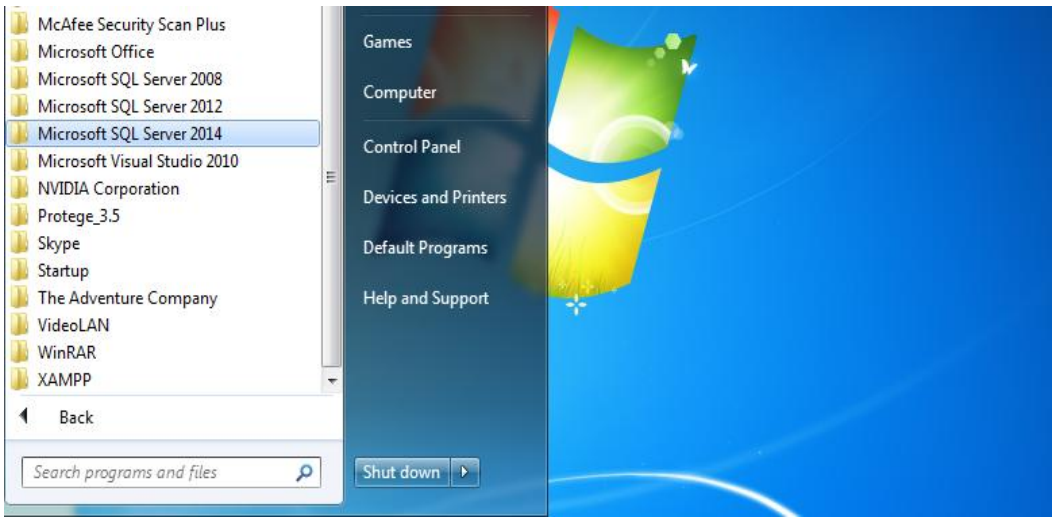
6.1. Εισαγωγή βάσης δεδομένων MovieClick

Σ' αυτήν την ενότητα επιθυμούμε, αρχικά, την εισαγωγή μιας βάσης δεδομένων που έχει δημιουργηθεί σε περιβάλλον Microsoft Access (*.mdb) σε ένα project του MS SQL Server 2014. Στη συνέχεια, θα επεξεργαστούμε αυτά τα δεδομένα. Πρέπει να τονίσουμε ότι η βάση δεδομένων MovieClick περιέχει στοιχεία καταγραφής με τις προτιμήσεις των χρηστών για ταινίες και, επομένως, θα χρησιμοποιηθεί στα Κεφάλαια 7, 8 και 9. Συνοπτικά τα βήματα που θα περιγραφούν στην ενότητα αυτή είναι τα παρακάτω:

- Δημιουργία μιας βάσης δεδομένων με τον MS SQL Server 2014.
- Εισαγωγή των δεδομένων της Microsoft Access βάσης (*.mdb) σ' αυτήν του SQL Server που μόλις δημιουργήσαμε.
- Δημιουργία ενός νέου project με τον SQL Server 2014 για επεξεργασία της βάσης που δημιουργήσαμε.
- Αποκατάσταση των συσχετίσεων (relationship) στην βάση του SQL Server.

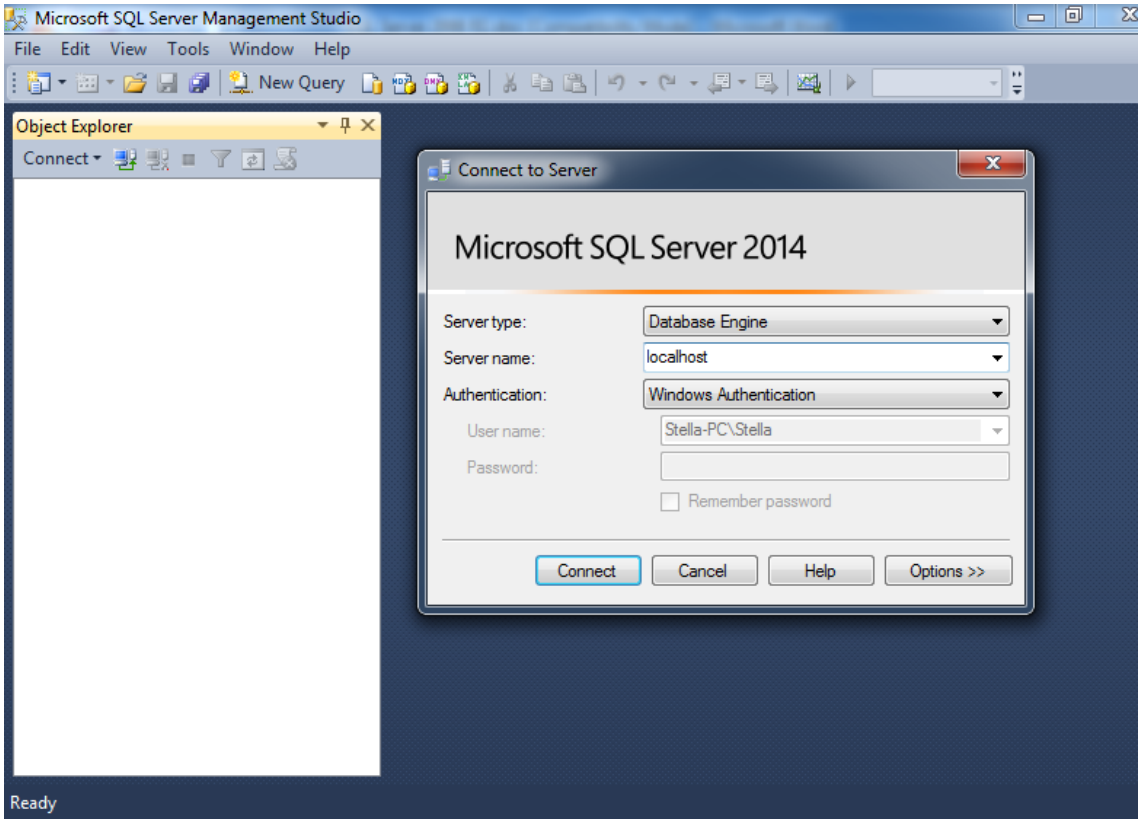
Αναλυτικά Βήματα

1. Όπως φαίνεται στην Εικόνα 6.1, για να δημιουργήσουμε μια βάση δεδομένων, θα χρησιμοποιήσουμε τον SQL Server Management Studio. Στο περιβάλλον των Windows ακολουθούμε την εξής διαδρομή: Έναρξη ► Όλα τα Προγράμματα ► Microsoft SQL Server 2014 ► SQL Server Management Studio.



Εικόνα 6.1

2. Όπως φαίνεται στην Εικόνα 6.2, επιλέγουμε τον διακομιστή με τον οποίο θα συνδεθούμε, ώστε να έχουμε πρόσβαση στις βάσεις δεδομένων του.

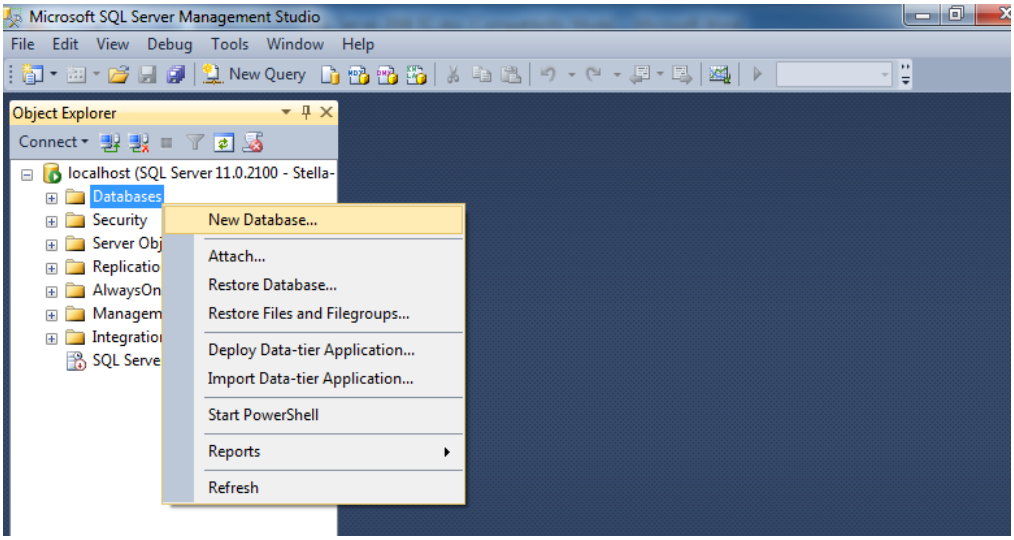


Εικόνα 6.2

Συγκεκριμένα συμπληρώνουμε τα πεδία ως εξής:

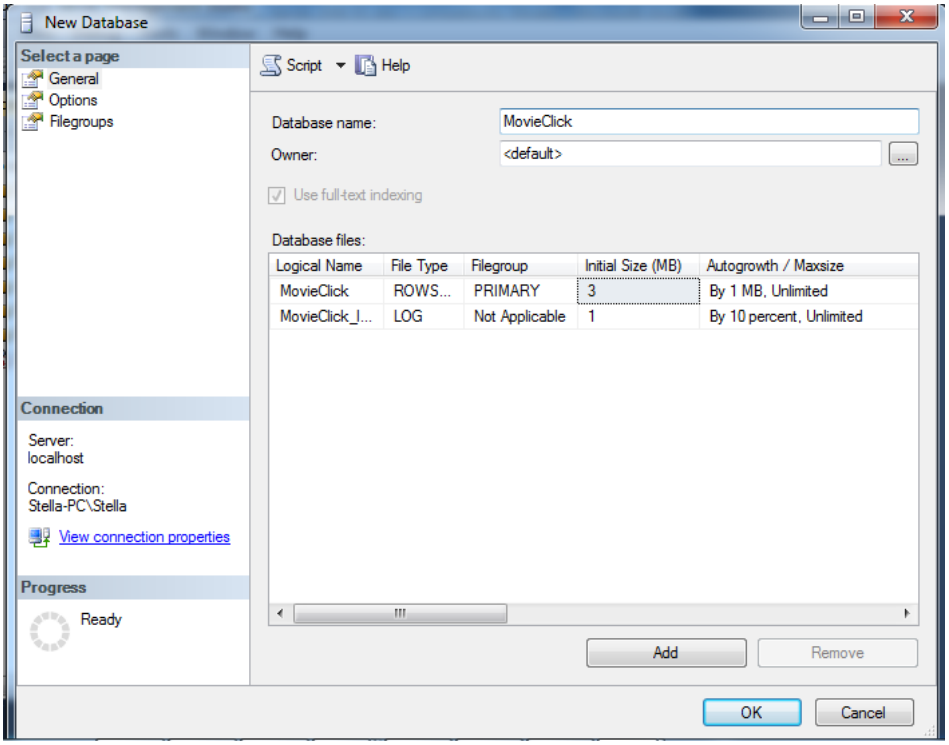
- Στο πεδίο **Server type** επιλέγουμε Database Engine.
- Στο πεδίο **Server name** συμπληρώνουμε το όνομα του υπολογιστή μας. Αν δεν το γνωρίζουμε, συμπληρώνουμε τον όρο «localhost» που αναφέρεται τοπικά στον υπολογιστή μας. Στη συγκεκριμένη περίπτωση το πεδίο έχει συμπληρωθεί με τον όρο «localhost».
- Στο πεδίο **Authentication** υπάρχουν δύο επιλογές: Windows Authentication και SQL Server Authentication (όπου εισάγουμε username και password για την σύνδεση με τον διακομιστή). Στη συγκεκριμένη περίπτωση έχουμε επιλέξει την πρώτη επιλογή.
- Επιλέγουμε **Connect** και συνδεόμαστε με τον διακομιστή που έχουμε επιλέξει.

3. Όπως εμφανίζεται στην Εικόνα 6.3, θα δημιουργήσουμε μια καινούρια βάση δεδομένων. Έτσι, επιλέγουμε την καρτέλα Object Explorer, κάνουμε δεξί κλικ στο πεδίο ¨Databases¨ και επιλέγουμε ¨New Database¨.



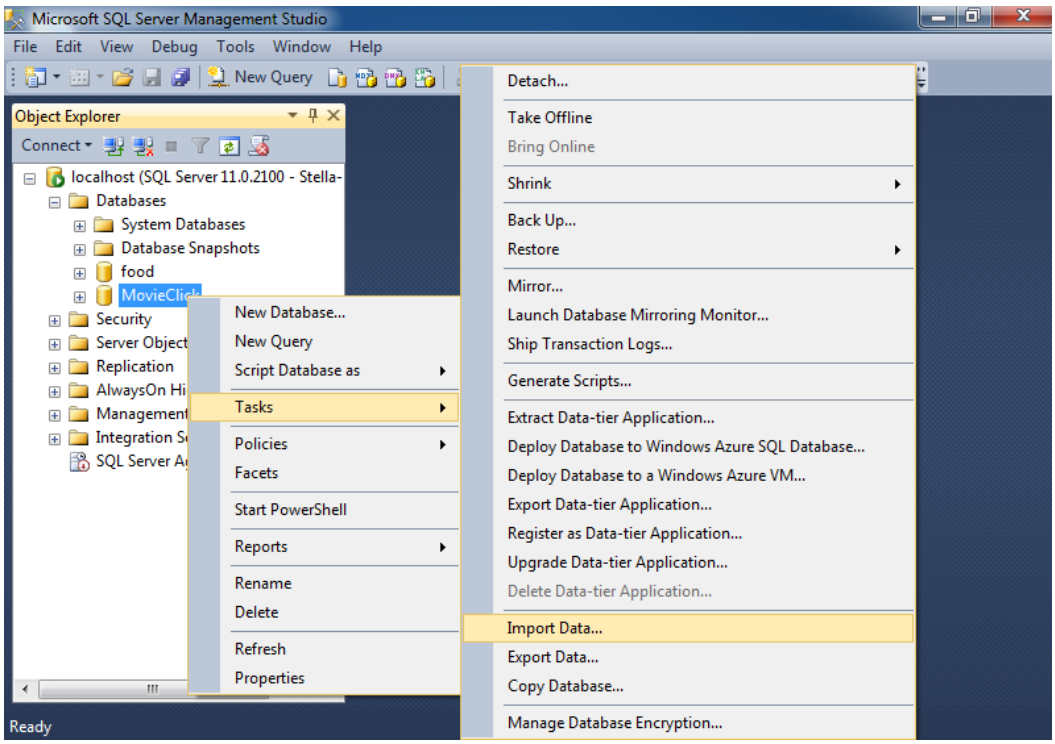
Εικόνα 6.3

4. Όπως φαίνεται στην Εικόνα 6.4, συμπληρώνουμε τα στοιχεία της βάσης που θα δημιουργήσουμε.
   - Στο πεδίο Database Name συμπληρώνουμε το όνομα της βάσης δεδομένων. Στη συγκεκριμένη περίπτωση έχει συμπληρωθεί το όνομα ¨MovieClick¨.
   - Στο πεδίο Owner δηλώνεται ο ιδιοκτήτης της βάσης δεδομένων. Στη συγκεκριμένη περίπτωση έχει αφεθεί η προεπιλεγμένη επιλογή <default>.
   - Επιλέγουμε OK, ώστε να δημιουργηθεί η βάση δεδομένων που έχουμε επιλέξει.



Εικόνα 6.4

5. Στη συνέχεια, πρέπει να εισάγουμε τα δεδομένα της βάσης MovieClick.mdb (αρχείο της Access) στην βάση που δημιουργήσαμε. Επιλέγουμε την καρτέλα Object Explorer ► Databases και κάνουμε δεξί κλικ στην βάση MovieClick. Επιλέγουμε Tasks και, στη συνέχεια, Import Data, όπως φαίνεται στην Εικόνα 6.5,

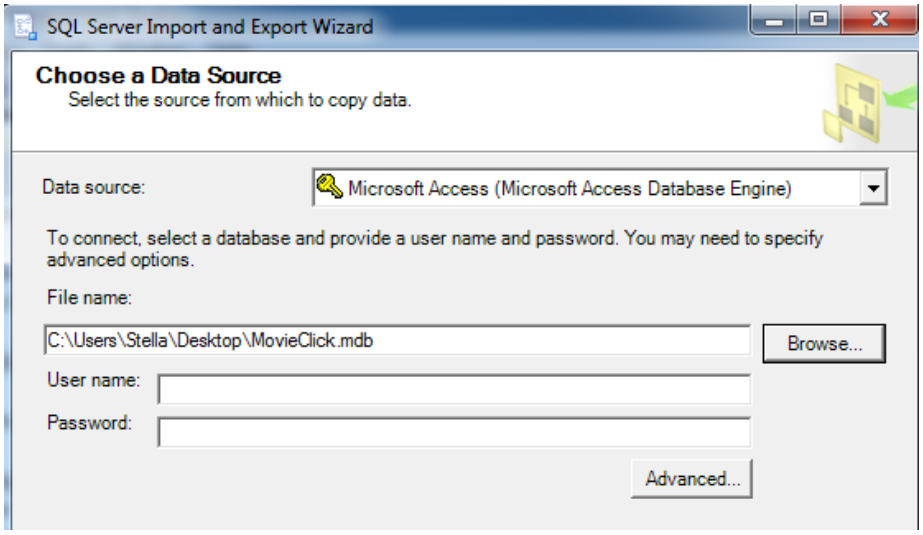


Εικόνα 6.5

6. Στην Εικόνα 6.6 εμφανίζεται ο οδηγός εισαγωγής/εξαγωγής δεδομένων «Import and Export Wizard» του SQL Server. Επιλέγουμε Next>, ώστε να προχωρήσουμε στο επόμενο βήμα.

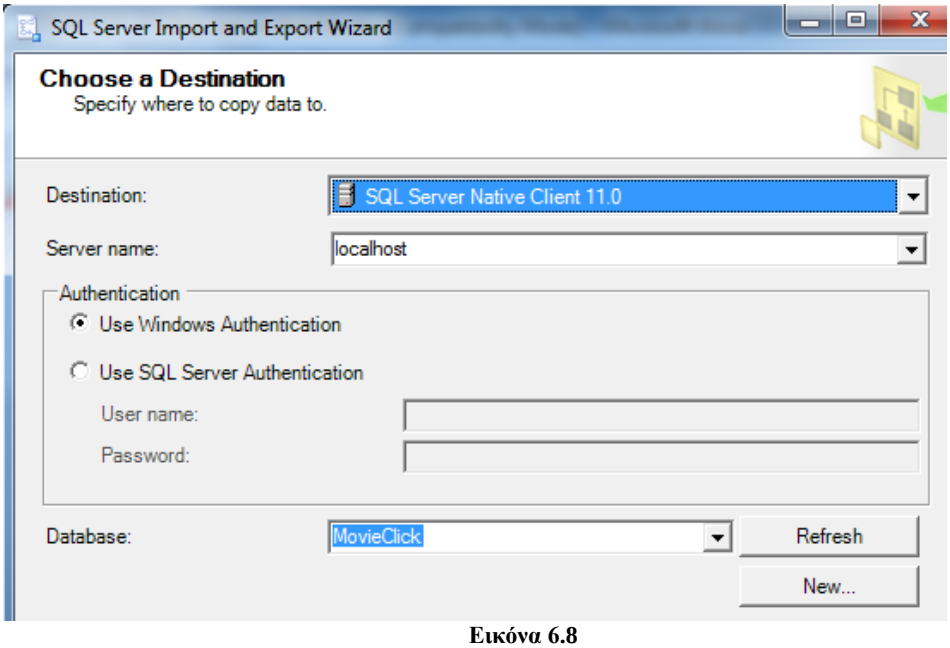
Εικόνα 6.6

7. Στο συγκεκριμένο βήμα, όπως φαίνεται στην Εικόνα 6.7, επιλέγουμε τα στοιχεία της βάσης MovieClick την οποία θα εισάγουμε στον SQL Server. Συγκεκριμένα, στο πεδίο Data source (όπου επιλέγουμε το είδος της βάσης που θέλουμε να εισάγουμε), εμείς επιλέγουμε Microsoft Access, καθώς η βάση μας έχει δημιουργηθεί στο περιβάλλον της Microsoft Access. Στο πεδίο File name συμπληρώνουμε τη διεύθυνση του αρχείου όπου βρίσκεται η βάση MovieClick στον υπολογιστή μας. Τέλος, επιλέγουμε Next>, ώστε να προχωρήσουμε στο επόμενο βήμα.



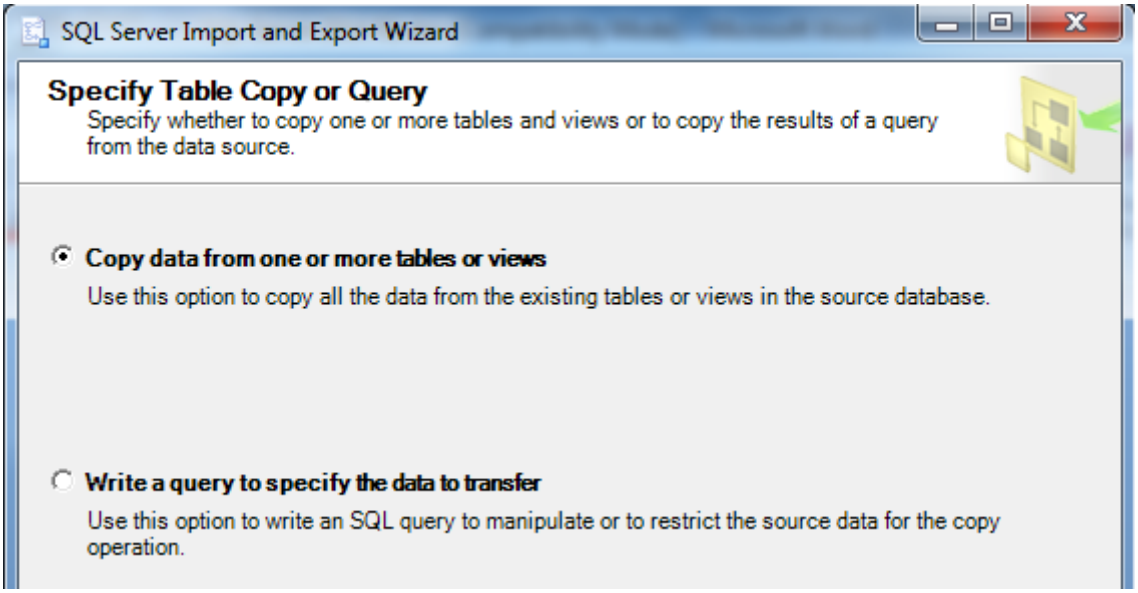
Εικόνα 6.7

8. Σ' αυτό το βήμα επιλέγουμε πού θα εισάγουμε τα δεδομένα της βάσης MovieClick.mdb. Όπως φαίνεται στην Εικόνα 6.8, στο πεδίο Destination επιλέγουμε SQL Native Client 10.0, στο πεδίο Server name επιλέγουμε τον υπολογιστή μας ή localhost, στο πεδίο Authentication επιλέγουμε Use Windows Authentication, στο πεδίο Database επιλέγουμε τη βάση MovieClick που δημιουργήσαμε. Τέλος, επιλέγουμε Next> για το επόμενο βήμα.



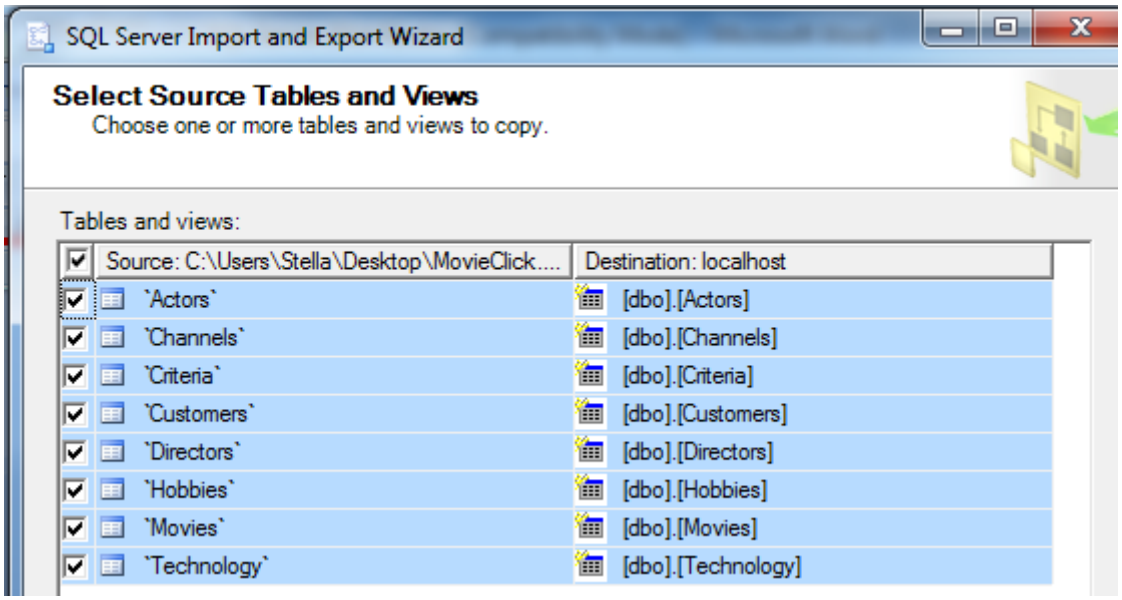
Εικόνα 6.8

9. Στο συγκεκριμένο βήμα καλούμαστε να επιλέξουμε αν θα εισάγουμε όλα τα δεδομένα από τους πίνακες της βάσης MovieClick ή αν θα γράψουμε ένα SQL Query, για να διαχειριστούμε ή να αποκλείσουμε κάποια δεδομένα. Στη συγκεκριμένη περίπτωση, όπως φαίνεται στην Εικόνα 6.9, επιλέγουμε Copy data from one or more tables or views.



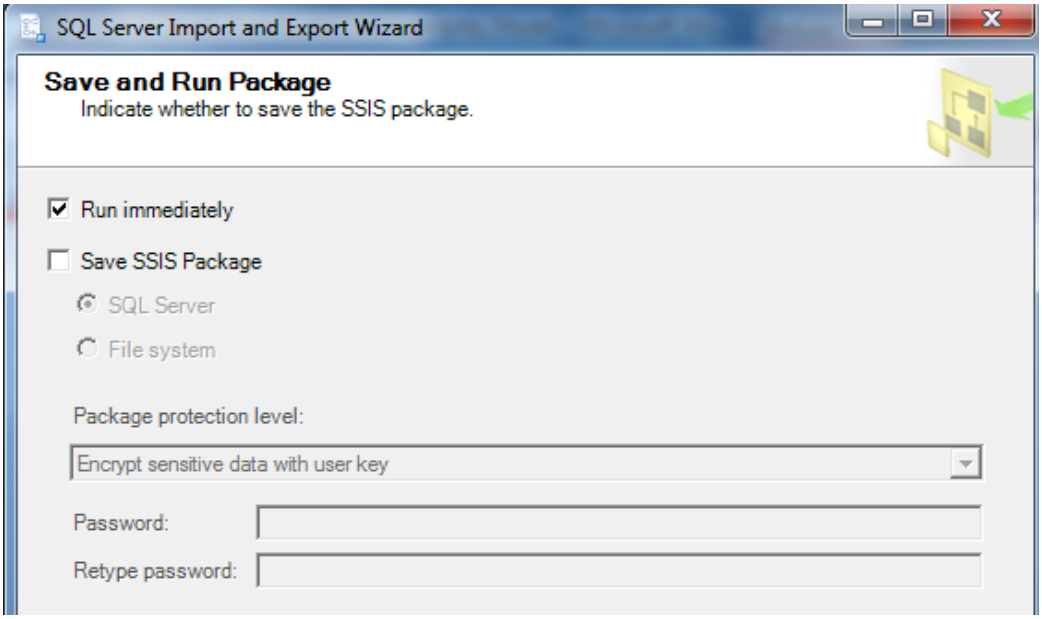
Εικόνα 6.9

10. Σ' αυτό το βήμα επιλέγουμε τους πίνακες της βάσης MovieClick.mdb που θα εισάγουμε στην βάση του SQL Server. Στη συγκεκριμένη περίπτωση, όπως φαίνεται στην Εικόνα 6.10, επιλέγουμε όλους τους πίνακες και, στη συνέχεια, επιλέγουμε Next>, ώστε να προχωρήσουμε στο επόμενο βήμα.



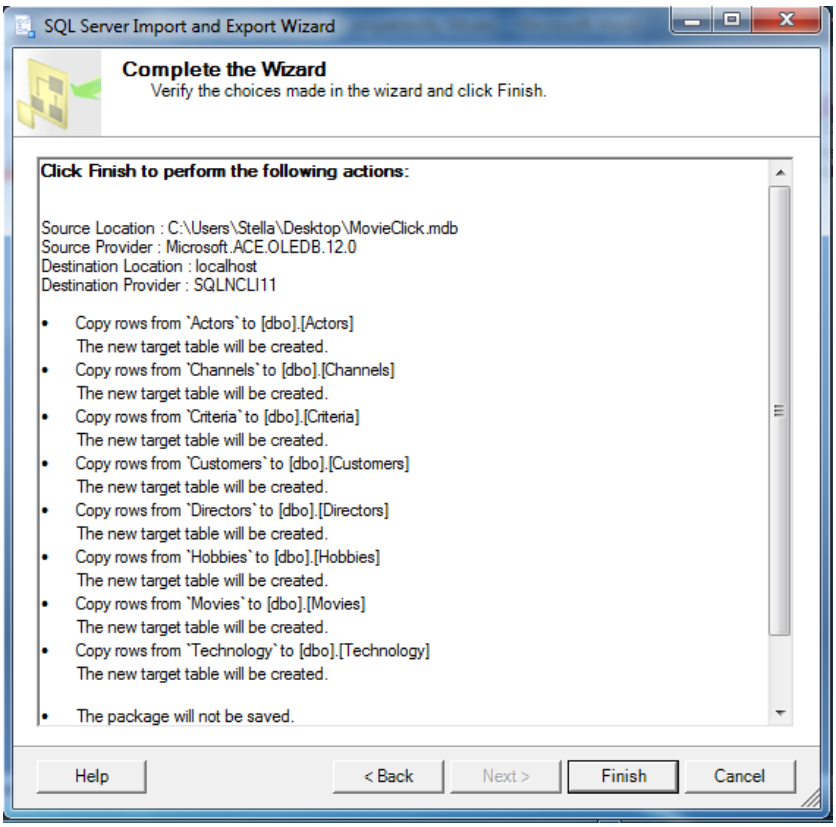
Εικόνα 6.10

11. Επιλέγουμε Run immediately, για να γίνει άμεση εκτέλεση, όπως φαίνεται στην Εικόνα 6.11. Στη συνέχεια, επιλέγουμε Next>, ώστε να προχωρήσουμε στο επόμενο βήμα.



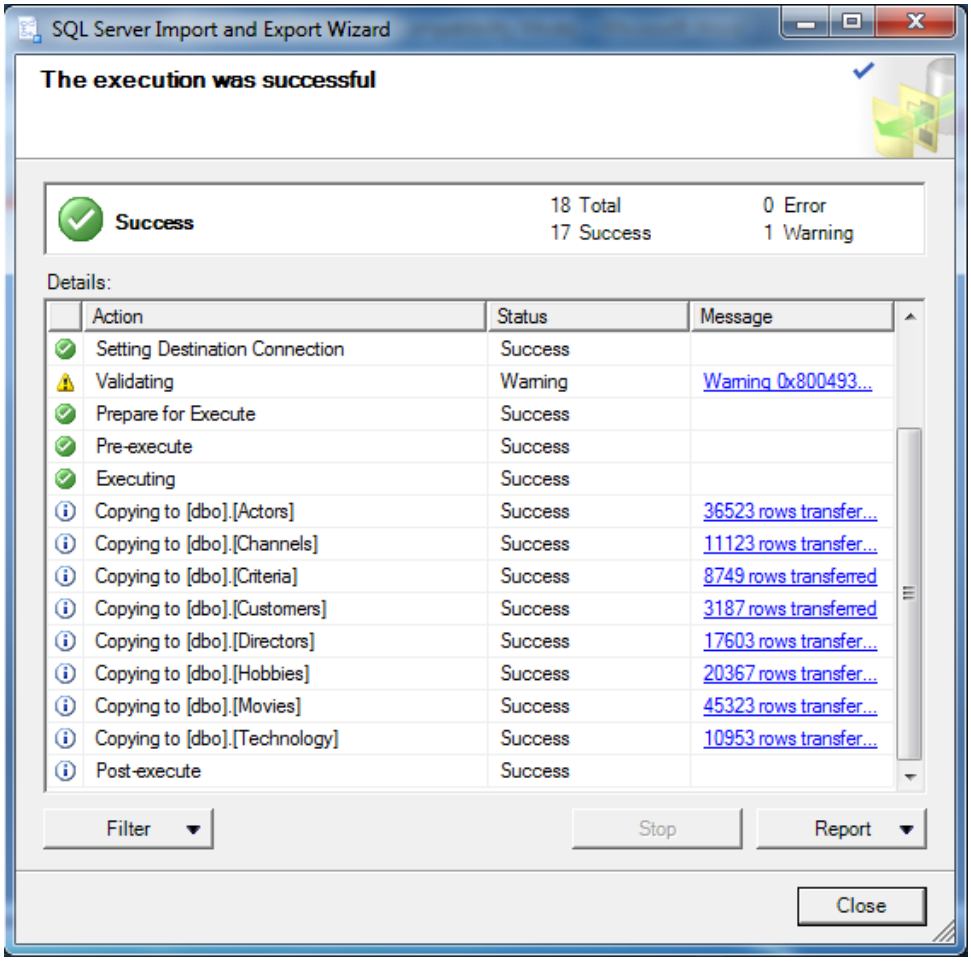
Εικόνα 6.11

12. Εμφανίζεται μια σύνοψη των επιλογών που έχουμε κάνει μέχρι τώρα, όπως φαίνεται στην Εικόνα 6.12. Επιλέγουμε Finish, για να ολοκληρωθεί η διαδικασία εισαγωγής.



Εικόνα 6.12

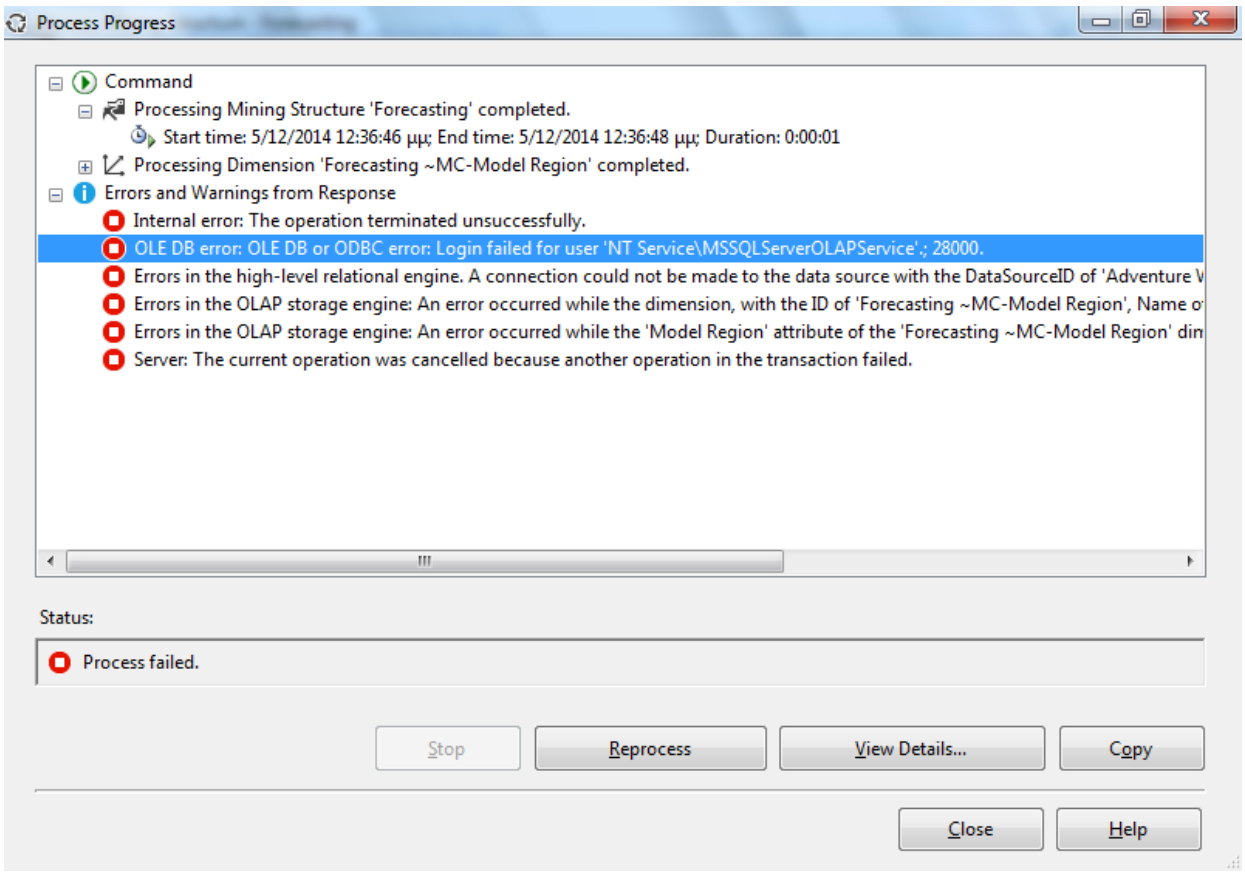
13. Όπως φαίνεται στην Εικόνα 6.13, εφόσον όλα τα βήματα έχουν ολοκληρωθεί με επιτυχία, δεν θα πρέπει να υπάρχουν καθόλου Errors ή Warnings. Η διαδικασία της εισαγωγής της βάσης MovieClick στον SQL Server έχει πλέον ολοκληρωθεί. Κατόπιν επιλέγουμε Close για να αφήσουμε τον οδηγό.



Εικόνα 6.13

14. Σ' αυτό το σημείο πρέπει να τονίσουμε ότι η βάση δεδομένων MovieClick αποτελείται από 7 πίνακες, για τους οποίους αφενός μεν δεν έχουν οριστεί πρωτεύοντα κλειδιά και αφετέρου δεν έχουν οριστεί συσχετίσεις μεταξύ τους. Γι' αυτόν τον λόγο, στην Ενότητα 6.4. θα ορίσουμε μέσα στο περιβάλλον του Visual Studio τόσο πρωτεύοντα κλειδιά σε κάθε πίνακα όσο και συσχετίσεις μεταξύ των πινάκων, προκειμένου να μπορούμε να τρέξουμε αλγορίθμους εξόρυξης δεδομένων. Για τους περισσότερους αλγορίθμους εξόρυξης δεδομένων του περιβάλλοντος του Business Intelligence απαιτείται η ύπαρξη τόσο πρωτεύοντος κλειδιού σε κάθε πίνακα όσο και αναφορικής ακεραιότητας μεταξύ των πινάκων, προκειμένου να τρέξουν ομαλά και να δώσουν αποτελέσματα.

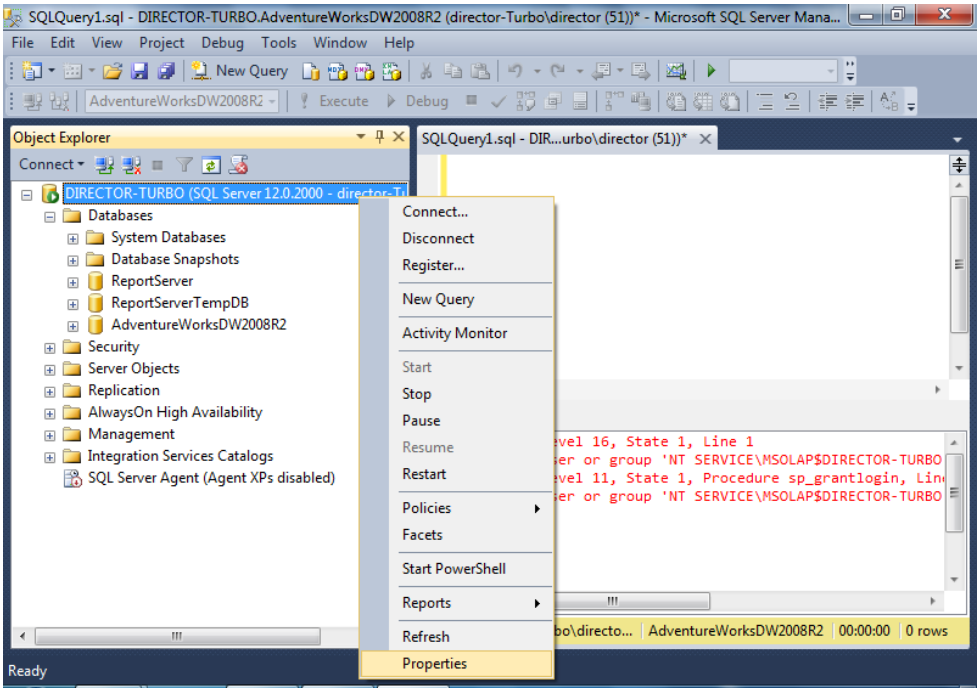
15. Στη συνέχεια, πρέπει να δημιουργηθεί ένας λογαριασμός χρήστη (**NT SERVICE\MSSQLServerOLAPService**) και να γίνει απόδοση δικαιωμάτων πρόσβασης σ' αυτόν για τη βάση δεδομένων MovieClick. Διαφορετικά, ο εξ ορισμού χρήστης **NT SERVICE\MSSQLServerOLAPService** δεν θα έχει τη δυνατότητα να τρέξει κάποιο αλγόριθμο εξόρυξης δεδομένων στο περιβάλλον του SQL Server Data Tools του Visual Studio γι' αυτήν τη βάση δεδομένων. Ένα τυπικό σφάλμα στην περίπτωση μη απόδοσης δικαιωμάτων πρόσβασης στον παραπάνω χρήστη εμφανίζεται στην Εικόνα 6.14.



Εικόνα 6.14

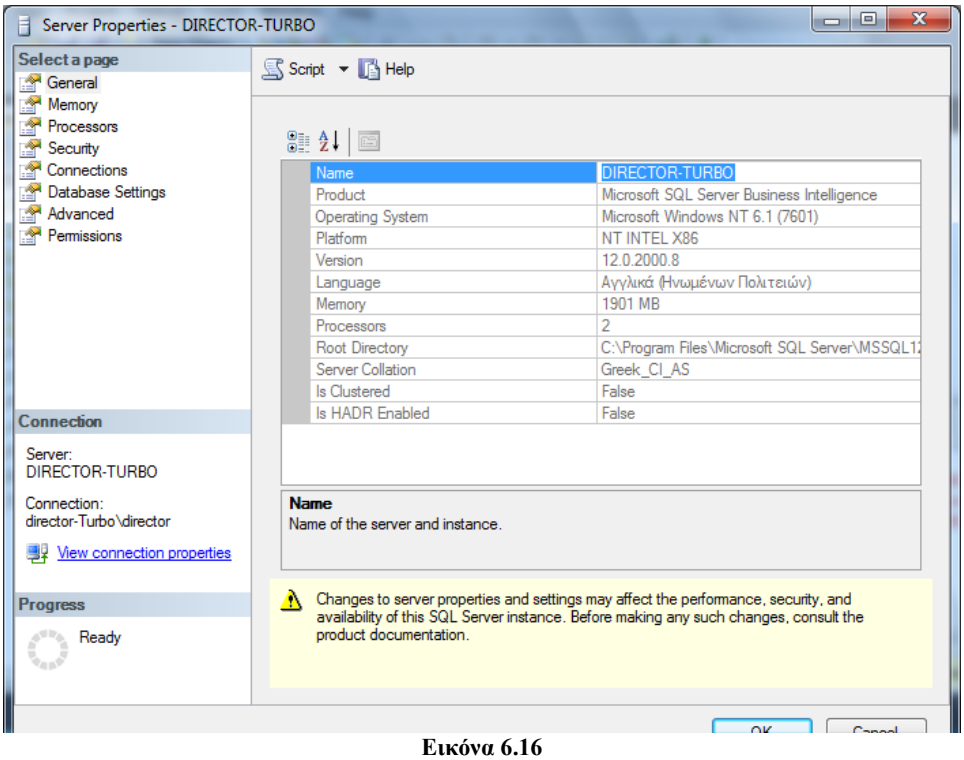
Συνεπώς, ακολουθούμε τα παρακάτω βήματα (Α έως Δ), προκειμένου ο χρήστης **NT SERVICE\MSSQLServerOLAPService** να έχει αργότερα τη δυνατότητα να τρέχει μοντέλα εξόρυξης δεδομένων στο Visual Studio. Τα βήματα περιγράφονται αναλυτικά παρακάτω:

Βήμα Α: Στον κεντρικό μας φάκελο, όπως φαίνεται στην Εικόνα 6.15, πατάμε δεξί κλικ στον Object Explorer και επιλέγουμε Properties. Σκοπός μας είναι δούμε το όνομα που έχει δοθεί στον server.



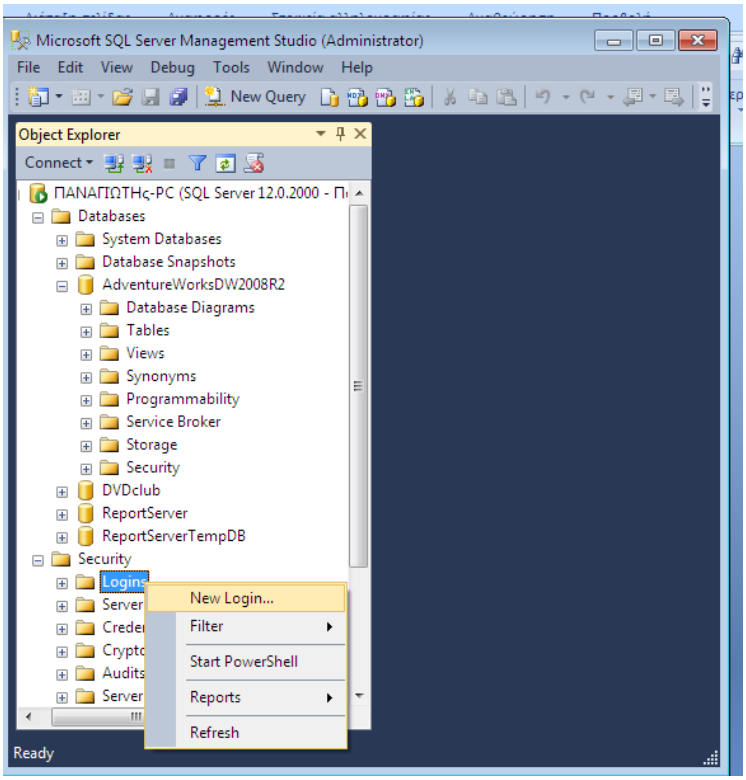
Εικόνα 6.15

Στην Εικόνα 6.16 βλέπουμε το όνομα του διακομιστή και το αντιγράφουμε.



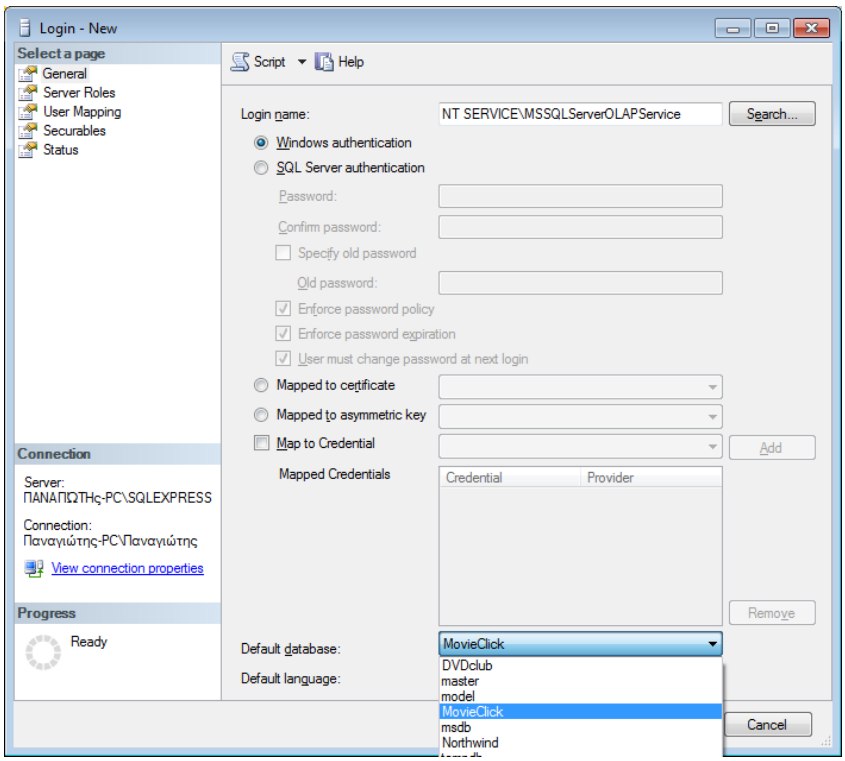
Εικόνα 6.16

Βήμα Β: Επιλέγουμε **Logins** από την επιλογή **Security** και, στη συνέχεια, με δεξί κλικ επιλέγουμε **New Login,** όπως φαίνεται στην Εικόνα 6.17.



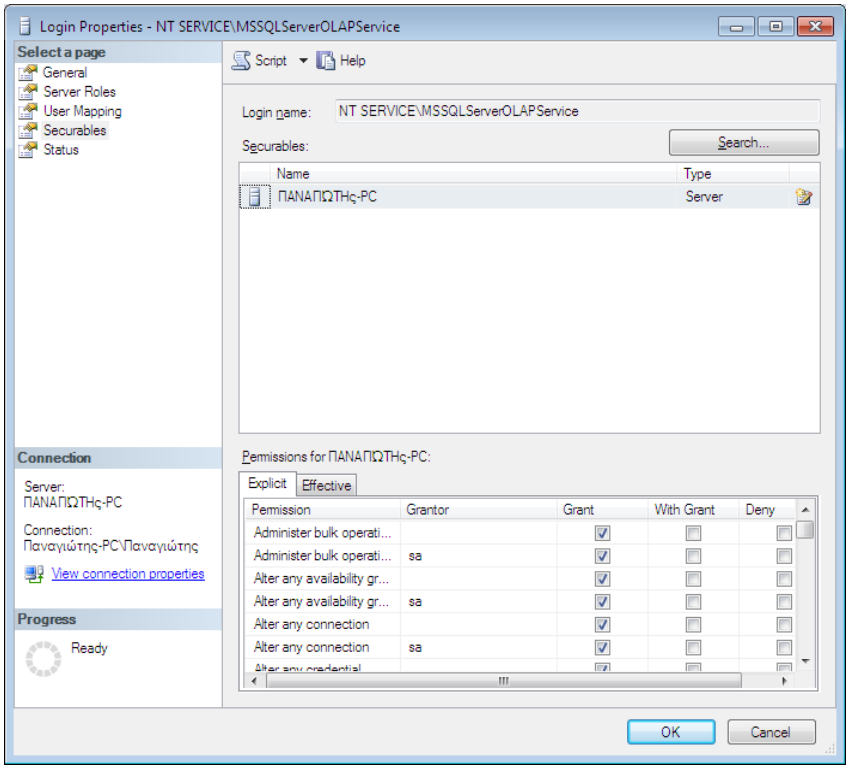
Εικόνα 6.17

Βήμα Γ: Ένα χαρακτηριστικό παράδειγμα δημιουργίας φαίνεται στην Εικόνα 6.18. Δημιουργούμε το Login **"NT SERVICE\MSSQLServerOLAPService"** (για το σφάλμα 28000 στο Visual Studio) και επιλέγουμε ως Default database την **MovieClick.**



Εικόνα 6.18

Βήμα Δ: Έπειτα, για την απόδοση των δικαιωμάτων σε ένα **login**, πηγαίνουμε στον **object explorer** και επιλέγουμε **Logins** από την επιλογή **Security** και, στη συνέχεια, επιλέγουμε με διπλό κλικ το νέο login που δημιουργήσαμε στο προηγούμενο βήμα. Στη νέα καρτέλα, όπως φαίνεται στην Εικόνα 6.19, επιλέγουμε από τα αριστερά το tab **Securables,** φροντίζοντας να είναι επιλεγμένη η βάση MovieClick, και δίνουμε **Grant** σε όλα τα **Permissions**.



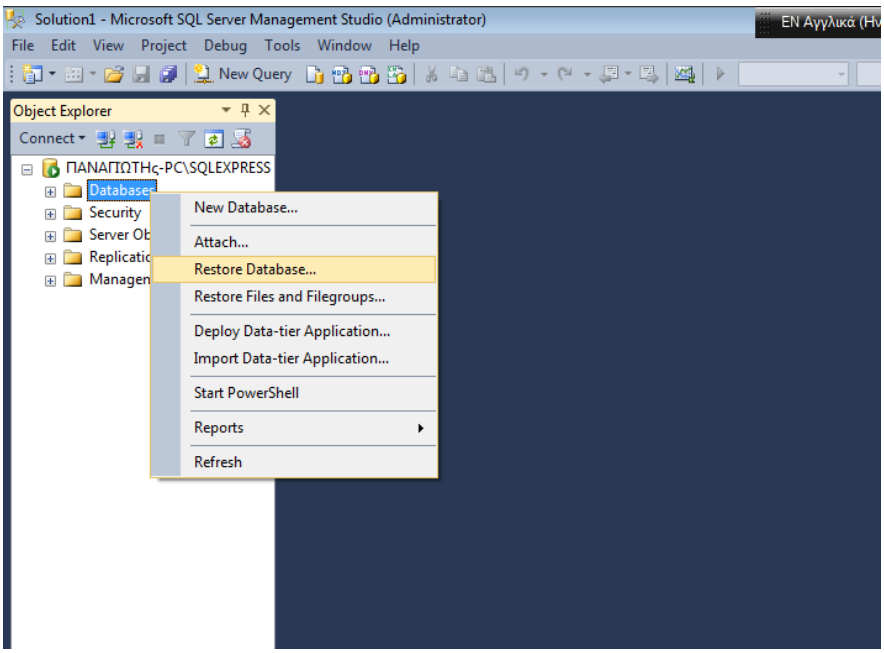
Εικόνα 6.19

Προσοχή! Τονίζεται ότι υπάρχει περίπτωση να απαιτείται η ενεργοποίηση και δεύτερου ή άλλου login, π.χ. το login **NT SERVICE\MSOPLAP$όνομα_instance** (για το σφάλμα 42000 στο Visual Studio), όπου όνομα_instance είναι το όνομα του SQL Server στον υπολογιστή μας.

6.2. Εισαγωγή Βάσης Δεδομένων FoodMart

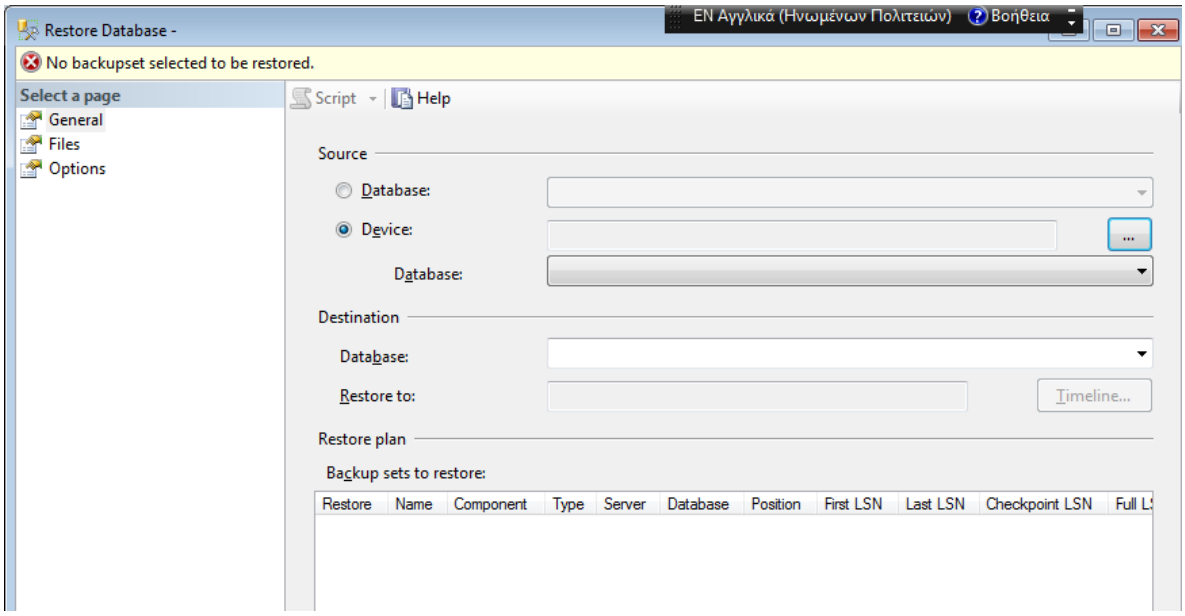
Σ' αυτήν την ενότητα θα εισάγουμε τη βάση δεδομένων foodmart2005_full.bak. Μ' αυτήν τη βάση θα ασχοληθούμε στο κεφάλαιο 11, για να δημιουργήσουμε έναν κύβο πωλήσεων (sales cube), στον οποίο θα εφαρμόσουμε τεχνικές εξόρυξης δεδομένων. Τονίζεται ότι η βάση δεδομένων FoodMart περιέχει στοιχεία πωλήσεων μιας αλυσίδας παντοπωλείων (SuperMarket).

1. Όπως φαίνεται στην Εικόνα 6.20, μεταβαίνουμε στην καρτέλα Object Explorer, κάνουμε δεξί κλικ στο Databases και, στη συνέχεια, επιλέγουμε Restore.



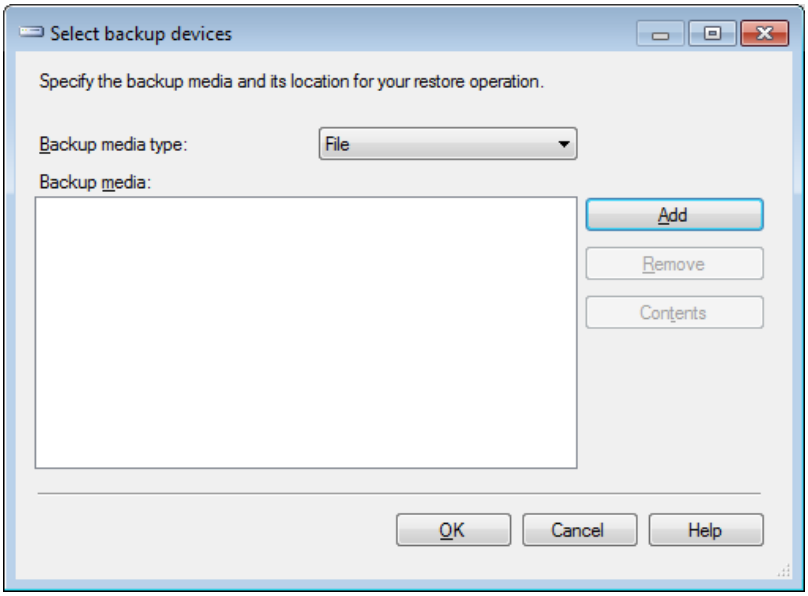
Εικόνα 6.20

2. Όπως φαίνεται στην Εικόνα 6.21, επιλέγουμε καταρχήν το Device και, στη συνέχεια, το κουμπί με τις τρεις τελείες („), ώστε να προσδιορίσουμε τη φυσική θέση στην οποία βρίσκεται η βάση δεδομένων.



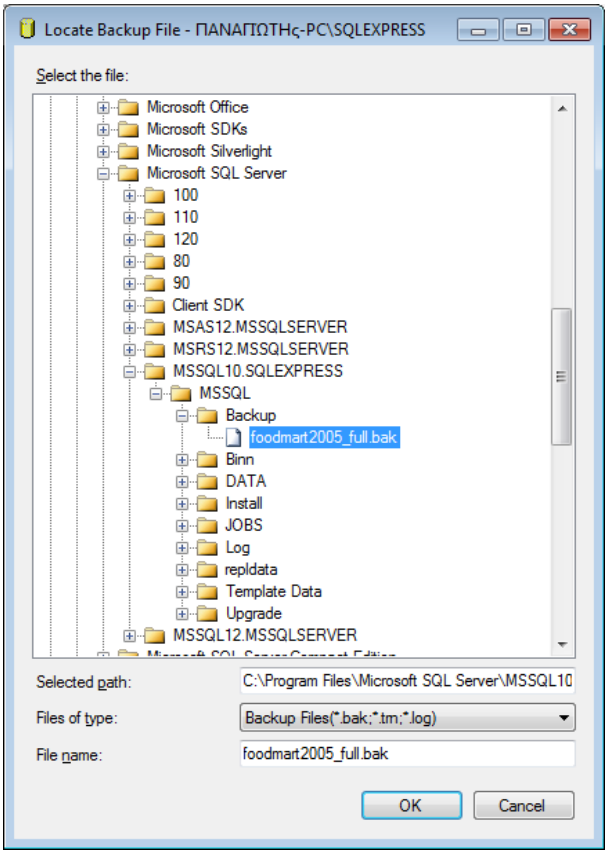
Εικόνα 6.21

3. Όπως φαίνεται στην Εικόνα 6.22, κάνουμε κλικ στο κουμπί Add, προκειμένου να εμφανιστεί το μονοπάτι από το οποίο θα αντλήσουμε το backup της βάσης μας.



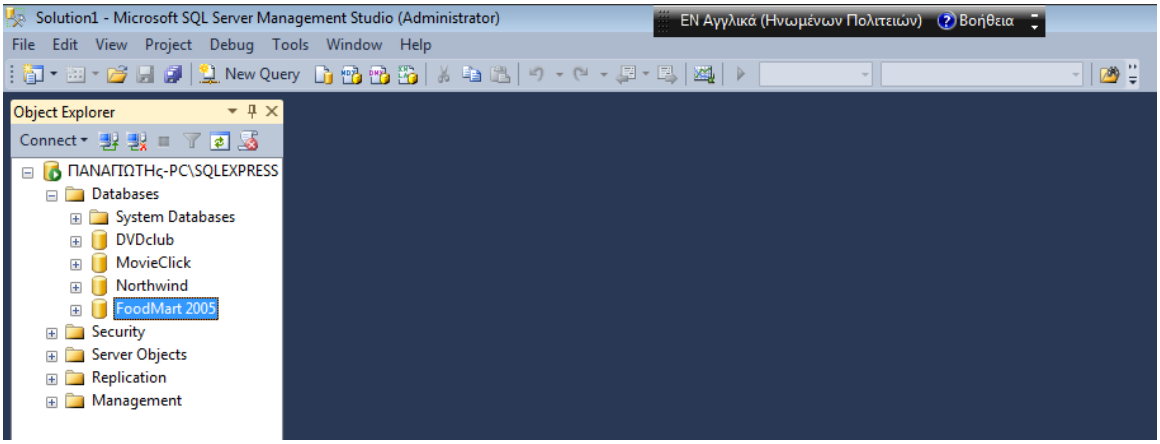
Εικόνα 6.22

4. Εντοπίζουμε τη θέση όπου έχουμε αποθηκεύσει την βάση μας, όπως φαίνεται στην Εικόνα 6.23, και πατάμε OK και ξανά OK.



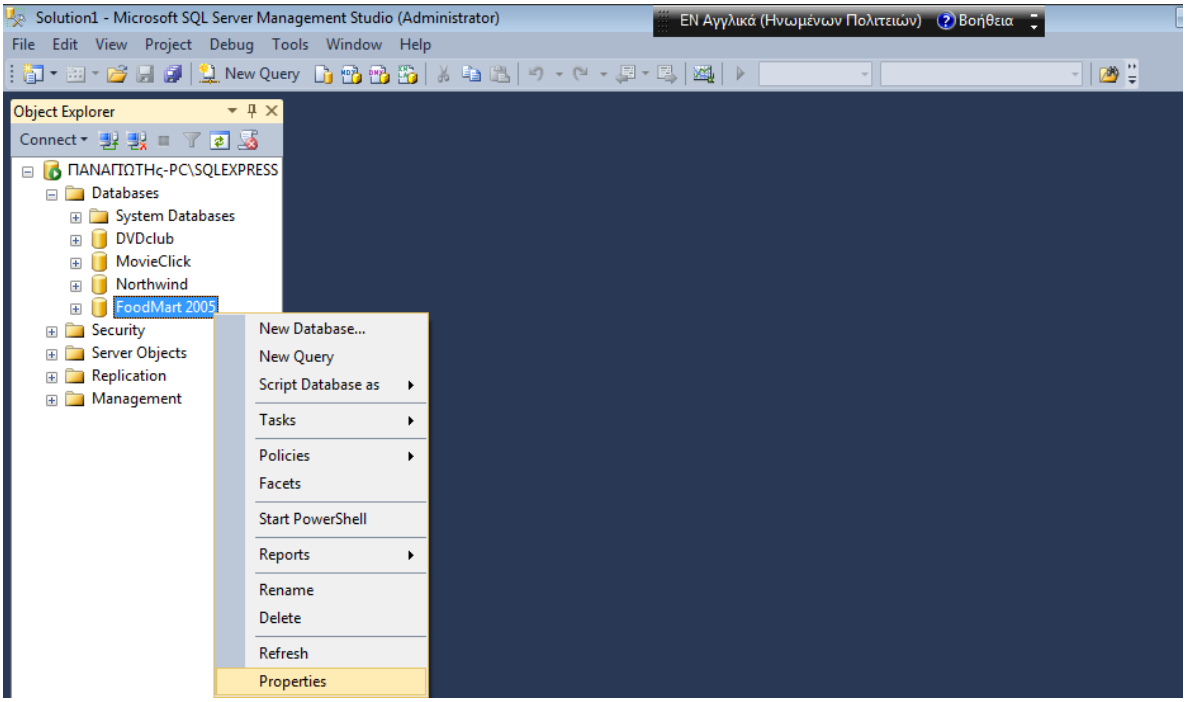
Εικόνα 6.23

5. Εμφανίζεται η καρτέλα του Object Explorer, όπως φαίνεται στην Εικόνα 6.24, και, έτσι, βλέπουμε τη βάση δεδομένων food που εισάγαμε προηγουμένως.



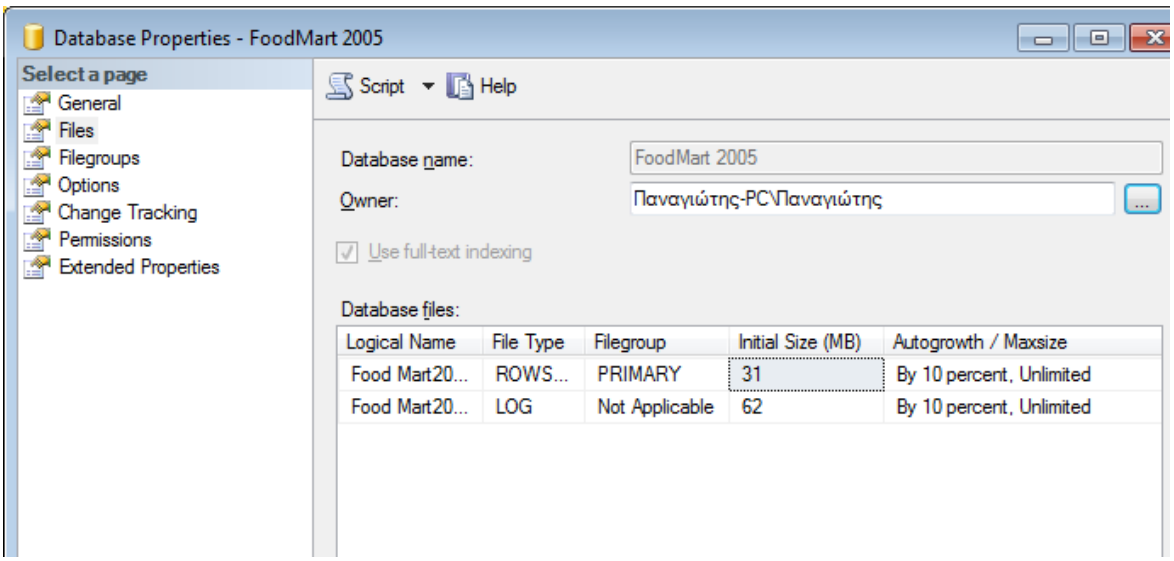
Εικόνα 6.24

6. Σ' αυτό το σημείο θα πρέπει να προσθέσουμε στους owners της βάσης δεδομένων τον χρήστη με τον οποίο συνδεθήκαμε στα Windows. Μεταβαίνουμε, λοιπόν, στον Object Explorer, κάνουμε δεξί κλικ πάνω στην βάση δεδομένων food και επιλέγουμε Properties, όπως φαίνεται στην Εικόνα 6.25.



Εικόνα 6.25

7. Επιλέγουμε την καρτέλα Files, όπως φαίνεται στην Εικόνα 6.26. Στο πεδίο owner θα πρέπει να προσθέσουμε τον χρήστη με τον οποίο συνδεθήκαμε στα Windows (στο παράδειγμα μας: Παναγιώτης). Γι' αυτόν τον λόγο, κάνουμε κλικ δίπλα στο κουμπί και εμφανίζεται το παρακάτω παράθυρο. Σε αυτό κάνουμε κλικ στο Browse (κουμπί με τις τρεις τελείες …) και επιλέγουμε τον χρήστη που έχει τα πλήρη δικαιώματα χρήσης του SQL Server. Μ' αυτόν τον τρόπο θα μπορούμε να προσπελάσουμε την βάση χωρίς πρόβλημα μέσα από το login **service account** στο κεφάλαιο 11 και δεν θα έχουμε προβλήματα δικαιωμάτων πρόσβασης στη βάση δεδομένων μας.

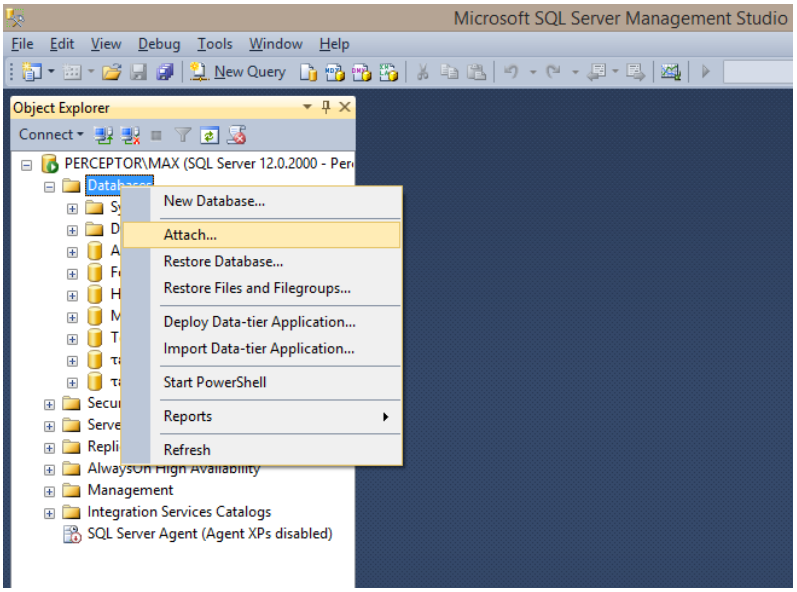


Εικόνα 6.26

8. Τέλος, είναι χρήσιμο να εφαρμόσουμε και για τη βάση δεδομένων FoodMart τα βήματα Α έως Δ που περιγράφονται στο τέλος της Ενότ. 6.1, προκειμένου ο εξ ορισμού χρήστης NT SERVICE\MSSQLServerOLAPService να έχει τη δυνατότητα να τρέξει κάποιον αλγόριθμο εξόρυξης δεδομένων στο περιβάλλον του SQL Server Data Tools του Visual Studio γι' αυτήν τη βάση δεδομένων.

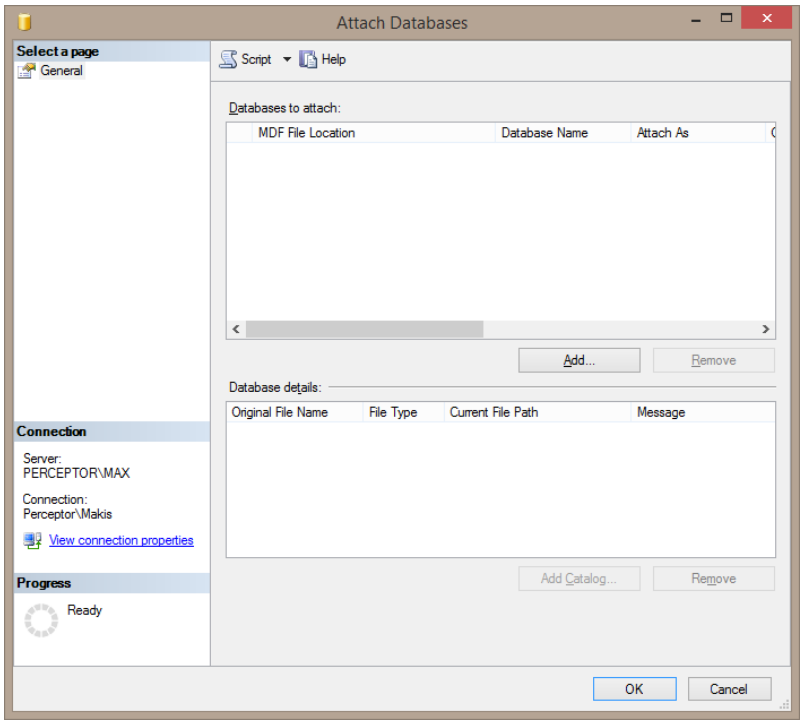
6.3. Εισαγωγή βάσης δεδομένων AdventureWorksDW2008R2

Σ' αυτήν την ενότητα θα εισάγουμε τη βάση δεδομένων **AdventureWorksDW2008R2.mdf**, η οποία θα μας απασχολήσει στο **Κεφάλαιο 10** για την δημιουργία ενός μοντέλου πρόβλεψης χρονοσειράς (time series). Τονίζεται ότι η **Adventure Works** είναι μια πολυεθνική εταιρία που εμπορεύεται ποδήλατα σε διάφορες χώρες. Στην καρτέλα **Object Explorer,** όπως φαίνεται στην Εικόνα 6.27, κάνουμε δεξί κλικ στο **Databases** και, στη συνέχεια, επιλέγουμε **Attach.**



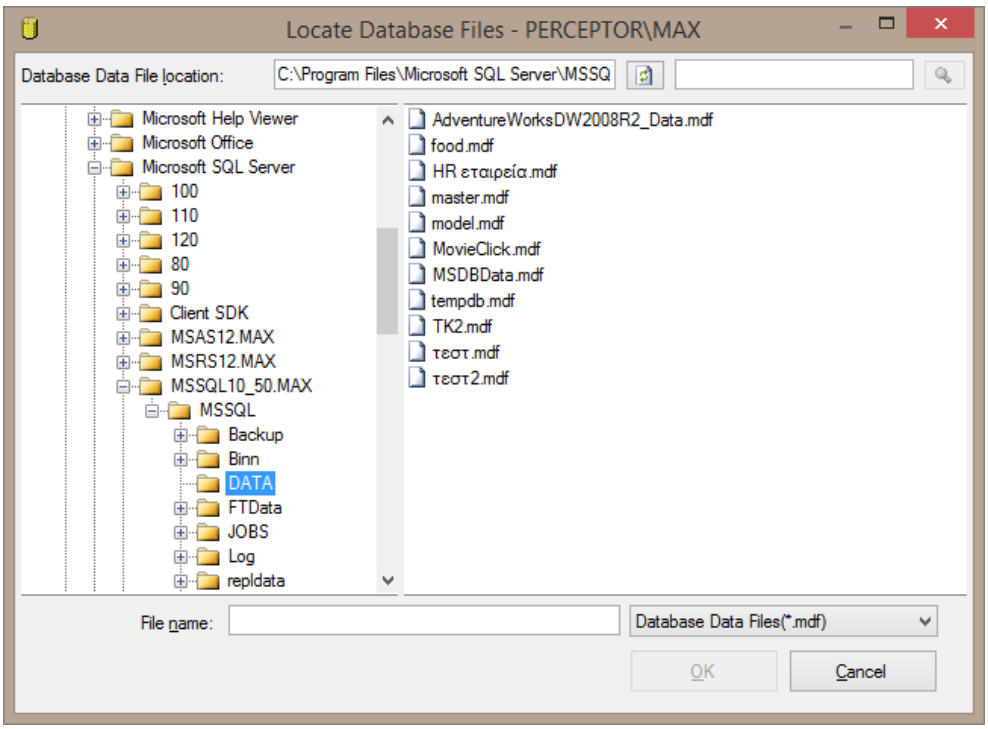
Εικόνα 6.27

1. Στο παράθυρο που εμφανίζεται, όπως φαίνεται στην Εικόνα 6.28, επιλέγουμε **Add,** ώστε να καθορίσουμε τον προορισμό στον οποίο βρίσκεται η βάση δεδομένων.



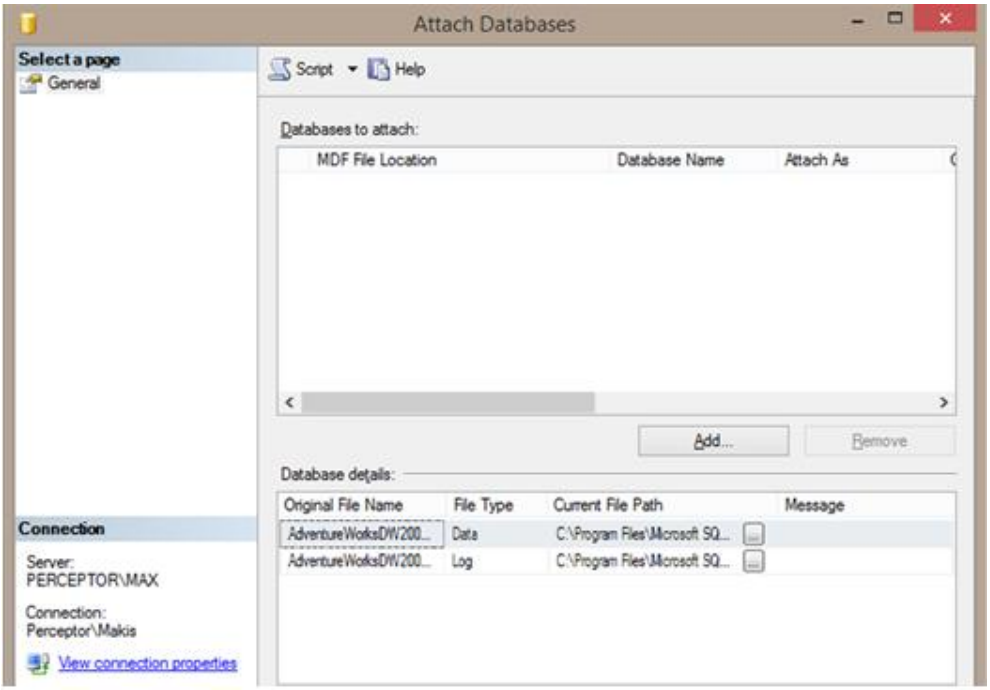
Εικόνα 6.28

2. Στο παράθυρο που εμφανίζεται, όπως φαίνεται στην Εικόνα 6.29, εντοπίζουμε τη θέση όπου έχουμε αποθηκεύσει την βάση δεδομένων με κατάληξη *.mdf και την επιλέγουμε. Επιλέγουμε OK, ώστε να αποθηκεύσουμε την επιλογή μας.

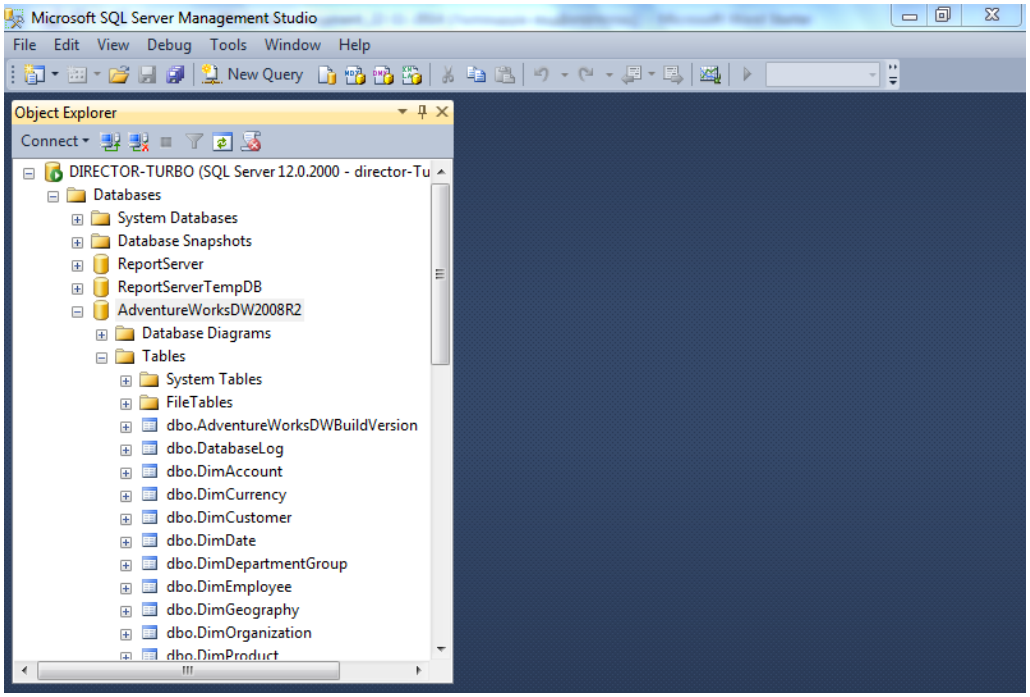


Εικόνα 6.29

3. Εμφανίζεται το παράθυρο με συγκεντρωμένες όλες τις πληροφορίες της βάσης δεδομένων που θέλουμε να εισάγουμε, όπως φαίνεται στην Εικόνα 6.30. Επιλέγουμε OK, ώστε να την εισάγουμε, όπως φαίνεται στην Εικόνα 6.31.



Εικόνα 6.30



Εικόνα 6.31

4. Τέλος, **είναι χρήσιμο να εφαρμόσουμε και για τη βάση δεδομένων** AdventureWorksDW**2008R2 τα** βήματα Α έως Δ **που περιγράφονται στο τέλος της Ενότητας 6.1, προκειμένου ο εξ ορισμού χρήστης** NT SERVICE\MSSQLServerOLAPService να έχει τη δυνατότητα να τρέξει κάποιον αλγόριθμο εξόρυξης δεδομένων στο περιβάλλον του SQL Server Data Tools του Visual Studio γι' αυτήν τη βάση **δεδομένων μας.**

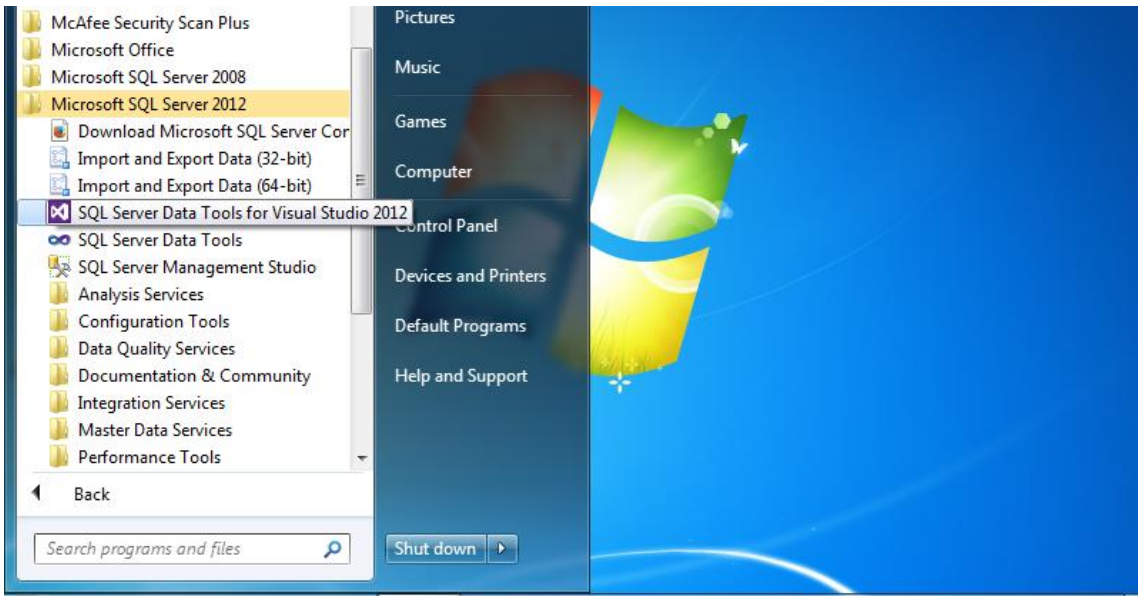
.

6.4. Επεξεργασία βάσης δεδομένων MovieClick

Σ' αυτήν την ενότητα θα επεξεργαστούμε τη βάση δεδομένων MovieClick που έχουμε δημιουργήσει στον SQL Server χρησιμοποιώντας το περιβάλλον SQL Server Data Tools for Visual Studio. Πιο συγκεκριμένα, αρχικά θα δημιουργήσουμε ένα νέο project με το περιβάλλον του SQL Server Data Tools for Visual Studio για την επεξεργασία της βάσης που δημιουργήσαμε. Ακολούθως, θα μας απασχολήσει η αποκατάσταση των συσχετίσεων (relationship) στην βάση του SQL Server 2014.

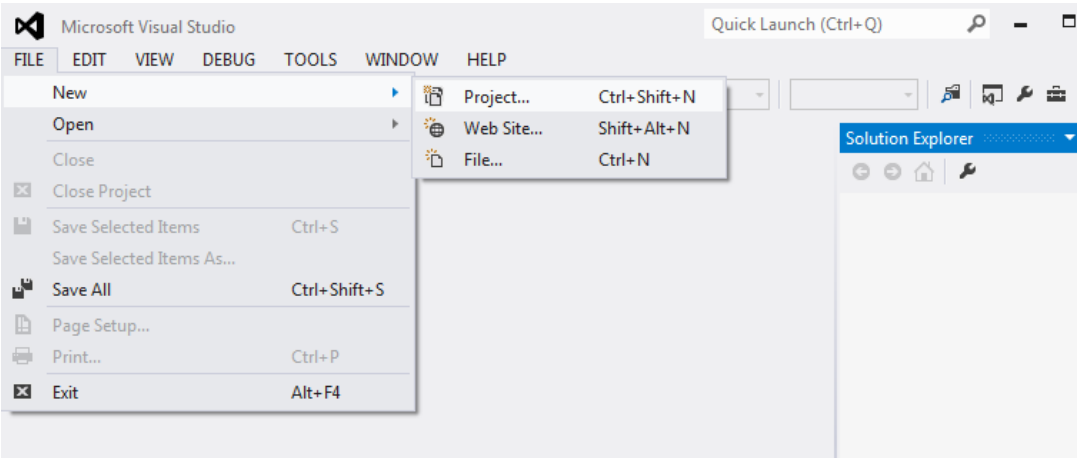
Αναλυτικά βήματα

1. Στο περιβάλλον των Windows επιλέγουμε τη διαδρομή Έναρξη ► Όλα τα Προγράμματα ► SQL Server Data Tools for Visual Studio 2012, όπως φαίνεται στην Εικόνα 6.32.



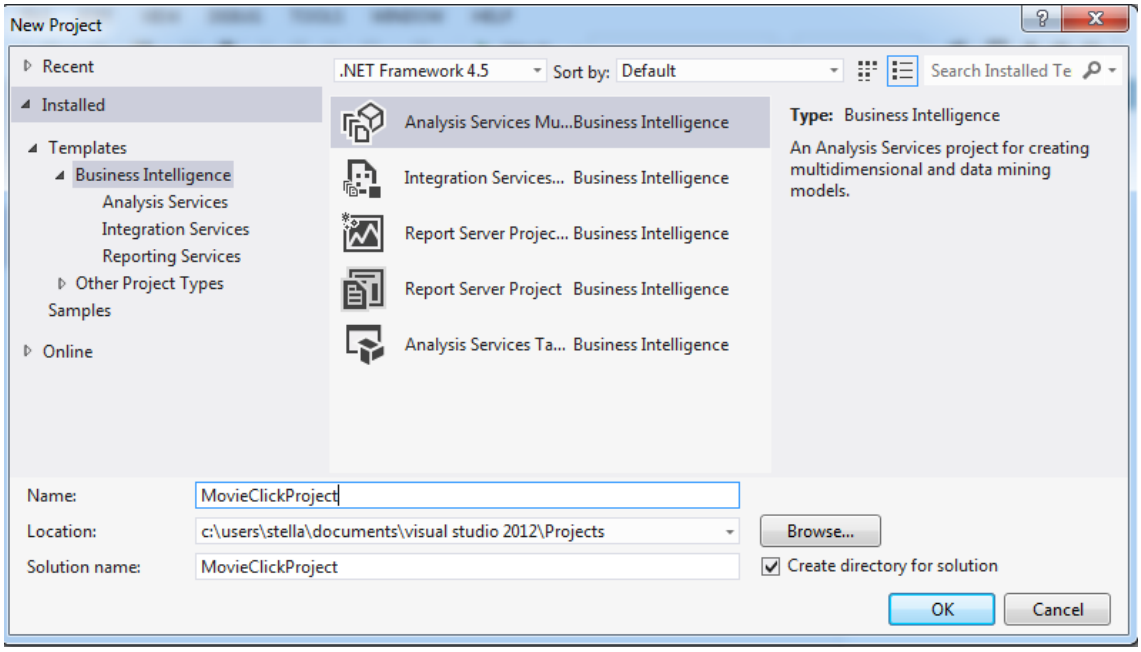
Εικόνα 6.32

2. Όπως φαίνεται στην Εικόνα 6.33, δημιουργούμε ένα νέο project στο οποίο θα εισάγουμε τα δεδομένα από την βάση MovieClick που έχουμε δημιουργήσει στον SQL Server Management Studio. Συγκεκριμένα, επιλέγουμε New ► Project.



Εικόνα 6.33

3. Στον οδηγό που εμφανίζεται, όπως φαίνεται στην Εικόνα 6.34, επιλέγουμε Business Intelligence Projects ► Analysis Services Project.

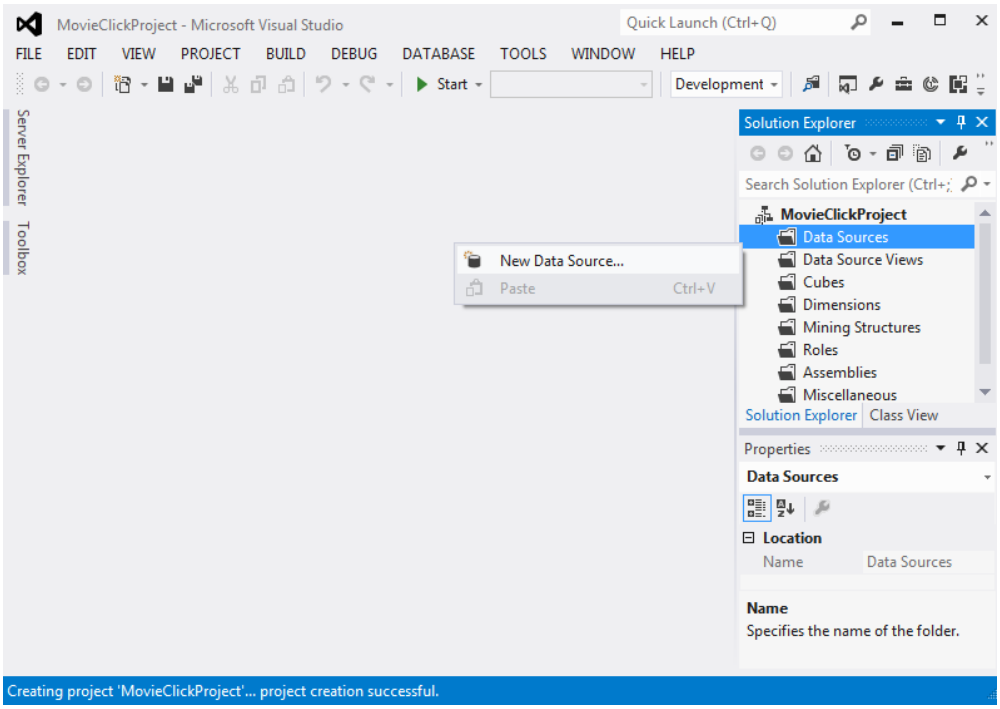


Εικόνα 6.34

Στη συνέχεια, όπως φαίνεται στην ίδια Εικόνα, συμπληρώνουμε τα στοιχεία του project ως εξής:

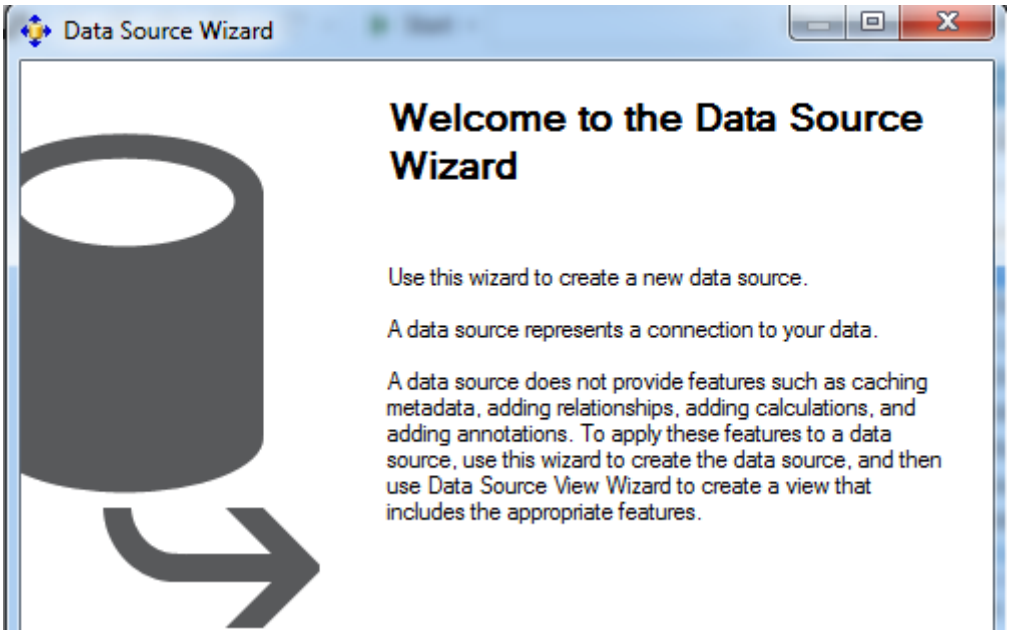
- Στο πεδίο Name δίνουμε το όνομα του Project. Στη συγκεκριμένη περίπτωση δίνουμε το όνομα MovieClickProject.
- Στο πεδίο Location εισάγουμε τον προορισμό αποθήκευσης του project. Στη συγκεκριμένη περίπτωση είναι καθορισμένος ο προεπιλεγμένος προορισμός.
- Στο πεδίο Solution name δίνουμε όνομα στο Solution που θα περιέχει το Project μας. Ένα Solution μπορεί να περιέχει πολλά projects που έχουν κάποια σχέση μεταξύ τους. Στη συγκεκριμένη περίπτωση δίνεται το όνομα MovieClickProject.
- Επιλέγουμε Create directory for solution
- Επιλέγουμε OK, ώστε να δημιουργηθεί το project.

4. Στην καρτέλα Solution Explorer επιλέγουμε MovieClickProject, κάνουμε δεξί κλικ στο Data Sources και πατάμε New Data Source, όπως φαίνεται στην Εικόνα 6.35.



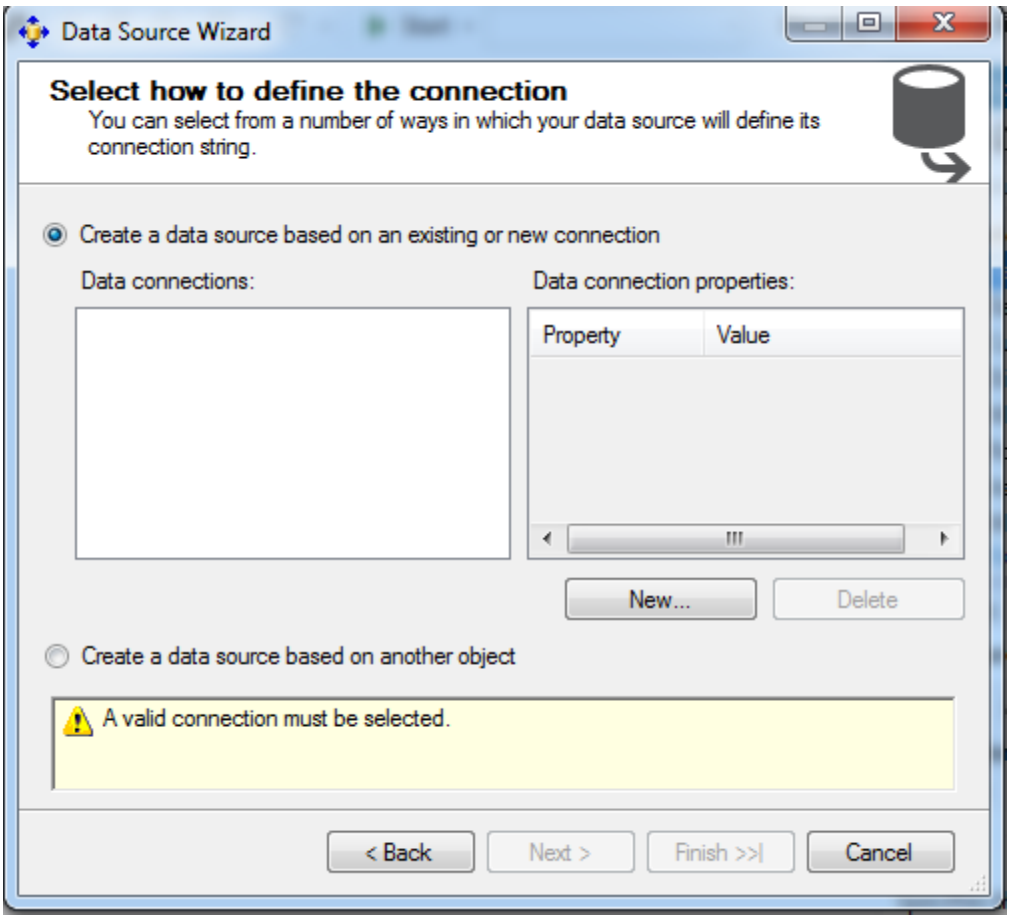
Εικόνα 6.35

5. Στο μήνυμα καλωσορίσματος του οδηγού Data Source Wizard, όπως φαίνεται στην Εικόνα 6.36, επιλέγουμε Next>, ώστε να προχωρήσουμε στο επόμενο βήμα.



Εικόνα 6.36

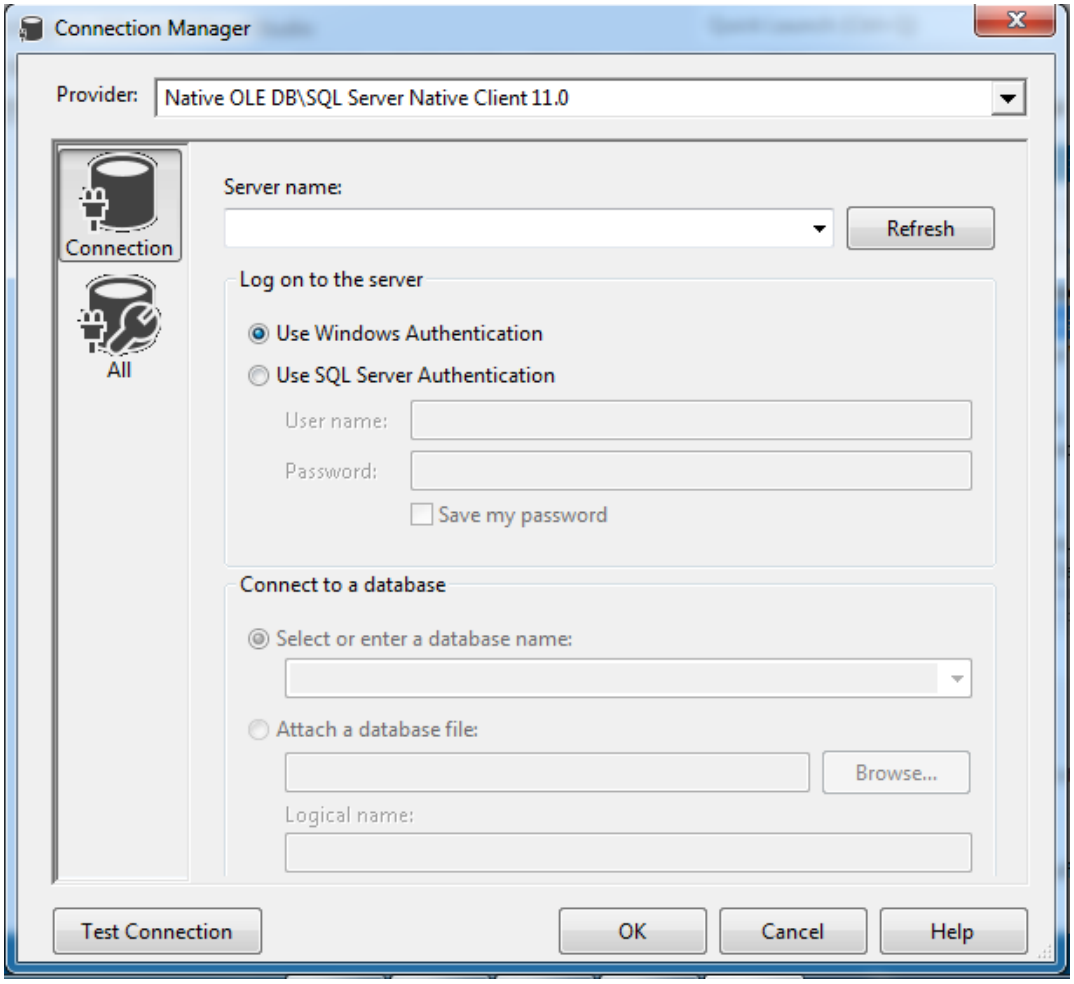
6. Σε αυτό το βήμα, όπως φαίνεται στην Εικόνα 6.37, πρέπει να επιλέξουμε αν θα δημιουργήσουμε μια νέα σύνδεση με έναν διακομιστή για να δημιουργήσουμε το Data Source ή αν θα επιλέξουμε μια ήδη υπάρχουσα σύνδεση. Στη συγκεκριμένη περίπτωση επιλέγουμε "Create a data source based on an existing or new connection" και, στη συνέχεια, επιλέγουμε New…



Εικόνα 6.37

7. Στον οδηγό σύνδεσης που εμφανίζεται, όπως φαίνεται στην Εικόνα 6.38, συμπληρώνουμε τα στοιχεία ως εξής:

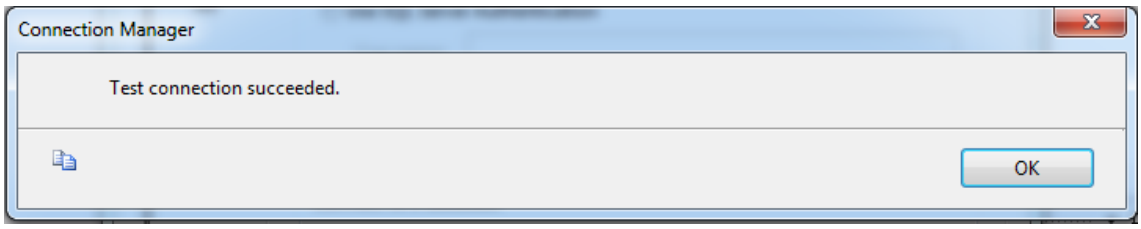
   - Στο πεδίο provider επιλέγουμε Native OLE DB/SQL Native Client.
   - Στο πεδίο Server name συμπληρώνουμε localhost ή το όνομα του υπολογιστή.
   - Επιλέγουμε Use Windows Authentication.
   - Στο πεδίο Connect to a database επιλέγουμε Select or enter a database name, αφού έχουμε ήδη δημιουργήσει τη βάση. Στη συνέχεια, επιλέγουμε τη βάση MovieClick.



Εικόνα 6.38

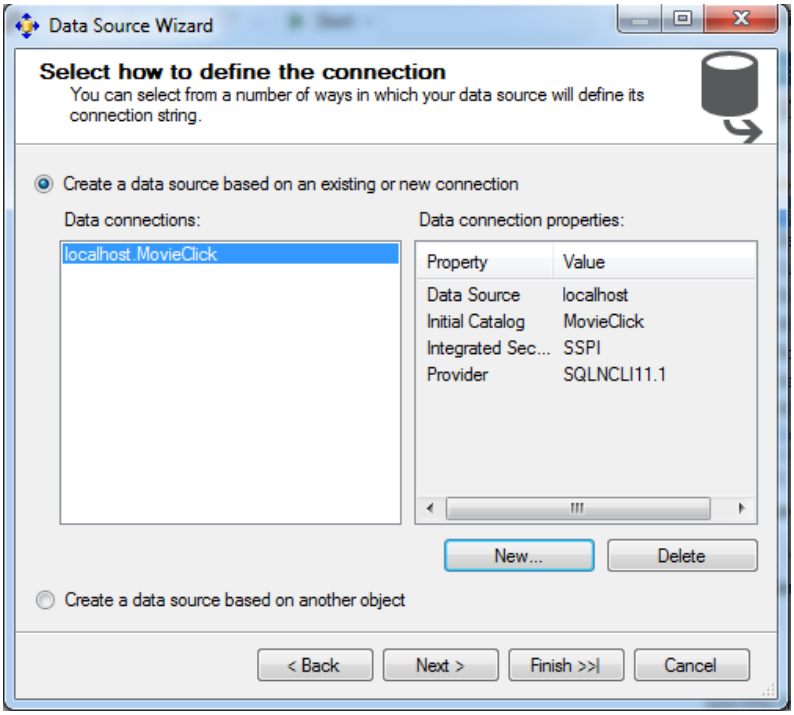
   - Επιλέγουμε Test Connection, για να ελέγξουμε αν μπορούμε να συνδεθούμε στη βάση δεδομένων.

   - Στο παράθυρο που εμφανίζεται επιλέγουμε OK και στη συνέχεια επίσης OK.



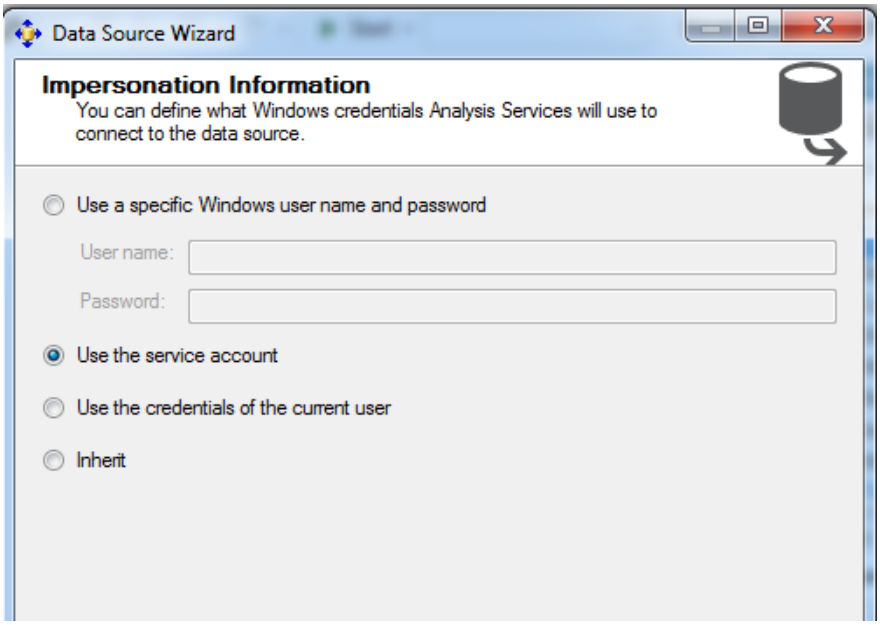
Εικόνα 6.39

8. Επιστρέφοντας στο προηγούμενο παράθυρο, στο πεδίο Data connections βλέπουμε τη σύνδεση localhost.MovieClick (όπως φαίνεται στην Εικόνα 6.40) ή την αντίστοιχη σύνδεση με το όνομα του υπολογιστή. Στη συνέχεια, επιλέγουμε Next>, για να προχωρήσουμε στο επόμενο βήμα.



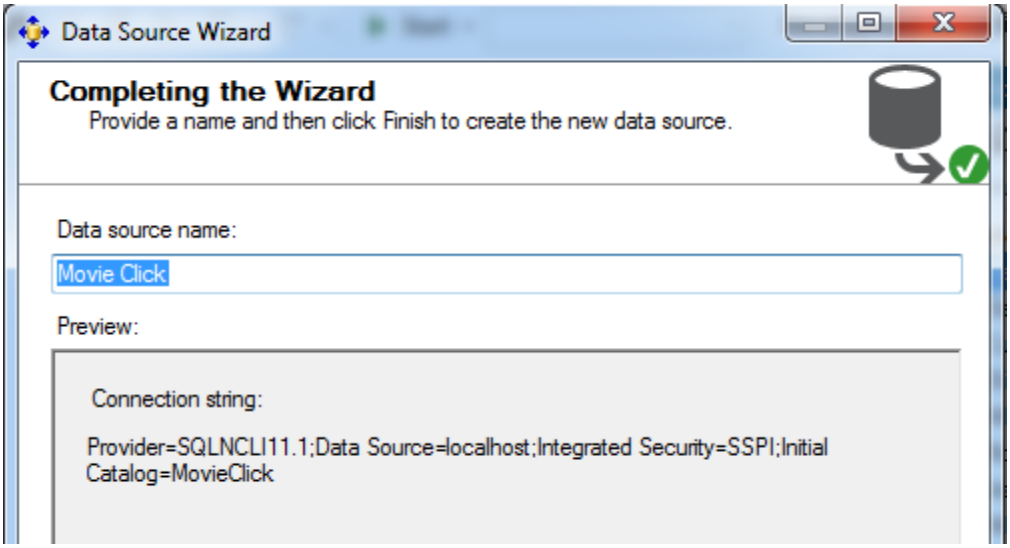
Εικόνα 6.40

9. Όπως εμφανίζεται στην Εικόνα 6.41, επιλέγουμε Use the service account, αφού δεν θέλουμε να ορίσουμε κάποιο άλλο username και password στο data source. Στη συνέχεια, επιλέγουμε Next>.



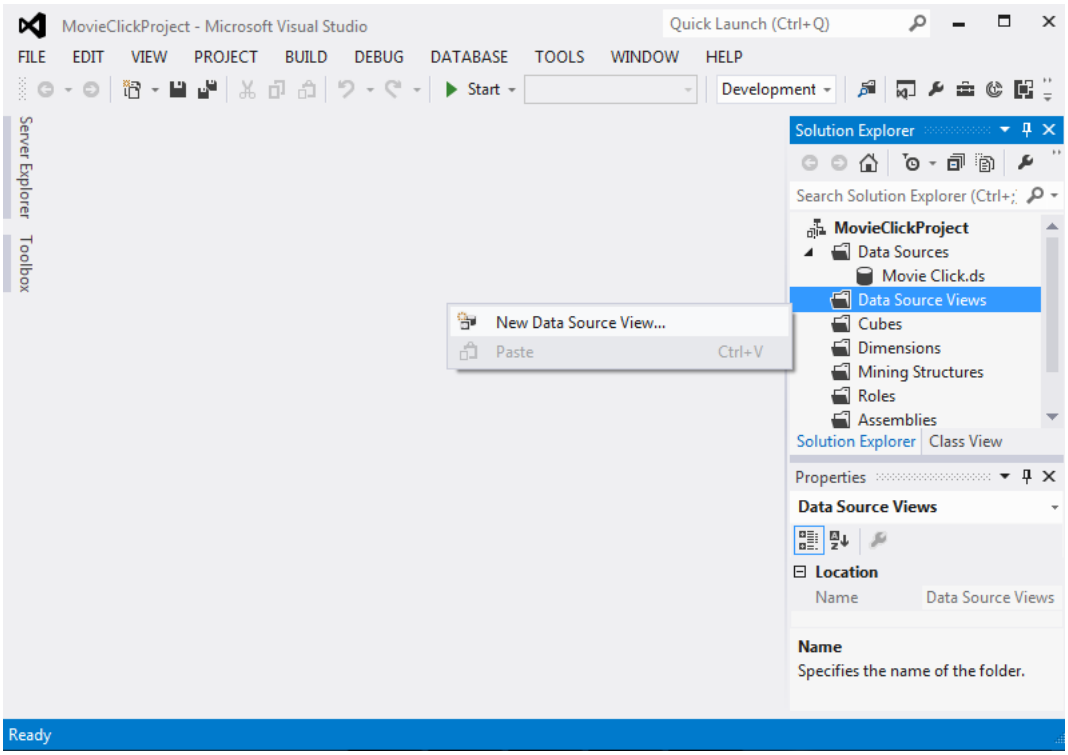
Εικόνα 6.41

10. Σ’ αυτό το βήμα ορίζουμε όνομα στο DataSource. Στη συγκεκριμένη περίπτωση συμπληρώνουμε το όνομα MovieClick, όπως φαίνεται στην Εικόνα 6.42, και, στη συνέχεια, επιλέγουμε Finish, ώστε να δημιουργηθεί το Data Source.



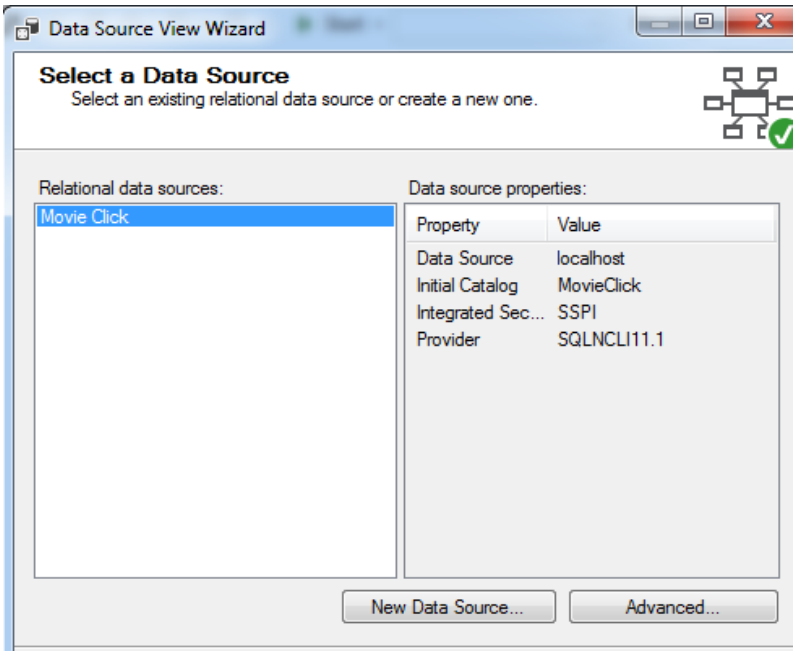
Εικόνα 6.42

11. Στο Data Sources του MovieClickProject βλέπουμε ότι έχει δημιουργηθεί το MovieClick.ds. Στη συνέχεια, θα δημιουργήσουμε ένα Data Source View που θα έχει τα δεδομένα του MovieClick.ds και θα μας προσφέρει τη γραφική αναπαράσταση της βάσης που έχουμε συνδέσει με το MovieClick.ds. Επιλέγουμε την καρτέλα Solution Explorer και, όπως φαίνεται στην Εικόνα 6.43, κάνουμε δεξί κλικ στο Data Source Views και επιλέγουμε New Data Source View…



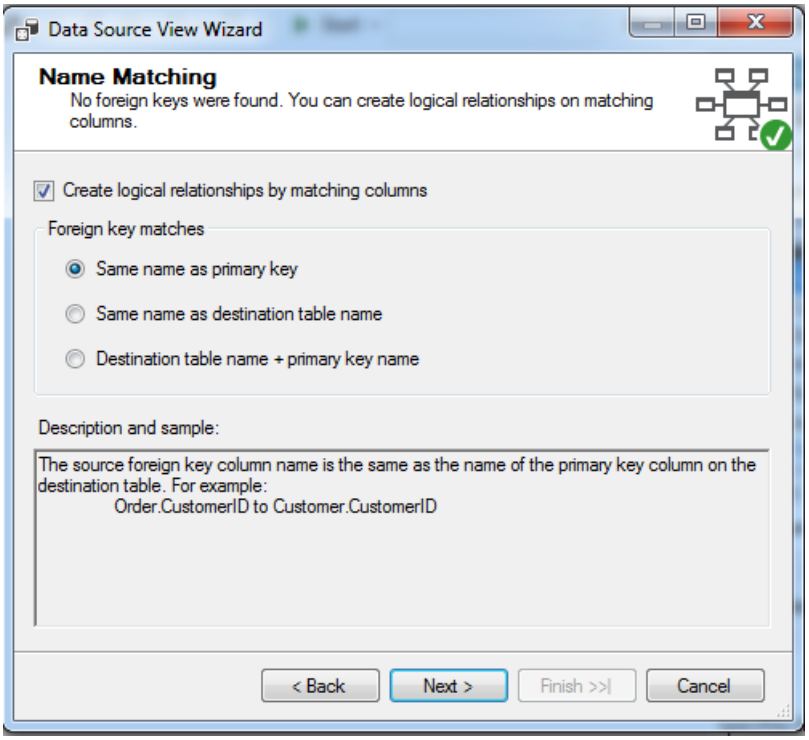
Εικόνα 6.43

12. Όπως εμφανίζεται στην Εικόνα 6.44, επιλέγουμε το Data Source με το οποίο θα συσχετίσουμε το Data Source View που θέλουμε να δημιουργήσουμε. Στη συγκεκριμένη περίπτωση επιλέγουμε το MovieClick και στη συνέχεια επιλέγουμε Next>, ώστε να προχωρήσουμε στο επόμενο βήμα.



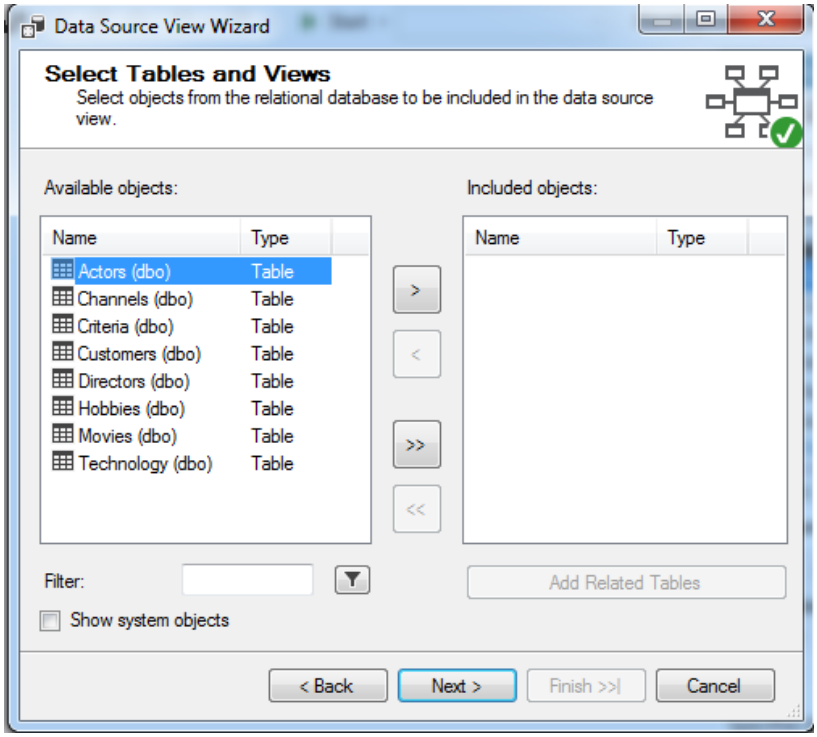
Εικόνα 6.44

13. Επειδή τα δεδομένα που έχει η βάση MovieClick έχουν εισαχθεί από τη βάση της Access, οι συσχετίσεις μεταξύ των πινάκων δεν έχουν μεταφερθεί. Έτσι, επιλέγουμε Create logical relationships by matching columns και Same name as primary key, όπως φαίνεται στην Εικόνα 6.45. Στη συνέχεια, επιλέγουμε Next>, ώστε να προχωρήσουμε στο επόμενο βήμα.



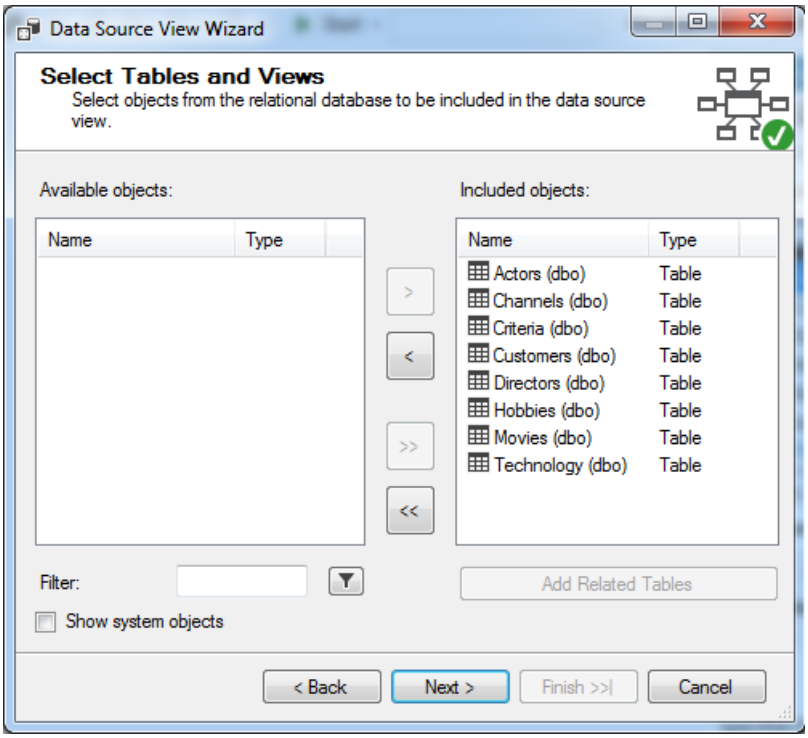
Εικόνα 6.45

14. Όπως φαίνεται στην Εικόνα 6.46, εμφανίζονται όλοι οι πίνακες που είναι διαθέσιμοι για να εισαχθούν. Για να εισάγουμε όλους τους πίνακες επιλέγουμε >> .



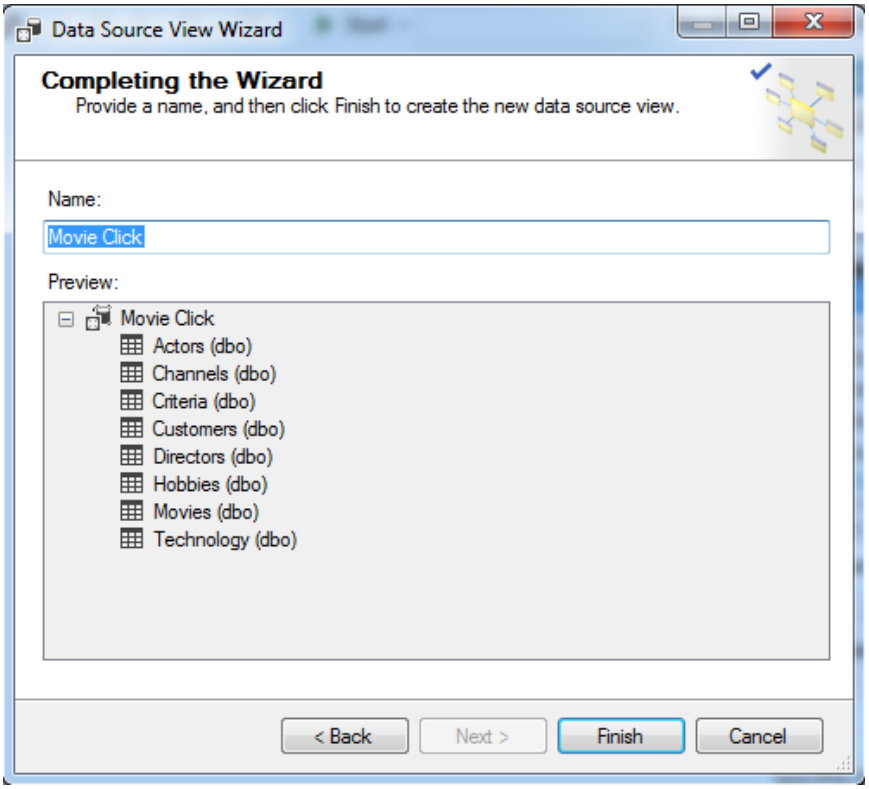
Εικόνα 6.46

Όπως φαίνεται στην Εικόνα 6.47, όλοι οι πίνακες έχουν εισαχθεί. Στη συνέχεια, επιλέγουμε Next>, ώστε να προχωρήσουμε στο επόμενο βήμα.



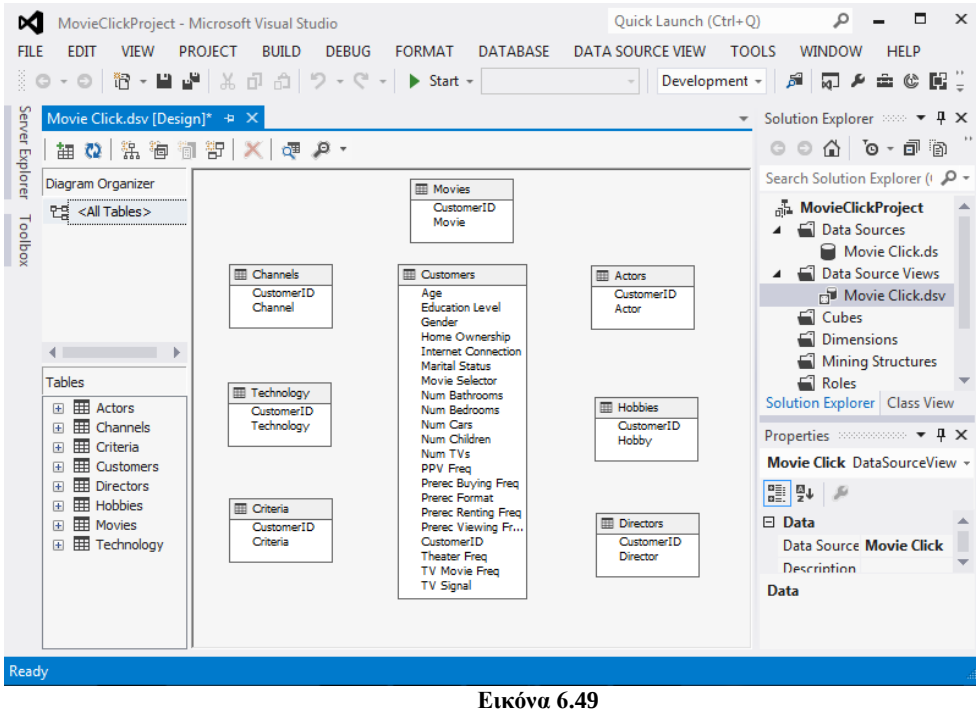
Εικόνα 6.47

15. Σ’ αυτό το στάδιο ορίζουμε όνομα στο Data Source View. Στη συγκεκριμένη περίπτωση, όπως φαίνεται στην Εικόνα 6.48, το ονομάζουμε MovieClick. Στη συνέχεια, επιλέγουμε Finish, ώστε να ολοκληρωθεί η διαδικασία.



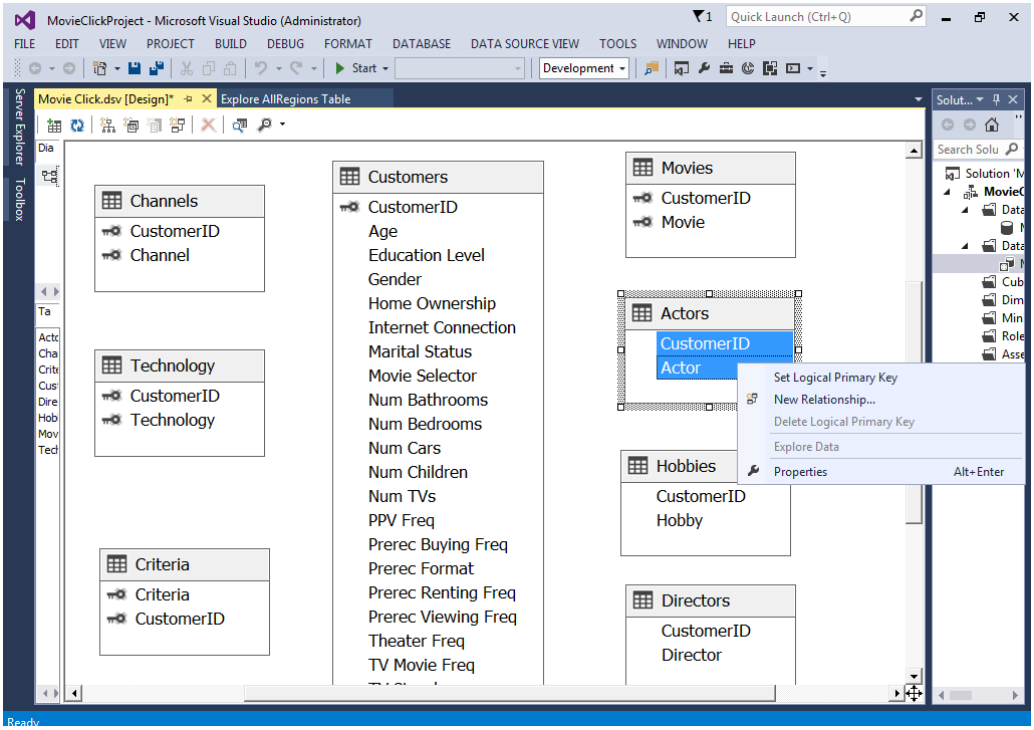
Εικόνα 6.48

16. Στη συνέχεια, όπως φαίνεται στην Εικόνα 6.49, εμφανίζεται το διάγραμμα με τους πίνακες της βάσης μας.



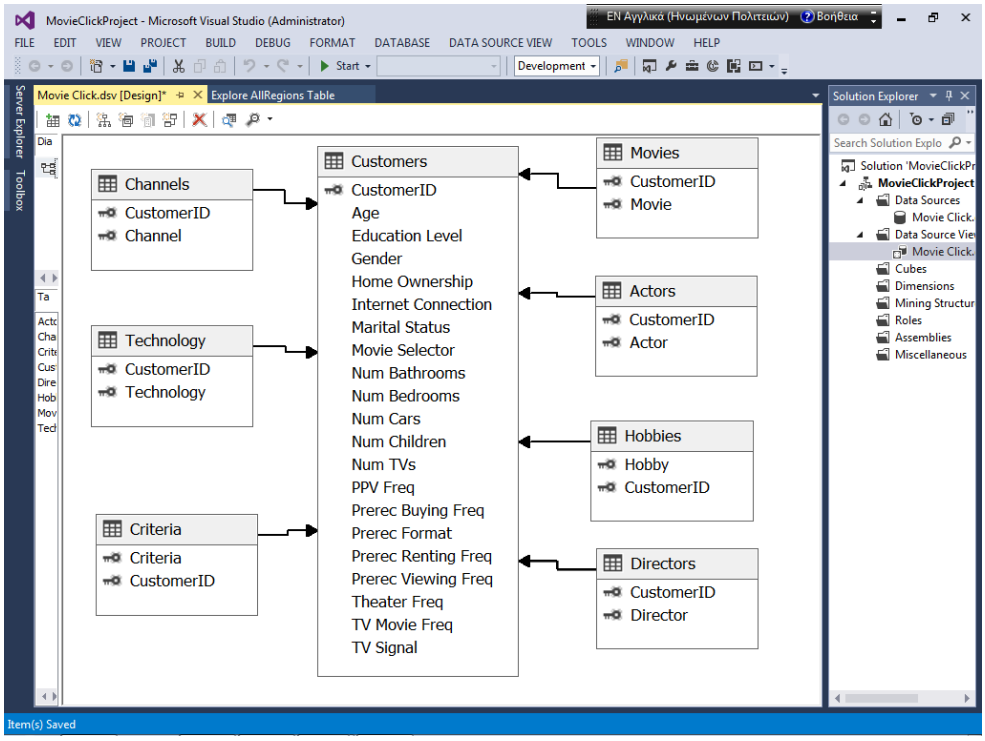
Εικόνα 6.49

17. Στη συνέχεια, θα πρέπει να αποκαταστήσουμε τα πρωτεύοντα κλειδιά στη βάση μας. Έτσι, για τον πίνακα Customers κάνουμε δεξί κλικ στο πεδίο CustomerID και επιλέγουμε Set Logical Primary Key. Για όλους τους υπόλοιπους πίνακες θα πρέπει να ορίσουμε ένα σύνθετο πρωτεύον κλειδί που θα αποτελείται από το πεδίο CustomerID και το αντίστοιχο δεύτερο πεδίο του κάθε πίνακα, όπως φαίνεται στην Εικόνα 6.50.



Εικόνα 6.50

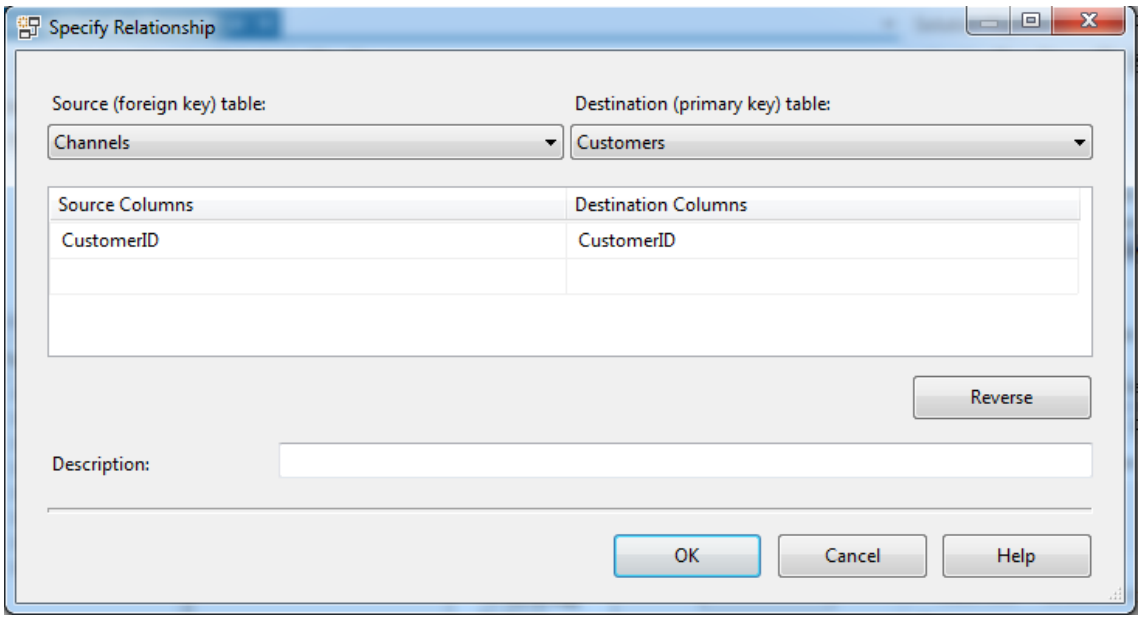
18. Στη συνέχεια θα πρέπει να αποκαταστήσουμε τις συσχετίσεις όλων των πινάκων με τον κεντρικό πίνακα Customers. Έτσι, σε κάθε πίνακα κάνουμε δεξί κλικ στο πεδίο του CustomerID και επιλέγουμε New Relationship, όπως φαίνεται στην Εικόνα 6.51.



Εικόνα 6.51

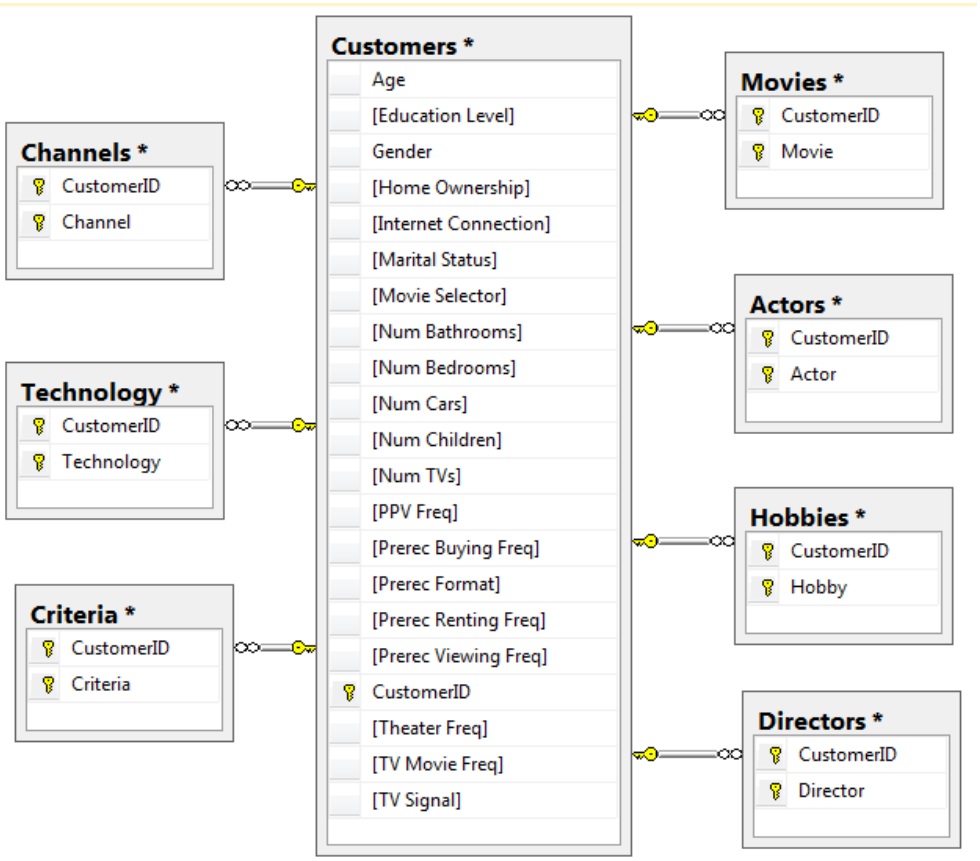
19. Στο αναδυόμενο παράθυρου που φαίνεται στην Εικόνα 6.52, συμπληρώνουμε τα στοιχεία ως εξής:

- Στο πεδίο Source (foreign key) table επιλέγουμε τον πίνακα που θέλουμε να λάβει foreign key. Στην περίπτωσή μας επιλέγουμε οποιονδήποτε πίνακα εκτός από τον πίνακα Customers (π.χ. στην Εικόνα 6.52 έχει επιλεγεί ο πίνακας Channels).
- Στο πεδίο Destination (primary key) table επιλέγουμε τον πίνακα με τον οποίον ταυτίζεται το foreign key. Στη συγκεκριμένη περίπτωση επιλέγουμε τον πίνακα dbo.Customers για όλες τις συνδέσεις.
- Στο πεδίο Source Columns επιλέγουμε το πεδίο του πίνακα που αποτελεί foreign key. Στη συγκεκριμένη περίπτωση επιλέγουμε για όλους τους πίνακες το CustomerID.
- Στο πεδίο Destination Columns επιλέγουμε το πεδίο του πίνακα που ταυτίζεται με το foreign key. Στη συγκεκριμένη περίπτωση επιλέγουμε το CustomerID του πίνακα Customers για όλους τους πίνακες.



Εικόνα 6.52

20. Στην Εικόνα 6.53 εμφανίζουμε τους πίνακες της βάσης μας, αφού έχουν αποκατασταθεί τα πρωτεύοντα κλειδιά και οι συσχετίσεις μεταξύ των δεδομένων. Το συγκεκριμένο σχήμα της βάσης δεδομένων MovieClick έχει παραχθεί στο περιβάλλον του SQL Server Management Studio. Όπως μπορούμε να παρατηρήσουμε, ο κεντρικός πίνακας της βάσης είναι ο πίνακας Customers, ενώ οι συσχετιζόμενοι πίνακες αποθηκεύουν τις προτιμήσεις του κάθε πελάτη. Για παράδειγμα, ο πίνακας Movies καταγράφει τους τίτλους των ταινιών που έχει δει κάθε πελάτης. Ο πίνακας Actors καταγράφει τους αγαπημένους ηθοποιούς κάθε πελάτη. Ο πίνακας Hobbies καταγράφει τα ενδιαφέροντα (travel, computer, photography, κτλ.) του κάθε πελάτη. Ο πίνακας Channels καταγράφει τα τηλεοπτικά κανάλια/παρόχους (Sci-Fi Channel, HBO, Cinemax, κτλ.) που προτιμάει κάθε πελάτης. Ο πίνακας Technology καταγράφει τον εξοπλισμό (DVD player, ηλεκτρονικό υπολογιστή, δορυφορική Τηλεόραση, κτλ.) που διαθέτει κάθε πελάτης, ενώ ο πίνακας Criteria αποθηκεύει τους λόγους (Previews, Genre, Friends Recommendation, κτλ.) για τους οποίους επιλέγει να δει μια ταινία. Τονίζεται ότι το παρακάτω σχεσιακό σχήμα δεν απαιτείται να παραχθεί ξανά, διότι τα βήματα που υλοποιήσαμε σ' αυτήν την ενότητα το έχουν ήδη δημιουργήσει.



Εικόνα 6.53

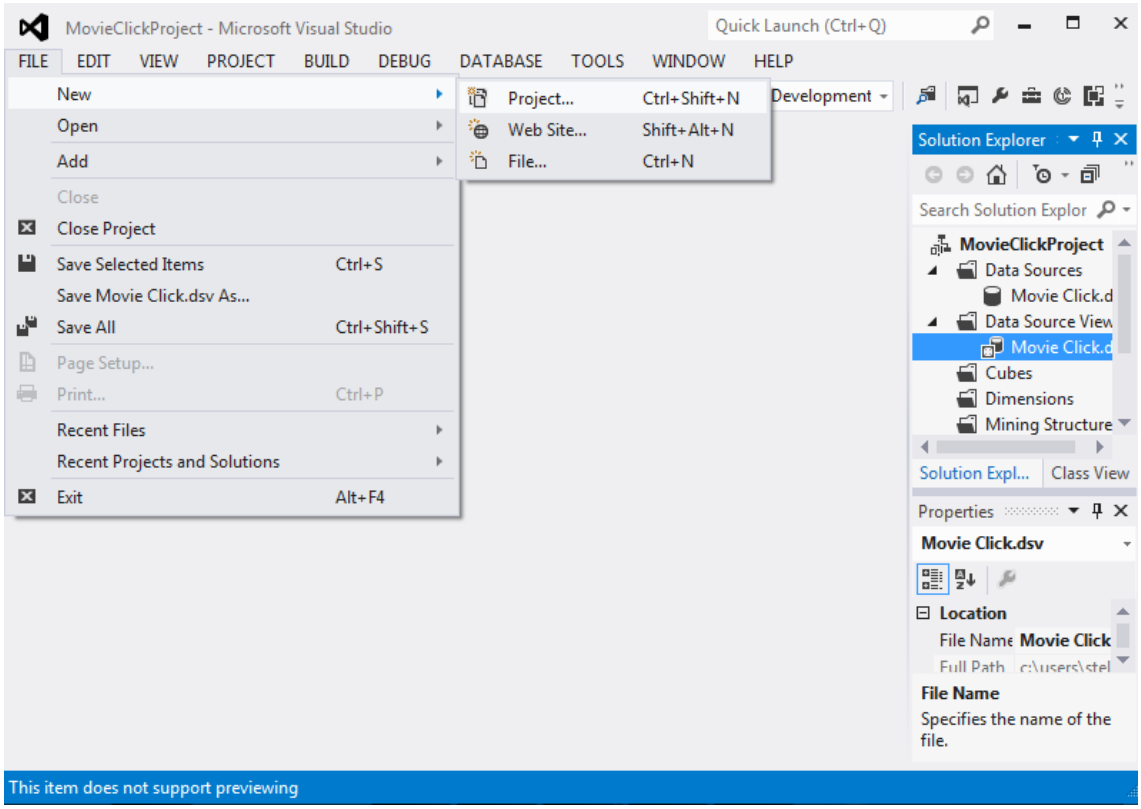
21. Στη συνέχεια, στην καρτέλα Solution Explorer ► κάνουμε δεξί κλικ στο MovieClickProject ► Deploy έτσι ώστε να αποθηκευτούν οι αλλαγές που πραγματοποιήσαμε στο project.

6.5. Επεξεργασία βάσης δεδομένων FoodMart

Σ' αυτήν την ενότητα θα επεξεργαστούμε τη βάση δεδομένων FoodMart που έχουμε δημιουργήσει στον SQL Server χρησιμοποιώντας το περιβάλλον SQL Server Data Tools του Visual Studio.

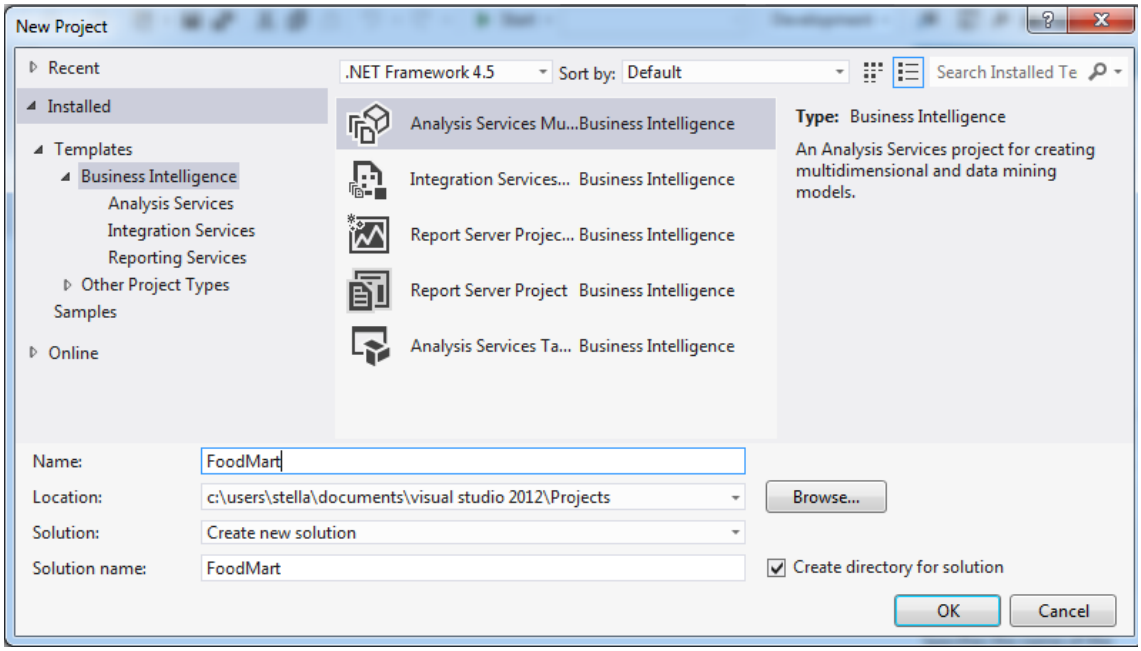
Αναλυτικά βήματα

1. Για να δημιουργήσουμε ένα νέο project στο περιβάλλον του MS SQL Server Data Tools του Visual Studio, ανοίγουμε, όπως φαίνεται στην Εικόνα 6.54, τις επιλογές New και, στη συνέχεια, Project, για να εισάγουμε τα δεδομένα από την βάση FoodMart που έχουμε δημιουργήσει στον SQL Server Management Studio.



Εικόνα 6.54

2. Στον οδηγό που εμφανίζεται, όπως φαίνεται στην Εικόνα 6.55, επιλέγουμε Business Intelligence Templates στο αριστερό pane και Analysis Services Project στα δεξιά. Επιλέγουμε OK, ώστε να δημιουργηθεί το project.

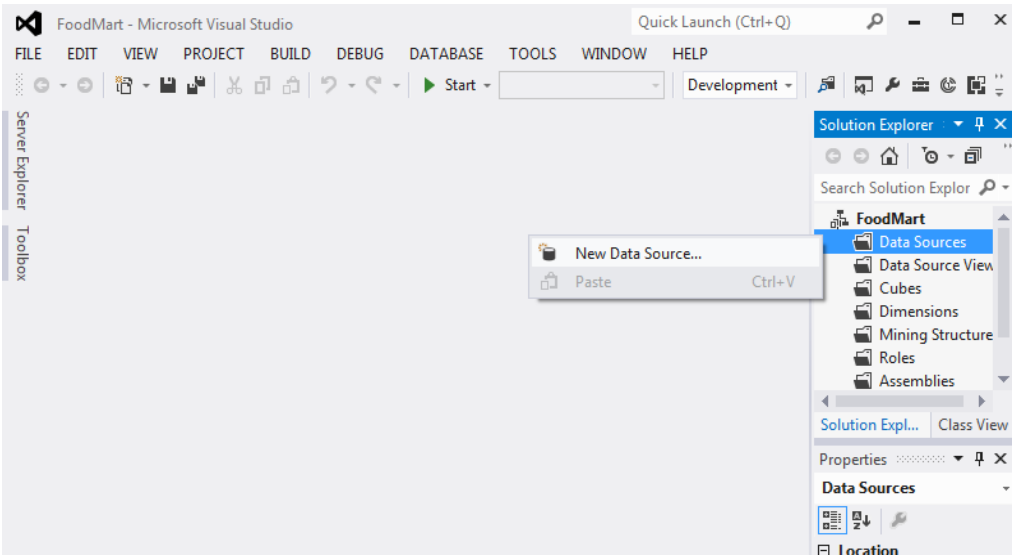


Εικόνα 6.55

Στη συνέχεια, συμπληρώνουμε τα στοιχεία του project ως εξής:

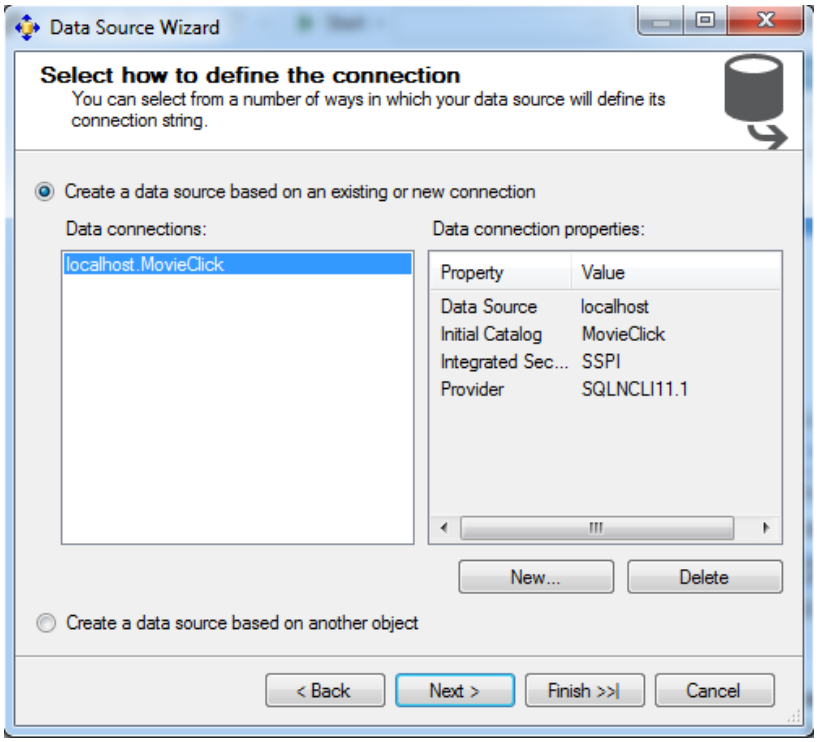
- Στο πεδίο Name δίνουμε το όνομα του Project. Στη συγκεκριμένη περίπτωση δίνουμε το όνομα FoodMart.
- Στο πεδίο Location εισάγουμε τον προορισμό αποθήκευσης του project. Στη συγκεκριμένη περίπτωση είναι καθορισμένος ο προεπιλεγμένος προορισμός.
- Στο πεδίο Solution name δίνουμε όνομα στο Solution που θα περιέχει το Project μας. Ένα Solution μπορεί να περιέχει πολλά projects που έχουν κάποια σχέση μεταξύ τους. Στη συγκεκριμένη περίπτωση δίνεται το όνομα FoodMart.
- Επιλέγουμε Create directory for solution

3. Στην καρτέλα Solution Explorer επιλέγουμε FoodMart. Κάνουμε δεξί κλικ στο Data Sources και επιλέγουμε New Data Source, όπως φαίνεται στην Εικόνα 6.56.



Εικόνα 6.56

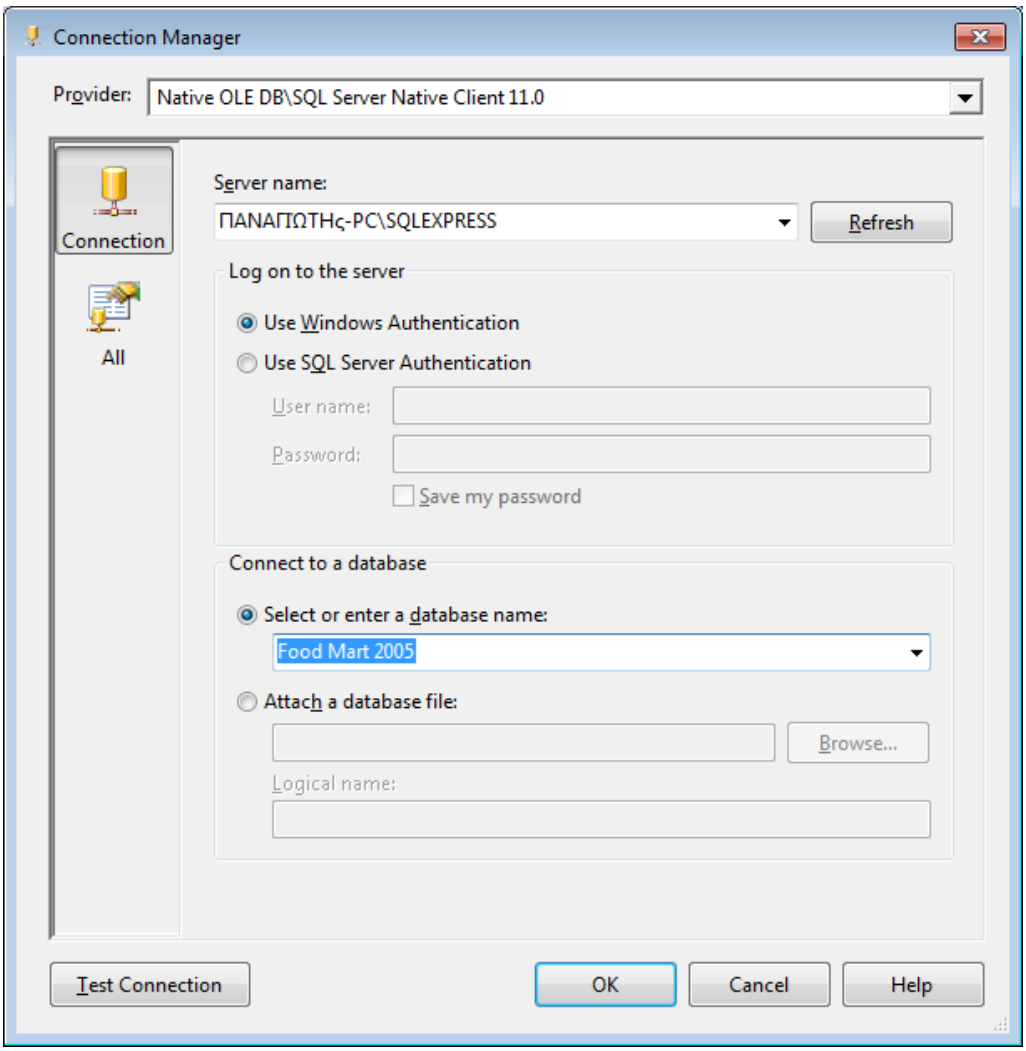
4. Σ' αυτό το βήμα, όπως φαίνεται στην Εικόνα 6.57, πρέπει να επιλέξουμε αν θα δημιουργήσουμε μια νέα σύνδεση με έναν διακομιστή για να δημιουργήσουμε το Data Source ή αν θα επιλέξουμε μια ήδη υπάρχουσα σύνδεση. Στη συγκεκριμένη περίπτωση επιλέγουμε Create a data source based on an existing or new connection και, στη συνέχεια, επιλέγουμε New…



Εικόνα 6.57

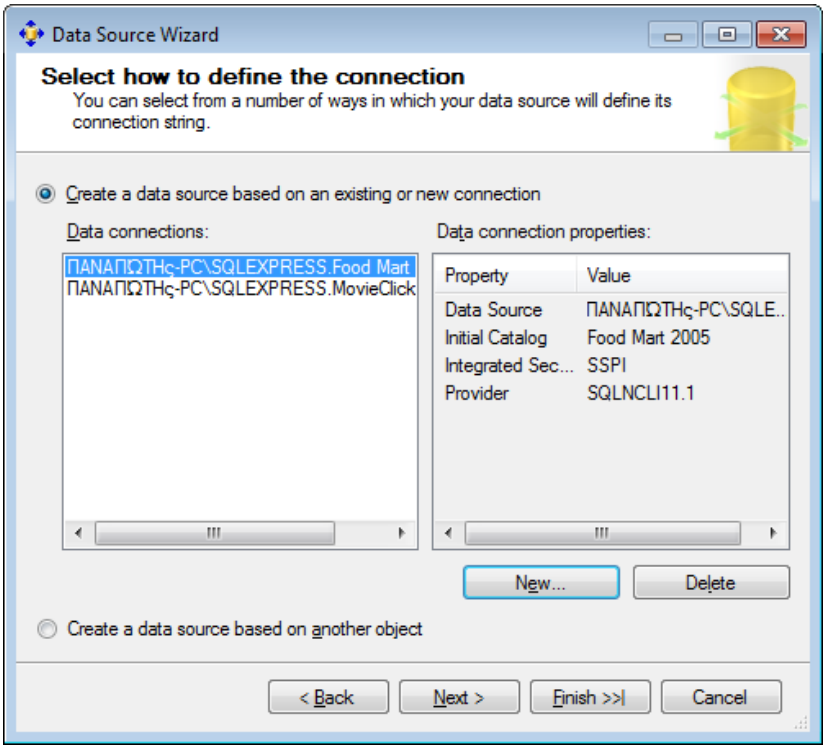
5. Στον οδηγό σύνδεσης, όπως φαίνεται στην Εικόνα 6.58, συμπληρώνουμε τα στοιχεία ως εξής:

- Στο πεδίο provider επιλέγουμε Native OLE DB/SQL Native Client
- Στο πεδίο Server name συμπληρώνουμε localhost ή το όνομα του υπολογιστή.
- Επιλέγουμε Use Windows Authentication
- Στο πεδίο Connect to a database επιλέγουμε Select or enter a database name, αφού έχουμε ήδη δημιουργήσει την βάση. Στη συνέχεια, επιλέγουμε τη βάση Food.



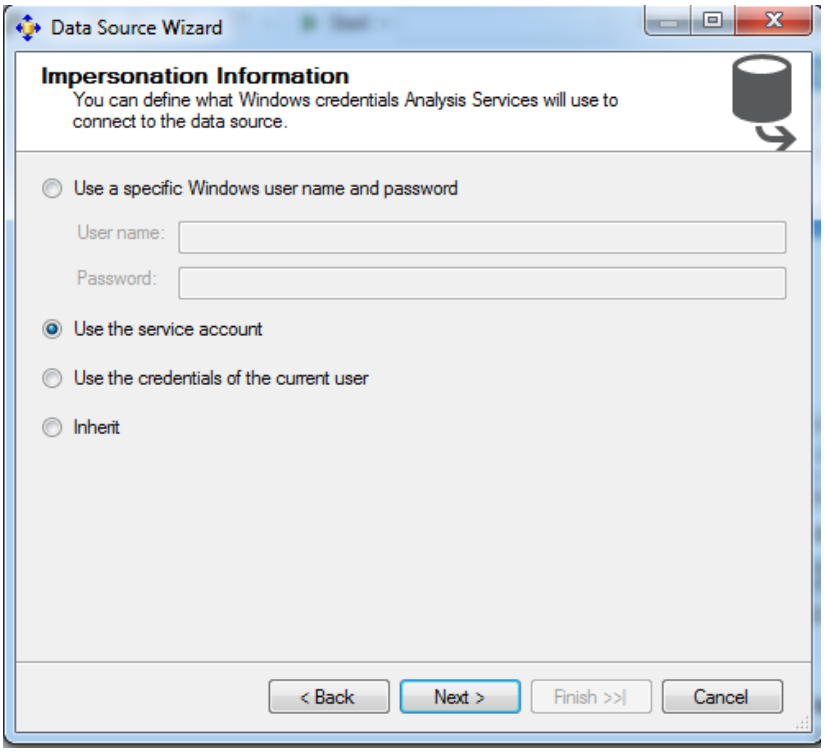
Εικόνα 6.58

6. Επιστρέφοντας στο προηγούμενο παράθυρο, όπως φαίνεται στην Εικόνα 6.59, στο πεδίο Data connections βλέπουμε τη σύνδεση localhost.Food Mart 2005 ή την αντίστοιχη σύνδεση με το όνομα του υπολογιστή. Στη συνέχεια, επιλέγουμε Next>, για να προχωρήσουμε στο επόμενο βήμα.



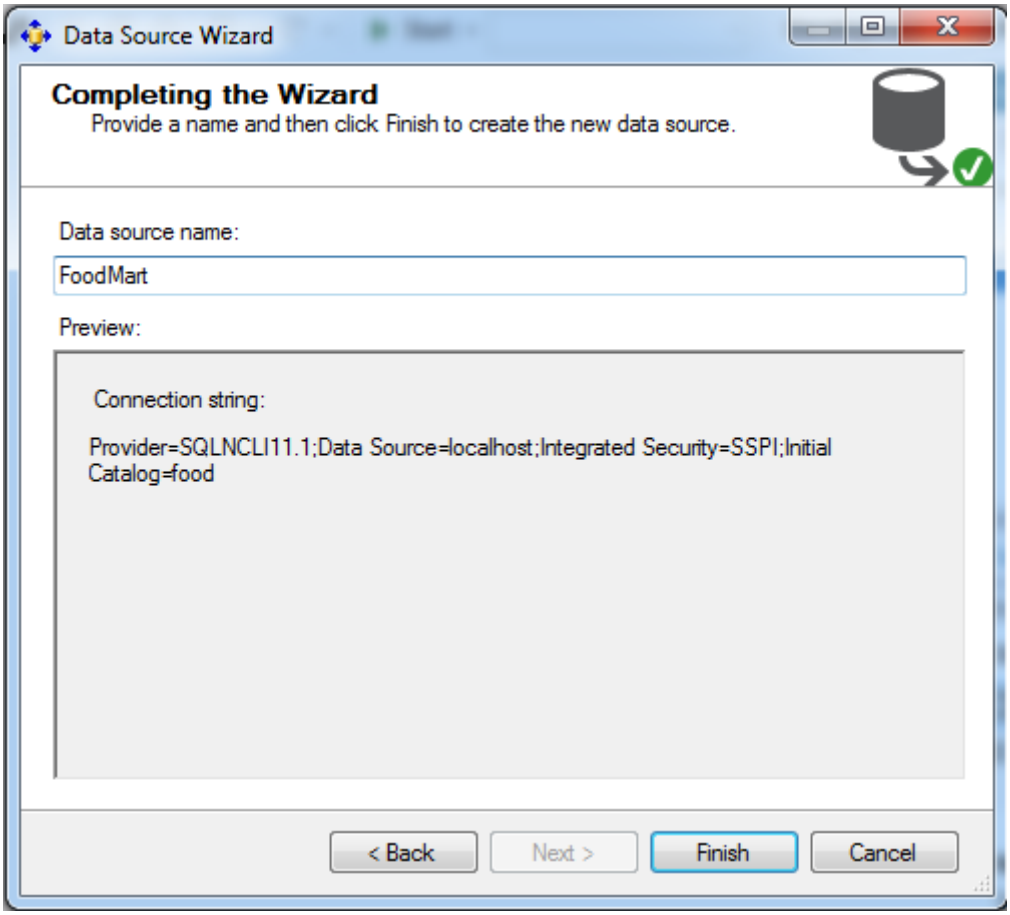
Εικόνα 6.59

7. Επιλέγουμε Use the service account, όπως φαίνεται στην Εικόνα 6.60, και, στη συνέχεια, Next>.



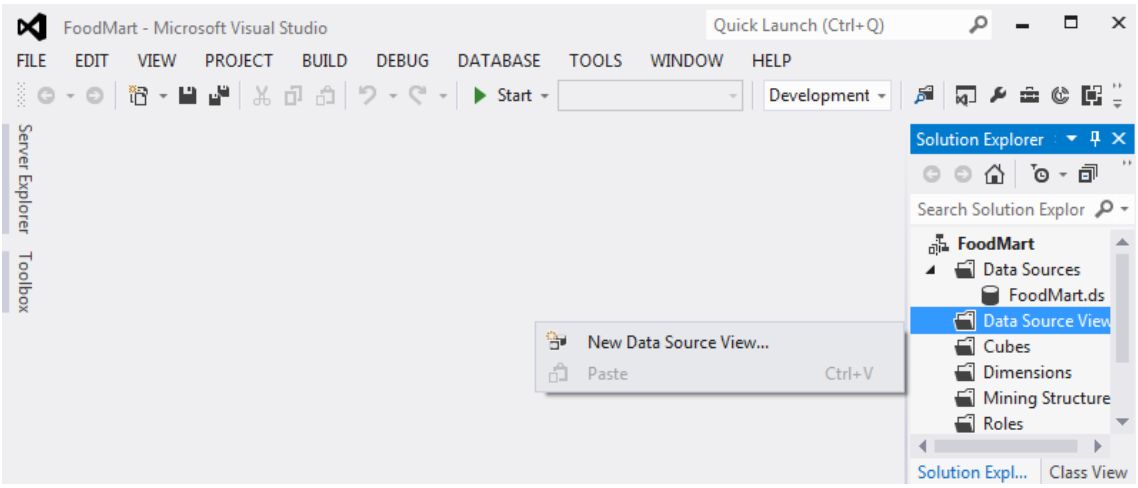
Εικόνα 6.60

8. Σ’ αυτό το βήμα δίνουμε ένα όνομα στο DataSource. Στη συγκεκριμένη περίπτωση, όπως φαίνεται στην Εικόνα 6.61, συμπληρώνουμε το όνομα FoodMart και, στη συνέχεια, επιλέγουμε Finish, ώστε να δημιουργηθεί το DataSource.



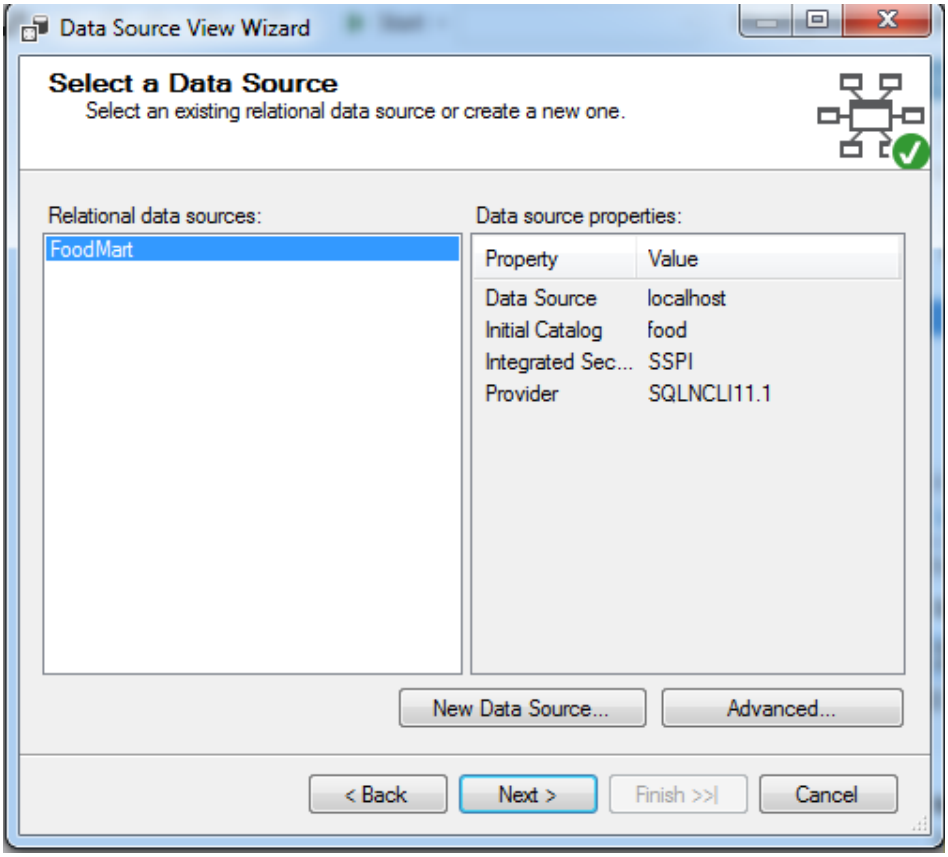
Εικόνα 6.61

9. Στο Data Sources του project FoodMart βλέπουμε ότι έχει δημιουργηθεί το FoodMart.ds. Στη συνέχεια, θα δημιουργήσουμε ένα Data Source View που θα έχει τα δεδομένα του MovieClick.ds και θα μας προσφέρει τη γραφική αναπαράσταση της βάσης που έχουμε συνδέσει με το FoodMart.ds. Επιλέγουμε, λοιπόν, την καρτέλα Solution Explorer, κάνουμε δεξί κλικ στο Data Source Views και πατάμε New Data Source View…, όπως φαίνεται στην Εικόνα 6.62.



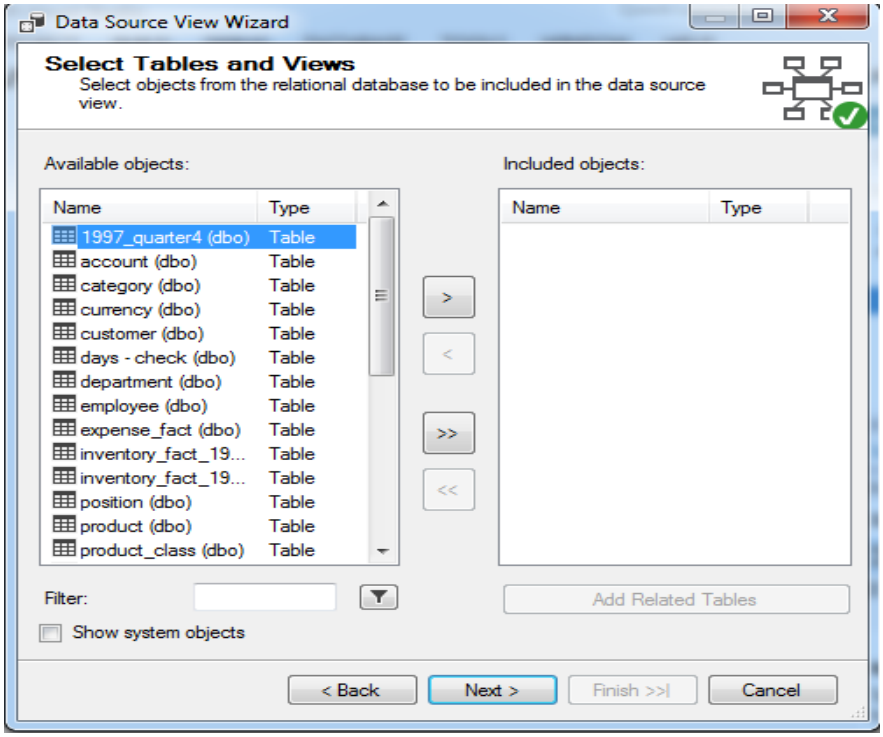
Εικόνα 6.62

10. Όπως φαίνεται στην Εικόνα 6.63, επιλέγουμε το Data Source με το οποίο θα συσχετίσουμε το Data Source View που θέλουμε να δημιουργήσουμε. Στη συγκεκριμένη περίπτωση, επιλέγουμε το FoodMart και, στη συνέχεια, επιλέγουμε Next>, ώστε να προχωρήσουμε στο επόμενο βήμα.



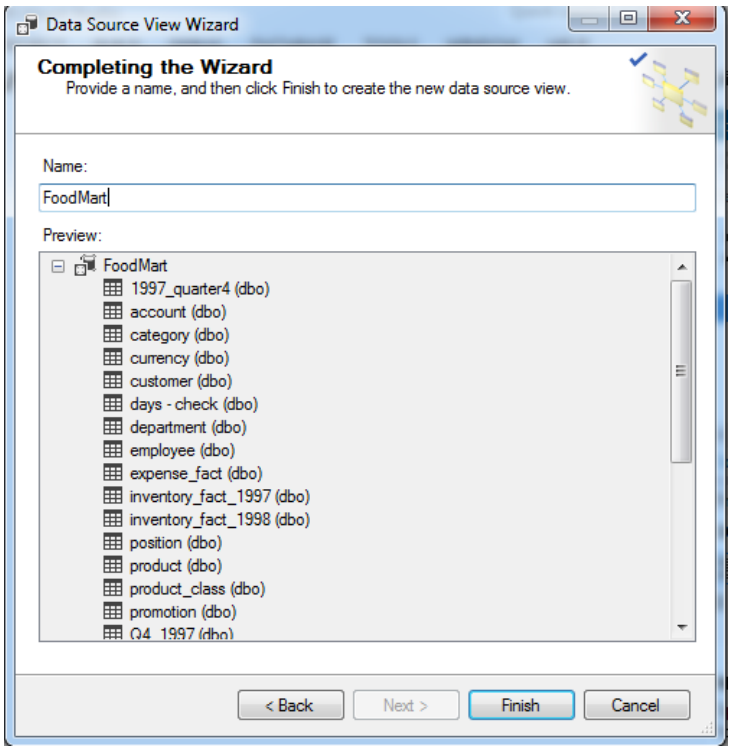
Εικόνα 6.63

11. Στην Εικόνα 6.64 απεικονίζονται όλοι οι πίνακες που είναι διαθέσιμοι για να εισαχθούν. Για να εισάγουμε όλους τους πίνακες, επιλέγουμε >> .



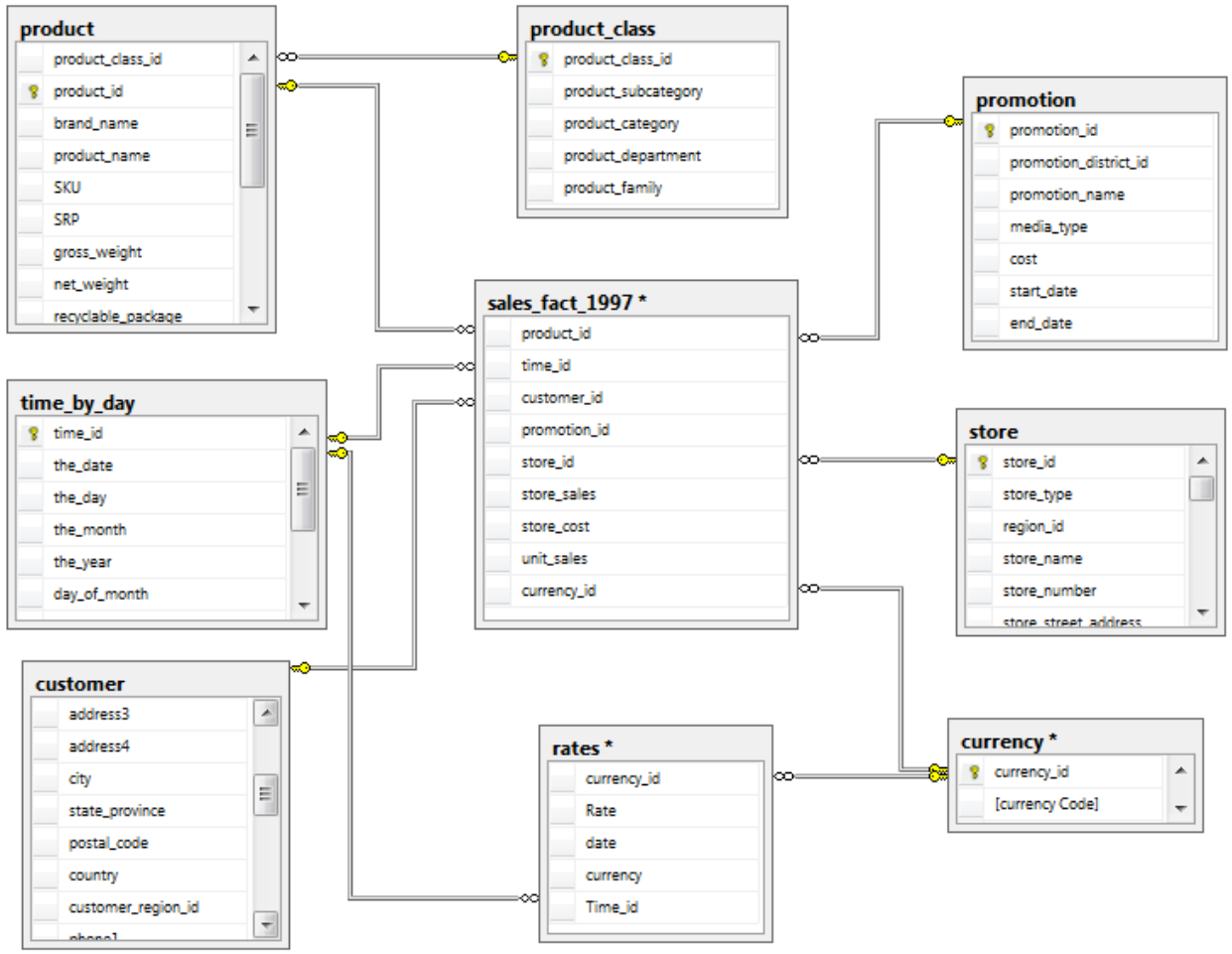
Εικόνα 6.64

12. Σ' αυτό το στάδιο ορίζουμε όνομα στο Data Source View. Στη συγκεκριμένη περίπτωση, όπως φαίνεται στην Εικόνα 6.65, το ονομάζουμε FoodMart. Στη συνέχεια, επιλέγουμε Finish, ώστε να ολοκληρωθεί η διαδικασία.



Εικόνα 6.65

13. Στη συνέχεια, όπως φαίνεται στην Εικόνα 6.66, εμφανίζονται οι συσχετίσεις μεταξύ των πινάκων της βάσης που θα ασχοληθούμε. Όπως παρατηρούμε, ο κεντρικός πίνακας είναι ο Sales_fact_1997, στον οποίο καταλήγουν έξι ξένα κλειδιά (product_id, time_id, customer_id, promotion_id, store_id και currency_id), τα οποία είναι και πρωτεύοντα κλειδιά σε έξι αντίστοιχους πίνακες (Product, time_by_day, customer, promotion, store, currency). Ο πίνακας sales_fact_1997 είναι ένας κύβος πωλήσεων που αφορά τις πωλήσεις και το κόστος πωλήσεων (store_sales και store_cost) ανά κατάστημα. Τονίζουμε ότι οι πίνακες της βάσης δεδομένων FoodMart είναι πολύ περισσότεροι. Εμείς εστιάζουμε μόνο στους παρακάτω επιλεγμένους πίνακες της βάσης δεδομένων FoodMart, για να γίνουν κατανοητοί από τον αναγνώστη, καθώς στο κεφάλαιο 11 θα ασχοληθούμε με αυτούς διεξοδικότερα.



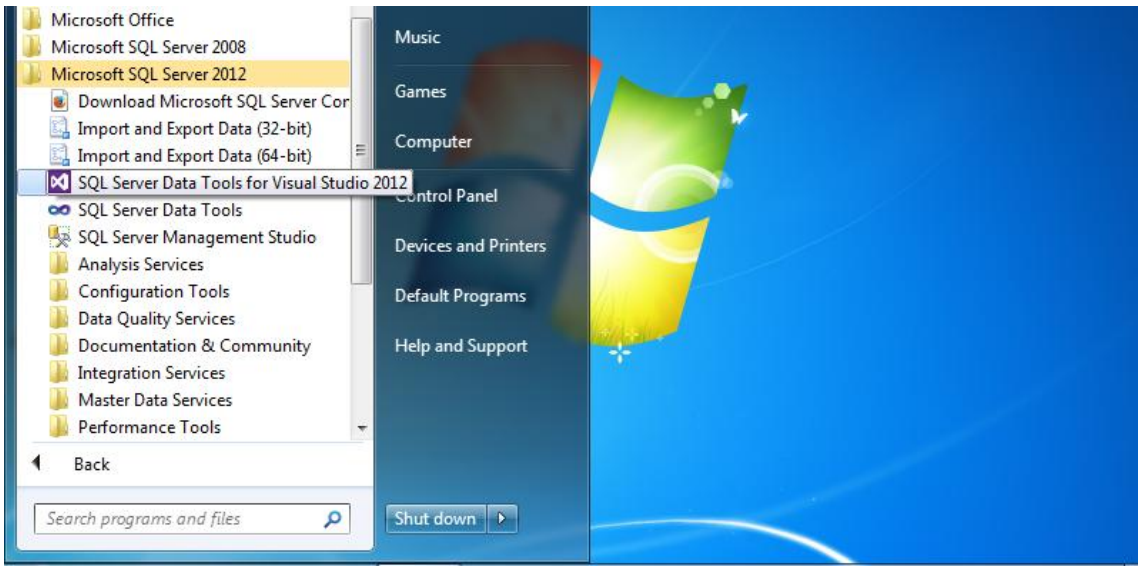
Εικόνα 6.66

6.6. Επεξεργασία βάσης AdventureWorks

Σ' αυτήν την ενότητα θα επεξεργαστούμε τη βάση δεδομένων AdventureWorksDW2008R2 που έχουμε δημιουργήσει στον SQL Server χρησιμοποιώντας το περιβάλλον SQL Server Data Tools for Visual Studio 2012.

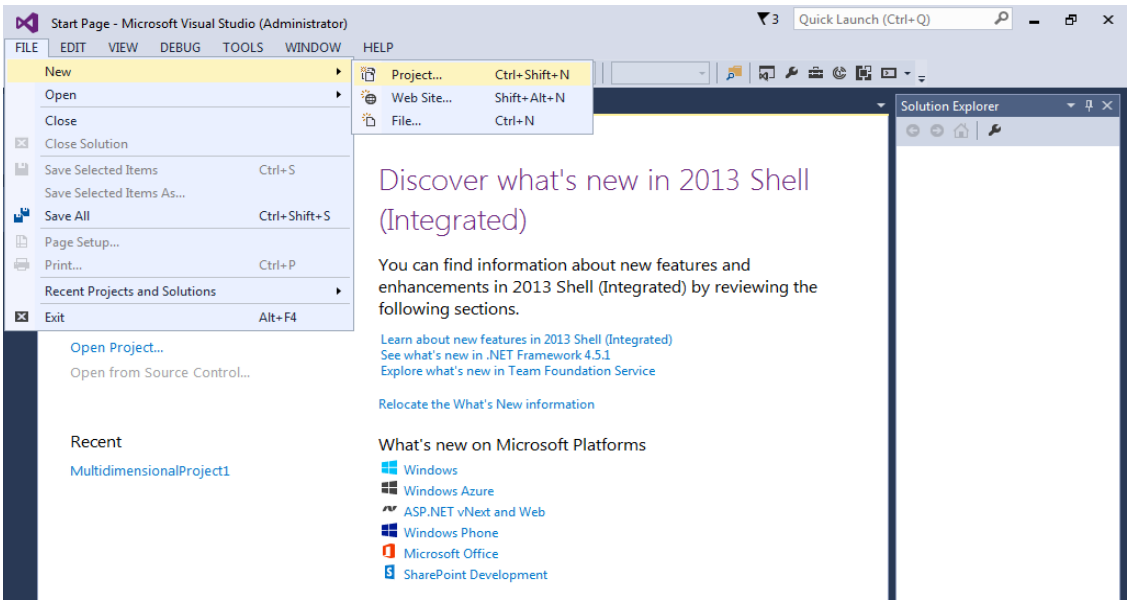
Αναλυτικά βήματα

1. Στο περιβάλλον των Windows ακολουθούμε τη διαδρομή Έναρξη ► Όλα τα Προγράμματα και, από τον φάκελο Microsoft SQL Visual Studio 2012, επιλέγουμε SQL Server Data Tools for Visual Studio 2012, όπως φαίνεται στην Εικόνα 6.67.



Εικόνα 6.67

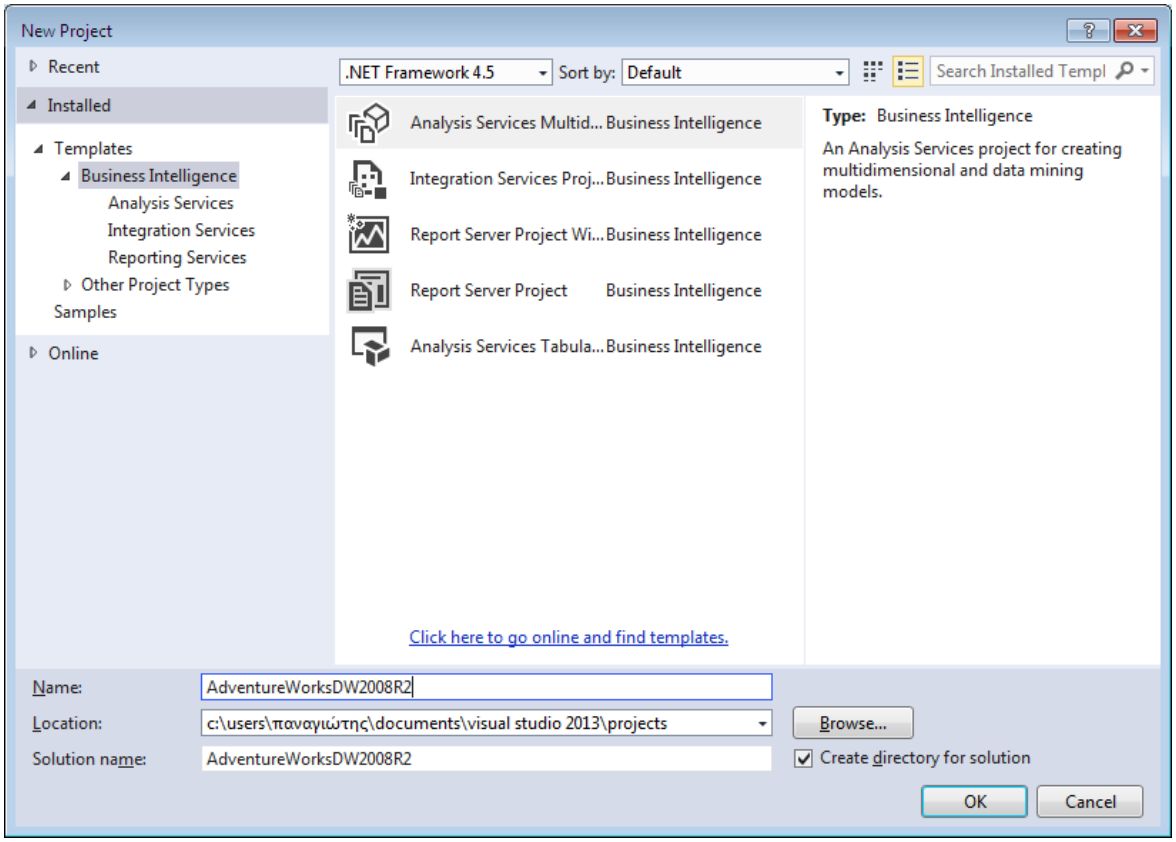
2. Στο νέο παράθυρο, όπως φαίνεται στην Εικόνα 6.68, δημιουργούμε ένα νέο project στο οποίο θα εισάγουμε τα δεδομένα από τη βάση **AdventureWorksDW2008R2** που έχουμε δημιουργήσει στον **SQL Server Management Studio**. Επιλέγουμε, λοιπόν, αρχικά New ► Project.



Εικόνα 6.68

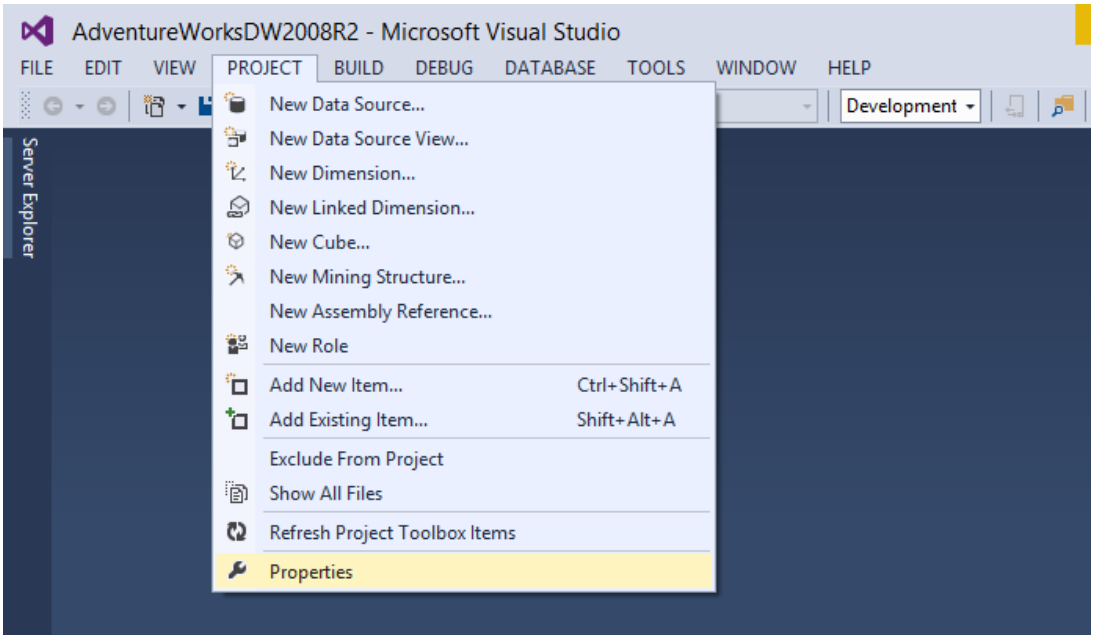
3. Στον οδηγό που εμφανίζεται, επιλέγουμε Business Intelligence Projects ► **Analysis Services Multidimensional and Data Mining Project**. Στη συνέχεια, συμπληρώνουμε τα στοιχεία του project με τον τρόπο που φαίνεται στην Εικόνα 6.69. Πιο συγκεκριμένα:

   - Στο πεδίο **Name** δίνουμε το όνομα του **Project.** Στη συγκεκριμένη περίπτωση δίνουμε το όνομα **AdventureWorksDW2008R2**.
   - Στο πεδίο **Location** εισάγουμε τον προορισμό όπου θα αποθηκεύεται το **project**. Στη συγκεκριμένη περίπτωση είναι καθορισμένος ο προεπιλεγμένος προορισμός.
   - Στο πεδίο **Solution name** δίνουμε όνομα στο Solution που θα περιέχει το Project μας. Ένα Solution μπορεί να περιέχει πολλά projects που έχουν κάποια σχέση μεταξύ τους. Στη συγκεκριμένη περίπτωση δίνεται το όνομα **AdventureWorksDW2008R2** αυτόματα.
   - Επιλέγουμε το **Create directory for solution** και κάνουμε κλικ στο OK, ώστε να δημιουργηθεί το project.



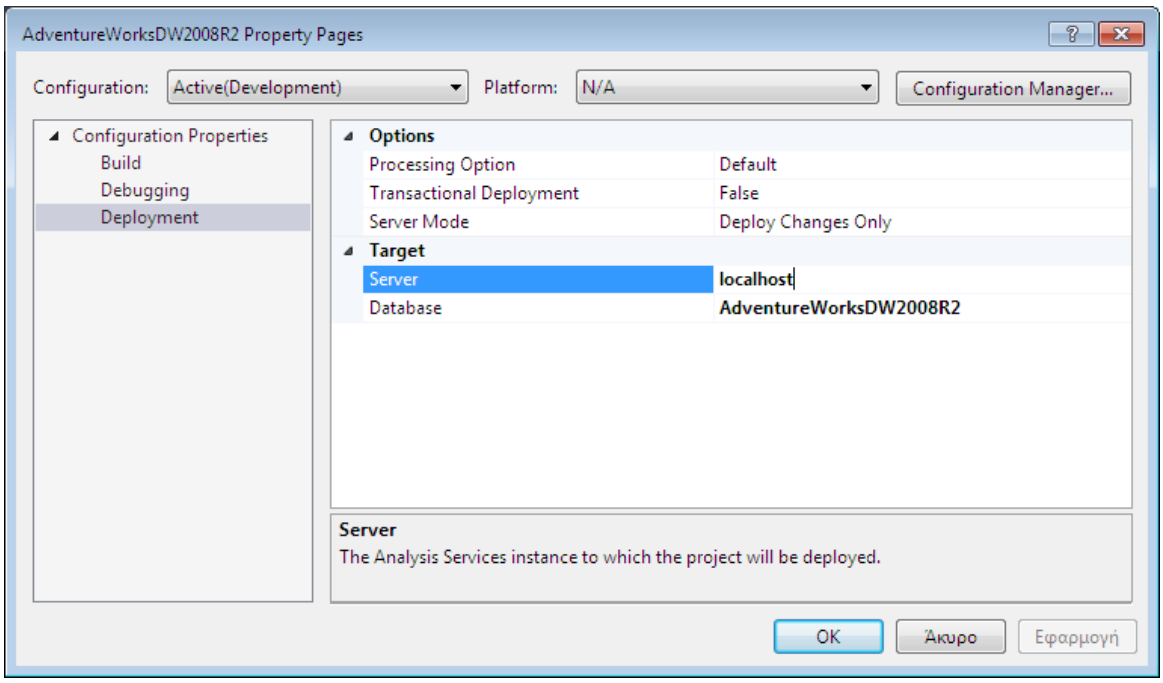
Εικόνα 6.69

4. Επιστρέφουμε στο **Visual Studio** και επιλέγουμε με κλικ το **Project** και, έπειτα, το **Properties,** όπως φαίνεται στην εικονα 6.70.



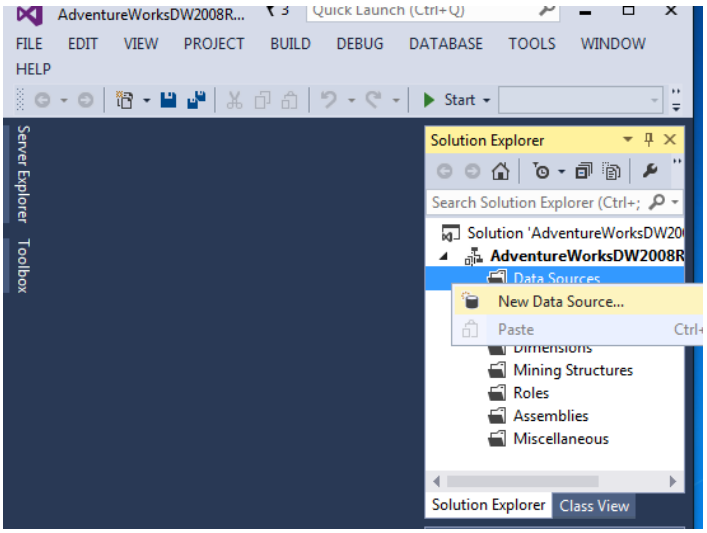
Εικόνα 6.70

5. Στην αριστερή στήλη, όπως φαίνεται στην Εικόνα 6.71, κάνουμε κλικ στο **Deployment** και αλλάζουμε το όνομα του **Server** σε **localhost**, ή εναλλακτικά, στο όνομα που έχουμε δώσει στο instance που δημιουργήσαμε κατά την εγκατάσταση του SQL. Πατάμε OK (Εικόνα 6.71).



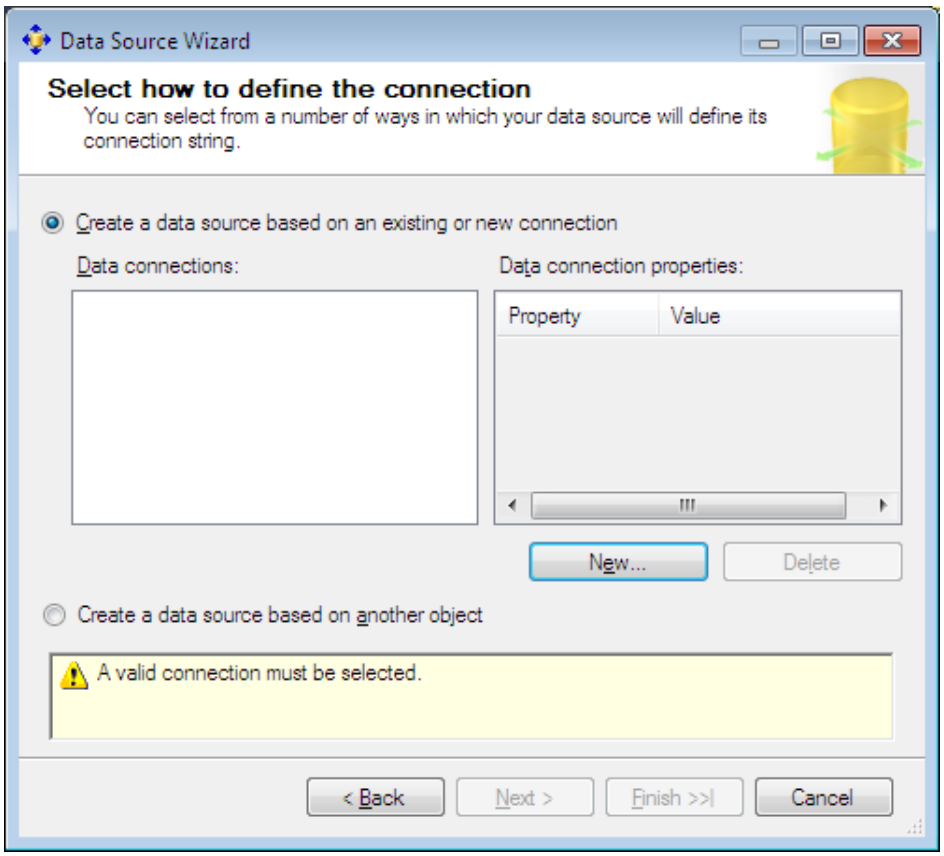
Εικόνα 6.71

6. Στην καρτέλα **Solution Explorer** επιλέγουμε **AdventureWorksDW2008R2**, και, όπως φαίνεται στην Εικόνα 6.72, κάνουμε δεξί κλικ στο Data Sources και επιλέγουμε **New Data Source.**



Εικόνα 6.72

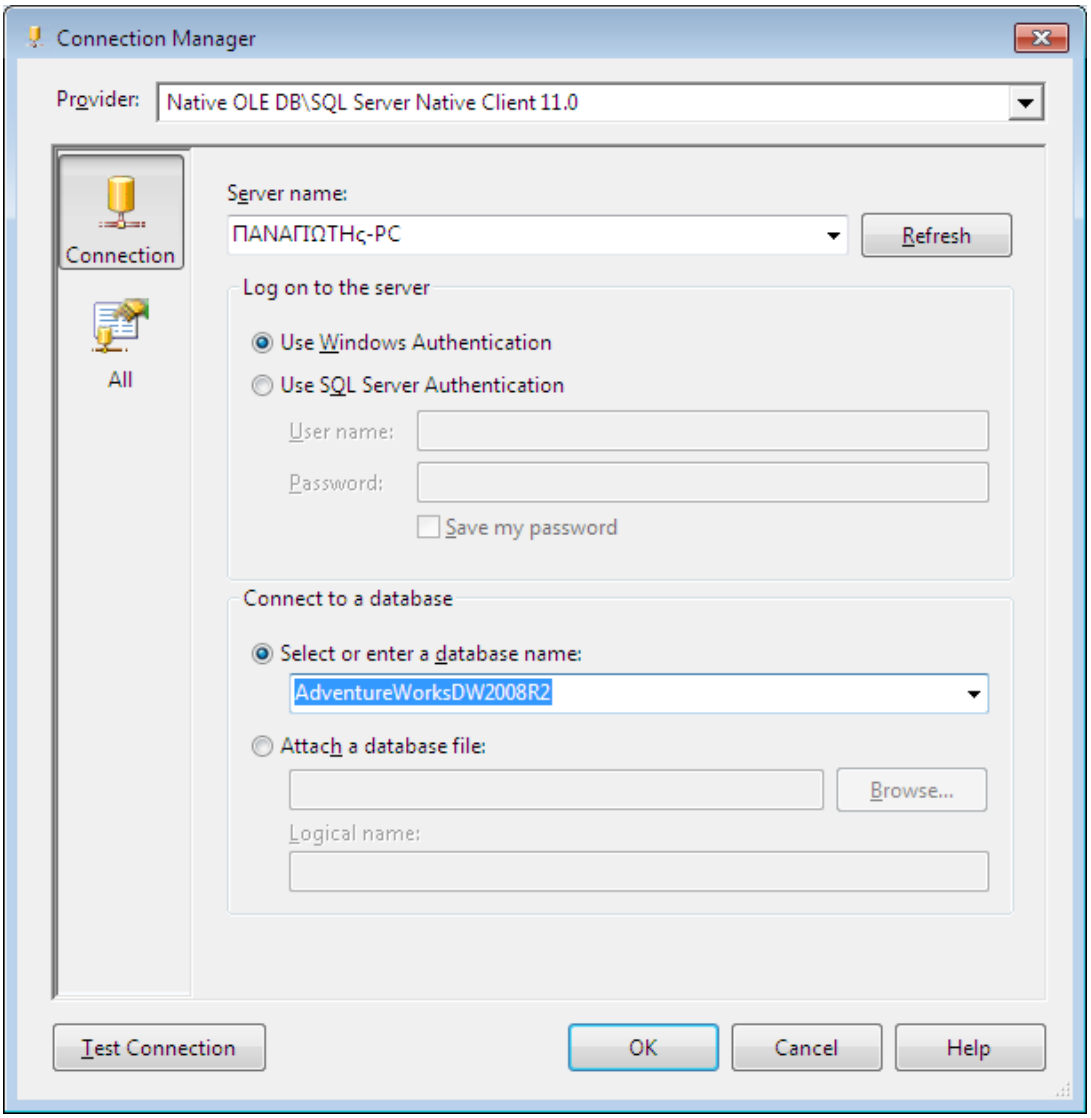
7. Σ' αυτό το βήμα πρέπει να επιλέξουμε αν θα δημιουργήσουμε μια νέα σύνδεση με έναν διακομιστή, για να δημιουργήσουμε το Data Source, ή αν θα επιλέξουμε μια ήδη υπάρχουσα σύνδεση. Στη συγκεκριμένη περίπτωση επιλέγουμε **Create a data source based on an existing or new connection** και, στη συνέχεια, επιλέγουμε **New…,** όπως φαίνεται στην Εικόνα 6.73.



Εικόνα 6.73

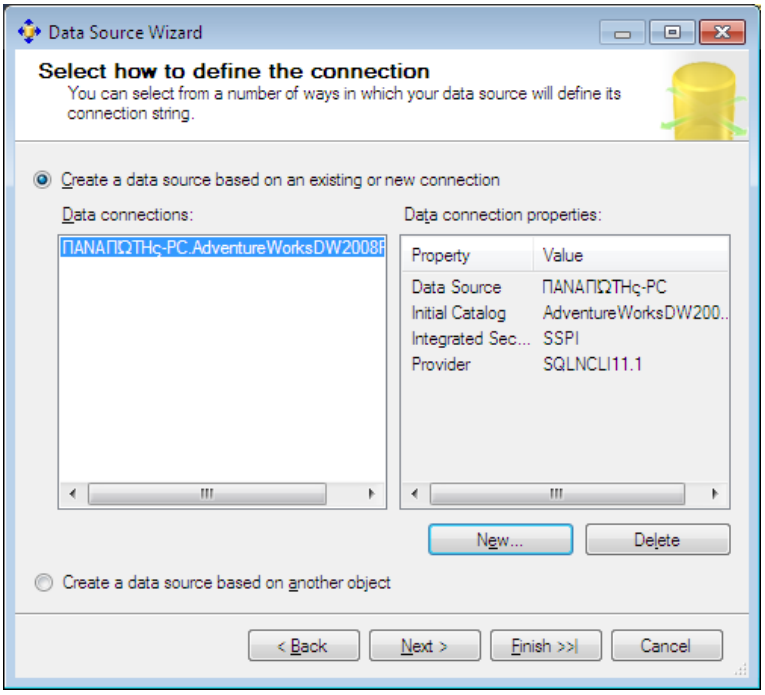
8. Στον διαχειριστή σύνδεσης που εμφανίζεται, όπως φαίνεται στην Εικόνα 6.74, συμπληρώνουμε τα στοιχεία ως εξής:

   - Στο πεδίο **Provider** επιλέγουμε **Native OLE DB/SQL Native Client.**
   - Στο πεδίο **Server name** συμπληρώνουμε **localhost** ή το όνομα του instance.
   - Επιλέγουμε **Use Windows Authentication.**
   - Στο πεδίο **Connect to a database** επιλέγουμε **Select or enter a database name,** αφού έχουμε ήδη δημιουργήσει τη βάση, και, στη συνέχεια, επιλέγουμε τη βάση **AdventureWorksDW2008R2.**



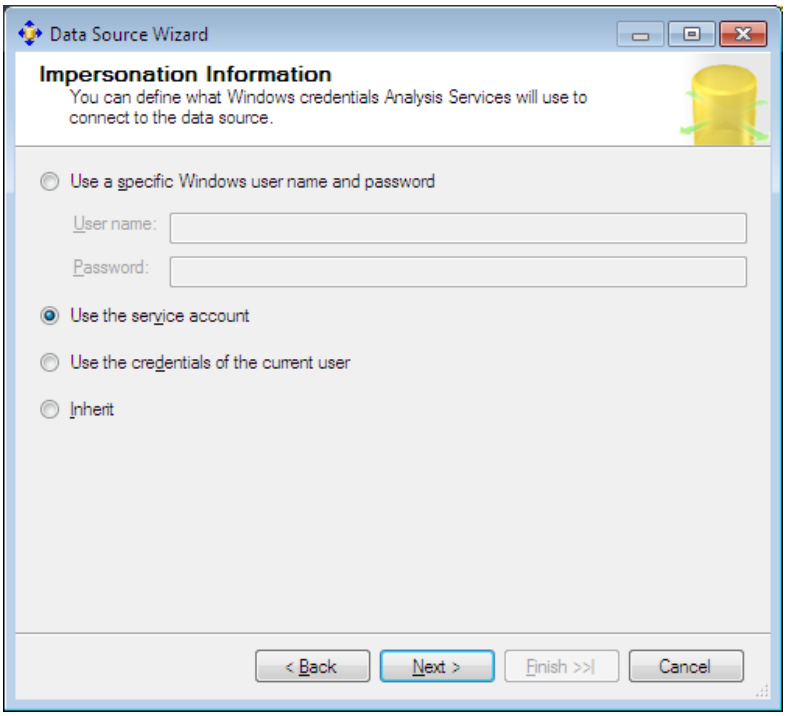
Εικόνα 6.74

9. Επιστρέφοντας στο προηγούμενο παράθυρο, όπως φαίνεται στην Εικόνα 6.75, στο πεδίο Data connections βλέπουμε τη σύνδεση **localhost.AdventureWorksDW2008R2** ή την αντίστοιχη σύνδεση με το όνομα του υπολογιστή. Στη συνέχεια, επιλέγουμε **Next,** για να προχωρήσουμε στο επόμενο βήμα.



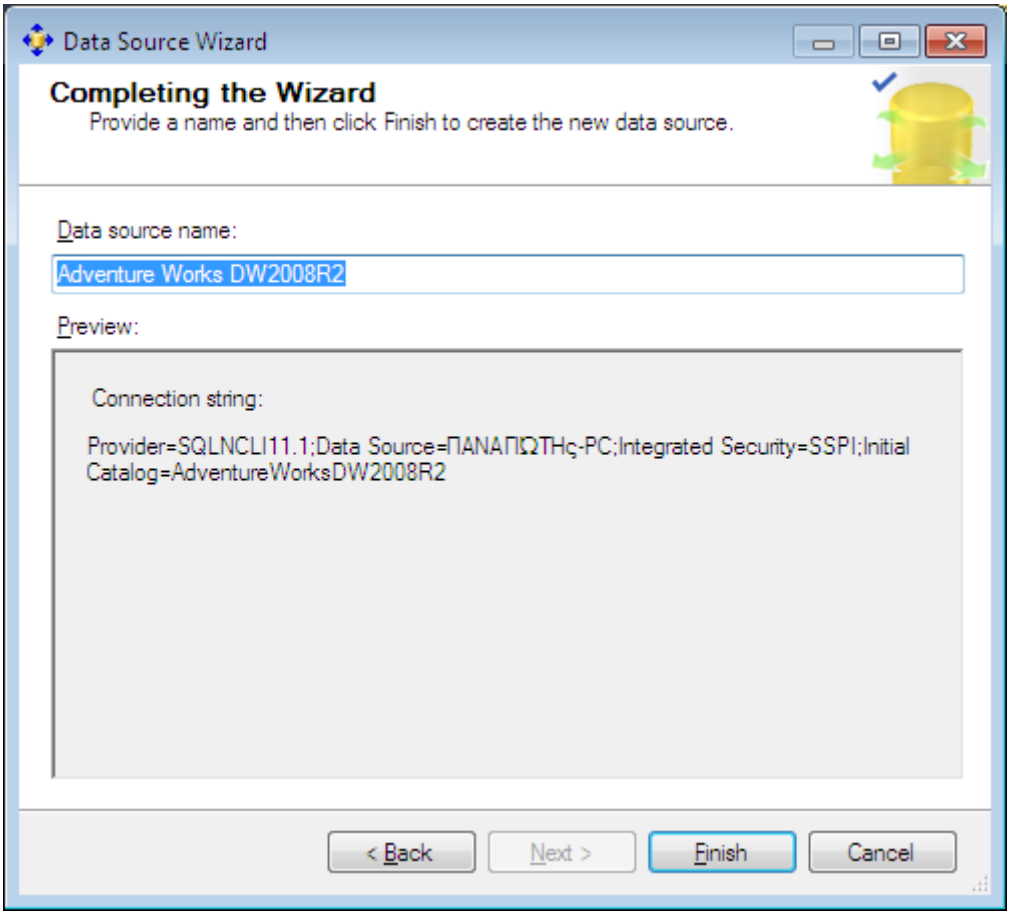
Εικόνα 6.75

10. Στο νέο παράθυρο, όπως φαίνεται στην Εικόνα 6.76, επιλέγουμε **Use the service account,** αφού δεν θέλουμε να ορίσουμε κάποιο άλλο **username** και **password** στο **data source.** Στη συνέχεια, επιλέγουμε **Next**.



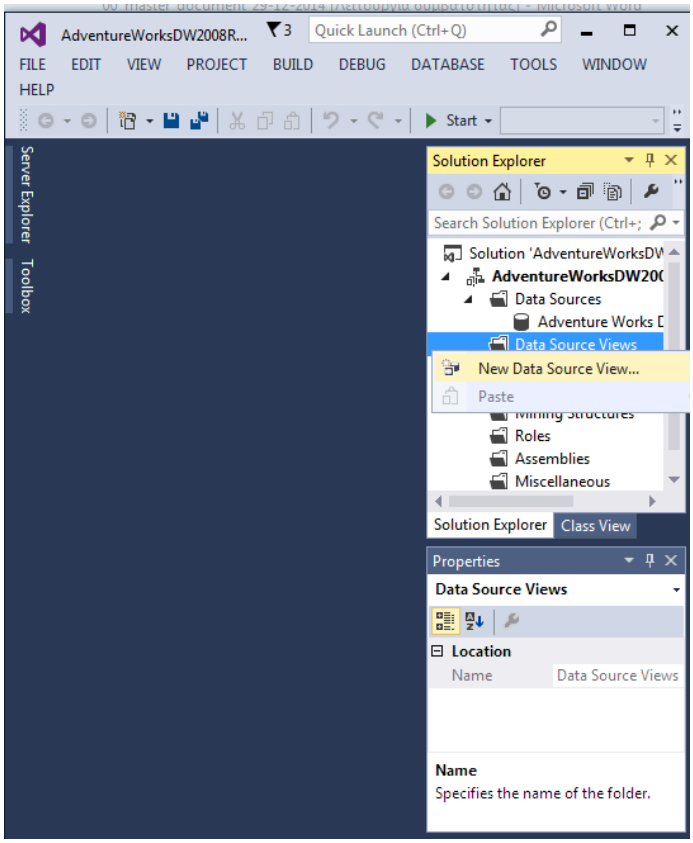
Εικόνα 6.76

11. Σ’ αυτό το βήμα ορίζουμε όνομα στο **DataSource**, Στη συγκεκριμένη περίπτωση**,** όπως φαίνεται στην Εικόνα 6.77, συμπληρώνουμε το όνομα **AdventureWorksDW2008R2** και, στη συνέχεια, επιλέγουμε **Finish,** ώστε να δημιουργηθεί το DataSource.



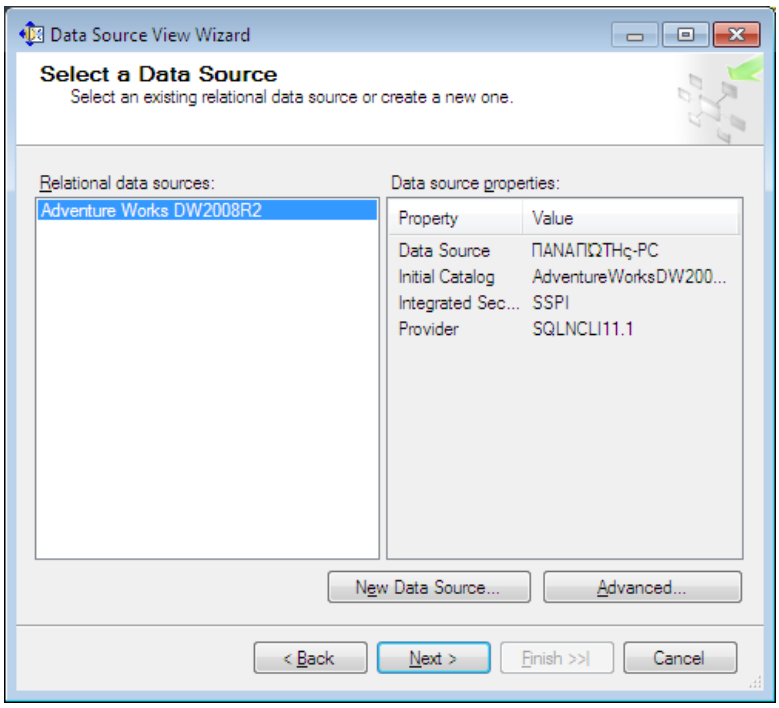
Εικόνα 6.77

12. Στο Data Sources του **AdventureWorksDW2008R2** βλέπουμε ότι έχει δημιουργηθεί το **AdventureWorksDW2008R2 source.ds**. Στη συνέχεια, θα δημιουργήσουμε ένα **Data Source View** που θα έχει τα δεδομένα του **AdventureWorksDW2008R2 source.ds** και θα μας προσφέρει τη γραφική αναπαράσταση της βάσης που έχουμε συνδέσει με το **AdventureWorksDW2008R2 source.ds**. Επιλέγουμε, λοιπόν, την καρτέλα **Solution Explorer και,** όπως φαίνεται στην Εικόνα 6.78, κάνουμε δεξί κλικ στο Data Source Views και επιλέγουμε **New Data Source View**.



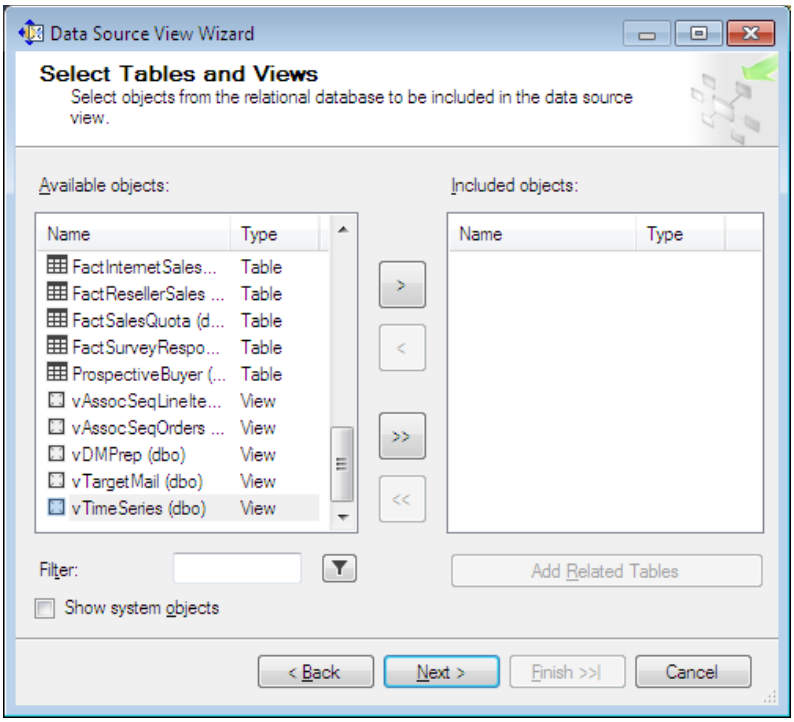
Εικόνα 6.78

13. Στο παράθυρο που εμφανίζεται, όπως φαίνεται στην Εικόνα 6.79, επιλέγουμε το **Data Source** με το οποίο θα συσχετίσουμε το Data Source View που θέλουμε να δημιουργήσουμε. Στη συγκεκριμένη περίπτωση, επιλέγουμε το **Adventure Works2014 source** και, στη συνέχεια, επιλέγουμε **Next,** ώστε να προχωρήσουμε στο επόμενο βήμα.



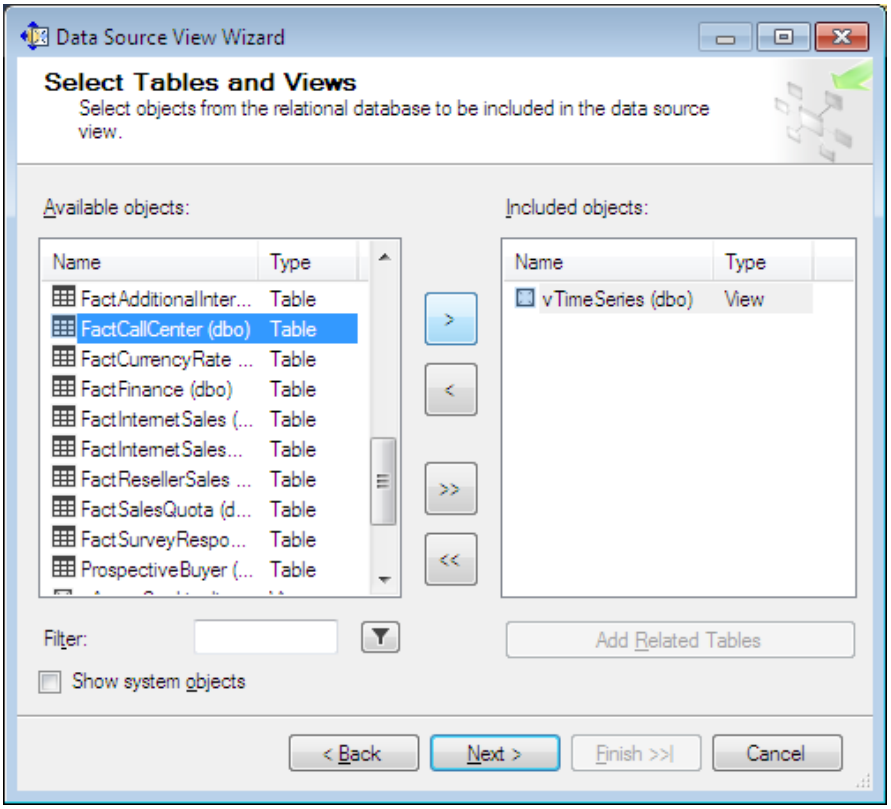
Εικόνα 6.79

14. Στο νέο παράθυρο, όπως φαίνεται στην Εικόνα 6.80, εμφανίζονται όλοι οι πίνακες που είναι διαθέσιμοι για να εισαχθούν. Παρατηρούμε ότι εμπεριέχονται όχι μόνο πίνακες (tables) αλλά και όψεις (views).



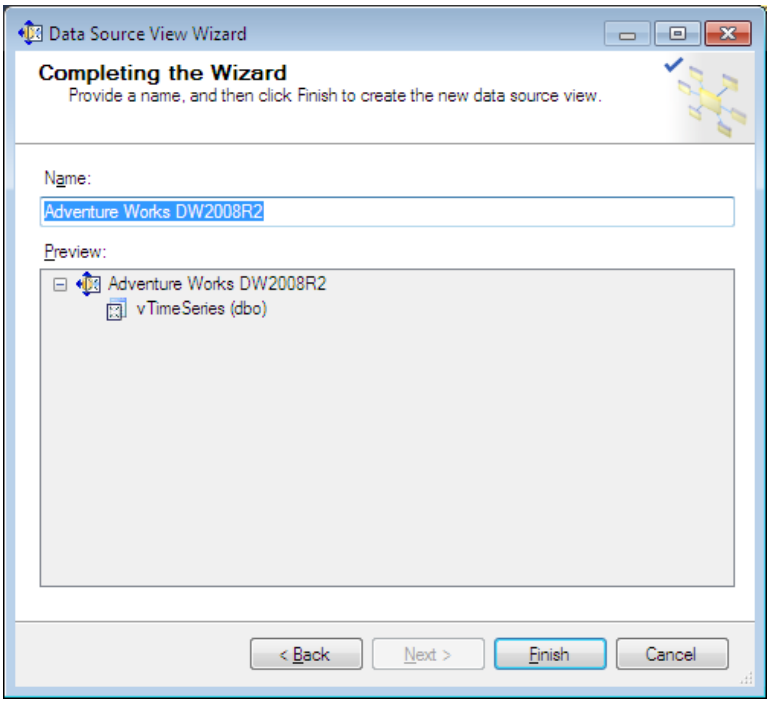
Εικόνα 6.80

15. Η όψη που πρέπει να εισαχθεί ονομάζεται **vTimeSeries.** Στη συνέχεια**,** όπως φαίνεται στην Εικόνα 6.81, επιλέγουμε Next, ώστε να προχωρήσουμε στο επόμενο βήμα.



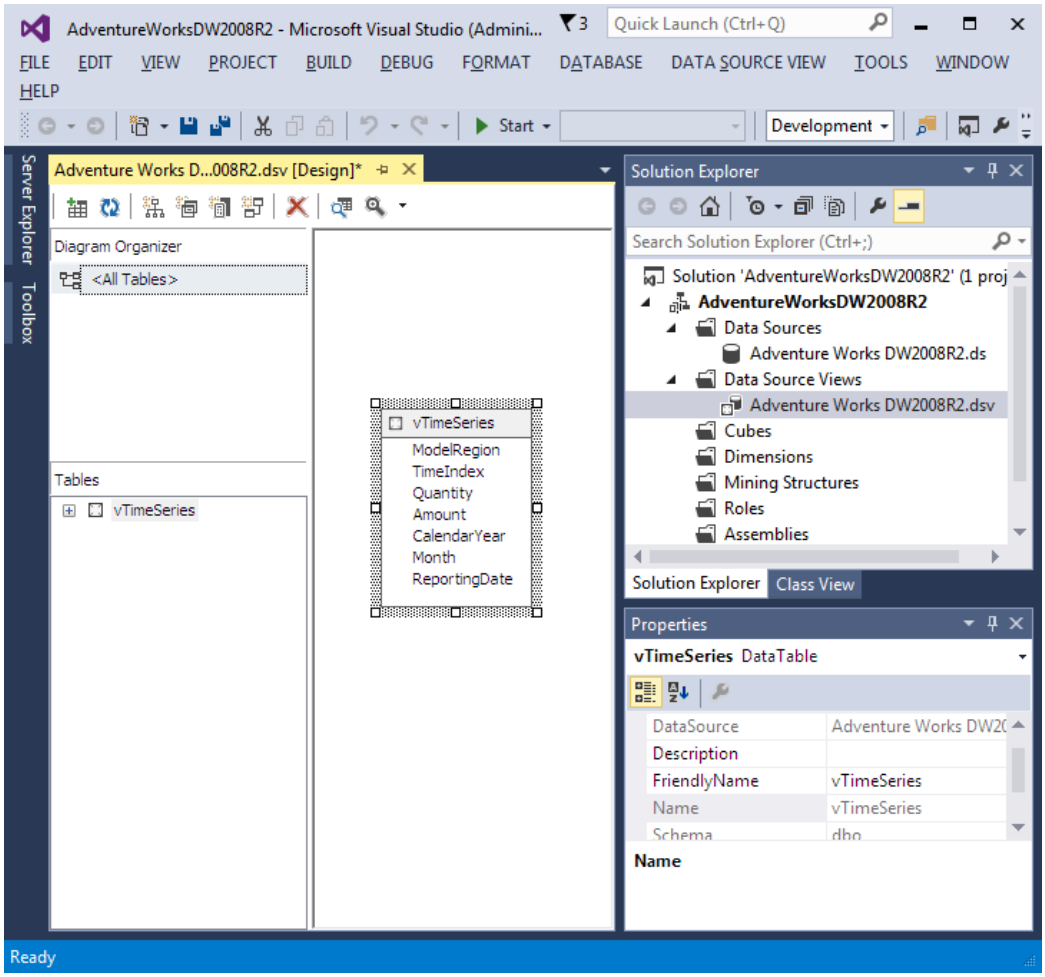
Εικόνα 6.81

16. Σ' αυτό το στάδιο, όπως φαίνεται στην Εικόνα 6.82, ορίζουμε όνομα στο **Data Source View**. Στη συγκεκριμένη περίπτωση το ονομάζουμε **AdventureWorksDW2008R2**. Στη συνέχεια, επιλέγουμε **Finish,** ώστε να ολοκληρωθεί η διαδικασία.



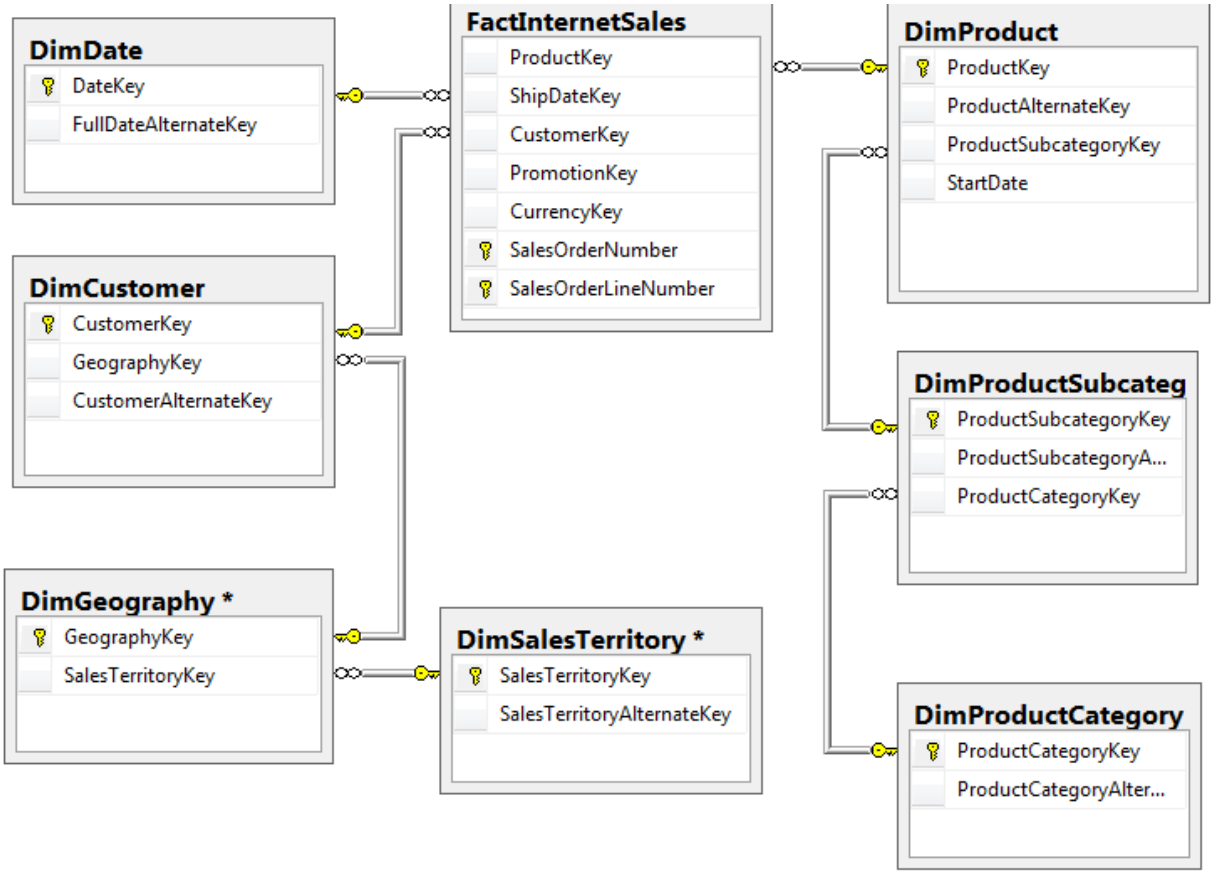
Εικόνα 6.82

17. Τέλος, εμφανίζεται το παράθυρο που περιέχει το διάγραμμα με την όψη vTimeSeries της βάσης μας, όπως φαίνεται στην Εικόνα 6.83.



Εικόνα 6.83

18. Θα πρέπει να τονιστεί ότι η παραπάνω όψη (view) έχει προκύψει μετά από κατάλληλο ερώτημα της βάσης δεδομένων **AdventureWorksDW2008R2**. Για την αμεσότερη κατανόηση των βασικών χαρακτηριστικών της παραπάνω βάσης δεδομένων, παρουσιάζουμε, όπως φαίνεται στην Εικόνα 6.84, ένα μέρος του σχεσιακού σχήματος της βάσης δεδομένων **AdventureWorksDW2008R2**.



Εικόνα 6.84

6.7. Ασκήσεις

1. Να κατεβάσετε από τον δικτυακό τόπο http://delab.csd.auth.gr/~symeon/courses/dm/index.htm τη βάση δεδομένων FoodMart και να την εγκαταστήσετε στο SQL Server. Να εκχωρήσετε, επίσης, τα κατάλληλα δικαιώματα πρόσβασης στο όνομά σας, προκειμένου να μπορείτε να τρέξετε αλγορίθμους εξόρυξης δεδομένων στο περιβάλλον του Data Tools του Visual Studio.

2. Να δημιουργήσετε ένα project και ένα Data Source View για τη βάση δεδομένων FoodMart στο περιβάλλον του Data Tools του Visual Studio.

3. Να κατεβάσετε από τον δικτυακό τόπο http://delab.csd.auth.gr/~symeon/courses/dm/index.htm τη βάση δεδομένων AdventureWorksDW2008R2 και να την εγκαταστήσετε στο SQL Server. Να εκχωρήσετε, επίσης, τα κατάλληλα δικαιώματα πρόσβασης στο όνομά σας, προκειμένου να μπορείτε να τρέξετε αλγορίθμους εξόρυξης δεδομένων στο περιβάλλον του Data Tools του Visual Studio.

4. Να δημιουργήσετε ένα project και ένα Data Source View για τη βάση δεδομένων AdventureWorksDW2008R2 στο περιβάλλον του Data Tools του Visual Studio.

5. Να κατεβάσετε από τον δικτυακό τόπο http://delab.csd.auth.gr/~symeon/courses/dm/index.htm τη βάση δεδομένων MovieClick και να την εγκαταστήσετε στο SQL Server. Να εκχωρήσετε, επίσης, τα κατάλληλα δικαιώματα πρόσβασης στο όνομά σας, προκειμένου να μπορείτε να τρέξετε αλγορίθμους εξόρυξης δεδομένων στο περιβάλλον του Data Tools του Visual Studio.

6. Να δημιουργήσετε ένα project και ένα Data Source View για την βάση δεδομένων MovieClick στο περιβάλλον του Data Tools του Visual Studio.

Κεφάλαιο 7. Κατηγοριοποίηση Δεδομένων με Δέντρα Απόφασης

Σύνοψη
Σ' αυτό το κεφάλαιο θα παρουσιάσουμε τα δέντρα αποφάσεων που αποτελούν την πιο δημοφιλή τεχνική εξόρυξης δεδομένων. Διαχωρίζουν τα δεδομένα σε αναδρομικά υποσύνολα, έτσι ώστε το κάθε υποσύνολο που προκύπτει να περιέχει στοιχεία με μεγαλύτερη ή μικρότερη ομοιογένεια σε σχέση με τον τελικό στόχο.

7.1. Θεωρητικό υπόβαθρο των αλγορίθμων κατηγοριοποίησης του SQL Server

Με τον όρο **κατηγοριοποίηση (classification)** προσδιορίζουμε την πράξη της ανάθεσης ενός αντικειμένου σε μία από τις κλάσεις ενός προκαθορισμένου συνόλου κλάσεων (Νανόπουλος, & Μανωλόπουλος, 2008· Χαλκίδη, & Βεζυργιάννης, 2005). Κάθε αντικείμενο ενός συνόλου δεδομένων διαθέτει έναν αριθμό χαρακτηριστικών ($X_1,\ldots,X_\kappa$), όπου $\Pi(X_i)$ είναι το πεδίο ορισμού του χαρακτηριστικού X_i. Επιπλέον, κάθε αντικείμενο έχει ένα χαρακτηριστικό C, το οποίο δηλώνει την κλάση όπου αυτό ανήκει, με το Π(C) να συμβολίζει το πεδίο ορισμού του χαρακτηριστικού της κλάσης C. Η κατηγοριοποίηση περιλαμβάνει την εξεύρεση μιας συνάρτησης $f: \Pi(X_1) \times \ldots \times \Pi(X_\kappa) \rightarrow \Pi(C)$, η οποία ονομάζεται **μοντέλο κατηγοριοποίησης** (classification model). Αν γνωρίζουμε τις τιμές των χαρακτηριστικών $X_1,\ldots,X_\kappa$ ενός αντικειμένου, αλλά όχι την τιμή του χαρακτηριστικού C, τότε εφαρμόζουμε ένα μοντέλο κατηγοριοποίησης και αναθέτουμε το αντικείμενο στην κλάση $f(X_1,\ldots, X_\kappa)$.

Τα δέντρα απόφασης (decision trees) είναι από τα πιο γνωστά μοντέλα κατηγοριοποίησης. Το δέντρο απόφασης είναι ένας γράφος με την κλασική δενδρική δομή (Νανόπουλος, & Μανωλόπουλος, 2008· Χαλκίδη, & Βεζυργιάννης, 2005), όπου διακρίνουμε: (α) έναν αρχικό κόμβο, τη ρίζα, (β) τους εσωτερικούς κόμβους και (γ) τους εξωτερικούς κόμβους, τα φύλλα. Σε κάθε κόμβο (εσωτερικό ή εξωτερικό) εκτός της ρίζας εισέρχεται μια κατευθυνόμενη ακμή από έναν άλλο κόμβο. Σε κάθε εσωτερικό κόμβο αντιστοιχεί ένα χαρακτηριστικό που χρησιμοποιείται για περαιτέρω διαχωρισμό του δέντρου. Στις ακμές που εξέρχονται από τη ρίζα ή κάθε εσωτερικό κόμβο, αντιστοιχεί μια συνθήκη ελέγχου με βάση το διαχωριστικό χαρακτηριστικό. Η διαδικασία κατασκευής ενός δέντρου απόφασης είναι επαναληπτική και μπορεί να περιγραφεί συνοπτικά ως ακολούθως: Αρχικά, επιλέγουμε ένα χαρακτηριστικό, το οποίο αναφέρεται στη ρίζα του δέντρου, και, στη συνέχεια, κατασκευάζουμε μια ακμή και έναν κόμβο για καθεμία από τις διακριτές τιμές του χαρακτηριστικού. Αυτά τα δύο βήματα επαναλαμβάνονται συνεχώς, μέχρις ότου όλα τα χαρακτηριστικά να εισαχθούν στους κόμβους του δέντρου.

Τονίζεται ότι στο περιβάλλον του SQL Server υπάρχει σχετική παράμετρος (**Split method**), η οποία καθορίζει τη μέθοδο με την οποία θα διαχωρίζονται κάθε φορά οι κόμβοι ενός δέντρου. Ο διαχωρισμός μπορεί να γίνει είτε με δυαδικό τρόπο (binary) είτε με περισσότερες από δύο ακμές (complete). Επιπροσθέτως, για την αντιστοίχιση ενός χαρακτηριστικού με έναν κόμβο του δέντρου, λαμβάνεται υπόψη κάθε φορά η πληροφορία που μεταφέρει ένα χαρακτηριστικό. Συγκεκριμένα, το κριτήριο επιλογής ενός διαχωριστικού χαρακτηριστικού βασίζεται στην ομοιογένεια των κόμβων που αυτό παράγει, ώστε να επιλέγεται αυτό που επιφέρει τη μεγαλύτερη ομοιογένεια (δηλαδή να εμφανίζονται σε ένα κόμβο μόνο αντικείμενα της ίδιας κλάσης) στους νέους κόμβους που δημιουργούνται κάθε φορά. Στο περιβάλλον του SQL Server υπάρχει μια συγκεκριμένη παράμετρος (**score_method**), η οποία προσδιορίζει ποια μέθοδος επιλογής διαχωριστικού χαρακτηριστικού θα χρησιμοποιηθεί (π.χ. Εντροπία, ή Bayesian with K2 Prior, ή Bayesian Dirichlet Equivalent with Uniform Prior).

Η **εντροπία (entropy)** μετράει τον βαθμό αβεβαιότητας (ανομοιογένειας των αντικειμένων που εντάσσονται σε ένα κόμβο) που δημιουργεί ένα διαχωριστικό χαρακτηριστικό.

Για τον υπολογισμό της εντροπίας χρησιμοποιείται η παρακάτω εξίσωση:

$$E(N) = -\sum_{i=1}^{c} p_i \log_2 p_i$$

όπου με *N* συμβολίζουμε έναν υπό εξέταση κόμβο και με *c* τον αριθμό των υπαρχουσών κλάσεων. Επίσης, $p_i = n_i / n$, όπου n_i είναι ο αριθμός των αντικειμένων που ανήκουν τόσο στο κόμβο *N* όσο και στην κλάση ***i***, ενώ *n* είναι ο συνολικός αριθμός των αντικειμένων που ανήκουν στον κόμβο *N*.

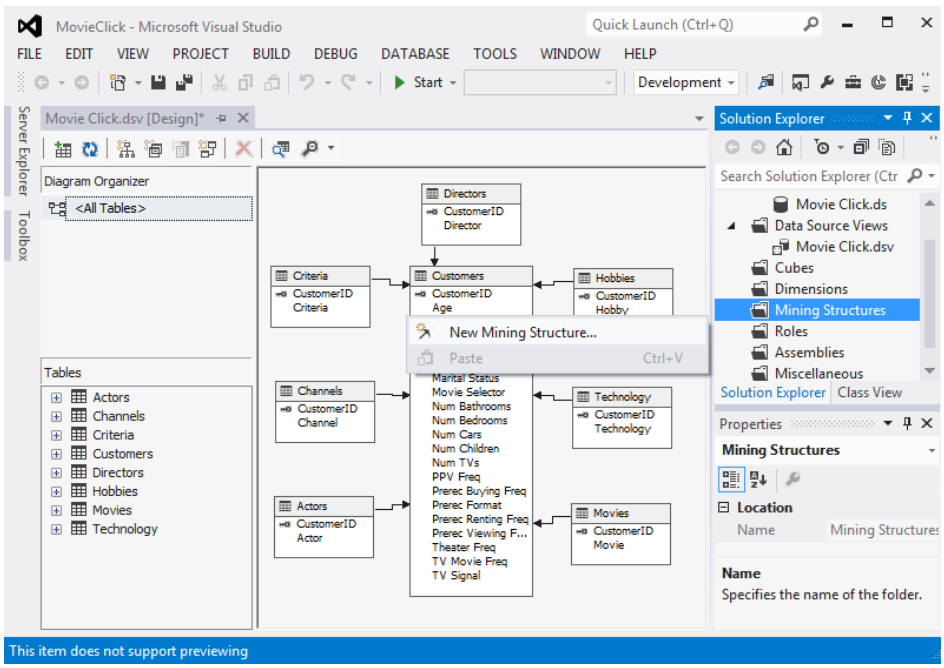
Τέλος, ο SQL Server παρέχει δύο ακόμη μεθόδους επιλογής ενός διαχωριστικού χαρακτηριστικού (Bayesian with K2 Prior και Bayesian Dirichlet Equivalent with Uniform Prior) που βασίζονται στα Bayesian δίκτυα. Το Bayesian δίκτυο είναι ένας κατευθυνόμενος ακυκλικός γράφος, όπου κάθε κόμβος αναπαριστά ένα χαρακτηριστικό, ενώ κάθε κατευθυνόμενη ακμή αναπαριστά μια εξαρτημένη πιθανότητα μετάβασης από έναν κόμβο σε έναν άλλο. Η μέθοδος Bayesian with K2 Prior (BK2) προσπαθεί να δώσει απαντήσεις που αφορούν το ερώτημα ποιος προηγούμενος (prior) κόμβος πρέπει να χρησιμοποιηθεί στον υπολογισμό των πιθανοτήτων μετάβασης για τους επόμενους κόμβους. Η μέθοδος Bayesian Dirichlet Equivalent with Uniform Prior (BDE) βασίζεται στην πολυωνυμική κατανομή Dirichlet, η οποία περιγράφει τη δεσμευμένη πιθανότητα μετάβασης σε έναν κόμβο ενός Bayesian δικτύου βάσει των συνδέσεών του με προηγούμενους κόμβους στο δίκτυο. Ένα κοινό γνώρισμα των δύο παραπάνω μεθόδων είναι ότι όσο πιο κοντά βρίσκεται ένα χαρακτηριστικό στη ρίζα του δέντρου, τόσο πιο σημαντική είναι η πληροφορία σύνδεσής του με το επόμενο χαρακτηριστικό.

7.2. Δημιουργία ενός μοντέλου με δέντρα απόφασης

Έστω ότι ένα Video Club, το οποίο τηρεί τη βάση δεδομένων MovieClick (βλέπε ενότητα 6.1) και για διαφημιστικούς λόγους, θέλει να προβλέψει το φύλλο των πελατών του με βάση την ηλικία τους, το μορφωτικό τους επίπεδο, την οικογενειακή τους κατάσταση και τα ενδιαφέροντά τους. Θα εξετάσουμε αυτό το παράδειγμα με τη χρήση του SQL Server Data Tools του Visual Studio.

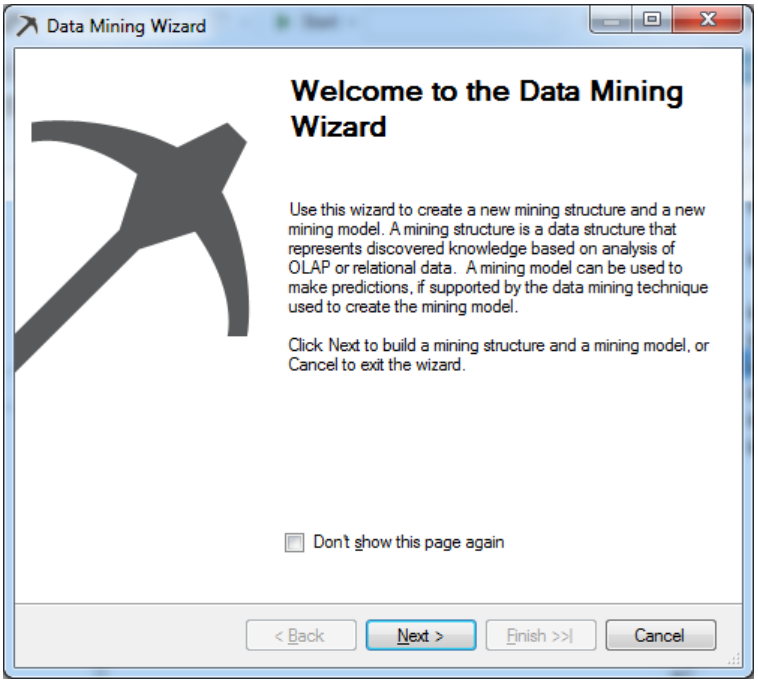
Αναλυτικά βήματα

1. Στην καρτέλα Solution Explorer κάνουμε δεξί κλικ στο Mining Structure και επιλέγουμε New Mining Structure, όπως φαίνεται στην Εικόνα 7.1.



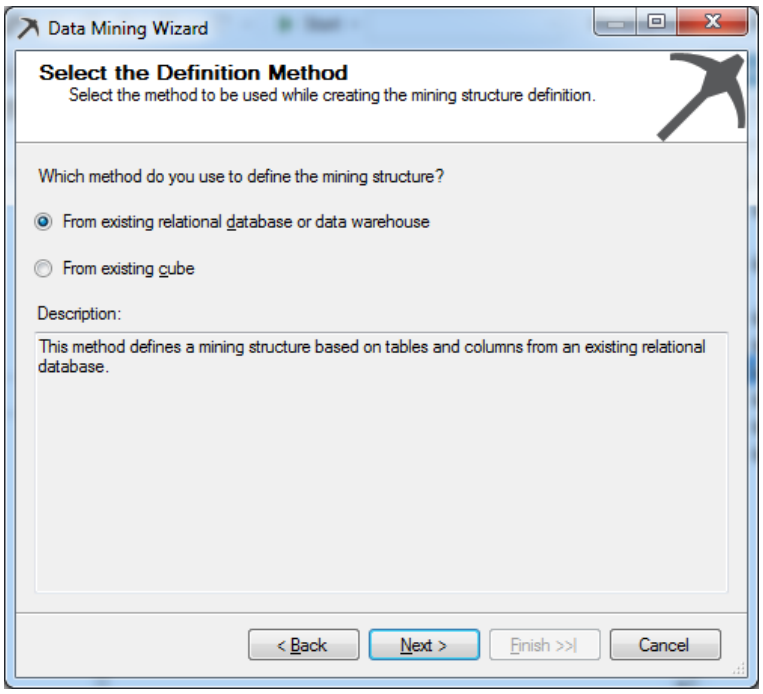
Εικόνα 7.1

2. Στο παράθυρο καλωσορίσματος του οδηγού Data Mining Wizard, όπως φαίνεται στην Εικόνα 7.2, επιλέγουμε Next>, ώστε να προχωρήσουμε στο επόμενο βήμα.



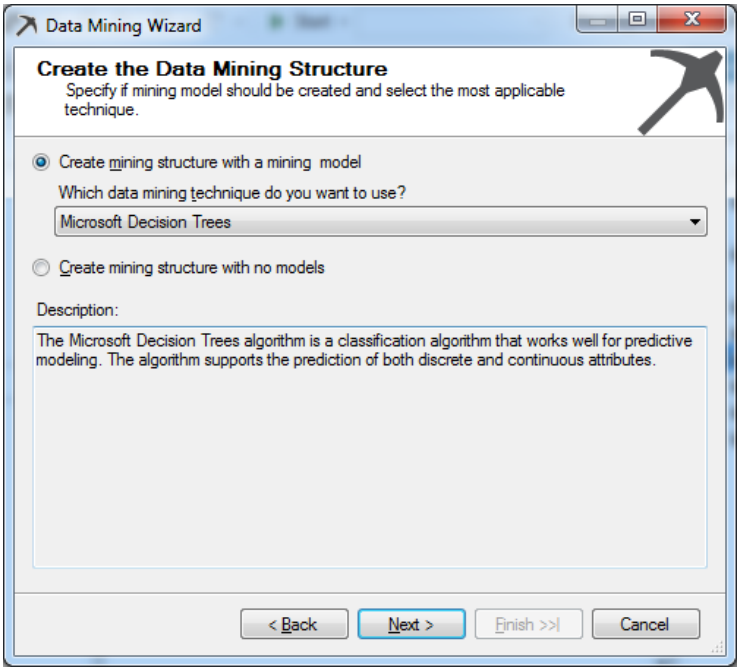
Εικόνα 7.2

3. Στο παράθυρο που εμφανίζεται, όπως φαίνεται στην Εικόνα 7.3, επιλέγουμε From existing relational database or data warehouse, καθώς τα δεδομένα μας θα εισαχθούν από την σχεσιακή βάση που εισάγαμε προηγουμένως στο project μας. Στη συνέχεια, επιλέγουμε Next>, ώστε να προχωρήσουμε στο επόμενο βήμα.



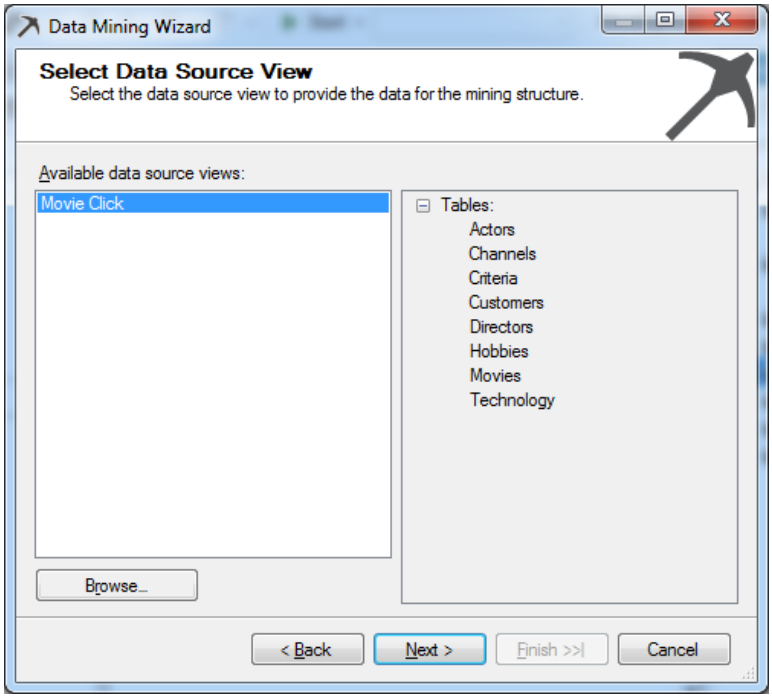
Εικόνα 7.3

4. Στο νέο παράθυρο, όπως φαίνεται στην Εικόνα 7.4, επιλέγουμε τον αλγόριθμο με τον οποίο θα επεξεργαστούμε τα δεδομένα. Στη συγκεκριμένη περίπτωση επιλέγουμε τον Microsoft Decision Trees, καθώς ασχολούμαστε με τα δέντρα αποφάσεων. Στη συνέχεια, επιλέγουμε Next>, ώστε να προχωρήσουμε στο επόμενο βήμα.



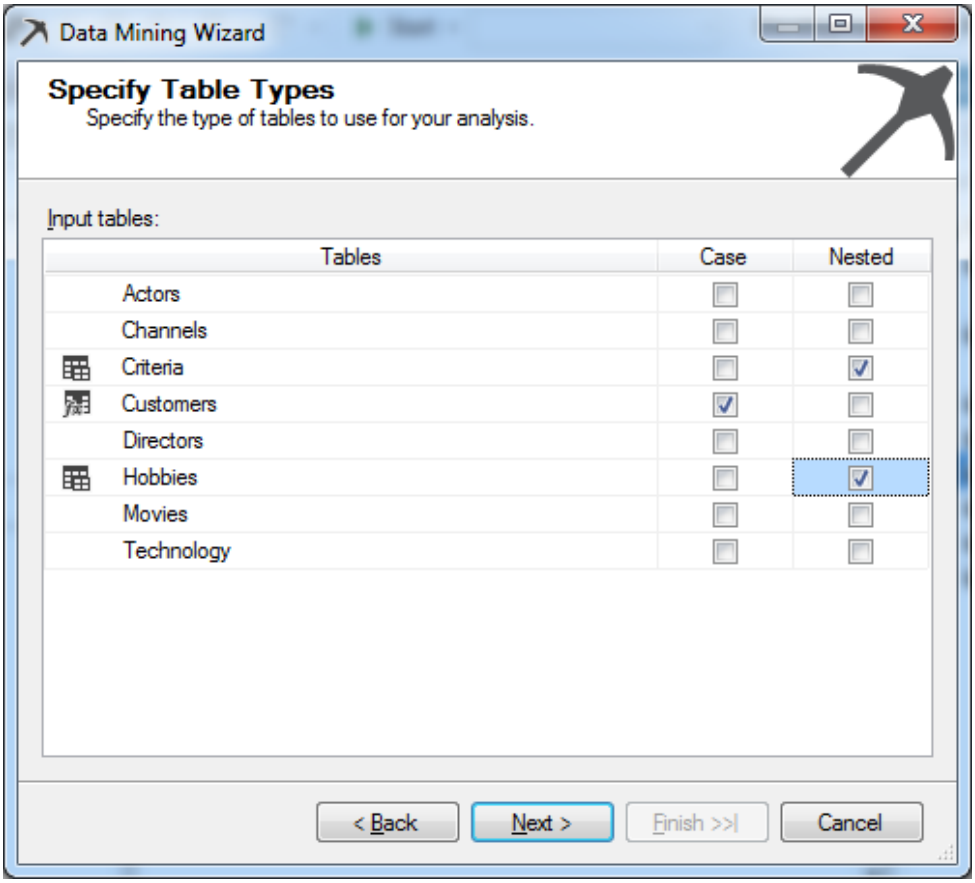
Εικόνα 7.4

5. Στο νέο παράθυρο, όπως φαίνεται στην Εικόνα 7.5, βλέπουμε τα διαθέσιμα source views του project μας. Επιλέγουμε το MovieClick και, στη συνέχεια, Next>, ώστε να προχωρήσουμε στο επόμενο βήμα.



Εικόνα 7.5

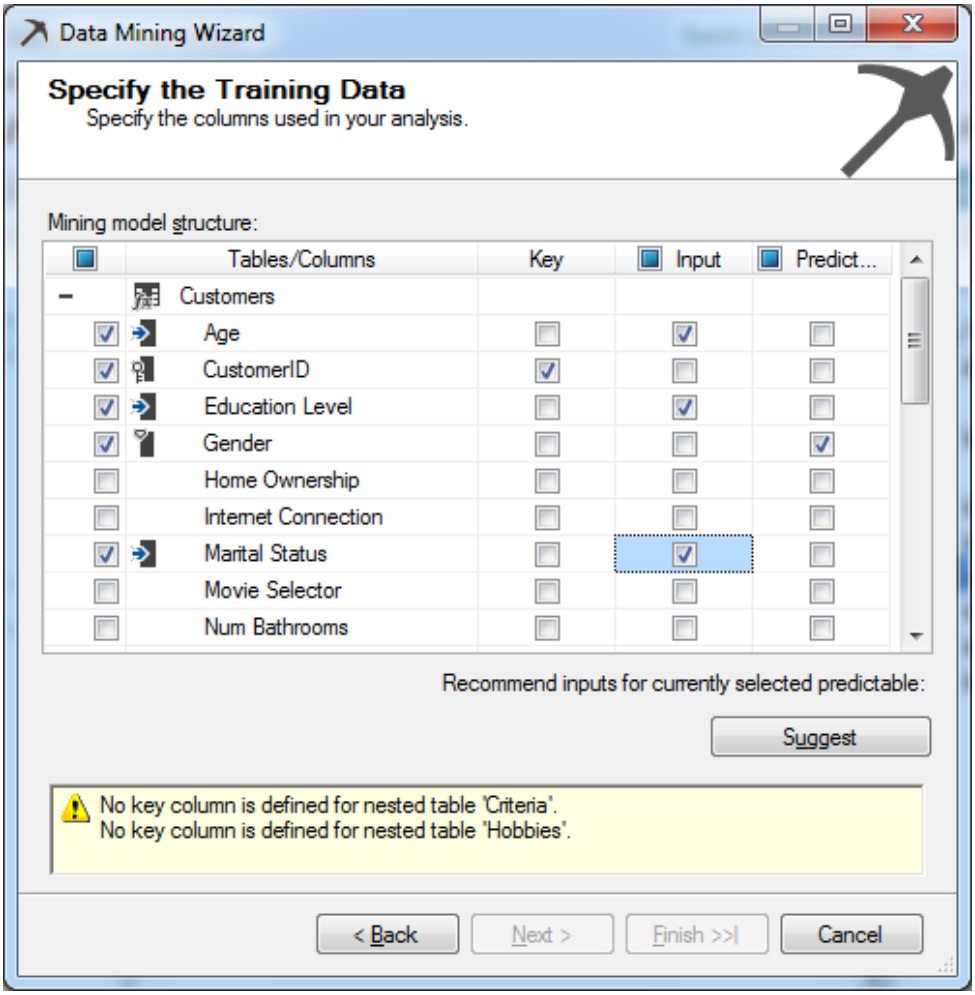
6. Σ’ αυτό το στάδιο, όπως φαίνεται στην Εικόνα 7.6, επιλέγουμε ποιος πίνακας θα είναι case και ποιοι πίνακες θα είναι nested. Case είναι ο πίνακας που περιέχει τα δεδομένα που θέλουμε να προβλέψουμε, ενώ Nested είναι οι πίνακες τα δεδομένα των οποίων είναι παράμετροι στον Case. Στη συγκεκριμένη περίπτωση επιλέγουμε τον πίνακα Customers ως Case και τους πίνακες Criteria και Hobbies ως Nested, καθώς θέλουμε να μελετήσουμε κάποια στοιχεία των πελατών σε σχέση με τα Criteria και τα Hobbies.



Εικόνα 7.6

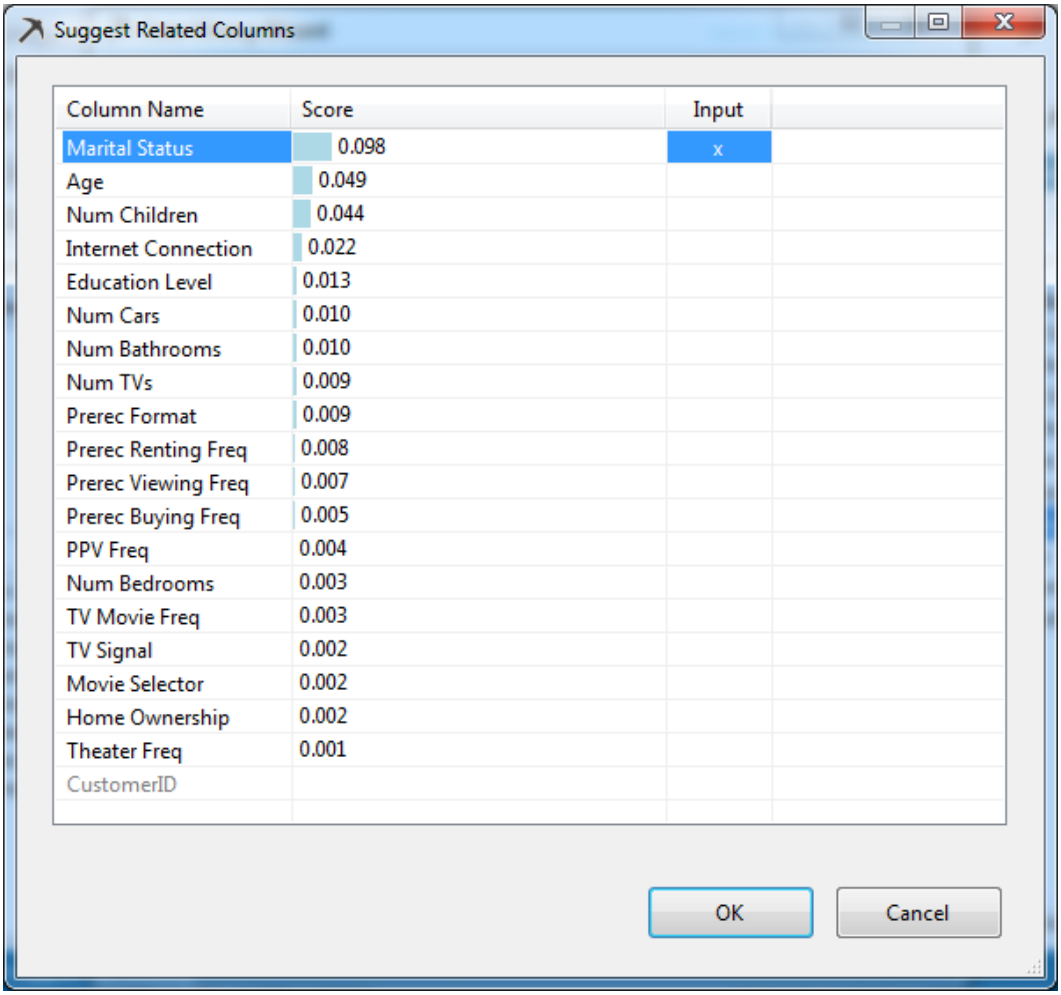
7. Σ’ αυτό το στάδιο, όπως φαίνεται στην Εικόνα 7.7, επιλέγουμε ποια δεδομένα των πινάκων που επιλέξαμε στο προηγούμενο βήμα θα είναι είσοδος στο δέντρο απόφασης και ποια δεδομένα θέλουμε να προβλέψουμε με το δέντρο απόφασης. Συγκεκριμένα, κάνουμε τις εξής επιλογές:

   - Για κάθε πίνακα επιλέγουμε ένα κλειδί Key. Στη συγκεκριμένη περίπτωση επιλέγουμε τα CustomerID, Hobby και Criteria.
   - Ορίζουμε ως Input τις στήλες των πινάκων που μας ενδιαφέρουν. Στη συγκεκριμένη περίπτωση επιλέγουμε τα Age, Education Level, Marital Status, Criteria και Hobby.
   - Ορίζουμε ως Predictable τη στήλη που μας ενδιαφέρει να προβλέψουμε. Αυτή θα είναι και η έξοδος του δέντρου. Στη συγκεκριμένη περίπτωση επιλέγουμε το Gender.



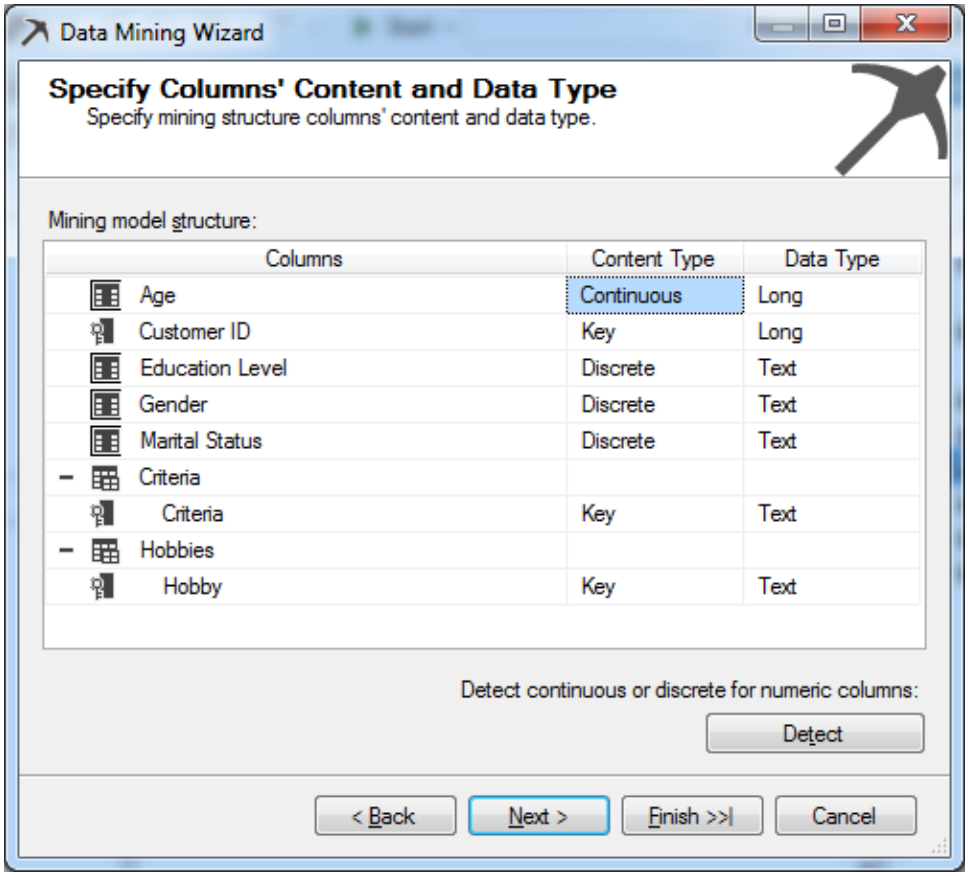
Εικόνα 7.7

8. Αν κάνουμε κλικ στο κουμπί Suggest, εμφανίζεται, όπως φαίνεται στην Εικόνα 7.8, μια εκτίμηση σχετικά με τα πιο σημαντικά χαρακτηριστικά για την πρόβλεψη της Predictable μεταβλητής. Αν επιλέγαμε OK, τότε όλα τα χαρακτηριστικά που φαίνεται να συσχετίζονται θα συμπεριλημβάνονταν στο Mining Structure που θέλουμε να δημιουργήσουμε. Γι΄ αυτό, εμείς επιλέγουμε Cancel. Στη συνέχεια, επιλέγουμε Next>, ώστε να προχωρήσουμε στο επόμενο βήμα.



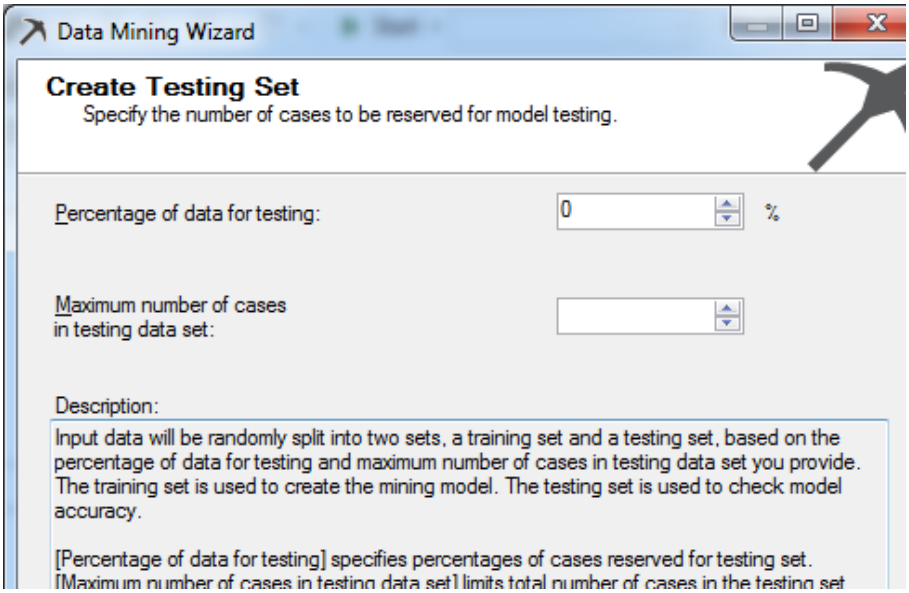
Εικόνα 7.8

9. Στη συνέχεια, όπως φαίνεται στην Εικόνα 7.9, εμφανίζεται μια σύνοψη-επιβεβαίωση του περιεχομένου του Mining Structure. Επιλέγουμε Detect, για να επιλεχθεί ο κατάλληλος τύπος δεδομένων από το σύστημα (είτε continuous ή discrete) που γίνεται ύστερα από δειγματοληψία και ανάλυση των δεδομένων από το σύστημα. Στη συνέχεια, επιλέγουμε Next.



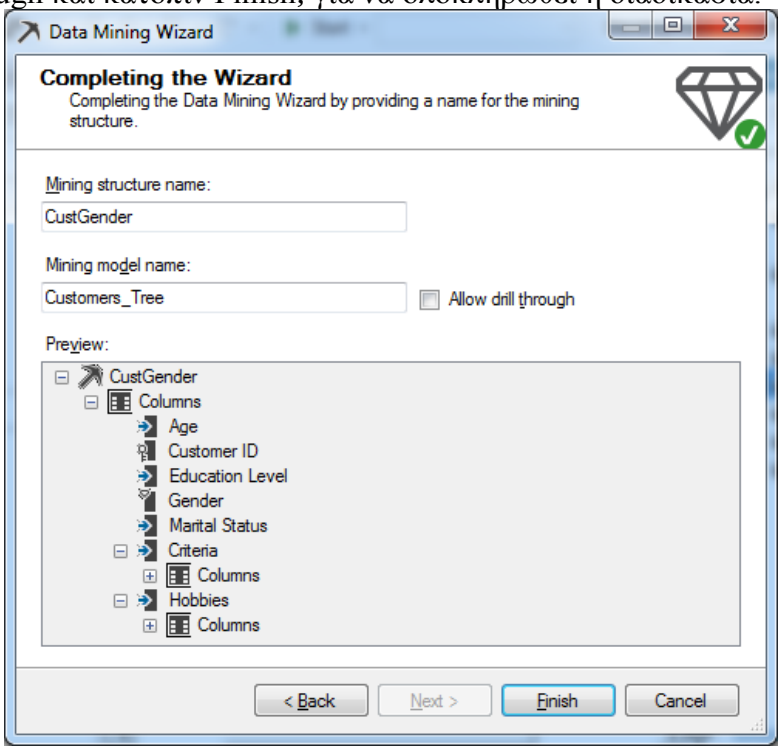
Εικόνα 7.9

10. Στο νέο παράθυρο, όπως φαίνεται στην Εικόνα 7.10, ορίζουμε το ποσοστό των δεδομένων που το μοντέλο θα διατηρήσει για την επαλήθευσή του. Στη συγκεκριμένη περίπτωση δηλώνουμε το ποσοστό 0%, καθώς θέλουμε να χρησιμοποιήσουμε όλο το σύνολο των δεδομένων της βάσης.



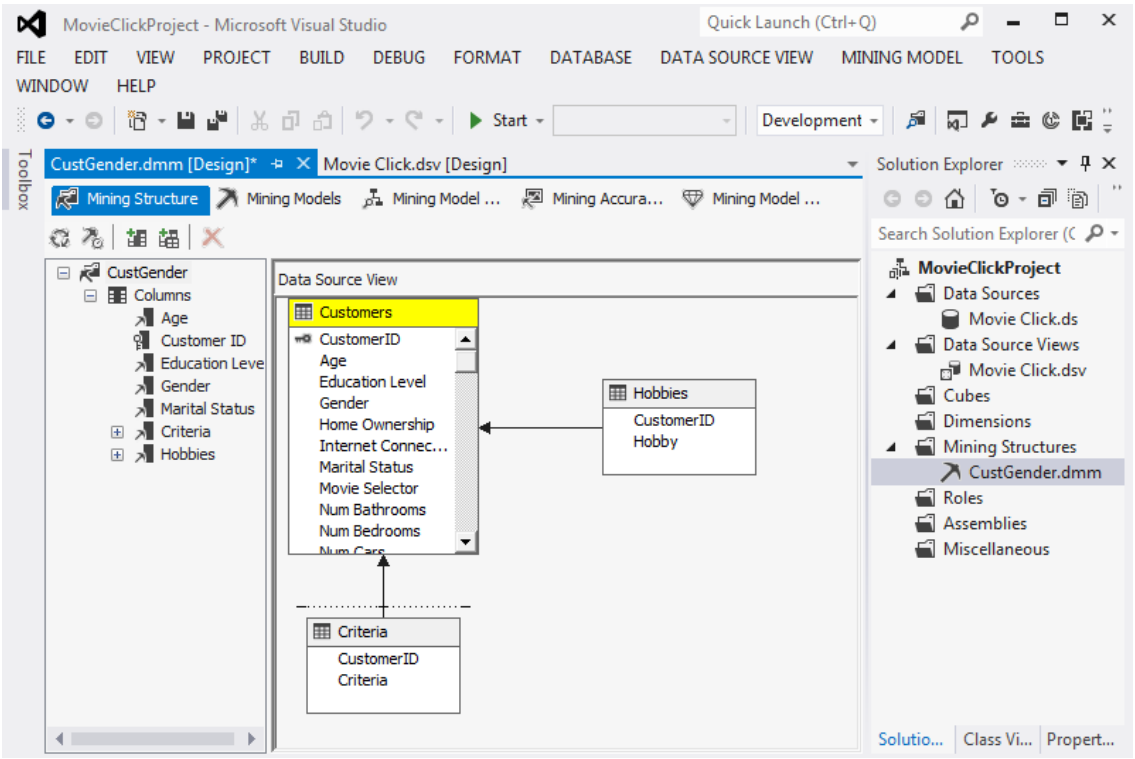
Εικόνα 7.10

11. Στη συνέχεια, ορίζουμε όνομα για το Mining Structure και το Mining Model. Στη συγκεκριμένη περίπτωση, όπως φαίνεται στην Εικόνα 7.11, συμπληρώνουμε CustGender στο Mining structure name και Customers_Tree στο Mining model name. Τέλος, επιλέγουμε Allow drill through και κατόπιν Finish, για να ολοκληρωθεί η διαδικασία.



Εικόνα 7.11

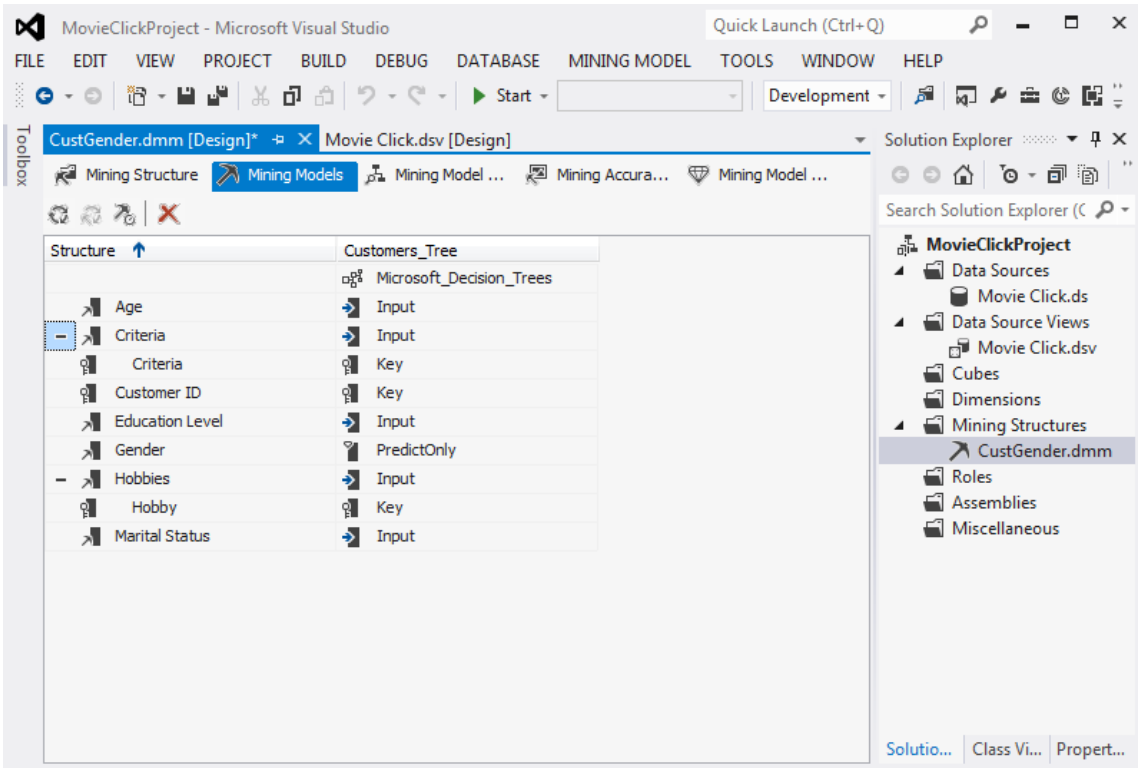
12. Στο παράθυρο του Data Source View και στην καρτέλα Mining Structure βλέπουμε το μοντέλο που δημιουργήσαμε, όπως φαίνεται στην Εικόνα 7.12.



Εικόνα 7.12

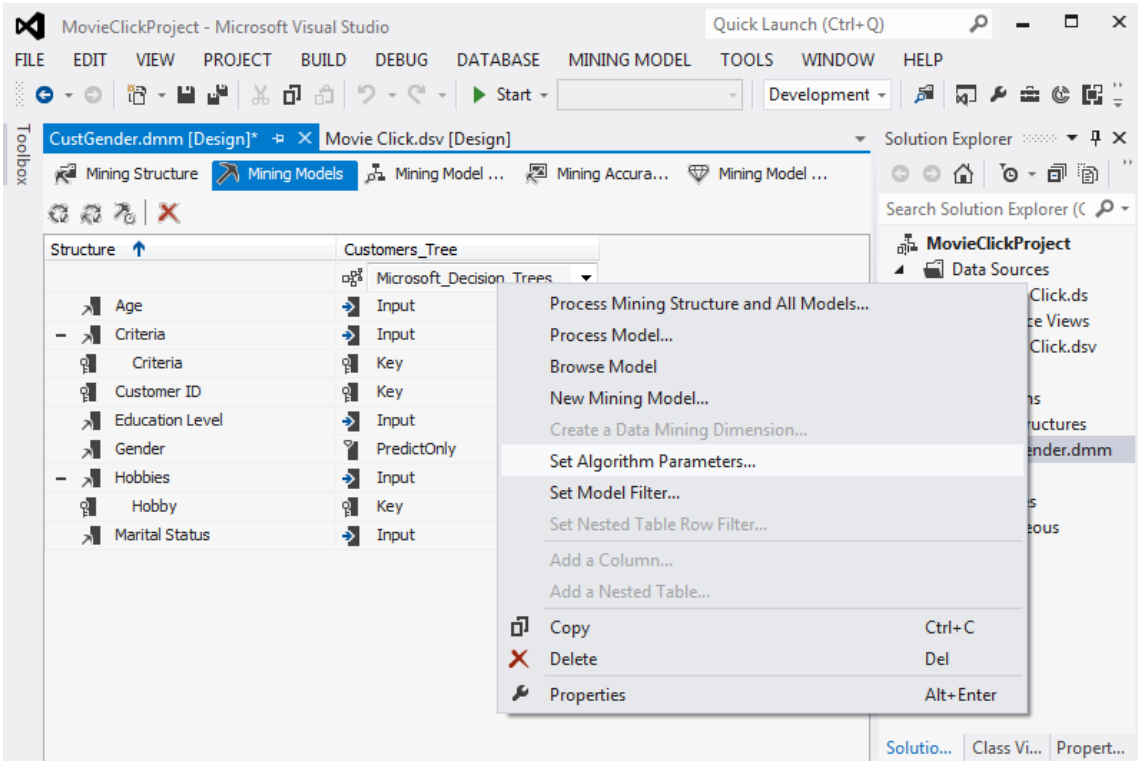
13. Στη συνέχεια, επιλέγουμε την καρτέλα Mining Models, ώστε να καθορίσουμε τις παραμέτρους για τα δέντρα που θα μελετήσουμε. Όπως φαίνεται στην Εικόνα 7.13, κάθε δεδομένο έχει οριστεί ως Input, Key, Predict ή PredictOnly. Η διαφορά ανάμεσα σε Predict και PredictOnly είναι ότι στο πρώτο τα δεδομένα μπορούμε να τα χρησιμοποιήσουμε ως είσοδο, αλλά και ως έξοδο (πρόβλεψη) στον αλγορίθμου, ενώ στο δεύτερο μπορούμε να τα χρησιμοποιήσουμε μόνο ως έξοδο. Στη συγκεκριμένη περίπτωση θέλουμε να προβλέψουμε το φύλλο των πελατών ανάλογα με την ηλικία, τη μόρφωση, τα hobbies και την οικογενειακή κατάσταση. Επομένως, ορίζουμε τα χαρακτηριστικά ως εξής:

- Age: Input
- Criteria: Ignore
- CustomerID: Key
- Education Level: Input
- Gender: PredictOnly
- Hobies: Input
- Marital Status: Input



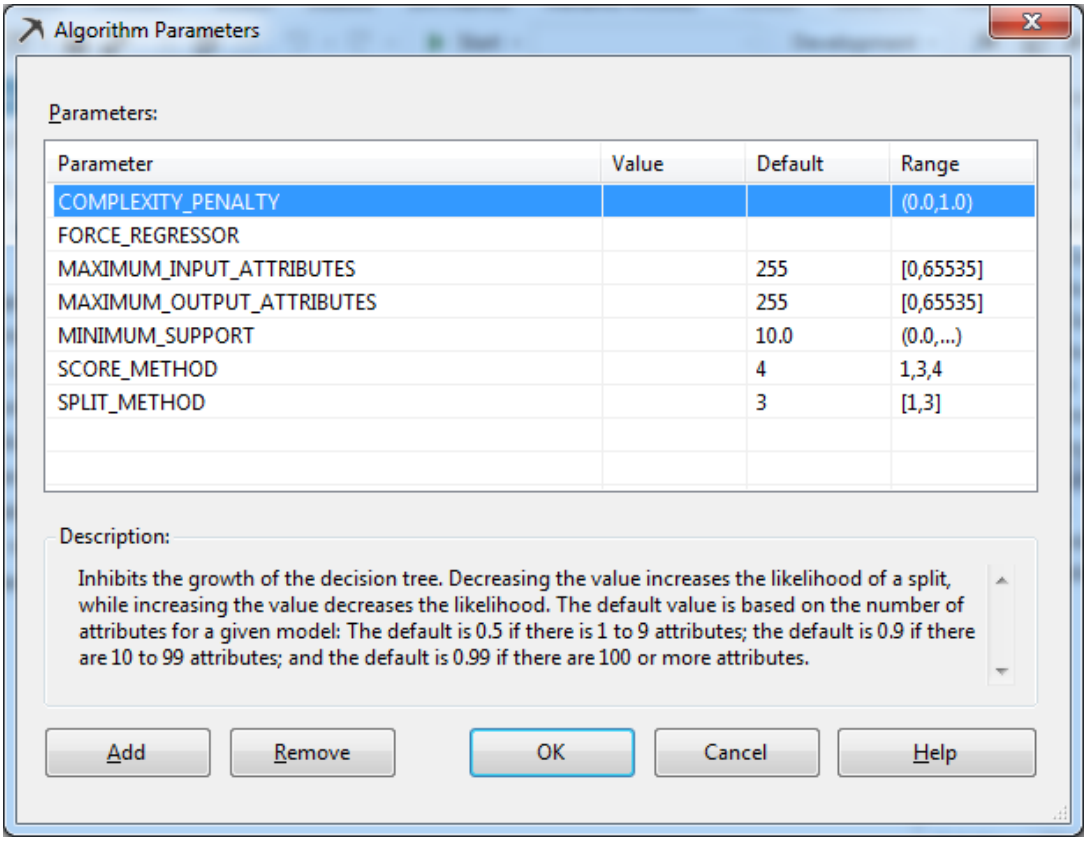
Εικόνα 7.13

14. Στη συνέχεια, θα μελετήσουμε αφενός τις παραμέτρους με τις οποίες κατασκευάζεται το δέντρο απόφασης και αφετέρου τις προεπιλεγμένες τιμές που παίρνουν. Αρχικά, όπως φαίνεται στην Εικόνα 7.14, κάνουμε δεξί κλικ στον αλγόριθμο Microsoft Decision Trees και επιλέγουμε Set Algorithm Parameters.



Εικόνα 7.14

15. Στο παράθυρο που εμφανίζεται, όπως φαίνεται στην Εικόνα 7.15, βλέπουμε τις μεταβλητές που μπορούμε να παραμετροποιήσουμε.



Εικόνα 7.15

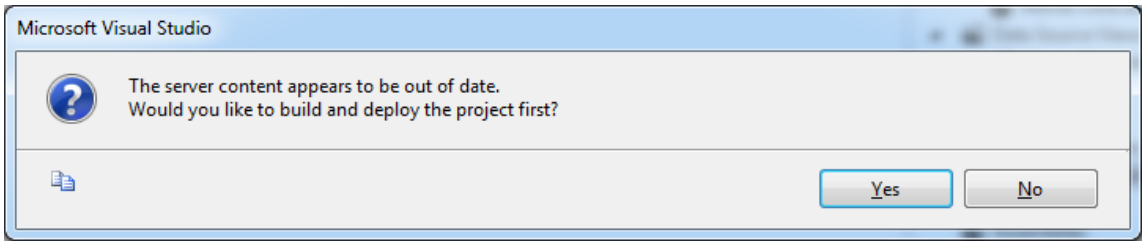
Ακολουθεί η αναλυτική περιγραφή της κάθε παραμέτρου του αλγορίθμου Decision Tree:

- **COMPLEXITY_PENALTY:** Αυτή η παράμετρος καθορίζει το μέγεθος του δέντρου. Η τιμή της εξαρτάται από το πλήθος των χαρακτηριστικών του μοντέλου και λαμβάνει τιμές [0,1]. Όσο η τιμή τείνει στο 1, τόσο μικρότερο είναι το δέντρο που προκύπτει. Αν τα χαρακτηριστικά είναι λιγότερα από 10, τότε η προεπιλεγμένη τιμή της παραμέτρου είναι 0,5. Αν τα χαρακτηριστικά είναι περισσότερα από 100, τότε η προεπιλεγμένη τιμή είναι 0,9. Τέλος, αν τα χαρακτηριστικά είναι περισσότερα ή ίσα από 100, τότε η προεπιλεγμένη τιμή είναι 0,99. Στη συγκεκριμένη περίπτωση τα χαρακτηριστικά είναι περισσότερα από 10 και λιγότερα από 100, οπότε η προεπιλεγμένη τιμή είναι 0.9.
- **FORCE_REGRESSOR:** Αυτή η παράμετρος προσδιορίζει τον αριθμό των στιγμιοτύπων που απαιτούνται, προκειμένου ένας κόμβος να διασπαστεί σε δύο ή περισσότερους κόμβους.
- **MAXIMUM_INPUT_ATTRIBUTES:** Αυτή η παράμετρος καθορίζει τον μέγιστο αριθμό των χαρακτηριστικών εισόδου που ο αλγόριθμος μπορεί να χειριστεί πριν αυτός αρχίσει να επιλέγει χαρακτηριστικά. Η τιμή 0 απενεργοποιεί τη δυνατότητα επιλογής των χαρακτηριστικών εισόδου. Στη συγκεκριμένη περίπτωση αφήνουμε την προεπιλεγμένη τιμή.
- **MAXIMUM_OUTPUT_ATTRIBUTES:** Η συγκεκριμένη παράμετρος καθορίζει τον μέγιστο αριθμό των χαρακτηριστικών εξόδου που ο αλγόριθμος μπορεί να χειριστεί πριν αυτός αρχίσει να επιλέγει χαρακτηριστικά. Η τιμή 0 απενεργοποιεί την δυνατότητα επιλογής των χαρακτηριστικών εξόδου. Στη συγκεκριμένη περίπτωση αφήνουμε την προεπιλεγμένη τιμή.
- **MINIMUM_SUPPORT:** Αυτή η παράμετρος προσδιορίζει το ελάχιστο πλήθος των περιπτώσεων στα φύλλα του δέντρου, που απαιτούνται για τη δημιουργία του. Όταν η τιμή αυτή είναι μικρότερη ή ίση με 1, τότε εκφράζει ποσοστό σε σχέση με το πλήθος όλων των περιπτώσεων. Διαφορετικά, όταν η τιμή είναι μεγαλύτερη από 1, εκφράζει πλήθος. Μ' αυτήν την παράμετρο καθορίζουμε στο σύστημα

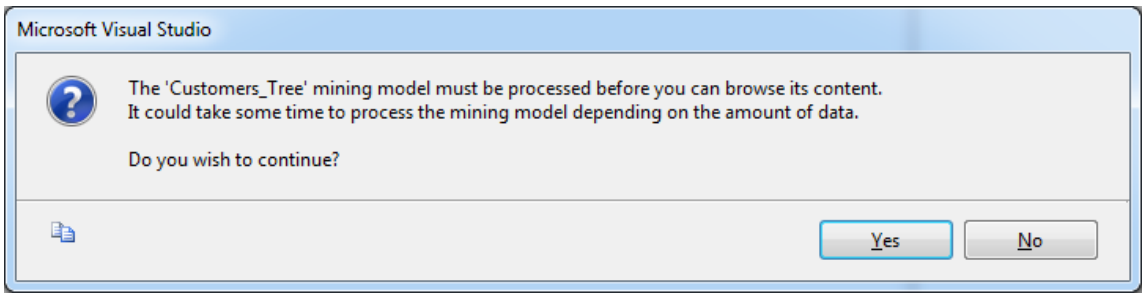
πότε θα σταματήσει η ανάλυση του δέντρου, δηλαδή καθορίζουμε το μέγεθός του. Στη συγκεκριμένη περίπτωση η προεπιλεγμένη τιμή είναι 10.

- **SCORE_METHOD:** Αυτή η παράμετρος καθορίζει τη μέθοδο επιλογής διαχωριστικού χαρακτηριστικού για τη δημιουργία ενός δέντρου απόφασης. Μπορεί να πάρει τις τιμές 1, ή 3, ή 4. Η τιμή 1 είναι για την Εντροπία, η τιμή 3 είναι για τα δίκτυα Bayesian with K2 Prior (BK2) και η τιμή 4 είναι για τα δίκτυα Bayesian Dirichlet Equivalent with Uniform Prior (BDE). Στη συγκεκριμένη περίπτωση αφήνουμε την προεπιλεγμένη τιμή 4. Όπως περιγράφηκε στην ενότητα 7.1, η παράμετρος score_method μετράει την βαθμό βεβαιότητας/αβεβαιότητας που δημιουργεί στο μοντέλο ένα χαρακτηριστικό έναντι των υπόλοιπων χαρακτηριστικών. Έτσι, θα μπορούμε να αποφασίσουμε αν αυτό το χαρακτηριστικό θα επιλεγεί ή όχι, για να γίνει κόμβος του δέντρου απόφασης.
- **SPLIT_METHOD:** Αυτή η παράμετρος καθορίζει τη μέθοδο με την οποία διαχωρίζονται οι κόμβοι του δέντρου. Μπορεί να πάρει τις τιμές [1,3] όπου 1 είναι η τιμή για Binary δέντρο, 2 η τιμή για Complete (multi-way) δέντρο και 3 η τιμή και για τα δύο μαζί.

16. Για να προβάλουμε το δέντρο, επιλέγουμε την καρτέλα Mining Model Viewer. Σε περίπτωση που δεν έχουν αποθηκευτεί οι αλλαγές που έχουμε κάνει, θα εμφανιστούν τα παρακάτω μηνύματα στα οποία επιλέγουμε Yes.

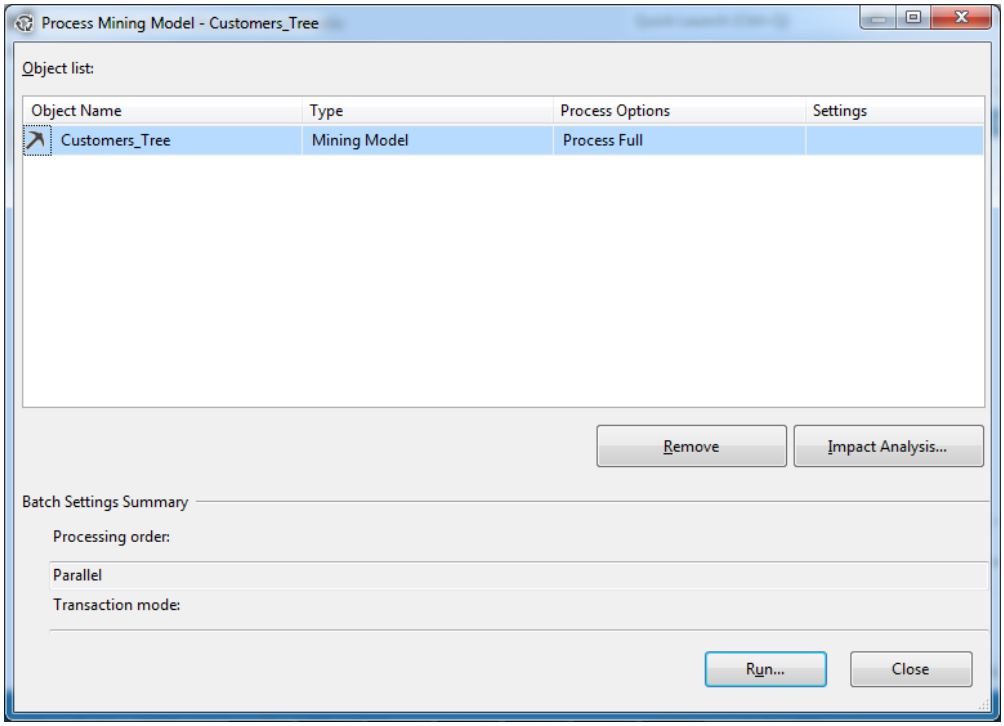


Εικόνα 7.16



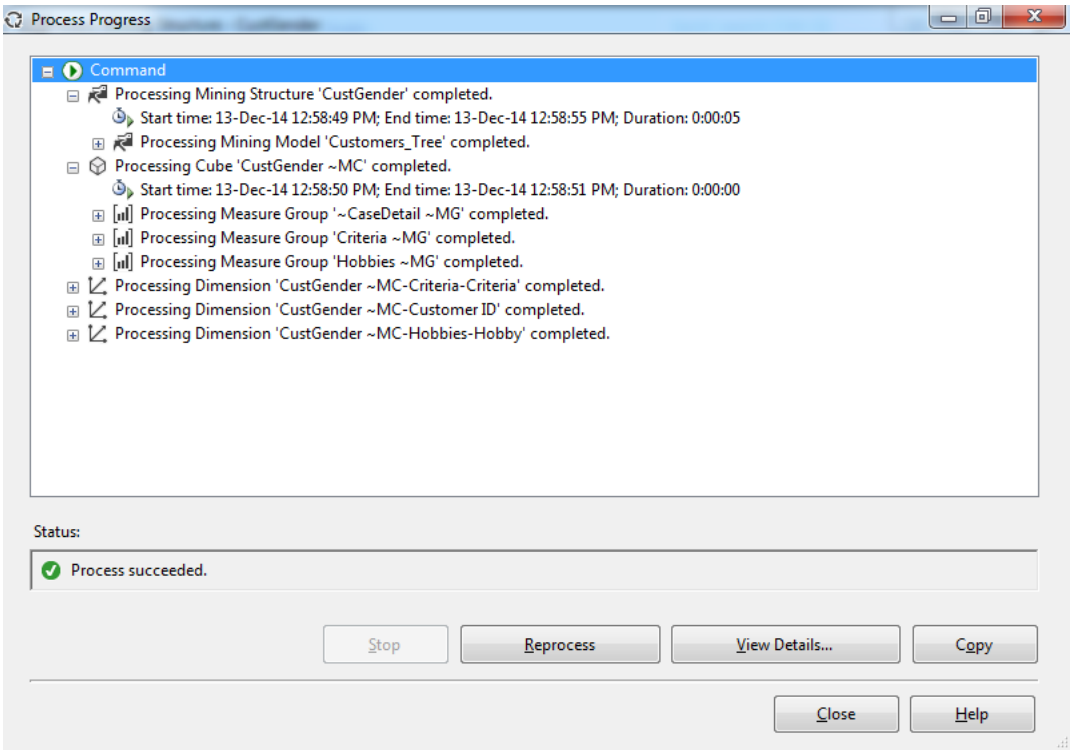
Εικόνα 7.17

Στη συνέχεια, όπως φαίνεται στην Εικόνα 7.18, εμφανίζονται συγκεντρωμένες οι επιλογές μας. Επιλέγουμε Run, ώστε να δημιουργηθεί το δέντρο και να γίνει deploy.



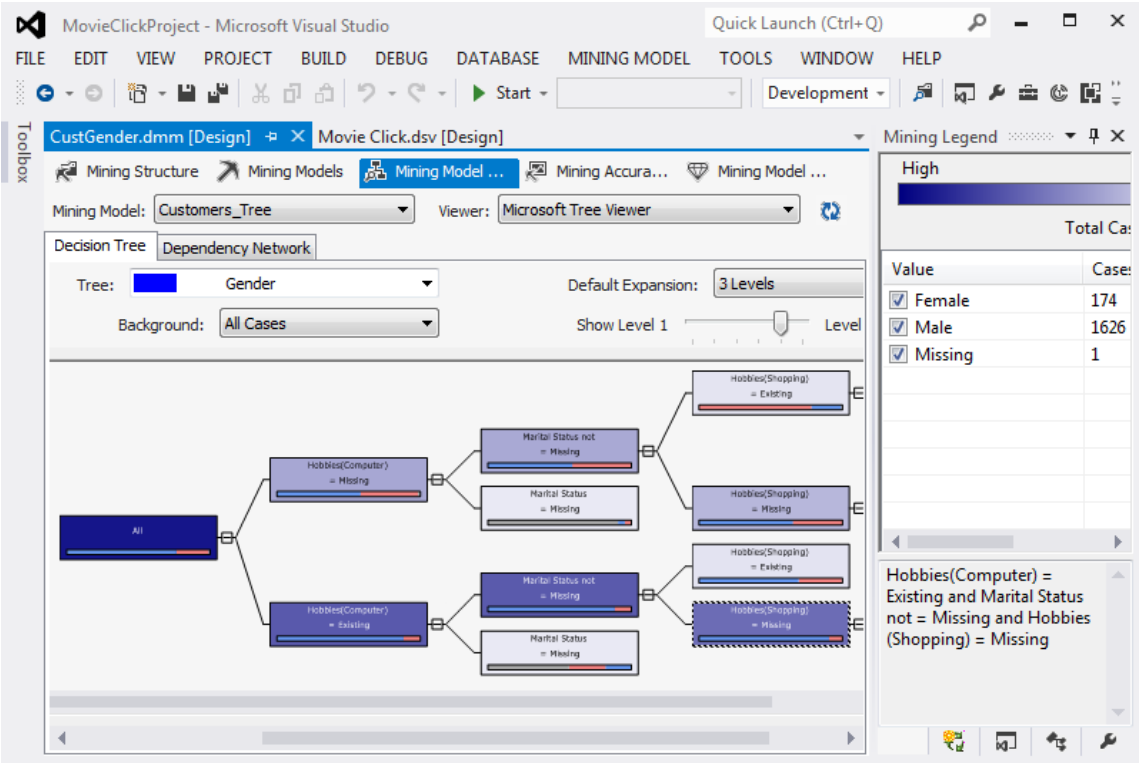
Εικόνα 7.18

Στο νέο παράθυρο, όπως φαίνεται στην Εικόνα 7.19, παρουσιάζονται οι ενέργειες που έγιναν για τη δημιουργία του δέντρου, συνοδευόμενες από μια παρατήρηση επιτυχούς ολοκλήρωσης. Επιλέγουμε Close, ώστε να ολοκληρώσουμε τη διαδικασία και να προβάλουμε το δέντρο.



Εικόνα 7.19

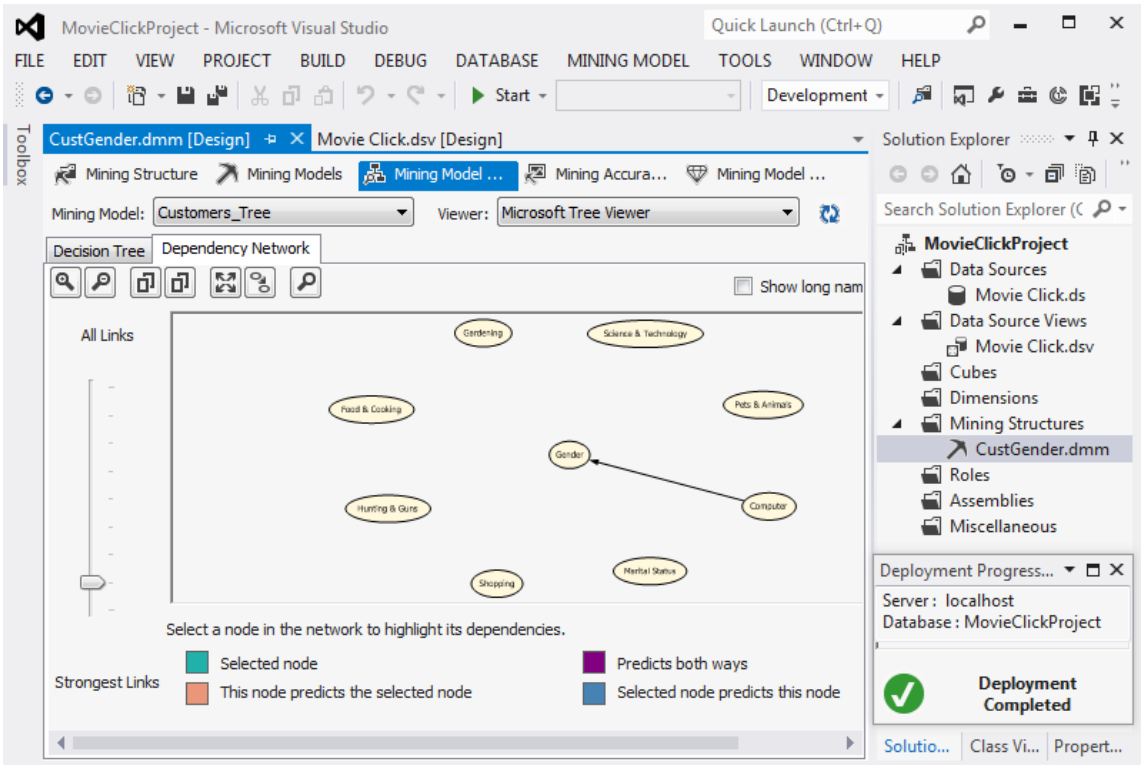
17. Στη συνέχεια, εμφανίζεται το παράθυρο με το δέντρο που έχει δημιουργηθεί, όπως φαίνεται στην Εικόνα 7.20. Αν επιλέξουμε Size to Fit, εμφανίζεται όλο το δέντρο (με όλα τα επίπεδά του), ώστε να έχουμε μια ολοκληρωμένη εικόνα του.



Εικόνα 7.20

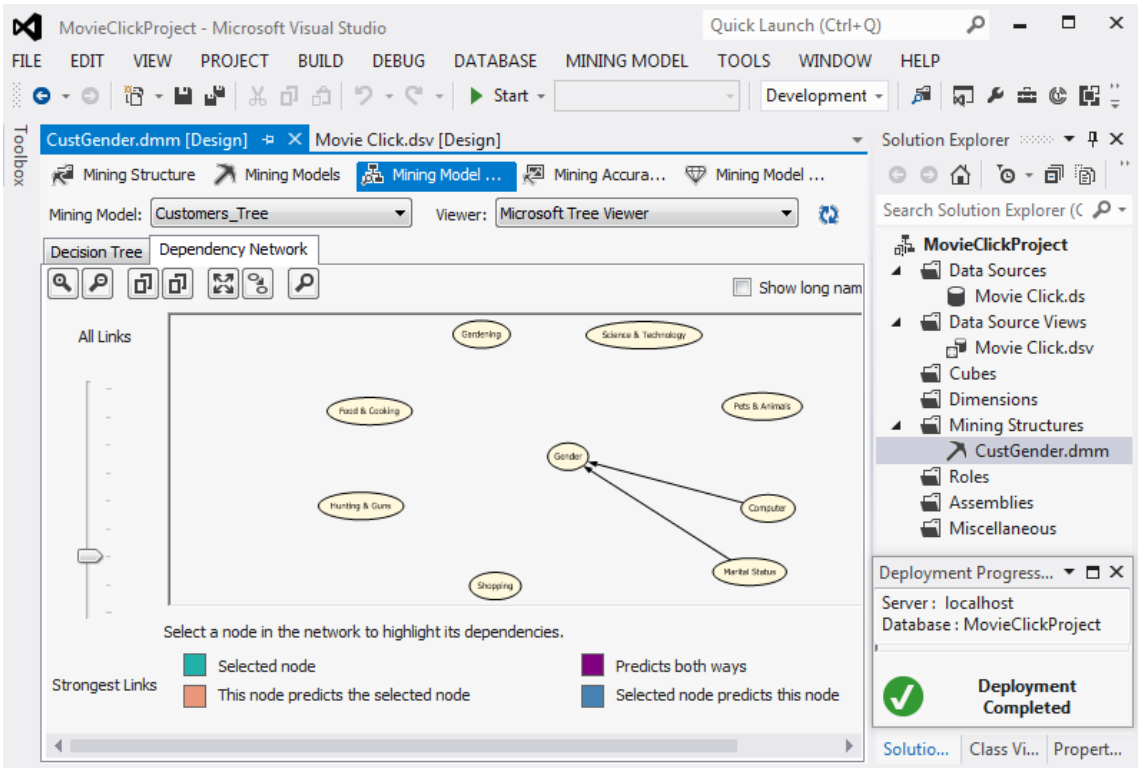
Αφήνοντας τον κέρσορα πάνω σε έναν κόμβο, βλέπουμε ότι εμφανίζονται κάποια στατιστικά στοιχεία σχετικά με το σύνολο των περιπτώσεων που ανήκουν σε κάθε κόμβο του δέντρου και σχετικά με την τιμή που έχουν: Male ή Female. επίσης, παρατηρούμε ότι σε κανένα φύλλο του δέντρου η τιμή των συνολικών περιπτώσεων δεν είναι μικρότερη του 10, έτσι όπως είναι καθορισμένη από την παράμετρο MINIMUM_SUPPORT. Τέλος, το σύνολο των περιπτώσεων κάθε κόμβου του δέντρου είναι ίσο με το άθροισμα των περιπτώσεων των παιδιών του, κάτι που ισχύει και για τις τιμές των χαρακτηριστικών.

18. Στη συνέχεια, επιλέγουμε την καρτέλα Dependency Network, όπως φαίνεται στην Εικόνα 7.21, ώστε να μελετήσουμε τη σχέση μεταξύ του χαρακτηριστικού που θέλουμε να προβλέψουμε και των υπολοίπων που σχετίζονται με αυτό και συμμετέχουν στη δημιουργία του δέντρου. Στα αριστερά του γραφήματος υπάρχει μια μπάρα, η μετακίνηση της οποίας παρουσιάζει τον βαθμό εξάρτησης του χαρακτηριστικού που θέλουμε να προβλέψουμε από τα υπόλοιπα χαρακτηριστικά. Η διαβάθμιση γίνεται από το χαμηλότερο προς το υψηλότερο επίπεδο της μπάρας, με το χαμηλότερο να δηλώνει τη μεγαλύτερη εξάρτηση και το υψηλότερο τη μικρότερη. Στη συγκεκριμένη περίπτωση, το χαρακτηριστικό που θέλουμε να προβλέψουμε (Gender) επηρεάζεται πρώτα από το χαρακτηριστικό Computer.



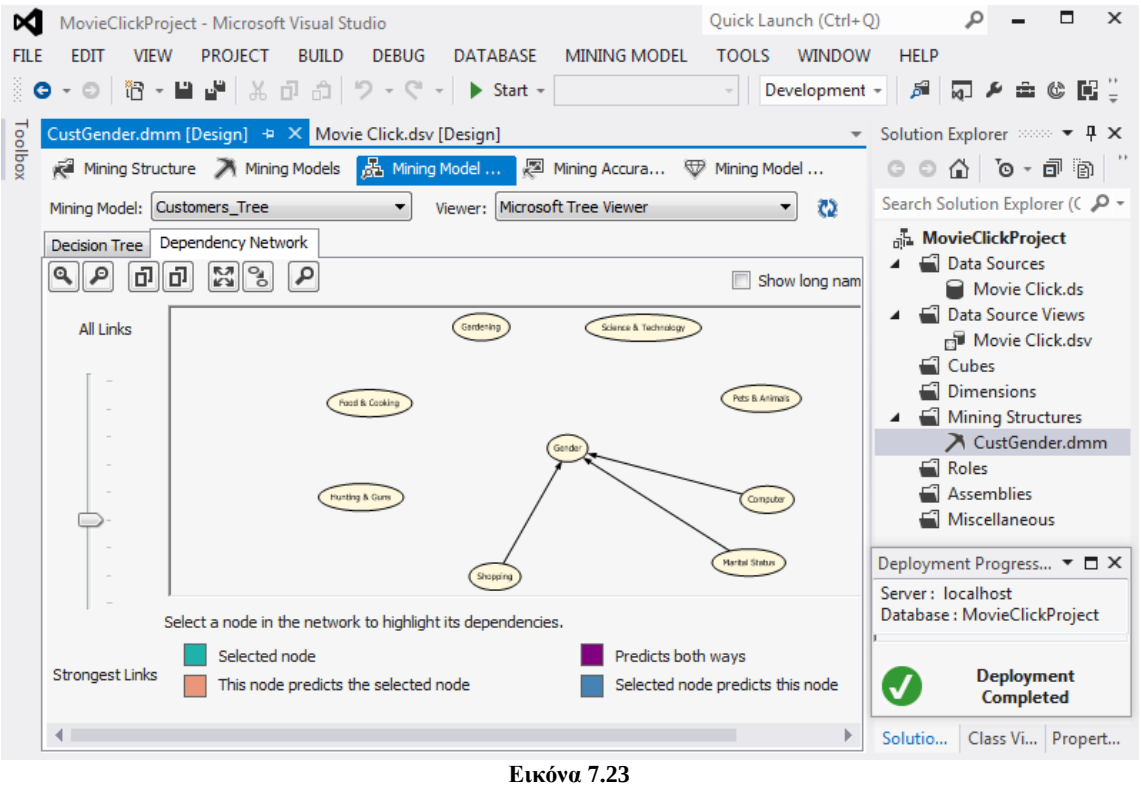
Εικόνα 7.21

Στο δεύτερο επίπεδο εξάρτησης το gender επηρεάζεται από το Computer και το Marital Status, όπως φαίνεται στην Εικόνα 7.22.



Εικόνα 7.22

Στο τρίτο επίπεδο εξάρτησης το gender επηρεάζεται από το Computer, το Marital Status και το Shopping, όπως φαίνεται στην Εικόνα 7.23.



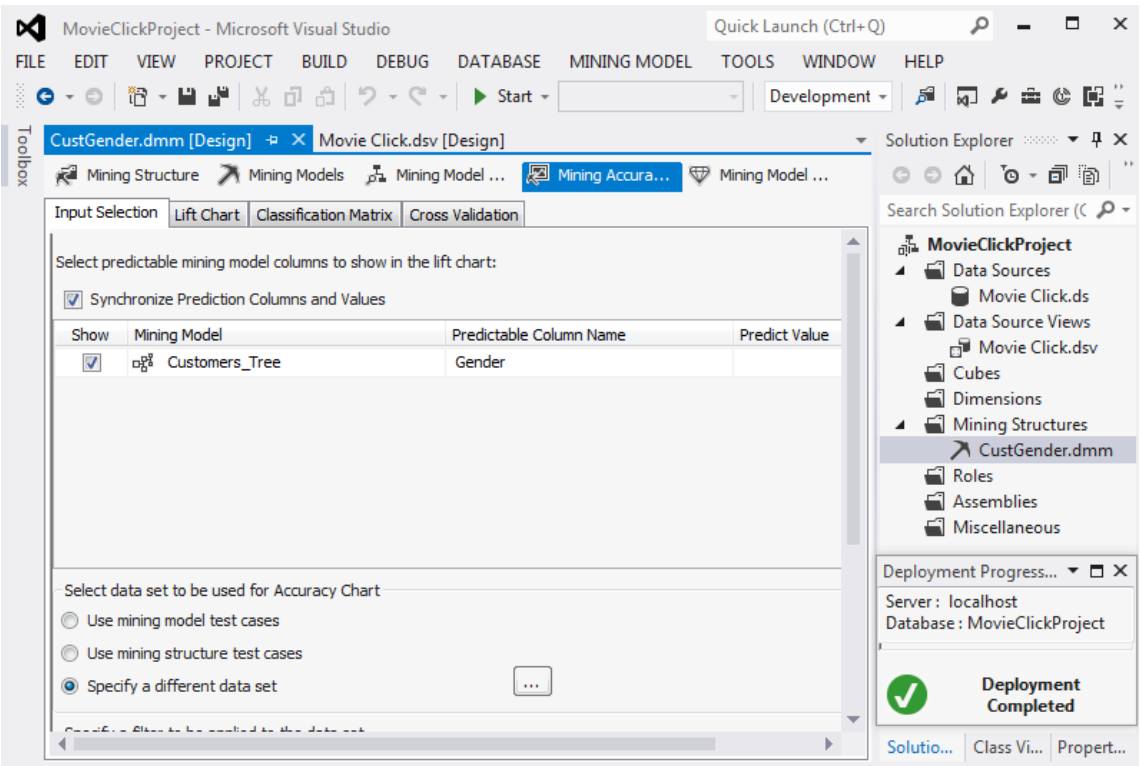
Εικόνα 7.23

7.3. Αξιολόγηση δέντρων απόφασης

Σ' αυτήν την ενότητα θα παρουσιάσουμε με αναλυτικά βήματα τη μέθοδο με την οποία αξιολογούνται τα δέντρα απόφασης. Πιο συγκεκριμένα, θα εξακριβώσουμε κατά πόσο το δέντρο που έχουμε δημιουργήσει είναι αποτελεσματικό, δηλαδή αν είναι σχετικά ακριβές στα δεδομένα που θέλουμε να προβλέψουμε. Μ' αυτόν τον τρόπο, θα μπορέσουμε να αξιολογήσουμε το δέντρο απόφασης, αποτυπώνοντας τα αποτελέσματα της ακρίβειας πρόβλεψης είτε με γραφήματα είτε με πίνακες σύγκρισης.

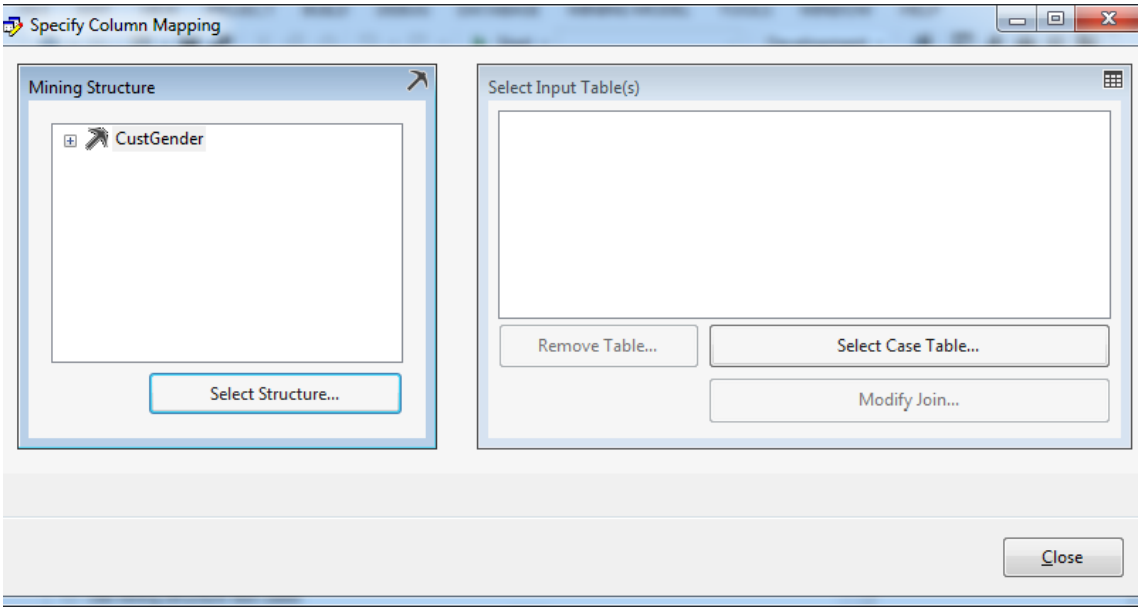
Αναλυτικά Βήματα

1. Επιλέγουμε την καρτέλα Mining Accuracy Chart και, όπως φαίνεται στην Εικόνα 7.24, στην καρτέλα Input Selection επιλέγουμε Specify a different data set, ώστε να καθορίσουμε το Mining Structure, τον Case πίνακα και τους Nested πίνακες με τους οποίους θα εργαστούμε.



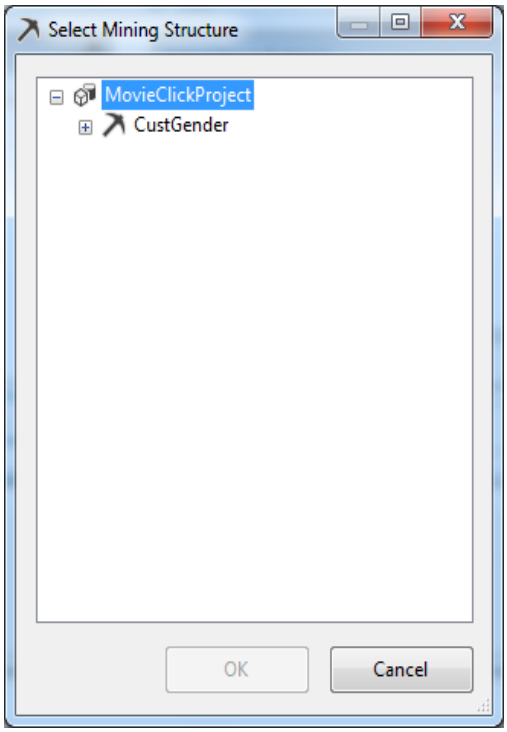
Εικόνα 7.24

2. Στο νέο παράθυρο, όπως φαίνεται στην Εικόνα 7.25, επιλέγουμε Select Structure, ώστε να καθορίσουμε το mining structure με το οποίο θα εργαστούμε.



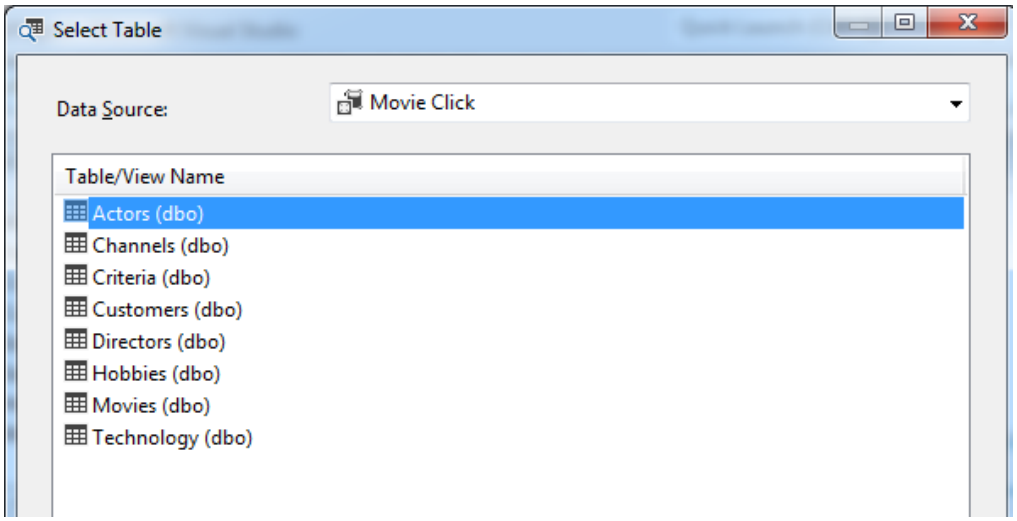
Εικόνα 7.25

3. Εμφανίζεται το παράθυρο επιλογής του mining structure, όπως φαίνεται στην Εικόνα 7.26. Επιλέγουμε το CustGender, που είναι το mining structure με το οποίο εργαζόμαστε, και, στη συνέχεια, επιλέγουμε OK, ώστε να επιστρέψουμε στο προηγούμενο παράθυρο.



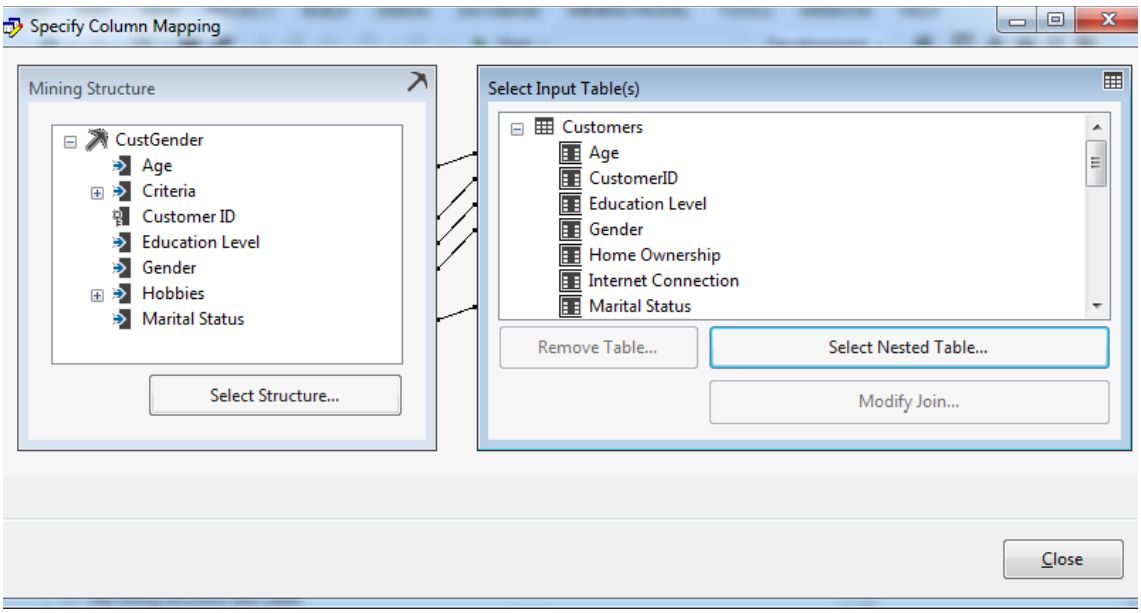
Εικόνα 7.26

4. Στη συνέχεια, επιλέγουμε Select Case Table και, όπως φαίνεται στην Εικόνα 7.27, εμφανίζεται το παράθυρο επιλογής του πίνακα Case, στο οποίο επιλέγουμε τον πίνακα Customers και στη συνέχεια επιλέγουμε OK ώστε να επιστρέψουμε στο προηγούμενο παράθυρο.



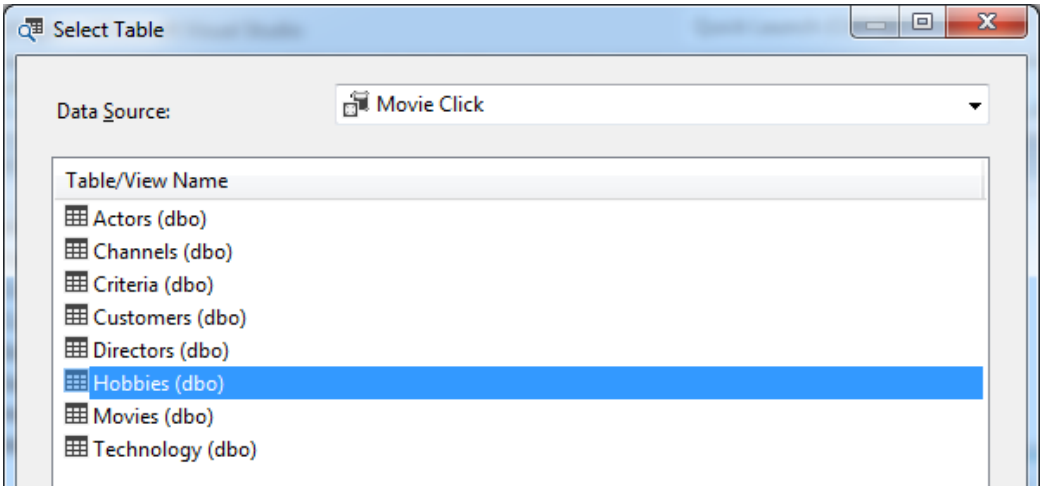
Εικόνα 7.27

5. Στο νέο παράθυρο, όπως φαίνεται στην Εικόνα 7.28, εμφανίζεται συνολικά το Mining Structure με τους πίνακες, όπου βλέπουμε τις σχέσεις που έχουν δημιουργηθεί. Στη συνέχεια, επιλέγουμε Select Nested Table, ώστε να επιλέξουμε τους Nested πίνακες.



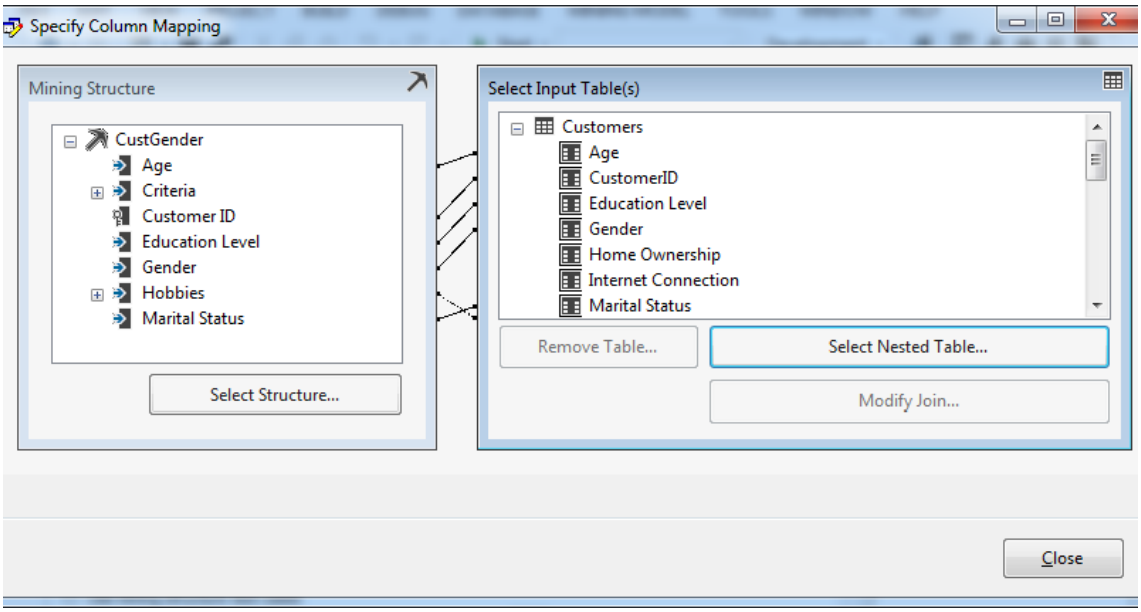
Εικόνα 7.28

6. Εμφανίζεται το παράθυρο επιλογής του πίνακα Nested, όπως φαίνεται στην Εικόνα 7.29. Επιλέγουμε τον πίνακα Hobbies, καθώς είναι ο μόνος πίνακας του Data Mining Structure στον οποίο εργαζόμαστε και δίνει χαρακτηριστικά ως input στον αλγόριθμο που φτιάχνει το δέντρο. Στη συνέχεια, επιλέγουμε OK, ώστε να επιστρέψουμε στο προηγούμενο παράθυρο.



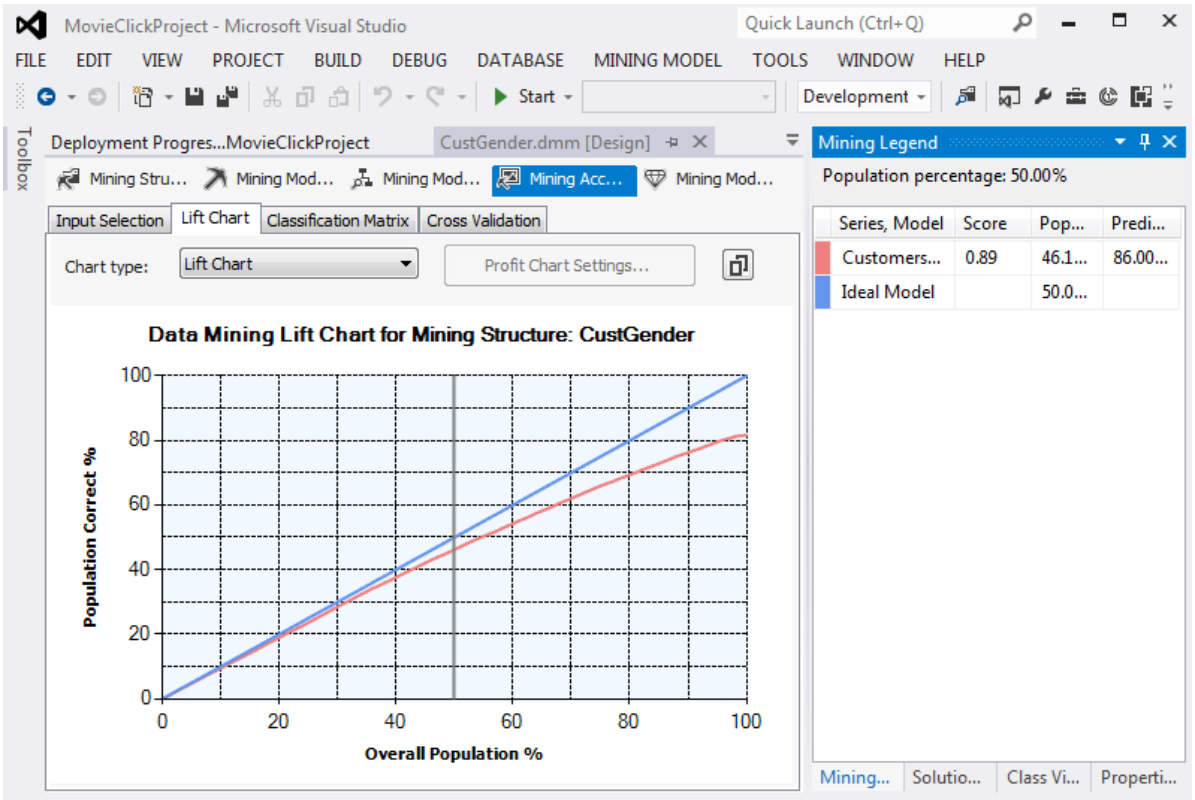
Εικόνα 7.29

7. Εμφανίζεται ξανά το παράθυρο του Mining Structure με τους πίνακες, όπως φαίνεται στην Εικόνα 7.30, όπου πλέον βλέπουμε όλες τις συσχετίσεις που έχουν δημιουργηθεί. Στη συνέχεια, επιλέγουμε Close, ώστε να αφήσουμε αυτό το παράθυρο.



Εικόνα 7.30

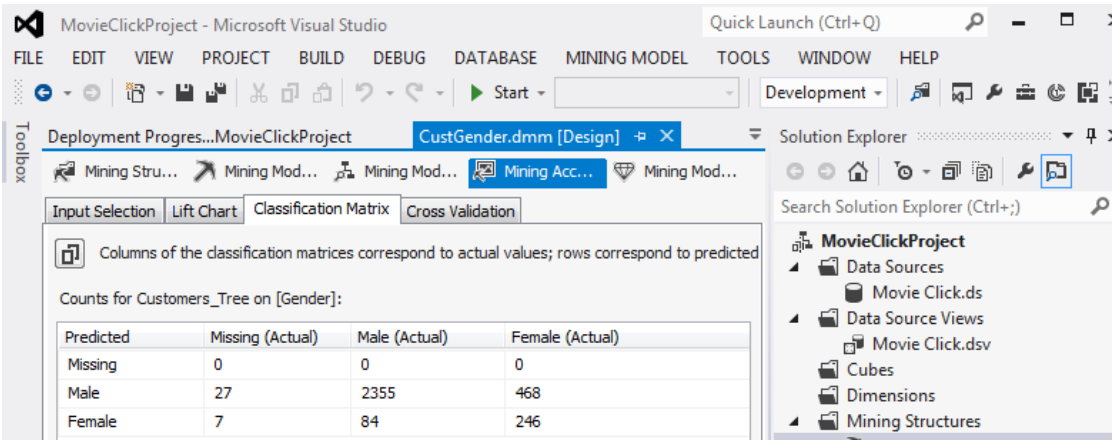
8. Επιλέγοντας την καρτέλα Lift Chart, όπως φαίνεται στην Εικόνα 7.31, διαπιστώνουμε πόσο αποτελεσματικό είναι το δέντρο που δημιουργήσαμε, καθώς σ' αυτό εμφανίζεται το διάγραμμα του ποσοστού του συνολικού πληθυσμού (άξονας X) σε σχέση με το ποσοστό του πληθυσμού που έχουμε προβλέψει σωστά (άξονας Y). Η μπλε γραμμή με κλίση 45^{o} με τον άξονα X απεικονίζει το ιδανικό μοντέλο, ενώ η κόκκινη γραμμή που βρίσκεται κάτω από αυτήν απεικονίζει το δικό μας μοντέλο. Κάνοντας κλικ πάνω στο διάγραμμα εμφανίζεται μια παράλληλη προς τον άξονα Y ευθεία, καθώς και κάποια στατιστικά στοιχεία που καταγράφονται στο παράθυρο Mining Legend.



Εικόνα 7.31

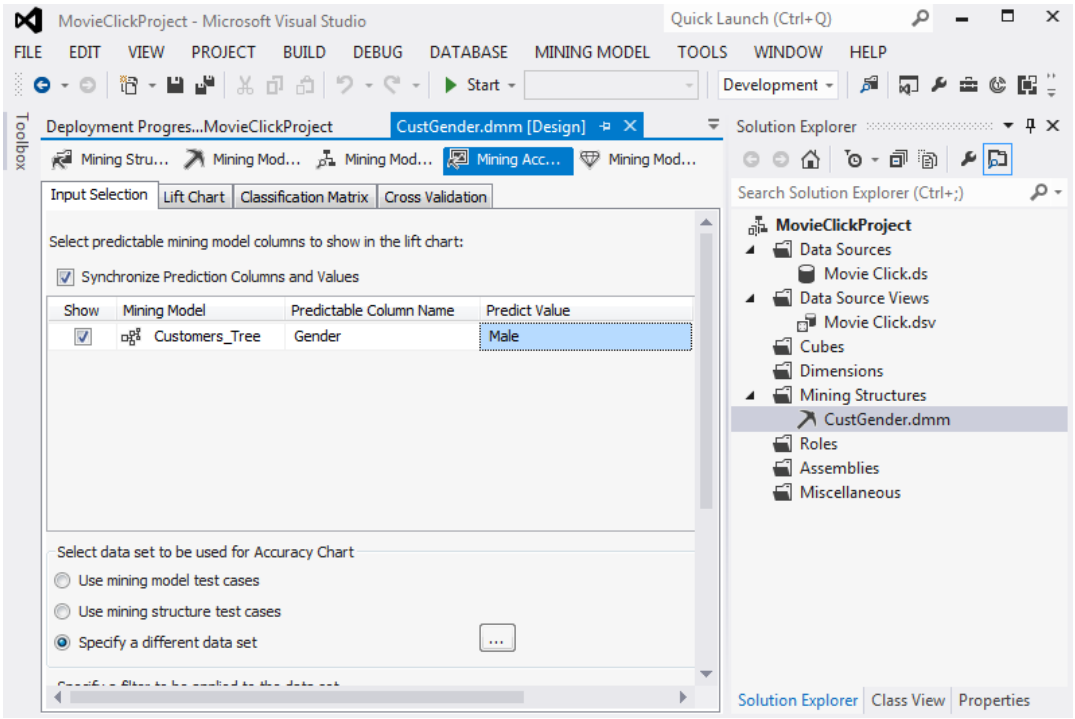
Συγκεκριμένα, στο δεξί μέρος της Εικόνας 7.31 βλέπουμε ότι στο 50% του συνολικού πληθυσμού (που είναι το δείγμα) το δέντρο προβλέπει σωστά το 46,1% του δείγματος, ενώ το ιδανικό είναι να προβλέπει σωστά το 50%. Από τη στιγμή που δεν έχει προσδιοριστεί συγκεκριμένη κατάσταση που θέλουμε να προβλέψουμε, η γραμμή του μοντέλου μας θα βρίσκεται πάντα κάτω από την ευθεία του ιδανικού μοντέλου.

9. Στη συνέχεια, θα αξιολογήσουμε την αξιοπιστία του μοντέλου μας μ' έναν άλλο τρόπο. Επιλέγουμε την καρτέλα Classification Matrix. Στον πίνακα που εμφανίζεται, όπως φαίνεται στην Εικόνα 7.32, βλέπουμε ότι στις 34 περιπτώσεις που το φύλλο δεν ήταν καταχωρημένο (Missing), ο αλγόριθμος εσφαλμένα το προέβλεψε 27 φορές ως Male και 7 φορές ως Female. Όσον αφορά τους άνδρες (Male), ο αλγόριθμος τούς προβλέπει πολύ σωστά (2355 σωστές προβλέψεις επί συνόλου 2439 ανδρών) με ποσοστό επιτυχημένης πρόβλεψης 96,5%. Όσον αφορά, όμως, τις γυναίκες (Female), ο αλγόριθμος δεν τις προβλέπει καθόλου σωστά (246 σωστές προβλέψεις επί συνόλου 714) με ποσοστό επιτυχημένης πρόβλεψης 34,5%.



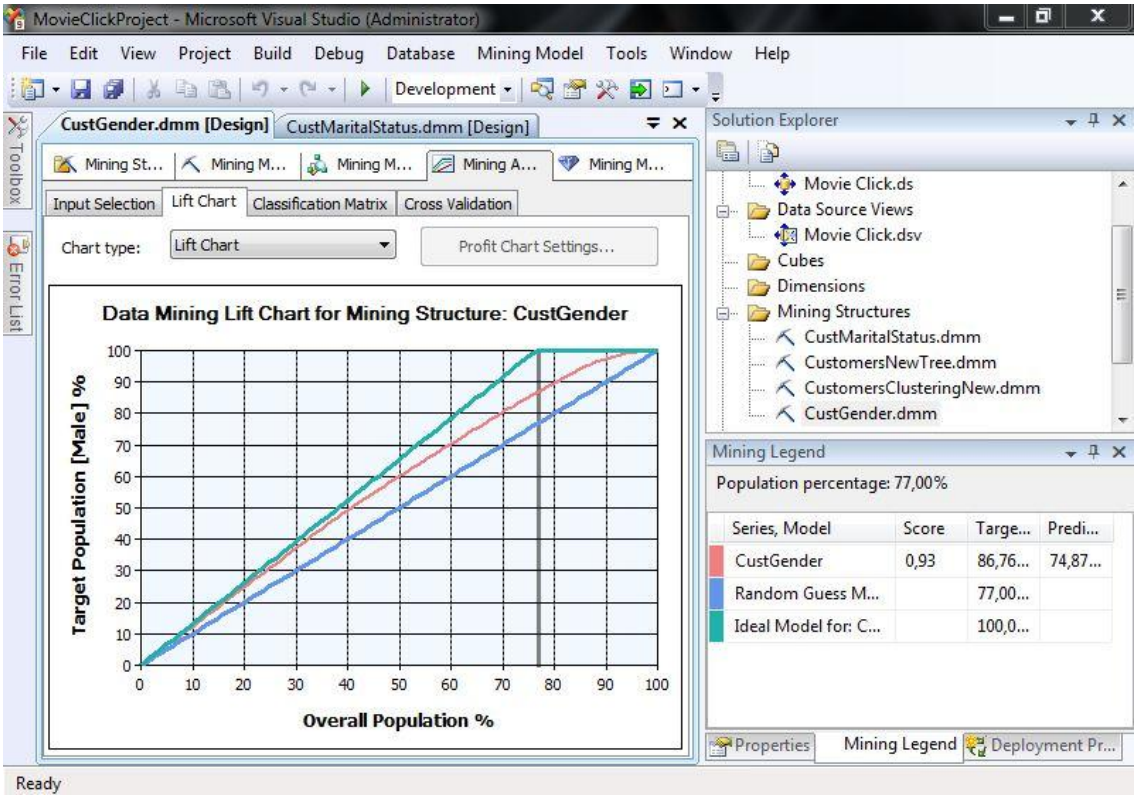
Εικόνα 7.32

10. Στη συνέχεια, θα προσδιορίσουμε μια συγκεκριμένη κατάσταση που θέλουμε να προβλέψουμε. Το χαρακτηριστικό που μας ενδιαφέρει παίρνει μόνο δύο τιμές (Male και Female), οπότε έχουμε μόνο αυτές τις επιλογές. Επιστρέφουμε στην καρτέλα Input Selection, όπως φαίνεται στην Εικόνα 7.33. Στο πεδίο Predictable Column Name επιλέγουμε την τιμή Gender. Στο πεδίο Predict Value επιλέγουμε την τιμή Male.



Εικόνα 7.33

11. Τέλος, επιλέγουμε πάλι Lift Chart, ώστε να εμφανιστεί η γραφική απεικόνιση του μοντέλου. Σ' αυτήν την περίπτωση, όπως φαίνεται στην Εικόνα 7.34, ο άξονας X απεικονίζει το ποσοστό του συνολικού πληθυσμού, ενώ ο άξονας Y το ποσοστό των ανδρών που έχει προβλεφθεί σωστά. Η γραμμή με κλίση 45° απεικονίζει το τυχαίο μοντέλο, ενώ η γραμμή που βρίσκεται υψηλότερα από τις υπόλοιπες δείχνει το ιδανικό μοντέλο. Η άλλη γραμμή αντιπροσωπεύει το δικό μας μοντέλο. Στη συγκεκριμένη περίπτωση, το ιδανικό μοντέλο πετυχαίνει το 100% των προβλέψεων στο 77% του συνολικού πληθυσμού. Στον πίνακα Mining Legend βλέπουμε ότι το δικό μας μοντέλο έχει Score 0.93 και προβλέπει με ακρίβεια 86,76% (δηλαδή, σχετικά καλά).



Εικόνα 7.34

7.4. Ασκήσεις στην παραμετροποίηση του αλγορίθμου δέντρου απόφασης

1. Να αλλάξετε στο ήδη δημιουργηθέν μοντέλο (όπως φαίνεται στην Εικόνα 7.13) την τιμή της παραμέτρου COMPLEXITY_PENALTY, ορίζοντας την σε 0.001. Να εμφανίσετε και να σχολιάσετε τα παρακάτω:

   a) το νέο δέντρο απόφασης που θα δημιουργηθεί,
   b) το dependency network με τα κύρια χαρακτηριστικά που προσδιορίζουν το φύλο (gender),
   c) το Lift Chart και το ποσοστό των ανδρών που προβλέπονται σωστά από το νέο μοντέλο,
   d) το Classification Matrix.

   Τέλος, να συγκριθεί το νέο μοντέλο με αυτό που δημιουργούν οι προεπιλεγμένες (default) τιμές του αλγορίθμου Decision Tree.

2. Να επαναληφθεί η άσκηση 1, δίνοντας την τιμή 0.999 στην παράμετρο COMPLEXITY_PENALTY του αλγορίθμου Decision Tree.

3. Να επαναληφθεί η άσκηση 1, δίνοντας την τιμή 200 στην παράμετρο COMPLEXITY_PENALTY, η οποία υποχρεώνει να βρίσκονται τουλάχιστον διακόσιες κατ' ελάχιστο περιπτώσεις σε κάθε φύλλο του δέντρου.

4. Να επαναληφθεί η άσκηση 1, επιλέγοντας τη δημιουργία ενός μη δυαδικού δέντρου απόφασης (multi-way) και διατηρώντας τις προεπιλεγμένες τιμές στις υπόλοιπες παραμέτρους του αλγορίθμου decision tree.

5. Να επαναληφθεί η άσκηση 1, επιλέγοντας τη δημιουργία ενός δέντρου απόφασης, το οποίο θα βασίζεται στο μέτρο της Εντροπίας (τρόπος υπολογισμού της καταλληλότητας ενός πεδίου/ χαρακτηριστικού ως κόμβου του δέντρου).

6. Να επαναληφθεί η άσκηση 1, επιλέγοντας τη δημιουργία ενός δέντρου απόφασης, το οποίο θα βασίζεται στο μέτρο της Εντροπίας και θα επιτρέπει πενήντα τουλάχιστον περιπτώσεις στο καθένα από τα φύλλα του.

7.5. Λύσεις ασκήσεων στην παραμετροποίηση του αλγορίθμου δέντρων απόφασης

Άσκηση 1

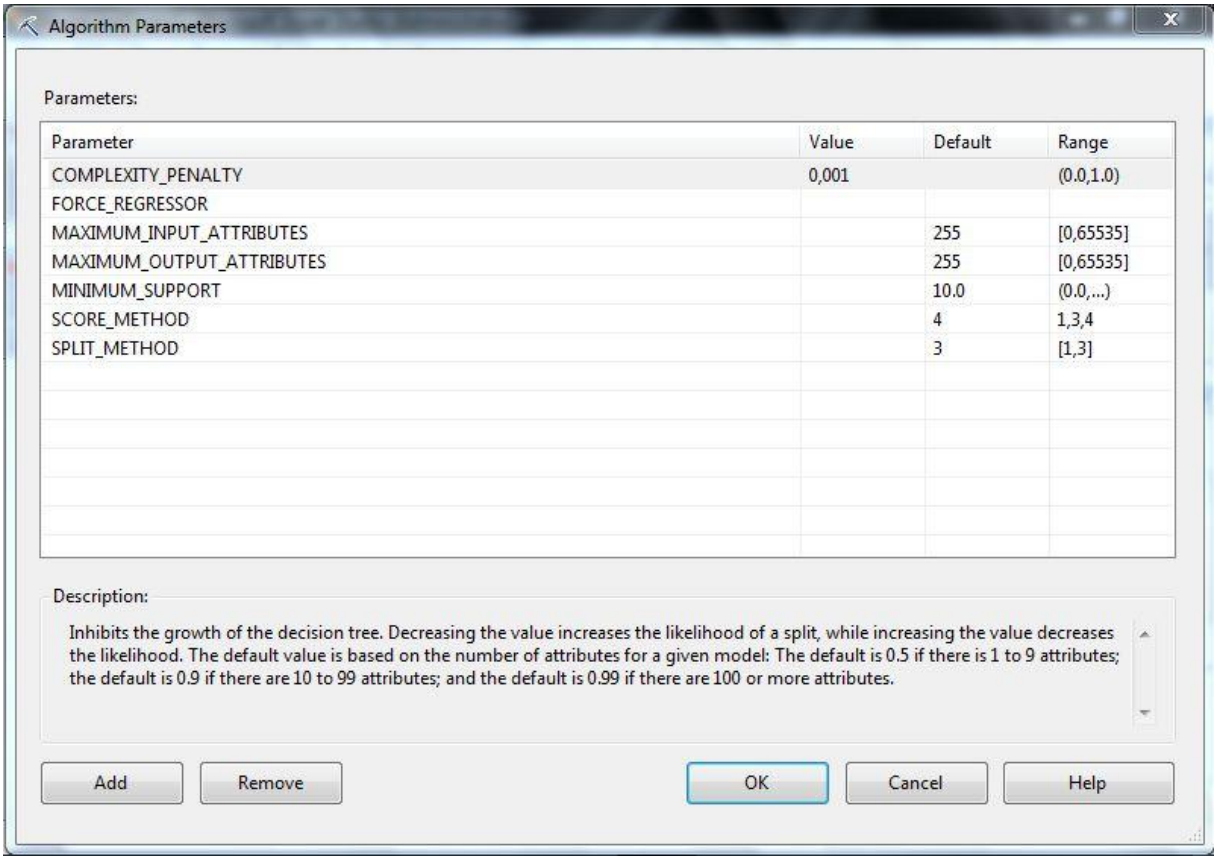
Να αλλάξετε στο ήδη δημιουργηθέν μοντέλο (όπως φαίνεται στην Εικόνα 7.13) την τιμή της παραμέτρου COMPLEXITY_PENALTY, ορίζοντας την σε 0.001. Να εμφανίσετε και να σχολιάσετε τα παρακάτω:

a) το νέο δέντρο απόφασης που θα δημιουργηθεί,
b) το dependency network με τα κύρια χαρακτηριστικά που προσδιορίζουν το φύλο (gender),
c) το Lift Chart και το ποσοστό των ανδρών που προβλέπονται σωστά από το νέο μοντέλο,
d) το Classification Matrix.

Τέλος, να συγκριθεί το νέο μοντέλο με αυτό που δημιουργούν οι προεπιλεγμένες (default) τιμές του αλγορίθμου Desicion Tree.

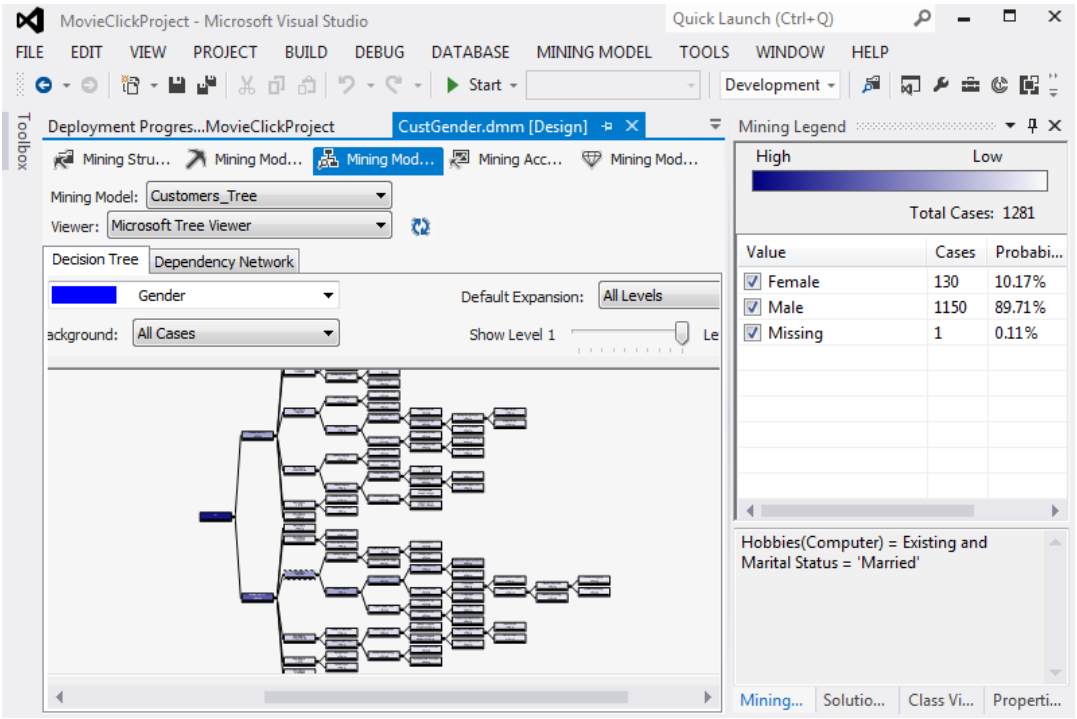
Λύση

1. Αλλάζουμε την τιμή της παραμέτρου σε 0.001, όπως φαίνεται στην Εικόνα 7.35, και κάνουμε run το μοντέλο.



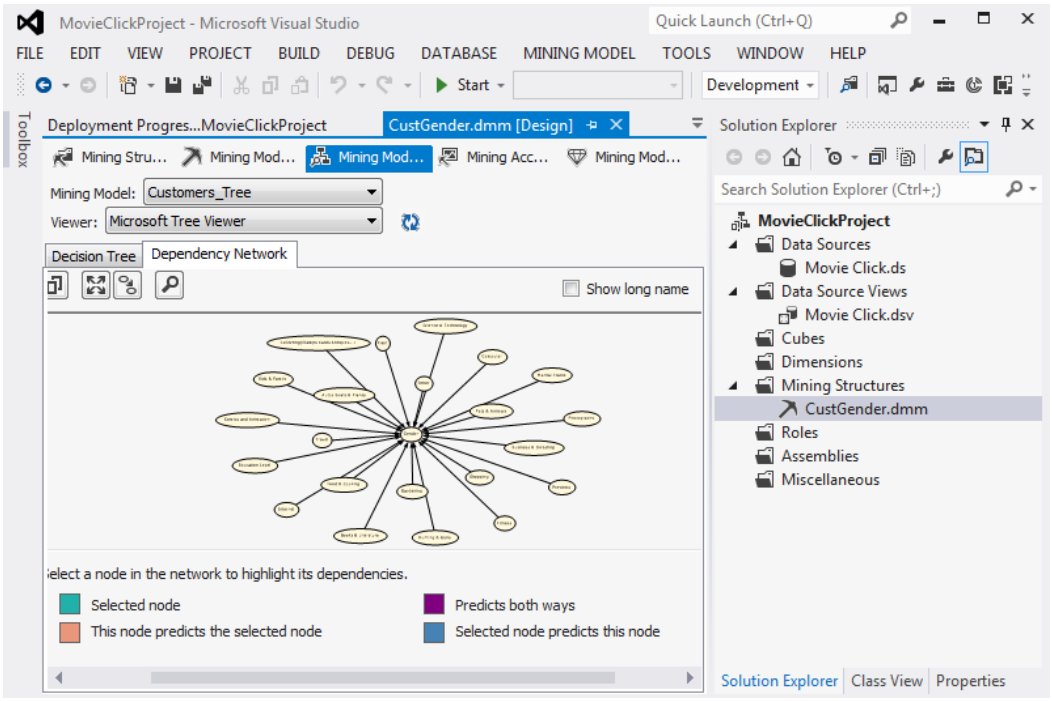
Εικόνα 7.35

2. Στην Εικόνα 7.36 παρατηρούμε ότι το δέντρο είναι πολύ πιο πολύπλοκο σε σχέση μ' αυτό του προηγούμενου μοντέλου (βλέπε Εικόνα 7.20). Ωστόσο, τα συμπεράσματα στα οποία καταλήγουμε είναι παρόμοια με αυτά του προηγούμενου μοντέλου. Ακόμα και η διάκριση στα πρώτα στάδια δημιουργίας του δέντρου είναι η ίδια.



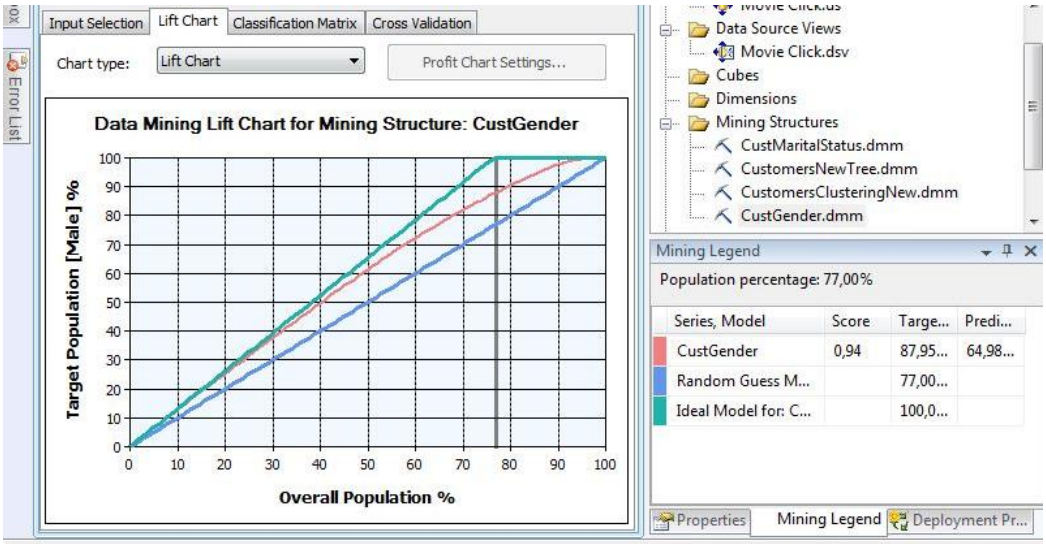
Εικόνα 7.36

3. Οι παράμετροι που καθορίζουν το φύλο έχουν αυξηθεί, όπως φαίνεται στην Εικόνα 7.37, χωρίς αυτό να σημαίνει απαραίτητα ότι η ικανότητα πρόβλεψης του δέντρου έχει βελτιωθεί.



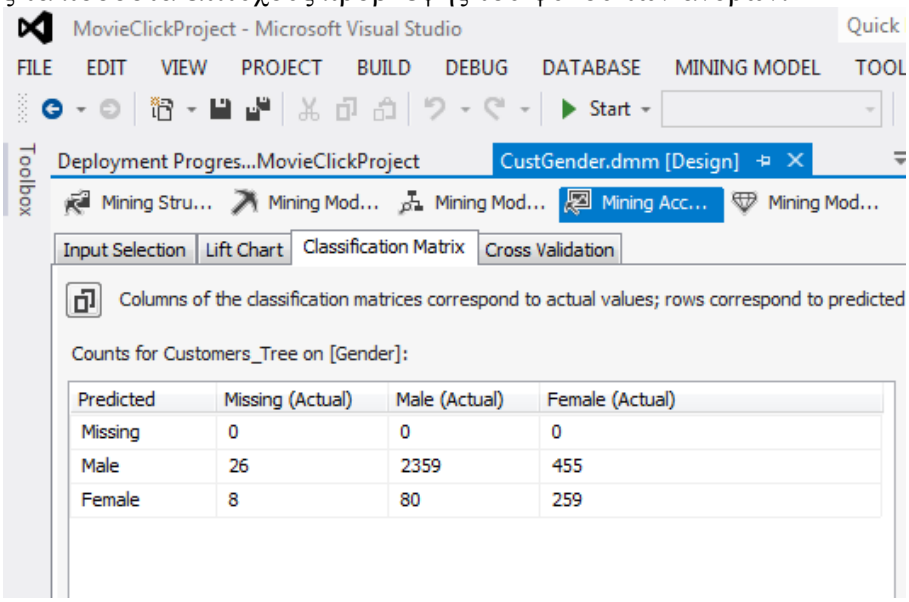
Εικόνα 7.37

4. Στο Lift Chart, όπως φαίνεται στην Εικόνα 7.38, με δείγμα 77% του πληθυσμού μπορούμε να προβλέψουμε σωστά το 87,95% των ανδρών του δείγματος. Το ποσοστό αυτό είναι καλύτερο απ' αυτό που είχαμε με τιμή 0,9 στην παράμετρο COMPLEXITY_PENALTY (προβλέπαμε σωστά μόνο το 86,76% των ανδρών), αλλά το δέντρο είναι πολύ πιο πολύπλοκο και, συνεπώς, πιο δύσκολα ερμηνεύσιμο. Το Score είναι επίσης καλύτερο (0,94) σε σχέση με το προηγούμενο (0.93) (βλέπε Εικόνα 7.34).



Εικόνα 7.38

5. Στο Classification Matrix, όπως φαίνεται στην Εικόνα 7.39, βλέπουμε να αυξάνονται ελαφρώς τα ποσοστά επιτυχούς πρόβλεψης του φύλου των ανδρών.



Εικόνα 7.39

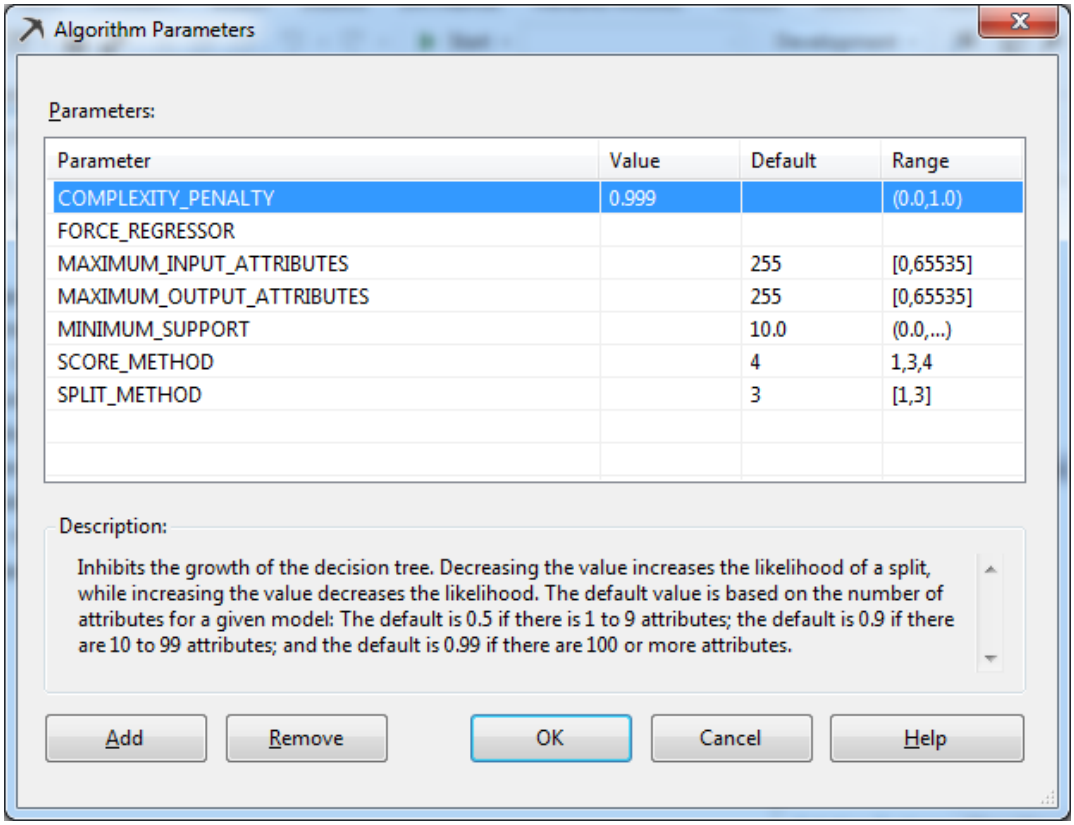
6. Συμπερασματικά, το δέντρο μας έχει πλέον μεγαλύτερη ακρίβεια. Πρέπει να τονίσουμε ότι ένα πιο πολύπλοκο δέντρο μπορεί σχεδόν πάντα να εξασφαλίζει ελαφρώς καλύτερη πρόβλεψη. Η αύξηση της πολυπλοκότητάς του, όμως, δεν είναι πάντα ανάλογη της βελτίωσης της πρόβλεψης του δέντρου, γεγονός που το κάνει μη εύκολα ερμηνεύσιμο. Γι' αυτό, δεν πρέπει να θεωρούμε πάντα ότι τα πολυπλοκότερα δέντρα είναι τα καλύτερα.

Άσκηση 2

Να επαναληφθεί η άσκηση 1, δίνοντας την τιμή 0.999 στην παράμετρο COMPLEXITY_PENALTY του αλγορίθμου Decision Tree.

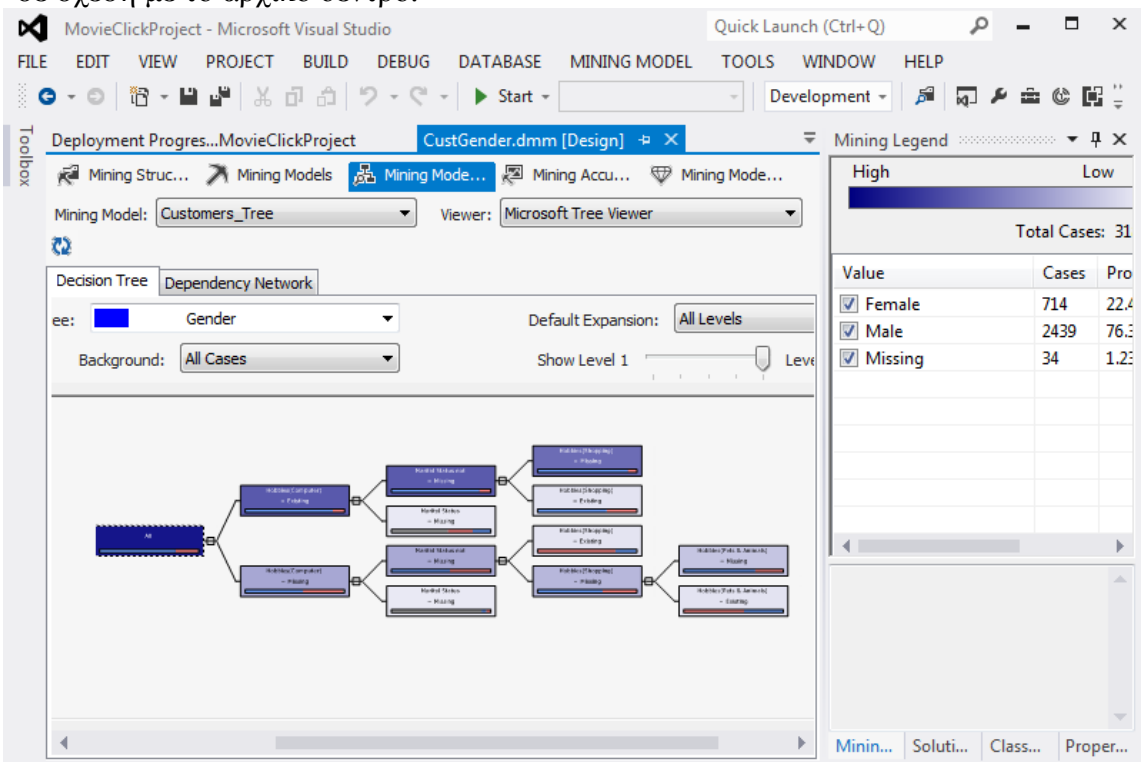
Λύση

1. Αλλάζουμε την τιμή της παραμέτρου σε 0.999, όπως φαίνεται στην Εικόνα 7.40, και κάνουμε run το μοντέλο.



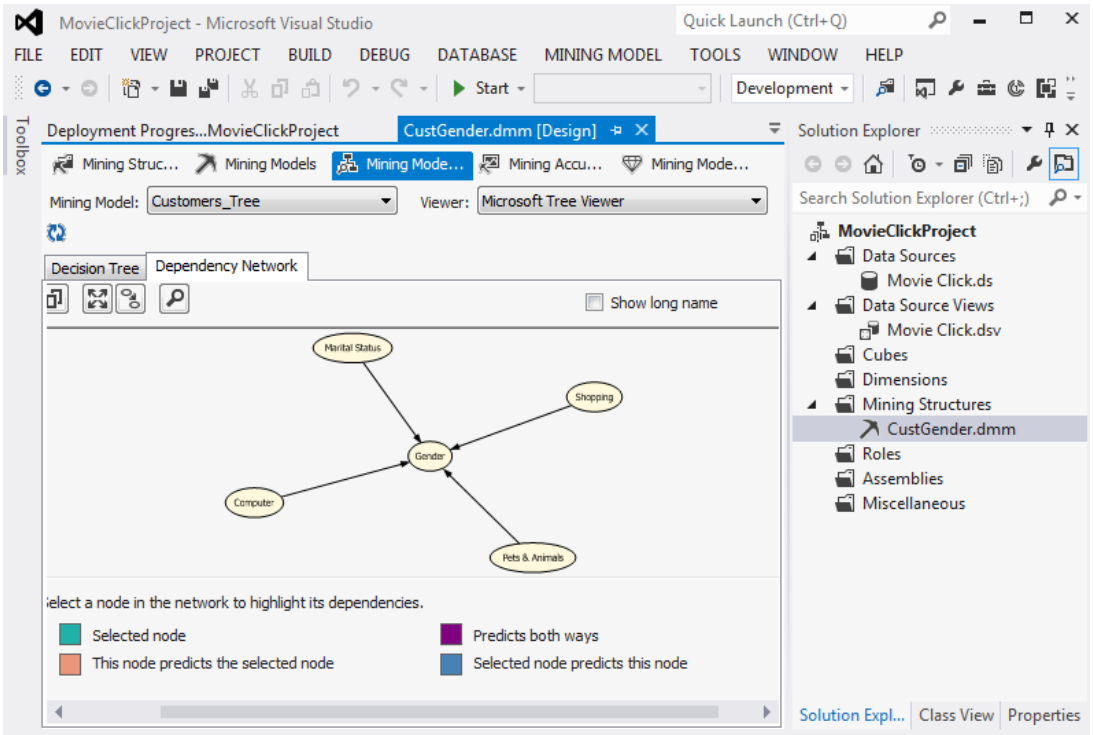
Εικόνα 7.40

2. Όπως φαίνεται στην Εικόνα 7.41, το δέντρο που προκύπτει είναι πολύ πιο απλό, ακόμα και σε σχέση με το αρχικό δέντρο.



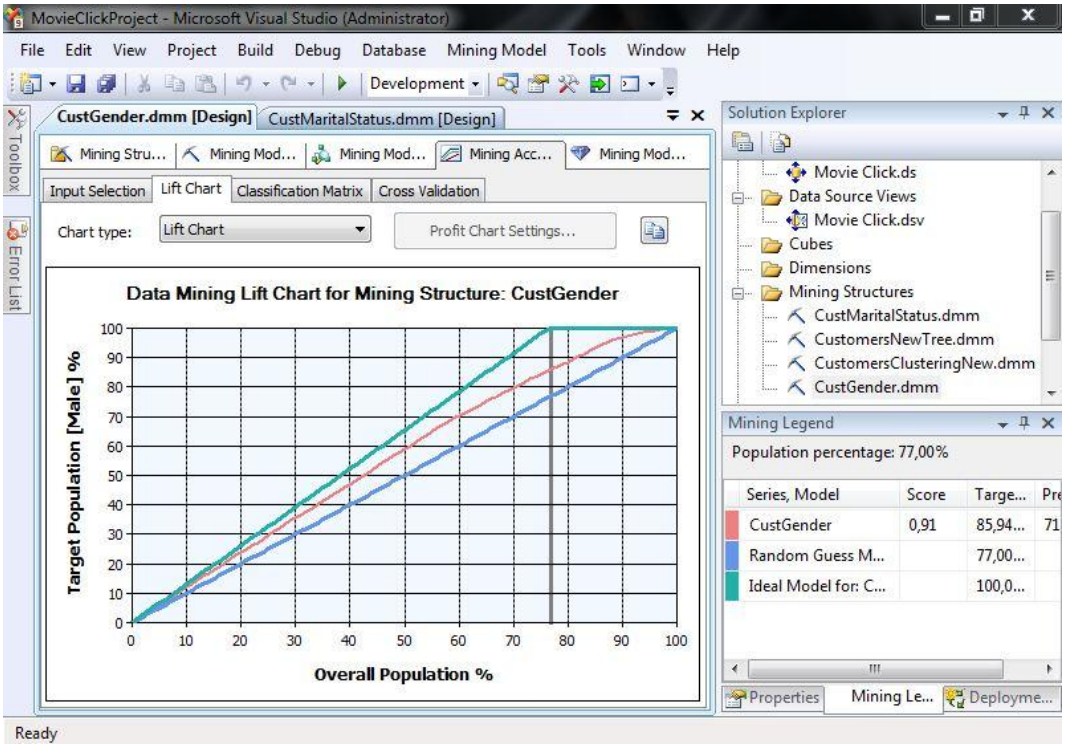
Εικόνα 7.41

3. Μόλις τέσσερα χαρακτηριστικά πλέον χρησιμοποιούνται για την πρόβλεψη του Gender, όπως φαίνεται στην Εικόνα 7.42.



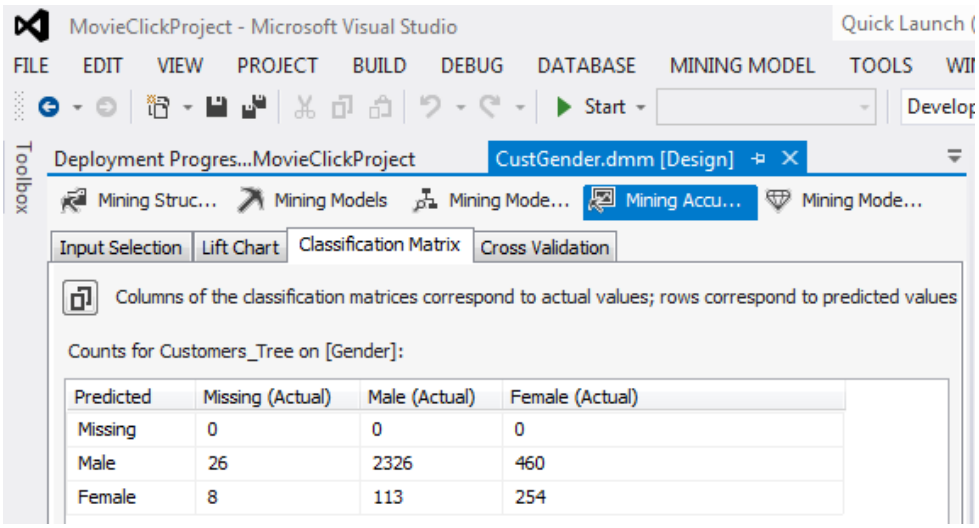
Εικόνα 7.42

4. Στο Lift Chart, όπως φαίνεται στην Εικόνα 7.43, με 77% του πληθυσμού προβλέπουμε σωστά το 85,94% των ανδρών του δείγματος, ποσοστό που είναι μικρότερο απ' αυτό που έδωσε το δέντρο με τις προεπιλεγμένες τιμές στις παραμέτρους του αλγορίθμου, ενώ το Score είναι 0,91. Τονίζεται ότι με το αρχικό δέντρο απόφασης προβλέπαμε σωστά το 86,76% των ανδρών.



Εικόνα 7.43

5. Στο Classification Matrix, όπως φαίνεται στην Εικόνα 7.44, βλέπουμε ότι τα στατιστικά στοιχεία είναι περίπου στα ίδια επίπεδα.



Εικόνα 7.44

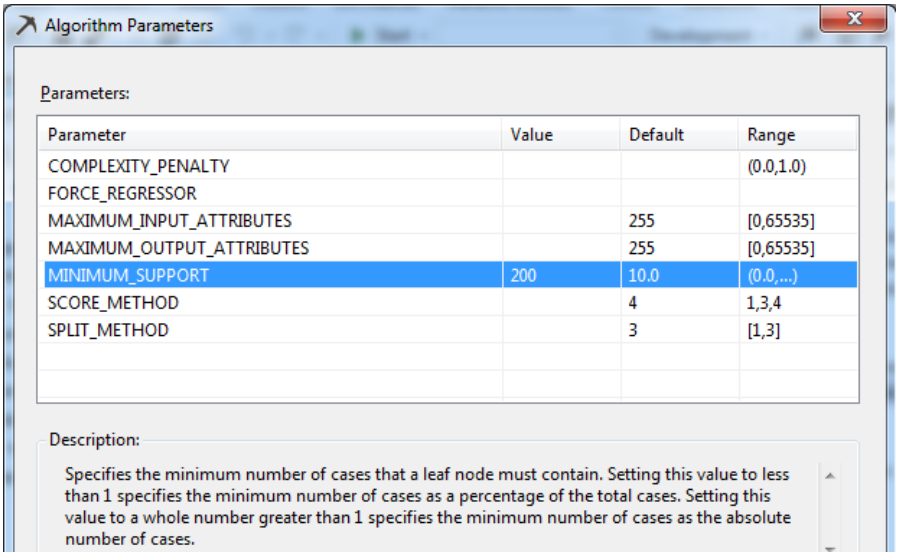
6. Συμπερασματικά, όταν ένα δέντρο είναι υπερβολικά μικρό, οι προβλέψεις του δεν είναι πολύ ακριβείς, διότι δεν λαμβάνει υπόψη του όλα τα διαθέσιμα χαρακτηριστικά που πιθανόν να επηρεάζουν την δεσμευμένη μεταβλητή (στην περίπτωση μας, το Φύλο).

Άσκηση 3

Να επαναληφθεί η άσκηση 1, δίνοντας την τιμή 200 στην παράμετρο COMPLEXITY_PENALTY, η οποία υποχρεώνει να βρίσκονται τουλάχιστον διακόσιες κατ' ελάχιστο περιπτώσεις σε κάθε φύλλο του δέντρου.

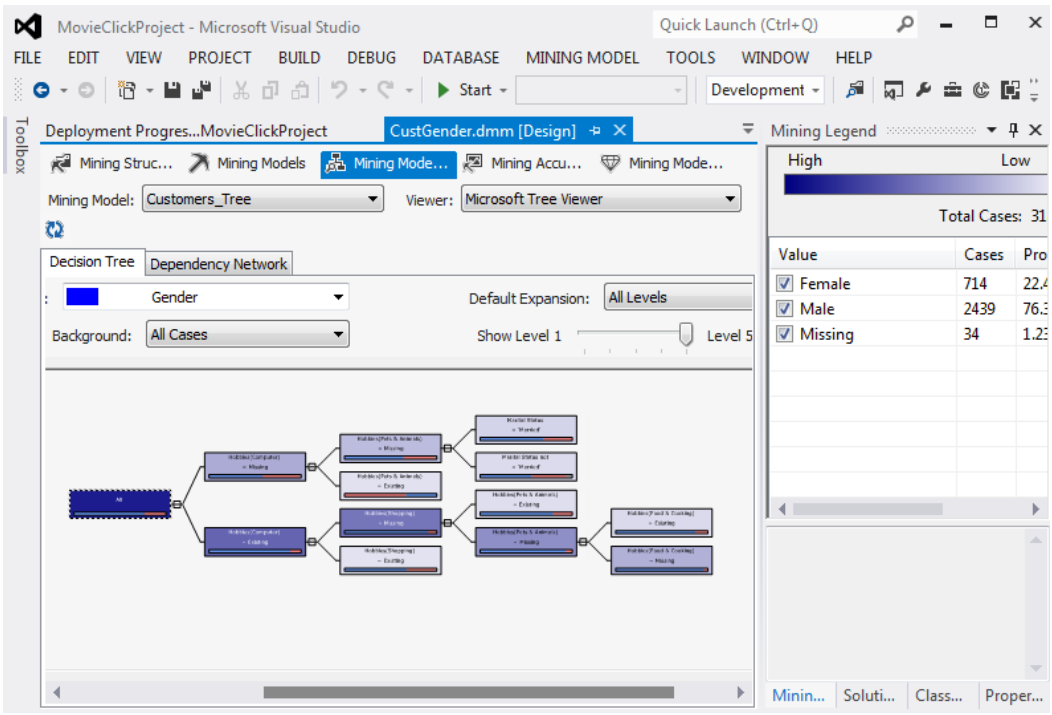
Λύση

1. Αλλάζουμε την τιμή της παραμέτρου σε 200, όπως φαίνεται στην Εικόνα 7.45, και κάνουμε run το μοντέλο.



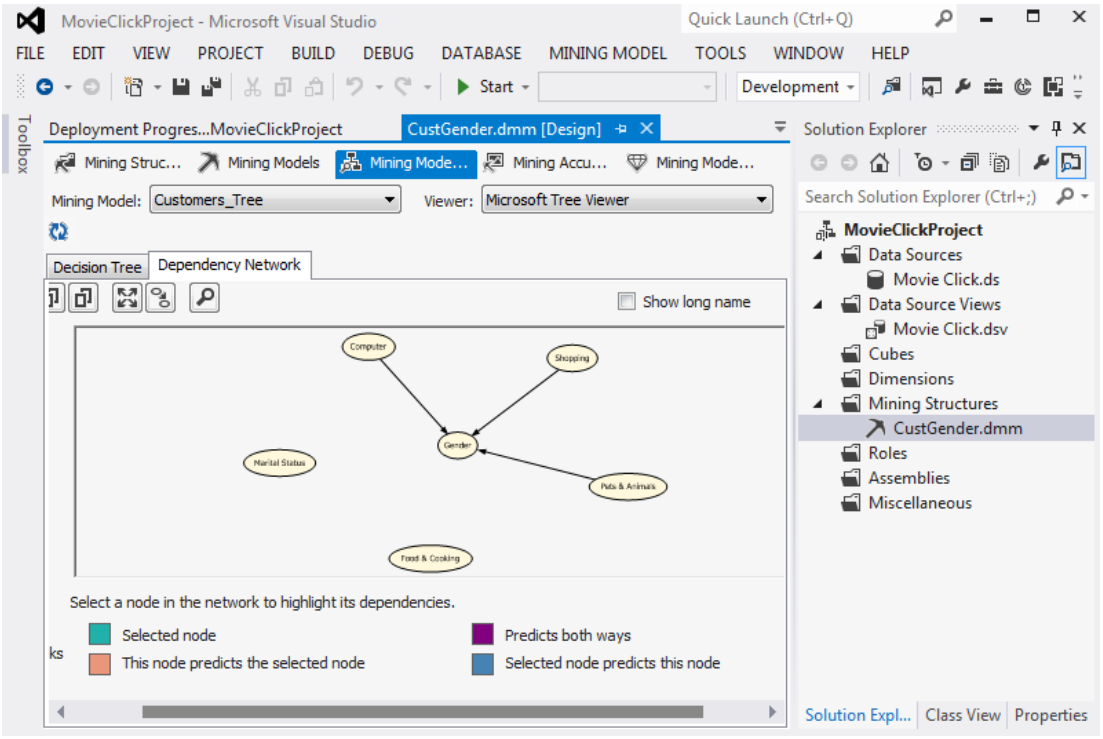
Εικόνα 7.45

2. Παρατηρούμε, όπως φαίνεται στην Εικόνα 7.46, ότι το δέντρο που δημιουργείται είναι απλό και ότι κάθε φύλλο του δέντρου ενσωματώνει περισσότερες από 200 περιπτώσεις (εγγραφές που βρίσκονται στον πίνακα case).



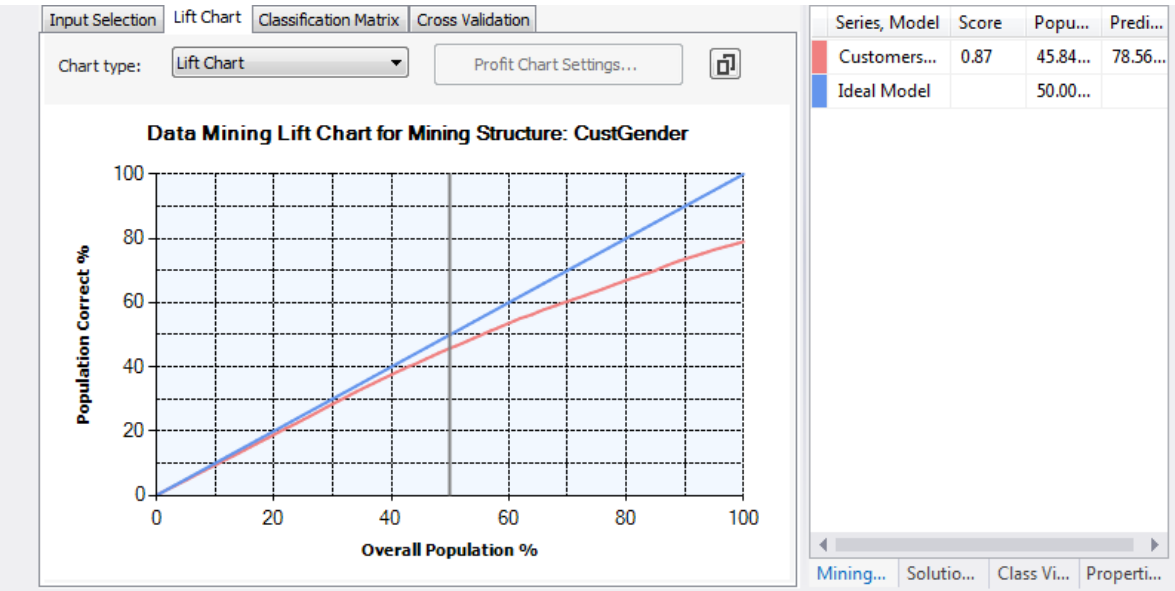
Εικόνα 7.46

3. Ο περιορισμός των διακοσίων περιπτώσεων είχε ως αποτέλεσμα να αλλάξουν τα βασικά κριτήρια με τα οποία γίνεται η διάκριση των πελατών στο δέντρο μας. Πράγματι, όπως φαίνεται στην Εικόνα 7.47, η οικογενειακή κατάσταση δεν θεωρείται τόσο σημαντικό κριτήριο όσο είναι τα κατοικίδια, κάτι που δεν εμφανίστηκε σε προηγούμενη παραμετροποίηση του μοντέλου μας.



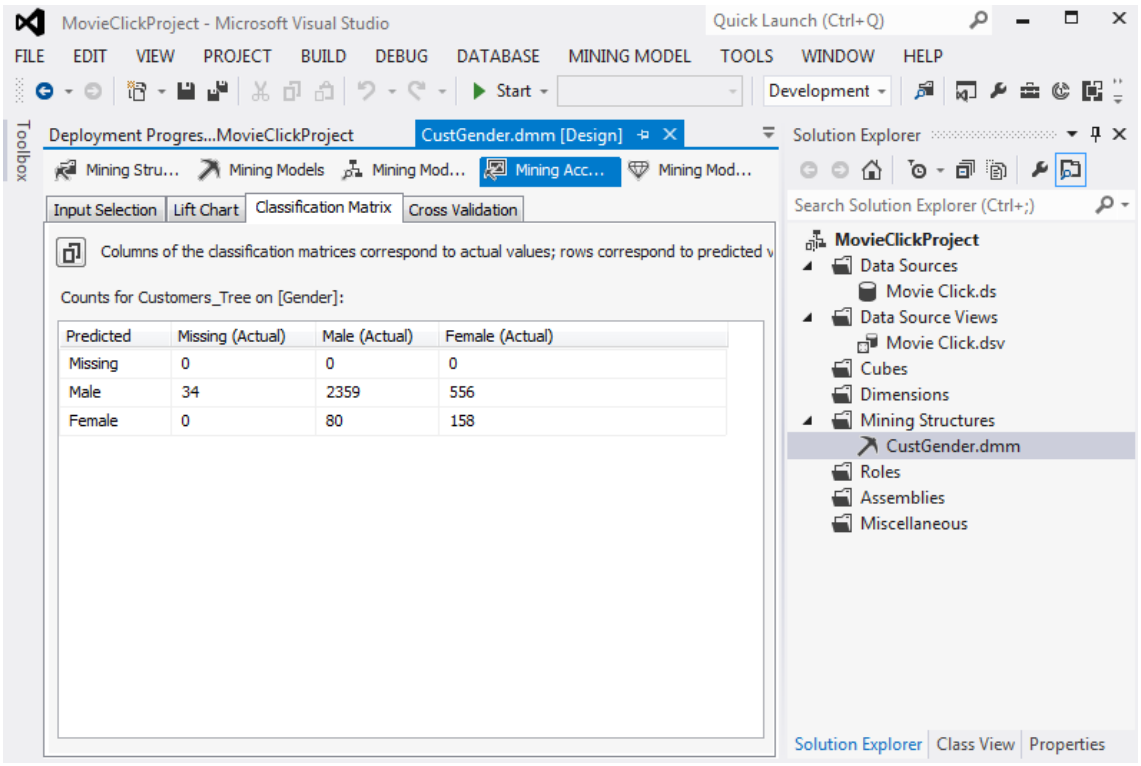
Εικόνα 7.47

4. Στο Lift Chart, όπως φαίνεται στην Εικόνα 7.48, σε δείγμα 50% του συνολικού πληθυσμού, προβλέπεται πλέον σωστά το 45,84% των ανδρών του δείγματος.



Εικόνα 7.48

12. Στο Classification Matrix, όπως φαίνεται στην Εικόνα 7.49, βλέπουμε ότι τα ποσοστά πρόβλεψης των ανδρών ελαφρώς αυξάνονται έναντι του αρχικού μοντέλου πρόβλεψης. Όσον αφορά όμως τις γυναίκες (Female), τα ποσοστά πρόβλεψης μειώνονται δραματικά (μόνο 158 σωστές προβλέψεις επί συνόλου 714) με ποσοστό επιτυχημένης πρόβλεψης μόλις 22,1%.



Εικόνα 7.49

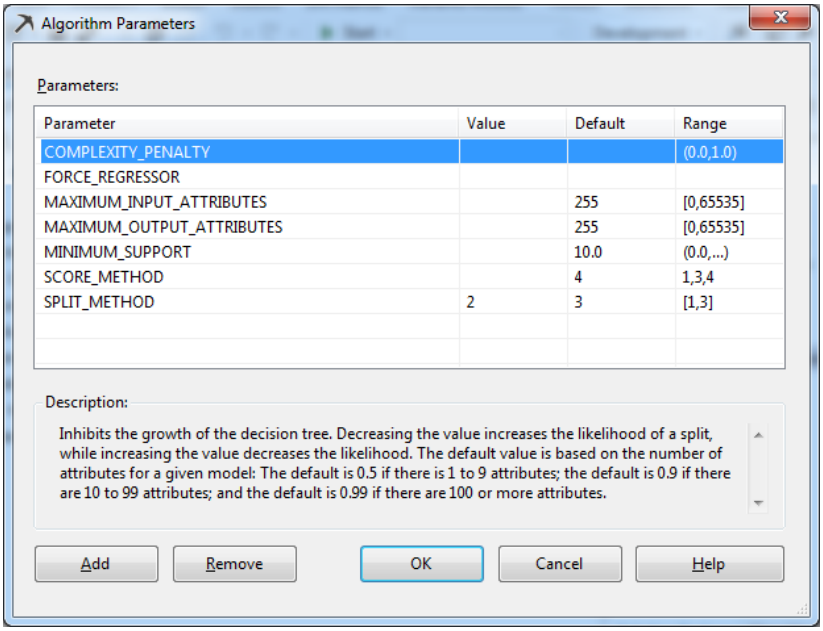
5. Συμπερασματικά, αυτή η παράμετρος πρέπει να χρησιμοποιείται μόνο σε περιπτώσεις που το δέντρο αποτελείται από φύλλα με πολύ μικρό αριθμό περιπτώσεων. Σ' αυτήν την περίπτωση, μπορούμε να δημιουργήσουμε ένα πιο συμπαγές δέντρο που θα βοηθήσει να εξάγουμε πιο σαφή συμπεράσματα.

Άσκηση 4

Να επαναληφθεί η άσκηση 1, επιλέγοντας τη δημιουργία ενός μη δυαδικού δέντρου απόφασης (multi-way) και διατηρώντας τις προεπιλεγμένες τιμές στις υπόλοιπες παραμέτρους του αλγορίθμου decision tree.

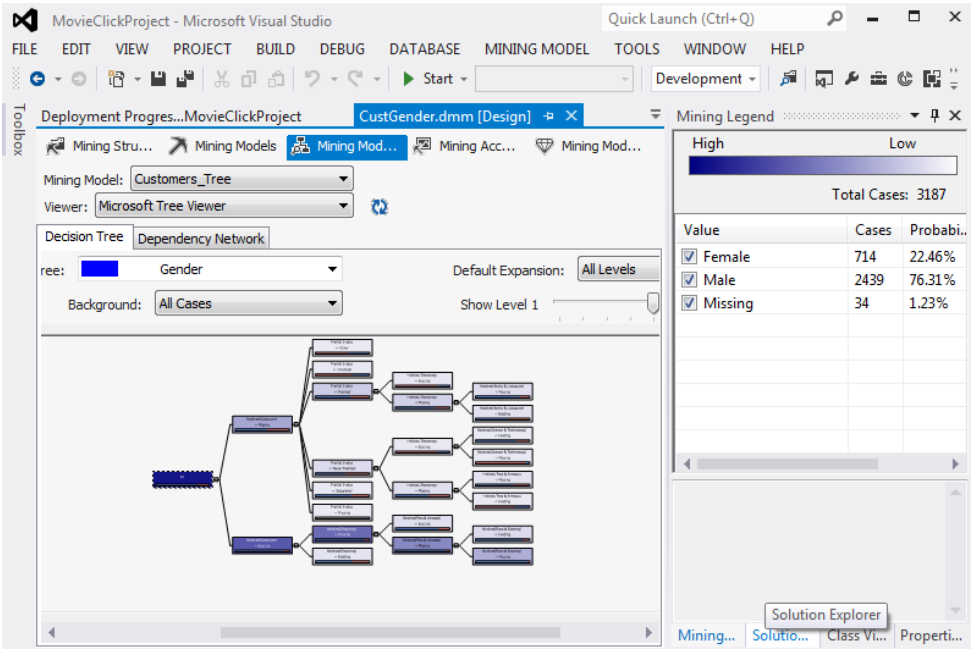
Λύση

1. Επιλέγουμε στην παράμετρο SPLIT_METHOD την τιμή 2, όπως φαίνεται στην Εικόνα 7.50.



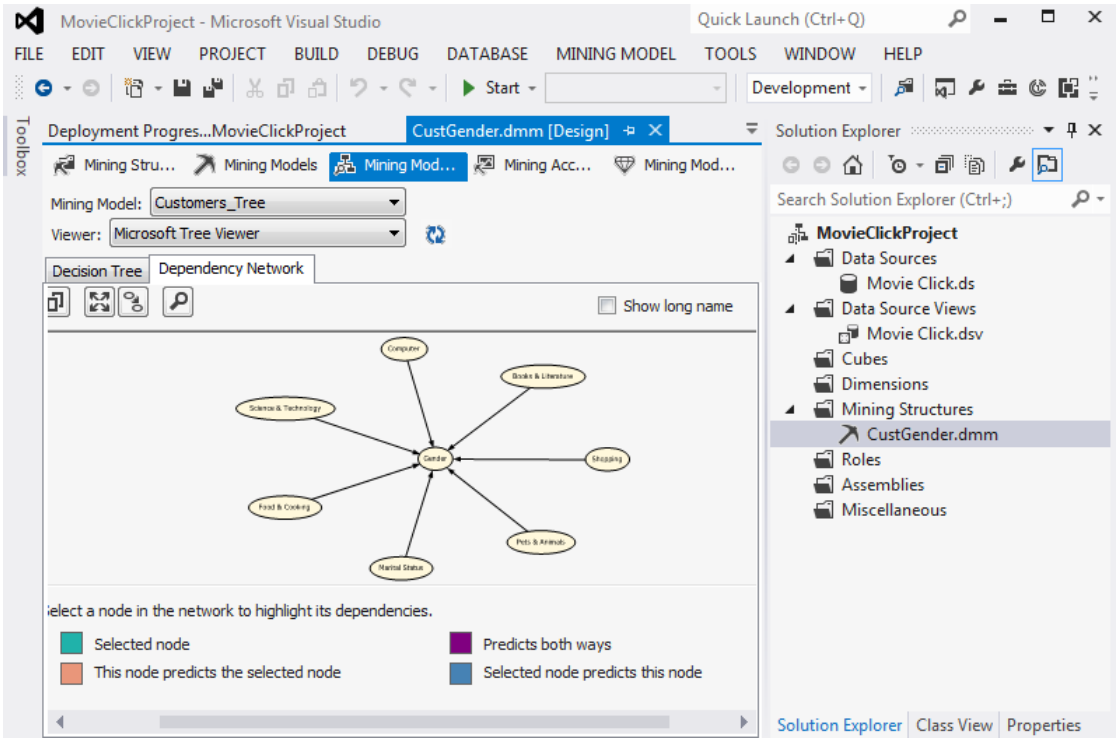
Εικόνα 7.50

2. Παρατηρούμε, όπως φαίνεται στην Εικόνα 7.51, ότι, από τη στιγμή που το δέντρο μας δεν είναι δυαδικό, αφενός η διάκριση των περιπτώσεων δεν γίνεται με σαφήνεια και, αφετέρου, πολλά χαρακτηριστικά επαναλαμβάνονται σε κατώτερα επίπεδα του δέντρου.



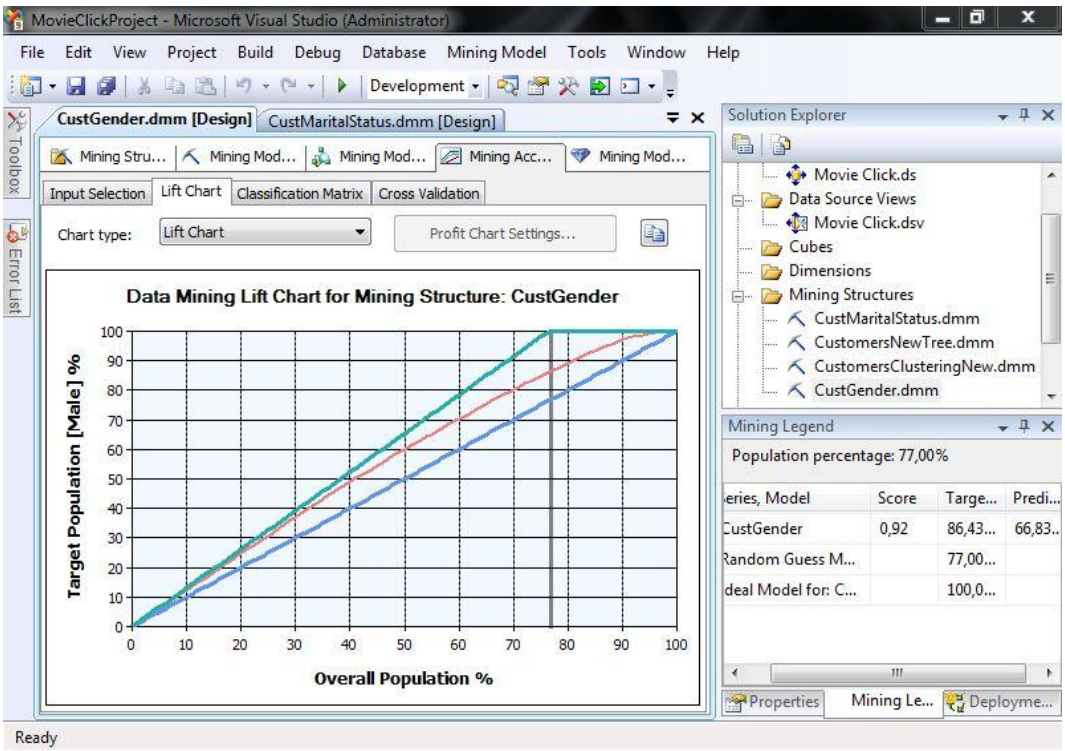
Εικόνα 7.51

3. Τα χαρακτηριστικά που προσδιορίζουν το Gender είναι πλέον περισσότερα απ' αυτά του δυαδικού δέντρου, όπως φαίνεται στην Εικόνα 7.52.



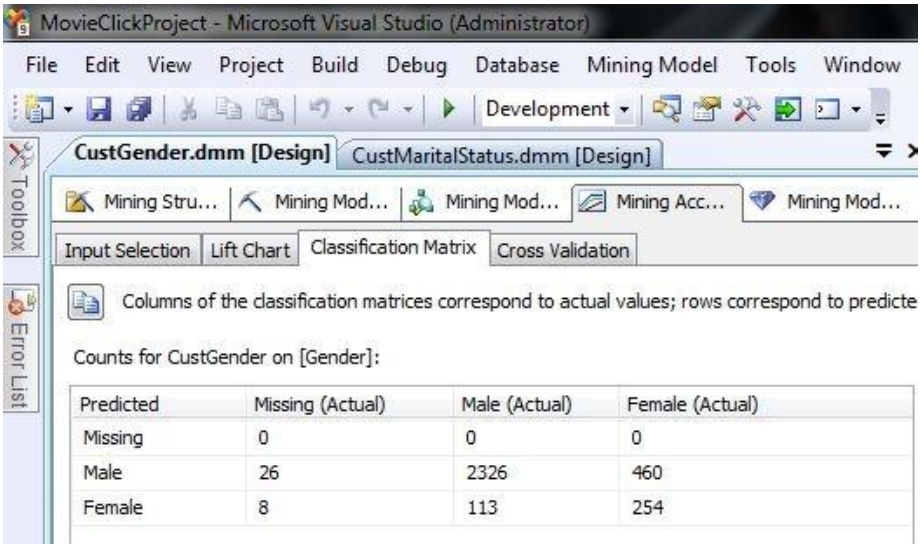
Εικόνα 7.52

4. Στο Lift Chart, όπως φαίνεται στην Εικόνα 7.53, σε δείγμα 77% του πληθυσμού προβλέπεται σωστά το 86,43% των ανδρών του δείγματος.



Εικόνα 7.53

5. Τα ποσοστά ακρίβειας είναι περίπου τα ίδια με το αρχικό δέντρο, κάτι που φαίνεται και από το classification matrix της Εικόνας 7.54.



Εικόνα 7.54

6. Συμπερασματικά, στα μη δυαδικά δέντρα απόφασης, η διάκριση των περιπτώσεων δεν γίνεται με σαφήνεια και πολλά χαρακτηριστικά επαναλαμβάνονται σε κατώτερα επίπεδα του δέντρου. Σε γενικές, όμως, γραμμές η ακρίβεια είναι ίδια με τα δυαδικά δέντρα απόφασης.

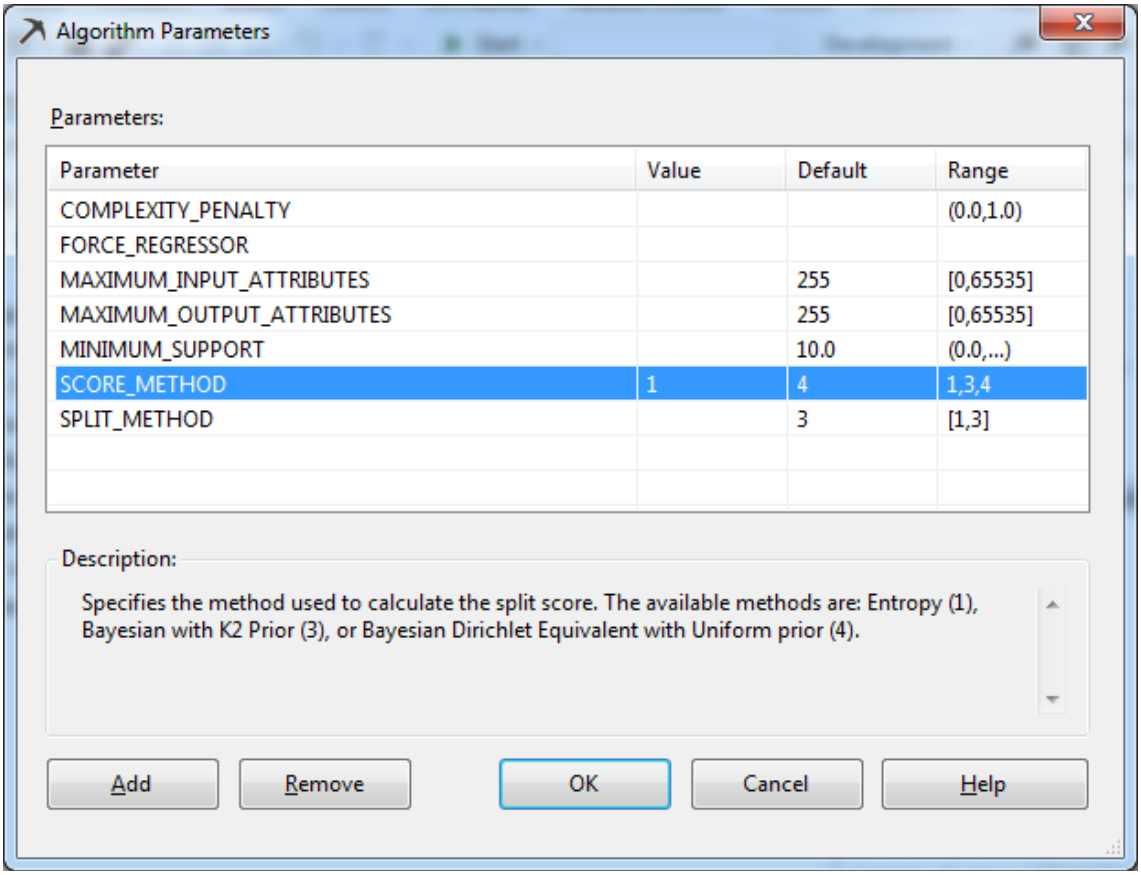
.

Άσκηση 5

Να επαναληφθεί η άσκηση 1, επιλέγοντας τη δημιουργία ενός δέντρου απόφασης, το οποίο θα βασίζεται στο μέτρο της Εντροπίας (τρόπος υπολογισμού της καταλληλότητας ενός πεδίου/χαρακτηριστικού ως κόμβου του δέντρου).

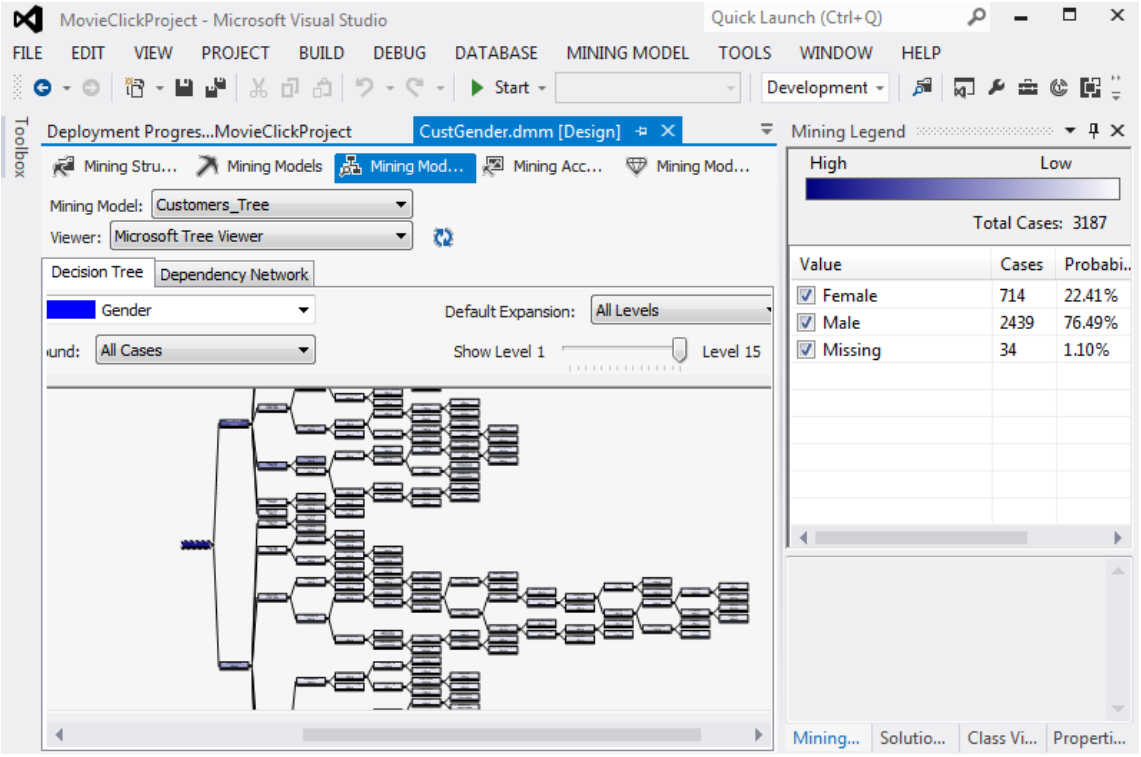
Λύση

1. Η εντροπία μετράει τον βαθμό βεβαιότητας/αβεβαιότητας που δημιουργεί στο μοντέλο ένα χαρακτηριστικό έναντι των υπόλοιπων χαρακτηριστικών (ώστε να επιλεχθεί ή όχι να γίνει κόμβος του δέντρου απόφασης). Για να επιλέξουμε την εντροπία ως μέτρο, δίνουμε στην παράμετρο Score_Method την τιμή 1.



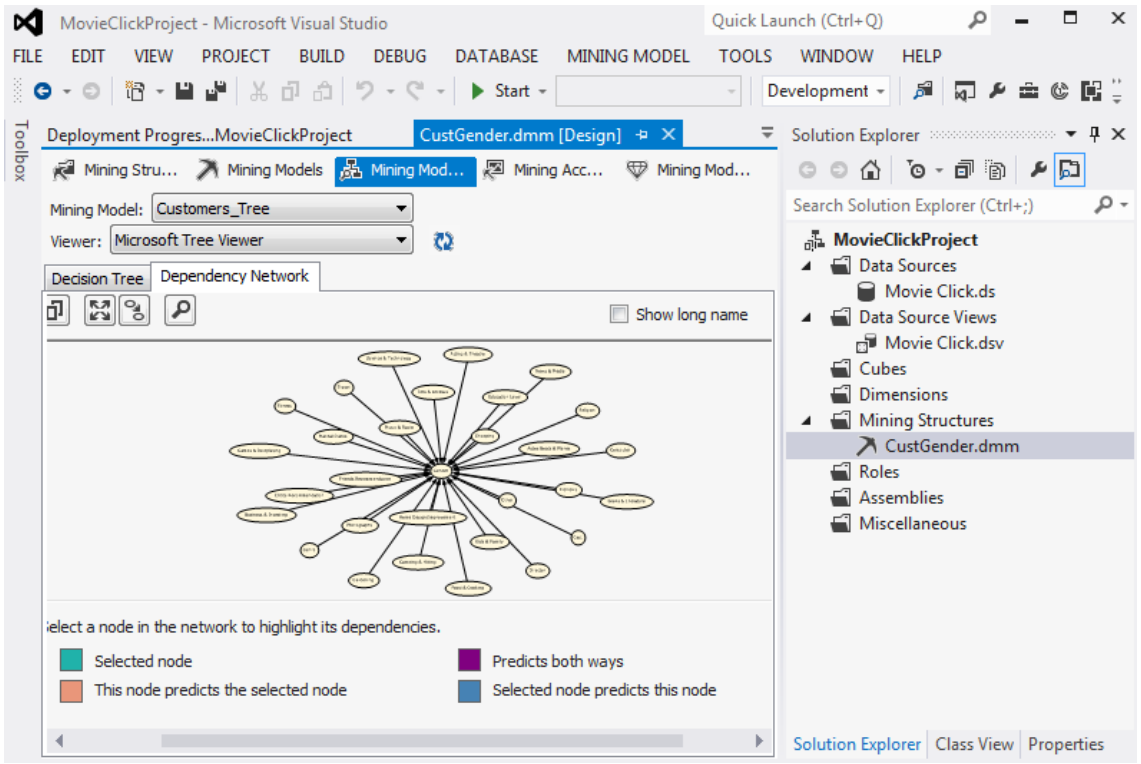
Εικόνα 7.55

2. Στην Εικόνα 7.56 παρατηρούμε ότι το δέντρο που προκύπτει είναι πολύπλοκο και αποτελείται από 14 επίπεδα.



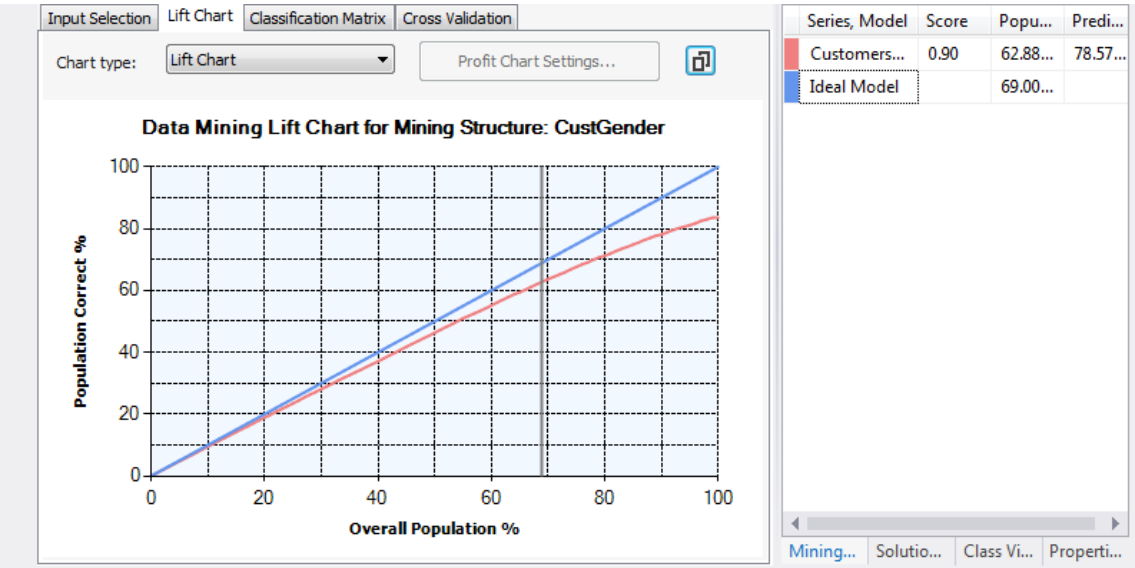
Εικόνα 7.56

3. Η τιμή που θέλουμε να προβλέψουμε εξαρτάται πλέον απ' όλα, σχεδόν, τα χαρακτηριστικά, όπως φαίνεται στην Εικόνα 7.57.



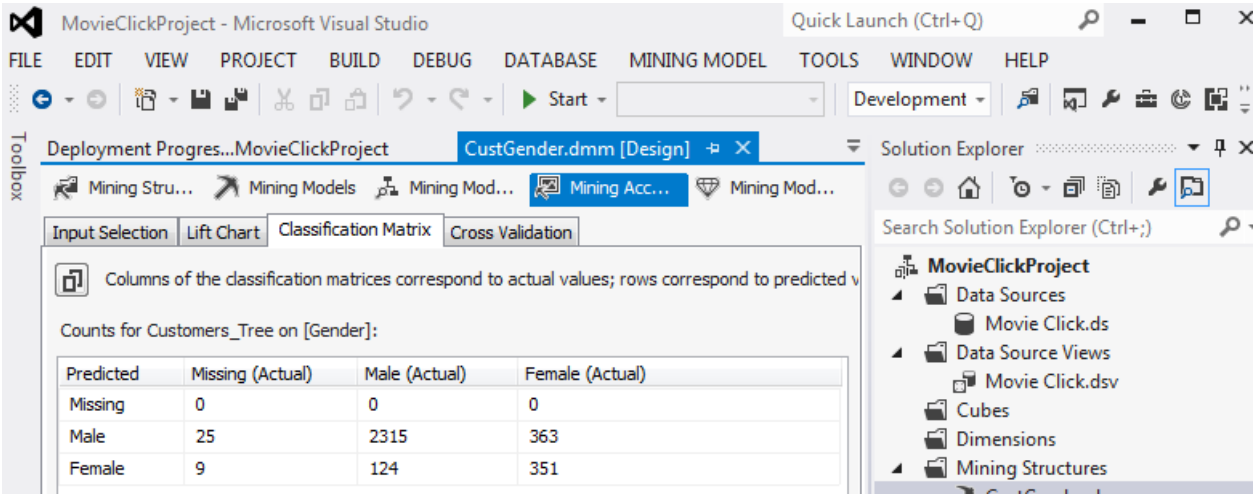
Εικόνα 7.57

4. Στο Lift Chart, όπως φαίνεται στην Εικόνα 7.58, σε δείγμα 77% του πληθυσμού προβλέπεται σωστά το 90,08% των ανδρών με Score 0,94. Αυτό το ποσοστό, που είναι το υψηλότερο που έχει εμφανιστεί στα παραδείγματά μας, αποδεικνύει ότι το δέντρο που παράγεται με την εντροπία μπορεί να προβλέψει με μεγαλύτερη ακρίβεια.



Εικόνα 7.58

5. Στο Classification Matrix, όπως φαίνεται στην Εικόνα 7.59, βλέπουμε ότι τα ποσοστά πρόβλεψης των γυναικών έχουν πλέον βελτιωθεί σημαντικά.



Εικόνα 7.59

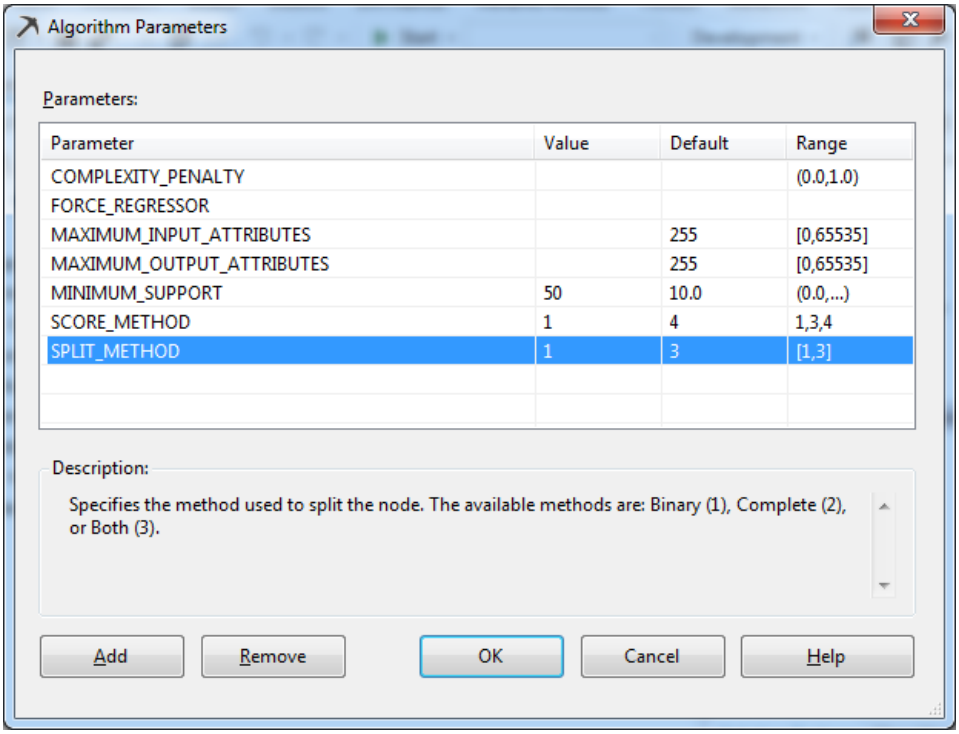
6. Συμπερασματικά, στην περίπτωση μας η εντροπία είναι ο πιο αποτελεσματικός τρόπος δημιουργίας ενός δέντρου απόφασης, αφού εξασφαλίζει την πιο ακριβή πρόβλεψη. Αυτό, όμως, δεν σημαίνει ότι είναι πάντα και ο καταλληλότερος τρόπος, καθώς αυξάνει σημαντικά την πολυπλοκότητα ενός δέντρου απόφασης, το οποίο πλέον αποκτά πολλά επίπεδα και κόμβους, με αποτέλεσμα να γίνεται δυσανάγνωστο και δυσερμήνευτο.

Άσκηση 6

Να επαναληφθεί η άσκηση 1, επιλέγοντας τη δημιουργία ενός δέντρου απόφασης, το οποίο θα βασίζεται στο μέτρο της Εντροπίας και θα επιτρέπει πενήντα τουλάχιστον περιπτώσεις σε καθένα από τα φύλλα του.

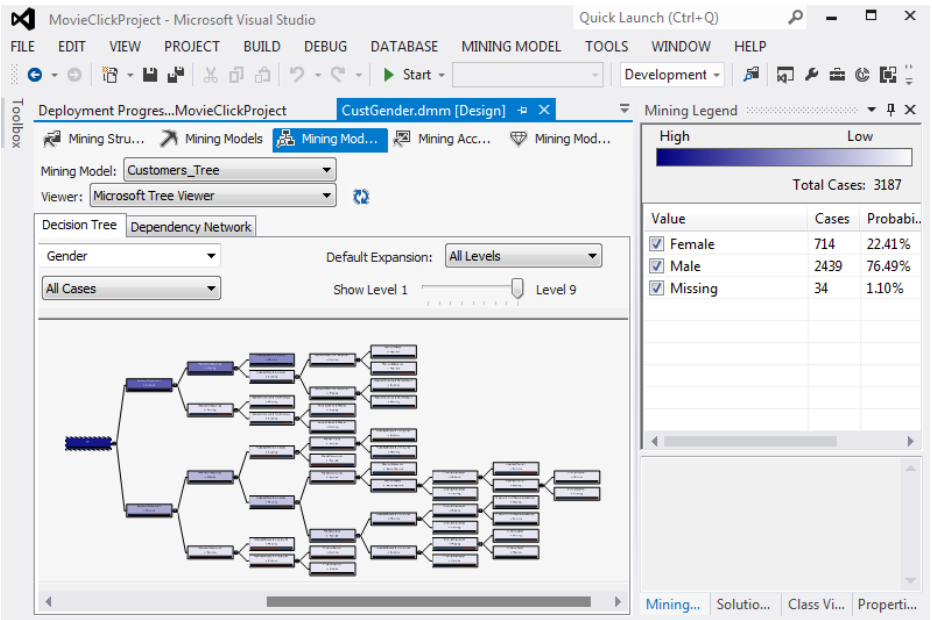
Λύση

1. Δίνουμε τις παρακάτω τιμές στις παραμέτρους:



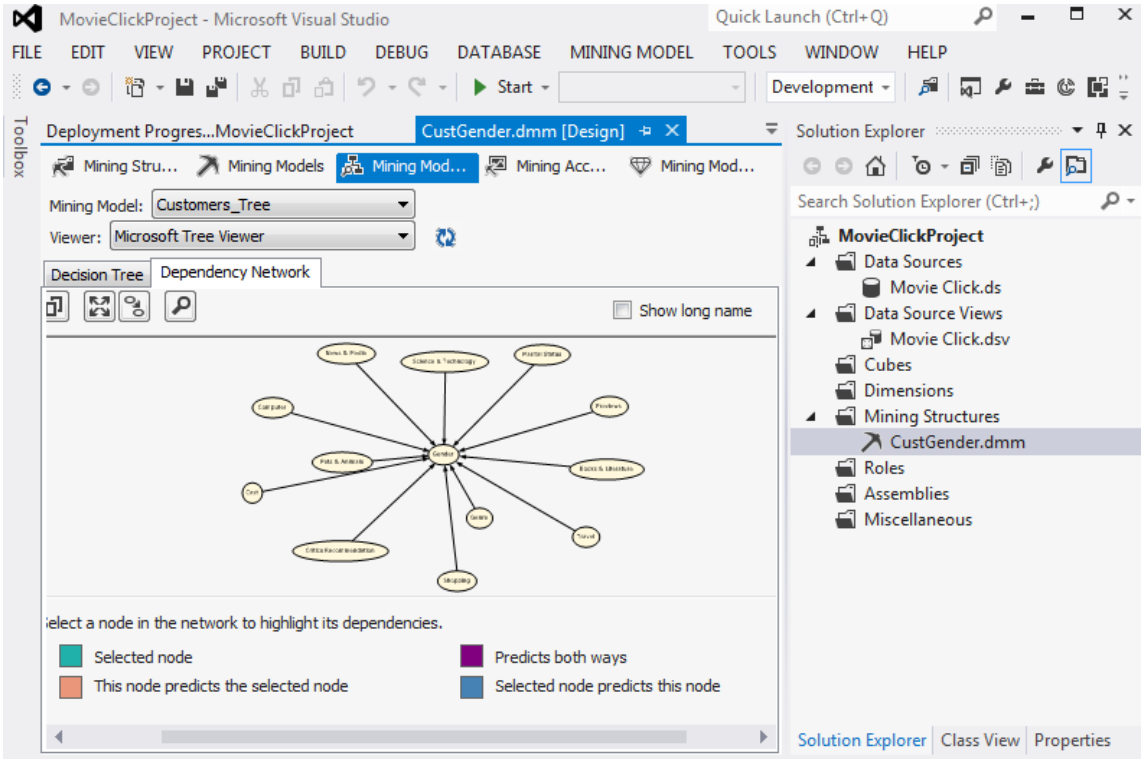
Εικόνα 7.60

2. Το δέντρο έχει την παρακάτω μορφή:



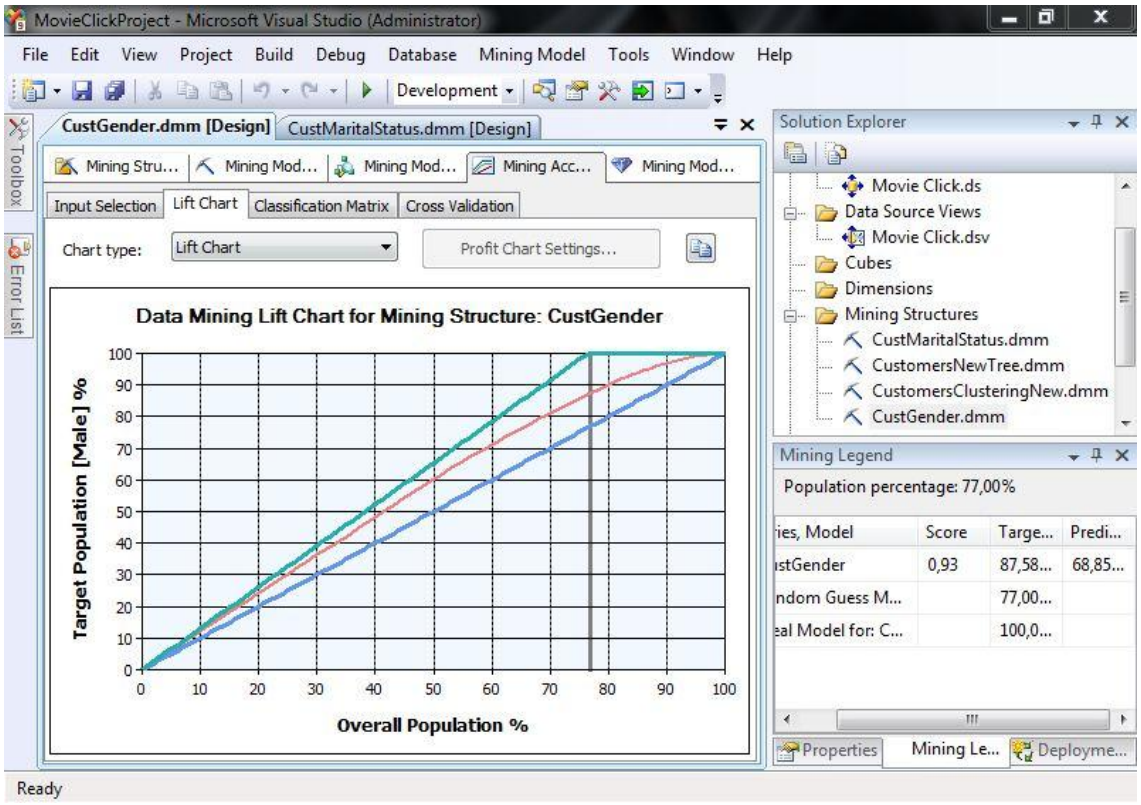
Εικόνα 7.61

3. Τα χαρακτηριστικά, από τα οποία εξαρτάται η προβλεπόμενη τιμή, είναι τα παρακάτω:



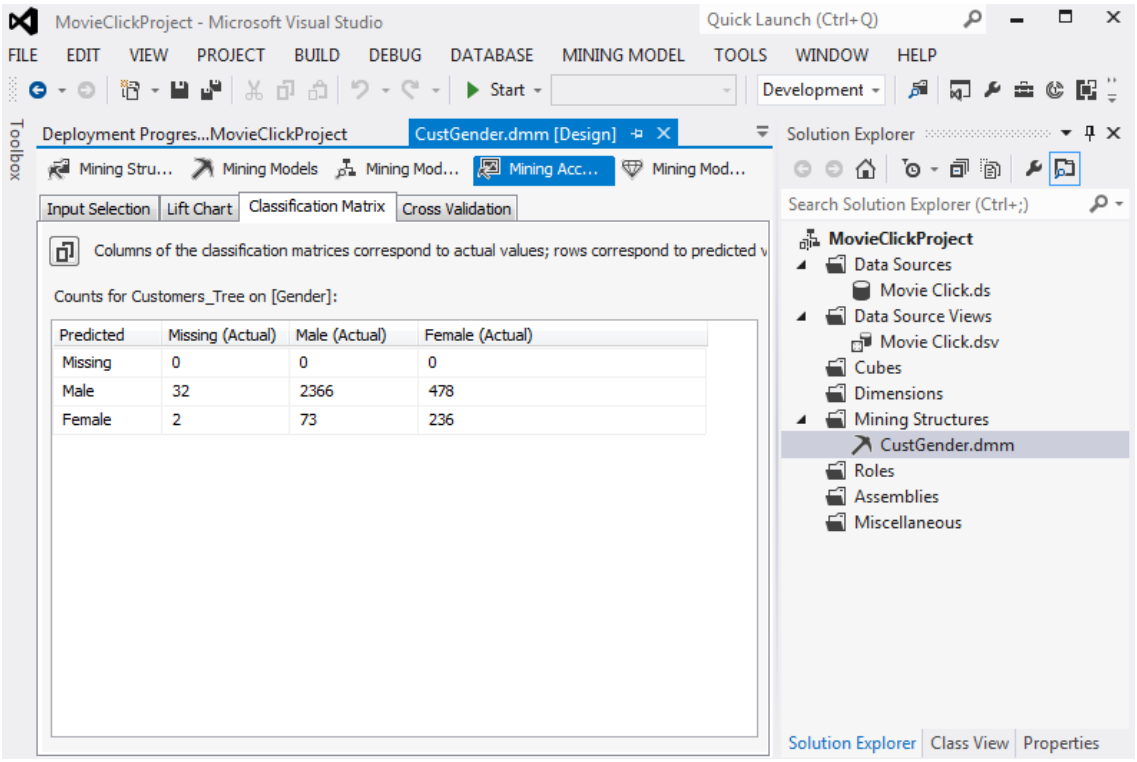
Εικόνα 7.62

4. Στο Lift Chart, όπως φαίνεται στην Εικόνα 7.63, με δείγμα 77% προβλέπουμε σωστά το φύλο σε ποστοστό 87,58% των ανδρών. Το ποσοστό είναι σχετικά καλό και το Score είναι στο 0,93. Οι τιμές είναι λίγο καλύτερες από το default δέντρο.



Εικόνα 7.63

5. Το ίδιο ισχύει και για τον Classification Matrix, όπως φαίνεται στην Εικόνα 7.64:



Εικόνα 7.64

6. Συμπερασματικά, αν χρησιμοποιήσουμε την εντροπία σε συνδυασμό με τις άλλες παραμέτρους (π.χ. minimum support), η ακρίβεια του δέντρου απόφασης μπορεί να μειώνεται αλλά, από την άλλη, το δέντρο που κατασκευάζεται είναι ευανάγνωστο, αφού η πολυπολοκότητά του έχει μειωθεί.

7.6. Βιβλιογραφία/Αναφορές

Νανόπουλος, Α., & Μανωλόπουλος, Ι. (2008). *Εισαγωγή στην Εξόρυξη και τις Αποθήκες Δεδομένων*, Αθήνα, Εκδόσεις Νέων Τεχνολογιών.

Χαλκίδη, Μ., & Βεζυργιάννης, Μ. (2005). *Εξόρυξη Γνώσης από Βάσεις Δεδομένων και τον Παγκόσμιο Ιστό*, Αθήνα, Τυπωθήτω.

Κεφάλαιο 8. Ομαδοποίηση δεδομένων

Σύνοψη
Σ' αυτό το κεφάλαιο θα μελετήσουμε την τεχνική της ομαδοποίησης (clustering). Το Clustering αποτελεί μια τεχνική ομαδοποίησης των δεδομένων μιας βάσης σε υποσύνολα (clusters), με τέτοιο τρόπο ώστε τα δεδομένα που βρίσκονται στο ίδιο cluster να έχουν όσο το δυνατό περισσότερα κοινά στοιχεία μεταξύ τους. Αυτή η ομαδοποίηση διευκολύνει την πρόβλεψη των χαρακτηριστικών που μας ενδιαφέρουν.

8.1. Θεωρητικό υπόβαθρο των αλγορίθμων ομαδοποίησης του SQL Server

Η **ομαδοποίηση (clustering)** ή, αλλιώς, συσταδοποίησης εντοπίζει ομάδες αντικειμένων στα δεδομένα βάσει των κοινών χαρακτηριστικών τους και της απόστασης μεταξύ τους. Οι ομάδες των αντικειμένων διαμορφώνονται με βάση ένα μέτρο απόστασης, έτσι ώστε τα αντικείμενα της ίδιας ομάδας να είναι όσο το δυνατόν πλησιέστερα μεταξύ τους, ενώ τα αντικείμενα διαφορετικών ομάδων να είναι όσο το δυνατόν πιο απομακρυσμένα. Καθώς δεν υπάρχει καθολικός ορισμός για την αποδεκτή ομάδα, η επιλογή εξαρτάται κάθε φορά από το συγκεκριμένο πρόβλημα και τα δεδομένα που καλούμαστε να εξορύξουμε. Το περιβάλλον του SQL Server διαθέτει δύο ευρέως γνωστούς αλγόριθμους ομαδοποίησης, τον *k-means* και τον Expectation-Maximization Clustering, τα βασικά χαρακτηριστικά των οποίων παρουσιάζονται παρακάτω.

Ο αλγόριθμος *k*-means

Ο *k-means* ανήκει στην κατηγορία των αλγορίθμων διαιρετικής (partition-based) ομαδοποίησης, οι οποίοι διαιρούν τον χώρο σε *k* περιοχές, με τα αντικείμενα καθεμιάς από τις περιοχές να αντιστοιχούν σε μία ομάδα (Chakrabarti, 2003· Liu, 2007). Ο *k-means* αναπαριστά κάθε ομάδα *C*, με το κέντρο της κάθε ομάδας να ορίζεται από την παρακάτω εξίσωση:

$$m_i = \frac{1}{m} \sum_{j=1}^{m} d_{ij}$$

όπου m_i είναι η μέση τιμή μιας ομάδας C_i, m είναι ο αριθμός των αντικειμένων που ανήκουν στην ομάδα και d_{ij} είναι η απόσταση ενός αντικειμένου j από το κέντρο της ομάδας C_i. Ο *k-means* είναι ένας επαναληπτικός αλγόριθμος, όπου σε κάθε επανάληψη τα αντικείμενα ενός συνόλου δεδομένων μετακινούνται μεταξύ των διαφόρων ομάδων, με στόχο να ελαχιστοποιηθεί η παρακάτω αντικειμενική συνάρτηση, που εκφράζει τη μέση τετραγωνική απόσταση (Sum of Squared Error ή, αλλιώς, μέσο τετραγωνικό σφάλμα) των αντικειμένων από τα πλησιέστερα κέντρα των ομάδων:

$$\text{SSE} = \sum_{i} \sum_{x \in C_i} (x - m_i)^2 ,$$

όπου x είναι ένα αντικείμενο, C_i είναι μια ομάδα i και m_i είναι η μέση τιμή του C_i. Η ελαχιστοποίηση της παραπάνω αντικειμενικής συνάρτησης πετυχαίνει τον εντοπισμό ομάδων που παρουσιάζουν μεγάλο βαθμό ομοιότητας μεταξύ των αντικειμένων της ίδιας ομάδας, ενώ ταυτόχρονα τα αντικείμενα μιας ομάδας διαφέρουν σημαντικά από τα αντικείμενα που ανήκουν σε διαφορετικές ομάδες. Τα βήματα του αλγορίθμου είναι, συνοπτικά, τα εξής:

1. Διάλεξε *k* τυχαία αντικείμενα ως κέντρα των ομάδων.
2. Ανάθεσε κάθε αντικείμενο στην ομάδα με το πλησιέστερο κέντρο.
3. Υπολόγισε το νέο κέντρο για καθεμία από τις *k* ομάδες.

4. Αν όλα τα κέντρα συμπίπτουν με τα προηγούμενα κέντρα των ομάδων (δηλαδή, αν δεν υπήρξε μεταβολή), τότε τερμάτισε, διότι ο αλγόριθμος έχει συγκλίνει. Διαφορετικά, επανάλαβε το βήμα 2.

Όπως είπαμε και προηγουμένως, το κριτήριο σύγκλισης μπορεί να βασιστεί στην ελαχιστοποίηση του μέσου τετραγωνικού σφάλματος SSE. Εναλλακτικά, ο αλγόριθμος μπορεί να τερματίζει, είτε όταν δεν υπάρχει κανένα αντικείμενο που να ανατίθεται σε διαφορετική ομάδα ή μετά από ένα συγκεκριμένο πλήθος επαναλήψεων που εγγυάται τον τερματισμό του αλγορίθμου, ακόμη και όταν αυτός δεν συγκλίνει. Τονίζεται ότι στο περιβάλλον του SQL Server υπάρχει σχετική παράμετρος (**stopping_tolerance**), η οποία καθορίζει τον ελάχιστο αριθμό των αντικειμένων που θα πρέπει να μετακινούνται μεταξύ των ομάδων σε κάθε επανάληψη, για να μην τερματίσει ο αλγόριθμος.

Ο αλγόριθμος Expectation-Maximization Clustering

Ο αλγόριθμος Expectation-Maximization (EM) Clustering βασίζεται σε ένα μοντέλο πιθανοτήτων που συνδυάζει διαφορετικές Gaussian κατανομές (Gaussian Mixture Model), όπου κάθε κατανομή αντιπροσωπεύει μία ομάδα (Rajaraman, Leskovec, & Ullman, 2015· Roiger, & Geatz, 2003· Tan, Steinback, & Kumar, 2006). Ο αλγόριθμος EM υπολογίζει επαναληπτικά την πιθανότητα που έχει ένα αντικείμενο του συνόλου δεδομένων να παράγεται από την i-οστή Gaussian κατανομή. Δηλαδή, υπολογίζει την πιθανότητα που έχει ένα αντικείμενο να ανήκει στη ομάδα i. Αυτό σημαίνει ότι ένα αντικείμενο μπορεί να ανατίθεται σε περισσότερες ομάδες με διαφορετική πιθανότητα κάθε φορά. Συγκεκριμένα, για κάθε αντικείμενο x προσδιορίζεται η δεσμευμένη πιθανότητα prob($x \mid \theta_i$) να έχει συγκεκριμένες τιμές στα χαρακτηριστικά του, με την προϋπόθεση ότι ανήκει σε μια ομάδα i, όπου θ_i είναι το σύνολο το παραμέτρων της υπό εξέταση ομάδας. Μια γενική περιγραφή των βημάτων του αλγορίθμου EM φαίνεται παρακάτω:

1. Επίλεξε τυχαίες αρχικές τιμές για το σύνολο Θ των παραμέτρων που προσδιορίζουν την κάθε κατανομή/ ομάδα (π.χ. μέση τιμή, τυπική απόκλιση κτλ.)

2. Όσο αλλάζουν οι τιμές των παραμέτρων του συνόλου Θ ή όσο δεν έχουμε φτάσει σε έναν ανώτατο αριθμό επαναλήψεων, επανάλαβε:

α. (Expectation-Step) Για κάθε αντικείμενο υπολόγισε την πιθανότητα να ανήκει σε μια κατανομή/ ομάδα.

β. (Maximization-Step) Βάσει των τιμών των πιθανοτήτων του βήματος 2α, υπολόγισε τις νέες τιμές των παραμέτρων Θ.

Σ' αυτό το σημείο τονίζεται ότι ο αλγόριθμος k-means είναι μια ειδική περίπτωση του αλγορίθμου EM. Συγκεκριμένα, το expectation step του αλγορίθμου EM αντιστοιχεί στο βήμα της ανάθεσης κάθε αντικειμένου σε μια ομάδα (βήμα 2 του k-means). Επιπροσθέτως, το maximization step του αλγορίθμου EM αντιστοιχεί στον υπολογισμό των νέων κέντρων των ομάδων (βήμα 3 του k-means). Στο περιβάλλον του SQL Server, ο αλγόριθμος EM είναι ο προεπιλεγμένος αλγόριθμος, επειδή, σε σύγκριση με το k-means, προσφέρει πολλαπλά πλεονεκτήματα, τα οποία περιγράφονται συνοπτικά παρακάτω:

1. Απαιτεί το πολύ μία σάρωση της βάσης δεδομένων.

2. Τρέχει ακόμη και σε περιπτώσεις πολύ περιορισμένης μνήμης RAM.

3. Αποδίδει ταχύτερα έναντι άλλων μεθόδων δειγματοληψίας.

Τέλος, στο περιβάλλον του SQL Server οι αλγόριθμοι k-means και EM διατίθενται σε δύο εναλλακτικές μορφές: Scalable και non-scalable. Στην πρώτη μορφή, σαρώνονται οι πρώτες 50.000 εγγραφές της βάσης δεδομένων. Αν η δειγματοληψία των δεδομένων εκπαίδευσης είναι επαρκής και το μοντέλο προσαρμοστεί επιτυχώς στα δεδομένα, τότε δεν χρειάζεται να γίνει προσπέλαση άλλων δεδομένων. Διαφορετικά, σαρώνονται ακόμη 50.000 εγγραφές από την βάση δεδομένων. Στη δεύτερη μορφή, σαρώνονται όλες οι

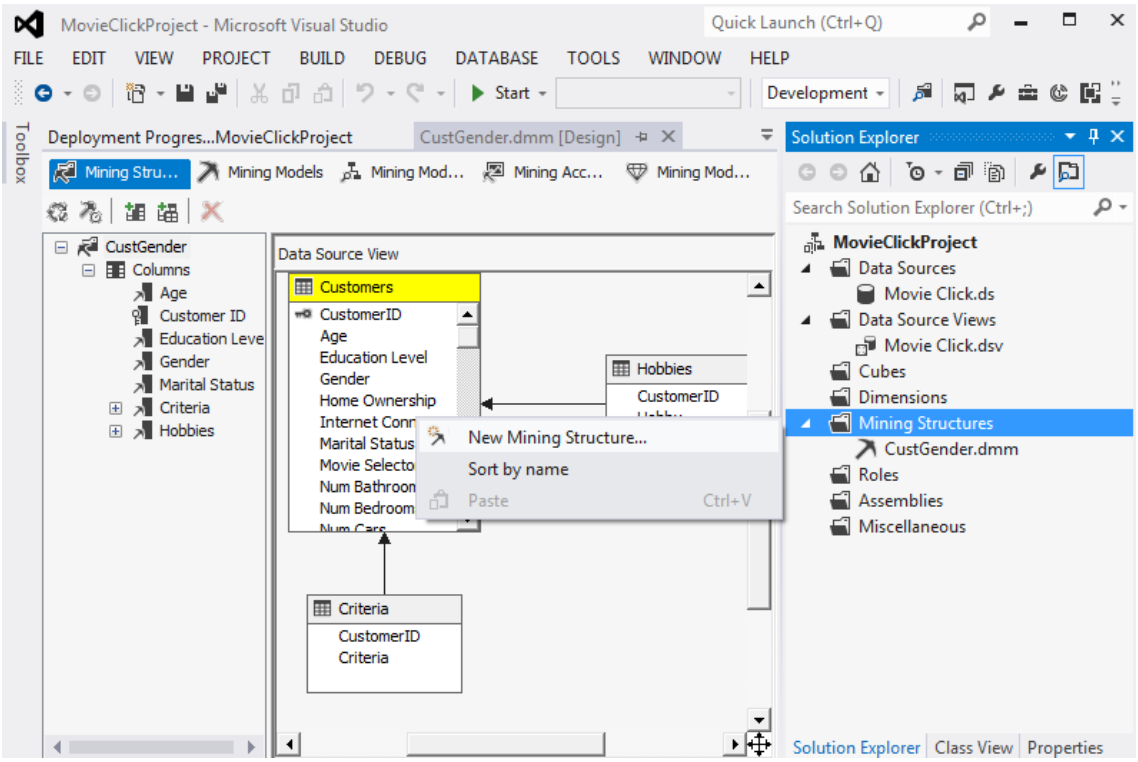
εγγραφές της βάσης δεδομένων, κάτι που, βέβαια, έχει αυξημένες απαιτήσεις κάθε φορά σε χρόνο, μνήμη RAM και υπολογιστική ισχύ.

8.2. Δημιουργία ενός μοντέλου ομαδοποίησης δεδομένων

Ας υποθέσουμε ότι ένα Video Club θέλει να προβλέψει την οικογενειακή κατάσταση των πελατών του, χρησιμοποιώντας την τεχνική της ομαδοποίησης αυτών και έχοντας ως βάση τα ενδιαφέροντά τους (Hobbies), την τεχνολογία (Technology) και κάποια άλλα στοιχεία των πελατών. Παρακάτω παρουσιάζονται τα αναλυτικά βήματα για την δημιουργία ενός μοντέλου βάσει της ομαδοποίησης δεδομένων.

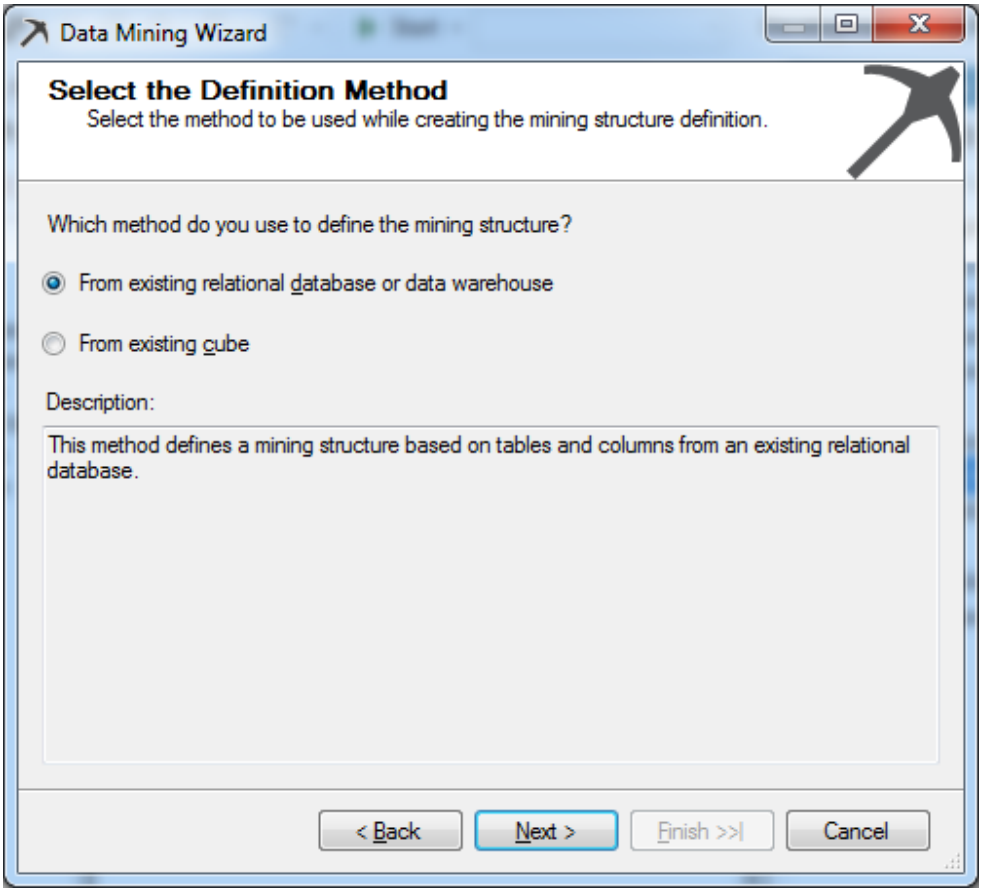
Αναλυτικά βήματα

1. Βρισκόμαστε στο MovieClick Project της καρτέλας Solution Explorer, όπως φαίνεται στην Εικόνα 8.1. Κάνουμε δεξί κλικ στο Mining Structure και επιλέγουμε New Mining Structure.



Εικόνα 8.1

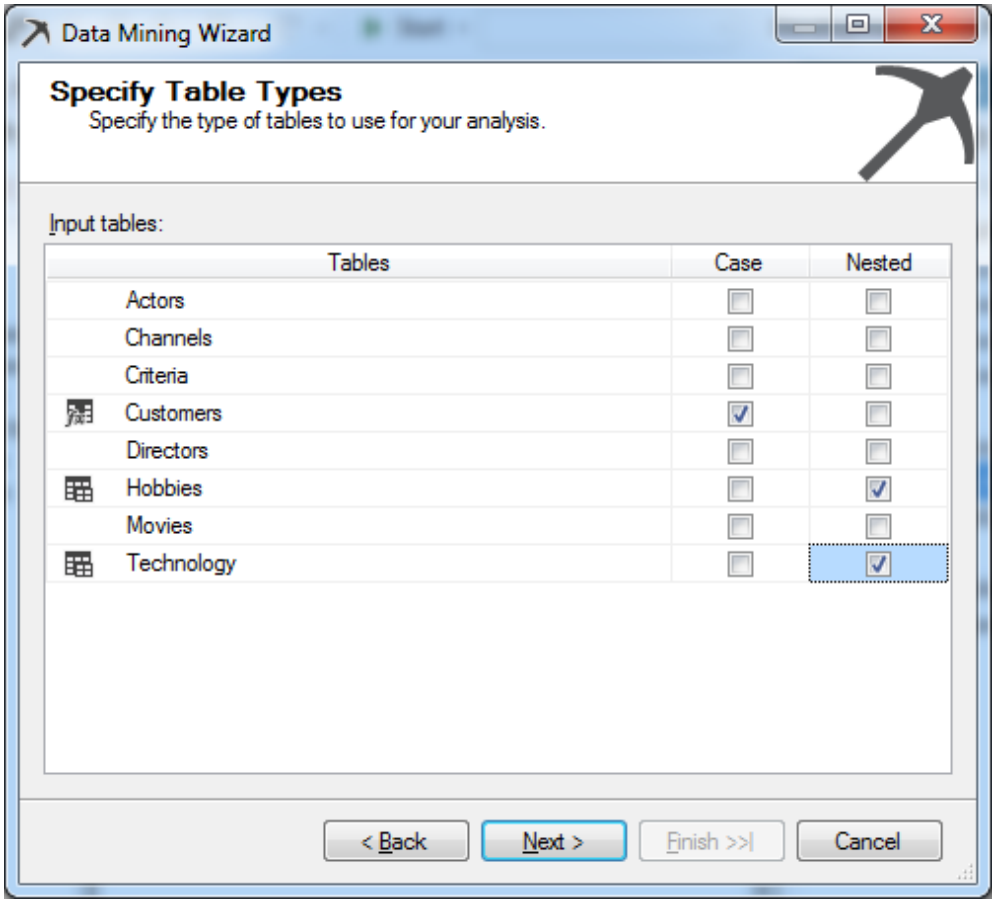
2. Επιλέγουμε From existing relational data or data warehouse, όπως φαίνεται στην Εικόνα 8, καθώς θα χρησιμοποιήσουμε τη βάση που έχουμε εισάγει στον SQL Server. Επιλέγουμε Next>, ώστε να προχωρήσουμε στο επόμενο βήμα.



Εικόνα 8.2

3. Στο παράθυρο που εμφανίζεται επιλέγουμε τον αλγόριθμο Microsoft Clustering και, στη συνέχεια, Next>, ώστε να προχωρήσουμε στο επόμενο βήμα. Στη συνέχεια, επιλέγουμε το MovieClick στο πεδίο με τα διαθέσιμα Data Source Views. Τέλος, επιλέγουμε Next>, ώστε να προχωρήσουμε στο επόμενο βήμα.

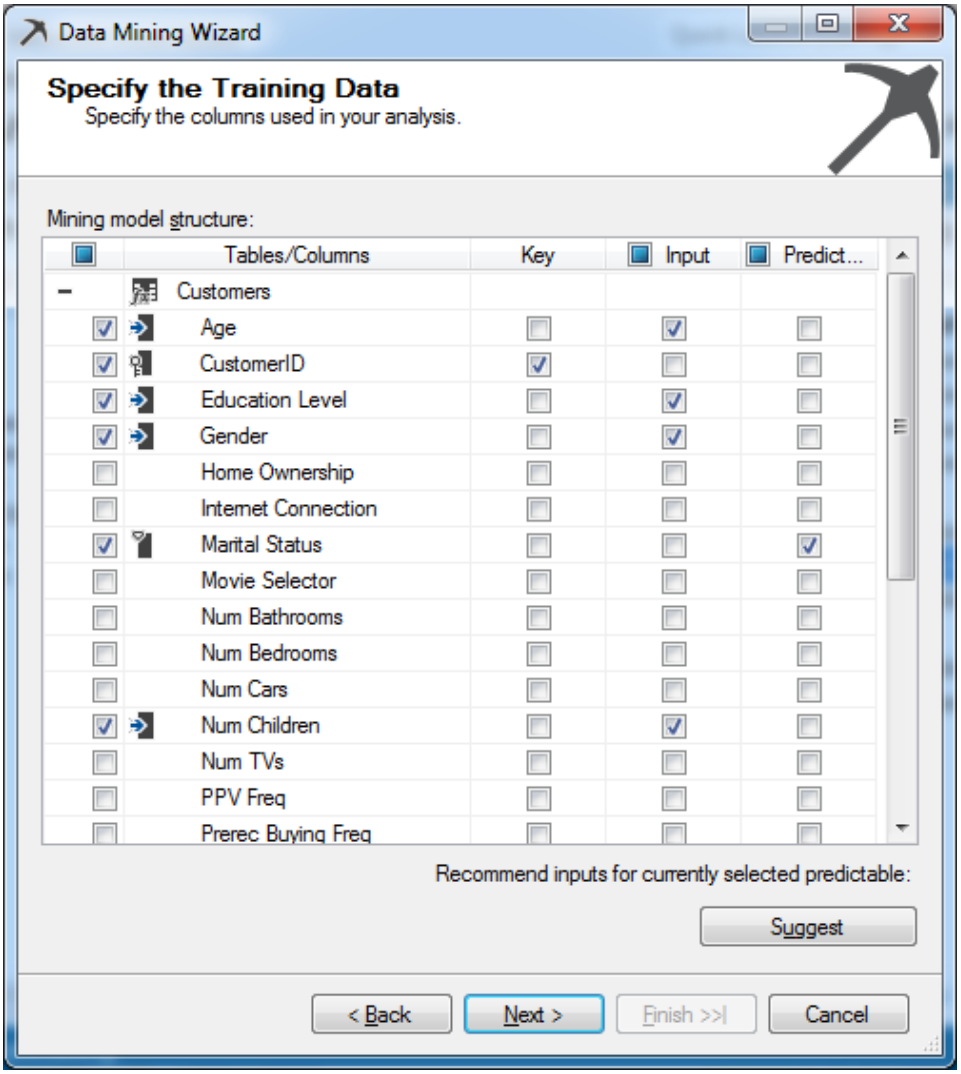
4. Στο παράδειγμα μας, όπως προαναφέραμε, θέλουμε να προβλέψουμε την οικογενειακή κατάσταση των πελατών μας σε σχέση με τα ενδιαφέροντά τους και την τεχνολογία. Σ' αυτό το στάδιο επιλέγουμε ποιος πίνακας θα οριστεί ως Case και ποιοι πίνακες θα είναι οι Nested. Case είναι ο πίνακας που περιέχει τα δεδομένα που θέλουμε να προβλέψουμε, ενώ Nested είναι οι πίνακες τα δεδομένα των οποίων είναι παράμετροι στον Case. Στη συγκεκριμένη περίπτωση, όπως φαίνεται στην Εικόνα 8.3, επιλέγουμε τον πίνακα Customers ως Case και τους πίνακες Hobbies και Technology ως Nested, καθώς, όπως αναφέραμε, θέλουμε να προβλέψουμε την οικογενειακή κατάσταση των πελατών σε σχέση με τα ενδιαφέροντά τους και την τεχνολογία. Κατόπιν, επιλέγουμε Next>, ώστε να προχωρήσουμε στο επόμενο βήμα.



Εικόνα 8.3

5. Σ’ αυτό το στάδιο επιλέγουμε ποια δεδομένα των πινάκων που επιλέξαμε στο προηγούμενο βήμα θα είναι είσοδος στο μοντέλο και ποια δεδομένα θέλουμε να προβλέψουμε. Συγκεκριμένα, όπως φαίνεται στην Εικόνα 8.4, κάνουμε τις εξής επιλογές:

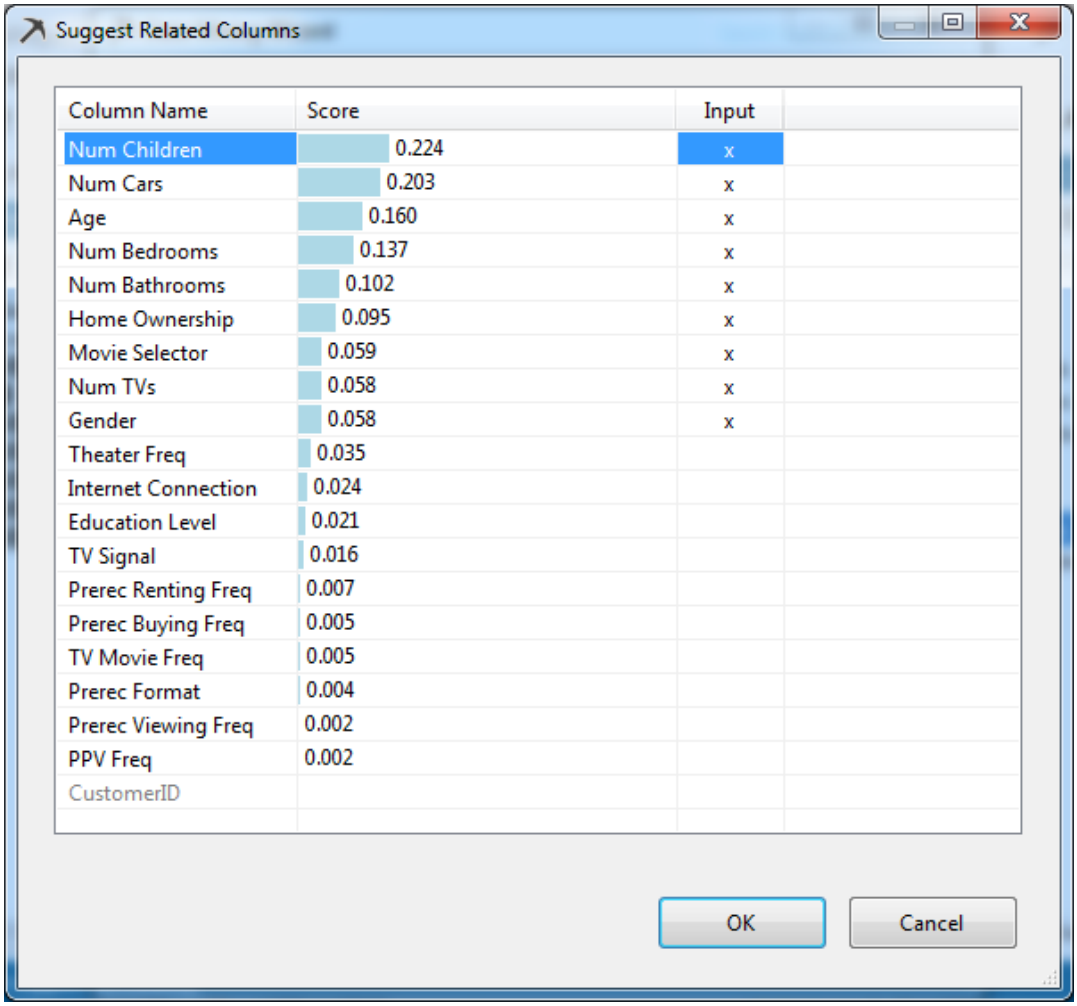
   - Για κάθε πίνακα επιλέγουμε ένα κλειδί Key. Στη συγκεκριμένη περίπτωση, επιλέγουμε τα CustomerID, Hobby και Technology.
   - Ορίζουμε ως Input τις στήλες των πινάκων που μας ενδιαφέρουν. Στη συγκεκριμένη περίπτωση, επιλέγουμε τα Age, Education Level, Gender, Num Children, Technology και Hobby.
   - Ορίζουμε ως Predictable τη στήλη που μας ενδιαφέρει να προβλέψουμε (αυτή, εξάλλου, θα είναι η έξοδος του μοντέλου). Στη συγκεκριμένη περίπτωση, επιλέγουμε το Marital Status.



Εικόνα 8.4

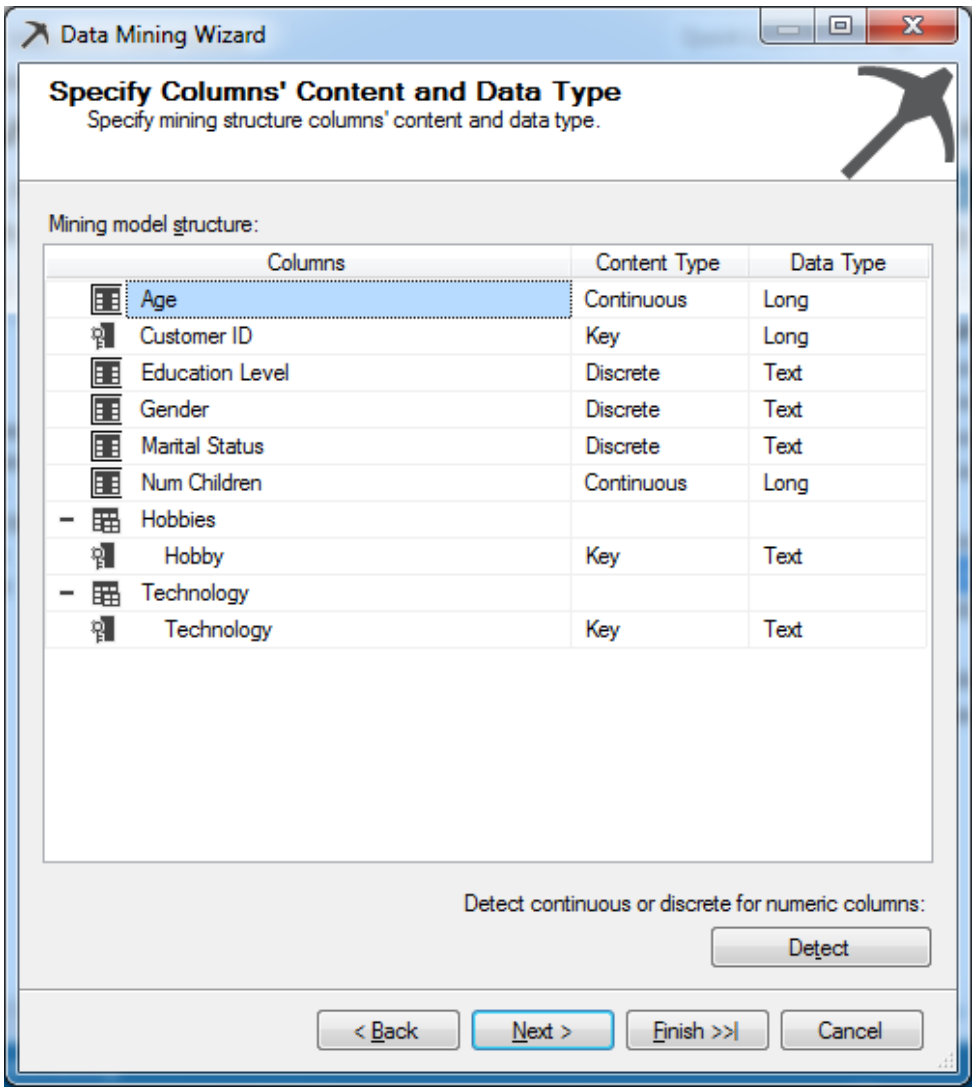
Δεν θα πατήσουμε τώρα Next, καθώς θέλουμε απλώς να λάβουμε υπόψη μας το αποτέλεσμα που δίνει η επιλογή Suggest.

Πράγματι, στο παραπάνω παράθυρο κάνουμε κλικ στο Suggest. Τότε, όπως φαίνεται στην Εικόνα 8.5, εμφανίζεται μια σχέση της predictable τιμής με τα άλλα στοιχεία των πινάκων. Επιλέγουμε Cancel, καθώς, αν επιλέξουμε OK, τότε όλες οι στήλες που φαίνεται να συσχετίζονται θα συμπεριληφθούν στο Mining Structure. Τώρα, καθώς επανήλθαμε στο παράθυρο της Εικόνας 8.4 επιλέγουμε Next>, ώστε να προχωρήσουμε στο επόμενο βήμα.



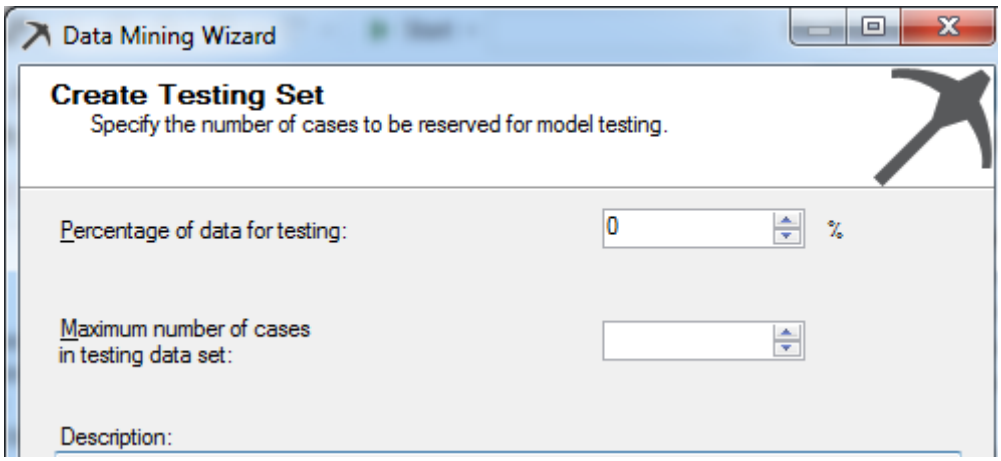
Εικόνα 8.5

6. Εμφανίζεται μια σύνοψη-επιβεβαίωση του περιεχομένου του Mining Structure, όπως φαίνεται στην Εικόνα 8.6. Επιλέγουμε Detect, για να επιλεχθεί ο κατάλληλος τύπος δεδομένων από το σύστημα και να γίνει ο σαφής διαχωρισμός διακριτών και συνεχών τιμών, που γίνεται ύστερα από δειγματοληψία και ανάλυση δεδομένων. Κατόπιν, επιλέγουμε Next>, ώστε να προχωρήσουμε στο επόμενο βήμα.



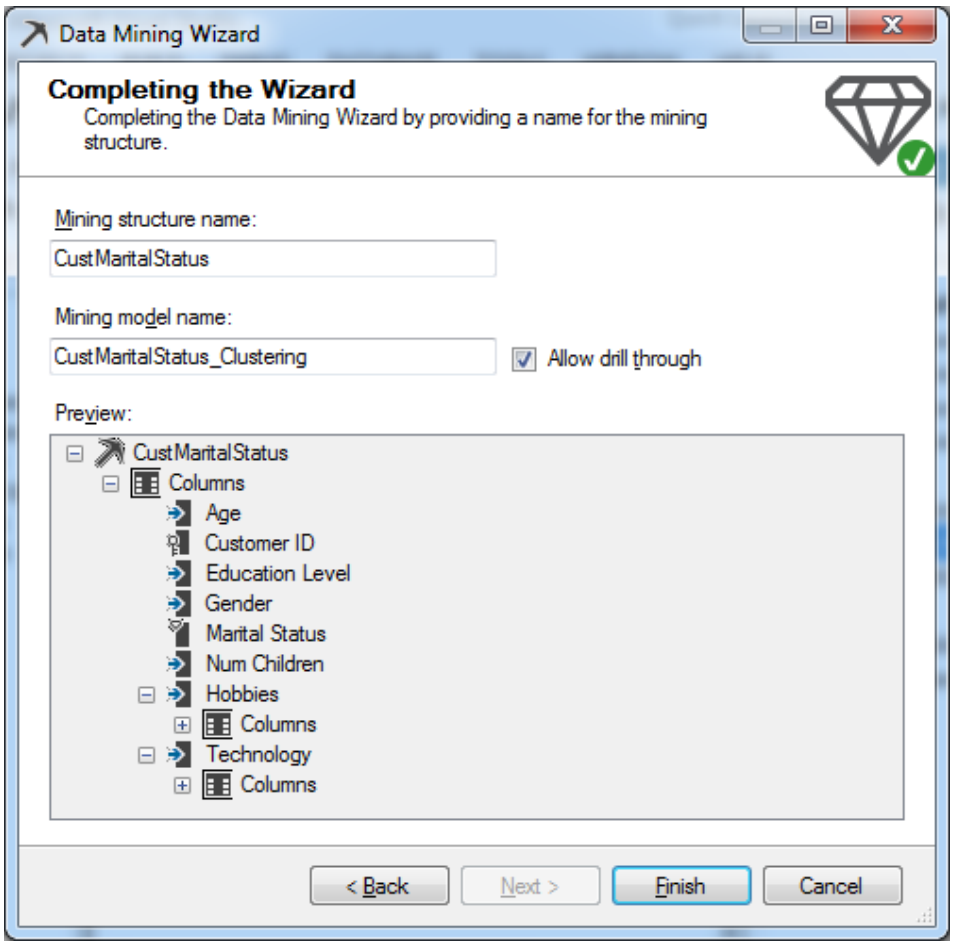
Εικόνα 8.6

7. Στον νέο πίνακα, όπως φαίνεται στην Εικόνα 8.7, ορίζουμε το ποσοστό των δεδομένων που το μοντέλο θα διατηρήσει για την επαλήθευσή του. Στη συγκεκριμένη περίπτωση, προσδιορίζουμε το test set σε 0%, διότι θέλουμε να μετρήσουμε με ακρίβεια σε ολόκληρο το train set (αισιόδοξη πρόβλεψη).



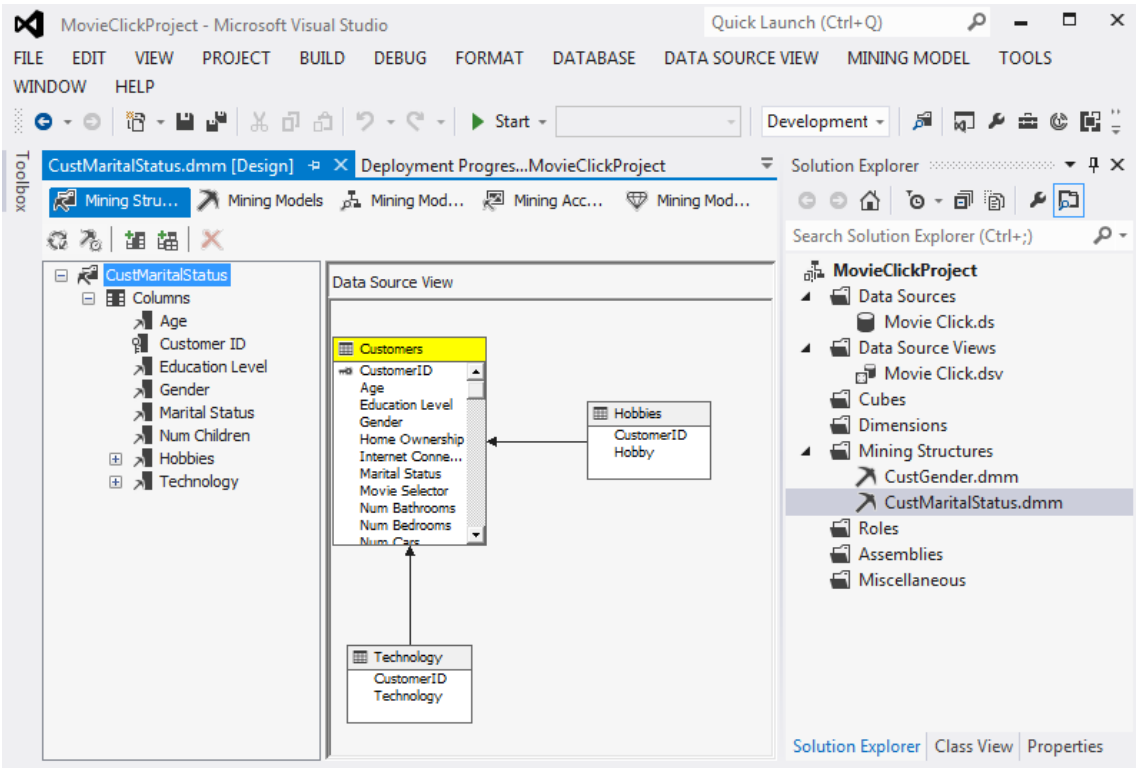
Εικόνα 8.7

8. Στη συνέχεια, όπως φαίνεται στην Εικόνα 8.8, ορίζουμε όνομα για το Mining structure name και το Mining model. Στη συγκεκριμένη περίπτωση, συμπληρώνουμε CustMaritalStatus στο πεδίο Mining structure name και CustMaritalStatus_Clustering στο πεδίο Mining model name. Κατόπιν, επιλέγουμε Allow drill through, προκειμένου να μπορούμε να δούμε τα δεδομένα των πινάκων μας. Τέλος, πατάμε Finish, ώστε να ολοκληρωθεί η διαδικασία.



Εικόνα 8.8

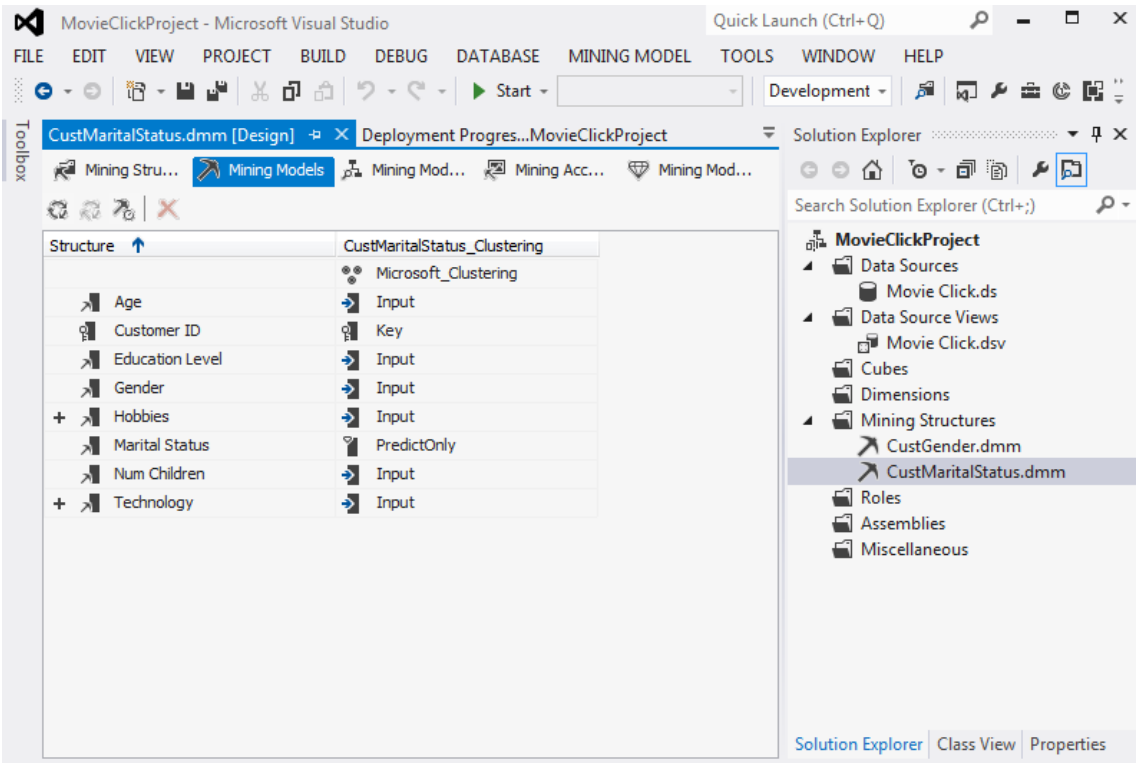
9. Εμφανίζεται το παράθυρο του Data Mining Designer, όπως φαίνεται στην Εικόνα 8.9. Επιλέγοντας την καρτέλα Mining Structure, βλέπουμε το Mining Structure που δημιουργήσαμε.



Εικόνα 8.9

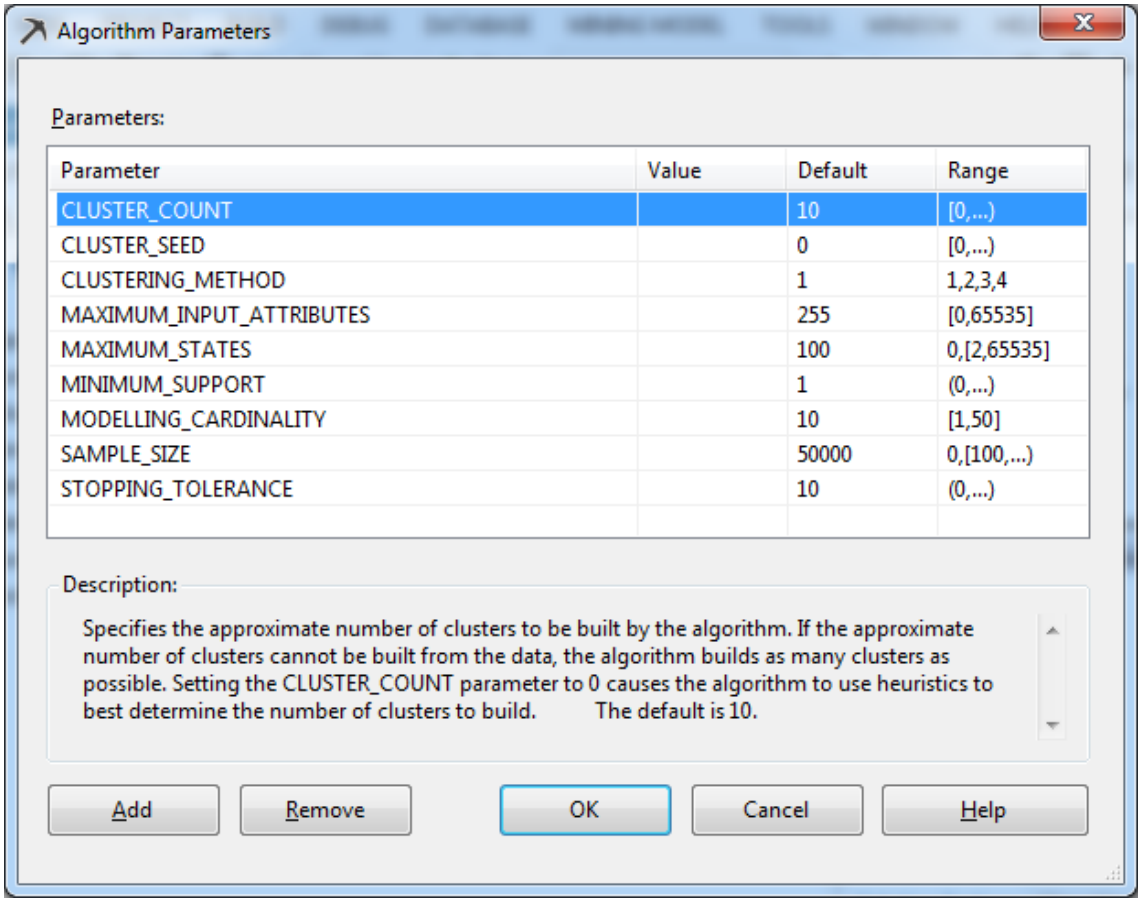
10. Στη συνέχεια, όπως φαίνεται στην Εικόνα 8.10, επιλέγουμε την καρτέλα Mining Models, ώστε να καθορίσουμε τις παραμέτρους για το μοντέλο που θα μελετήσουμε. Βλέπουμε ότι κάθε δεδομένο έχει οριστεί ως Input, Key, Predict ή PredictOnly. Η διαφορά ανάμεσα σε Predict και PredictOnly είναι ότι τα πρώτα δεδομένα μπορούμε να τα χρησιμοποιήσουμε και ως είσοδο αλλά και ως έξοδο του αλγορίθμου. Αντίθετα, τα PredictOnly μπορούμε να τα χρησιμοποιήσουμε μόνο ως έξοδο. Στη συγκεκριμένη περίπτωση θέλουμε να προβλέψουμε το Marital Status των πελατών ανάλογα με την ηλικία, τη μόρφωση, το φύλο, τον αριθμό των παιδιών, τα hobbies και την τεχνολογία. Επομένως, ορίζουμε τα χαρακτηριστικά ως εξής:

    - Age: Input
    - CustomerID: Key
    - Education Level: Input
    - Gender: Input
    - Hobbies: Input
    - Marital Status: PredictOnly
    - Num Children: Input
    - Technology: Input



Εικόνα 8.10

11. Στη συνέχεια, θα μελετήσουμε τις παραμέτρους με τις οποίες κατασκευάζεται το μοντέλο και τις προεπιλεγμένες τιμές που παίρνουν. Στον πίνακα της Εικόνας 8.10 κάνουμε δεξί κλικ στον αλγόριθμο Microsoft_Clustering και επιλέγουμε Set Algorithm Parameters. Εμφανίζεται ένα νέο παράθυρο με 9 παραμέτρους, όπως βλέπουμε πλέον στην Εικόνα 8.11.



Εικόνα 8.11

Ακολουθεί η αναλυτική περιγραφή της κάθε παραμέτρου του αλγορίθμου Clustering:

- **CLUSTER_COUNT:** Αυτή η παράμετρος καθορίζει κατά προσέγγιση τον αριθμό των clusters που θέλουμε να κατασκευαστούν από τον αλγόριθμο. Αν ο αριθμός των clusters δεν έχει προσδιοριστεί από τον χρήστη, τότε ο αλγόριθμος επιλέγει τον αριθμό των clusters ευρηστικά, χωρίς να υπάρχει κάποια εγγύηση ότι η ομαδοποίηση είναι βέλτιστη. Ο εξ ορισμού αριθμός των cluster που δημιουργούνται αυτόματα από τον αλγόριθμο είναι 10.

- **CLUSTER_SEED**: Αυτή η παράμετρος καθορίζει τον αριθμό των αρχικών σπόρων για την τυχαία δημιουργία των clusters που θα δημιουργηθούν στο πρώτο στάδιο κατασκευής του μοντέλου. Στη συγκεκριμένη περίπτωση διατηρείται η προεπιλεγμένη τιμή.

- **CLUSTERING_METHOD**: Αυτή η παράμετρος καθορίζει τον αλγόριθμο που θα χρησιμοποιηθεί για clustering. Υπάρχουν 2 διαθέσιμοι αλγόριθμοι:
  - Ο αλγόριθμος Expectation Maximization (EM) είναι πιθανοκρατικός και προσδιορίζει την πιθανότητα ενός στιγμιοτύπου/εγγραφής να ανήκει σε ένα cluster.
  - Ο αλγόριθμος k-means θεωρεί ότι υπάρχουν K αρχικοί μέσοι στους οποίους ανήκουν τα στιγμιότυπα/εγγραφές του πίνακα case. Κάθε στιγμιότυπο ανήκει στον μέσο εκείνο από τον οποίο απέχει την μικρότερη απόσταση. Κατά την εκτέλεση του αλγορίθμου, μεταβάλλονται τα κέντρα αλλά και τα clusters στα οποία ανήκει κάθε εγγραφή του πίνακα case. Στη συγκεκριμένη περίπτωση διατηρείται η προεπιλεγμένη τιμή 1.

- **MAXIMUM_INPUT_ATTRIBUTES**: Αυτή η παράμετρος καθορίζει τον μέγιστο αριθμό των χαρακτηριστικών εισόδου πριν ο αλγόριθμος αρχίσει να επιλέγει χαρακτηριστικά. Η τιμή 0 δηλώνει ότι δεν υπάρχει μέγιστος αριθμός χαρακτηριστικών. Στη συγκεκριμένη περίπτωση αφήνουμε την προεπιλεγμένη τιμή.

- **MAXIMUM_STATES:** Αυτή η παράμετρος καθορίζει τον μέγιστο αριθμό των καταστάσεων ενός χαρακτηριστικού. Αν ο αριθμός των καταστάσεων ενός χαρακτηριστικού είναι μεγαλύτερος από τον μέγιστο αριθμό των καταστάσεων που έχει οριστεί, ο αλγόριθμος χρησιμοποιεί εκείνες τις καταστάσεις των χαρακτηριστικών που είναι πιο δημοφιλείς και θεωρεί τις υπόλοιπες ως missing. Στη συγκεκριμένη περίπτωση διατηρείται η προεπιλεγμένη τιμή.

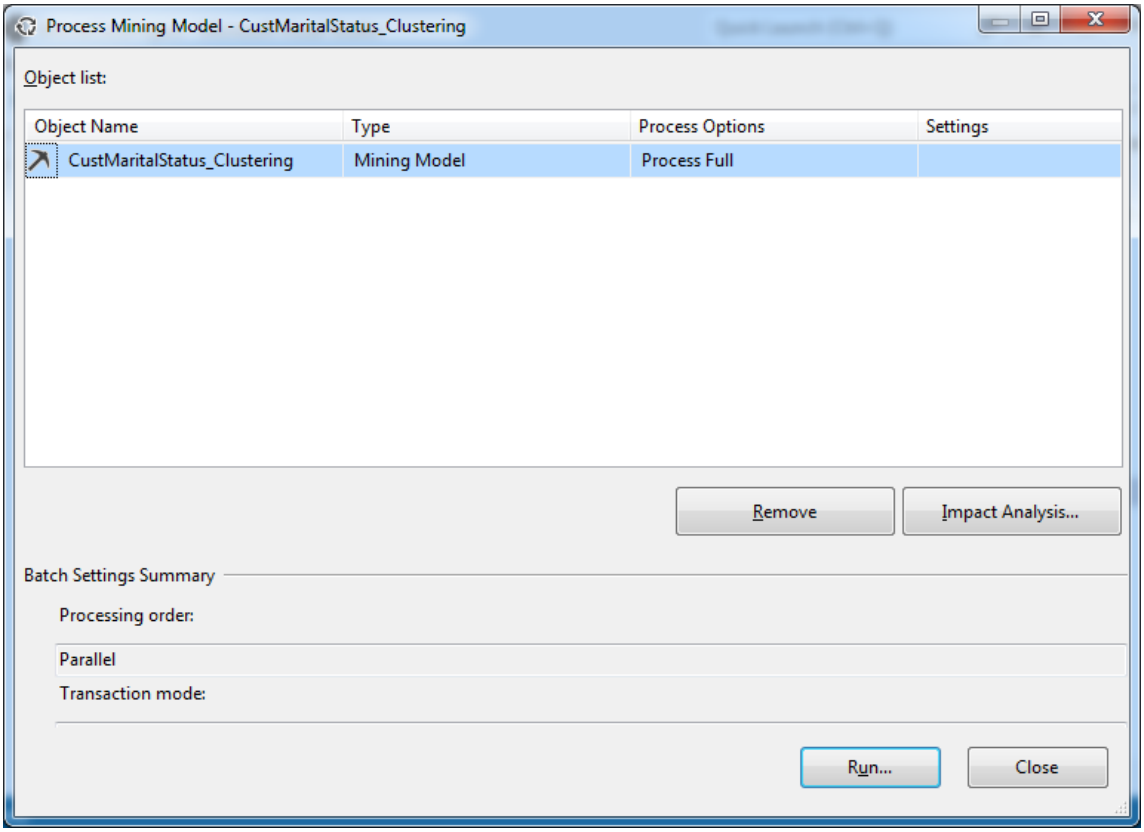
- **MINIMUM_SUPPORT:** Αυτή η παράμετρος προσδιορίζει το ελάχιστο πλήθος των περιπτώσεων που θα περιέχει κάθε cluster. Η προεπιλεγμένη τιμή γι' αυτήν την παράμετρο είναι 1.

- **MODELLING_CARDINALITY:** Αυτή η παράμετρος καθορίζει τα υποψήφια μοντέλα που θα δημιουργηθούν από τον αλγόριθμο που θα κάνει το clustering. Ο αλγόριθμος δημιουργεί ένα σύνολο υποψήφιων μοντέλων με τυχαίες αρχικοποιήσεις και, στη συνέχεια, επιλέγει το καλύτερο εξ αυτών μοντέλο. Με άλλα λόγια, αυτή η παράμετρος καθορίζει το σύνολο των υποψήφιων μοντέλων που θα δημιουργηθούν. Η προεπιλεγμένη τιμή είναι 10.

- **SAMPLE_SIZE:** Αυτή η παράμετρος καθορίζει τον αριθμό των στιγμιότυπων που χρησιμοποιεί ο αλγόριθμος σε κάθε πέρασμα, αν η παράμετρος CLUSTERING_METHOD έχει οριστεί σε μία από τις μεθόδους Scalable. Δίνοντας την τιμή 0, ο αλγόριθμος θα ομαδοποιήσει το σύνολο των δεδομένων σε ένα μόνο πέρασμα, γεγονός που μπορεί να λύσει προβλήματα περιορισμένης, ενδεχομένως, μνήμης και επίδοσης του υπολογιστή. Στη συγκεκριμένη περίπτωση διατηρείται η προεπιλεγμένη τιμή.

- **STOPPING_TOLERANCE:** Αυτή η παράμετρος καθορίζει τον αριθμό των περιπτώσεων που μετακινούνται μεταξύ των clusters σε κάθε πέρασμα του αλγορίθμου. Ο αλγόριθμος εφαρμόζεται επαναληπτικά στα δεδομένα και σχηματίζει τα cluster με την μορφή που εμείς τα βλέπουμε, ύστερα από ένα σύνολο επαναλήψεων. Επειδή σε κάθε επανάληψη προστίθενται διαρκώς και νέες περιπτώσεις, η τιμή της παραμέτρου μπορεί να θεωρηθεί ως ποσοστό και όχι ως ένας συγκεκριμένος αριθμός. Η προεπιλεγμένη τιμή της παραμέτρου είναι 10.

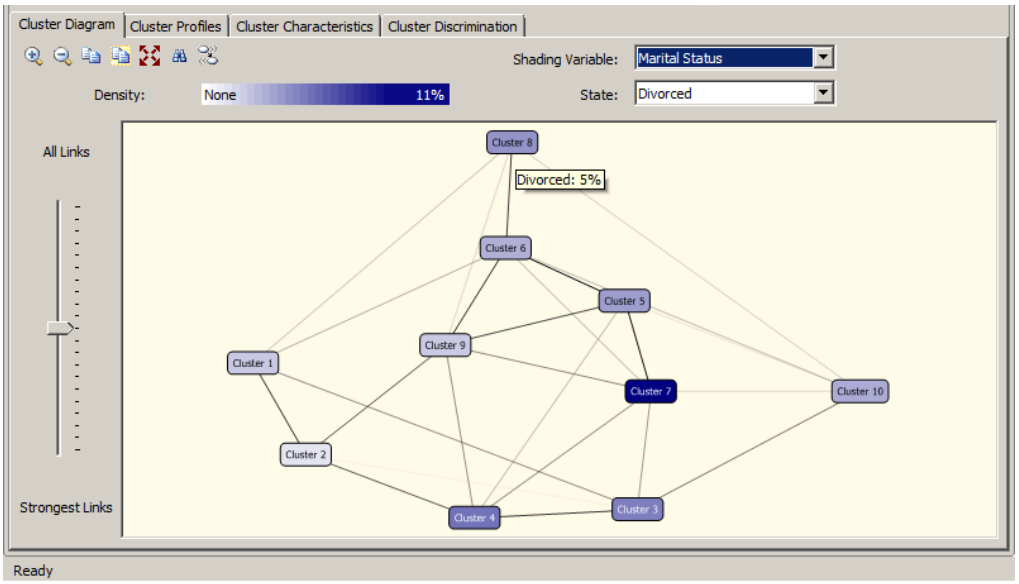
12. Επιλέγουμε την καρτέλα Mining Model Viewer, για να προβάλουμε το μοντέλο. Επιλέγουμε Run, ώστε να αποθηκευτεί και να «τρέξει» το μοντέλο μας.



Εικόνα 8.12

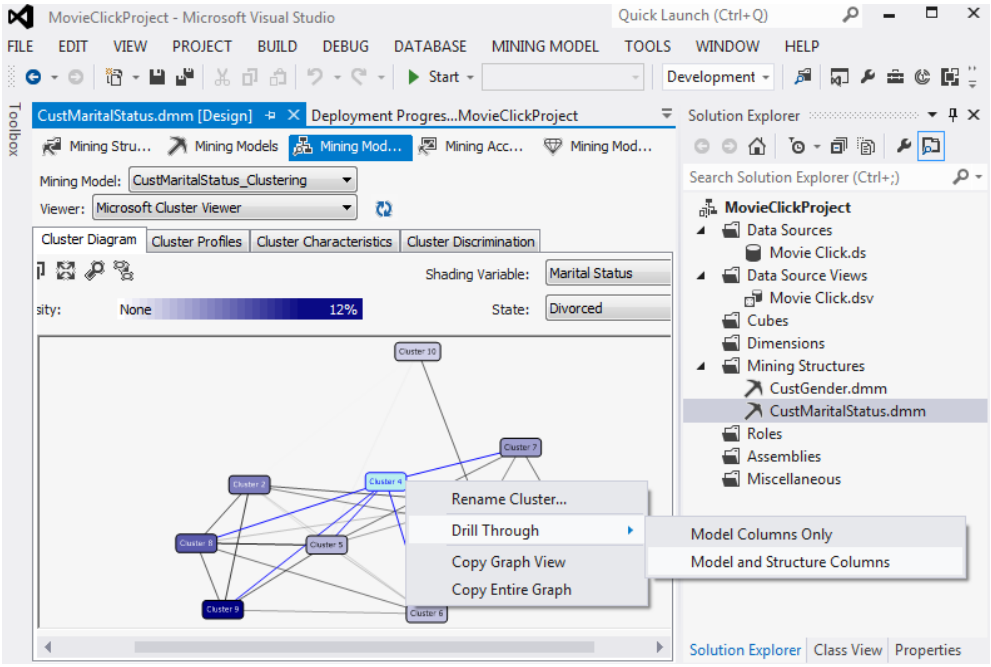
13. Ακολούθως, εμφανίζεται ένα παράθυρο που παρουσιάζει τις ενέργειες που έγιναν για τη δημιουργία του μοντέλου και πληροφορεί αν αυτές ολοκληρώθηκαν με επιτυχία. Επιλέγουμε Close, ώστε να ολοκληρωθεί η διαδικασία και να προβάλουμε το μοντέλο.

14. Επιλέγοντας την καρτέλα Cluster Diagram στον Mining Model Viewer εμφανίζονται τα Clusters που έχουν δημιουργηθεί. Παρατηρούμε, όπως φαίνεται στην Εικόνα 8.13, ότι όσο πιο ανοιχτό είναι το χρώμα ενός cluster τόσο πιο μικρός είναι ο αριθμός των περιπτώσεων που ομαδοποιούνται σ' αυτό το cluster, δηλαδή τόσο πιο μικρό είναι το πλήθος των περιπτώσεων που το αποτελούν.
Αν αφήσουμε τον κέρσορα του ποντικιού πάνω σε ένα cluster, παρατηρούμε ότι εμφανίζονται κάποια στατιστικά στοιχεία. Αυτά τα στατιστικά στοιχεία για κάθε cluster σχετίζονται με την παράμετρο που θέτουμε στο πεδίο Shading Variable. Για παράδειγμα, επιλέγοντας Shading Variable = Marital Status και State = Divorced, όπως φαίνεται στην Εικόνα 8.13, βλέπουμε ότι το Cluster 8 εμπεριέχει το 5% των διαζευγμένων (divorced).
Παρατηρούμε ότι όσο πιο κοντά είναι τα clusters μεταξύ τους και όσο παχύτερη είναι η γραμμή που τα συνδέει, τόσο μεγαλύτερες είναι οι μεταξύ τους ομοιότητες. Επομένως, παρατηρώντας αυτό το διάγραμμα, μπορούμε να καταλάβουμε αφενός πόσο μεγάλη είναι η σχέση μεταξύ των clusters και αφετέρου ποια clusters έχουν τις περισσότερες ομοιότητες. Παρατηρούμε, ακόμη, ότι ο αριθμός των clusters που σχημάτισε ο αλγόριθμος είναι 10, κάτι που συνέβη επειδή η τιμή της παραμέτρου CLUSTER_COUNT είχε αφεθεί στην προεπιλεγμένη τιμή 10.
Παρατηρούμε, τέλος, ότι στα αριστερά του γραφήματος υπάρχει μια μπάρα, η μετακίνηση της οποίας παρουσιάζει τον βαθμό συσχέτισης μεταξύ των clusters. Η κλίμακα διαβάθμισης γίνεται από το χαμηλότερο προς το υψηλότερο επίπεδο της μπάρας με το το χαμηλότερο να δηλώνει τη μεγαλύτερη συσχέτιση μεταξύ των clusters και το υψηλότερο τη μικρότερη.



Εικόνα 8.13

15. Στη συνέχεια, όπως φαίνεται στην Εικόνα 8.14, κάνουμε δεξί κλικ σε ένα συγκεκριμένο cluster. το cluster 4. Επιλέγουμε Drill Through και Model and Structure Colums, προκειμένου να δούμε αναλυτικά τις εγγραφές που εντάσσονται σ' αυτό το cluster.



Εικόνα 8.14

Στην Εικόνα 8.15 εμφανίζονται αναλυτικά όλες οι περιπτώσεις που ομαδοποιούνται στο συγκεκριμένο cluster. Τα στοιχεία που εμφανίζονται σ' αυτό το παράθυρο είναι πάρα πολύ σημαντικά, καθώς μπορούμε να δούμε την ομοιογένεια των περιπτώσεων που περιέχονται σε κάθε cluster.

Drill Through

Cases Classified to:

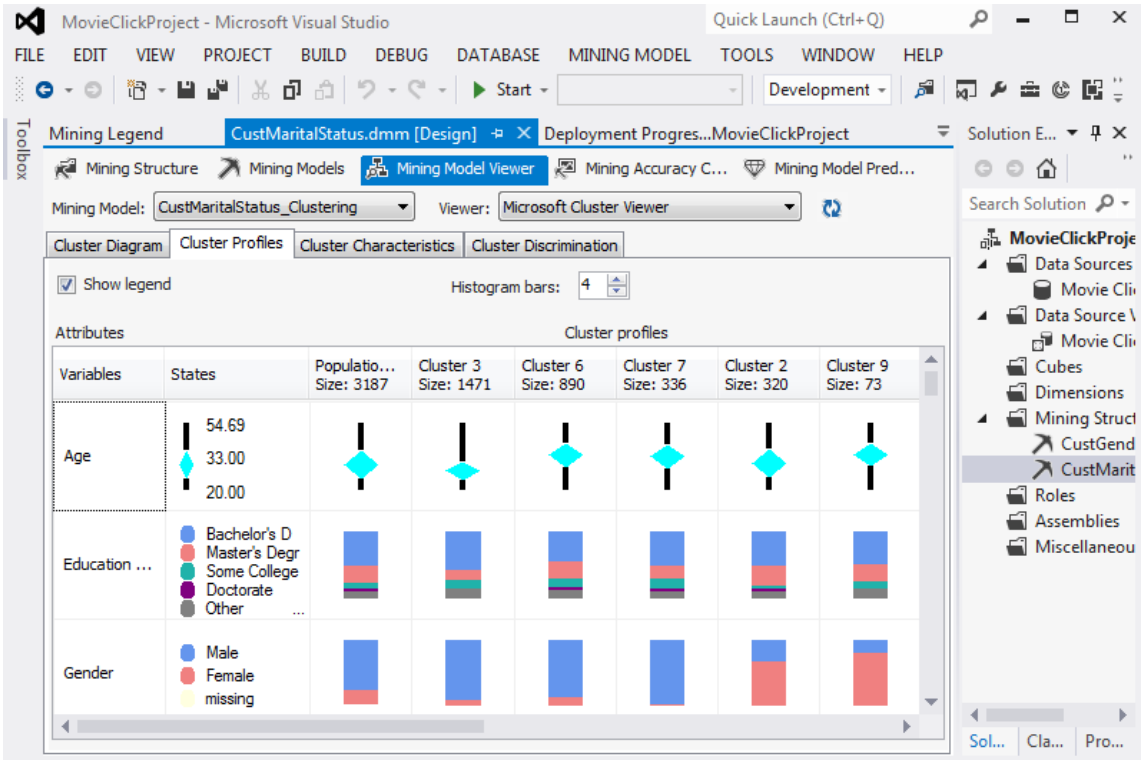
Cluster 4

Age	Customer ID	Education Level	Gender	Marital Status	Num Children	Hobbies
32	878993	Bachelor's De...	Male	Married	1	+ Hobbies
32	882332	Master's Degree	Male	Married	2	+ Hobbies
31	882342	Some College	Male	Separated	3	+ Hobbies
35	882616	Associate's De...	Male	Married	2	+ Hobbies
37	883449	Bachelor's De...	Male	Married	3	+ Hobbies
28	885269	Post-Doc	Male	Married	1	+ Hobbies
30	885344	Bachelor's De...	Male	Married	1	+ Hobbies
34	886455	Associate's De...	Male	Married	3	+ Hobbies
26	888104	Some College	Male	Married	2	+ Hobbies
30	888378	Bachelor's De...	Male	Married	1	+ Hobbies
32	888389	Bachelor's De...	Male	Married	1	+ Hobbies
27	888736	Bachelor's De...	Male	Married	2	+ Hobbies
33	888868	Some College	Male	Divorced	1	+ Hobbies
25	889109	Bachelor's De...	Male	Married	1	+ Hobbies
31	889592	Master's Degree	Male	Married	1	+ Hobbies
35	889743	Bachelor's De...	Male	Married	1	+ Hobbies
28	889894	Master's Degree	Male	Married	1	+ Hobbies
31	889960	Bachelor's De...	Male	Married	2	+ Hobbies
37	890353	Bachelor's De...	Male	Married	2	+ Hobbies
36	890698	High School	Male	Married	2	+ Hobbies

45

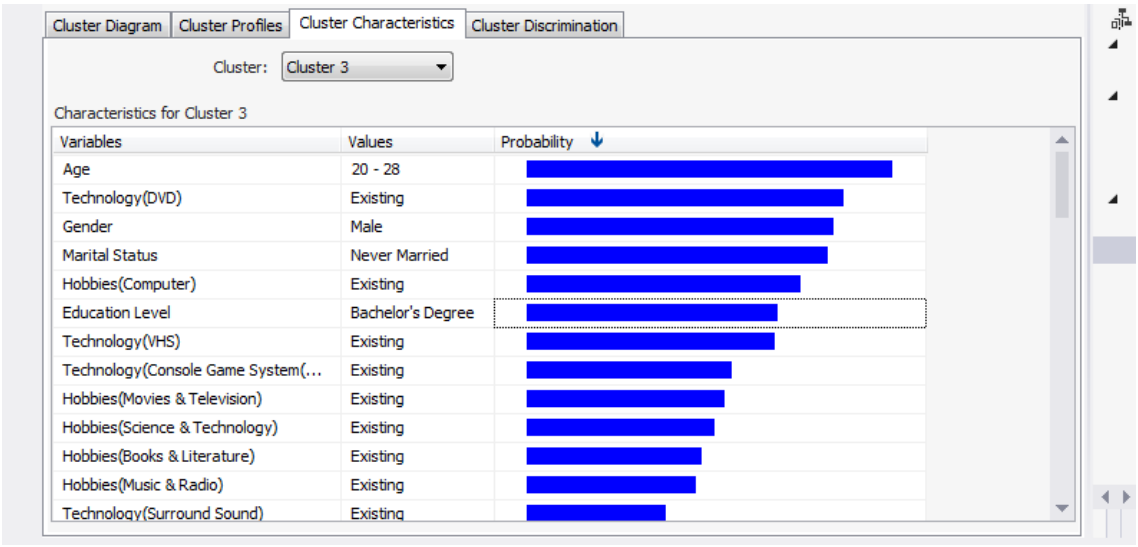
Εικόνα 8.15

16. Στη συνέχεια, επιλέγουμε την καρτέλα Cluster Profiles, όπως φαίνεται στην Εικόνα 8.16, ώστε να εμφανιστούν συγκεντρωτικά οι τιμές όλων των χαρακτηριστικών που επιλέχθηκαν κατά την κατασκευή του μοντέλου, για κάθε cluster ξεχωριστά. Παρατηρούμε τη σύνοψη των βασικών τάσεων των χαρακτηριστικών των clusters, η οποία είναι απλή, κατανοητή και κατατοπιστική. Για παράδειγμα, για το Cluster 3 και το χαρακτηριστικό Επίπεδο μόρφωσης (Education Level), το οποίο παίρνει διακριτές τιμές, βλέπουμε ότι με μπλε χρώμα υπερέχουν αυτοί που είναι κάτοχοι πτυχίου πανεπιστημίου (Bachelor's degree). Επίσης, για το χαρακτηριστικό ηλικία (age), το οποίο παίρνει συνεχείς τιμές, βλέπουμε ότι ο μέσος όρος ηλικίας των πελατών που εντάσσονται στο cluster 3 είναι κάτω των 33 ετών. Όπως παρατηρούμε, η -με τον ίδιο τρόπο- τοποθέτηση των χαρακτηριστικών ανά cluster επιτρέπει την ταυτόχρονη οπτική σύγκριση των τιμών τους μεταξύ των διαφορετικών ομάδων.



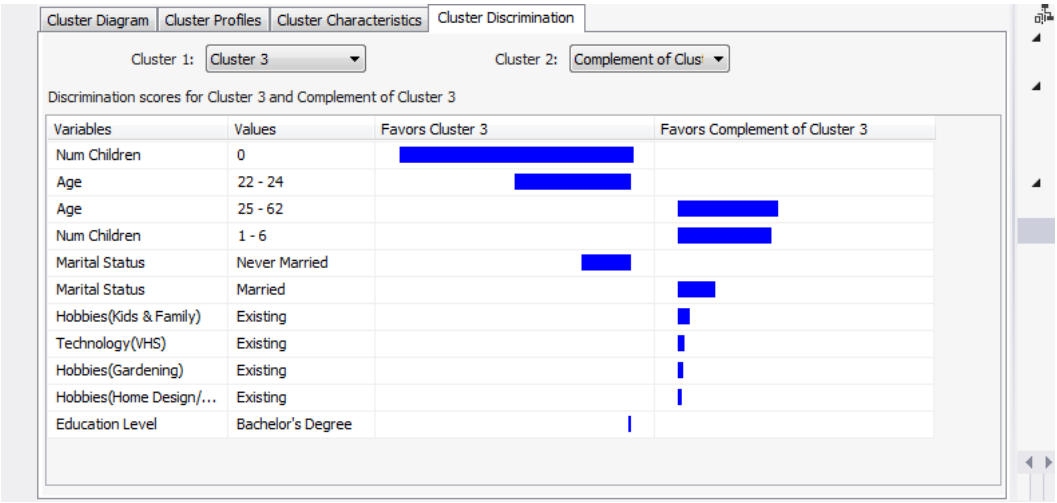
Εικόνα 8.16

17. Στη συνέχεια, όπως φαίνεται στην Εικόνα 8.17, επιλέγουμε την καρτέλα Cluster Characteristics, ώστε να δούμε ακόμα πιο αναλυτικά τα χαρακτηριστικά και τις τιμές για κάθε συγκεκριμένο cluster που μας ενδιαφέρει. Για παράδειγμα, όπως φαίνεται στην ίδια Εικόνα, για το cluster 3 εμφανίζεται ένας πίνακας που περιγράφει τα χαρακτηριστικά των εγγραφών που εντάσσονται σ' αυτό, προβάλλοντας ταυτόχρονα τις τιμές αυτών των χαρακτηριστικών με φθίνουσα πιθανότητα εμφάνισης. Στο συγκεκριμένο cluster, όπως βλέπουμε, ανήκουν πελάτες που είναι ηλικιακά μεταξύ 20-28 ετών, νοικιάζουν DVD ταινίες, είναι άγαμοι άνδρες κτλ.



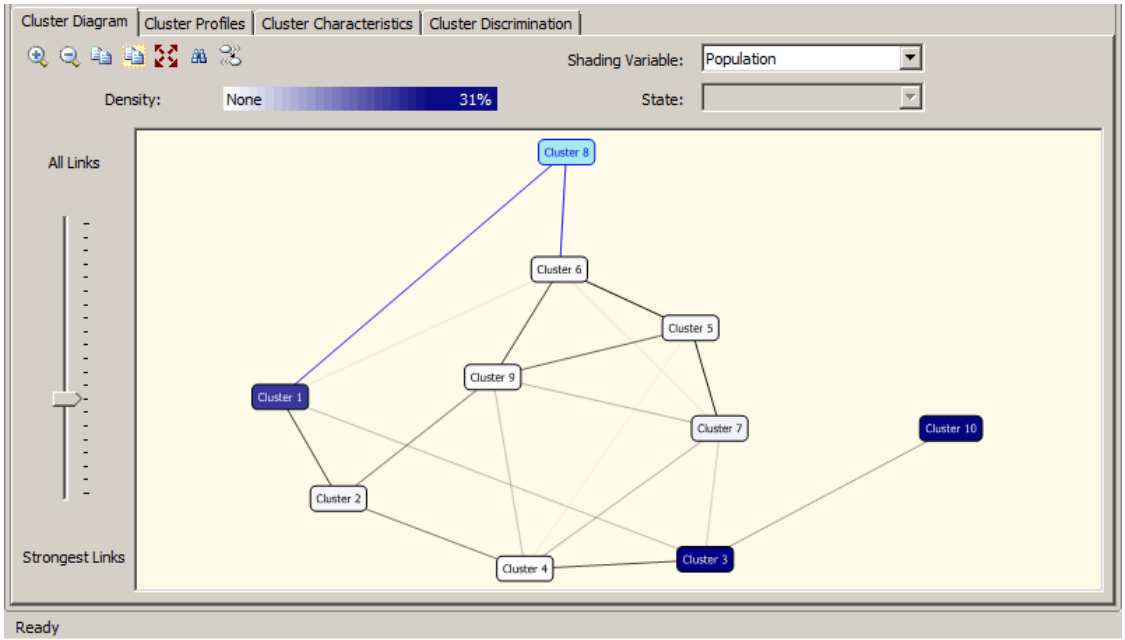
Εικόνα 8.17

18. Στη συνέχεια, όπως φαίνεται στην Εικόνα 8.18, επιλέγουμε την καρτέλα Cluster Discrimination, ώστε να βρούμε τα σημαντικότερα χαρακτηριστικά στοιχεία του cluster που μας ενδιαφέρει και να καταλάβουμε την ομάδα πελατών που εντάσσονται σ' αυτό. Θα εντοπίσουμε τα σημαντικότερα στοιχεία του cluster μέσα από τη σύγκριση των στοιχείων που υπάρχουν στο ίδιο cluster με όλα τα στοιχεία που βρίσκονται έξω από αυτό, δηλαδή με το συμπλήρωμά του (complement of cluster). Στο πεδίο Cluster 1, όπως φαίνεται στην ίδια Εικόνα, συμπληρώνουμε το cluster που μας ενδιαφέρει και στο πεδίο Cluster 2 συμπληρώνουμε το συμπλήρωμα του. Στη συγκεκριμένη περίπτωση, επιλέγουμε το Cluster 3 και το συμπλήρωμα του αντίστοιχα. Τα αποτελέσματα του πίνακα επιβεβαιώνουν την εκτίμηση που πήραμε από την προηγούμενη καρτέλα (Cluster Characteristics). καθώς οι πελάτες που εντάσσονται και σε αυτό το cluster (Cluster Discrimination) είναι νέοι, άγαμοι και άτεκνοι. Βάσει των χαρακτηριστικών αυτών, μπορούμε να καταλάβουμε τι είδους πελάτες ανήκουν στο cluster 3 και, ενδεχομένως, να μετονομάσουμε το συγκεκριμένο cluster με κάποιο όνομα που το προσδιορίζει καλύτερα.



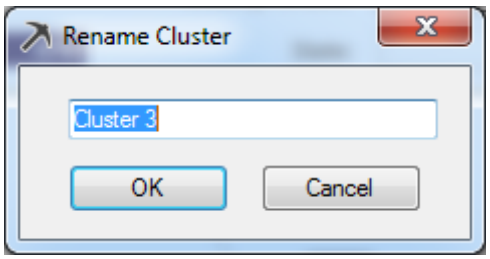
Εικόνα 8.18

19. Πριν όμως μετονομάσουμε το cluster, καλό είναι να λάβουμε υπόψη μας ότι μπορεί να είναι παρόμοιο με άλλα clusters. Γι' αυτό, πρέπει πρώτα να συγκριθεί με τα άλλα γειτονικά του clusters. Επιστρέφουμε, λοιπόν, στην καρτέλα Cluster Diagram και, όπως φαίνεται στην Εικόνα 8.19, παρατηρούμε ότι, αν ανεβάσουμε διαδοχικά τη μπάρα από το χαμηλότερο προς το ανώτερο επίπεδο, το cluster 3 σχετίζεται περισσότερο με τα clusters 10, 4, 1 και 7.



Εικόνα 8.19

Επειδή, όμως, μας ενδιαφέρουν τα clusters με τα οποία το cluster 3 έχει την ισχυρότερη σχέση, μελετάμε τις περιπτώσεις που ομαδοποιούνται σ' αυτά τα clusters επιλέγοντας τις καρτέλες Cluster Profiles, Cluster Characteristics και Cluster Discrimination. Όταν καταλήξουμε στο όνομα που προσδιορίζει το cluster με μεγαλύτερη σαφήνεια, το μετονομάζουμε, κάνοντας δεξί κλικ επάνω στο cluster και επιλέγοντας Rename Cluster, όπως φαίνεται στην Εικόνα 8.20.



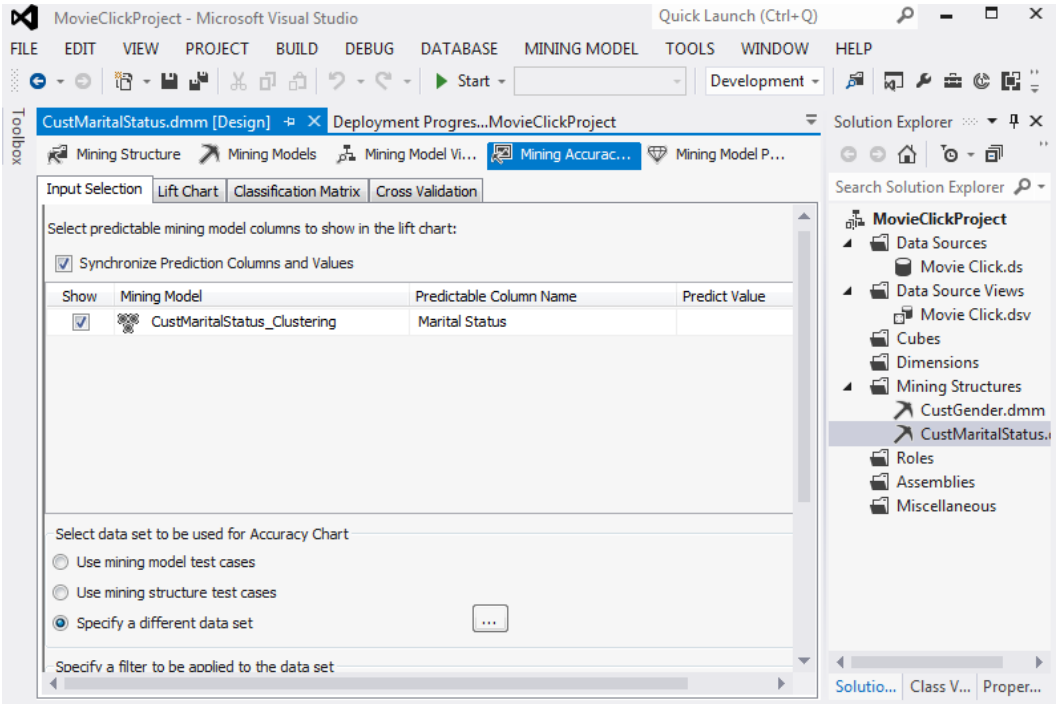
Εικόνα 8.20

8.3. Αξιολόγηση Μοντέλου Clustering

Σ' αυτήν την ενότητα θα εξετάσουμε αν το μοντέλο που δημιουργήσαμε μπορεί (ή δεν μπορεί) να προβλέψει με ακρίβεια την οικογενειακή κατάσταση των πελατών. Η αξιολόγηση του μοντέλου θα πραγματοποιηθεί με δύο τρόπους: α) την ερμηνεία του Lift Chart και β) τη μελέτη των περιπτώσεων που ανήκουν σε κάθε cluster, επιλέγοντας Drill Through στα clusters.

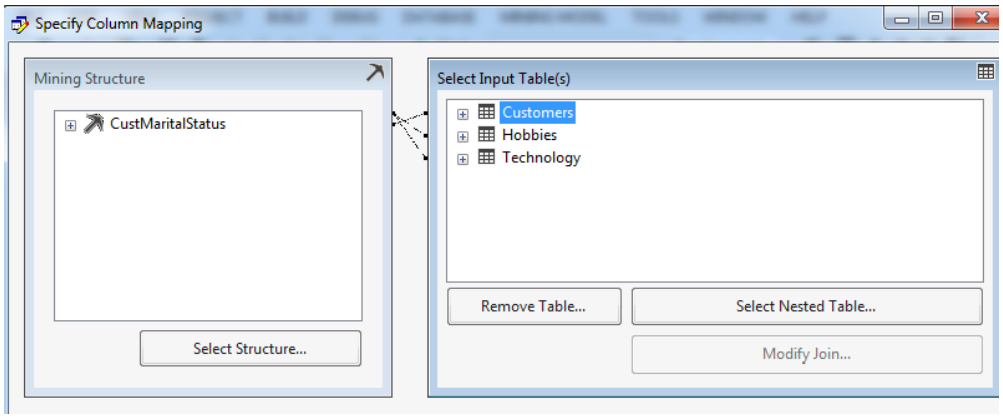
8.3.1. Αξιολογώντας το μοντέλο με τη χρήση του Lift chart

1. Επιλέγουμε την καρτέλα Mining Accuracy Chart και, στη συνέχεια, την καρτέλα Column Mapping, όπως φαίνεται στην Εικόνα 8.21. Στο πεδίο Select data set to be used for Accuracy Chart επιλέγουμε το Specify a different data set.



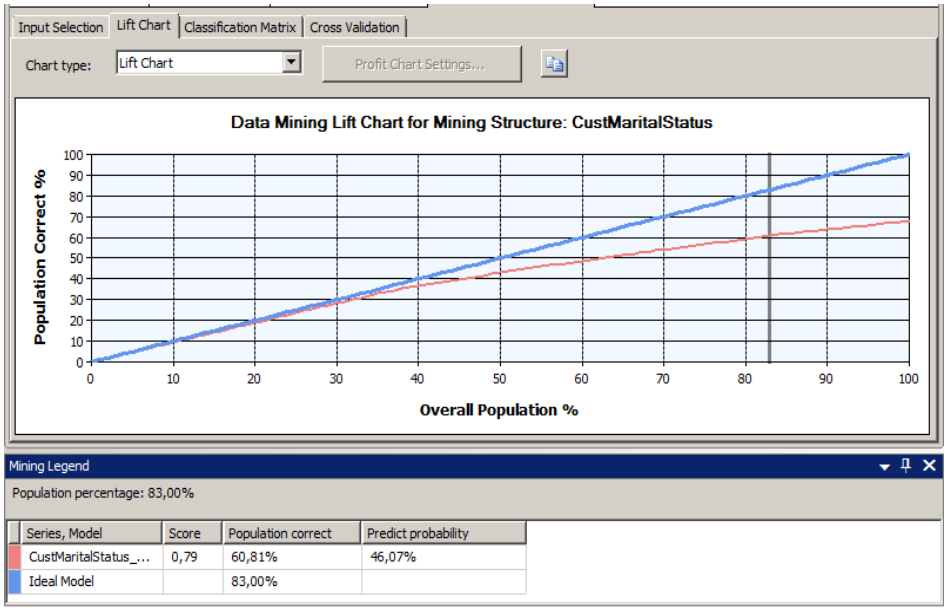
Εικόνα 8.21

2. Εμφανίζεται το παράθυρο με το Mining Structure. Επιλέγουμε **Select Structure,** ώστε να προσδιορίσουμε το μοντέλο μας (CustMaritalStatus). Στη συνέχεια, κάνουμε κλικ στο Select Case Table και επιλέγουμε τον πίνακα **Customers.** Εμφανίζεται ξανά το παράθυρο με το Mining Structure και τους πίνακες, όπου βλέπουμε τις σχέσεις που έχουν δημιουργηθεί. Στη συνέχεια, επιλέγουμε Select Nested Table, ώστε να επιλέξουμε τους πίνακες **Technology** και **Hobbies**, όπως φαίνεται στην Εικόνα 8.22.



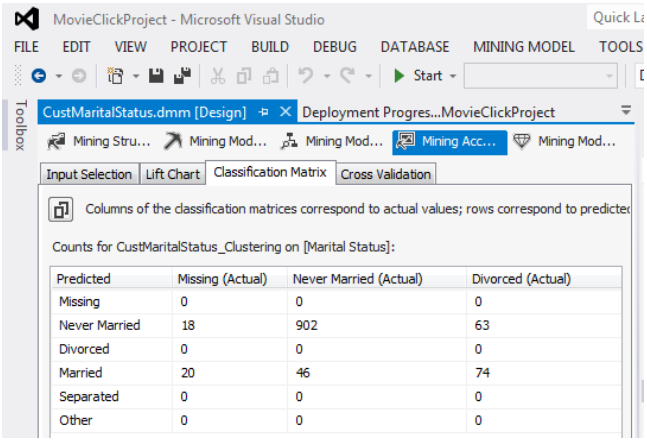
Εικόνα 8.22

3. Στη συνέχεια, όπως φαίνεται στην Εικόνα 8.23, επιλέγουμε την καρτέλα Lift Chart και εμφανίζεται το σχετικό διάγραμμα. Το διάγραμμα εκφράζει το ποσοστό του συνολικού πληθυσμού (άξονας X) σε σχέση με το ποσοστό του πληθυσμού που έχουμε προβλέψει σωστά (άξονας Y). Η μπλε γραμμή του άξονα X, που έχει κλήση 45^{o}, δείχνει την επίδοση του ιδανικού μοντέλου που θα προέβλεπε σωστά την οικογενειακή κατάσταση όλων των πελατών. Η κόκκινη καμπύλη που βρίσκεται κάτω από την μπλε γραμμή εκφράζει την επίδοση του δικού μας μοντέλου. Κάνουμε κλικ στο 83% πάνω στο διάγραμμα, όπως φαίνεται στην Εικόνα 8.23, για να δούμε τα σχετικά στατιστικά τα οποία καταγράφονται στο παράθυρο Mining Legend. Βλέπουμε ότι στο 83% του συνολικού πληθυσμού (που είναι το δείγμα μας), το μοντέλο μας προβλέπει σωστά το 60.81% του δείγματος, ενώ το ιδανικό είναι να προβλέπει σωστά το 83%. Το Score είναι 0.79.



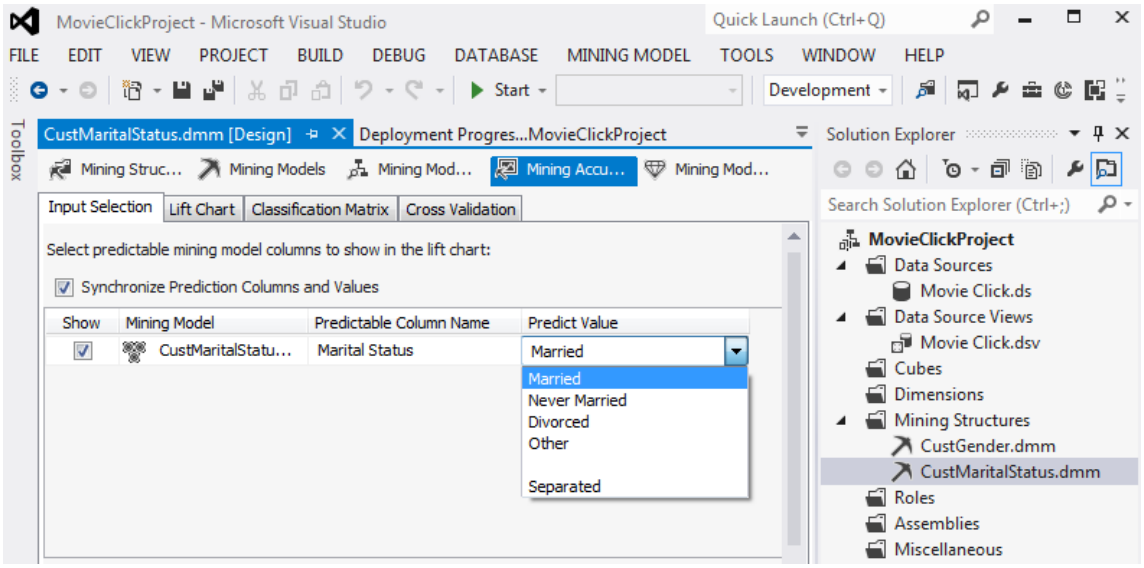
Εικόνα 8.23

4. Στην συνέχεια, επιλέγουμε την καρτέλα Classification Matrix, όπου βλέπουμε ποιες είναι οι πραγματικές τιμές των χαρακτηριστικών και ποιες τιμές αποδόθηκαν στα χαρακτηριστικά αυτά από το μοντέλο μας. Όπως φαίνεται στην Εικόνα 8.24, όσον αφορά τους πελάτες που δεν έχουν παντρευτεί (Never Married), ο αλγόριθμος τούς προβλέπει πολύ σωστά (902 σωστές προβλέψεις επί συνόλου 948 πελατών που δεν έχουν παντρευτεί) με ποσοστό επιτυχούς πρόβλεψης περίπου 95%. Όσον αφορά, όμως, τους διαζευγμένους (Divorced), ο αλγόριθμος δεν τους προβλέπει καθόλου καλά (0 σωστές προβλέψεις επί συνόλου 137) με ποσοστό επιτυχούς πρόβλεψης 0%. Ίσως αυτό το μη επιτυχές αποτέλεσμα να οφείλεται σε μη επαρκές δείγμα διαζευγμένων μέσα στο σύνολο δεδομένων μας.



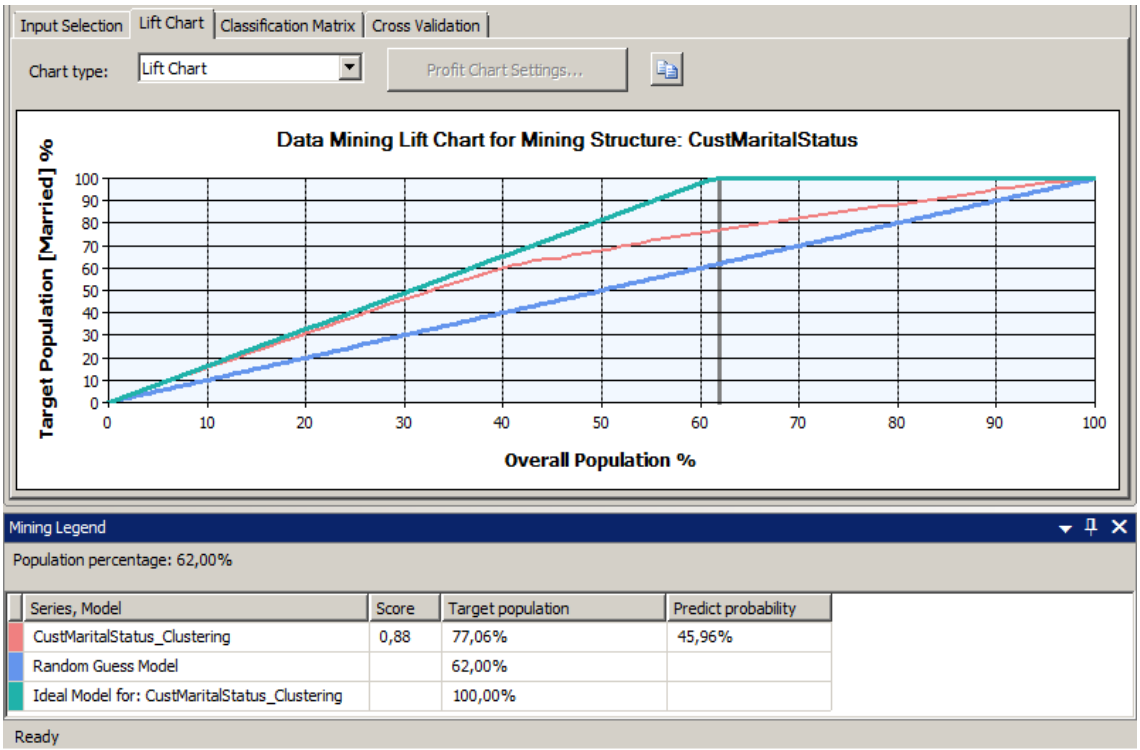
Εικόνα 8.24

5. Στη συνέχεια, θα προβλέψουμε την τιμή Married για το χαρακτηριστικό Marital Status. Επιλέγουμε την καρτέλα Column Mapping, όπως φαίνεται στην Εικόνα 8.25. Στο πεδίο Predictable Column Name επιλέγουμε Marital Status. Στο πεδίο Predict Value επιλέγουμε Married.



Εικόνα 8.25

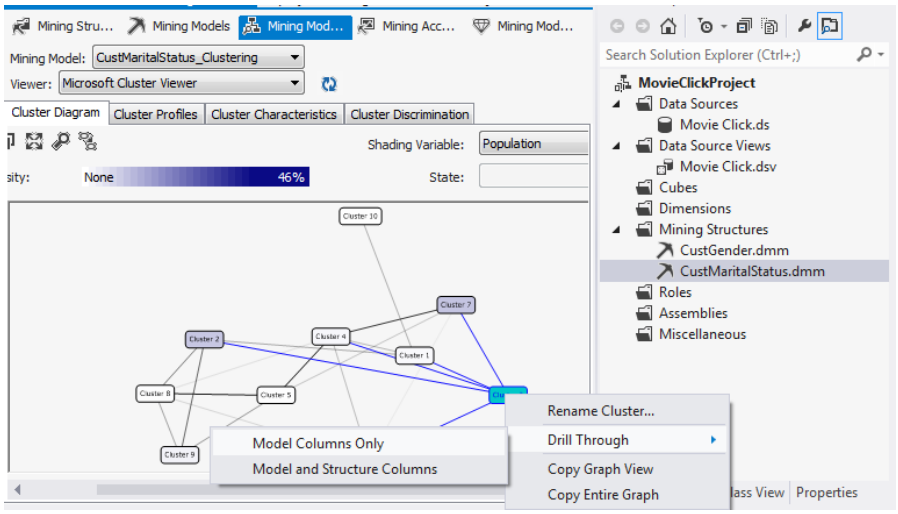
6. Στη συνέχεια, επιλέγουμε την καρτέλα Lift Chart και εμφανίζεται το σχετικό διάγραμμα, όπως φαίνεται στην Εικόνα 8.26. Σ' αυτήν την περίπτωση, ο άξονας Y δείχνει το ποσοστό των παντρεμένων που προβλέφθηκε σωστά. Η μπλε γραμμή, που έχει κλήση 45°, δείχνει το τυχαίο μοντέλο. Η πράσινη γραμμή, που βρίσκεται πιο ψηλά απ' όλες, δείχνει το ιδανικό μοντέλο. Η κόκκινη γραμμή αντιπροσωπεύει το δικό μας μοντέλο. Στο παράδειγμά μας, το ιδανικό μοντέλο πετυχαίνει το 100% των προβλέψεων με το 62% του συνολικού πληθυσμού. Στον πίνακα Mining Legend βλέπουμε ότι το μοντέλο μας έχει Score 0.88 και προβλέπει την τιμή που θέλουμε με ακρίβεια 77.06%.



Εικόνα 8.26

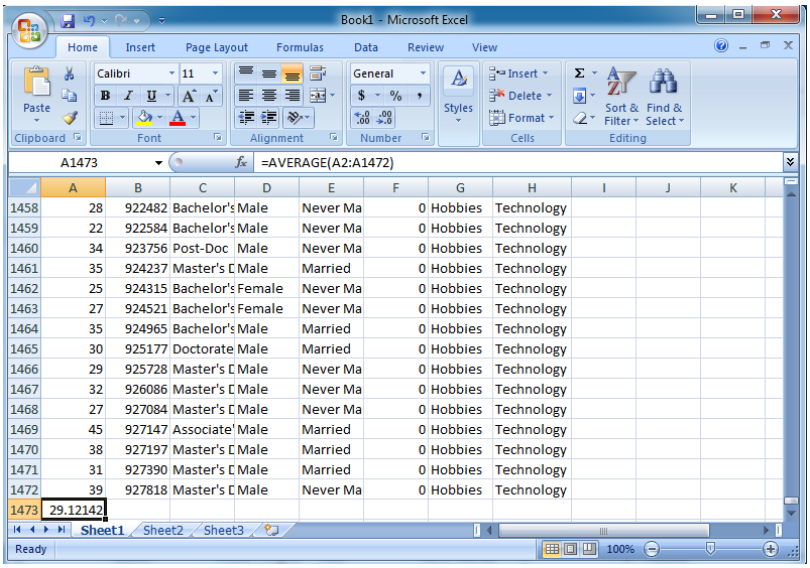
8.3.2. Αξιολόγηση ενός μοντέλου με τη χρήση του Drill through

Σ' αυτήν την ενότητα θα μελετήσουμε τη δυνατότητα αξιολόγησης ενός μοντέλου με έναν εναλλακτικό τρόπο. Αυτό που θα πρέπει να προσέξουμε είναι ότι με το Lift Chart, που περιγράφηκε στην προηγούμενη ενότητα, αξιολογούμε την απόδοση του αλγορίθμου σε όλα τα clusters συνολικά. Στην περίπτωση, όμως, που θέλουμε να αξιολογήσουμε την ομοιογένεια των μελών ενός cluster, πρέπει να έχουμε πρόσβαση στα instances που συγκροτούν το κάθε cluster, έτσι ώστε να κάνουμε τις κατάλληλες συγκρίσεις. Αυτό γίνεται κάνοντας Drill Through σε κάθε cluster και συγκρίνοντας την ομοιογένειά του ως προς τα μέλη που ενσωματώνει (στην περίπτωσή μας, τους πελάτες). Ας υποθέσουμε, για την περίπτωση μας, ότι επιλέγουμε το cluster 3. Επιλέγουμε την καρτέλα Mining Model Viewer και, στη συνέχεια, την καρτέλα Cluster Diagram, όπως φαίνεται στην Εικόνα 8.27, Κάνουμε δεξί κλικ επάνω σε ένα cluster και επιλέγουμε Drill Through και Model Columns Only.



Εικόνα 8.27

Εμφανίζεται το παράθυρο με όλες τις περιπτώσεις του cluster, όπως φαίνεται στην Εικόνα 8.28. Στη συνέχεια, κάνουμε δεξί κλικ και επιλέγουμε Copy All. Επειδή δεν μπορούμε να επεξεργαστούμε αυτά τα δεδομένα στον SQL Server, μπορούμε να τα αντιγράψουμε σε διάφορα άλλα εργαλεία (Microsoft Excel, SPSS κτλ). Στη συγκεκριμένη περίπτωση, χρησιμοποιούμε το Microsoft Excel επιλέγοντας Επικόλληση σε ένα φύλλο. Μπορούμε τώρα να κάνουμε τους υπολογισμούς που θέλουμε, ώστε να αξιολογήσουμε την ομοιογένεια των στοιχείων του cluster 3, δηλαδή την αποτελεσματικότητα του αλγορίθμου. Για παράδειγμα, μπορούμε να υπολογίσουμε τον μέσο όρο ηλικίας των πελατών στο Cluster 3.



Εικόνα 8.28

8.4. Ασκήσεις στην ομαδοποίηση δεδομένων

1. Να συγκριθούν τα δεδομένα του cluster 3 με τα δεδομένα του cluster 4 στο ήδη δημιουργηθέν μοντέλο (της Εικόνας 8.10) που αναπτύχθηκε στο κεφάλαιο 8.

2. Να βρεθούν τα χαρακτηριστικά των πελατών του cluster 4 και, στη συνέχεια, να μετονομαστεί αυτό καταλλήλως.

3. Στο ήδη δημιουργηθέν μοντέλο (της Εικόνας 8.10) να αλλάξετε την τιμή της παραμέτρου CLUSTER_COUNT, ορίζοντάς την σε 5. Να εμφανίσετε και να σχολιάσετε τα παρακάτω:

   a) τα νέα clusters που θα δημιουργηθούν και
   b) το Lift Chart και την ακρίβεια πρόβλεψης του νέου μοντέλου.

4. Να επαναλάβετε την άσκηση 3, επιλέγοντας τον αλγόριθμο k-means (CLUSTERING_METHOD = 3), όπου το πλήθος των clusters να υπολογίζεται ευρηστικά από τον ίδιο τον αλγόριθμο.

5. Να επαναλάβετε την άσκηση 3, επιλέγοντας τον αλγόριθμο k-means (CLUSTERING_METHOD = 3), όπου το πλήθος των clusters να υπολογίζεται ευρηστικά από τον ίδιο τον αλγόριθμο. Επιπροσθέτως, να ορίσετε ως κατώτατο πλήθος περιπτώσεων ανά cluster τις 50.

6. Στην καρτέλα Mining Models να δημιουργήσετε δύο μοντέλα. Ένα μοντέλο να γίνει με τη χρήση του αλγορίθμου Decision Tree και ένα με τη χρήση του αλγορίθμου Clustering. Οι αλγόριθμοι να επεξεργάζονται τα ίδια ακριβώς δεδομένα μ' αυτά της Εικόνας 8.10. Στη συνέχεια, να συγκριθούν αυτά τα δύο μοντέλα.

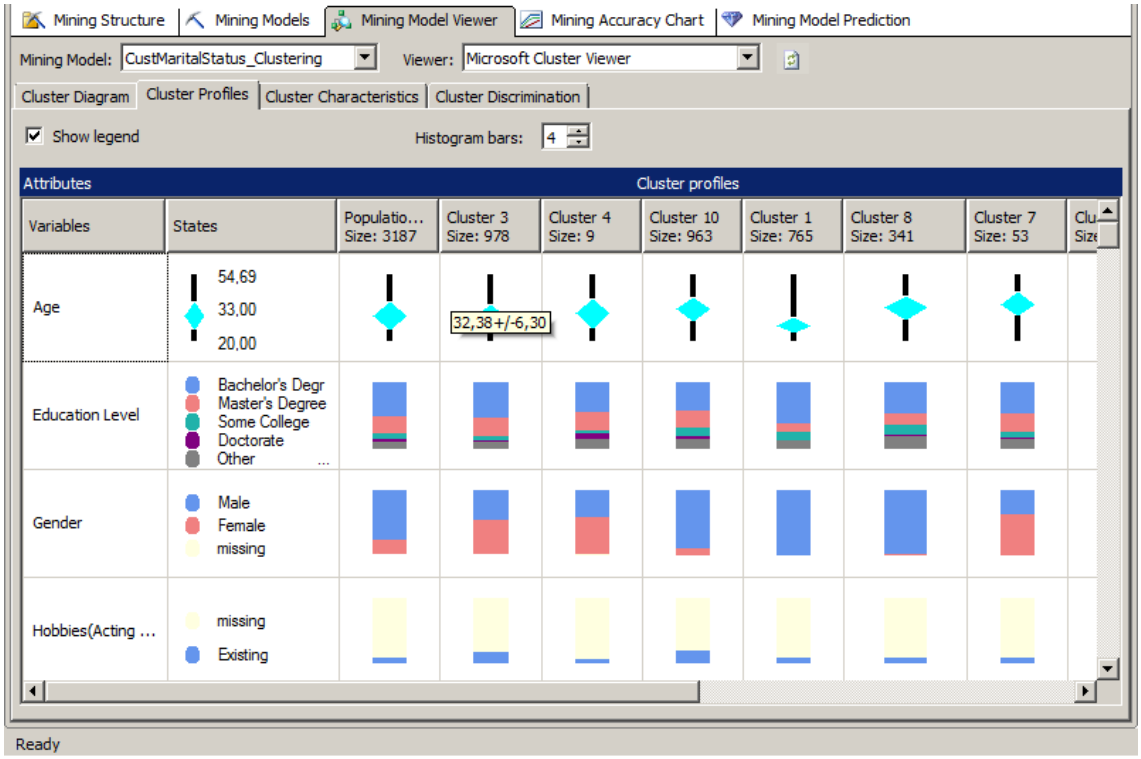
8.5. Λύσεις ασκήσεων στην ομαδοποίηση δεδομένων

Άσκηση 1

Να συγκριθούν τα δεδομένα του cluster 3 με τα δεδομένα του cluster 4 στο ήδη δημιουργηθέν μοντέλο (της Εικόνας 8.10) που αναπτύχθηκε στο κεφάλαιο 8.

Λύση

1. Επιλέγουμε την καρτέλα Cluster Profiles, ώστε να εμφανιστούν αναλυτικά οι τιμές όλων των χαρακτηριστικών για τα cluster 3 και cluster 4. Κοιτάζοντας με προσοχή τα δεδομένα, παρατηρούμε ότι οι πελάτες που ανήκουν σ' αυτά τα δύο clusters έχουν πάρα πολλές ομοιότητες και δικαιολογημένα στη καρτέλα Cluster Diagram, αυτά τα δύο clusters εμφανίζονται να συσχετίζονται τόσο πολύ. Πιο συγκεκριμένα, όπως φαίνεται στην Εικόνα 8.29, οι πελάτες αυτών των clusters ανήκουν στο ίδιο φύλο, έχουν την ίδια περίπου ηλικία και διάθετουν το ίδιο περίπου μορφωτικό επίπεδο.



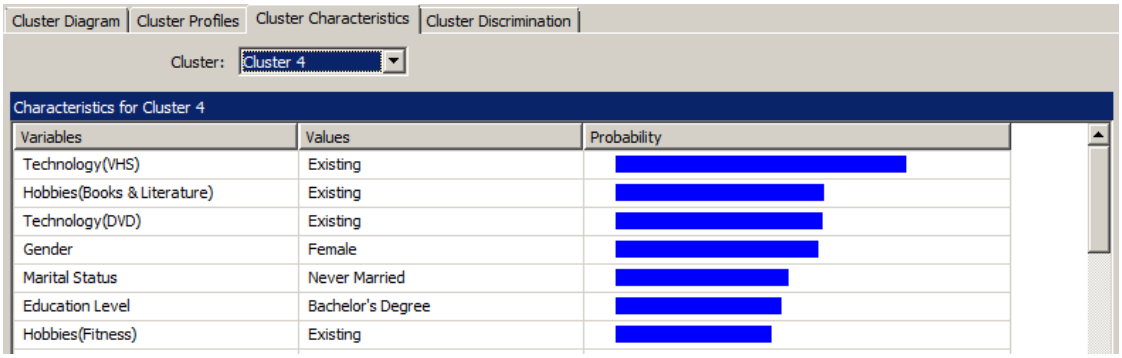
Εικόνα 8.29

Άσκηση 2

Να βρεθούν τα χαρακτηριστικά των πελατών του cluster 4 και, στη συνέχεια, να μετονομαστεί αυτό καταλλήλως.

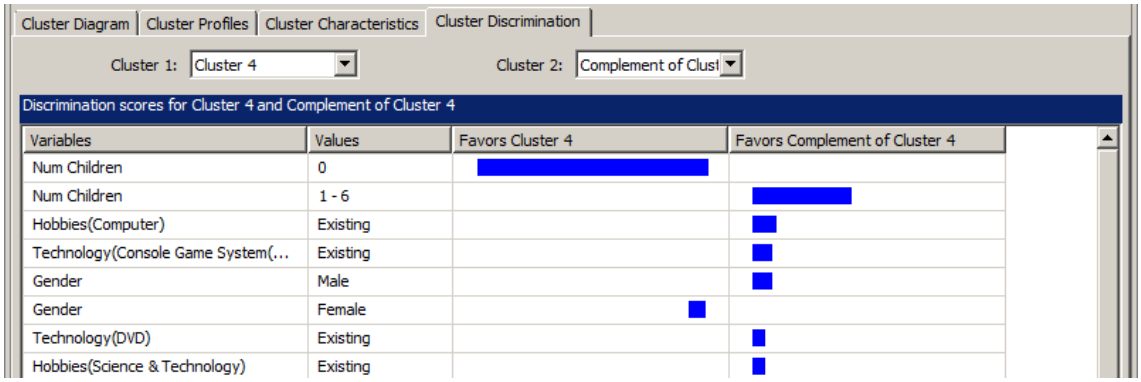
Λύση

1. Στο Cluster Characteristics tab, όπως φαίνεται στην Εικόνα 8.30, βλέπουμε τις τιμές των χαρακτηριστικών των πελατών του cluster 4 σε φθίνουσα σειρά ως προς την πιθανότατα εμφάνισης ενός χαρακτηριστικού. Αυτή η κατηγορία πελατών είναι κυρίως άγαμες γυναίκες, που ενδιαφέρονται κυρίως για την παρακαλούθηση ταινιών και τη λογοτεχνία.



Εικόνα 8.30

2. Αν, όμως, θέλουμε να μελετήσουμε τα πιο σημαντικά χαρακτηριστικά που προσδιορίζουν αυτό το cluster, πρέπει να ανοίξουμε την καρτέλα Cluster Discrimination tab. Βλέπουμε τότε, όπως φαίνεται στην Εικόνα 8.30, ότι αυτό το cluster αποτελείται από γυναίκες που δεν έχουν παιδιά, δεν αρέσουν τους υπολογιστές, δεν έχουν DVD player ή Console Game System κλπ. Επομένως, τώρα που έχουμε μια πιο ολοκληρωμένη εικόνα αυτού του cluster, μπορούμε να το μετονομάσουμε, αφού πρώτα το συγκρίνουμε με τα άλλα clusters με τα οποία συνδέεται στενά, κάτι που βλέπουμε στο Cluster Diagram tab.



Εικόνα 8.31

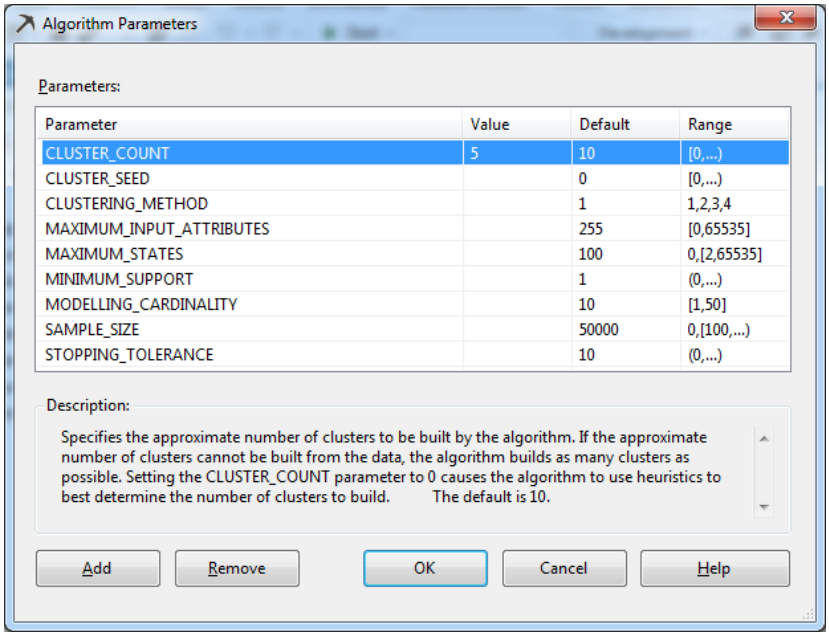
Άσκηση 3

Στο ήδη δημιουργηθέν μοντέλο (της Εικόνας 8.10) να αλλάξετε την τιμή της παραμέτρου CLUSTER_COUNT, ορίζοντάς την σε 5. Να εμφανίσετε και να σχολιάσετε τα παρακάτω:

c) τα νέα clusters που θα δημιουργηθούν και
d) το Lift Chart και την ακρίβεια πρόβλεψης του νέου μοντέλου.

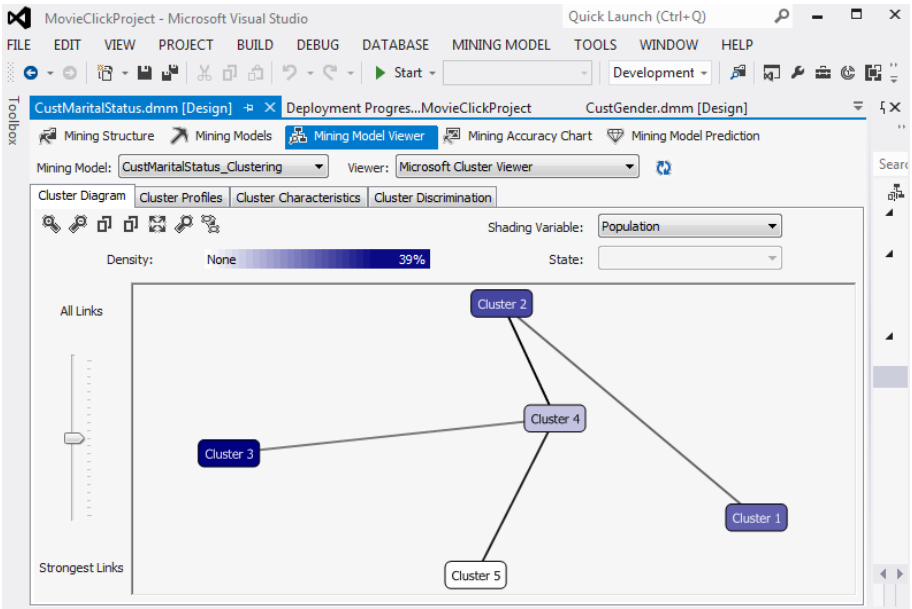
Λύση

1. Αλλάζουμε την τιμή της παραμέτρου CLUSTER_COUNT, δίνοντάς της την τιμή 5.



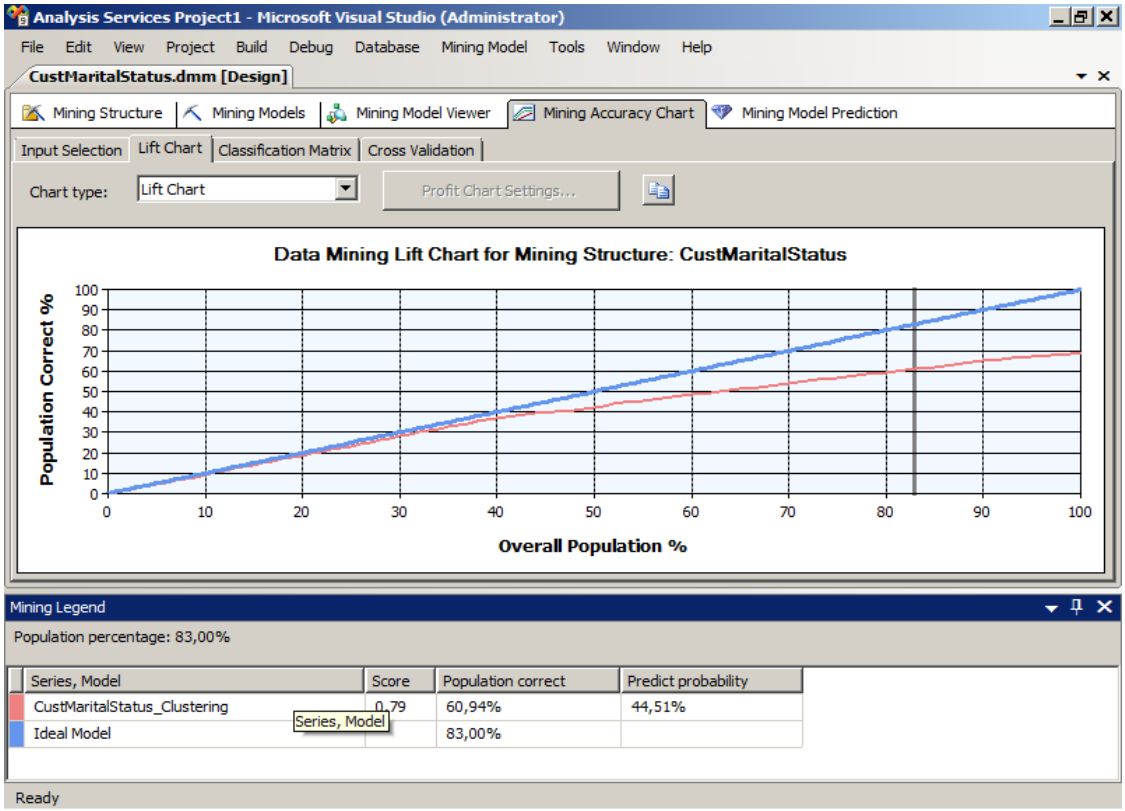
Εικόνα 8.32

2. Δημιουργούνται 5 clusters, όπως εμφανίζονται στο Cluster Diagram της Εικόνας 8.33.



Εικόνα 8.33

3. Επιλέγουμε την καρτέλα Lift Chart, οπότε εμφανίζεται το σχετικό διάγραμμα. Κάνουμε κλικ στο 83%, πάνω στο διάγραμμα, για να δούμε τα σχετικά στατιστικά που καταγράφονται στο παράθυρο Mining Legend. Όπως φαίνεται στην Εικόνα 8.34, στο 83% του συνολικού πληθυσμού (που είναι το δείγμα μας) το μοντέλο μας προβλέπει σωστά το 60.94% του δείγματος. Ας θυμηθούμε ότι όταν τα clusters ήταν 10, το μοντέλο πρόβλεπε το 60.81% (βλέπε Εικόνα 8.23).



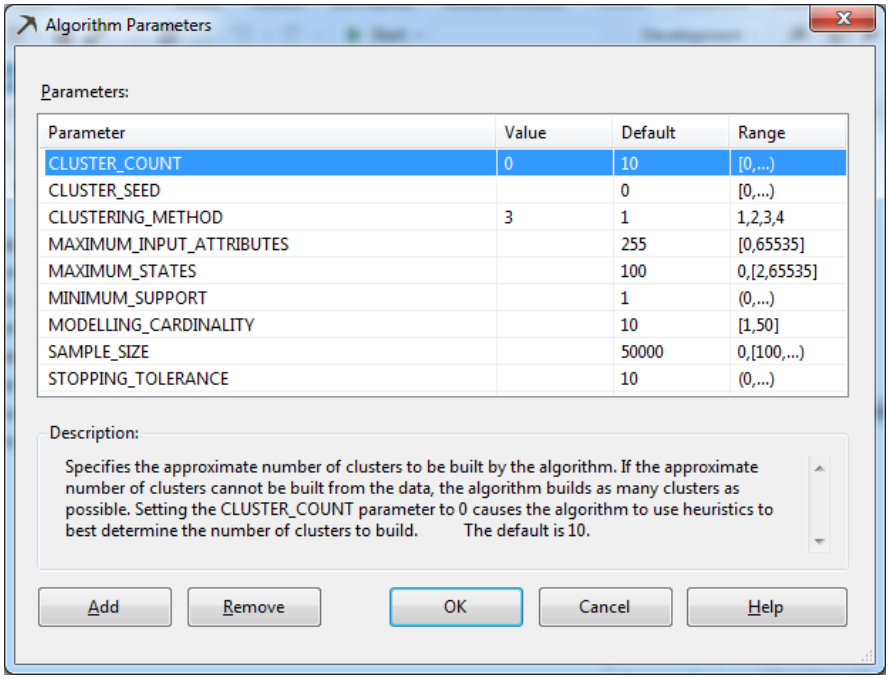
Εικόνα 8.34

Άσκηση 4

Να επαναλάβετε την άσκηση 3, επιλέγοντας τον αλγόριθμο k-means (CLUSTERING_METHOD = 3), όπου το πλήθος των clusters να υπολογίζεται ευρηστικά από τον ίδιο τον αλγόριθμο.

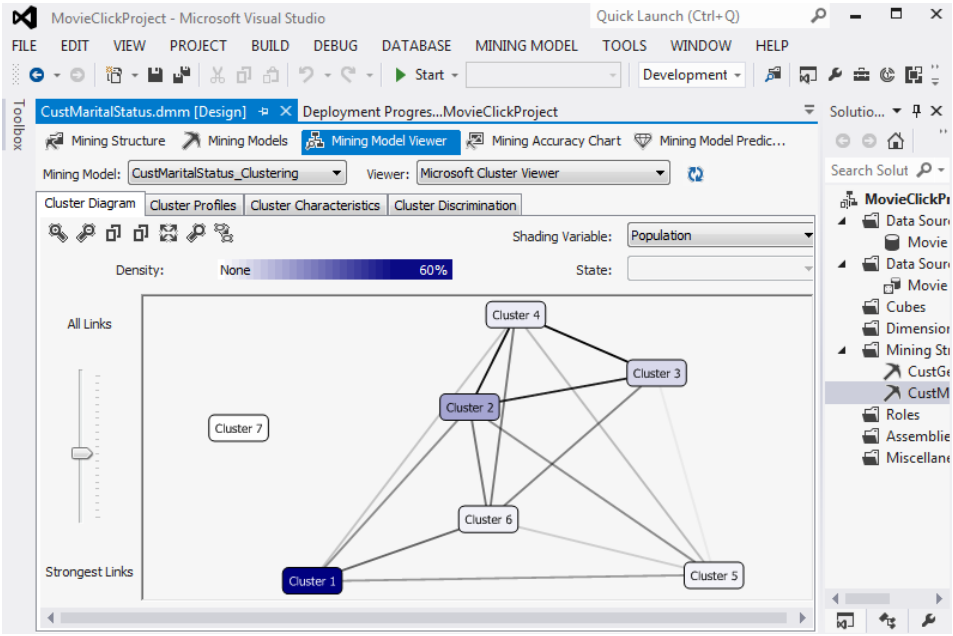
Λύση

1. Όπως φαίνεται στην Εικόνα 8.35, στην παράμετρο CLUSTERING_METHOD δίνουμε την τιμή 3, επειδή σ' αυτήν την τιμή αντιστοιχεί ο Scalable k-means. Στην παράμετρο CLUSTER_COUNT δίνουμε την τιμή 0, για να υπολογίσει ο ίδιος ο αλγόριθμος ευρηστικά το πλήθος των clusters που είναι καλύτερο να δημιουργηθούν.



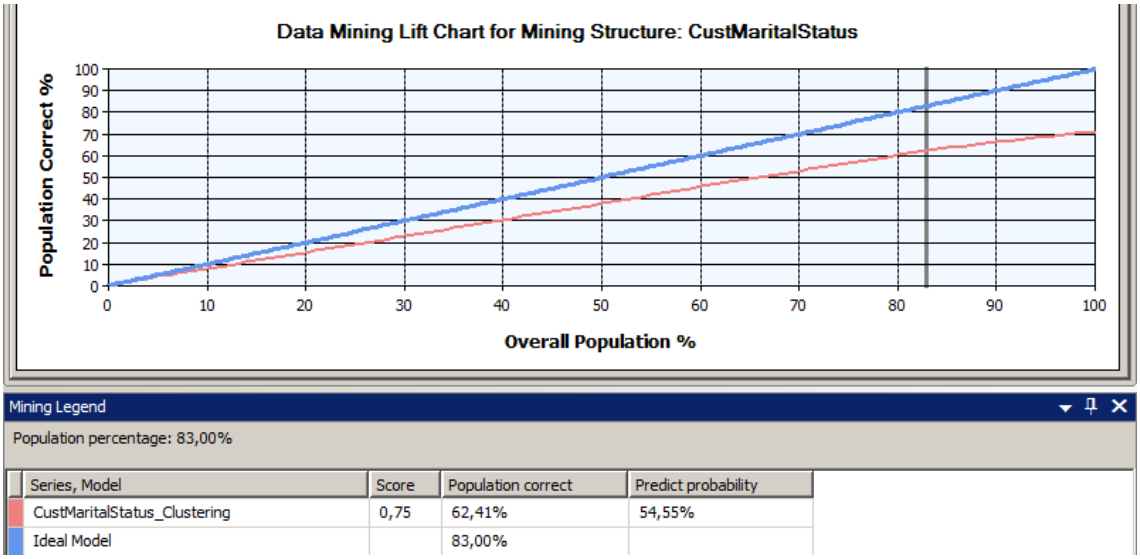
Εικόνα 8.35

2. Όπως φαίνεται στην Εικόνα 8.36, ο αλγόριθμος χώρισε τα δεδομένα σε 7 clusters.



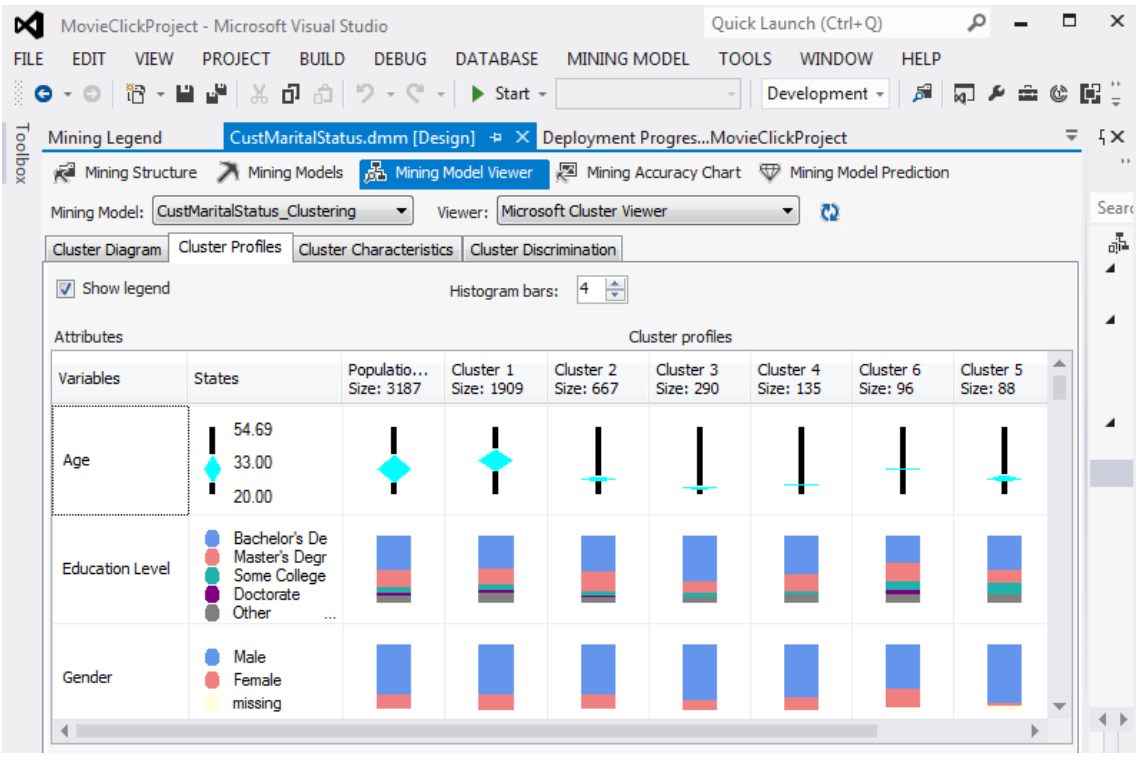
Εικόνα 8.36

2. Επιλέγουμε την καρτέλα Lift Chart, οπότε εμφανίζεται το σχετικό διάγραμμα. Κάνουμε κλικ στο 83%, πάνω στο διάγραμμα, για να δούμε τα σχετικά στατιστικά που καταγράφονται στο παράθυρο Mining Legend. Όπως φαίνεται στην Εικόνα 8.37, στο 83% του συνολικού πληθυσμού (που είναι το δείγμα μας) το μοντέλο μας προβλέπει σωστά το 62.41% του δείγματος. Ας θυμηθούμε ότι όταν τα clusters ήταν 10, το μοντέλο πρόβλεπε το 60.81% (βλέπε Εικόνα 8.23).



Εικόνα 8.37

3. Παρατηρούμε, επίσης, στο Cluster Profiles tab τις τιμές των χαρακτηριστικών σε κάθε cluster, όπως αυτές εμφανίζονται στην Εικόνα 8.38. Όπως φαίνεται, υπάρχουν και clusters (cluster 6, cluster 7) που έχουν πολύ μικρό size και θα πρέπει να ενοποιηθούν. Αυτό θα γίνει στην επόμενη άσκηση.



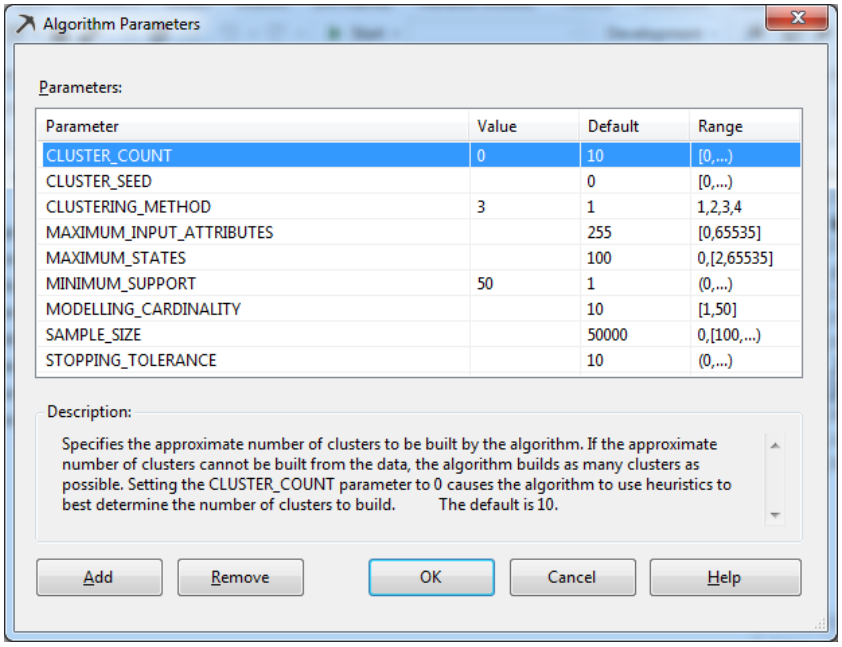
Εικόνα 8.38

Άσκηση 5

Να επαναλάβετε την άσκηση 3, επιλέγοντας τον αλγόριθμο k-means (CLUSTERING_METHOD = 3), όπου το πλήθος των clusters να υπολογίζεται ευρηστικά από τον ίδιο τον αλγόριθμο. Επιπροσθέτως, να ορίσετε ως κατώτατο πλήθος περιπτώσεων ανά cluster τις 50.

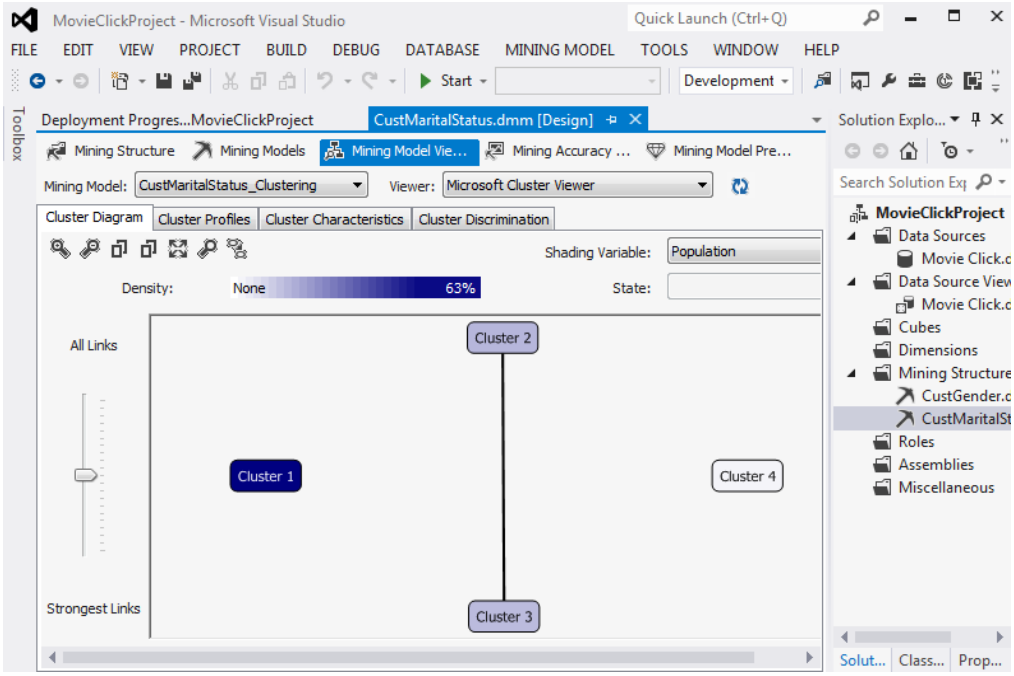
Λύση

1. Αφήνουμε τις παραμέτρους ως έχουν στην άσκηση 4, με τη διαφορά ότι στην παράμετρο MINIMUM_SUPPORT δίνουμε την τιμή 50.



Εικόνα 8.39

2. Στο Cluster Diagram, όπως φαίνεται στην Εικόνα 8.40, παρατηρούμε ότι το πλήθος των clusters έχει μειωθεί σημαντικά.



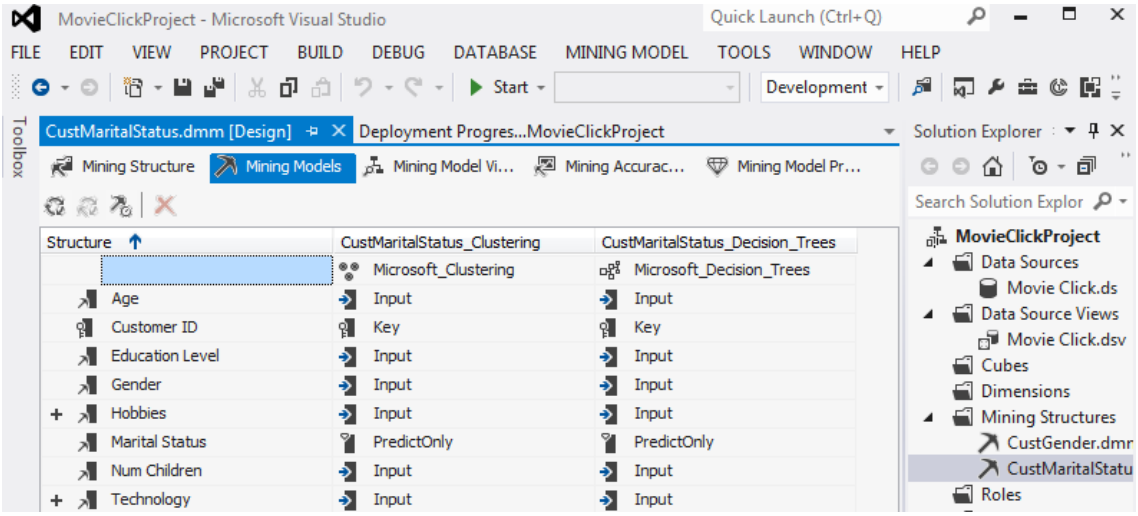
Εικόνα 8.40

Άσκηση 6

Στην καρτέλα Mining Models να δημιουργήσετε δύο διαφορετικά μοντέλα. Ένα μοντέλο να γίνει με τη χρήση του αλγορίθμου Decision Tree και ένα με τη χρήση του αλγορίθμου Clustering. Οι αλγόριθμοι να επεξεργάζονται τα ίδια ακριβώς δεδομένα μ' αυτά της Εικόνας 8.10. Στη συνέχεια, να συγκριθούν αυτά τα δύο μοντέλα.

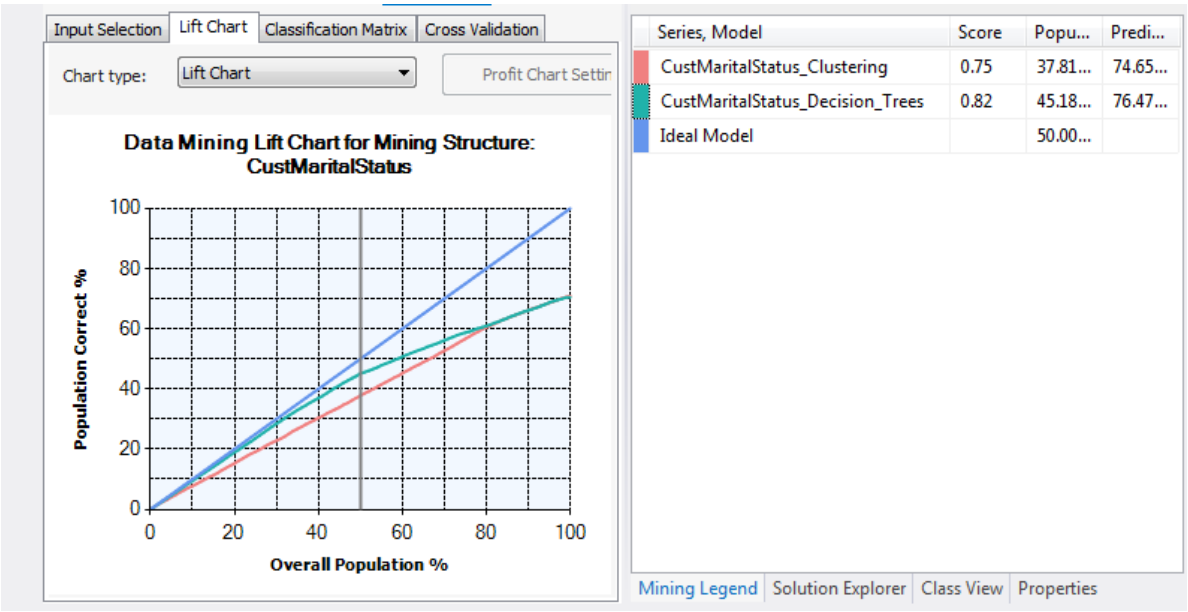
Λύση

1. Στην καρτέλα Mining Models επιλέγουμε New Mining Model. Επιλέγουμε τον αλγόριθμο Microsoft Decision Tree και δίνουμε όνομα στο μοντέλο μας (CustMaritalStatus_Decision Tree). Κάνουμε process το μοντέλο μας. Όπως φαίνεται στην Εικόνα 8.41, έχουμε πλέον φτιάξει δύο μοντέλα.



Εικόνα 8.41

2. Πηγαίνουμε στο Lift chart, όπως φαίνεται στην Εικόνα 8.43, και κάνουμε σύγκριση των δύο μοντέλων. Παρατηρούμε ότι το δεύτερο μοντέλο (πράσινη καμπύλη) προβλέπει λίγο καλύτερα από το πρώτο (κόκκινη καμπύλη).



Εικόνα 8.42

8.6. Βιβλιογραφία/Αναφορές

Chakrabarti, S. (2003). *Mining the Web: Discovering Knowledge from Hypertext Data*, Morgan – Kauffman.

Liu, B. (2007). *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*, Springer.

Rajaraman, A., Leskovec, J., & Ullman, J.D. (2015). *Mining of Massive Datasets*, Cambridge University Press.

Roiger, R., & Geatz, M. (2003). *Data Mining: A tutorial-based Primer*, Addison Wesley.

Tan, P - N., Steinbach, M., & Kumar, V. (2006). *Introduction to Data Mining*, Addison Wesley.

Κεφάλαιο 9. Εξαγωγή Κανόνων Συσχέτισης

Σύνοψη
Σ' αυτό το κεφάλαιο θα μελετήσουμε τον αλγόριθμο Assosiation Rules. Ο συγκεκριμένος αλγόριθμος παράγει συσχετίσεις μεταξύ αντικειμένων και ανήκει στην οικογένεια των Apriori αλγορίθμων. Οι ομαδοποιήσεις αντικειμένων που παράγει ονομάζονται itemsets. Με βάση τα itemssets που έχουν παραχθεί δημιουργούνται οι κανόνες συσχέτισης μεταξύ των αντικειμένων. Ένας κανόνας συσχέτισης σηματοδοτεί την εξάρτηση ενός συνόλου αντικειμένων από ένα άλλο σύνολο αντικειμένων.

9.1. Ο αλγόριθμος Assosiation Rules

Ο αλγόριθμος Assosiation Rules (Νανόπουλος, & Μανωλόπουλος, 2008· Χαλκίδη, & Βεζυργιάννης, 2005), εκτελείται σε δύο στάδια. Στο πρώτο στάδιο, γίνονται οι υπολογισμοί για να επιλεγούν τα Itemsets (σύνολα αντικειμένων) που εμφανίζονται με τη μεγαλύτερη συχνότητα. Στο δεύτερο στάδιο, δημιουργούνται οι κανόνες συσχέτισης με βάση τις συχνότητες των itemsets.

Θα μελετήσουμε ένα παράδειγμα για την καλύτερη κατανόηση της λειτουργίας του αλγορίθμου, ο οποίος εκτελείται σε δύο στάδια. Ας υποθέσουμε ότι έχουμε τις ταινίες Α, Β και τα σύνολα {Α} και {Α, Β} σ' έναν πίνακα μιας βάσης δεδομένων ενός Video Club.

1. Στο πρώτο στάδιο, γίνονται οι υπολογισμοί, για να επιλεγούν τα itemsets που εμφανίζονται με τη μεγαλύτερη συχνότητα με την εύρεση του Support που έχουν είτε μεμονωμένα αντικείμενα είτε μετέπειτα συνδυασμοί αυτών.

   - Support({A}): Το σύνολο των εγγραφών στις οποίες εμφανίζεται η ταινία Α (item {Α}) στον υπό εξέταση πίνακα της βάσης δεδομένων.

   - Support({A, B}): Το σύνολο των εγγραφών στις οποίες εμφανίζονται οι ταινίες Α και Β μαζί (itemset {Α, Β}).

   Αν **Support >= MINIMUM_SUPPORT**, τότε το itemset γίνεται αποδεκτό.

2. Στο δεύτερο στάδιο, δημιουργούνται οι κανόνες συσχέτισης με βάση τη **πιθανότητα (probability)** ή, αλλιώς, **εμπιστοσύνη (confidence)**. Έστω ότι ελέγχουμε αν η ταινία Β εξαρτάται από την ταινία Α. Στην περίπτωση που αυτός ο κανόνας ισχύει, τότε το Video Club θα μπορούσε να προτείνει την ταινία Β σ' έναν πελάτη που διαλέγει την ταινία Α. Ο τύπος του probability είναι ο ακόλουθος:

   Probability (A=>B) = Probability (B|A) = Support ({A, B})/Support ({A})

   Αν **Probability >= MINIMUM_PROBABILITY**, ο κανόνας είναι ισχυρός.

   Τονίζεται ότι ο δείκτης probability μάς βοηθά να ελέγξουμε αν ένας κανόνας είναι «ισχυρός». Όμως, ένας κανόνας μπορεί να είναι ισχυρός (να έχει υψηλό probability/confidence) και να είναι «παραπλανητικός». Δηλαδή, να υπάρχει αρνητική συσχέτιση μεταξύ των στοιχείων του κανόνα. Για τον εντοπισμό «παραπλανητικών κανόνων» υπάρχει ο δείκτης σημαντικότητας (importance) που αξιολογεί τους κανόνες συσχέτισης. Ο τύπος του importance δίνεται παρακάτω:

   Importance (A=>B) = log (Probability (B|A)/Probability (B| not A))

   Αν **importance = 0**, δεν υπάρχει καμία συσχέτιση μεταξύ Α και Β.
   Αν **importance < 0**, probability(B) μειώνεται αν το Α είναι αληθές.
   Αν **importance > 0**, probability(B) αυξάνεται αν το Α είναι αληθές.

Τέλος, ένας εναλλακτικός τρόπος που υπολογίζει τη **συσχέτιση (correlation)** μεταξύ των στοιχείων μέσα στα itemsets είναι ο παρακάτω:

Correlation({A,B})= Probability({A,B})/(Probability({A})*Probability({B}))

Αν **Correlation = 1**, οι ταινίες A και B είναι ανεξάρτητες.
Αν **Correlation < 1**, οι ταινίες έχουν αρνητική συσχέτιση.
Αν **Correlation > 1**, οι ταινίες έχουν θετική συσχέτιση.

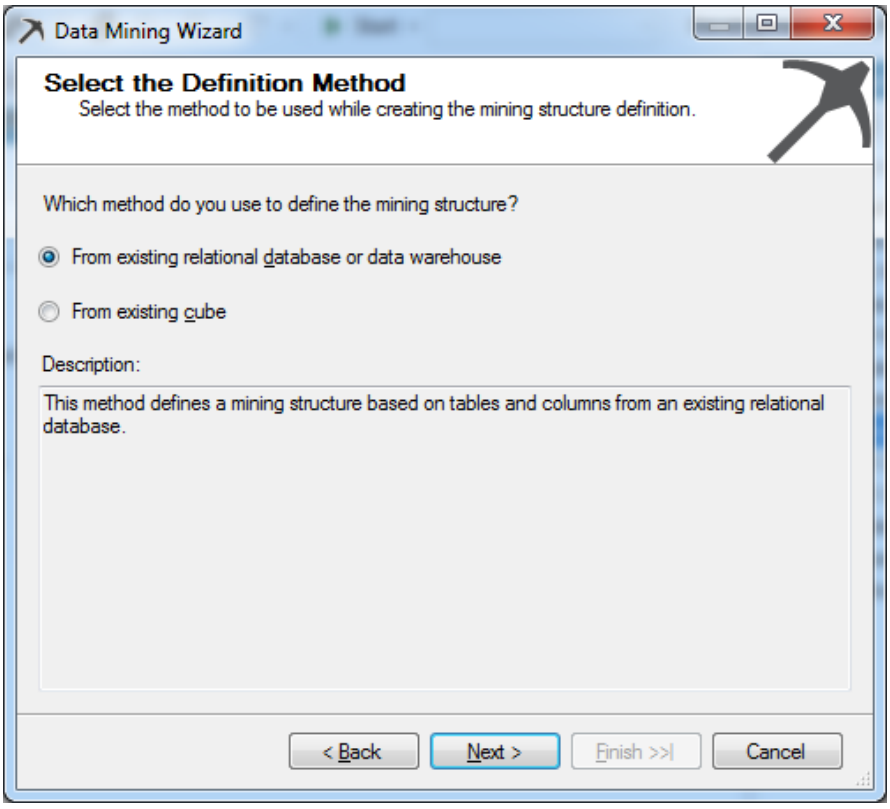
Τονίζεται ότι ο δείκτης correlation δεν χρησιμοποιείται στον SQL Server αλλά μπορεί να υπολογιστεί εξωτερικά από τον χρήστη.

9.2. Δημιουργία ενός μοντέλου Association Rules

Ας υποθέσουμε ότι ο ιδιοκτήτης ενός Video Club θέλει να συσχετίσει τις ταινίες που ενοικιάζονται από τους πελάτες του και να βρει την πιθανότητα να ενοικιάζει μια ταινία μαζί με κάποια άλλη. Μ' αυτόν τον τρόπο θα μπορεί να προτείνει στους πελάτες του ταινίες σχετικές μ' αυτήν που οι ίδιοι έχουν επιλέξει. Παρακάτω περιγράφονται αναλυτικά τα βήματα που ακολουθούμε, ώστε να δημιουργήσουμε ένα μοντέλο association rules.

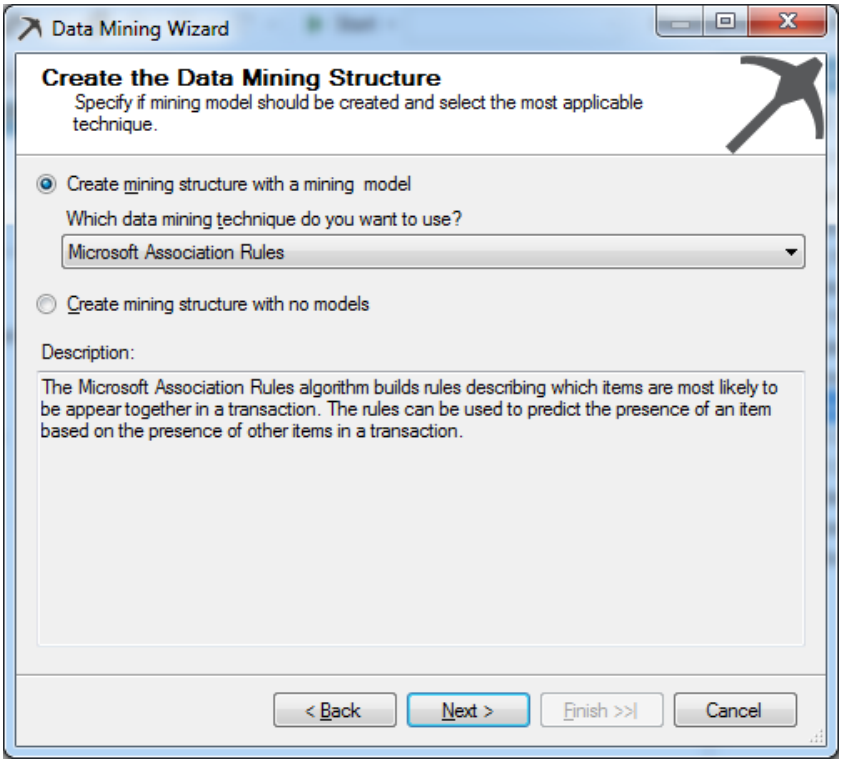
Αναλυτικά Βήματα

1. Στην καρτέλα Solution Explorer κάνουμε δεξί κλικ στο Mining Structures και, στη συνέχεια, επιλέγουμε New Mining Structure. Εμφανίζεται το παράθυρο καλωσορίσματος του οδηγού Data Mining Wizard, στο οποίο επιλέγουμε Next>, ώστε να προχωρήσουμε στο επόμενο βήμα. Στο παράθυρο που εμφανίζεται, όπως φαίνεται στην Εικόνα 9.1, επιλέγουμε From existing relational or data warehouse, καθώς θα χρησιμοποιήσουμε την βάση που ήδη έχουμε δημιουργήσει. Στη συνέχεια επιλέγουμε Next>, ώστε να προχωρήσουμε στο επόμενο βήμα.



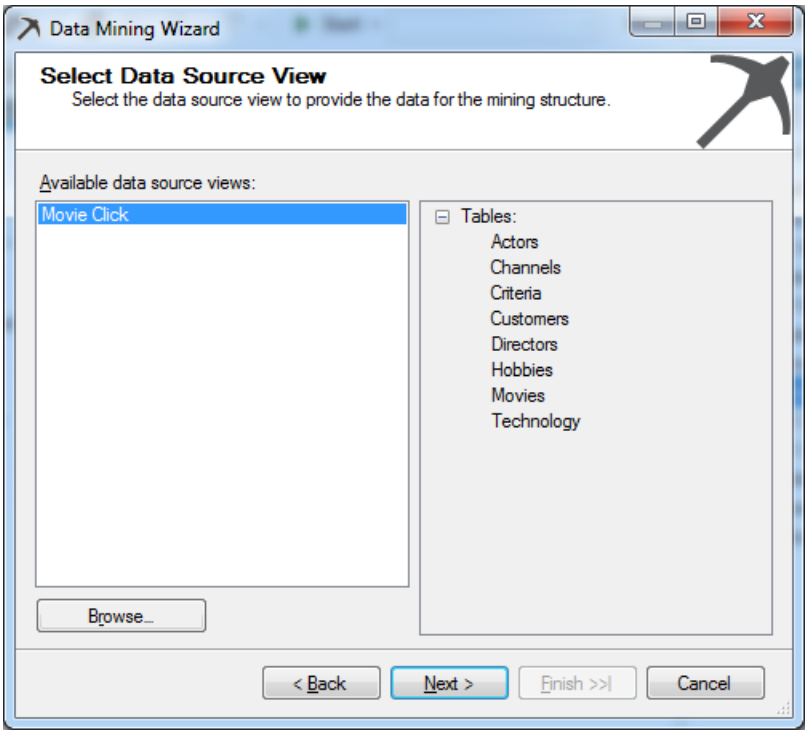
Εικόνα 9.1

2. Εμφανίζεται το παράθυρο επιλογής αλγορίθμου, όπως φαίνεται στην Εικόνα 9.2, στο οποίο επιλέγουμε Microsoft Assosiation Rules, καθώς με αυτόν τον αλγόριθμο θα ασχοληθούμε. Στη συνέχεια, επιλέγουμε Next>, ώστε να προχωρήσουμε στο επόμενο βήμα.



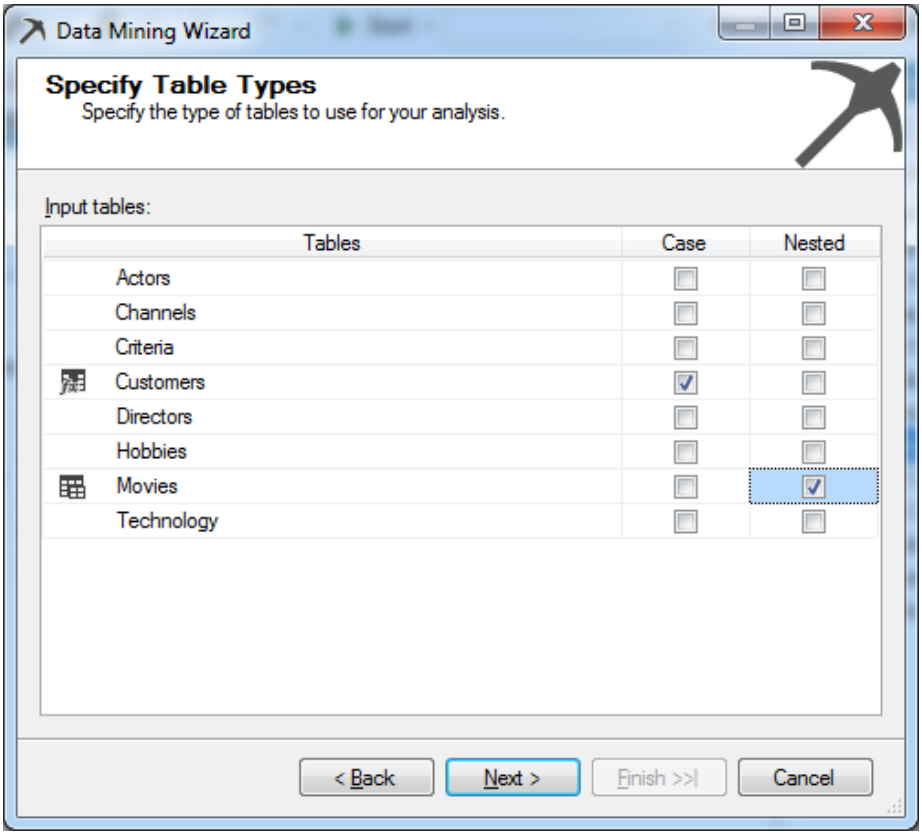
Εικόνα 9.2

3. Εμφανίζεται το παράθυρο επιλογής Data Source View, όπως φαίνεται στην Εικόνα 9.3, στο οποίο επιλέγουμε την βάση δεδομένων Movie Click. Στη συνέχεια επιλέγουμε Next>, ώστε να προχωρήσουμε στο επόμενο βήμα.



Εικόνα 9.3

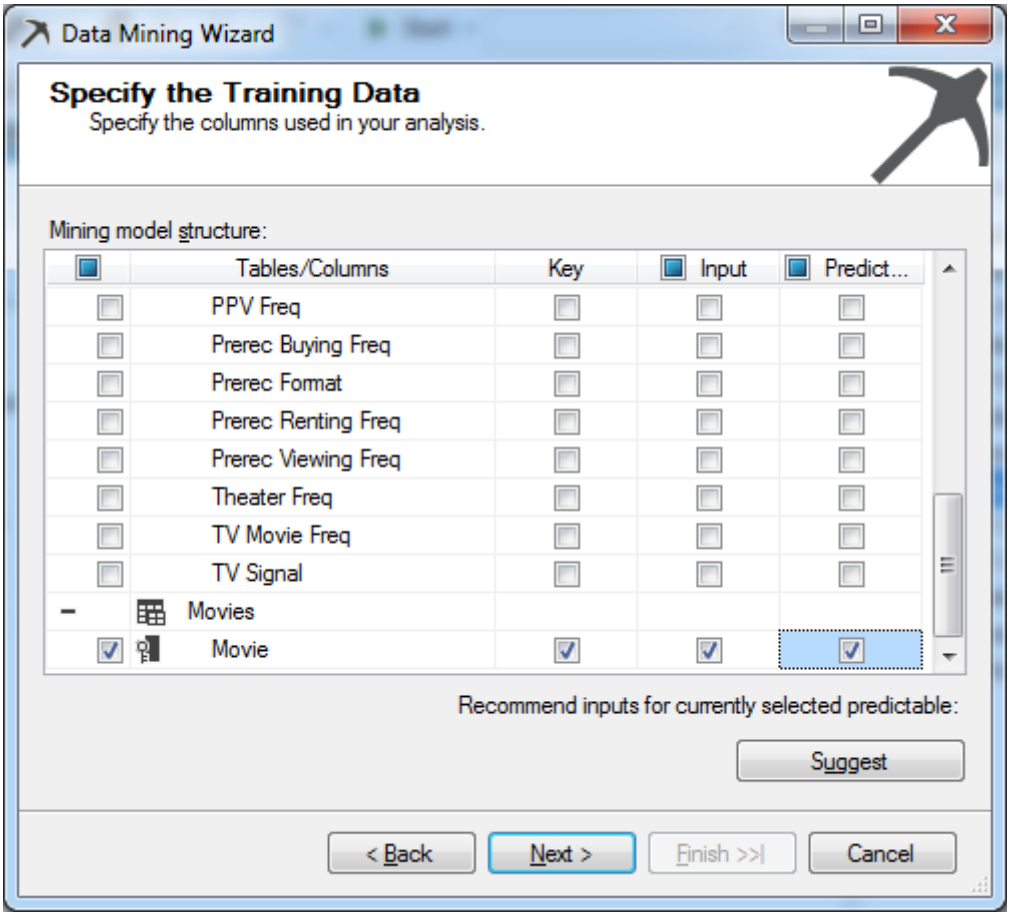
4. Σ’ αυτό το στάδιο επιλέγουμε ποιος πίνακας θα είναι ο case και ποιοι πίνακες θα είναι nested. Case είναι ο πίνακας που περιέχει τα δεδομένα που θέλουμε να προβλέψουμε. Nested είναι οι πίνακες τα δεδομένα των οποίων είναι παράμετροι στον Case (ξένα κλειδιά). Στη συγκεκριμένη περίπτωση, όπως φαίνεται στην Εικόνα 9.4, επιλέγουμε τον πίνακα Customers ως Case και τον πίνακα Movies ως Nested, καθώς θέλουμε να συσχετίσουμε τις ταινίες που έχουν επιλέξει οι πελάτες. Κατόπιν, επιλέγουμε Next>, ώστε να προχωρήσουμε στο επόμενο βήμα.



Εικόνα 9.4

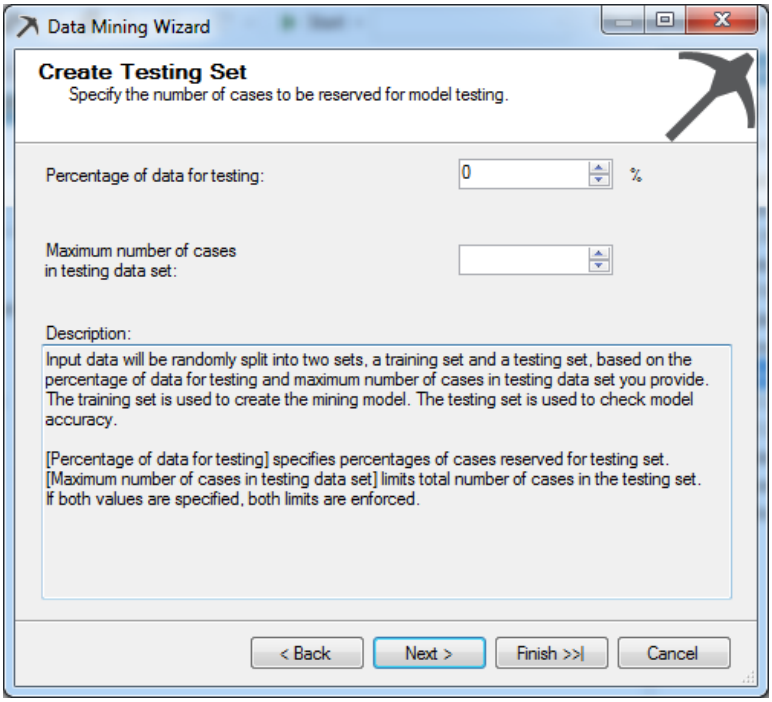
5. Σ’ αυτό το στάδιο επιλέγουμε ποια από τα δεδομένα των πινάκων που επιλέξαμε στο προηγούμενο βήμα θα είναι είσοδος στο μοντέλο και για ποια δεδομένα θέλουμε να προβλέψουμε συσχετίσεις. Συγκεκριμένα, όπως φαίνεται στην Εικόνα 9.5, κάνουμε τις εξής επιλογές:

   - Για κάθε πίνακα επιλέγουμε ένα κλειδί Key. Στη συγκεκριμένη περίπτωση, επιλέγουμε τα CustomerID και Movies.
   - Ορίζουμε ως Input τα πεδία των πινάκων που μας ενδιαφέρουν. Στη συγκεκριμένη περίπτωση, επιλέγουμε το Movies.
   - Ορίζουμε ως Predictable το πεδίο που μας ενδιαφέρει να συσχετίσουμε. Στη συγκεκριμένη περίπτωση, επιλέγουμε το Movies. Στη συνέχεια, επιλέγουμε Next>, ώστε να προχωρήσουμε στο επόμενο βήμα.



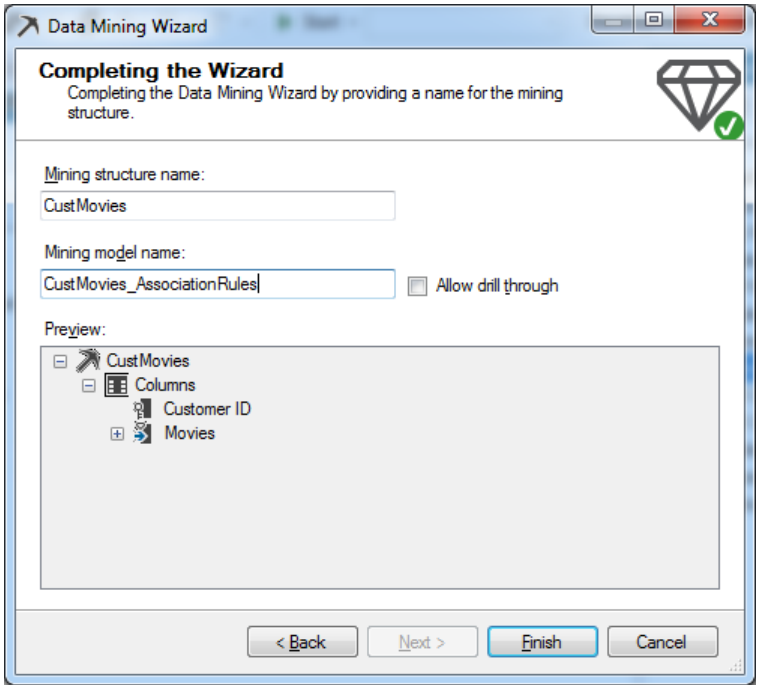
Εικόνα 9.5

6. Στο παράθυρο που εμφανίζεται, όπως φαίνεται στην Εικόνα 9.6, ορίζουμε το ποσοστό των δεδομένων που το μοντέλο θα διατηρήσει για την επαλήθευσή του. Στη συγκεκριμένη περίπτωση, επιλέγουμε την τιμή 0%. Αυτό σημαίνει ότι η αξιολόγηση του μοντέλου θα γίνει με την αισιόδοξη (optimistic) μέθοδο. Δηλαδή, θα προβλέψουμε τις εγγραφές που έχουν χρησιμοποιηθεί ως δεδομένα εκπαίδευσης. Στη συνέχεια, πατάμε Next.



Εικόνα 9.6

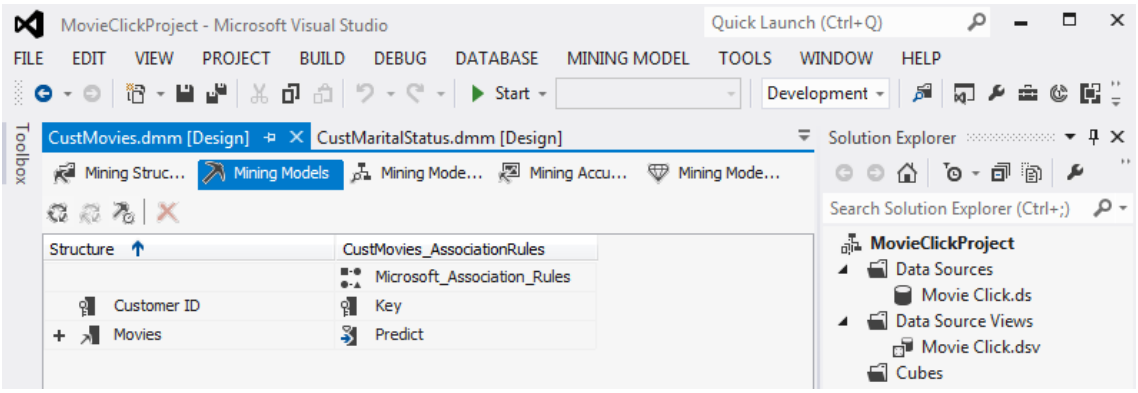
7. Στη συνέχεια, ορίζουμε όνομα για το μοντέλο μας στο πεδίο Mining structure name, όπως φαίνεται στην Εικόνα 9.7. Στη συγκεκριμένη περίπτωση συμπληρώνουμε CustMovies στο πεδίο Mining structure name και CustMovieClick_AssosiationRules στο πεδίο Mining model name. Κατόπιν, επιλέγουμε Allow drill through και, στη συνέχεια, Finish, ώστε να ολοκληρωθεί η διαδικασία.



Εικόνα 9.7

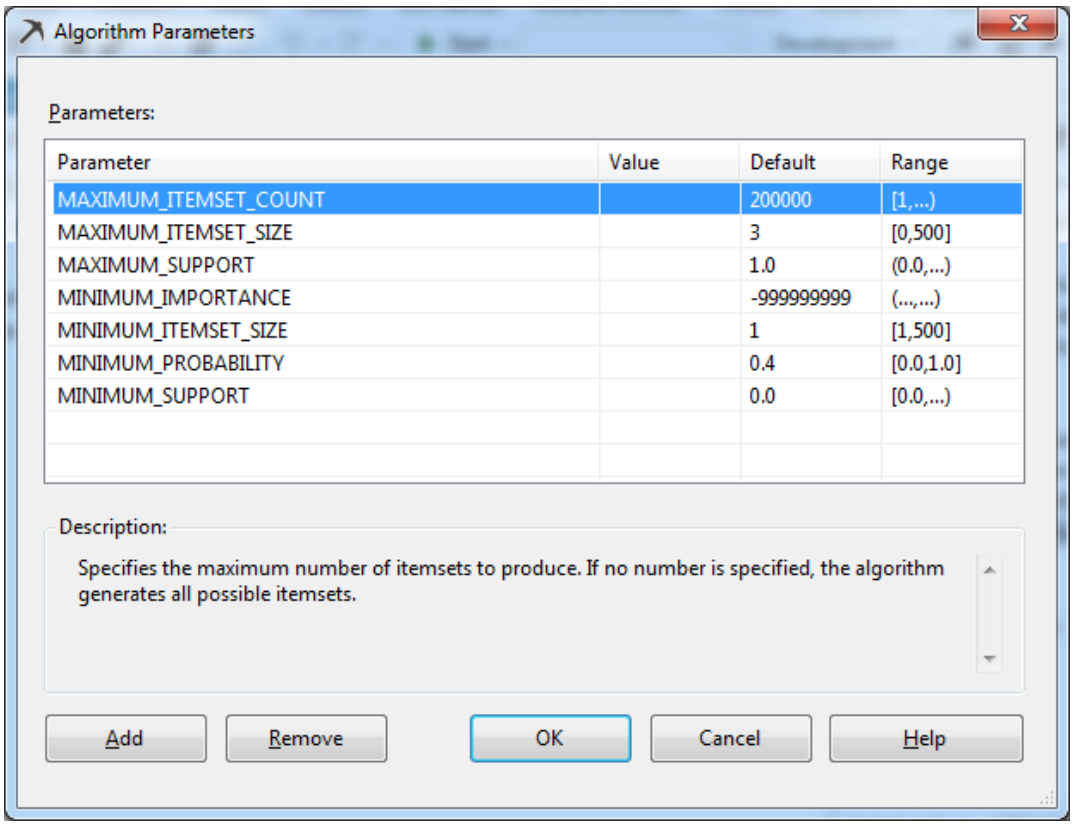
8. Στη συνέχεια, όπως φαίνεται στην Εικόνα 9.8, επιλέγουμε την καρτέλα Mining Models, ώστε να καθορίσουμε τις παραμέτρους για το μοντέλο που θα μελετήσουμε. Στη συγκεκριμένη περίπτωση θέλουμε να συσχετίσουμε τις ταινίες που επιλέγουν οι πελάτες. Έτσι, ορίζουμε τα χαρακτηριστικά ως εξής:

   • CustomerID: Key
   • Movies: Predict



Εικόνα 9.8

9. Στη συνέχεια, θα μελετήσουμε τις παραμέτρους με τις οποίες κατασκευάζεται το μοντέλο και τις προεπιλεγμένες τιμές που παίρνουν. Στην ίδια καρτέλα, όπως φαίνεται στην Εικόνα 9.8, κάνουμε δεξί κλικ στον αλγόριθμο Microsoft_Assosiation_Rules και επιλέγουμε Set Algorithm Parameters. Εμφανίζεται το παράθυρο Algorithm Parametrs, όπως φαίνεται στην Εικόνα 9.9, στο οποίο βλέπουμε επτά παραμέτρους.



Εικόνα 9.9

Ακολουθεί η αναλυτική περιγραφή της κάθε παραμέτρου του αλγορίθμου Association Rules:

- **MAXIMUM_ITEMSET_COUNT:** Αυτή η παράμετρος προσδιορίζει το μέγιστο πλήθος των Itemsets που παράγει ο αλγόριθμος. Στην περίπτωση που δεν προσδιορίσουμε τον ακριβή αριθμό, ο αλγόριθμος παράγει όλα τα δυνατά itemsets. Η προεπιλεγμένη τιμή αυτής της παραμέτρου είναι 200000.

- **MAXIMUM_ITEMSET_SIZE:** Αυτή η παράμετρος προσδιορίζει το μέγιστο πλήθος των items που επιτρέπονται σε ένα itemset. Δίνοντας στην παράμετρο την τιμή 0, ορίζουμε ότι δεν υπάρχει όριο στο πλήθος των items που επιτρέπονται στο Itemset. Η προεπιλεγμένη τιμή αυτής της παραμέτρου είναι 3.

- **MAXIMUM_SUPPORT:** Αυτή η παράμετρος προσδιορίζει τον μέγιστο αριθμό των εμφανίσεων ενός itemset που θα εξετάζεται από τον αλγόριθμο. Αν η τιμή της παραμέτρου είναι μικρότερη του 1, τότε αναπαριστά ένα ποσοστό επί των συνολικών περιπτώσεων. Τιμές μεγαλύτερες του ενός αναπαριστούν τον απόλυτο αριθμό των εμφανίσεων ενός itemset.

- **MINIMUM_IMPORTANCE:** Αυτή η παράμετρος προσδιορίζει το κατώτατο όριο που πρέπει να έχει ένας κανόνας συσχέτισης για να χαρακτηριστεί σημαντικός (βλέπε ενότητα 9.1). Οι κανόνες οι οποίοι έχουν τιμή μικρότερη απ' την τιμή που ορίζεται φιλτράρονται και απορρίπτονται.

- **MINIMUM_ITEMSET_SIZE:** Αυτή η παράμετρος προσδιορίζει τον ελάχιστο αριθμό στοιχείων από τα οποία θα πρέπει να αποτελείται ένα Itemset και λαμβάνει τιμές [1,500]. Η προεπιλεγμένη τιμή γι' αυτήν την παράμετρο είναι 1.

- **MINIMUM_PROPABILITY:** Αυτή η παράμετρος προσδιορίζει το ελάχιστο όριο της εμπιστοσύνης (confidence) για να γίνει ένας κανόνας αποδεκτός (βλέπε Ενότητα 9.1). Η προεπιλεγμένη τιμή αυτής της παραμέτρου είναι 0.4.

- **MINIMUM_SUPPORT:** Αυτή η παράμετρος προσδιορίζει τον ελάχιστο αριθμό των εμφανίσεων που πρέπει να έχει ένα itemset για να επιλεχθεί. Αν η τιμή της παραμέτρου είναι μικρότερη του 1, τότε αναπαριστά ένα ποσοστό επί των συνολικών περιπτώσεων. Τιμές μεγαλύτερες από 1 αναπαριστούν τον απόλυτο αριθμό των εμφανίσεων που πρέπει ένα itemset να έχει για να γίνει αποδεκτό. Η προεπιλεγμένη τιμής αυτής της παραμέτρου είναι 0,0.

Για να κατανοήσουμε καλύτερα τη λειτουργία των παραπάνω παραμέτρων, θα περιγράψουμε συνοπτικά τον τρόπο με τον οποίο εφαρμόζεται ο αλγόριθμος Apriori. Όπως αναφέρθηκε και στην Ενότητα 9.1, ο Apriori αλγόριθμος εκτελείται σε δύο στάδια:

- Στο πρώτο στάδιο, γίνονται οι υπολογισμοί για να επιλεγούν τα itemsets που εμφανίζονται με τη μεγαλύτερη συχνότητα. Ο αλγόριθμος, λοιπόν, υπολογίζει το support όλων των items, του καθενός ξεχωριστά. Τα items που έχουν support μεγαλύτερο ή ίσο από το ελάχιστο όριο υποστήριξης (**MINIMUM_SUPPORT**) γίνονται δεκτά και συγκροτούν το σύνολο L1. Στη συνέχεια, παράγονται όλα τα δυνατά ζευγάρια των στοιχείων του συνόλου L1, δηλαδή συγκροτούνται δυάδες από items. Για κάθε ζευγάρι υπολογίζεται ξανά το support και όσα από τα ζευγάρια γίνονται δεκτά συγκροτούν το σύνολο L2. Κατόπιν, παράγονται όλα τα δυνατά ζευγάρια των στοιχείων του συνόλου L2, δηλαδή τριάδες από items. Τελικά, ο αλγόριθμος συνεχίζει να παράγει n-άδες από items, έως ότου η τιμή του n να γίνει ίση με την τιμή της παραμέτρου **MAXIMUM_ITEMSET_SIZE.**
- Στο δεύτερο στάδιο, ο Apriori δημιουργεί τους κανόνες συσχέτισης. Από το τελευταίο σύνολο L που προκύπτει, ελέγχεται το confidence όλων των δυνατών κανόνων συσχέτισης που μπορεί να προκύψουν. Οι κανόνες που έχουν εμπιστοσύνη μεγαλύτερη από την ελάχιστη εμπιστοσύνη που έχει προσδιοριστεί (**MINIMUM_PROPABILITY**) γίνονται τελικά αποδεκτοί. Τέλος, οι κανόνες αυτοί ελέγχονται και ως προς τη σημαντικότητα τους, ένα στάδιο στο οποίο πρέπει να περάσουν το κατώφλι που έχει οριστεί από την παράμετρο (**MINIMUM_IMPORTANCE).**

9.3. Αξιολόγηση των Itemsets και των Association Rules

Σ' αυτήν την ενότητα θα αξιολογήσουμε την ποιότητα τόσο των Itemsets όσο και των κανόνων συσχέτισης.

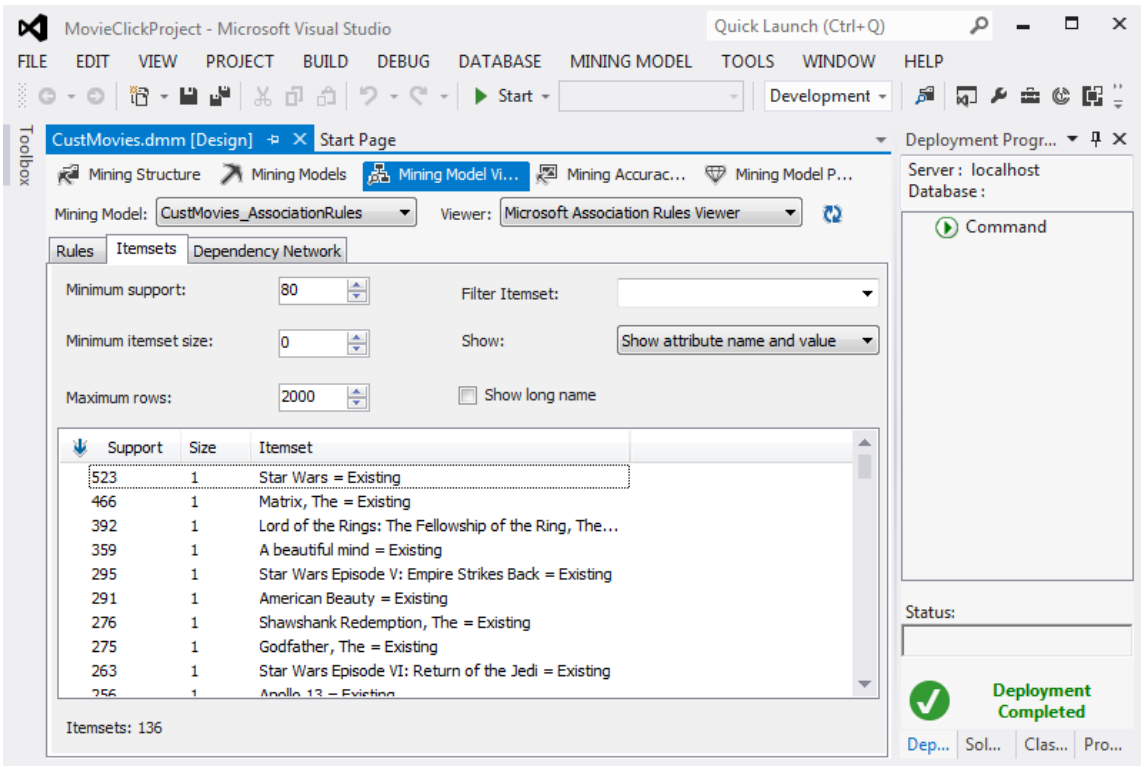
9.3.1. Αξιολόγηση των Itemsets

Για να αξιολογήσουμε τα Itemsets, επιλέγουμε την καρτέλα Mining Model Viewer και, κατόπιν, την καρτέλα Itemsets, ώστε να εμφανιστούν τα itemsets με τη μεγαλύτερη συχνότητα που έχουν δημιουργηθεί από τον αλγόριθμο association rules. Όπως φαίνεται στην Εικόνα 9.10, εμφανίζεται ένας πίνακας με τρεις στήλες.

1. Η στήλη **Support** εμφανίζει τη συχνότητα του κάθε itemset. Η ελάχιστη τιμή που παρατηρούμε σ' αυτήν την στήλη καθορίζεται από την τιμή της παραμέτρου **MINIMUM_SUPPORT**. Αν η τιμή της παραμέτρου είναι πολύ μικρή, είναι πιθανό να εμφανιστούν πολλά itemsets.
2. Η στήλη **Size** απεικονίζει το πλήθος των αντικειμένων που συγκροτούν το itemset. Η μέγιστη τιμή εξαρτάται από την τιμή της παραμέτρου **MAXIMUM_ITEMSET_SIZE**. Η προεπιλεγμένη τιμή της παραμέτρου είναι 3 και, επομένως, το μέγιστο πλήθος των ταινιών που περιέχει ένα itemset δεν ξεπερνάει τις 3.
3. Η στήλη **Itemset** περιέχει τα αντικείμενα (items) από τα οποία αποτελείται το κάθε itemset.

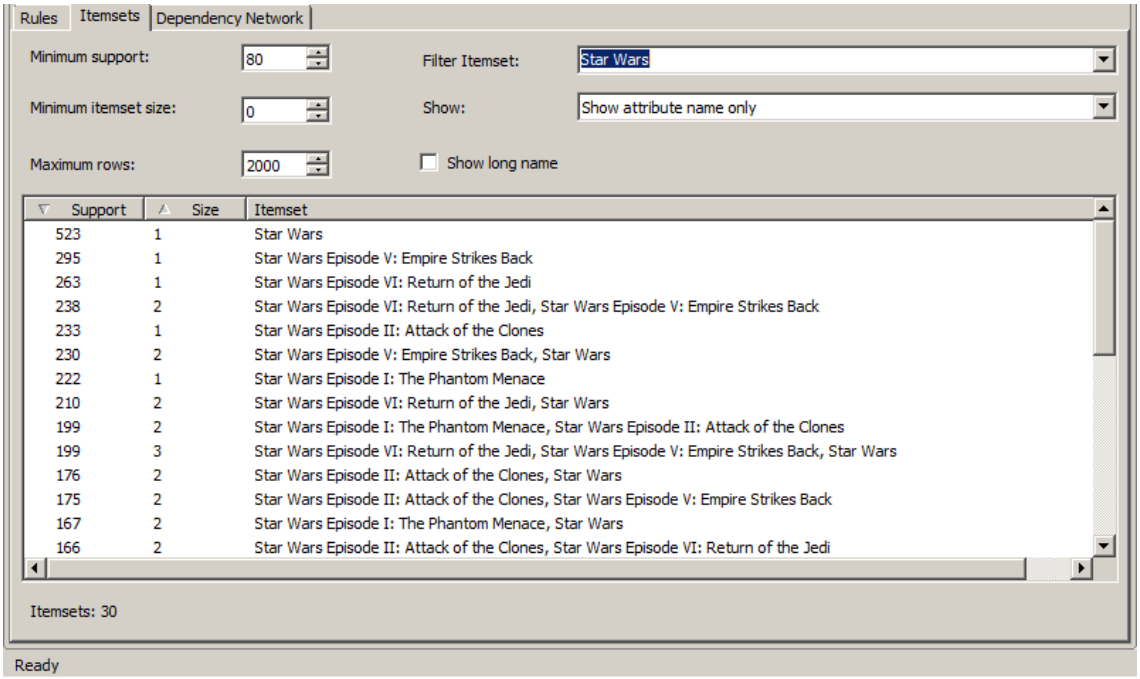
Επιπλέον, παρατηρούμε, όπως φαίνεται στην ίδια Εικόνα, ότι υπάρχουν διάφορα πεδία με τα οποία μπορούμε να παραμετροποιήσουμε τα αποτελέσματα. Για παράδειγμα, στη συγκεκριμένη περίπτωση:

- Στο πεδίο **Minimum support** καθορίζεται ο ελάχιστος αριθμός εμφανίσεων που απαιτούμε να έχει το κάθε itemset. Η τιμή του εδώ είναι 80.
- Στο πεδίο **Minimum itemset size** καθορίζεται το ελάχιστο πλήθος των items που συγκροτούν ένα itemset.
- Στο πεδίο **Maximum rows** καθορίζουμε το πλήθος των Itemsets. Η τιμή που δίνουμε εδώ ισούται με 2000 και περιορίζει την τιμή της παραμέτρου **MAXIMUM_ITEMSET_COUNT**.
- Στο πεδίο **Filter Itemset** καθορίζουμε το item με βάση το οποίο θέλουμε να φιλτράρουμε τα itemsets.
- Στο πεδίο Show επιλέγουμε τον τρόπο με τον οποίο θα εμφανίζονται τα itemsets (π.χ. show attribute name and value).



Εικόνα 9.10

Στη συνέχεια, ας υποθέσουμε ότι θέλουμε να βρούμε τα itemsets που εμπεριέχουν την ταινία Star Wars και, επίσης, να βλέπουμε μόνο τα names των attributes. Όπως φαίνεται στην Εικόνα 9.11, συμπληρώνουμε στο πεδίο Filter Itemset το όνομα της ταινίας, δηλαδή Star Wars. Επίσης, επιλέγουμε στο drop box Show την επιλογή Show attribute name only. Όπως φαίνεται στην ίδια Εικόνα, βλέπουμε πλέον αποκλειστικά itemsets που εμπεριέχουν την ταινία Star Wars.



Εικόνα 9.11

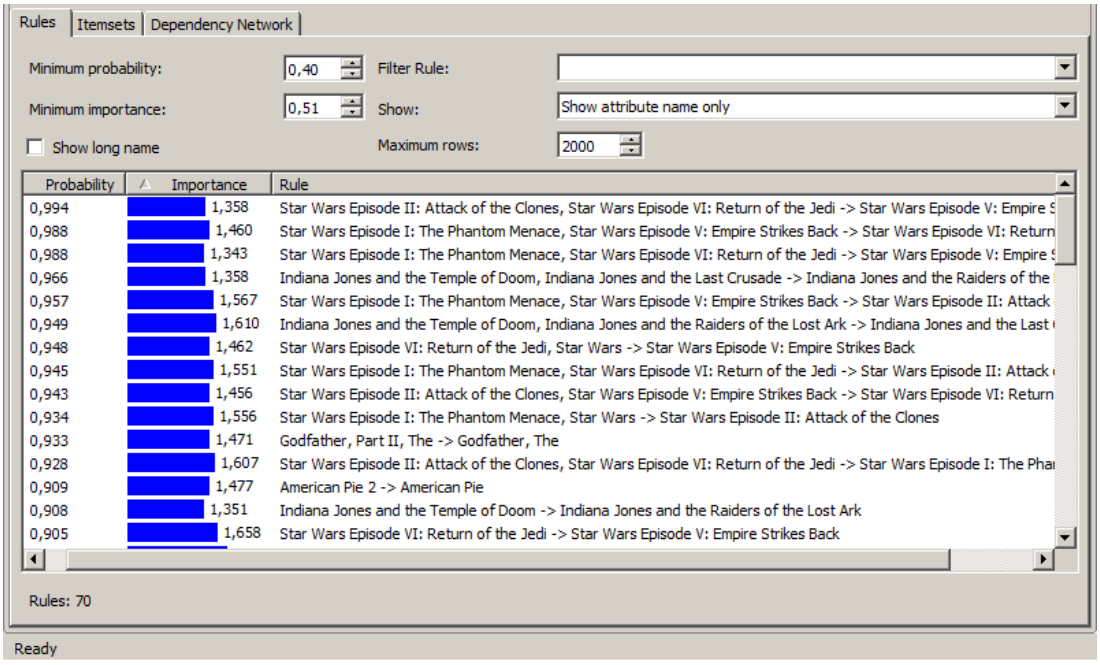
9.3.2. Αξιολόγηση των κανόνων συσχέτισης

Για να αξιολογήσουμε τους κανόνες συσχέτισης, επιλέγουμε την καρτέλα Mining Model Viewer και, κατόπιν, την καρτέλα Rules, ώστε να εμφανιστούν οι κανόνες συσχέτισης με τη μεγαλύτερη πιθανότητα/εμπιστοσύνη (probability/confidence) και βαθμό σημαντικότητας (importance). Όπως φαίνεται στην Εικόνα 9.12, εμφανίζεται ένας πίνακας με τρεις στήλες.

1. Στην στήλη **Probability** εμφανίζεται ο βαθμός εμπιστοσύνης/πιθανότητας του κανόνα. Όλα τα ποσοστά που βλέπουμε είναι πάνω από 0.4, καθώς η προεπιλεγμένη τιμή της παραμέτρου MINIMUM_PROBABILITY είναι 0.4.
2. Στην στήλη **Importance** εμφανίζεται ο βαθμός σημαντικότητας του κανόνα συσχέτισης (βλέπε Ενότητα 9.1).
3. Στην στήλη **Rule** απεικονίζονται οι κανόνες που τελικά παρήγαγε ο αλγόριθμος. Κάθε κανόνας αποτελείται από το αριστερό και το δεξιό μέρος και έχει την παρακάτω μορφή:

«Σύνολο μεταβλητών που εξαρτούν τις μεταβλητές του δεξιού μέλους → Σύνολο μεταβλητών που εξαρτώνται από τις μεταβλητές του αριστερού μέλους»

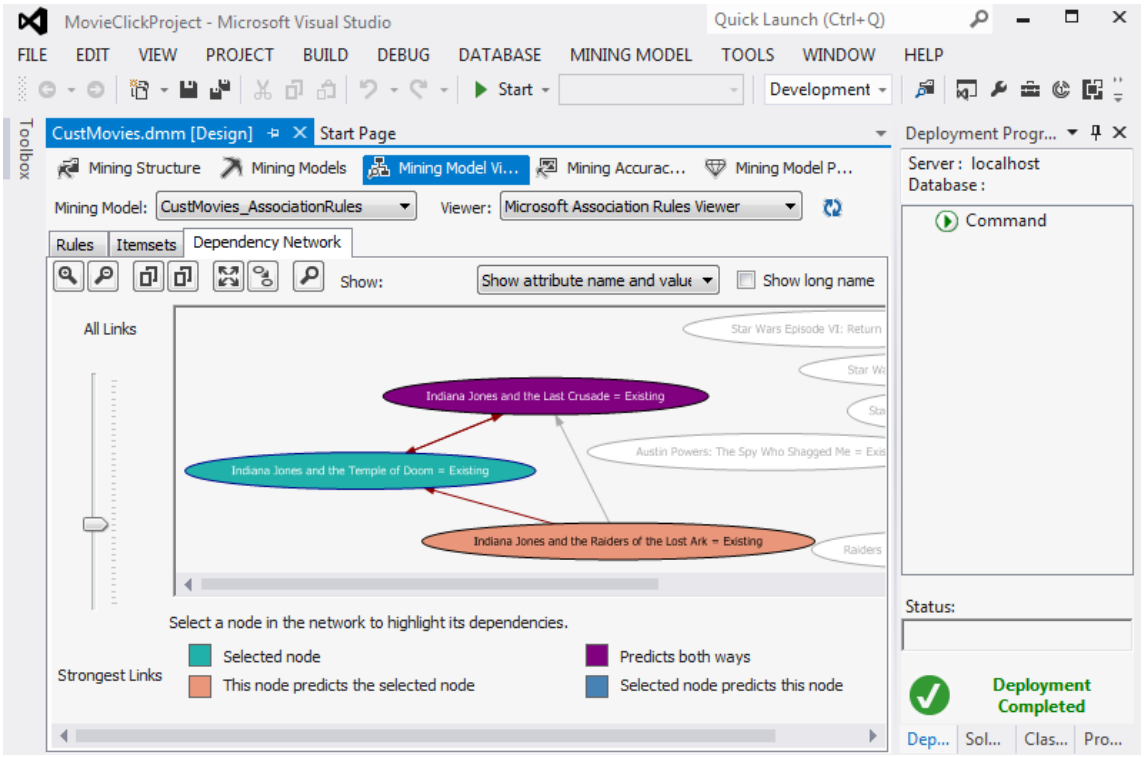
Έτσι, η ταινία που βρίσκεται στο δεξιό μέρος του κανόνα εξαρτάται από την ταινία ή τις ταινίες που βρίσκονται στο αριστερό μέρος του. Για παράδειγμα, ένας πελάτης που επέλεξε την ταινία «The Godfather: Part 2» συνήθως επιλέγει και την ταινία «The Godfather», όπως φαίνεται στην πέμπτη σειρά από το τέλος του πίνακα της Εικόνας 9.12.



Εικόνα 9.12

Στη συνέχεια, για να δούμε καλύτερα τις συσχετίσεις που έχει βρει ο αλγόριθμος, ανοίγουμε την καρτέλα Dependency Network του Mining Model Viewer, όπως φαίνεται στην Εικόνα 9.13. Παρατηρούμε ότι:

- Κάθε κόμβος αντιπροσωπεύει ένα ξεχωριστό item.
- Δύο ή περισσότερα items που συνδέονται με μια γραμμή σχηματίζουν ένα itemset.
- Το βέλος μεταξύ δύο items συμβολίζει την ύπαρξη κανόνα.
- Η γραμμή κύλισης στο αριστερό μέρος του παραθύρου ελέγχει τον δείκτη importance score.
- Εάν κάνουμε κλικ σ' έναν κόμβο, βλέπουμε τους υπόλοιπους κόμβους που σχετίζονται με αυτόν. Για παράδειγμα, κάνοντας κλικ σ' ένα node (π.χ. Indiana Jones and the Temple of Doom), βλέπουμε όλους τους nodes που σχετίζονται με αυτόν. Με διαφορετικά χρώματα βλέπουμε: αυτόν που επιλέξαμε, τους nodes που εξαρτώνται απ' αυτόν και τους nodes απ' τους οποίους αυτός εξαρτάται.



Εικόνα 9.13

9.4 Ασκήσεις αξιολόγησης Κανόνων Συσχέτισης

1. Να αλλάξετε την παράμετρο **(MAXIMUM_ITEMSET_SIZE)** του αλγορίθμου Association Rules, ώστε τα itemsets να εμπεριέχουν το πολύ μέχρι 5 αντικείμενα (items). Να αξιολογήσετε το νέο μοντέλο (αξιολογώντας τις καρτέλες itemsets και rules).

2. Να αλλάξετε την παράμετρο **(MINIMUM_SUPPORT)** του αλγορίθμου Association Rules, θέτοντας σ' αυτήν την τιμή 0.1. Να αξιολογήσετε το μοντέλο (αξιολογώντας τις καρτέλες itemsets και rules).

3. Να αλλάξετε τις παραμέτρους **(MINIMUM_SUPPORT, MINIMUM_PROBABILITY, MAXIMUM_ITEM_SIZE)** του αλγορίθμου Association Rules, θέτοντας σ' αυτές τις τιμές 0.05, 0.6 και 4, αντίστοιχα. Κατόπιν, να αξιολογήσετε το νέο μοντέλο (αξιολογώντας τις καρτέλες itemsets και rules).

4. Να αλλάξετε τις παραμέτρους **(MINIMUM_SUPPORT, MINIMUM_PROBABILITY, MAXIMUM_ITEM_SIZE, MINIMUM IMPORTANCE)** του αλγορίθμου Association Rules, θέτοντας σ' αυτές τις τιμές 0.05, 0.6, 4 και 1.5 αντίστοιχα. Κατόπιν, να βρείτε τους δέκα κανόνες με τις μεγαλύτερες τιμές στο δείκτη probabilty.

5. Να δημιουργήσετε ένα νέο μοντέλο που να συσχετίζει τις ταινίες με την ηλικία των πελατών. Αυτά τα δύο δεδομένα (ταινίες και ηλικία πελατών) να χρησιμοποιηθούν ως είσοδοι (input variables) στο νέο μοντέλο, αλλά να ζητείται και η πρόβλεψή τους (predicable variables).

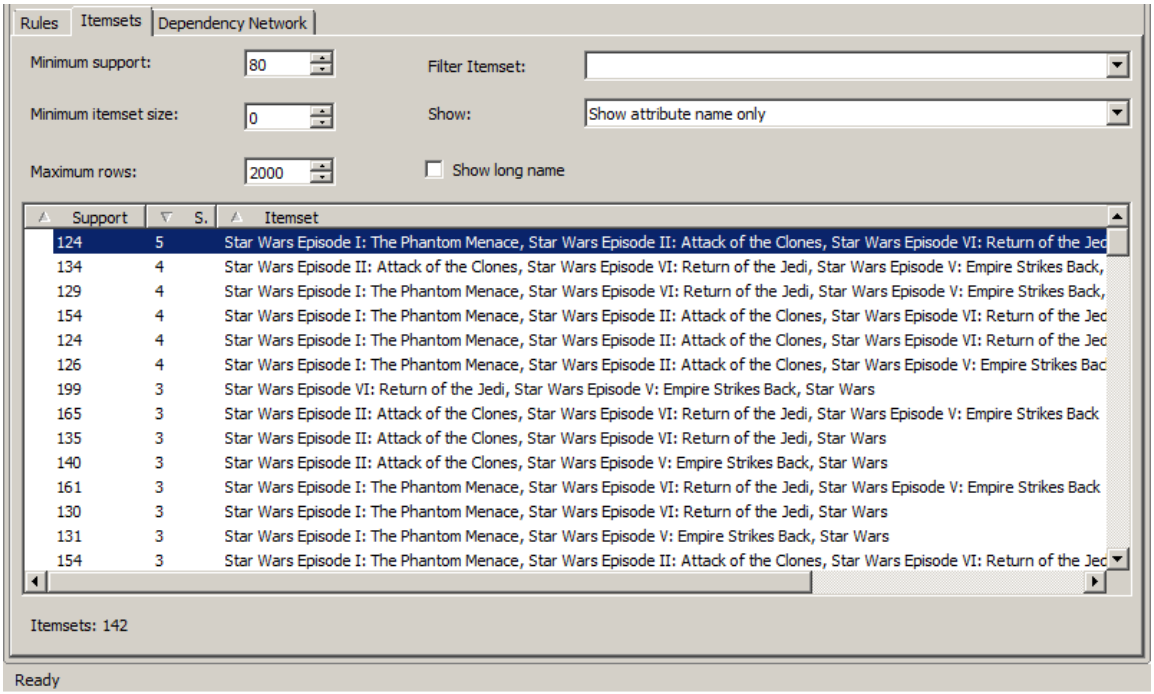
9.5. Λύσεις ασκήσεων αξιολόγησης Κανόνων Συσχέτισης

Άσκηση 1

Να αλλάξετε την παράμετρο **(MAXIMUM_ITEMSET_SIZE)** του αλγορίθμου Association Rules, έτσι ώστε τα itemsets να εμπεριέχουν το πολύ μέχρι 5 αντικείμενα. Να αξιολογήσετε το νέο μοντέλο (αξιολογώντας τις καρτέλες itemsets και rules).

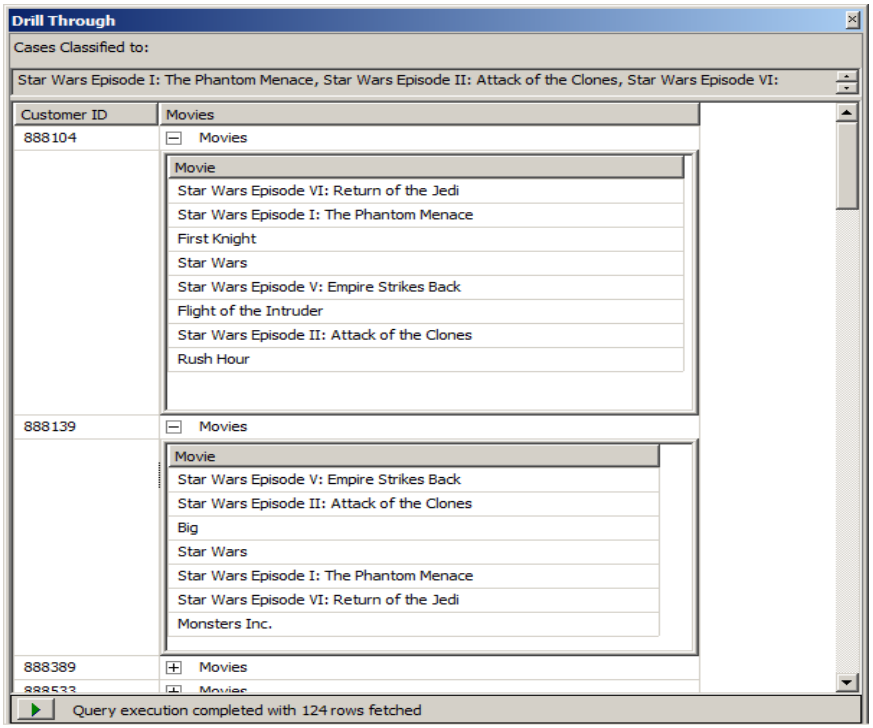
Λύση

1. Αλλάζουμε την τιμή της παραμέτρου του αλγορίθμου MAXIMUM_ITEMSET_SIZE, δίνοντας την τιμή πέντε. Όπως φαίνεται στην Εικόνα 9.14, έχει παραχθεί ένα itemset με μέγεθος 5 που αφορά τη σειρά ταινιών επιστημονικής φαντασίας Star Wars. Το συγκεκριμένο itemset εμφανίζεται 124 φορές στη βάση δεδομένων video club. Αυτό σημαίνει ότι 124 διαφορετικοί πελάτες έχουν ενοικιάσει στο παρελθόν τις πέντε ταινίες της σειράς Star Wars.



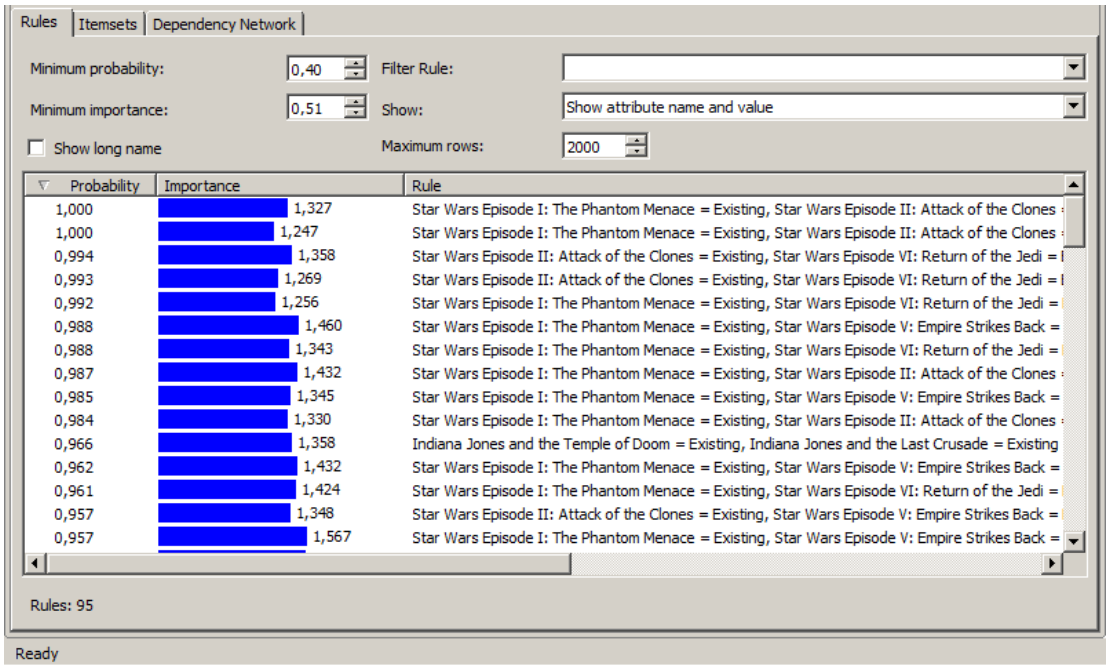
Εικόνα 9.14

2. Κάνοντας Drill Through πάνω στο συγκεκριμένο itemset, βλέπουμε το προφίλ των πελατών που τις έχουν νοικιάσει. Συγκεκριμένα, όπως φαίνεται στην Εικόνα 9.15, έχουμε δύο στήλες (Customer_ID, Movies). Παρατηρήστε ότι εμφανίζονται και άλλες ταινίες εκτός απ' αυτές που εμπεριέχονται στο itemset.



Εικόνα 9.15

3. Τέλος, όσον αφορά τους κανόνες συσχέτισης, ο αλγόριθμος βρήκε «ισχυρούς κανόνες» με Probability κοντά στην μονάδα. Θα πρέπει, όμως, να συνεκτιμήσουμε, σύμφωνα μ' αυτά που περιγράφηκαν στην Ενότητα 9.1 αν οι κανόνες είναι «παραπλανητικοί», λαμβάνοντας υπόψη την τιμή της στήλης importance, όπως φαίνεται στην Εικόνα 9.16,



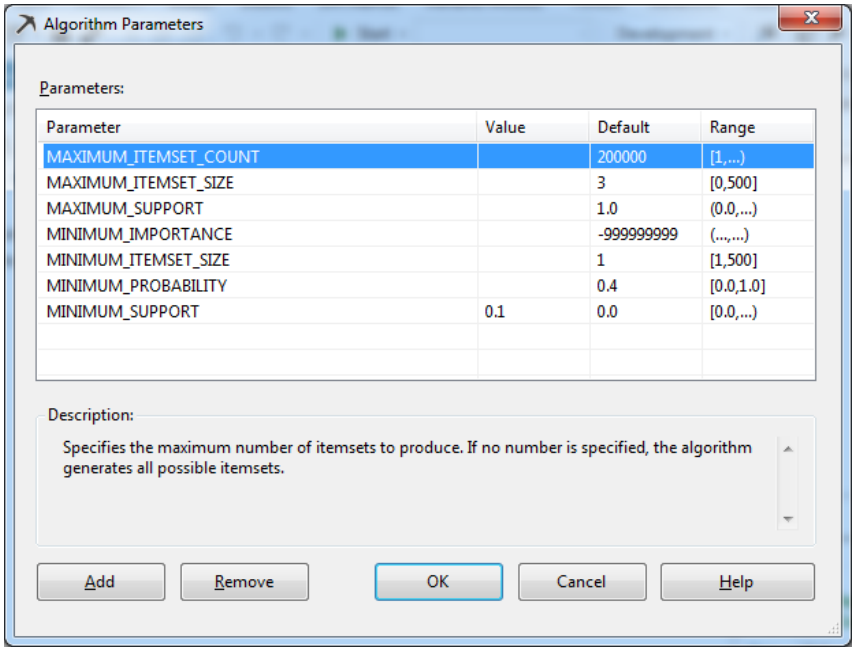
Εικόνα 9.15

Άσκηση 2

Να αλλάξετε την παράμετρο **(MINIMUM_SUPPORT)** του αλγορίθμου Association Rules, θέτοντας σ' αυτήν την τιμή 0.1. Να αξιολογήσετε το μοντέλο (αξιολογώντας τις καρτέλες itemsets και rules).

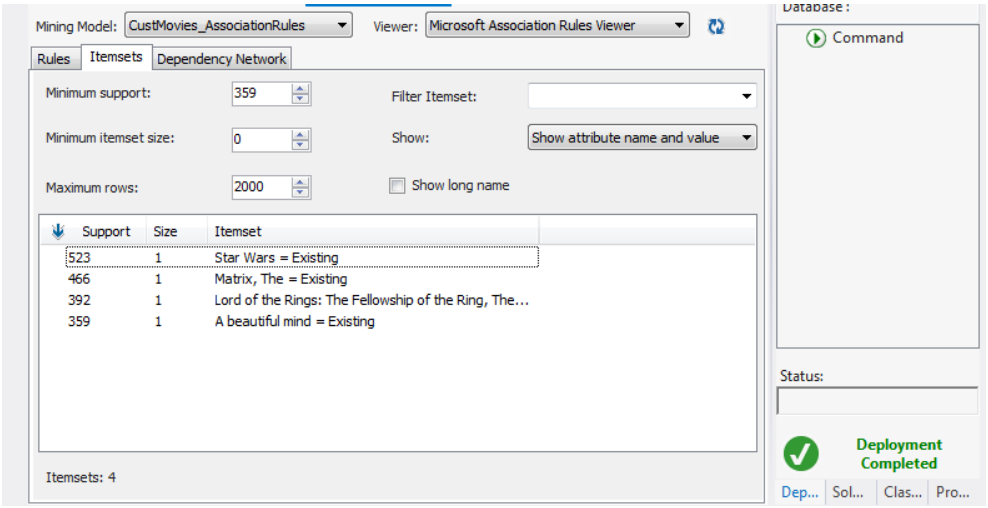
Λύση

1. Στην παράμετρο MINIMUM_SUPPORT δίνουμε την τιμή 0.1, όπως φαίνεται στην Εικόνα 9.17. Η παράμετρος αυτή προσδιορίζει το ελάχιστο όριο της υποστήριξης (support) που ένα itemset χρειάζεται για να γίνει αποδεκτό στην πρώτη φάση του αλγορίθμου Association Rules. Η τιμή 0.1 σημαίνει πρακτικά ότι ένα itemset, για να γίνει αποδεκτό, πρέπει να υπάρχει στο 10% των συνολικών εγγραφών που εξετάζει ο αλγόριθμος. Επομένως, για να γίνει δεκτό ένα itemset, στο Video Club θα πρέπει να περιέχονται όλες οι ταινίες του itemset στο 10% των προτιμήσεων όλων των πελατών.



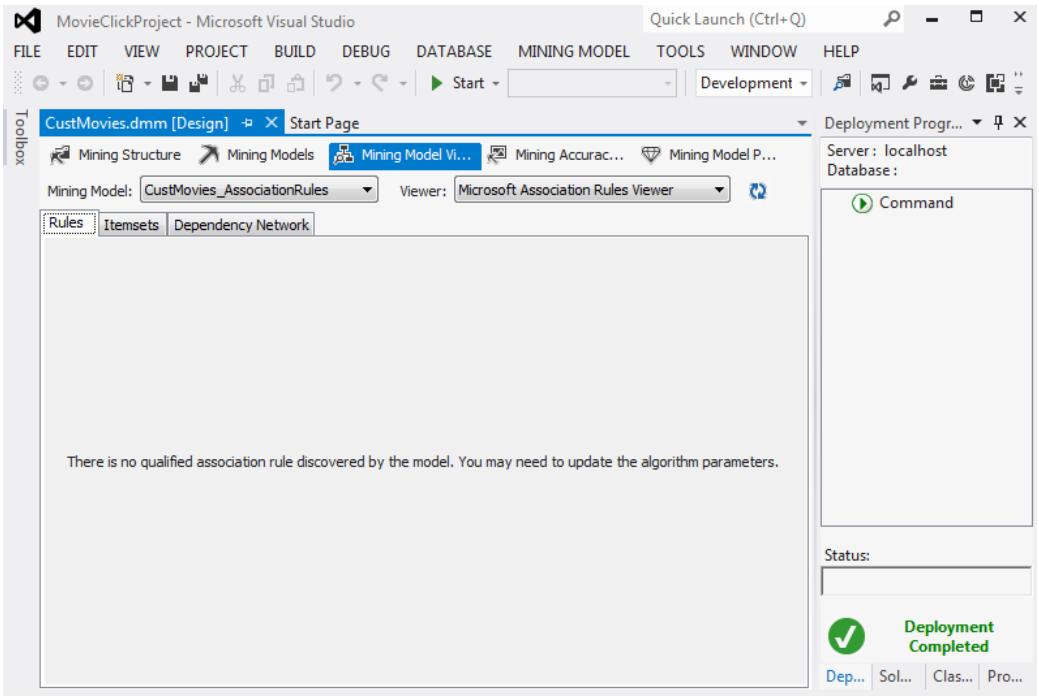
Εικόνα 9.17

2. Όπως φαίνεται στην καρτέλα itemsets της Εικόνας 9.18, τα itemsets που πληρούν την προϋπόθεση του 0.1 Minimum Support είναι μόλις 4. Βλέπουμε ότι η παράμετρος Minimum support πάνω από το Grid έχει τιμή 359, που ισοδυναμεί με το 10% όλων των εγγραφών του πίνακα Movies που εξετάζει ο αλγόριθμος.



Εικόνα 9.18

3. Τέλος, όπως φαίνεται στην καρτέλα Rules της Εικόνας 9.19, είναι πολύ δύσκολο να προχωρήσει με επιτυχία ο αλγόριθμος στη δεύτερη φάση υλοποίησής του, δηλαδή στην παραγωγή κανόνων συσχέτισης. Αυτό συμβαίνει διότι έχει ήδη απορριφθεί η πλειοψηφία των itemsets. Συνεπώς, δεν παράγεται κανένας κανόνας συσχέτισης.



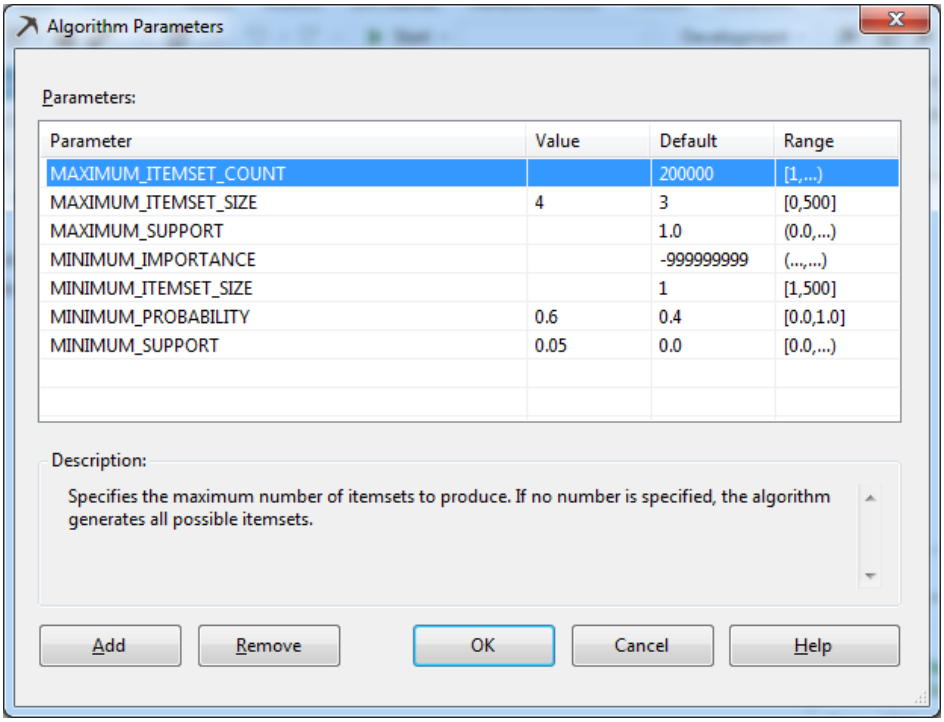
Εικόνα 9.19

Άσκηση 3

Να αλλάξετε τις παραμέτρους **(MINIMUM_SUPPORT, MINIMUM_PROBABILITY, MAXIMUM_ITEM_SIZE)** του αλγορίθμου Association Rules, θέτοντας σ' αυτές τις τιμές 0.05, 0.6 και 4, αντίστοιχα. Κατόπιν, να αξιολογήσετε το νέο μοντέλο (αξιολογώντας τις καρτέλες itemsets και rules).

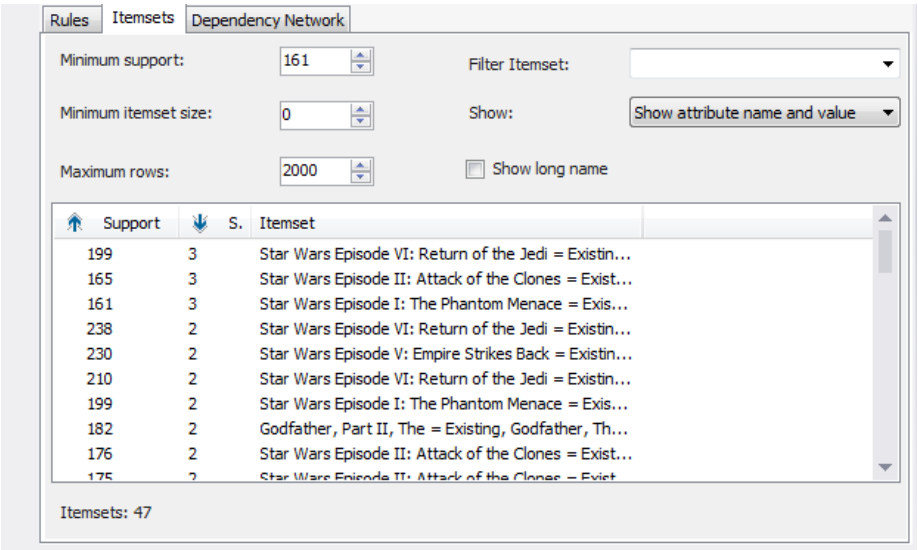
Λύση

1. Αλλάζουμε τις τιμές στις παραμέτρους, όπως φαίνεται στην Εικόνα 9.20.



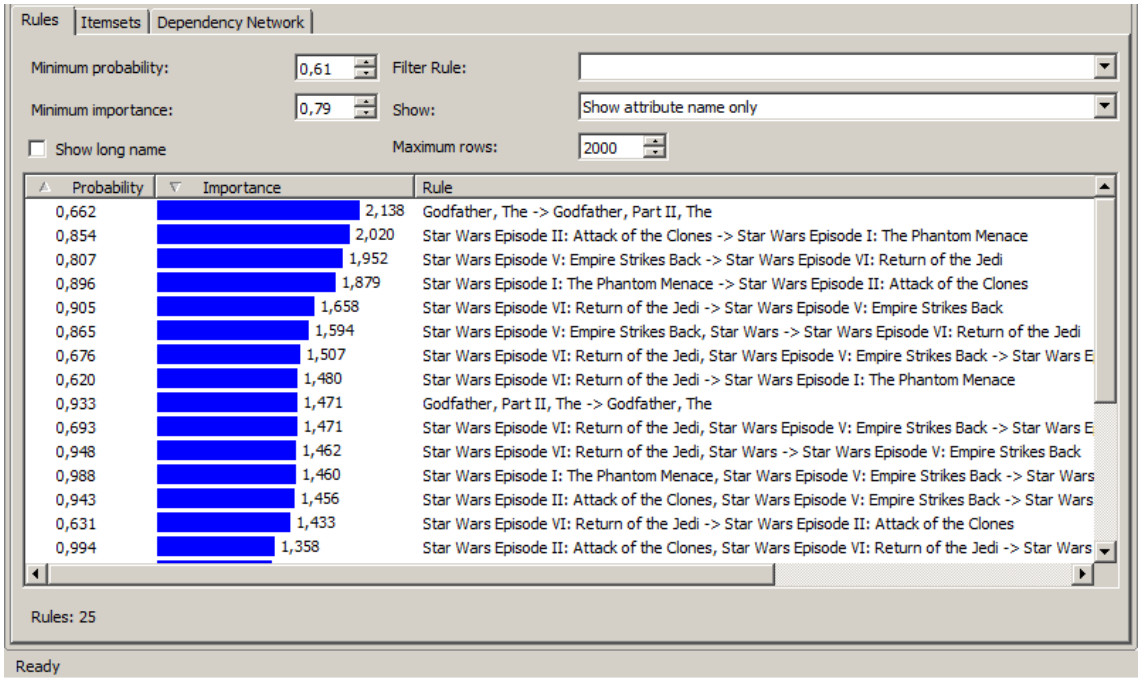
Εικόνα 9.20

2. Στην καρτέλα Itemsets, όπως φαίνεται στην Εικόνα 9.21, παρατηρήστε ότι δεν έχουν προκύψει itemsets με size 4.



Εικόνα 9.21

3. Στην καρτέλα Rules, όπως φαίνεται στην Εικόνα 9.22, παρατηρήστε ότι το Minimum Probability έχει προσδιοριστεί σε 0.61, ενώ το Minimum Importance έχει προσδιοριστεί σε 0.79. Όπως αναμενόταν, στην πρώτη στήλη του πίνακα δεν υπάρχει κανένας κανόνας με τιμή μικρότερη από 0.6 για τον δείκτη probability. Στη δεύτερη στήλη του πίνακα οι τιμές των κανόνων για το δείκτη importance είναι πολύ παραπάνω από τη μονάδα, υποδηλώνοντας θετική συσχέτιση μεταξύ των μεταβλητών του αριστερού και του δεξιού μέρους των κανόνων (βλέπε Ενότητα 9.1).



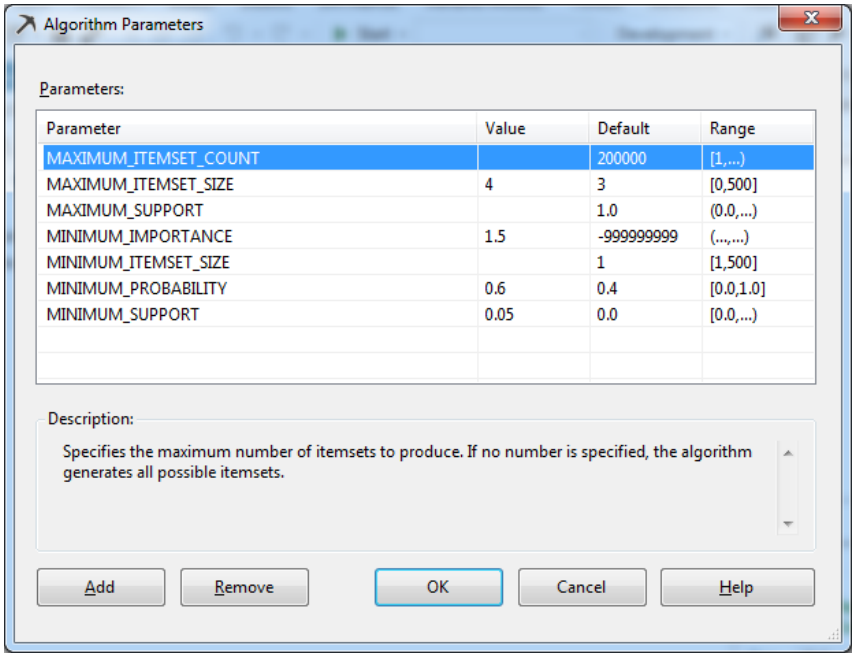
Εικόνα 9.22

Άσκηση 4

Να αλλάξετε τις παραμέτρους **(MINIMUM_SUPPORT, MINIMUM_PROBABILITY, MAXIMUM_ITEM_SIZE, MINIMUM IMPORTANCE)** του αλγορίθμου Association Rules, θέτοντας σ' αυτές τις τιμές 0.05, 0.6, 4 και 1.5 αντίστοιχα. Κατόπιν, να βρείτε τους δέκα κανόνες με τις μεγαλύτερες τιμές στο δείκτη probabilty.

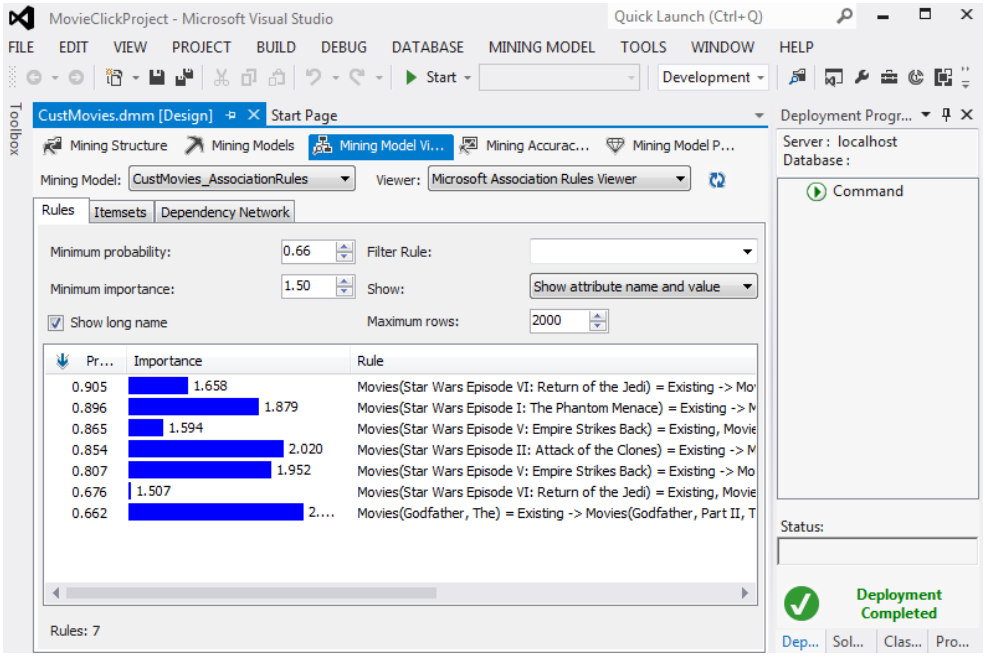
Λύση

1. Συμπληρώνουμε τις τιμές στις παραμέτρους, όπως φαίνεται στην Εικόνα 9.23.



Εικόνα 9.23

2. Στην καρτέλα Rules, όπως φαίνεται στην Εικόνα 9.24, παρατηρούμε ότι υπάρχουν επτά μόνο κανόνες, τους οποίους τους ταξινομούμε ως προς τον δείκτη probability (φθίνουσα ταξινόμηση).



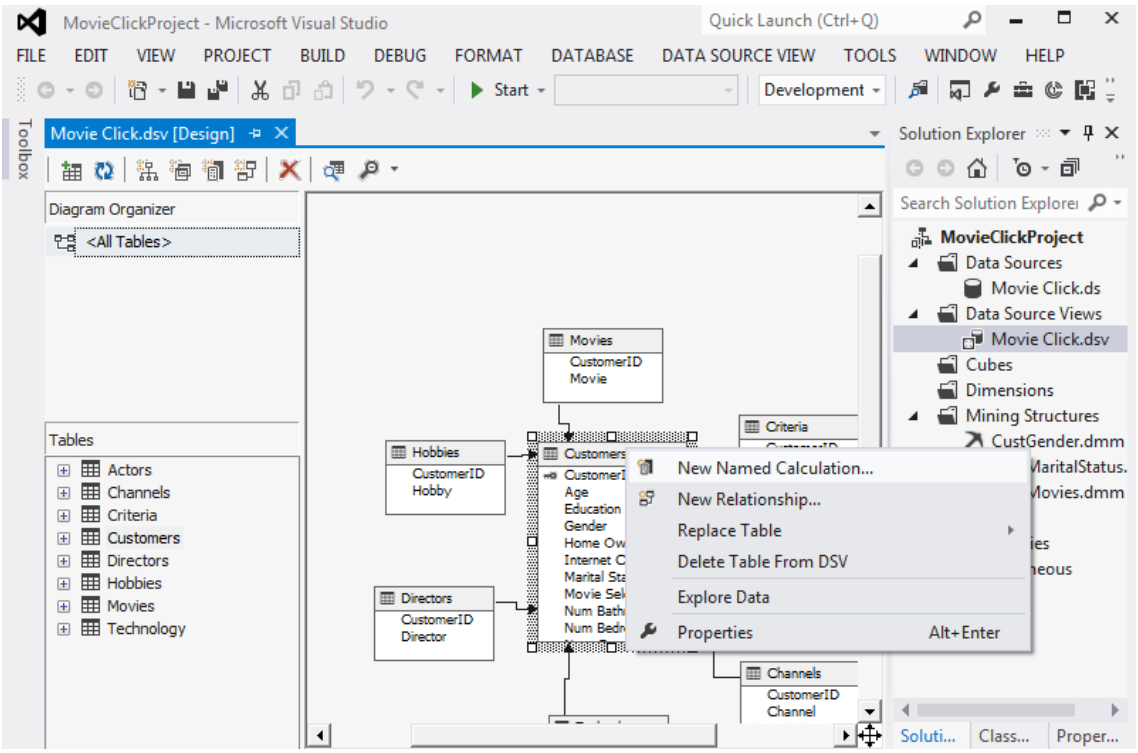
Εικόνα 9.24

Άσκηση 5

Να δημιουργήσετε ένα νέο μοντέλο που να συσχετίζει τις ταινίες με την ηλικία των πελατών. Αυτά τα δύο δεδομένα (ταινίες και ηλικία πελατών) να χρησιμοποιηθούν ως είσοδοι (input variables) στο νέο μοντέλο, αλλά να γίνεται και η πρόβλεψή τους (predicable variables).

Λύση

1. Καταρχήν, πρέπει να εισάγουμε τις ηλικίες των πελατών στο νέο μοντέλο. Ο αλγόριθμος, όμως, δεν μπορεί να επεξεργαστεί συνεχείς τιμές, οι οποίες περιέχονται στο πεδίο Age του πίνακα Customers. Συνεπώς, θα πρέπει να μετατραπούν σε διακριτές. Γι' αυτό, θα πρέπει να δημιουργήσουμε ένα νέο πεδίο στον πίνακα Customers. Επιλέγουμε, λοιπόν, την καρτέλα του designer της βάσης MovieClick.dsv, όπως φαίνεται στην Εικόνα 9.25. Στη συνέχεια, κάνουμε δεξί κλικ στον πίνακα Customers και επιλέγουμε New Named Calculation.



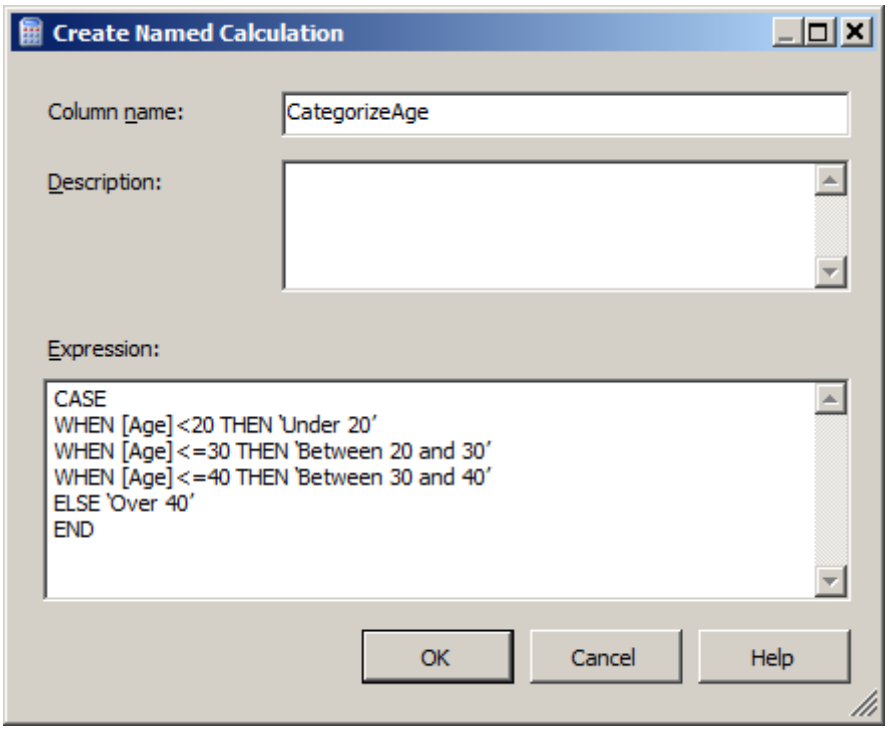
Εικόνα 9.25

2. Στο παράθυρο που εμφανίζεται, όπως φαίνεται στην Εικόνα 9.26, συμπληρώνουμε τα πεδία ως εξής:

   - Στο πεδίο Column name συμπληρώνουμε το όνομα του νέου πεδίου που δημιουργούμε στον πίνακα Customers. Στη συγκεκριμένη περίπτωση, συμπληρώνουμε το όνομα CategorizedAge.

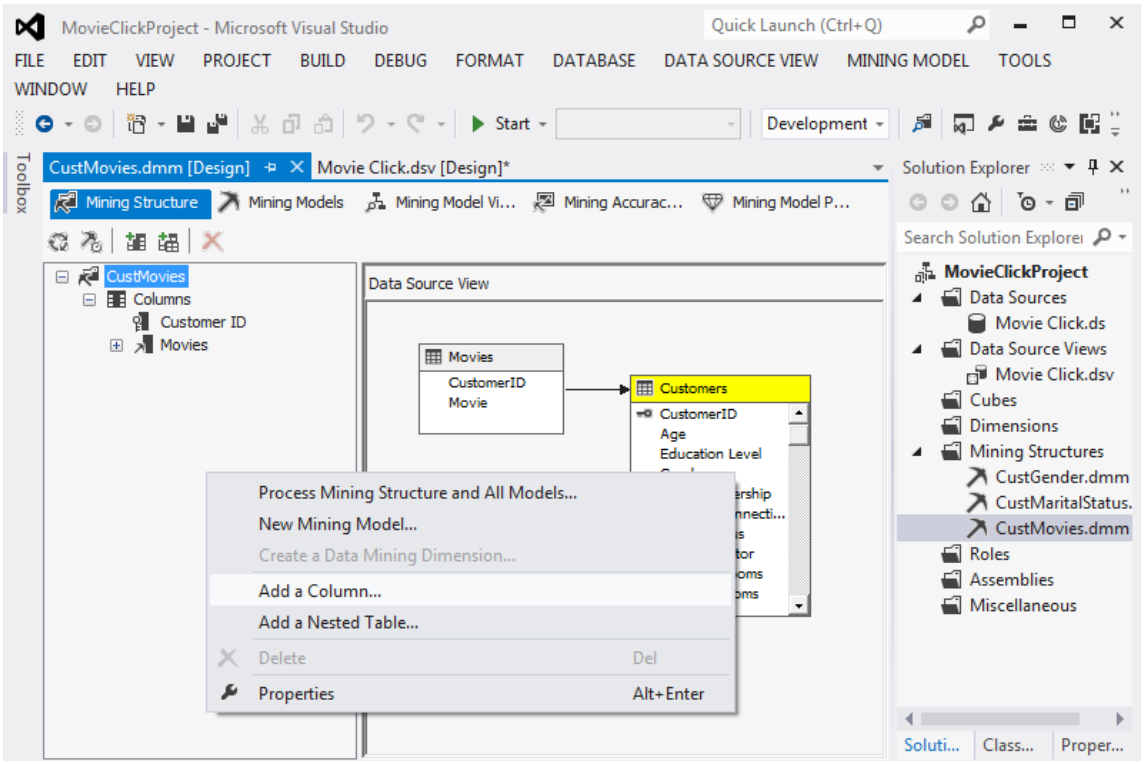
   - Στο πεδίο Expression συμπληρώνουμε τις εντολές με τις οποίες θα δημιουργηθεί το νέο πεδίο του πίνακα Customers:

     CASE
       WHEN [Age] < 20 THEN 'Under 20'
       WHEN [Age] <= 30 THEN 'Between 20 and 30'
       WHEN [Age] <= 40 THEN 'Between 30 and 40'
       ELSE 'Over 40'
     END



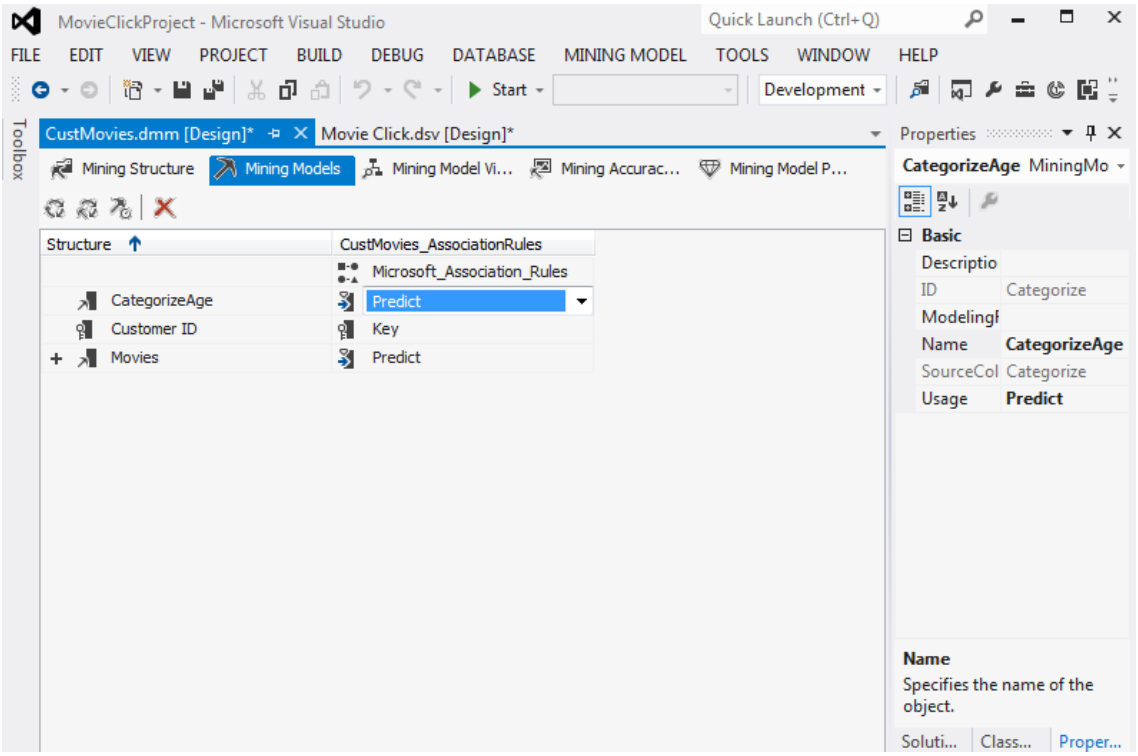
Εικόνα 9.26

3. Επιστρέφουμε στην καρτέλα Mining Structure του CustMovies.dmm, όπως φαίνεται στην Εικόνα 9.27, και κάνουμε add a Column, για να επιλέξουμε το πεδίο που μας ενδιαφέρει (CategorizeAge).



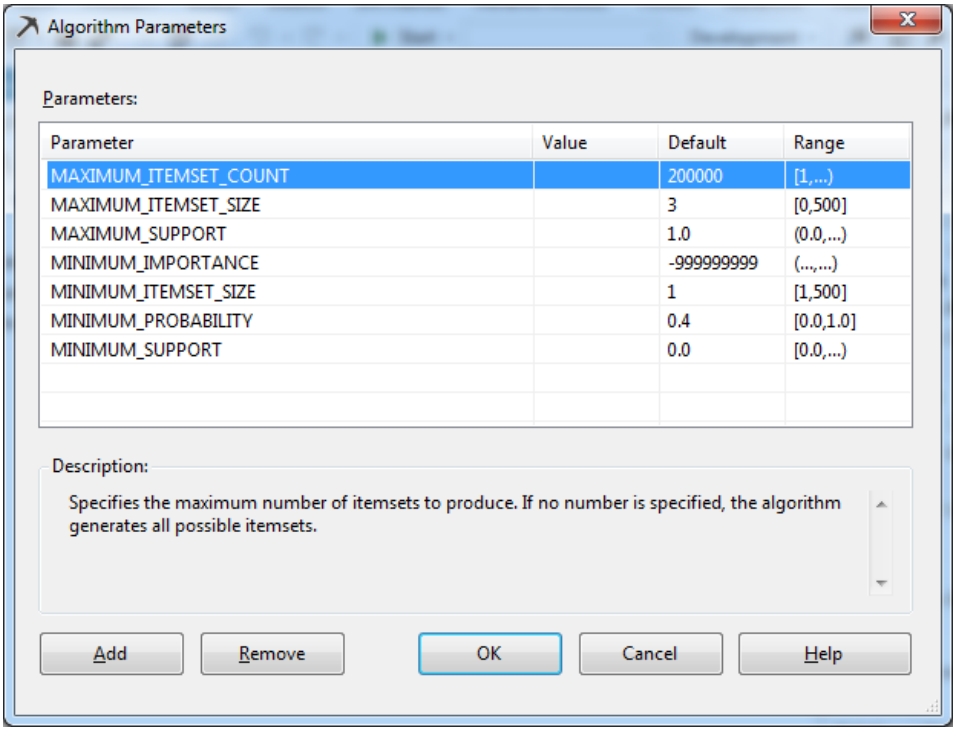
Εικόνα 9.27

4. Στην καρτέλα Mining Models, όπως φαίνεται στην Εικόνα 9.28, επιλέγουμε τα δεδομένα του πεδίου CategorizeAge να είναι Predict.



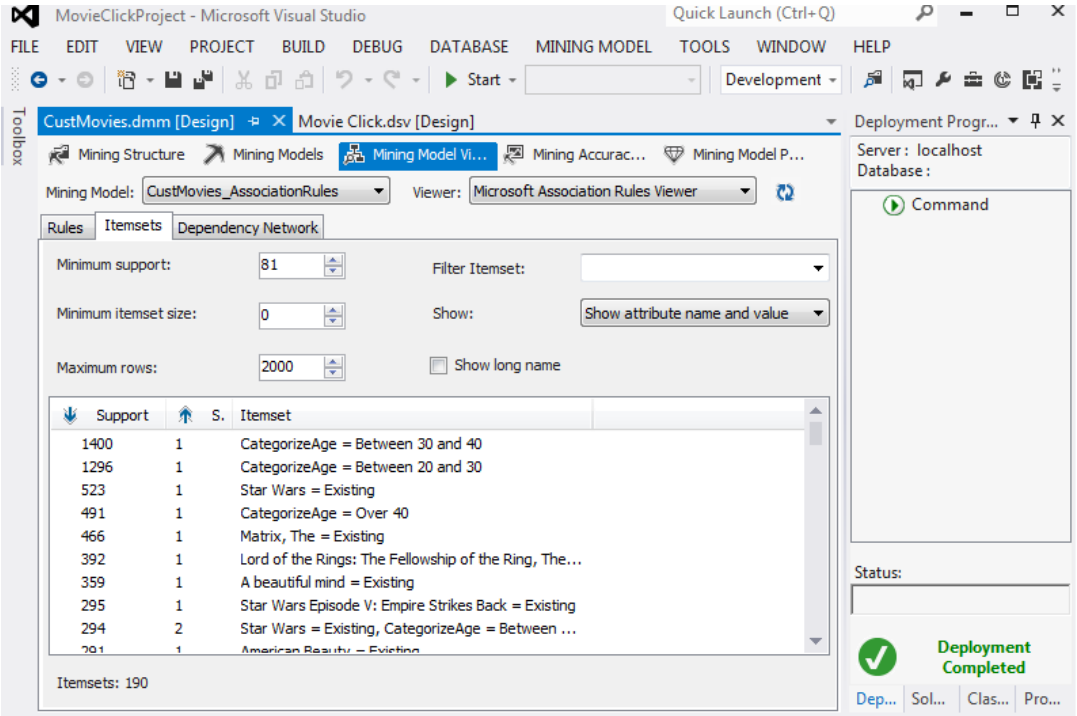
Εικόνα 9.28

5. Στο παράθυρο Set algorithms parameters, όπως φαίνεται στην Εικόνα 9.29, επαναφέρουμε τις τιμές των παραμέτρων στις αρχικές default τιμές.



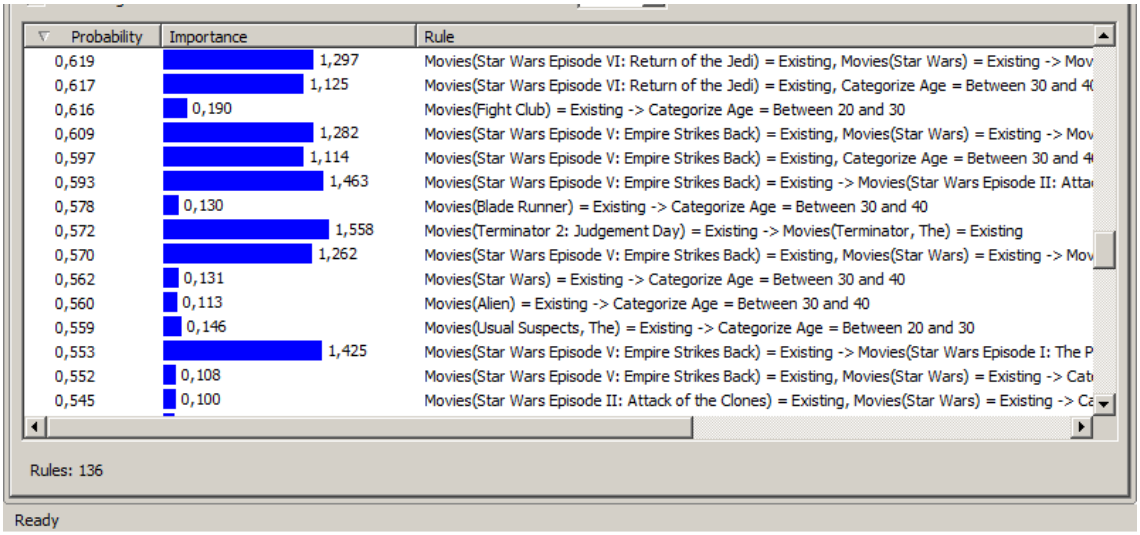
Εικόνα 9.29

6. Στην καρτέλα Itemsets, όπως φαίνεται στην Εικόνα 9.30, παρατηρούμε ότι έχουν δημιουργηθεί itemsets και για το πεδίο CategorizeAge.



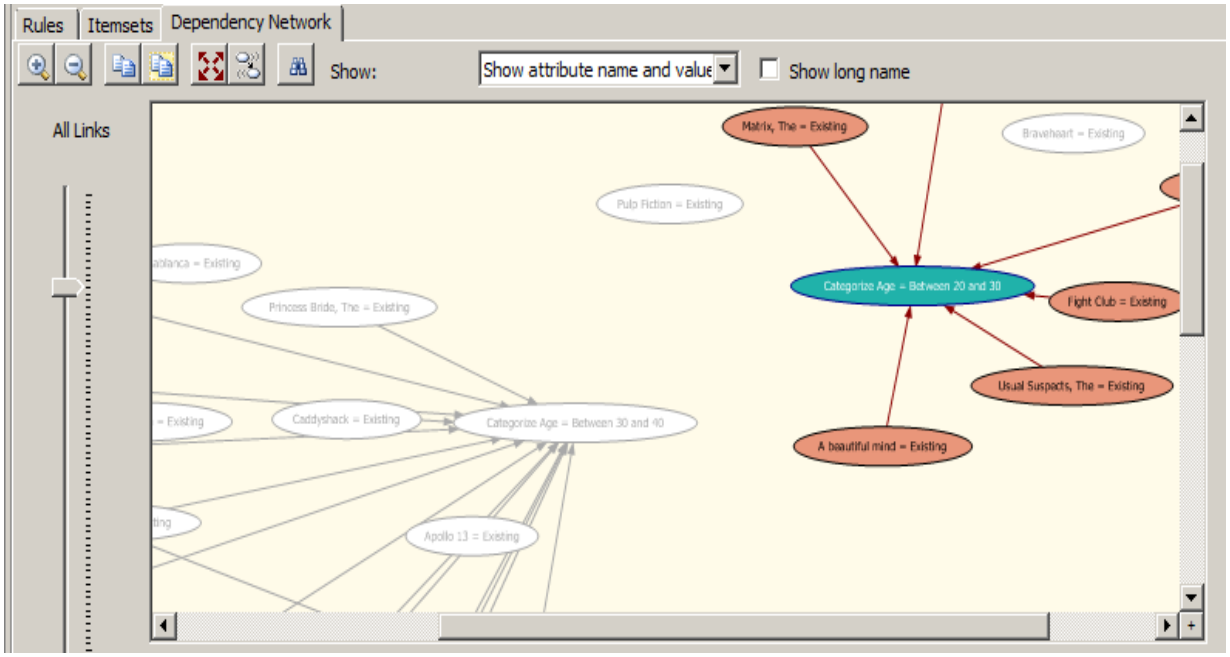
Εικόνα 9.30

7. Στην καρτέλα Rules, όπως φαίνεται στην Εικόνα 9.31, παρατηρούμε ότι έχουν δημιουργηθεί κανόνες που συσχετίζουν τις ταινίες με τις ηλικίες των πελατών. Για παράδειγμα, στην τρίτη γραμμή του πίνακα, την ταινία Fight Club την προτιμούν κυρίως πελάτες ηλικίας μεταξύ 20 και 30 ετών.



Εικόνα 9.31

8. Τέλος, στην καρτέλα Dependency Network, όπως φαίνεται στην Εικόνα 9.32, παρατηρούμε ότι τα δεδομένα μας έχουν χωριστεί κυρίως σε δύο μεγάλες ομάδες. Στη μία ομάδα ανήκουν οι ταινίες που προτιμούνται από πελάτες ηλικίας μεταξύ 30 και 40 ετών. Στην άλλη ομάδα ανήκουν οι ταινίες που σχετίζονται με πελάτες ηλικίας 20 έως 30 ετών.



Εικόνα 9.32

9.6. Βιβλιογραφία/Αναφορές

Νανόπουλος, Α., & Μανωλόπουλος, Ι. (2008). *Εισαγωγή στην Εξόρυξη και τις Αποθήκες Δεδομένων*, Αθήνα, Εκδόσεις Νέων Τεχνολογιών.

Χαλκίδη, Μ., & Βεζυργιάννης, Μ. (2005). *Εξόρυξη Γνώσης από Βάσεις Δεδομένων και τον Παγκόσμιο Ιστό*, Αθήνα, Τυπωθήτω.

Κεφάλαιο 10. Χρονοσειρές

Σύνοψη

Σ' αυτό το κεφάλαιο θα παρουσιάσουμε τη δημιουργία ενός μοντέλου με χρονοσειρές (time series). Συγκεκριμένα, θα μάθουμε τον τρόπο με τον οποίο δημιουργείται και χρησιμοποιείται ένα μοντέλο χρονοσειρών για την AdventureWorks. Η ***Adventure Works,*** *μια βάση δεδομένων που παρουσιάστηκε στις Ενότητες 6.3. και 6.6., αφορά μια πολυεθνική εταιρία που εμπορεύεται ποδήλατα σε διάφορες ηπείρους/χώρες. Το τμήμα πωλήσεων της εταιρίας επιθυμεί να προβλέψει τις μελλοντικές πωλήσεις ανά μοντέλο ποδηλάτου βάσει των πωλήσεων που σημειώθηκαν στο παρελθόν.*

10.1. Θεωρητικό υπόβαθρο των αλγορίθμων χρονοσειρών (time series) του SQL Server

Μια χρονοσειρά είναι το σύνολο διαδοχικών τιμών ενός χαρακτηριστικού για ένα χρονικό διάστημα. Πιο συγκεκριμένα, δοθέντος ενός χαρακτηριστικού *A*, μια χρονοσειρά είναι το σύνολο *N* παρατηρηθέντων ζευγών: $\{(t_1, a_1), (t_2, a_2), \dots, (t_n, a_n)\}$, όπου $T = \{1, 2, \dots, n\}$ είναι μια αλληλουχία διαδοχικών χρονικών στιγμών (π.χ. ημέρα, μήνας κτλ.), ενώ a_n είναι μια παρατηρηθείσα τιμή επί του χαρακτηριστικού *A* για τη χρονική στιγμή *n* ως παρατήρηση στο χρόνο αυτό (Aggarwal, 2015· Dunham, 2003· Han, & Kamber, 2001). Για παράδειγμα, η συλλογή των ημερήσιων τιμών κλεισίματος του χρηματιστηρίου Αθηνών για ένα έτος αποτελεί μια ακολουθία χρονοσειράς. Η ανάλυση μιας χρονοσειράς έχει ποικίλες εφαρμογές σε πολλούς επιστημονικούς τομείς, με σκοπό τη δημιουργία μοντέλων που θα προβλέπουν επαρκώς τις μελλοντικές τιμές ενός υπό εξέταση χαρακτηριστικού. Για παράδειγμα, ένας αλγόριθμος χρονοσειρών (time series) θα επέτρεπε την πρόβλεψη των μελλοντικών τιμών κλεισίματος των μετοχών εταιριών στο χρηματιστήριο Αθηνών.

Μερικά βασικά χαρακτηριστικά των χρονοσειρών είναι:

- Η **τάση** (trend): Είναι η συστηματική μεταβολή (γραμμική ή μη) των τιμών ενός χαρακτηριστικού στη μονάδα του χρόνου. Στον SQL Server μπορούμε να δούμε την τάση μιας χρονοσειράς σ' ένα διάγραμμα παρατηρώντας την καμπύλη πρόβλεψης, η οποία είτε αυξάνεται, είτε μειώνεται, είτε παραμένει σταθερή στη μονάδα του χρόνου.

- Η **περιοδικότητα (periodicity)**:Είναι ένα επαναλαμβανόμενο μοτίβο είτε υψηλών είτε χαμηλών παρατηρηθεισών τιμών ενός χαρακτηριστικού, το οποίο σχετίζεται με συγκεκριμένες χρονικές στιγμές (π.χ. κάθε φορά τα Χριστούγεννα αυξάνονται οι πωλήσεις). Στον αλγόριθμο χρονοσειρών του SQL Server μπορούμε να εντοπίσουμε ή να ορίσουμε την περιοδικότητα μέσα από τις παραμέτρους **AUTO_DETECT_PERIODICITY** και **PERIODICITY_HINT** αντίστοιχα.

- Οι **ακραίες τιμές (outliers)**:Είναι κάποιες παρατηρηθείσες τιμές ενός χαρακτηριστικού, οι οποίες διαφέρουν σημαντικά από τη μέση τιμή των υπόλοιπων τιμών του. Καθώς αυτές οι ακραίες τιμές συνήθως επηρεάζουν αρνητικά το μοντέλο πρόβλεψης, ενδείκνυται να εντοπίζονται και να απομακρύνονται από τα δεδομένα εκπαίδευσης. Στον αλγόριθμο χρονοσειρών του SQL Server μπορούμε να ορίσουμε ακρότατα με τις παραμέτρους **MAXIMUM_SERIES_VALUE** και **MINIMUM_SERIES_VALUE**.

Ο SQL Server διαθέτει δύο αλγορίθμους χρονοσειρών, τον **ARIMA** και τον **ARTXP**. Περιγράφουμε συνοπτικά τα πιο βασικά χαρακτηριστικά τους:

- Ο αλγόριθμος ARIMA (AutoRegressive Integrated Moving Average) χρησιμοποιείται κυρίως για μακροπρόθεσμες (long-term) προβλέψεις των τιμών ενός χαρακτηριστικού μιας χρονοσειράς. Η αυτοπαλινδρόμηση (autoregression) προβλέπει μελλοντικές τιμές μιας χρονοσειράς λαμβάνοντας υπόψη τις παλαιότερες παρατηρηθείσες τιμές της. Ο ARIMA υποθέτει ότι υπάρχει μια εξάρτηση μεταξύ των παρελθοντικών και μελλοντικών τιμών μιας χρονοσειράς, γνωστή ως μεταβαλλόμενη μέση τιμή (moving average). Η μεταβαλλόμενη μέση τιμή θεωρεί την υπό εξέταση χρονοσειρά ως μη στατική (non-stationary).

Αυτό σημαίνει ότι οι μελλοντικές τιμές της χρονοσειράς προκύπτουν από ένα μοντέλο μεταβαλλόμενης μέσης τιμής επί των παρελθοντικών τιμών της. Επιπροσθέτως, ο ARIMA είναι μονοπαραγοντικός (univariate). Αυτό σημαίνει ότι για τις προβλέψεις του δεν λαμβάνει υπόψη άλλα χαρακτηριστικά ως δεδομένα εισόδου παρά μόνο το υπό εξέταση χαρακτηριστικό της χρονοσειράς.

- Ο αλγόριθμος ARTXP (AutoRegressive Tree XP model) χρησιμοποιείται κυρίως για βραχυπρόθεσμες (short-term) και πολυπαραγοντικές (multivariate) προβλέψεις (Aggarwal, 2015· Dunham, 2003). Με τον όρο βραχυπρόθεσμες εννοούμε ότι ο ARTXP προβλέπει καλύτερα τις πιο άμεσες (πιο κοντινές) μελλοντικές τιμές ενός χαρακτηριστικού. Επίσης, είναι πολυπαραγοντικός και υποστηρίζει το λεγόμενο cross-prediction, επειδή λαμβάνει υπόψη και άλλα χαρακτηριστικά ως δεδομένα εισόδου για τη δημιουργία του μοντέλου χρονοσειράς. Ας υποθέσουμε, για παράδειγμα, ότι έχουμε μια βάση δεδομένων με πολλά χαρακτηριστικά για την οικονομία (ΑΕΠ, πληθωρισμό, επιτόκια καταθέσεων κτλ.). Ένας αλγόριθμος που υποστηρίζει το cross-prediction θα στηριχθεί στην εξελικτική πορεία όλων των παραπάνω διαθέσιμων χαρακτηριστικών, για να προβλέψει τις επόμενες τιμές τού υπό εξέταση χαρακτηριστικού (π.χ. του ΑΕΠ).

Στο περιβάλλον του SQL Server χρησιμοποιείται εξ ορισμού ένα μείγμα (mixed) των δύο παραπάνω αλγορίθμων (mixed model). Στην περίπτωση που ο χρήστης δεν επιθυμεί τη χρήση του μεικτού μοντέλου, η παράμετρος **FORECAST_METHOD** επιτρέπει την επιλογή μόνο ενός από τους δύο αλγορίθμους. Αντίστοιχα, η παράμετρος **PREDICTION_SMOOTHING** δίνει τη δυνατότητα να αποδώσουμε διαφορετική βαρύτητα στους δυο αλγορίθμους. Τονίζεται ότι στην Ενότητα 10.2. θα περιγραφούν οι υπόλοιπες παράμετροι του αλγορίθμου χρονοσειρών του SQL Server.

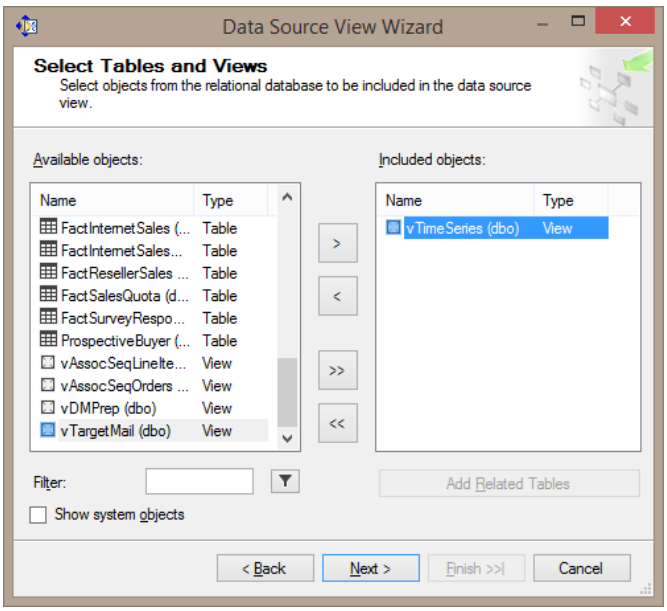
10.2. Δημιουργία ενός μοντέλου πρόβλεψης χρονοσειρών

Ας υποθέσουμε ότι έχει ζητηθεί απ' το τμήμα πωλήσεων να προβλέψουμε τις μελλοντικές πωλήσεις ανά κωδικό προϊόντος για το επόμενο έτος. Συγκεκριμένα, ζητήθηκε να βρούμε τις περιόδους αιχμής πώλησης ποδηλάτων και να μάθουμε πού αυξάνονται και πού υστερούν οι πωλήσεις σε σχέση με την κάθε περιοχή/ χώρα πώλησης. Επιπλέον, ζητήθηκε να καθορίσουμε αν οι πωλήσεις των προϊόντων διαφοροποιούνται ανάλογα με την εποχή του έτους. Τα στοιχεία της εταιρείας θα μελετηθούν σε μηνιαίο επίπεδο και θα αφορούν τις πωλήσεις για τρεις περιοχές: Ευρώπη, Βόρεια Αμερική και Ειρηνικό.

Τα βήματα για τη δημιουργία ενός μοντέλου χρονοσειρών είναι τα ακόλουθα:

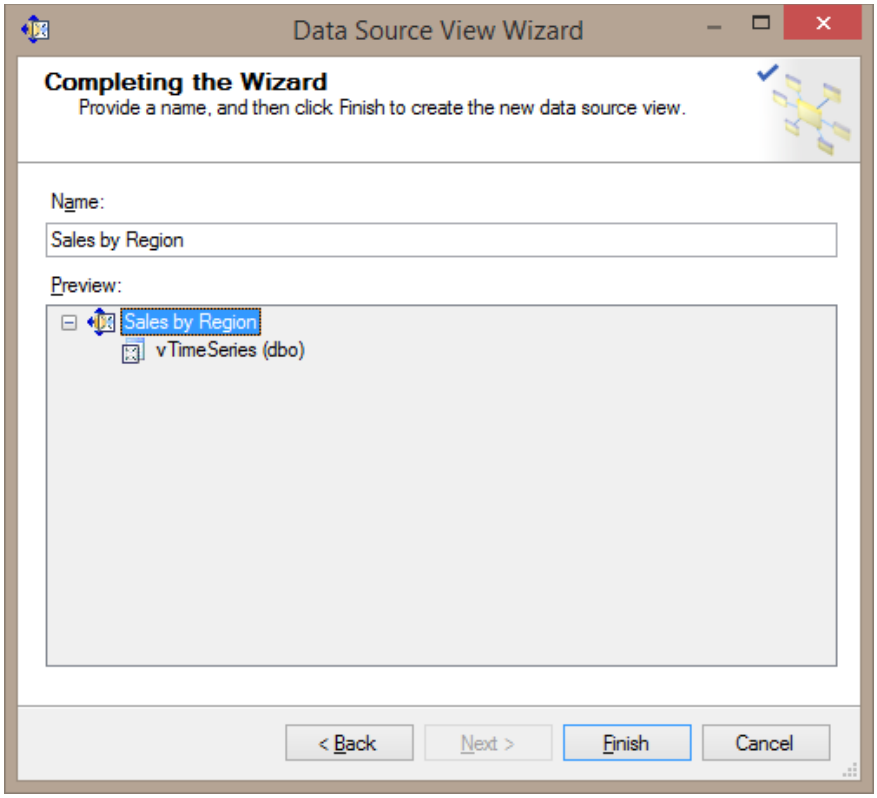
Αναλυτικά Βήματα

1. Αρχικά, όπως φαίνεται στην Εικόνα 10.1, δημιουργούμε ένα New Data Source View σύμφωνα με τα βήματα που προαναφέρθηκαν στην ενότητα 6.6. Κατόπιν, εισάγουμε τον πίνακα vTimeSeries (dbo).



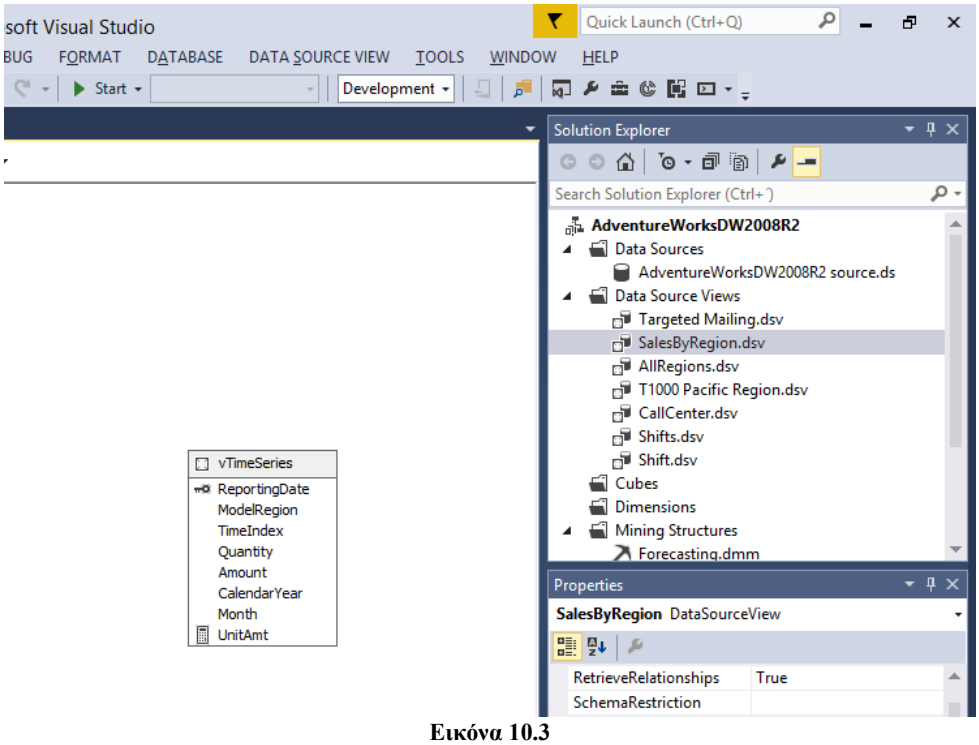
Εικόνα 10.1

2. Στη συνέχεια, δίνουμε στο Data Source View το όνομα SalesByRegion, όπως φαίνεται στην Εικόνα 10.2.



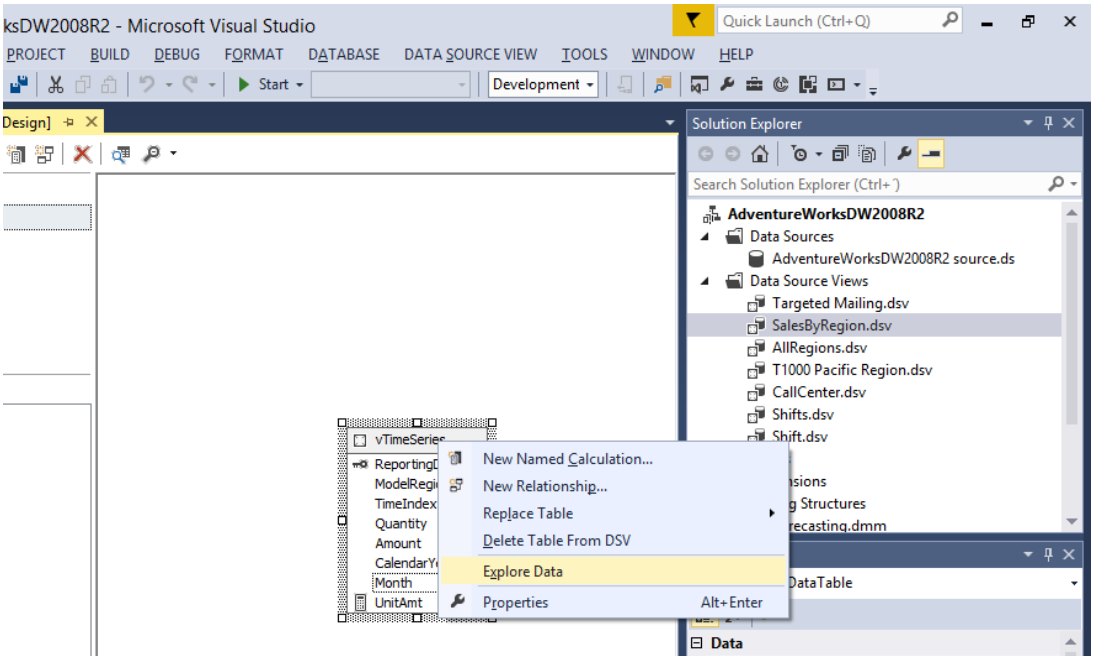
Εικόνα 10.2

3. Στη συνέχεια, όπως φαίνεται στην Εικόνα 10.3, εμφανίζεται το παράθυρο που περιέχει το σχεδιάγραμμα με τον πίνακα vTimeSeries.



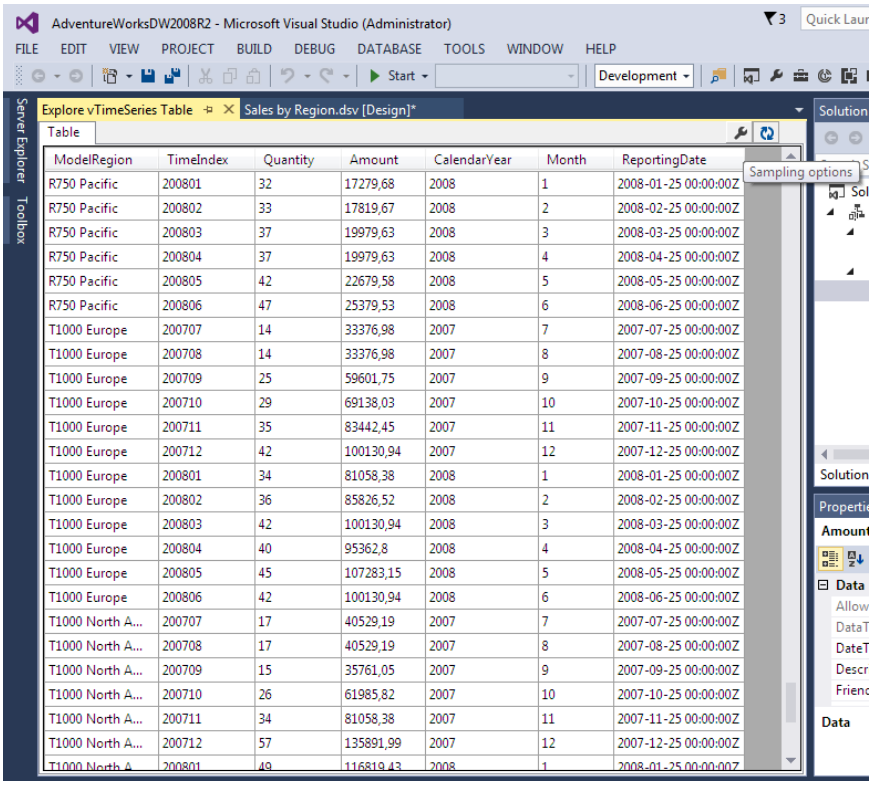
Εικόνα 10.3

4. Για να προσδιορίσουμε τον χρονικό ορίζοντα των δεδομένων του πίνακα vTimeSeries, θα πρέπει, όπως φαίνεται στην Εικόνα 10.4, να πάμε στο data source view του SalesByRegion, να κάνουμε δεξί κλικ στον πίνακα vTimeSeries και να επιλέξουμε Explore Data.



Εικόνα 10.4

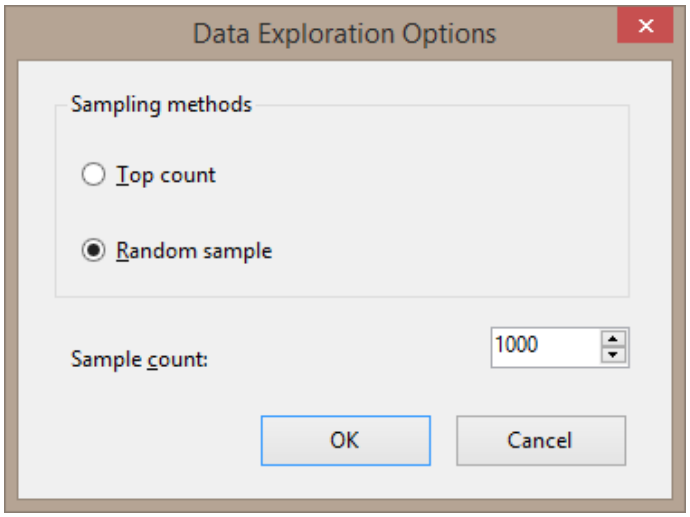
5. Η νέα καρτέλα που εμφανίζεται, όπως φαίνεται στην Εικόνα 10.5, περιέχει τον πίνακα vTimeSeries με διάφορες δυνατότητες επιλογής. Για παράδειγμα, μπορούμε να δούμε ότι δεν υπάρχουν δεδομένα για τα ποδήλατα με κωδικό "T1000 Europe" πριν από τον Ιούλιο του 2007, ενώ σε όλους τους λοιπούς κωδικούς ποδηλάτων υπάρχουν δεδομένα από τον Ιούλιο του 2005. Τονίζεται ότι αυτό δεν συνιστά πρόβλημα, καθώς όλοι οι κωδικοί προϊόντος έχουν δεδομένα μέχρι τον Ιούνιο του 2008.



ModelRegion	TimeIndex	Quantity	Amount	CalendarYear	Month	ReportingDate
R750 Pacific	200801	32	17279,68	2008	1	2008-01-25 00:00:00Z
R750 Pacific	200802	33	17819,67	2008	2	2008-02-25 00:00:00Z
R750 Pacific	200803	37	19979,63	2008	3	2008-03-25 00:00:00Z
R750 Pacific	200804	37	19979,63	2008	4	2008-04-25 00:00:00Z
R750 Pacific	200805	42	22679,58	2008	5	2008-05-25 00:00:00Z
R750 Pacific	200806	47	25379,53	2008	6	2008-06-25 00:00:00Z
T1000 Europe	200707	14	33376,98	2007	7	2007-07-25 00:00:00Z
T1000 Europe	200708	14	33376,98	2007	8	2007-08-25 00:00:00Z
T1000 Europe	200709	25	59601,75	2007	9	2007-09-25 00:00:00Z
T1000 Europe	200710	29	69138,03	2007	10	2007-10-25 00:00:00Z
T1000 Europe	200711	35	83442,45	2007	11	2007-11-25 00:00:00Z
T1000 Europe	200712	42	100130,94	2007	12	2007-12-25 00:00:00Z
T1000 Europe	200801	34	81058,38	2008	1	2008-01-25 00:00:00Z
T1000 Europe	200802	36	85826,52	2008	2	2008-02-25 00:00:00Z
T1000 Europe	200803	42	100130,94	2008	3	2008-03-25 00:00:00Z
T1000 Europe	200804	40	95362,8	2008	4	2008-04-25 00:00:00Z
T1000 Europe	200805	45	107283,15	2008	5	2008-05-25 00:00:00Z
T1000 Europe	200806	42	100130,94	2008	6	2008-06-25 00:00:00Z
T1000 North A...	200707	17	40529,19	2007	7	2007-07-25 00:00:00Z
T1000 North A...	200708	17	40529,19	2007	8	2007-08-25 00:00:00Z
T1000 North A...	200709	15	35761,05	2007	9	2007-09-25 00:00:00Z
T1000 North A...	200710	26	61985,82	2007	10	2007-10-25 00:00:00Z
T1000 North A...	200711	34	81058,38	2007	11	2007-11-25 00:00:00Z
T1000 North A...	200712	57	135891,99	2007	12	2007-12-25 00:00:00Z
T1000 North A	200801	49	116819,43	2008	1	2008-01-25 00:00:00Z

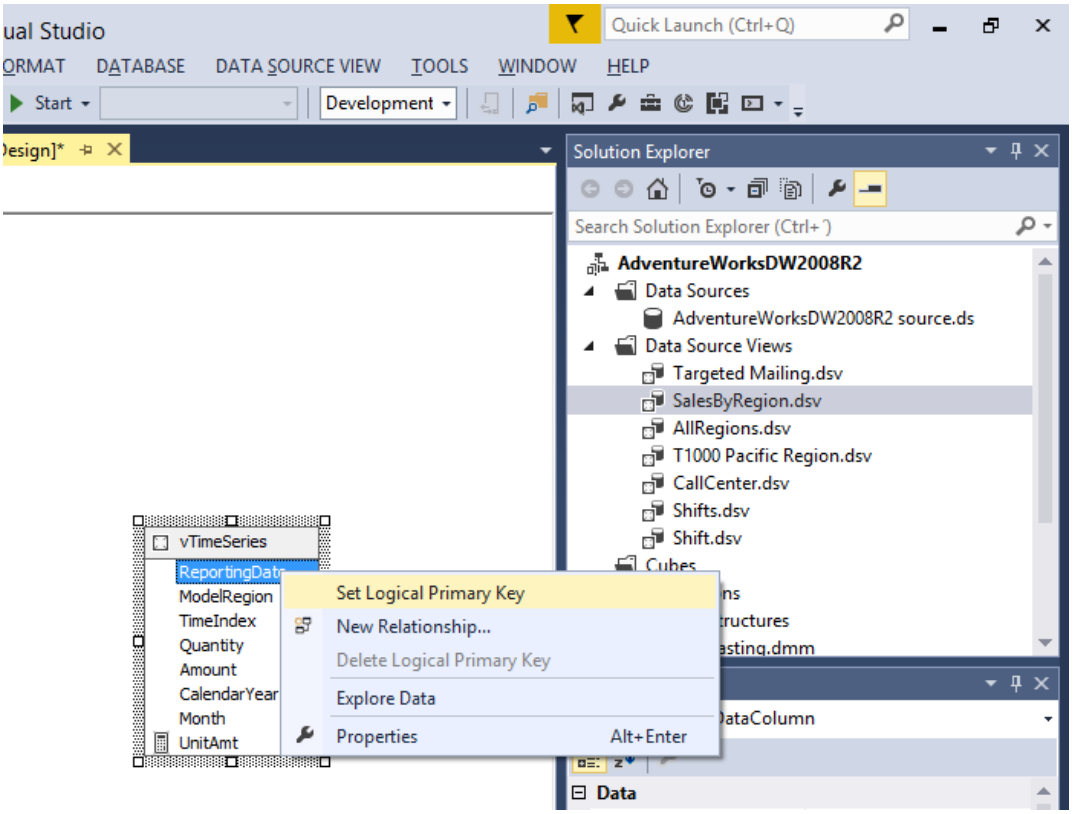
Εικόνα 10.5

6. Προκειμένου η επεξεργασία των δεδομένων να γίνει με τυχαία δειγματοληψία, κάνουμε κλικ στο κλειδί (υδραυλικό κλειδί) που βρίσκεται επάνω δεξιά στην Εικόνα 10.5, με αποτέλεσμα να εμφανίζεται η καρτέλα Data Exploration Options. Στο Sampling methods αυτής της καρτέλας της Εικόνας 10.6, όπως φαίνεται στην Εικόνα 10.6, φροντίζουμε να αποεπιλέξουμε το πεδίο Top count και να επιλέξουμε το πεδίο Random sample. Επίσης, στο Sample count ορίζουμε την τιμή 1000. Στη συνέχεια, πατάμε OK.



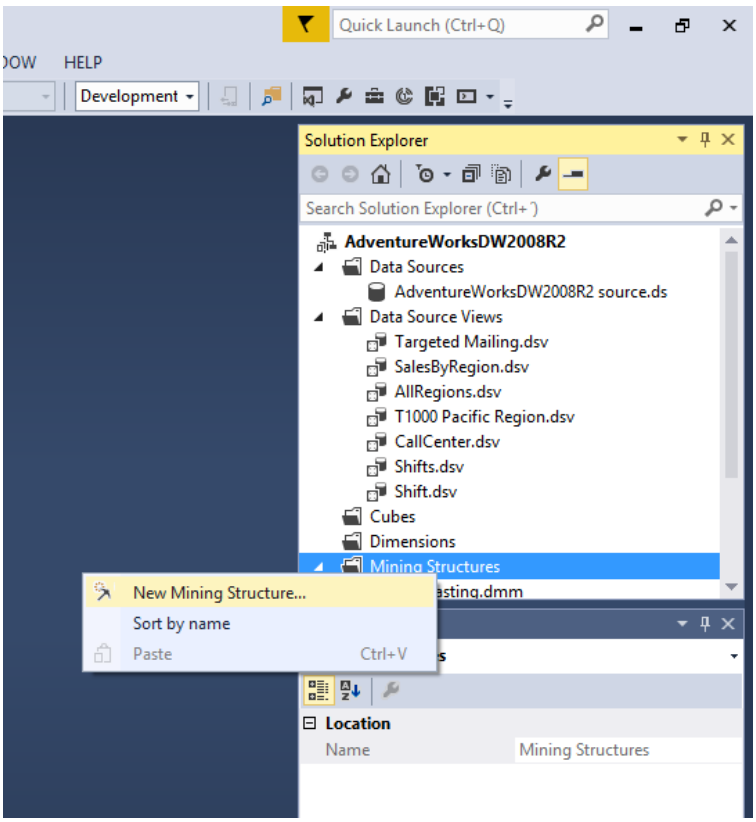
Εικόνα 10.6

7. Στη συνέχεια, επιλέγουμε το Reporting Date, για να το ορίσουμε ως Set Logical Primary Key, όπως φαίνεται στην Εικόνα 10.7.



Εικόνα 10.7

8. Στη συνέχεια, για να δημιουργήσουμε ένα mining structure, θα πρέπει να επιλέξουμε την καρτέλα Solution Explorer, να κάνουμε δεξί κλικ στο Mining Structure και να επιλέξουμε New Mining Structure, όπως φαίνεται στην Εικόνα 10.8.

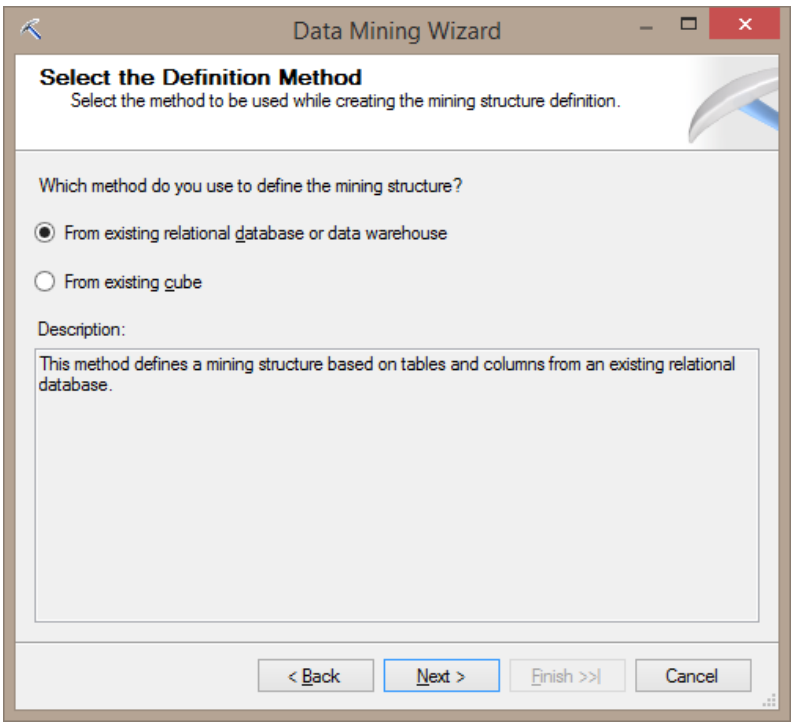


Εικόνα 10.8

9. Στο παράθυρο καλωσορίσματος του οδηγού Data Mining Wizard, όπως φαίνεται στην Εικόνα 10.9, επιλέγουμε Next>, ώστε να προχωρήσουμε στο επόμενο βήμα.

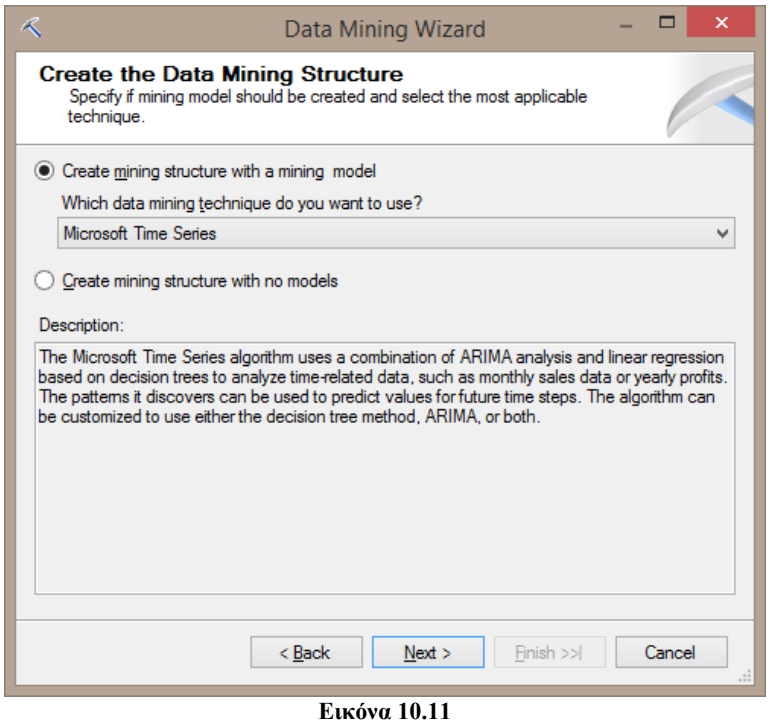
Εικόνα 10.9

10. Στο νέο παράθυρο, όπως φαίνεται στην Εικόνα 10.10, επιλέγουμε From existing relational database or data warehouse, καθώς τα δεδομένα μας θα εισαχθούν από την σχεσιακή βάση που εισάγαμε στην Ενότητα 6.3. Στη συνέχεια, πατάμε Next>, ώστε να προχωρήσουμε στο επόμενο βήμα.



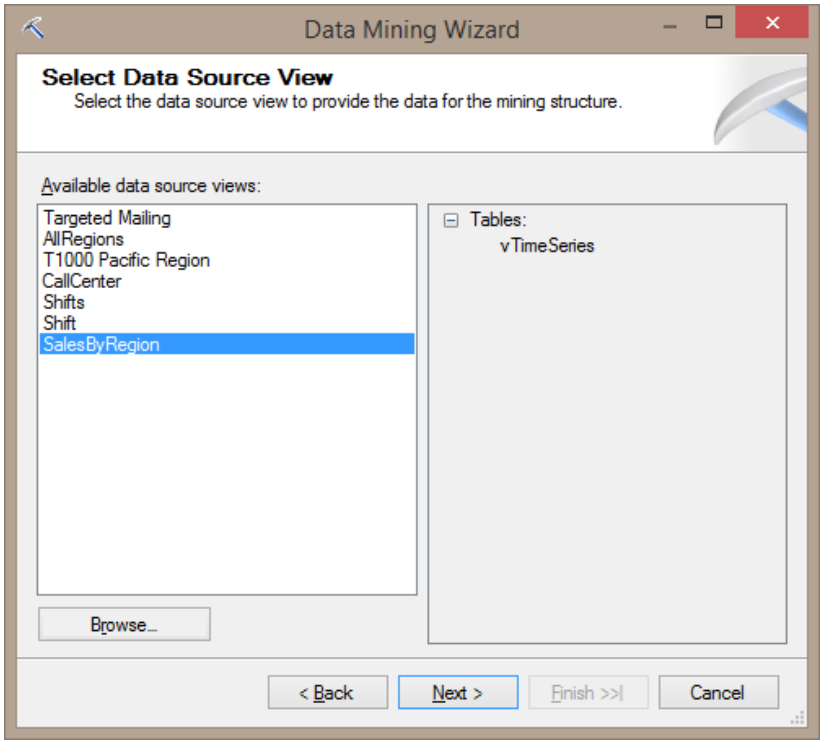
Εικόνα 10.10

11. Στο νέο παράθυρο, όπως φαίνεται στην Εικόνα 10.11, επιλέγουμε τον αλγόριθμο με τον οποίο θα επεξεργαστούμε τα δεδομένα. Στη συγκεκριμένη περίπτωση, επιλέγουμε τον Microsoft Time Series. Πατάμε Next>, ώστε να προχωρήσουμε στο επόμενο βήμα.



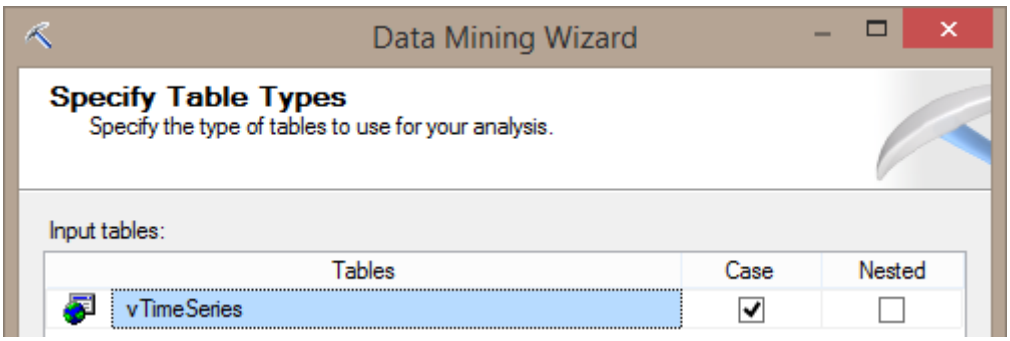
Εικόνα 10.11

13. Στη συνέχεια, εμφανίζεται το παράθυρο με τα διαθέσιμα data source views του project μας. Επιλέγουμε, όπως φαίνεται στην Εικόνα 10.12, το Sales By Region. Στη συνέχεια, πατάμε Next>, ώστε να προχωρήσουμε στο επόμενο βήμα.



Εικόνα 10.12

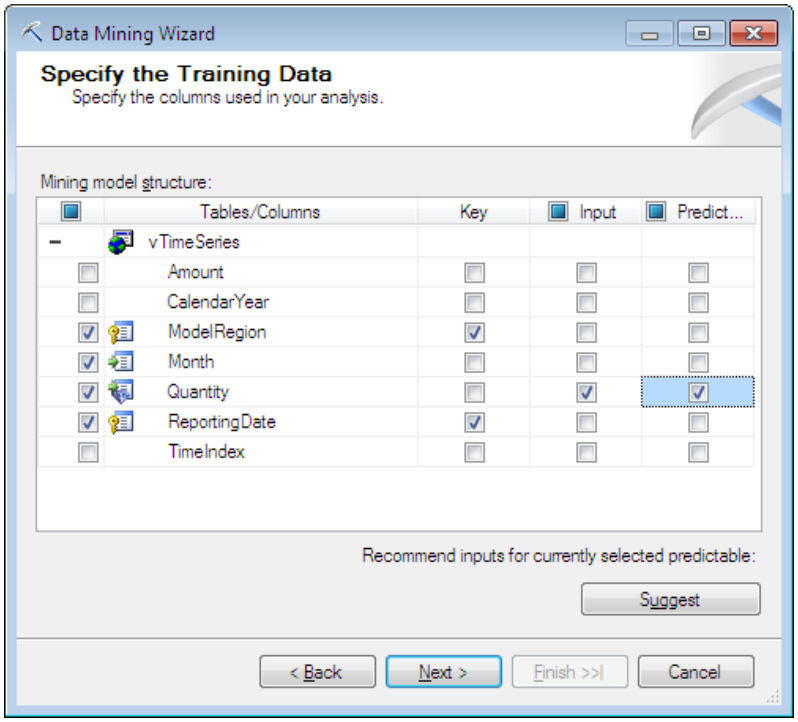
13. Σ' αυτό το βήμα θα επιλέξουμε ποιος πίνακας θα οριστεί ως case και ποιοι πίνακες θα είναι nested. Στη συγκεκριμένη περίπτωση, όπως φαίνεται στην Εικόνα 10.13, δεν θα ορίσουμε nested πίνακες. Επιλέγουμε ως Case τον πίνακα vTimeSeries table και επιλέγουμε Next.



Εικόνα 10.13

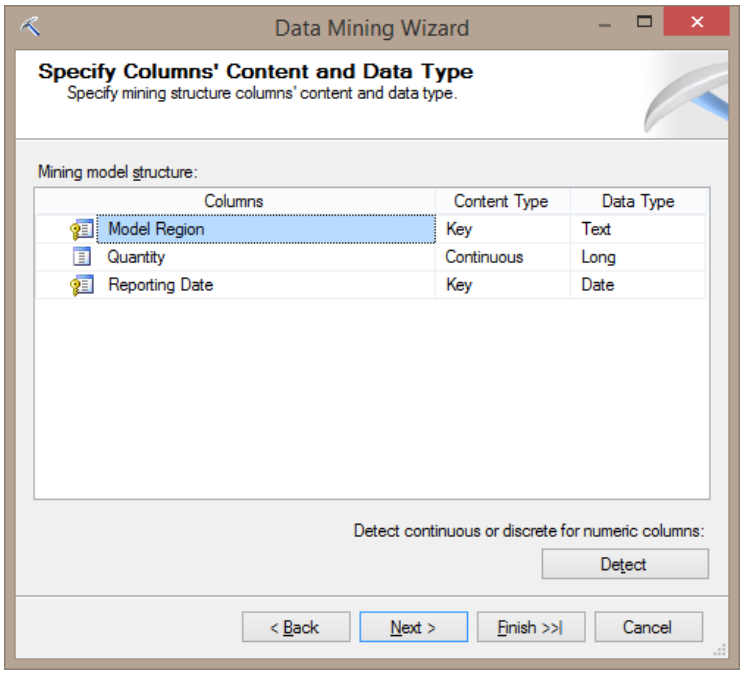
14. Σ’ αυτό το βήμα θα επιλέξουμε ποια δεδομένα που ορίσαμε στο προηγούμενο βήμα θα είναι είσοδος στη χρονοσειρά και ποια δεδομένα θέλουμε να προβλέψουμε.
    Συγκεκριμένα, όπως φαίνεται στην Εικόνα 10.14, κάνουμε τις εξής επιλογές:
    - Επιλέγουμε ως κλειδιά (**Keys)** τα **ModelRegion** και **ReportingDate** της όψης vTimeSeries**.**
    - Ορίζουμε ως **Input** το πεδίο **Quantity.**
    - Ορίζουμε ως **Predictable** ξανά το πεδίο **Quantity.**

    Στη συνέχεια, πατάμε Next >, ώστε να προχωρήσουμε στο επόμενο βήμα.



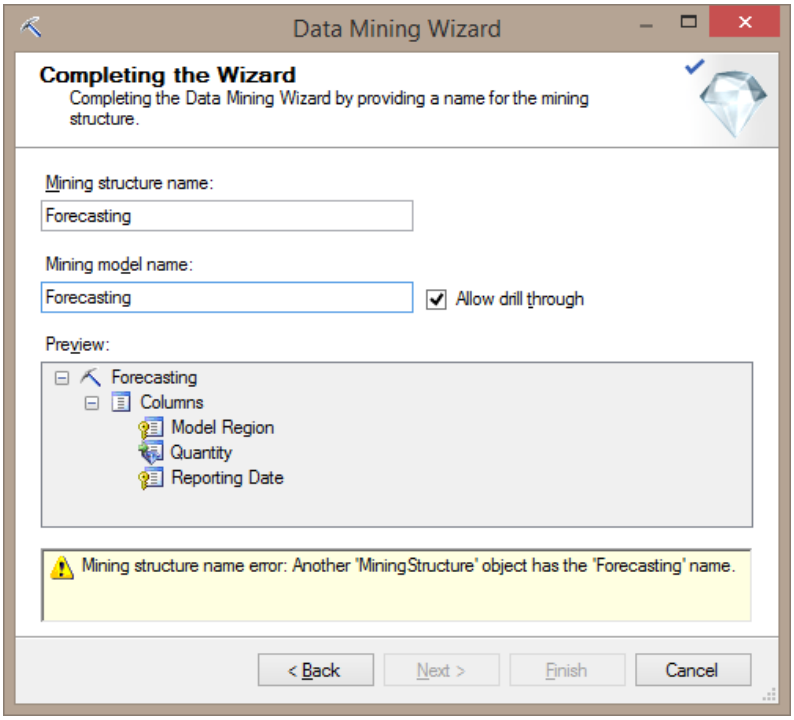
Εικόνα 10.14

15. Εμφανίζεται μια σύνοψη - επιβεβαίωση του περιεχομένου του Mining Structure, όπως φαίνεται στην Εικόνα 10.15. Κατόπιν, επιλέγουμε Next>, ώστε να προχωρήσουμε στο επόμενο βήμα.



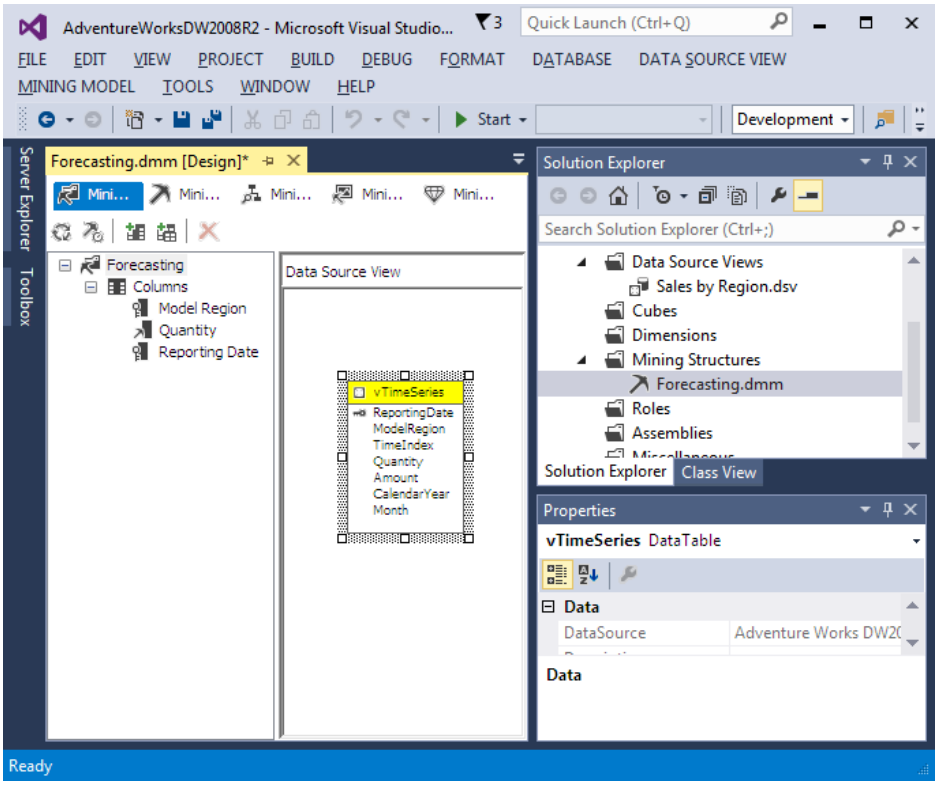
Εικόνα 10.15

16. Στο νέο παράθυρο θα ορίσουμε όνομα για το Mining Structure και το Mining Model. Στη συγκεκριμένη περίπτωση, συμπληρώνουμε Forecasting και στα δύο πεδία, όπως φαίνεται στην Εικόνα 10.16. Τέλος, επιλέγουμε Finish, για να ολοκληρωθεί η διαδικασία.



Εικόνα 10.16

17. Εμφανίζεται το παράθυρο του Data Mining Designer, όπως φαίνεται στην Εικόνα 10.17. Στην καρτέλα Mining Structures βλέπουμε το μοντέλο Forecasting.dmm που δημιουργήσαμε. Στη συνέχεια, μπορούμε να κάνουμε κλικ στο Start για να κάνουμε employment στο μοντέλο μας.

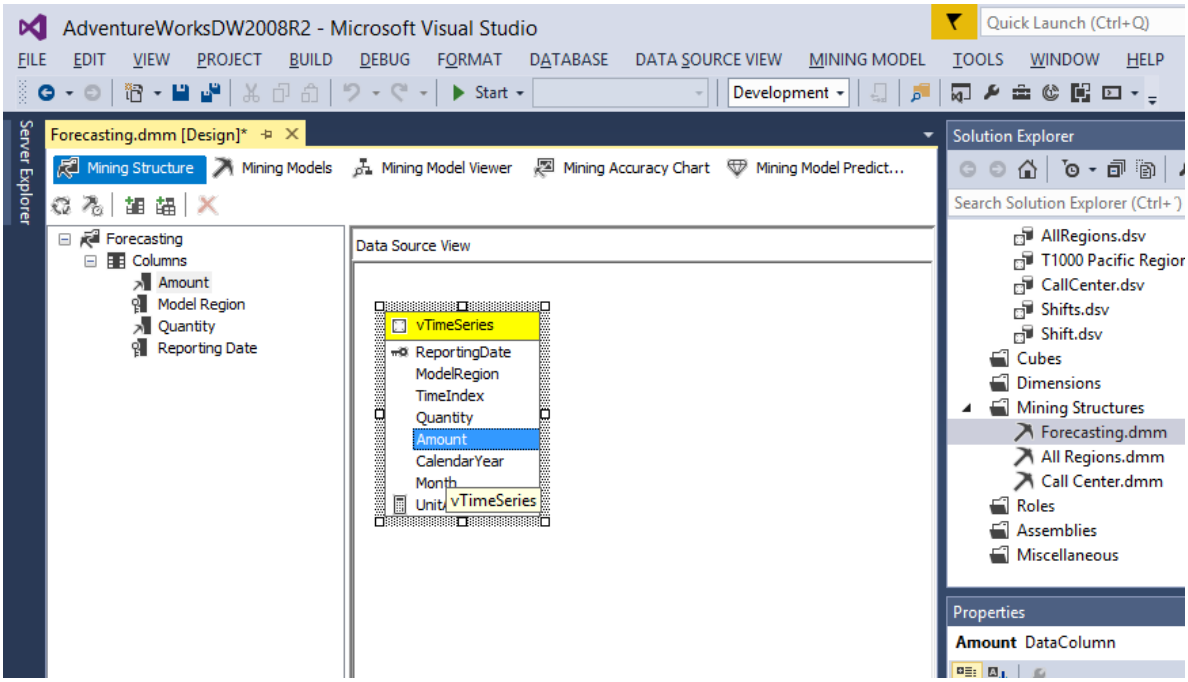


Εικόνα 10.17

10.3. Τροποποίηση και παραμετροποίηση του μοντέλου Time Series

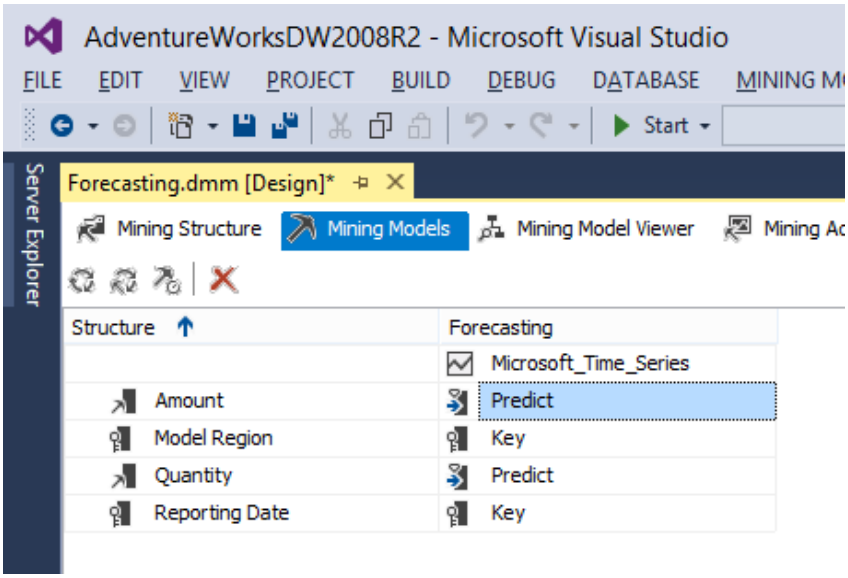
Μπορούμε να αλλάξουμε τη δομή του μοντέλου μας χρησιμοποιώντας την καρτέλα Mining Structure από το Data Mining Designer. Όταν δημιουργήθηκε το μοντέλο με τον Data Mining Wizard, χρησιμοποιήθηκαν τρία πεδία: ReportingDate, ModelRegion και Quantity. Ωστόσο, η υπό εξέταση όψη vTimeSeries περιέχει, επίσης, το πεδίο Amount, με το οποίο μπορούμε να προβλέψουμε τα εκάστοτε ποσά των πωλήσεών μας. Με τη χρήση της καρτέλας Mining Structure, μπορούμε να προσθέσουμε αυτήν τη στήλη από το data source view του mining structure.

1. Για να προστεθεί το Amount στο mining structure Forecasting, θα πρέπει να επιλέξουμε το Amount από τον πίνακα vTimeSeries και να το σύρουμε (με drag & drop) στη λίστα του Forecasting structure, όπως φαίνεται στην Εικόνα 10.18.



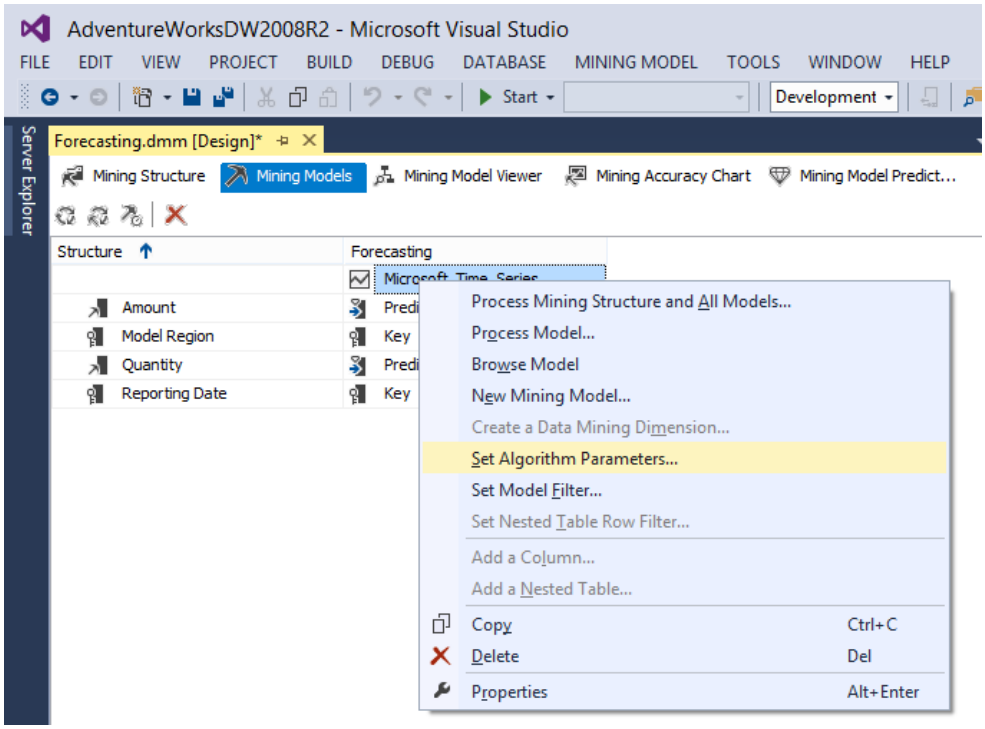
Εικόνα 10.18

2. Στο μοντέλο Forecasting, η στήλη Amount μπορεί να χρησιμοποιηθεί ως πεδίο πρόβλεψης. Ως εκ τούτου, πηγαίνουμε στην καρτέλα Mining Models του Forecasting, όπως φαίνεται στην Εικόνα 10.19, και επιλέγουμε στο κελί του Amount το Predict.



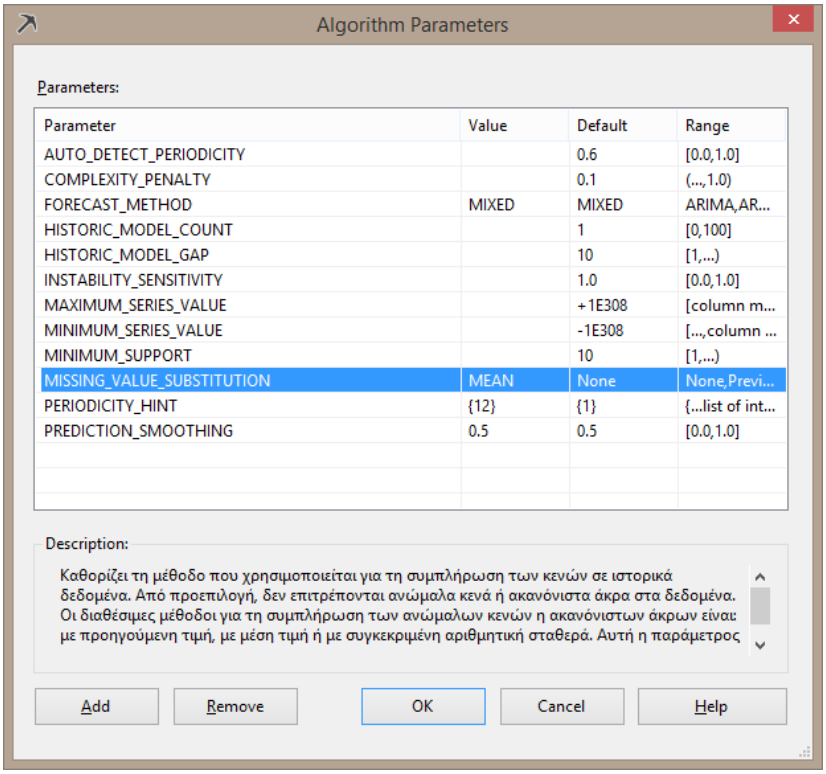
Εικόνα 10.19

3. Στη συνέχεια θα μελετήσουμε πώς επηρεάζονται η αποτελεσματικότητα και η συμπεριφορά του αλγορίθμου Time Series, αλλάζοντας μερικές απ' τις τιμές των παραμέτρων του. Όπως φαίνεται στην Εικόνα 10.20, κάνουμε δεξί κλικ στον τίτλο Microsoft Time Series και επιλέγουμε Set Algorithm Parameters.



Εικόνα 10.20

4. Όσον αφορά την παραμετροποίηση του αλγορίθμου Time Series, αυτός παρέχει δώδεκα διαφορετικές παραμέτρους, όπως φαίνεται στην Εικόνα 10.21, που επηρεάζουν το πώς ένα μοντέλο δημιουργείται και το πώς τα δεδομένα του χρόνου αναλύονται κάθε φορά.



Εικόνα 10.21

Ακολουθεί η αναλυτική περιγραφή της κάθε παραμέτρου του αλγορίθμου Time Series:

- **AUTO_DETECT_PERIODICITY**: Παίρνει τιμές μεταξύ 0 και 1 (default 0.6). Χρησιμοποιείται για την ανίχνευση περιοδικότητας. Όσο η τιμή τείνει στη μονάδα, ευνοείται η ανακάλυψη περισσότερων μοτίβων.

- **COMPLEXITY_PENALTY**: Ελέγχει την ανάπτυξη του δέντρου, οι κόμβοι του οποίου καθορίζουν την ύπαρξη διαφορετικής συμπεριφοράς ανά χρονική περίοδο στα δεδομένα της χρονοσειράς μας (default 0.1). Μείωση της τιμής της παραμέτρου επιφέρει αύξηση της πιθανότητας διαχωρισμού (split) στο δέντρο.

- **FORECAST_METHOD**: Καθορίζει τον αλγόριθμο που θα χρησιμοποιηθεί. Οι αλγόριθμοι είναι τρείς: ARTXP, ARIMA και MIXED (default τιμή: MIXED).

- **HISTORIC_MODEL_COUNT**: Καθορίζει τον αριθμό των μοντέλων που θα δημιουργηθούν (default τιμή: 1).

- **HISTORICAL_MODEL_GAP**: Καθορίζει τη χρονική καθυστέρηση ανάμεσα σε δύο διαδοχικά μοντέλα (default τιμή: 10). Ο αριθμός αντιστοιχεί σε χρονικές μονάδες βάσει του εκάστοτε μοντέλου.

- **INSTABILITY_SENSITIVITY**: Η συγκεκριμένη παράμετρος ελέγχει αν η διακύμανση των προβλέψεων ξεπερνά ένα κατώφλι που έχει οριστεί. Αφορά μόνο τον αλγόριθμο ARTPX (default τιμή: 1) και δεν χρησιμοποιείται για τον ARIMA. Όταν μια προβλεπόμενη τιμή ξεπεράσει το όριο που έχει οριστεί, ο ARTPX επιστρέφει NULL τιμές και σταματάει η διαδικασία.
- **MAXIMUM_SERIES_VALUE:** Η παράμετρος επιβάλλει έναν περιορισμό στην ανώτατη τιμή που

μπορεί να δοθεί σε μια πρόβλεψη. Χρησιμοποιείται για να θέσουμε ένα μέγιστο όριο στις τιμές των προβλέψεών μας.

- **MINIMUM_SERIES_VALUE**: Η παράμετρος επιβάλλει έναν περιορισμό στην κατώτατη τιμή που μπορεί να δοθεί σε μία πρόβλεψη. Χρησιμοποιείται για να θέσουμε **ένα** ελάχιστο όριο στις τιμές των προβλέψεών μας.

- **MINIMUM_SUPPORT**: Με αυτή την παράμετρο ορίζουμε τον ελάχιστο αριθμό περιπτώσεων (cases) που απαιτούνται για τη δημιουργία ενός κόμβου στο δέντρο που εκφράζει τα δεδομένα της χρονοσειράς μας (default τιμή: 10).

- **MISSING_VALUE_SUBSTITUTION**: Η παράμετρος καθορίζει τη μέθοδο συμπλήρωσης κενών στα δεδομένα της χρονοσειράς μας. Οι πιθανές τιμές που μπορεί να πάρει είναι: Previous (επανάληψη τιμής που σημειώθηκε την προηγούμενη χρονική στιγμή), Mean (μέσος), Numeric constant (μία σταθερή τιμή) και None (κενή τιμή).

- **PERIODICITY_HINT**: Η παράμετρος προσδιορίζει την ύπαρξη περιοδικότητας στα δεδομένα μας. Για παράδειγμα, αν οι πωλήσεις ποικίλλουν ανά έτος και οι μονάδες μέτρησης της χρονοσειράς είναι ανά μήνα, τότε η περιοδικότητα είναι {12}.

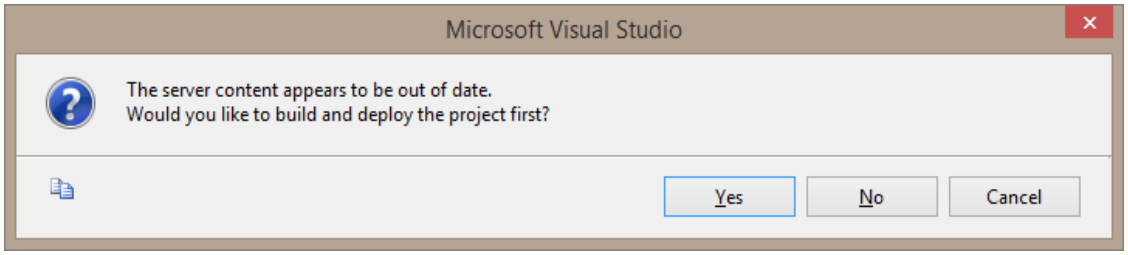
- **PREDICTION_SMOOTHING**: Η παράμετρος χρησιμοποιείται μόνο εφόσον έχει επιλεχθεί η τιμή MIXED στην παράμετρο FORECAST_METHOD. Η παράμετρος καθορίζει το ποσοστό συμμετοχής του κάθε αλγορίθμου (ARTXP και ARIMA).

5. Όσον αφορά την περιοδικότητα των δεδομένων (πόσο συχνά επαναλαμβάνεται ένα pattern στα δεδομένα), μπορούμε να ρυθμίσουμε την τιμή της παραμέτρου **PERIODICITY_HINT**, όπως φαίνεται στην Εικόνα 10.21. Τα δεδομένα της όψης vTimeSeries είναι διαμορφωμένα σε μηνιαία βάση και η περιοδικότητά τους ορίζεται σε ετήσιο επίπεδο. Ως εκ τούτου, θα ορίσουμε την παράμετρο **PERIODICITY_HINT** ίση με την τιμή {**12}**.

   Όπως ήδη αναφέρθηκε στην Ενότητα 10.1., η παράμετρος **FORECAST_METHOD** ρυθμίζει ποιος αλγόριθμος θα χρησιμοποιείται κάθε φορά (δηλαδή, αν θα χρησιμοποιηθεί ο αλγόριθμος ARTXP, για βραχυπρόθεσμες προβλέψεις ή ο αλγόριθμος ARIMA για μακροπρόθεσμες προβλέψεις). Βέβαια, η συγκεκριμένη παράμετρος έχει τεθεί εξ ορισμού σε **MIXED**, που σημαίνει ότι χρησιμοποιούνται και οι δύο αλγόριθμοι. Η παράμετρος **PREDICTION_SMOOTHING** είναι αυτή που ρυθμίζει το ποσοστό συμμετοχής του κάθε αλγορίθμου. Στην περίπτωσή μας, η παράμετρος αυτή ορίζεται στην τιμή **0.5**, επειδή παρέχει, γενικά, την καλύτερη ισορροπία μεταξύ βραχυπρόθεσμων και μακροπρόθεσμων προβλέψεων.

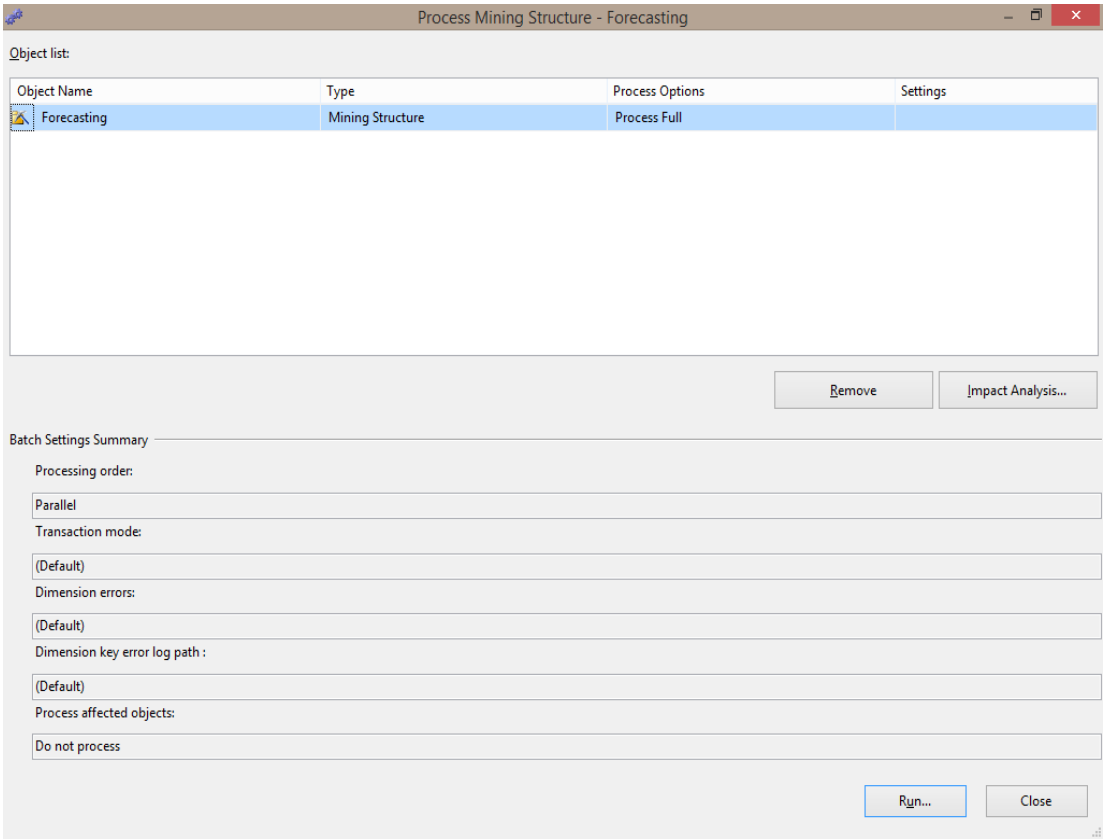
   Τέλος, η ύπαρξη κενών τιμών είναι ένα ακόμη πρόβλημα που μπορεί να παρουσιαστεί στα δεδομένα πωλήσεων. Για παράδειγμα, όπως φαίνεται στην Εικόνα 10.5, δεν υπάρχουν δεδομένα πωλήσεων για τα ποδήλατα με κωδικό T1000 Europe πριν από τον Ιούλιο του 2007. Για να αποφύγουμε την περίπτωση εμφάνισης κάποιου σφάλματος απ' το σύστημα, μπορούμε να καθορίσουμε στην παράμετρο **MISSING_VALUE_SUBSTITUTION** την τιμή **MEAN**, όπως φαίνεται στην Εικόνα 10.21, ώστε η μέση τιμή πώλησης της συνολικής χρονικής περιόδου πωλήσεων να καταχωρηθεί στις κενές τιμές.

6. Σε περίπτωση που δεν έχουν αποθηκευτεί οι αλλαγές που έχουμε κάνει, θα εμφανιστεί μήνυμα, όπως φαίνεται στην Εικόνα 10.22, στο οποίο επιλέγουμε Yes.



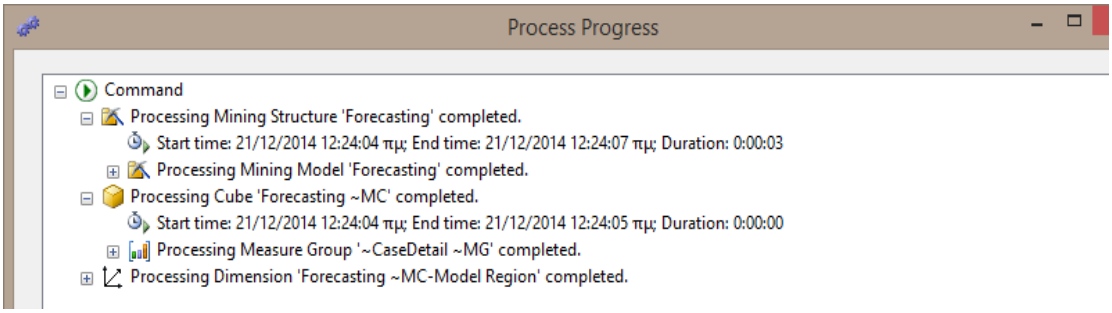
Εικόνα 10.22

7. Στη συνέχεια, εμφανίζεται ένα νέο παράθυρο, όπως φαίνεται στην Εικόνα 10.23, στο οποίο βρίσκονται συγκεντρωμένες οι επιλογές μας. Επιλέγουμε Run, ώστε να δημιουργηθεί το μοντέλο και να γίνει deployment.



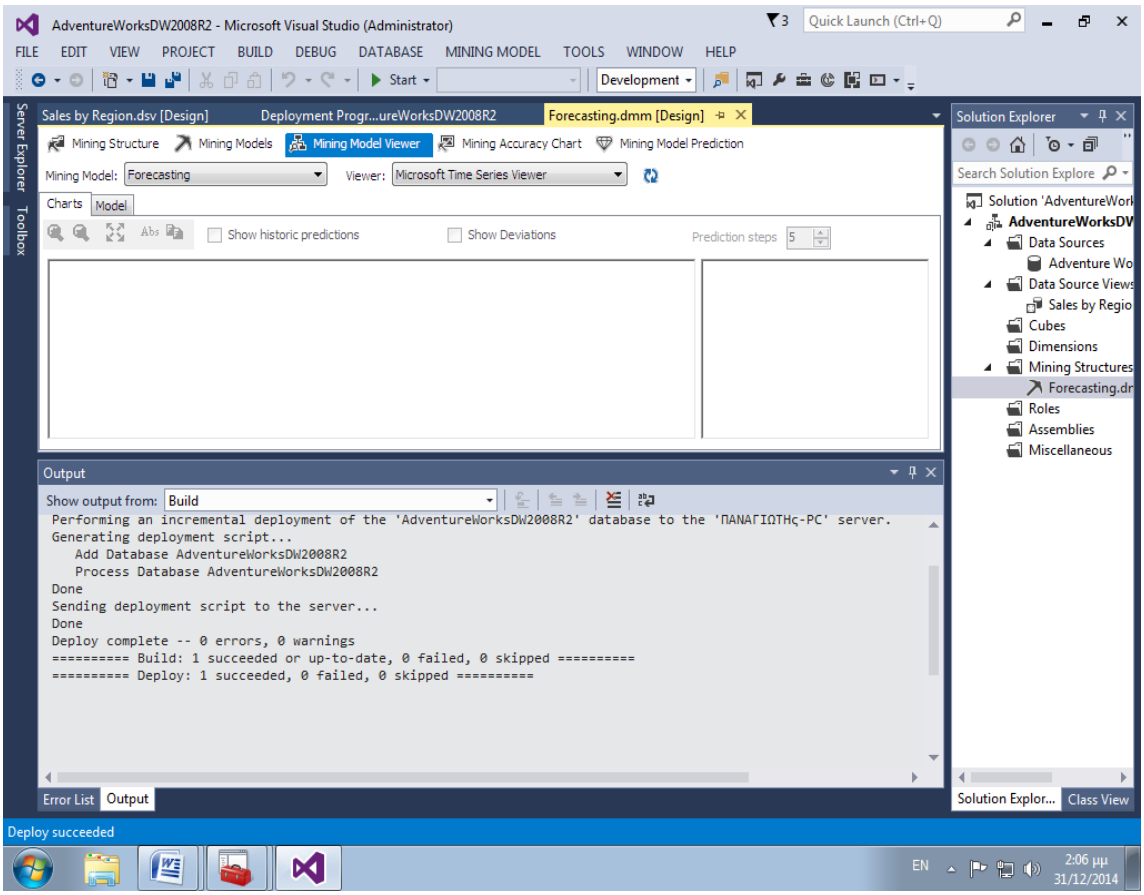
Εικόνα 10.23

8. Στη συνέχεια, εμφανίζεται ένα νέο παράθυρο, όπως φαίνεται στην Εικόνα 10.24, που παρουσιάζει τις ενέργειες που έγιναν για τη δημιουργία της πρόβλεψης. Παράλληλα, μας πληροφορεί αν αυτές οι ενέργειες ολοκληρώθηκαν με επιτυχία. Επιλέγουμε Close, ώστε να ολοκληρωθεί η διαδικασία.



Εικόνα 10.24

9. Αν το employment είναι επιτυχημένο, τότε θα εμφανιστεί το παράθυρο της Εικόνας 10.25.



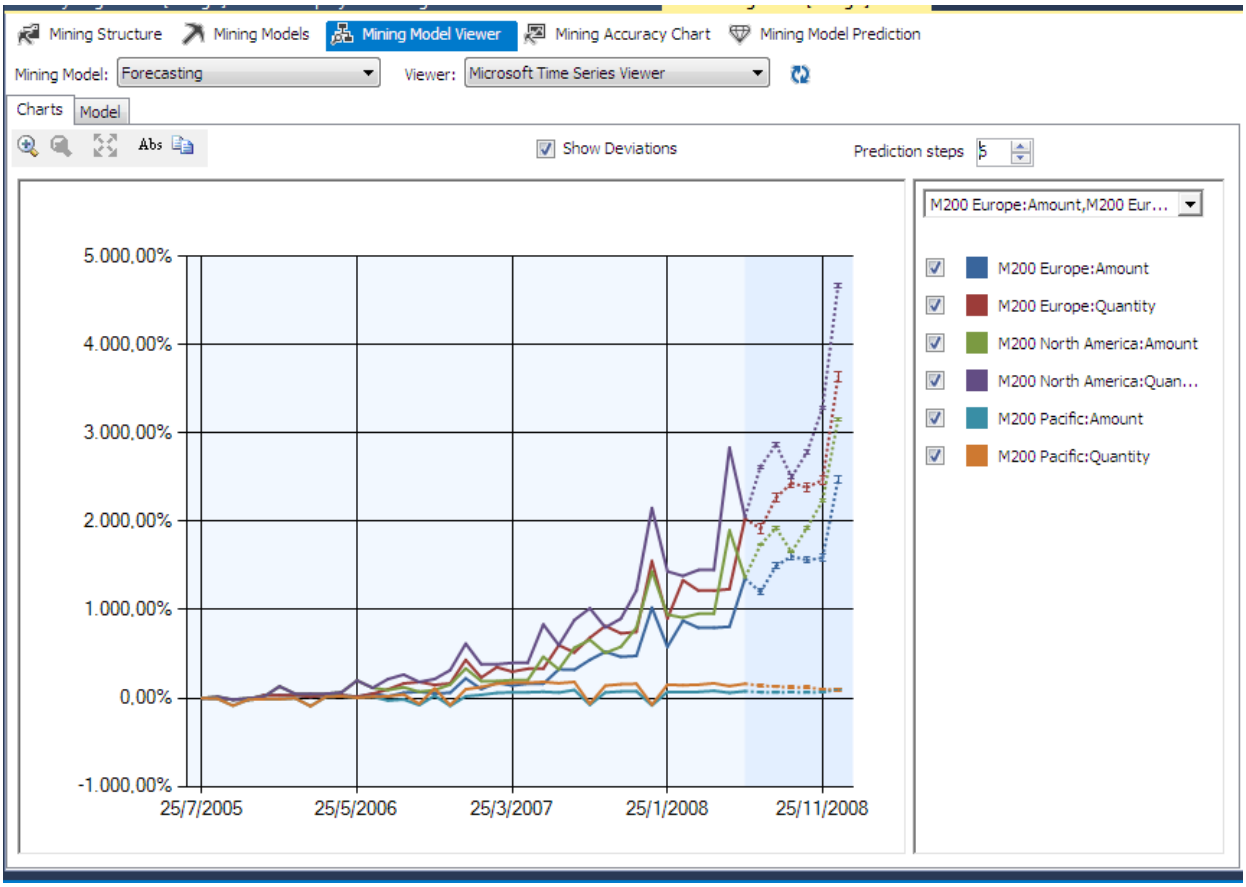
Εικόνα 10.25

10.4. Αξιολόγηση του μοντέλου Time Series.

Το μοντέλο πρόβλεψης που δημιουργήθηκε στην προηγούμενη ενότητα προβλέπει, όπως αναφέραμε, μελλοντικές πωλήσεις βάσει των πωλήσεων ποδηλάτων που έγιναν σε τρεις διαφορετικές περιοχές (Europe, North America και Pacific) για τα έτη 2005 - 2008. Μετά την επιτυχή δημιουργία του μοντέλου πρόβλεψης, μπορούμε να εξερευνήσουμε τα αποτελέσματα απ' την καρτέλα του Mining Model Viewer, η οποία περιέχει τις δύο καρτέλες Charts και Model.

10.4.1. Καρτέλα Charts

Η καρτέλα **Charts** εμφανίζει με γραφικό τρόπο μελλοντικές προβλέψεις για τα πεδία amount και/ή quantity με βάση το προϊόν και την εκάστοτε περιοχής πώλησής του. Όπως φαίνεται στην Εικόνα 10.26, το γράφημα εμφανίζει τόσο ιστορικά δεδομένα όσο και δεδομένα μελλοντικής πρόβλεψης. Συγκεκριμένα, το διάγραμμα εμφανίζει στοιχεία για το ποδήλατο με κωδικό προϊόντος M200 για τις περιοχές Europe, North America και Pacific, τόσο για το πεδίο amount όσο και για το πεδίο quantity. Οι καμπύλες τάσης δείχνουν ότι οι συνολικές πωλήσεις για όλες τις περιοχές γενικά (εκτός της περιοχής του Ειρηνικού) αυξάνονται κάθε 12 μήνες, και συγκεκριμένα τον μήνα Δεκέμβριο. Αυξάνοντας την επιλογή Prediction Steps, μπορούμε να αυξήσουμε τον χρονικό ορίζοντα των προβλέψεών μας.



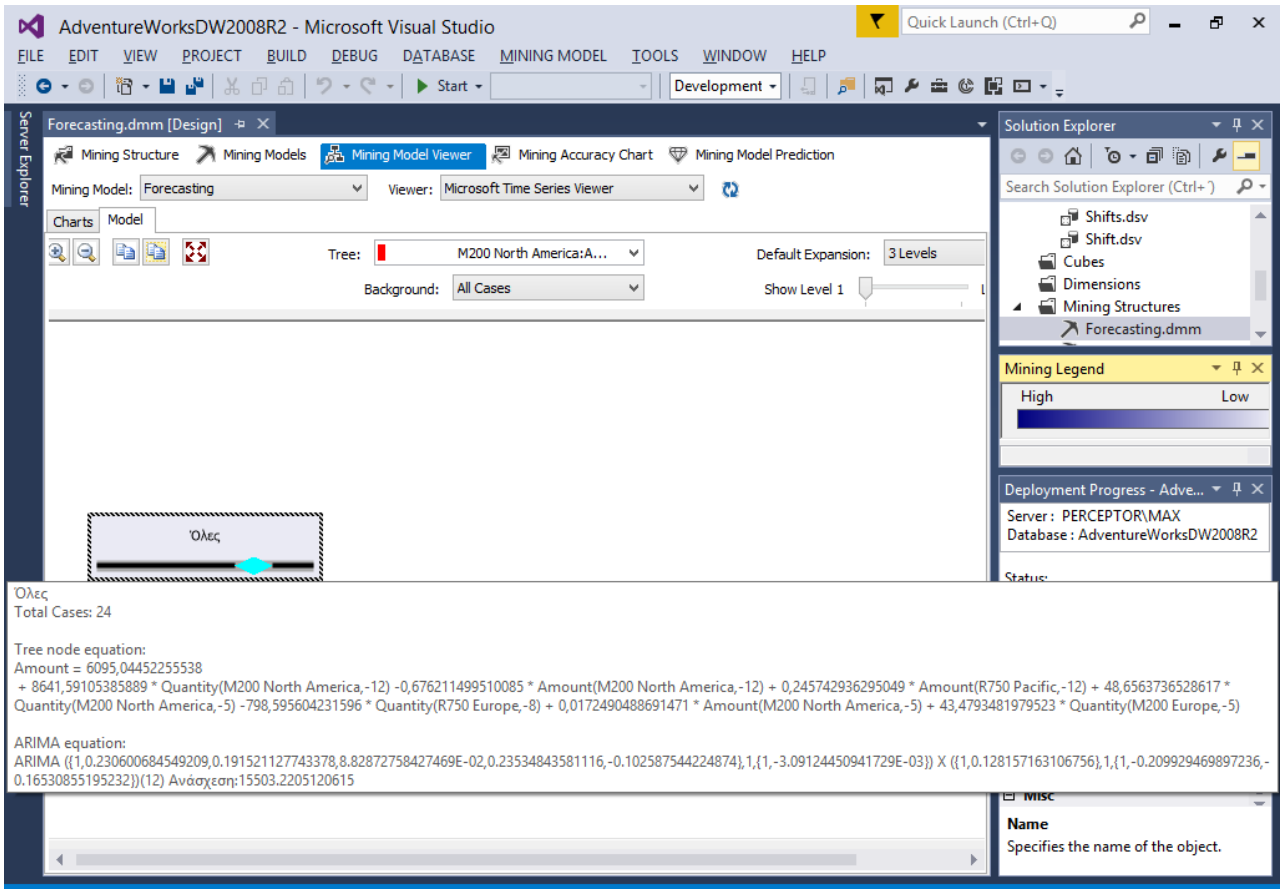
Εικόνα 10.26

10.4.2. Καρτέλα Model

Η καρτέλα **Model** εμφανίζει το μοντέλο πρόβλεψης με τη μορφή ενός δέντρου απόφασης. Συγκεκριμένα, εμφανίζεται ένα δέντρο για κάθε δυνατό συνδυασμό των τριών παραγόντων που υπάρχουν στα δεδομένα μας (κωδικός προϊόντος, περιοχή πώλησης και χαρακτηριστικό πρόβλεψης). Καθώς έχουμε ως δεδομένα τους τέσσερεις κωδικούς προϊόντων (M200, T1000, R250 και R750), τις τρεις περιοχές πώλησης (Europe, North America και Pacific) και τα δύο χαρακτηριστικά πρόβλεψης (Amount και Quantity), δημιουργούνται συνολικά 24 δέντρα απόφασης (4*3*2=24).

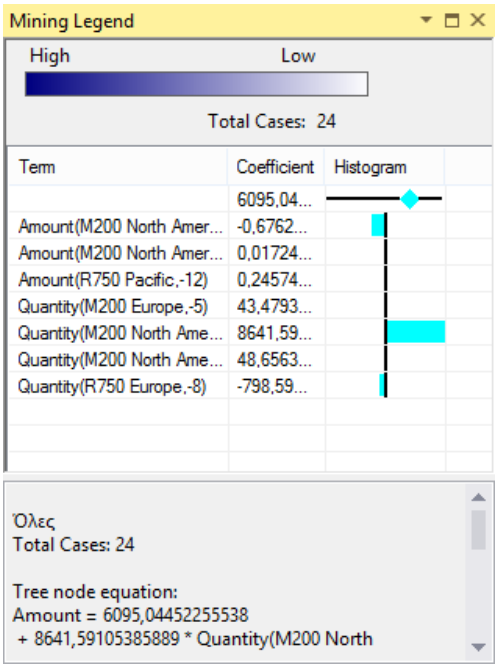
Τονίζεται ότι όταν ένα δέντρο απόφασης αποτελείται μόνο από έναν κόμβο, αυτό σημαίνει ότι η τάση των δεδομένων είναι ομοιογενής στη μονάδα του χρόνου και η χρονοσειρά μπορεί να εκφραστεί απλά με μία μόνο γραμμική εξίσωση. Από την άλλη, όταν ένα δέντρο απόφασης αποτελείται από πολλούς κόμβους και διακλαδώσεις, αυτό σημαίνει ότι η χρονοσειρά δεν είναι γραμμική και κάθε κλαδί του δέντρου πρέπει να εκφραστεί με μια διαφορετική εξίσωση.

1. Στην περίπτωση μας, επιλέγουμε την καρτέλα Model στο Mining Model Viewer και διαλέγουμε το M200 North America: Amount, όπως φαίνεται στην Εικόνα 10.27. Αυτό εμφανίζει έναν μόνο κόμβο. Αφήνοντας τον κέρσορα πάνω στον κόμβο, βλέπουμε πληροφορίες, όπως είναι ο αριθμός των περιπτώσεων που υπάρχουν στη χρονοσειρά και η εξίσωση χρονοσειράς που προκύπτει από την ανάλυση αυτών των δεδομένων.



Εικόνα 10.27

2. Κάνουμε δεξί κλικ στον κόμβο και επιλέγουμε Mining Legend. Το Mining Legend, όπως φαίνεται στην Εικόνα 10.28, περιλαμβάνει πληροφορίες για τις ανεξάρτητες μεταβλητές και τους συντελεστές που συμμετέχουν στην εξίσωση πρόβλεψης της χρονοσειράς για το M200 North America: Amount**.** Επιπλέον, το Mining Legend περιέχει ένα ιστόγραμμα που αφορά τις μεταβλητές που παίρνουν διακριτές τιμές και δείχνει την κατανομή των τιμών τους μέσα στο κόμβο..



Εικόνα 10.28

10.5. Ασκήσεις αξιολόγησης μοντέλου Time Series

1. Να αξιολογήσετε τις τάσεις πωλήσεων που προκύπτουν απ' το μοντέλο πρόβλεψης Time Series για τη χρονοσειρά που αφορά το προϊόν R250 στην περιοχή πώλησης Europe όσον αφορά το ποσό των πωλήσεων (amount).

2. Να δημιουργήσετε ένα ερώτημα SQL που να προβλέπει το ποσό των πωλήσεων σε δολάρια και τις ποσότητες των ποδηλάτων που θα πωληθούν κατά το διάστημα 25/7/2008 έως 25/11/2008, δηλαδή τους 5 μήνες που έπονται των δεδομένων της χρονοσειράς μας. Η πρόβλεψη να γίνει για κάθε δυνατό συνδυασμό ενός τύπου προϊόντος (M200, T1000, R250 και R750) και μιας περιοχής πώλησης (Europe, North America και Pacific).

3. Να συγκρίνετε τις τάσεις πωλήσεων μεταξύ των ποδηλάτων με κωδικό προϊόντος M200 και να προσδιορίσετε πιθανά προβλήματα.

4. Να δημιουργήσετε ένα γενικό μοντέλο πρόβλεψης που να μην αφορά μια μεμονωμένη περιοχή πώλησης (Europe, North America και Pacific) και ένα συγκεκριμένο κωδικό ποδηλάτου (M200, T1000, R250, R750) Αντιθέτως, να προβλέπει συγκεντρωτικά και σε παγκόσμιο επίπεδο τα συνολικά ποσά πωλήσεων, καθώς και τις συνολικές ποσότητες ποδηλάτων που θα πωληθούν παγκοσμίως.

5. Έστω ότι εργάζεστε για την AdventureWorks, μια πολυεθνική εταιρία που εμπορεύεται τέσσερις τύπους ποδηλάτων (M200, R250, R750 και T1000) σε τρεις περιοχές (Ευρώπη, Βόρεια Αμερική και Ειρηνικό). Το τμήμα πωλήσεων επιθυμεί να προβλέψει τις πωλήσεις του επόμενου εξαμήνου (Ιανουάριος 2008 έως Ιούνιος 2008) για το μοντέλο ποδηλάτου R750 στις τρεις παραπάνω περιοχές, λαμβάνοντας υπόψη τις πωλήσεις που σημειώθηκαν στο προγενέστερο διάστημα (Ιούλιος 2005 έως Δεκέμβριος του 2007). Δημιουργήστε, λοιπόν, ένα μοντέλο χρονοσειράς που θα έχει ως input και predictable το πεδίο amount, θέτοντας τις παραμέτρους του αλγορίθμου ως εξής: PERIODICITY_HINT = {12} και FORECAST_METHOD=MIXED. Τονίζεται ότι θα πρέπει να δημιουργήσετε ένα νέο ερώτημα (Data Source View & New named query) που θα επιλέγει δεδομένα μόνο μέχρι τις 31-12-2007. Ακόμη, τονίζεται ότι πρέπει να τρέξετε τον αλγόριθμο time series μόνο στο συγκεκριμένο χρονικό διάστημα τιμών.

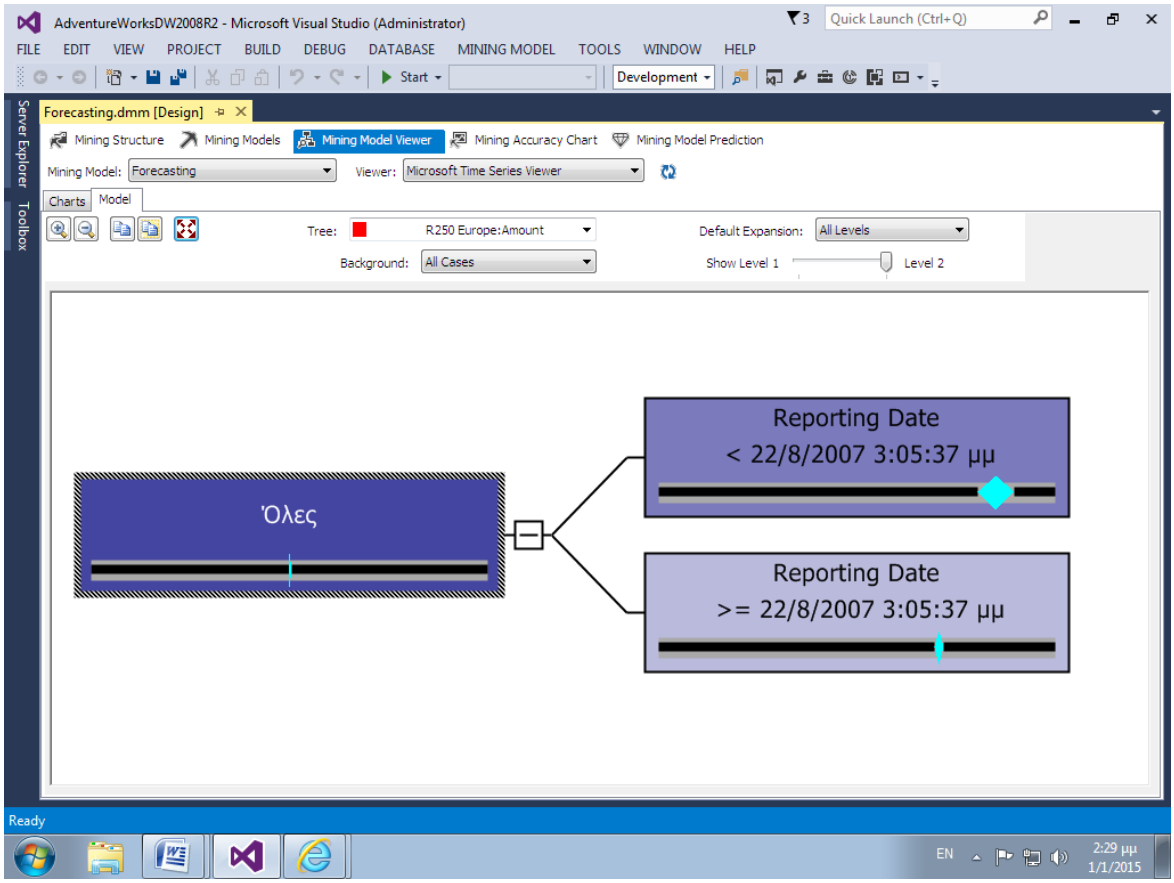
10.6. Λύσεις ασκήσεων αξιολόγησης μοντέλου Time Series

Άσκηση 1

Να αξιολογήσετε τις τάσεις πωλήσεων που προκύπτουν απ' το μοντέλο πρόβλεψης Time Series για τη χρονοσειρά που αφορά το προϊόν R250 στην περιοχή πώλησης Europe όσον αφορά το ποσό των πωλήσεων (amount).

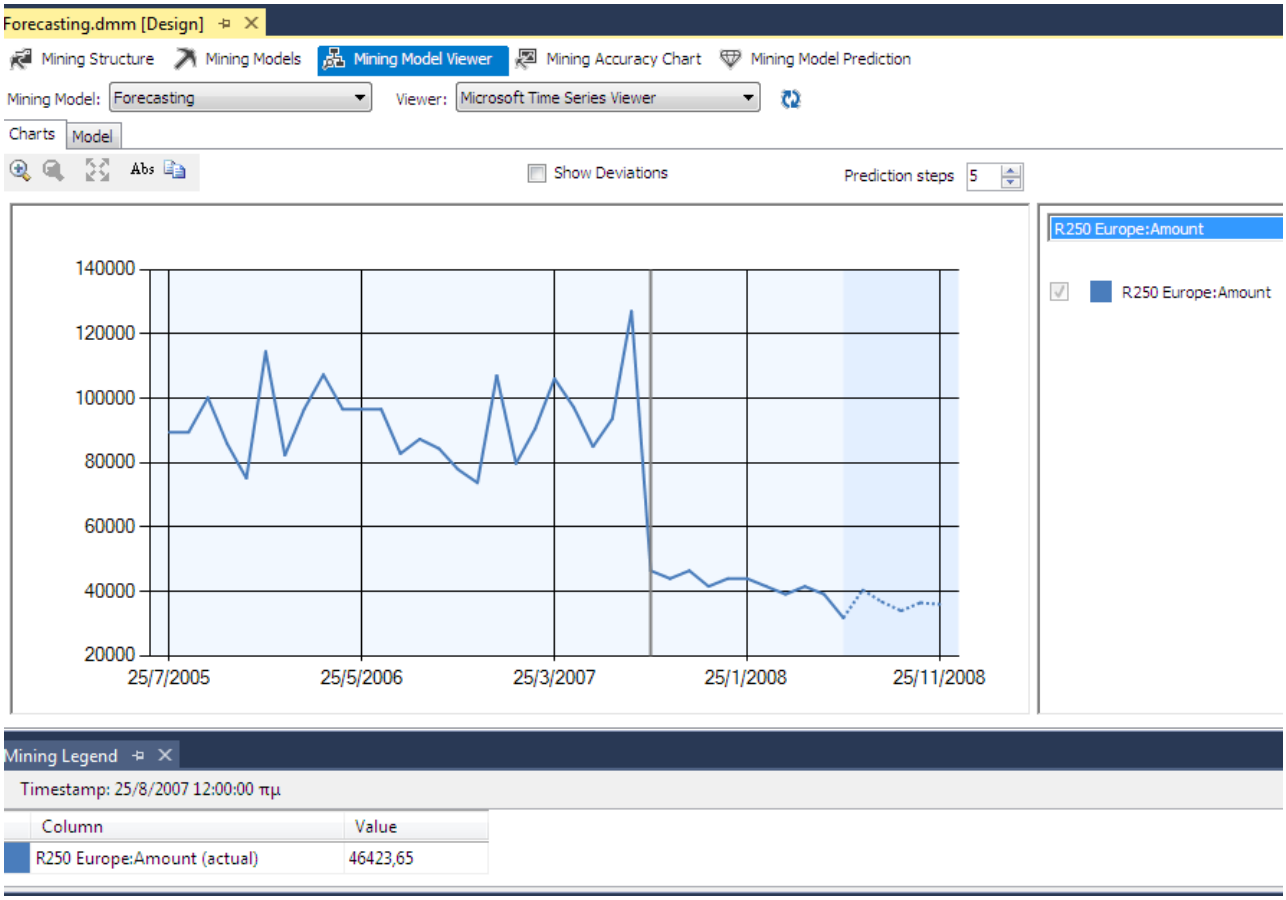
Λύση

1. Βρισκόμαστε στην καρτέλα Model του Mining Model Viewer. Επιλέγουμε τον συνδυασμό προϊόντος και περιοχής R250 Europe: Amount. Αυτή η ενέργεια εμφανίζει έναν κόμβο που σε κάποια χρονική στιγμή διασπάται σε δύο, κάτι που σημαίνει ότι έχει υπάρξει μια δραστική αλλαγή στη χρονοσειρά. Όπως φαίνεται στην Εικόνα 10.29, η ημερομηνία 22/8/2007 έχει προσδιοριστεί ως ορόσημο για μια σημαντική αλλαγή στα δεδομένα της χρονοσειράς.



Εικόνα 10.29

2. Βρισκόμαστε στην καρτέλα Charts του Mining Model Viewer. Επιλέγουμε το προϊόν R250 για την περιοχή Europe και προβλέπουμε το χαρακτηριστικό amount. Όπως φαίνεται στην Εικόνα 10.30, στις 25/8/2007 οι πωλήσεις του προϊόντος R250 έπεσαν δραματικά: από 127054 $ στα 46423 $. Δηλαδή, σημειώθηκε μια πτώση των πωλήσεων κατά τρείς φορές σε σχέση με την προτεραία κατάσταση. Ασφαλώς, ένα τέτοιο συμβάν θα πρέπει να αντιμετωπιστεί από το τμήμα πωλήσεων!



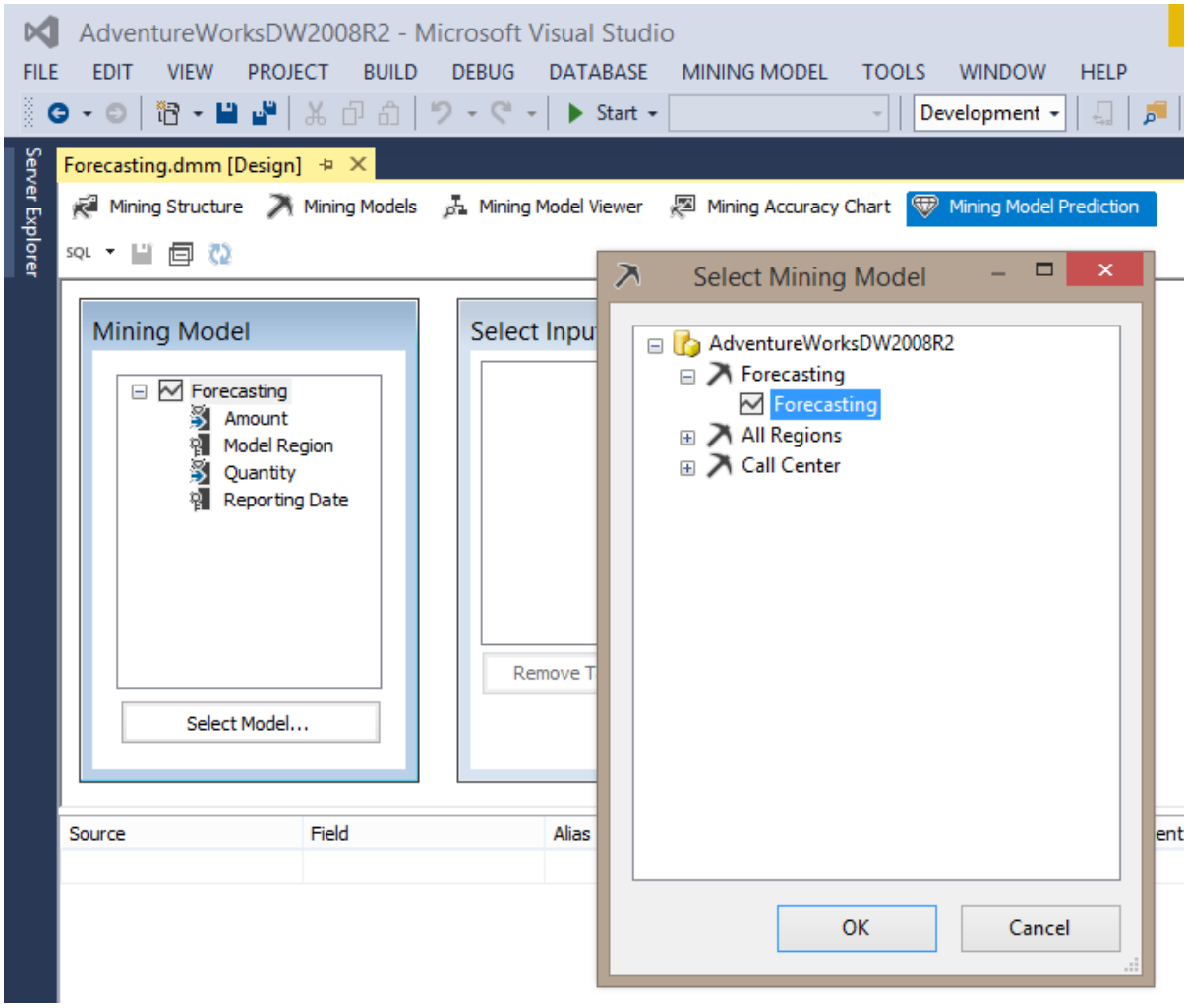
Εικόνα 10.30

Άσκηση 2

Να δημιουργήσετε ένα ερώτημα SQL που να προβλέπει το ποσό των πωλήσεων σε δολάρια και τις ποσότητες των ποδηλάτων που θα πωληθούν κατά το διάστημα 25/7/2008 έως 25/11/2008, δηλαδή τους 5 μήνες που έπονται των δεδομένων της χρονοσειράς μας. Η πρόβλεψη να γίνει για κάθε δυνατό συνδυασμό ενός τύπου προϊόντος (M200, T1000, R250 και R750) και μιας περιοχής πώλησης (Europe, North America και Pacific).

Λύση

1. Για να δημιουργήσουμε το ερώτημα, ανοίγουμε την καρτέλα Mining Model Prediction, όπως φαίνεται στην Εικόνα 10.31. Στο Mining Model κάνουμε κλικ στο Select Model και επιλέγουμε το Forecasting. Στη συνέχεια, πατάμε OK.



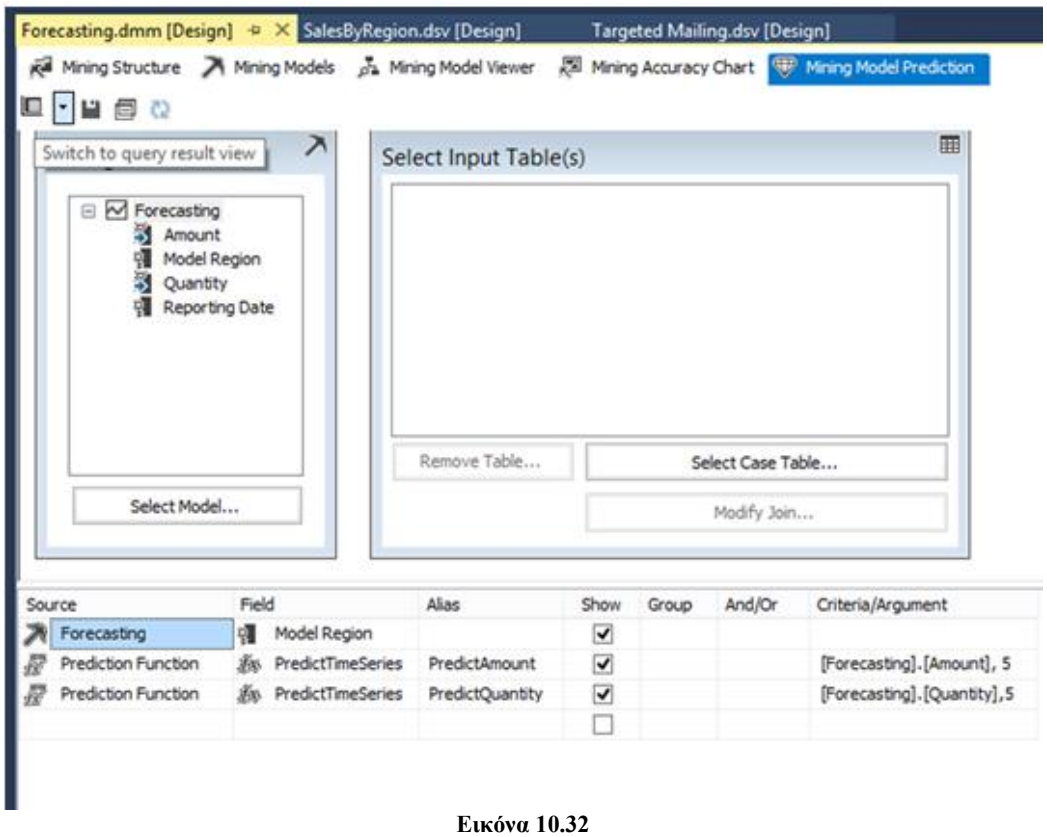
Εικόνα 10.31

2. Όπως φαίνεται στην Εικόνα 10.32, αγνοούμε το παράθυρο Select Input Table, διότι δεν έχουμε να εισάγουμε δεδομένα από άλλους συσχετιζόμενους πίνακες. Αντίθετα, ενδιαφερόμαστε για τον πίνακα με τις κενές γραμμές που βρίσκεται στην κάτω περιοχή του παραθύρου..

   Ξεκινάμε από την πρώτη γραμμή του πίνακα. Στη στήλη Source κάνουμε κλικ και επιλέγουμε Forecasting, επειδή αυτό είναι το μοντέλο για την πρόβλεψή μας. Στη στήλη Field επιλέγουμε Model Region, προκειμένου να ομαδοποιούνται τα δεδομένα μας ανά κωδικό προϊόντος και περιοχή πώλησης.

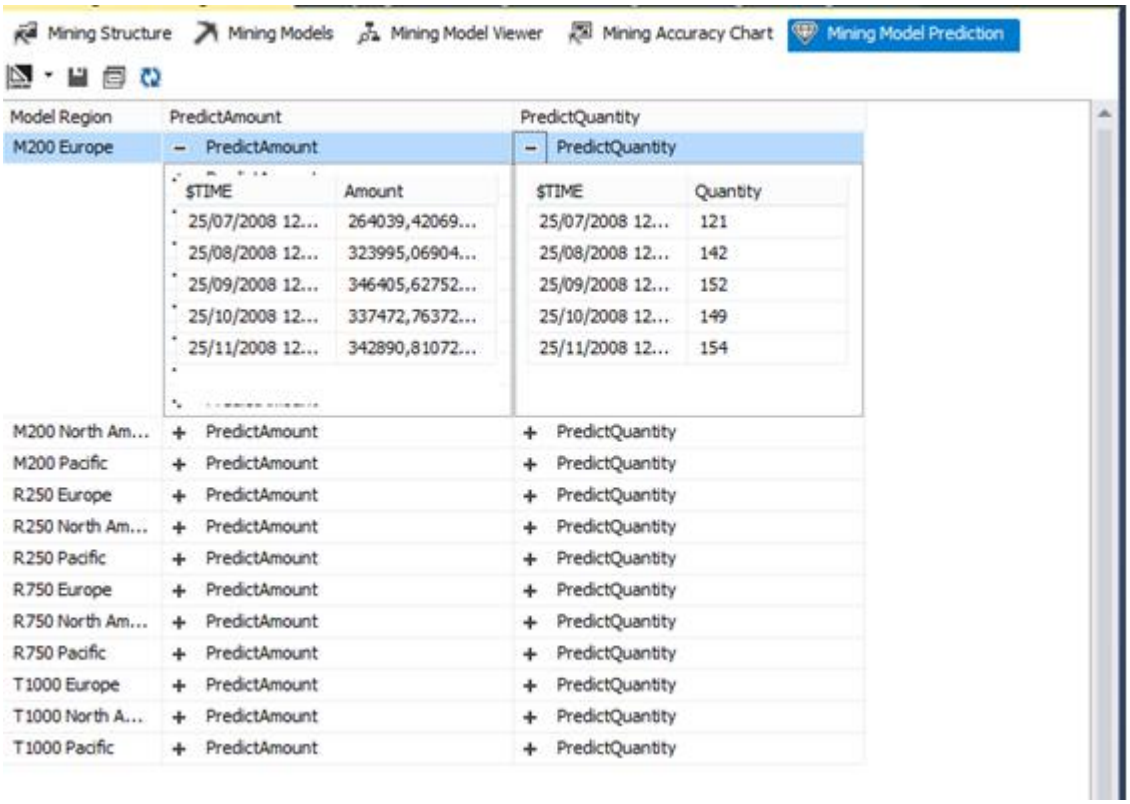
   Συνεχίζουμε στη δεύτερη γραμμή του πίνακα. Στη στήλη Source επιλέγουμε Prediction Function, στη στήλη Field επιλέγουμε PredictTimeSeries (μια ενσωματωμένη συνάρτηση του SQL Server) και στη στήλη Alias πληκτρολογούμε PredictAmount, που θα είναι το όνομα του πεδίου που θα προβλέπουμε. Στη στήλη Criteria/Arguments γράφουμε "[Forecasting].[Amount], 5", προκειμένου να προβλέψουμε το ποσό των πωλήσεων σε δολάρια για τους επόμενους 5 μήνες (25/7/2008 έως 25/11/2008) που έπονται των δεδομένων της χρονοσειράς μας.

   Συμπληρώνουμε, τέλος, την τρίτη γραμμή του πίνακα. Στη στήλη Source επιλέγουμε Prediction Function, στη στήλη Field επιλέγουμε PredictTimeSeries και στη στήλη Alias πληκτρολογούμε PredictQuantity, που θα είναι το όνομα του πεδίου που θα προβλέπουμε. Στη στήλη Criteria/Arguments γράφουμε "[Forecasting].[Quantity], 5", προκειμένου να προβλέψουμε τις ποσότητες των ποδηλάτων που θα πωληθούν τους επόμενους 5 μήνες (25/7/2008 έως 25/11/2008) που έπονται των δεδομένων της χρονοσειράς μας. Τέλος, κάνουμε κλικ στο Switch to query result view (το κουμπί πάνω αριστερά στην Εικόνα 10.32), για να εμφανιστούν τα αποτελέσματα του ερωτήματος.



Εικόνα 10.32

3. Όπως φαίνεται στην Εικόνα 10.33, τα αποτελέσματα του ερωτήματός μας περιέχονται σε τρεις ξεχωριστές στήλες. Η πρώτη στήλη περιέχει όλους τους δυνατούς συνδυασμούς των προϊόντων και της περιοχής πώλησης. Η δεύτερη στήλη περιέχει τις πέντε προβλεπόμενες τιμές ποσών πωλήσεων ανά κωδικό προϊόντος και περιοχή πώλησης. Τέλος, η τρίτη στήλη περιέχει τις πέντε προβλεπόμενες ποσότητες ποδηλάτων ανά κωδικό προϊόντος και περιοχή πώλησης. Τελικά, στην Εικόνα 10.33 παρουσιάζονται οι αναλυτικές προβλέψεις μας για το προϊόν 200 Europe.



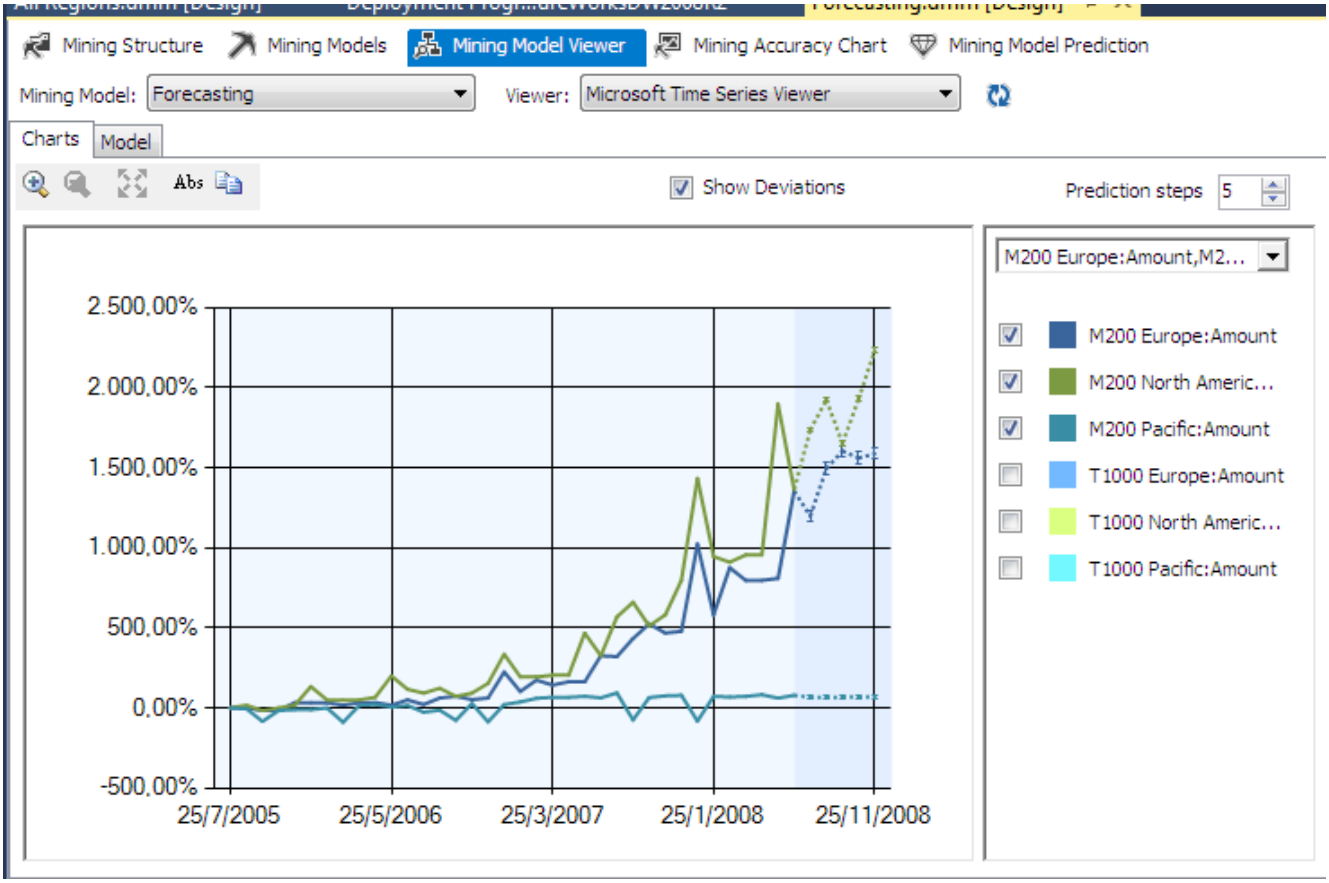
Εικόνα 10.33

Άσκηση 3

Να συγκρίνετε τις τάσεις πωλήσεων μεταξύ των ποδηλάτων με κωδικό προϊόντος M200 και να προσδιορίσετε πιθανά προβλήματα.

Λύση

Στην Εικόνα 10.34 εμφανίζονται οι τάσεις πωλήσεων του ποδηλάτου M200 στις τρεις περιοχές πώλησης. Οι καμπύλες πρόβλεψης πωλήσεων για το μοντέλο M200 στις περιοχές της Ευρώπης και της Βόρειας Αμερικής είναι σταθερά αυξανόμενες. Αντιθέτως, η καμπύλη πρόβλεψης των πωλήσεων για το M200 στην περιοχή του Ειρηνικού είναι χαμηλή και σχετικά επίπεδη. Προφανώς, το τμήμα πωλήσεων θα πρέπει να προσέξει ιδιαίτερα το μεγάλο χάσμα που εμφανίζεται στις προβλέψεις πωλήσεων μεταξύ των τριών περιοχών στο ίδιο μοντέλο ποδηλάτου.



Εικόνα 10.34

Άσκηση 4

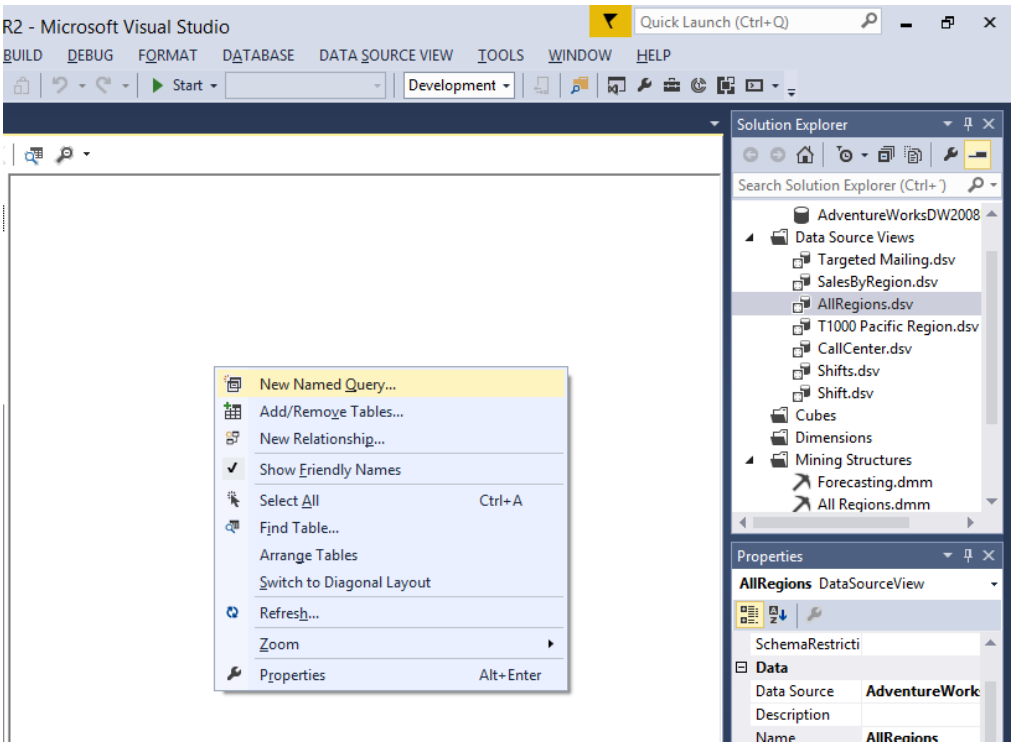
Να δημιουργήσετε ένα γενικό μοντέλο πρόβλεψης που να μην αφορά μια μεμονωμένη περιοχή πώλησης (Europe, North America και Pacific) και ένα συγκεκριμένο κωδικό ποδηλάτου (M200, T1000, R250, R750) Αντιθέτως, να προβλέπει συγκεντρωτικά και σε παγκόσμιο επίπεδο τα συνολικά ποσά πωλήσεων, καθώς και τις συνολικές ποσότητες ποδηλάτων που θα πωληθούν.

Λύση

Το πρώτο βήμα για τη δημιουργία του γενικευμένου μοντέλου είναι η συγκέντρωση των στοιχείων για τις πωλήσεις σ' όλον τον κόσμο. Μπορούμε να το κάνουμε αυτό, δημιουργώντας μια νέα προβολή προέλευσης δεδομένων (data source view) που θα εκτελέσει συγκεντρωτικούς υπολογισμούς **(sums** ή **averages)** πάνω στα πεδία ammount και quantity.

1. Για να δημιουργήσουμε μια νέα προβολή προέλευσης δεδομένων, στο Solution Explorer επιλέγουμε με δεξί κλικ New Data Source View. Στην επιλογή Select Tables and Views κάνουμε κλικ στο Next χωρίς να επιλέξουμε κάποιον πίνακα. Το Data Source View το ονομάζουμε AllRegions.

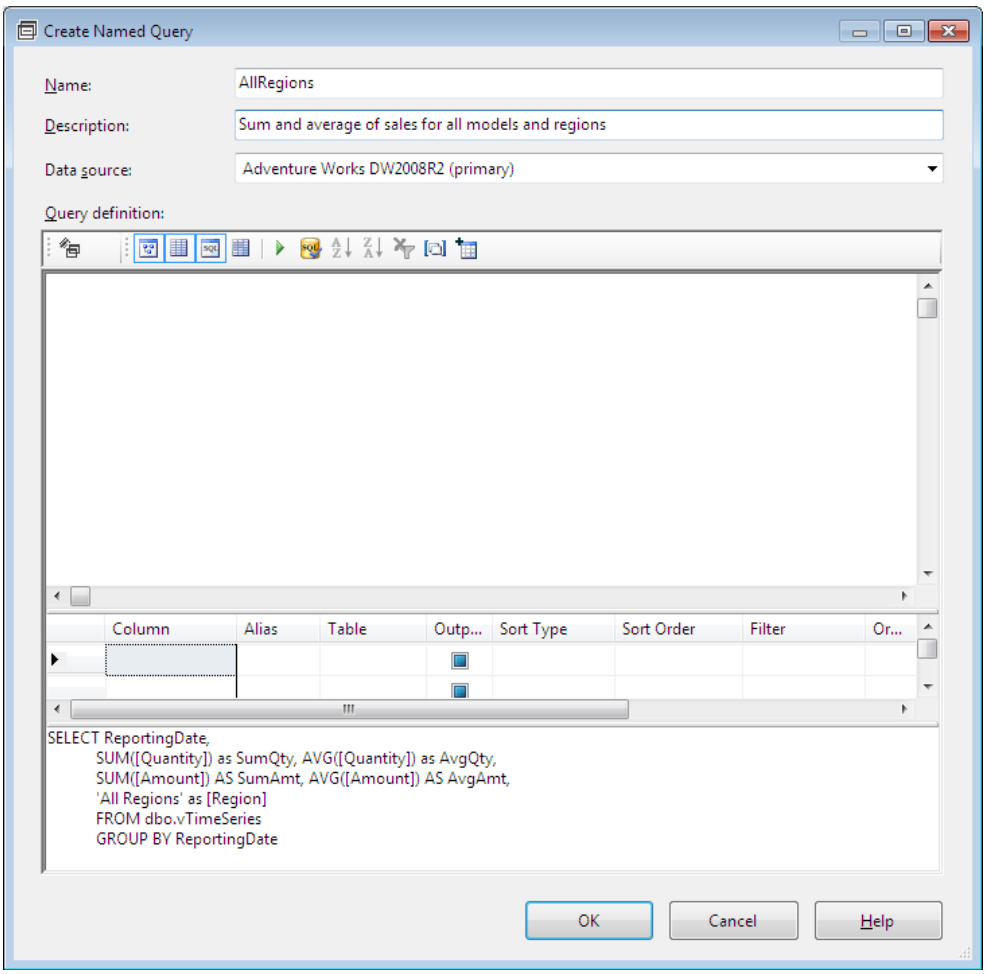
2. Μετά τη δημιουργία του, κάνουμε δεξί κλικ στην κενή προβολή δεδομένων σχεδίασης, όπως φαίνεται στην Εικόνα 10.35, και επιλέγουμε New Named Query.



Εικόνα 10.35

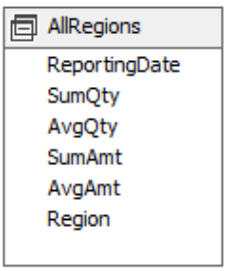
3. Στο παράθυρο Create Named Query, όπως φαίνεται στην Εικόνα 10.36, γράφουμε AllRegions στο πεδίο Name. Στη συνέχεια, γράφουμε Sum and average of sales for all models and regions στο πεδίο Description. Τέλος, στο SQL query editor (στο κάτω μέρος του παραθύρου) γράφουμε το παρακάτω ερώτημα ομαδοποίησης:

```
SELECT ReportingDate,
SUM([Quantity]) as SumQty, AVG([Quantity]) as AvgQty,
SUM([Amount]) AS SumAmt, AVG([Amount]) AS AvgAmt,
'All Regions' as [Region]
FROM dbo.vTimeSeries
GROUP BY ReportingDate
```



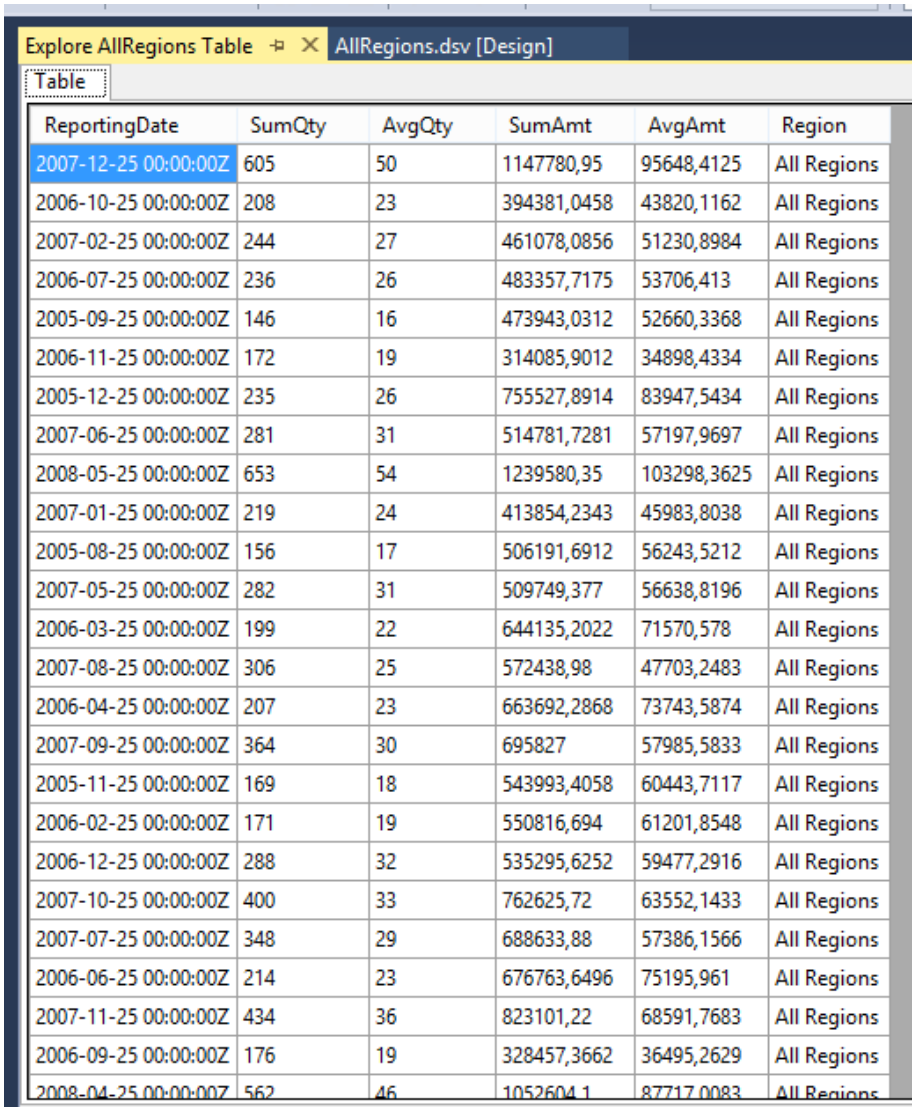
Εικόνα 10.36

4. Στο παραπάνω παράθυρο πατάμε OK, οπότε δημιουργείται το Data Source View, όπως φαίνεται στην Εικόνα 10.37.



Εικόνα 10.37

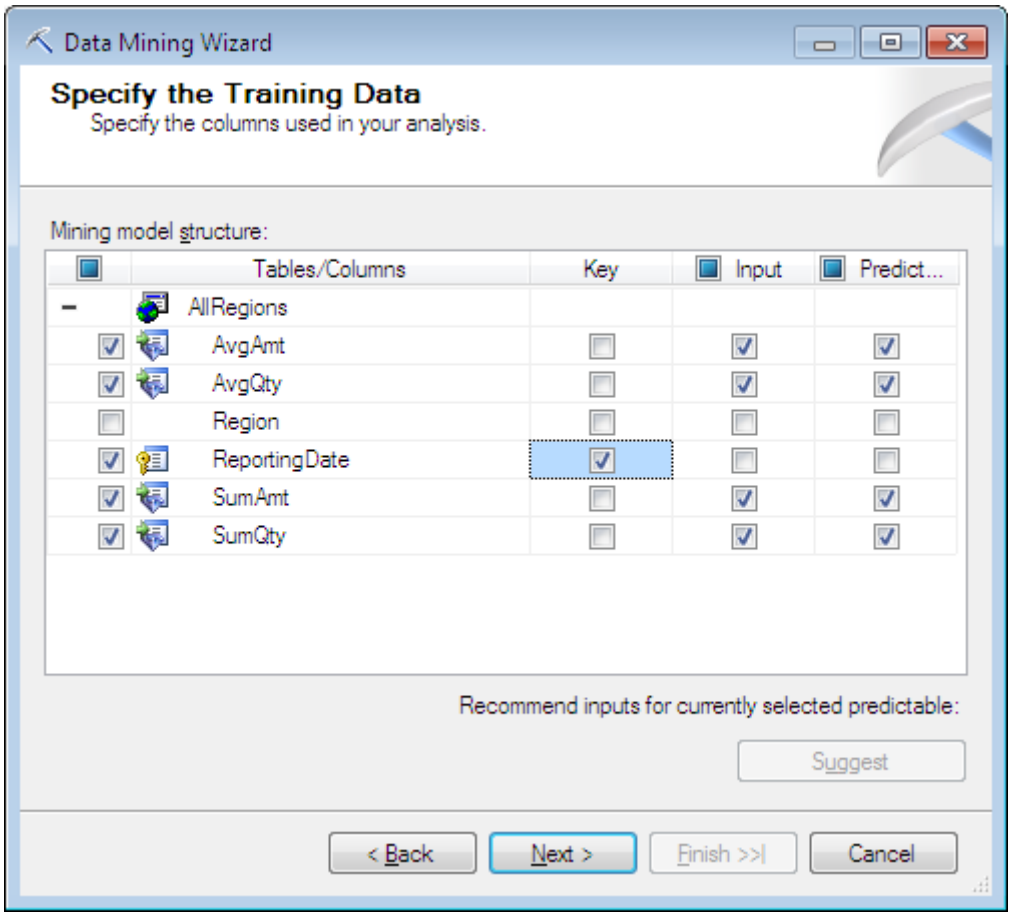
5. Στη συνέχεια, κάνουμε δεξί κλικ στην όψη AllRegions και επιλέγουμε Explore Data. Όπως φαίνεται στην Εικόνα 10.38, η νέα προβολή προέλευσης δεδομένων περιέχει συγκεντρωτικά αθροίσματα και μέσες τιμές για τις συνολικές πωλήσεις και τη συνολική ποσότητα ποδηλάτων που πουλήθηκε, για όλες τις περιοχές (Στήλες SumQty, AvgQty, SumAmt, AvgAmt) ανά χρονική περίοδο.



ReportingDate	SumQty	AvgQty	SumAmt	AvgAmt	Region
2007-12-25 00:00:00Z	605	50	1147780,95	95648,4125	All Regions
2006-10-25 00:00:00Z	208	23	394381,0458	43820,1162	All Regions
2007-02-25 00:00:00Z	244	27	461078,0856	51230,8984	All Regions
2006-07-25 00:00:00Z	236	26	483357,7175	53706,413	All Regions
2005-09-25 00:00:00Z	146	16	473943,0312	52660,3368	All Regions
2006-11-25 00:00:00Z	172	19	314085,9012	34898,4334	All Regions
2005-12-25 00:00:00Z	235	26	755527,8914	83947,5434	All Regions
2007-06-25 00:00:00Z	281	31	514781,7281	57197,9697	All Regions
2008-05-25 00:00:00Z	653	54	1239580,35	103298,3625	All Regions
2007-01-25 00:00:00Z	219	24	413854,2343	45983,8038	All Regions
2005-08-25 00:00:00Z	156	17	506191,6912	56243,5212	All Regions
2007-05-25 00:00:00Z	282	31	509749,377	56638,8196	All Regions
2006-03-25 00:00:00Z	199	22	644135,2022	71570,578	All Regions
2007-08-25 00:00:00Z	306	25	572438,98	47703,2483	All Regions
2006-04-25 00:00:00Z	207	23	663692,2868	73743,5874	All Regions
2007-09-25 00:00:00Z	364	30	695827	57985,5833	All Regions
2005-11-25 00:00:00Z	169	18	543993,4058	60443,7117	All Regions
2006-02-25 00:00:00Z	171	19	550816,694	61201,8548	All Regions
2006-12-25 00:00:00Z	288	32	535295,6252	59477,2916	All Regions
2007-10-25 00:00:00Z	400	33	762625,72	63552,1433	All Regions
2007-07-25 00:00:00Z	348	29	688633,88	57386,1566	All Regions
2006-06-25 00:00:00Z	214	23	676763,6496	75195,961	All Regions
2007-11-25 00:00:00Z	434	36	823101,22	68591,7683	All Regions
2006-09-25 00:00:00Z	176	19	328457,3662	36495,2629	All Regions
2008-04-25 00:00:00Z	562	46	1052604,1	87717,0083	All Regions

Εικόνα 10.38

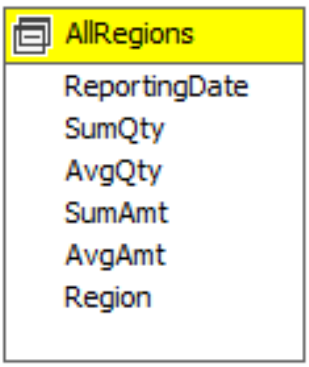
6. Για να δημιουργήσουμε ένα μοντέλο πρόβλεψης για τα νέα συγκεντρωτικά στοιχεία, θα πρέπει να δημιουργήσουμε ένα New Mining Structure, όπως προαναφέρθηκε. Στο Data Mining Wizard επιλέγουμε τον αλγόριθμο Microsoft Time Series, στο Data source view το AllRegion, και επιλέγουμε το AllRegions για Case. Επίσης, όπως φαίνεται στην Εικόνα 10.39, επιλέγουμε ως κλειδί (key) το ReportingDate, και ως Input και Predict τα πεδία AvgAmt, AvgQty, SumAmt, και SumQty. Κατόπιν, πατάμε Next.



Εικόνα 10.39

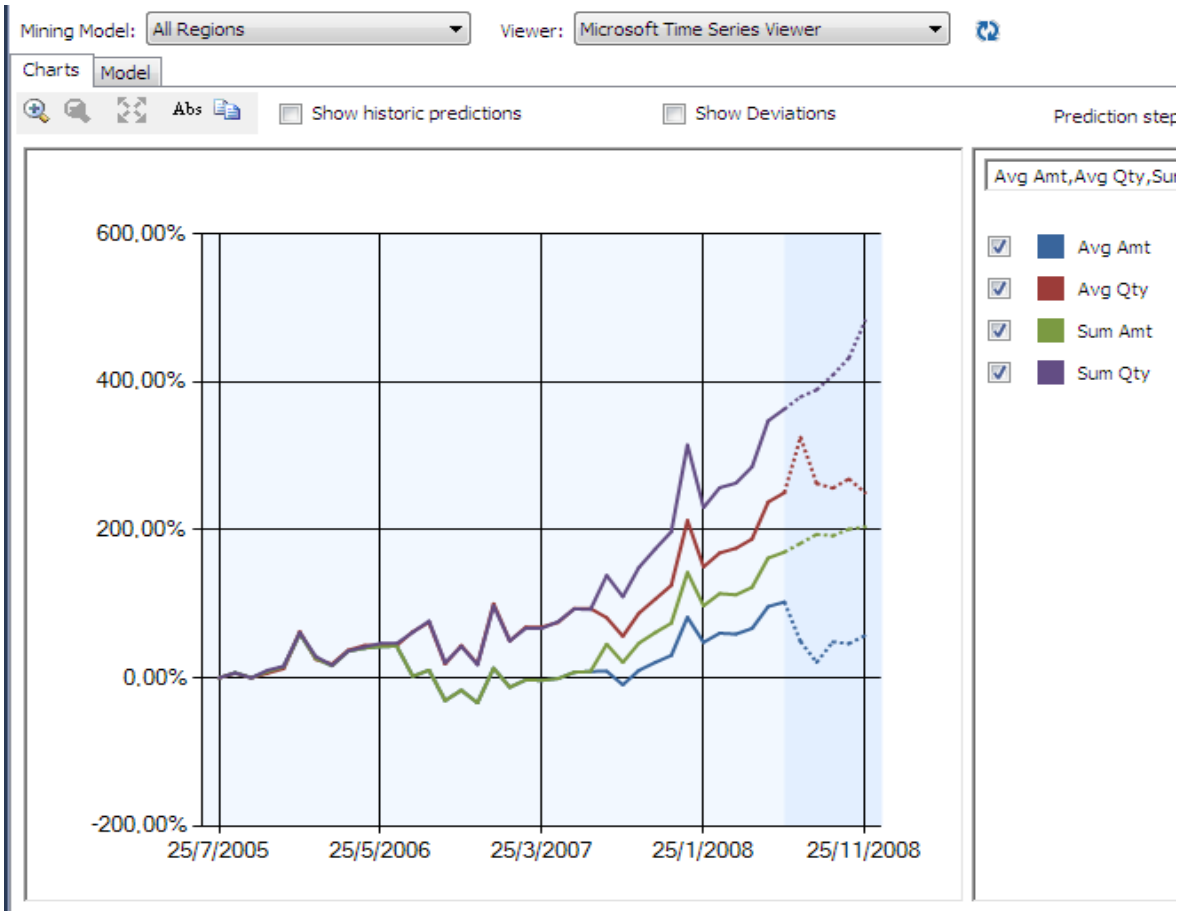
7. Στη συνέχεια, θα ονομάσουμε All Regions τόσο το Mining structure name όσο και το Mining model name.

8. Κατόπιν, κάνουμε Process the structure and the model και εμφανίζεται ο πίνακας της Εικόνας 10.40.



Εικόνα 10.40

9. Στη συνέχεια, πηγαίνουμε στην καρτέλα Charts, όπως φαίνεται στο διάγραμμα της Εικόνας 10.41. Βλέπουμε σ' αυτό τις προβλέψεις του αλγόριθμου Time Series για τα συγκεντρωτικά δεδομένα. Συγκεκριμένα, οι τέσσερις καμπύλες δείχνουν τις μέσες τιμές ποσών πώλησης και ποσοτήτων ποδηλάτων, καθώς επίσης και τα συνολικά εκτιμώμενα ποσά πωλήσεων και ποσότητες ποδηλάτων.



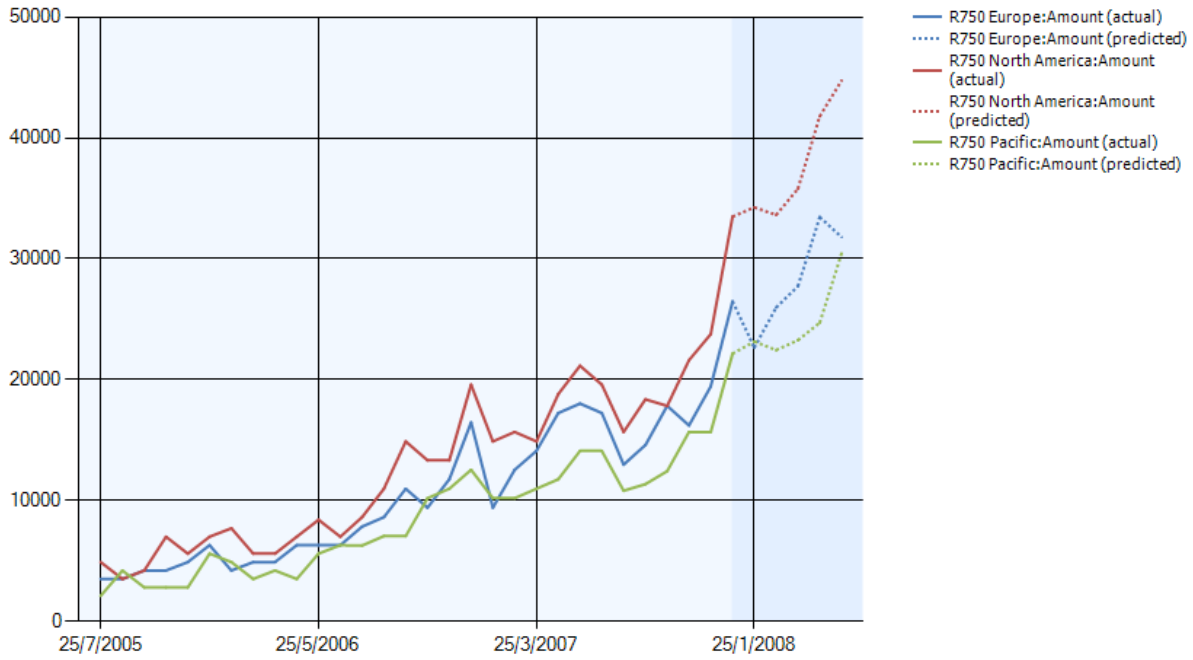
Εικόνα 10.41

Άσκηση 5

Έστω ότι εργάζεστε για την AdventureWorks, μια πολυεθνική εταιρία που εμπορεύεται τέσσερις τύπους ποδηλάτων (M200, R250, R750 και T1000) σε τρεις περιοχές (Ευρώπη, Βόρεια Αμερική και Ειρηνικό). Το τμήμα πωλήσεων επιθυμεί να προβλέψει τις πωλήσεις του επόμενου εξαμήνου (Ιανουάριος 2008 έως Ιούνιος 2008) για το μοντέλο ποδηλάτου R750 στις τρεις παραπάνω περιοχές, λαμβάνοντας υπόψη τις πωλήσεις που σημειώθηκαν στο προγενέστερο διάστημα (Ιούλιος 2005 έως Δεκέμβριος του 2007). Δημιουργήστε, λοιπόν, ένα μοντέλο χρονοσειράς που θα έχει ως input και predictable το πεδίο amount, θέτοντας τις παραμέτρους του αλγορίθμου ως εξής: PERIODICITY_HINT = {12} και FORECAST_METHOD=MIXED. Τονίζεται ότι θα πρέπει να δημιουργήσετε ένα νέο ερώτημα (Data Source View & New named query) που θα επιλέγει δεδομένα μόνο μέχρι τις 31-12-2007. Ακόμη, τονίζεται ότι πρέπει να τρέξετε τον αλγόριθμο time series μόνο στο συγκεκριμένο χρονικό διάστημα τιμών.

Λύση

Στην Εικόνα 10.42 παρουσιάζεται ένα διάγραμμα με την πρόβλεψη των πωλήσεων του ποδηλάτου R750 στην Ευρώπη, τη Βόρεια Αμερική και τον Ειρηνικό Ωκεανό για το χρονικό διάστημα Ιανουάριος – Ιούνιος 2008. Οι διακεκομμένες γραμμές αντιστοιχούν στις προβλέψεις του μοντέλου χρονοσειρών Mixed. Για την πρόβλεψη αυτών χρησιμοποιήθηκαν δεδομένα για το συγκεκριμένο ποδήλατο μέχρι και τον Δεκέμβριο του 2007.



Εικόνα 10.42

10.7. Βιβλιογραφία/Αναφορές

Aggarwal, C. C. (2015). *Data Mining: The Textbook*, Springer.

Dunham, M. H. (2003). *Data Mining: Introductory and Advanced Topics*, New Jersey, Prentice Hall.

Han, J., & Kamber, M. (2001). *Data Mining: Concepts and Techniques*, Academic Press.

Κεφάλαιο 11. Αποθήκες και κύβοι δεδομένων

Σύνοψη

Σ' αυτό το κεφάλαιο θα παρουσιάσουμε τη δημιουργία μιας αποθήκης δεδομένων ή, αλλιώς, ενός κύβου δεδομένων. Ο κύβος είναι μια πολυδιάστατη δομή δεδομένων που εμπεριέχει συναθροιστικές πληροφορίες για μία ή περισσότερες βάσεις δεδομένων. Η συνάθροιση της πληροφορίας οδηγεί σε γρήγορους χρόνους απόκρισης ερωτημάτων που τίθενται από υψηλόβαθμα στελέχη επιχειρήσεων, προκειμένου αυτά να λάβουν συνήθως στρατηγικές αποφάσεις για την επιχείρηση.

11.1. Θεωρητικό υπόβαθρο για τους κύβους δεδομένων και την πολυδιάστατη ανάλυση

Τα ανώτερα στελέχη μιας επιχείρησης καλούνται συχνά να πάρουν στρατηγικές αποφάσεις (π.χ. την είσοδο της επιχείρησης σε μια νέα αγορά, τη δημιουργία ενός νέου προϊόντος κτλ.) για το μέλλον της επιχείρησης, βασιζόμενοι σ' έναν τεράστιο όγκο δεδομένων που αφορούν την επιχείρηση και το περιβάλλον της (Νανόπουλος, & Μανωλόπουλος, 2008· Χαλκίδη, & Βεζυργιάννης, 2005). Η **αποθήκη δεδομένων (data warehouse)** είναι μια πολυδιάστατη δομή, η οποία, καθώς περιέχει συναθροιστικές πληροφορίες που προέρχονται συνήθως από περισσότερες ετερογενείς βάσεις δεδομένων, βοηθά στη λήψη τέτοιων στρατηγικών αποφάσεων. Συγκεκριμένα, μια αποθήκη δεδομένων έχει τα εξής χαρακτηριστικά:

- **Είναι προσανατολισμένη σε ένα θέμα (subject-oriented)** και αφορά βασικές επιχειρηματικές διεργασίες, όπως είναι η διαχείριση πελατών, οι πωλήσεις κτλ.

- **Είναι ολοκληρωμένη (integrated)** και ενοποιεί στην ίδια μορφή τα δεδομένα ετερογενών βάσεων δεδομένων της επιχείρησης.

- **Δεν είναι ευμετάβλητη (non volatile)** και δεν έχουμε σ' αυτήν συχνά τη διαγραφή εγγραφών, παρά μόνο την προσθήκη νέων εγγραφών.

- **Αφορά ιστορικά δεδομένα (time-variant)** σε βάθος χρόνου, ακόμα και πολλών δεκαετιών.

Η αποθήκη δεδομένων εστιάζει σε επιχειρηματικές διεργασίες, στις οποίες λαμβάνουν χώρα διάφορα **γεγονότα** (facts). Για παράδειγμα, για μια αποθήκη δεδομένων που αφορά τις πωλήσεις προϊόντων, γεγονός αποτελεί η πώληση ενός συγκεκριμένου προϊόντος σε κάποιο υποκατάστημα σε μια δεδομένη χρονική στιγμή. Μία αριθμητική ποσότητα που αφορά ένα γεγονός ονομάζεται **μέτρο ή αλλιώς μετρική** (measure). Στο γεγονός του προηγούμενου παραδείγματος, ένα μέτρο θα μπορούσε να είναι ο αριθμός των τεμαχίων του προϊόντος που πωλήθηκαν σε πελάτες. Οι πληροφορίες που περιγράφουν το γεγονός ονομάζονται **διαστάσεις** (dimensions). Διαστάσεις ενός γεγονότος πώλησης είναι, για παράδειγμα, το προϊόν που πωλήθηκε, το υποκατάστημα όπου έγινε η πώληση, η ημερομηνία πώλησης κ.λπ. Κάθε διάσταση μπορεί να εμπεριέχει μια **ιεραρχία** (hierarchy). Για παράδειγμα, η διάσταση του χρόνου μπορεί να αναλύεται στο επίπεδο της ώρας, της ημέρας, της εβδομάδας ή του μήνα μιας συναλλαγής κτλ. Η διάσταση του υποκαταστήματος μπορεί να αναλύεται στο επίπεδο της περιοχής, της πόλης, της χώρας κτλ.

Σε μια αποθήκη δεδομένων, τα γεγονότα αναπαρίστανται ως πολυδιάστατοι κύβοι δεδομένων. Κάθε άξονας του κύβου αντιστοιχεί σε μία διάσταση. Κάθε διάσταση αναπαρίσταται ως προς ένα επίπεδο της ιεραρχίας της. Οι τιμές των ιδιοτήτων όλων των διαστάσεων διαμερίζουν τον κύβο σε κελιά, όπου κάθε κελί του περιέχει την αντίστοιχη τιμή του μέτρου. Ο σχεδιασμός ενός κύβου δεδομένων γίνεται συνήθως με **το σχήμα αστέρα (star schema)**. Σ' ένα σχήμα αστέρα, τα γεγονότα αναπαρίστανται στον πίνακα γεγονότων (fact table), ενώ κάθε διάσταση αναπαρίσταται με ξεχωριστό πίνακα διαστάσεων (dimension table). Το σχήμα αστέρα παίρνει την ονομασία του από τη δομή που προκύπτει, με τον πίνακα γεγονότων στο κέντρο και τις διαστάσεις τοποθετημένες ακτινωτά γύρω του. Εκτός απ' το σχήμα αστέρα, υπάρχει το **σχήμα χιονονιφάδας (snowflake schema)**. Η διαφορά του με το σχήμα αστέρα βρίσκεται στο γεγονός ότι οι πίνακες διαστάσεων αποσυντίθενται σε περισσότερους από έναν πίνακες. Επίσης, υπάρχει το **σχήμα γαλαξία (galaxy schema),** όπου δύο ή περισσότεροι πίνακες γεγονότων διαμοιράζονται τους ίδιους πίνακες διαστάσεων.

Σ' έναν κύβο δεδομένων μπορούμε να αλλάζουμε το επίπεδο της ιεραρχίας σε κάθε διάστασή του, προσδιορίζοντας έτσι μια διαφορετική όψη του κύβου δεδομένων. Για παράδειγμα, μπορούμε να εξετάσουμε τις πωλήσεις ανά μήνα αντί ανά ημέρα και, έτσι, να διακρίνουμε ότι τους μήνες Αύγουστο και Νοέμβριο είχαμε μειωμένες πωλήσεις σε σχέση με άλλους μήνες. Αυτές οι ενέργειες εκτελούνται με τις πράξεις OLAP (On line Analytical Processing) που βοηθούν στην εύκολη διατύπωση αναλυτικών ερωτήσεων επί κύβων δεδομένων, καθώς και στη γρήγορη εκτέλεσή τους. Οι βασικές πράξεις OLAP αναλύονται παρακάτω:

Η πράξη **Roll-up** παράγει έναν κύβο δεδομένων με μειωμένο επίπεδο λεπτομέρειας και υλοποιείται όταν (α) σε κάποιες διαστάσεις επιλέγουμε ανώτερο επίπεδο στην ιεραρχία τους ή (β) αφαιρούμε κάποιες διαστάσεις [1, 2]. Η πράξη **Drill-down** παράγει έναν κύβο δεδομένων με αυξημένο επίπεδο λεπτομέρειας και υλοποιείται όταν (α) σε κάποιες διαστάσεις επιλέγουμε κατώτερο επίπεδο στην ιεραρχία τους ή (β) προσθέτουμε κάποιες διαστάσεις. Η πράξη **Slice** παράγει έναν κύβο δεδομένων όταν επιλέγουμε δεδομένα από μία μόνο διάσταση. Η πράξη **Dice** παράγει έναν κύβο δεδομένων όταν επιλέγουμε δεδομένα από μία ή περισσότερες διαστάσεις. Η πράξη **Pivot** παράγει έναν κύβο δεδομένων, του οποίου οι διαστάσεις έχουν αναδιαταχθεί. Τέλος, με τη βοήθεια των πράξεων OLAP είναι εύκολη η πολυδιάστατη ανάλυση ενός κύβου δεδομένων. Όμως, για την εφαρμογή τους απαιτείται ο ορισμός του τρόπου παραγωγής των κύβων-αποτελεσμάτων, μέσω μιας **συναθροιστικής συνάρτησης** (aggregation function) επί των τιμών των μέτρων. Οι βασικές συναθροιστικές συναρτήσεις είναι αυτές του αθροίσματος (sum), του πλήθους (count), του μέσου όρου (avg), του μεγίστου (max) και του ελαχίστου (min).

Τύποι συστημάτων OLAP

Όσον αφορά το φυσικό επίπεδο μιας αποθήκης δεδομένων, στο περιβάλλον του SQL Server υποστηρίζονται τρεις βασικοί **τύποι συστημάτων OLAP**:

1. **Multidimensional OLAP (MOLAP)**

Στα συστήματα MOLAP ο κύβος δεδομένων αποθηκεύεται σε πολυδιάστατους πίνακες. Μ' αυτόν τον τρόπο, επιτυγχάνεται γρήγορη εκτέλεση των πράξεων OLAP, καθώς η προσπέλαση του κύβου είναι άμεση. Ωστόσο, οι πίνακες είναι αραιοί, επειδή δεν αντιστοιχεί πάντα κάθε συνδυασμός διαστάσεων σε ένα γεγονός. Γι' αυτόν τον λόγο, πολλές φορές εφαρμόζεται συμπίεση, ώστε να μειωθεί ο αποθηκευτικός χώρος, κάτι το οποίο επιφέρει αύξηση του χρόνου της δημιουργίας του κύβου.

2. **Relational OLAP (ROLAP)**

Στα συστήματα ROLAP χρησιμοποιείται ένα σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων, όπου υπάρχουν ξεχωριστοί πίνακες γεγονότων και διαστάσεων. Αυτό αποτελεί και ένα από τα σπουδαιότερα πλεονεκτήματα αυτών των συστημάτων, καθώς υπάρχουν πρότυπα για να εφαρμοστεί η σχεσιακή τεχνολογία, με αποτέλεσμα να παρουσιάζουν πολύ καλή κλιμάκωση στη διαχείριση μεγάλου όγκου δεδομένων. Ωστόσο, η ταχύτητα εκτέλεσης αυτών των πράξεων είναι μειωμένη σε σχέση με τα MOLAP.

3. **Hybrid OLAP (HOLAP)**

Τα συστήματα HOLAP είναι τα πιο διαδεδομένα, καθώς δημιουργήθηκαν με σκοπό τον συνδυασμό των πλεονεκτημάτων των παραπάνω συστημάτων. Συγκεκριμένα, δίνεται η δυνατότητα αποθήκευσης ενός τμήματος του κύβου με την μορφή MOLAP για τη γρηγορότερη εκτέλεση πράξεων OLAP, ενώ ο υπόλοιπος κύβος μπορεί να αποθηκευτεί όπως στα συστήματα ROLAP, προκειμένου να επιτευχθεί υψηλή κλιμάκωση σε μεγάλο όγκο δεδομένων.

11.2. Δημιουργία ενός κύβου δεδομένων

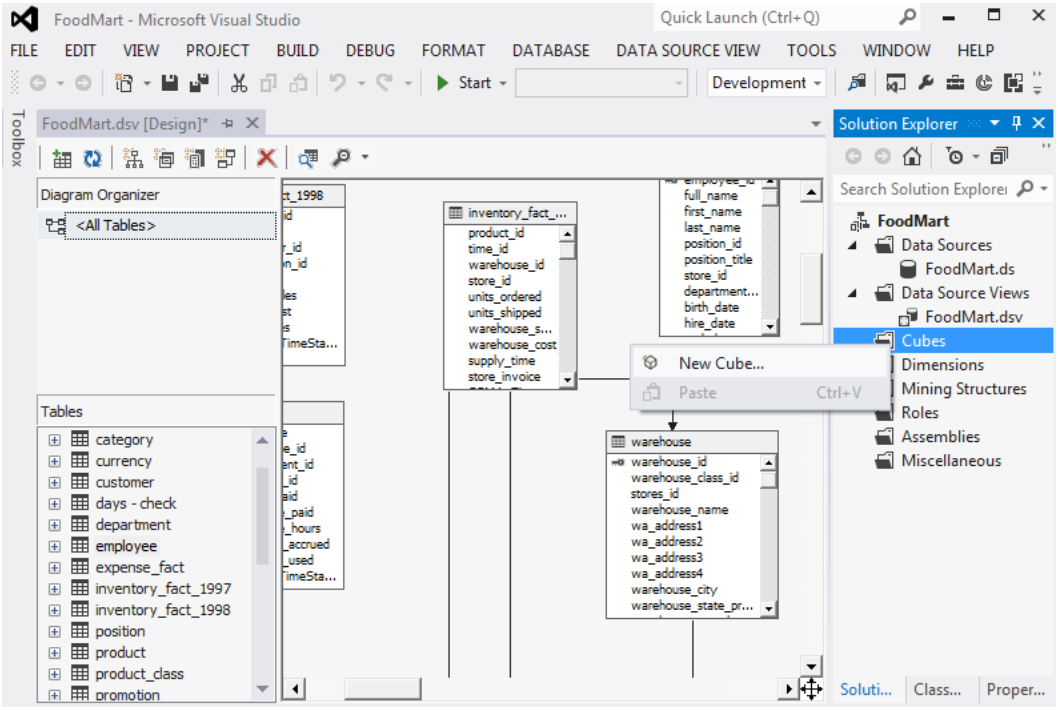
Ας υποθέσουμε ότι βρισκόμαστε στην εταιρία FoodMart ως διαχειριστές της βάσης δεδομένων που έχουμε ήδη επεξεργαστεί στις ενότητες 6.2 και 6.5. Η FoodMart είναι μια μεγάλη αλυσίδα παντοπωλείων με πωλήσεις στις Ηνωμένες Πολιτείες, το Μεξικό και τον Καναδά. Το εμπορικό τμήμα της εταιρείας θέλει να αναλύσει όλες τις πωλήσεις των προϊόντων της και την αγοραστική συμπεριφορά των πελατών της που έγιναν κατά τη διάρκεια του ημερολογιακού έτους 1997. Εμείς, χρησιμοποιώντας τα στοιχεία που αποθηκεύονται στη βάση δεδομένων της επιχείρησης, θα χτίσουμε μια πολυδιάστατη δομή δεδομένων (ένα κύβο), για να επιτρέψουμε τους γρήγορους χρόνους απόκρισης της βάσης, όποτε προστρέχουν σ' αυτήν οι εμπορικοί αναλυτές της εταιρείας. Σ' αυτήν την ενότητα, λοιπόν, θα δημιουργήσουμε, μέσα από αναλυτικά βήματα, έναν κύβο πωλήσεων (Sales Cube) με τα εξής στοιχεία:

a. Πίνακας γεγονότων: Sales_fact_1997
b. Πίνακες διαστάσεων: Product, Product class, Time By Day, Customer, Store
c. Μετρικά: store_sales, store_cost και unit_sales.

Αναλυτικά βήματα

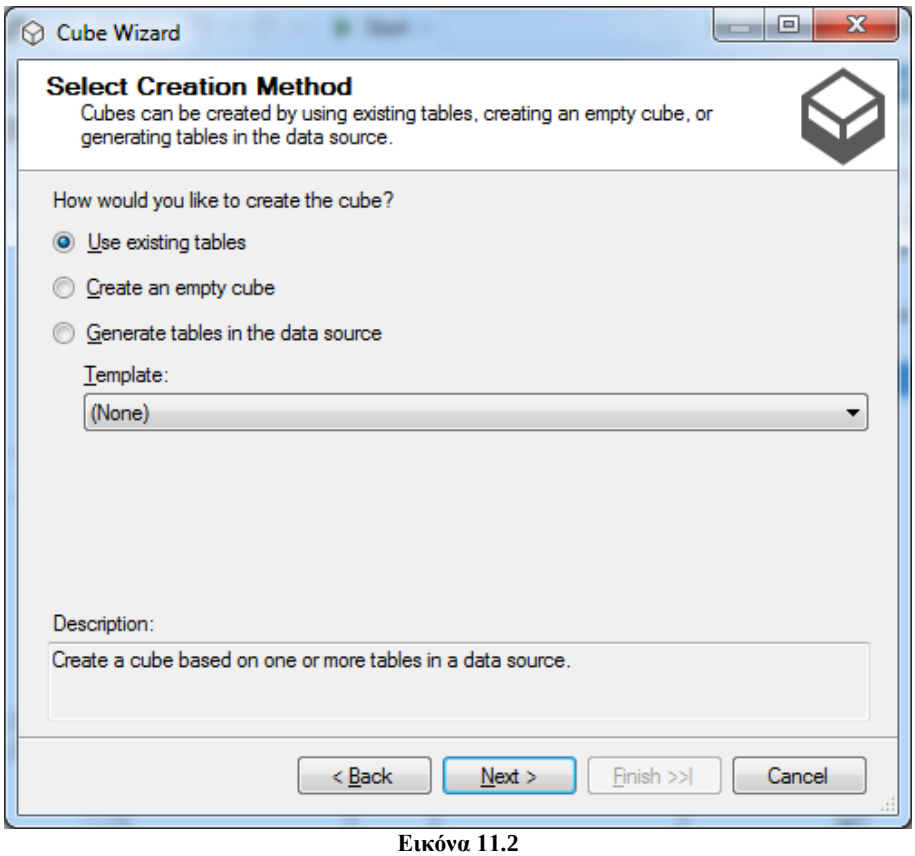
1. Η εισαγωγή της βάσης δεδομένων FoodMart στο περιβάλλον του Management Studio και Business Intelligence έχει γίνει ήδη στο Κεφάλαιο 6. Συγκεκριμένα, η εισαγωγή της βάσης δεδομένων FoodMart στο Management Studio περιγράφεται στην Ενότητα 6.2. Επιπλέον, η εισαγωγή της βάσης δεδομένων FoodMart στο περιβάλλον του Business Intelligence περιγράφεται στην Ενότητα 6.5, όπου το σχεσιακό σχήμα της βάσης δεδομένων περιγράφεται με την Εικόνα 6.66.

2. Από τον Solution Explorer επιλέγουμε Cubes. Στη συνέχεια, επιλέγουμε με δεξί κλικ New Cube, όπως φαίνεται στην Εικόνα 11.1, προκειμένου να δημιουργήσουμε έναν νέο κύβο.



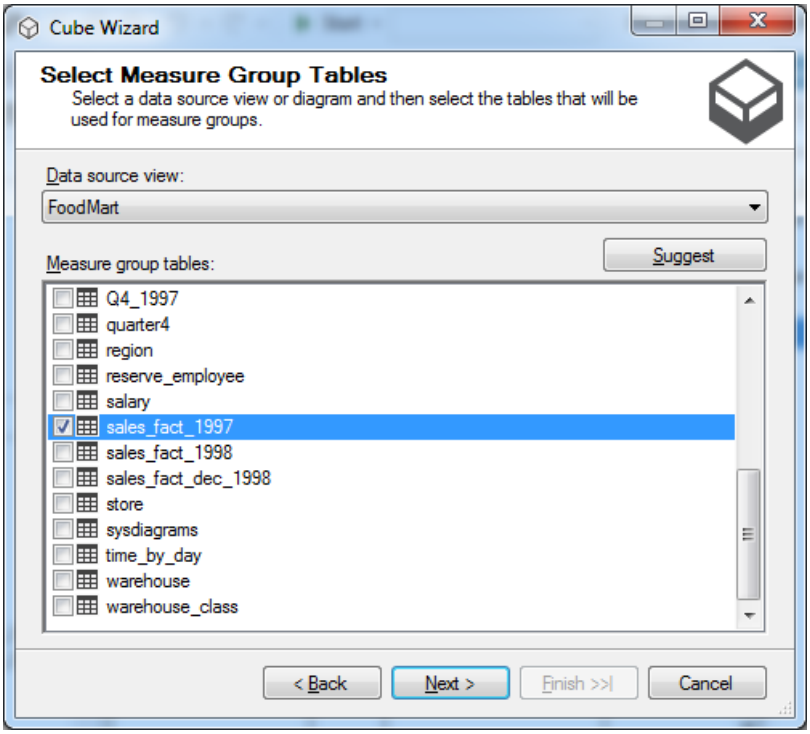
Εικόνα 11.1

3. Στο παράθυρο Cube Wizard, όπως φαίνεται στην Εικόνα 11.2, επιλέγουμε Use existing tables. Στη συνέχεια, πατάμε Next>.



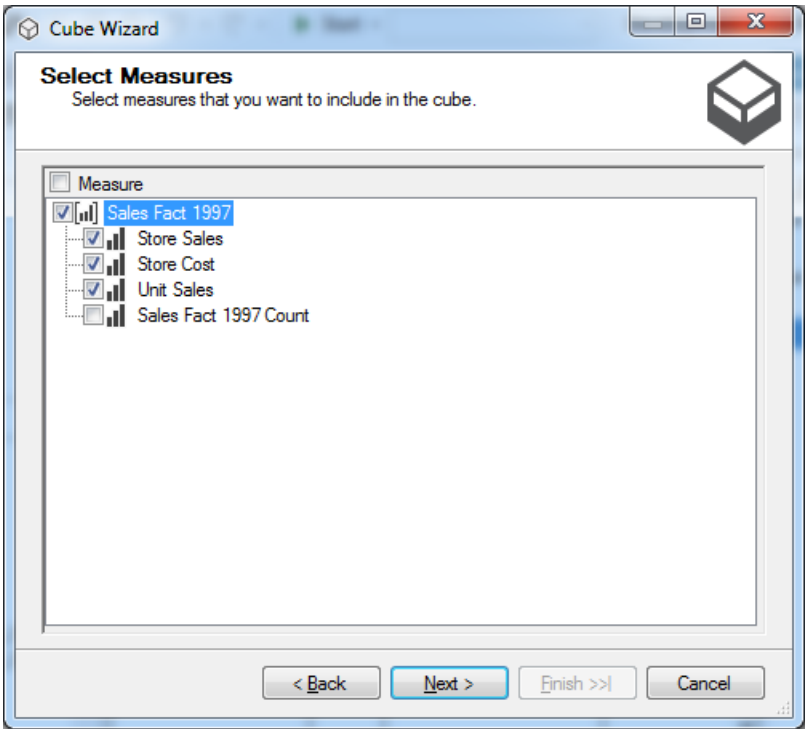
Εικόνα 11.2

4. Στο παράθυρο Select Measure Group Tables, όπως φαίνεται στην Εικόνα 11.3, επιλέγουμε τον πίνακα γεγονότων (fact table). Πιο συγκεκριμένα, επιλέγουμε τον πίνακα γεγονότων sales_fact_1997.



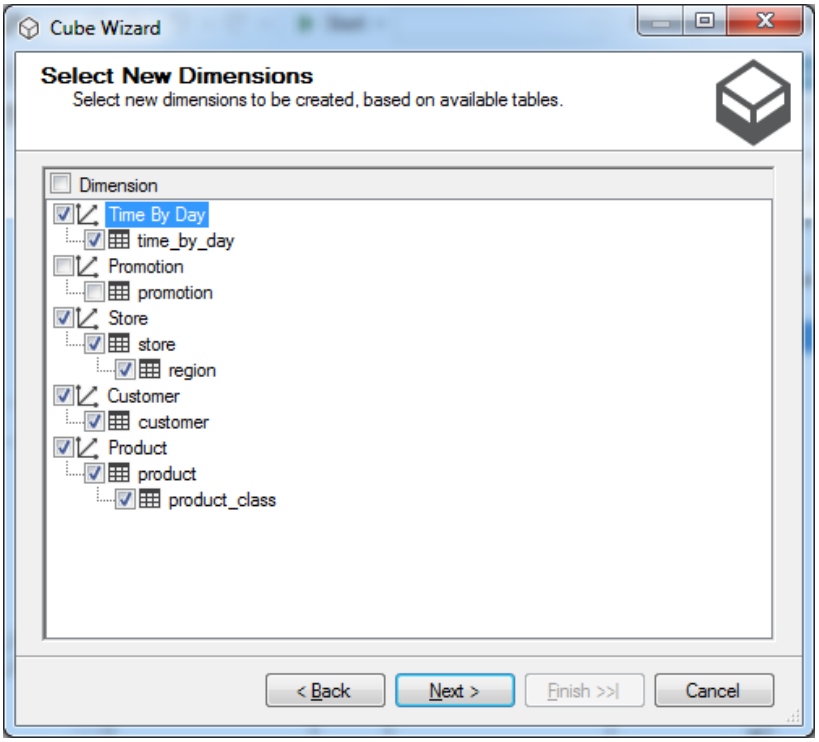
Εικόνα 11.3

5. Στο παράθυρο Select Measures επιλέγουμε τις μετρικές του κύβου μας. Πιο συγκεκριμένα, όπως φαίνεται στην Εικόνα 11.4, διατηρούμε επιλεγμένες τις μετρικές Store Sales, Store Cost και Unit Sales, ενώ αποεπιλέγουμε τη μετρική Sales Fact 1997 Count, την οποία προς το παρόν δεν χρειαζόμαστε.



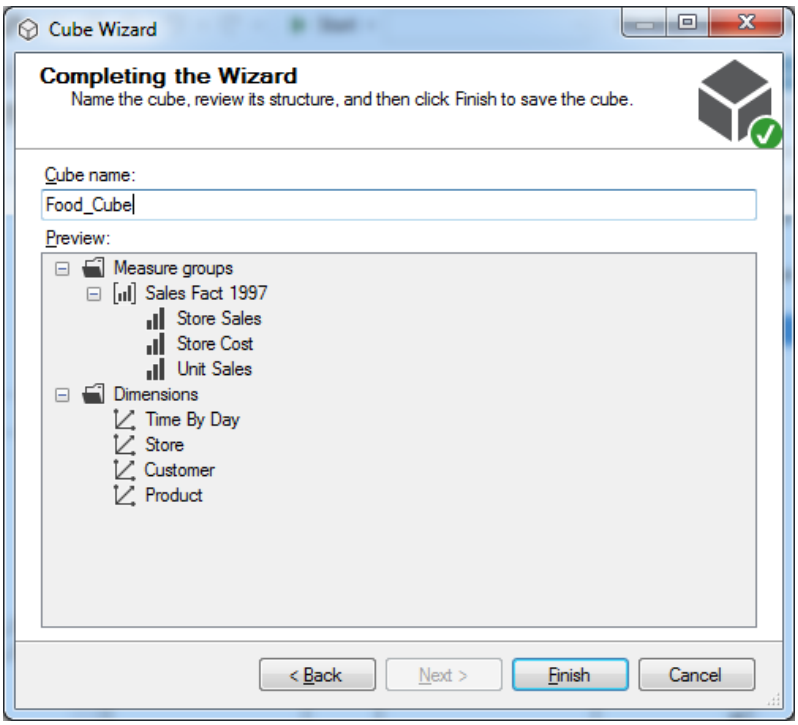
Εικόνα 11.4

6. Στο παράθυρο Select New Dimensions, όπως φαίνεται στην Εικόνα 11.5, εμφανίζονται όλοι οι πίνακες που είναι συνδεδεμένοι στον fact table. Εμείς επιλέγουμε ως πίνακες διαστάσεων τους παρακάτω πίνακες: Product, Product class, Time By Day, Customer, Store. Για τις ανάγκες του παραδείγματος μας αποεπιλέγουμε τον πίνακα Promotion.



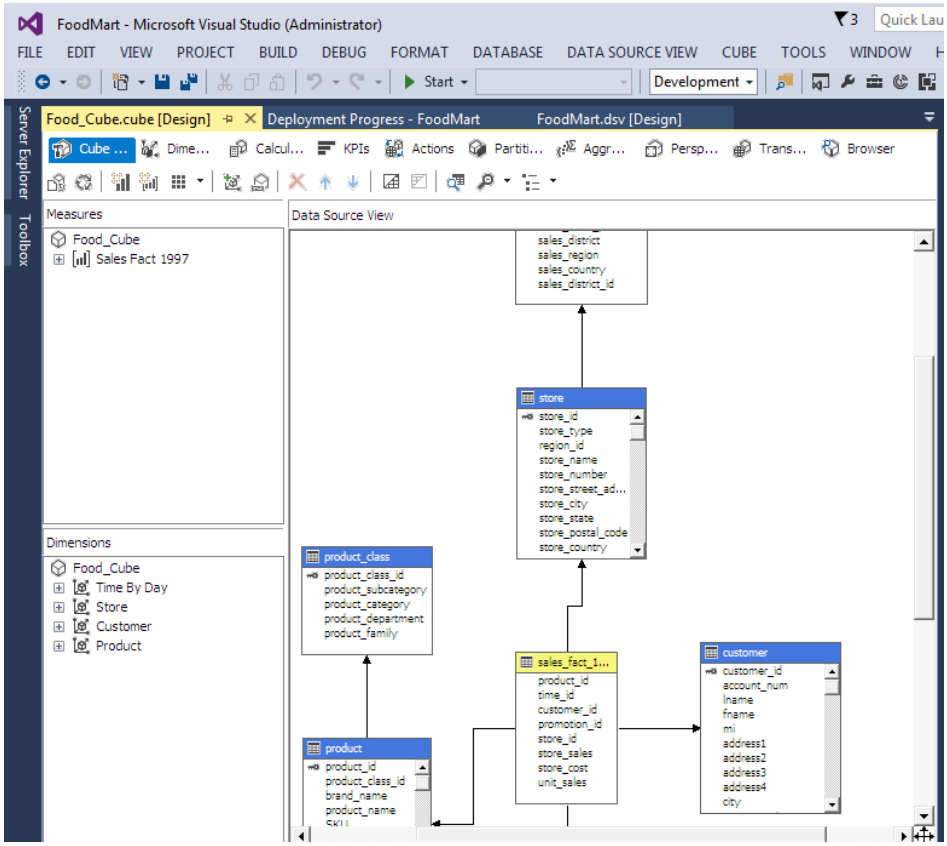
Εικόνα 11.5

7. Στο παράθυρο Completing the Wizard, όπως φαίνεται στην Εικόνα 11.6, ονομάζουμε τον κύβο μας Food_Cube. Στη συνσέχεια, πατάμε Finish.



Εικόνα 11.6

8. Εμφανίζεται διαγραμματικά ο κύβος μας. Όπως φαίνεται στην Εικόνα 11.7, το σχήμα του είναι τύπου χιονονιφάδας. Τώρα, επιλέγοντας Start (το πράσινο κουμπί στην επάνω μπάρα εργαλείων), μπορούμε να κάνουμε τον Κύβο μας process.

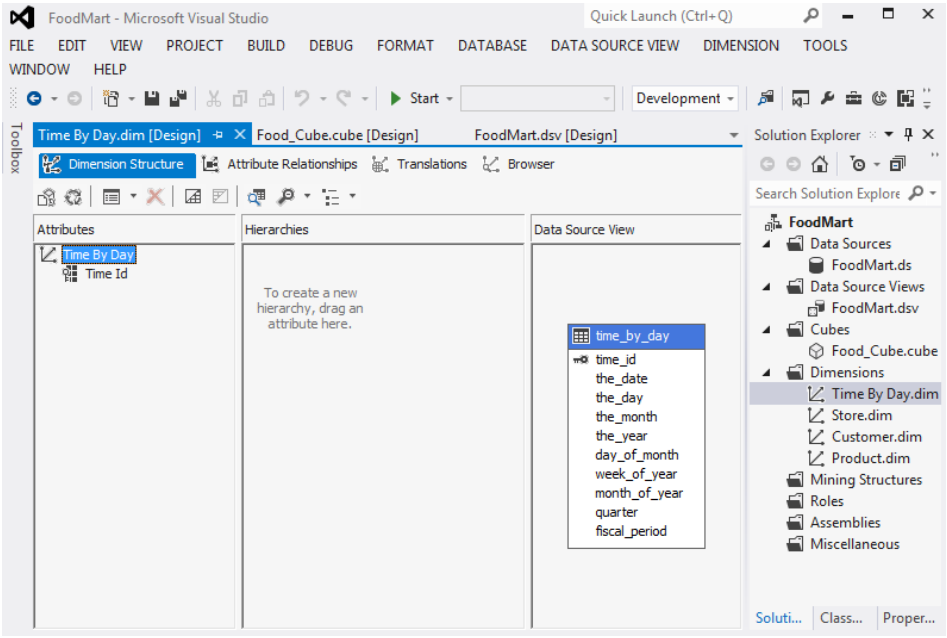


Εικόνα 11.7

11.3. Δημιουργία ιεραρχίας σε μια διάσταση του κύβου δεδομένων

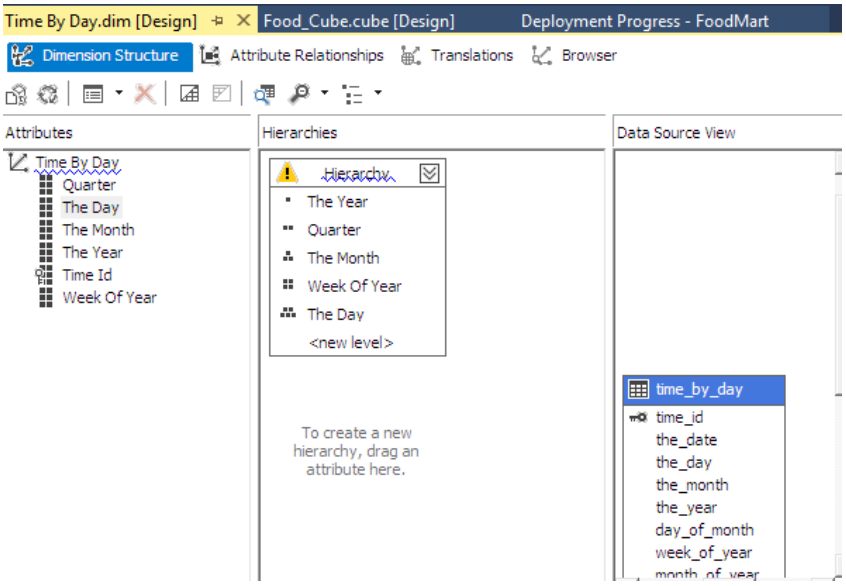
Στην ενότητα αυτή θα δημιουργήσουμε μια ιεραρχία μέσα σε κάθε μία από τις διαστάσεις (time, store, κτλ.) του κύβου πωλήσεων του παραδείγματος μας.

1. Για να δημιουργήσουμε μια ιεραρχία σε μια διάσταση (έστω την διάσταση του χρόνου), κάνουμε δεξί κλικ πάνω στο Dimensions του Solution Explorer και, στην συνέχεια, κάνουμε κλικ στο View Designer. Θα εμφανιστεί η παρακάτω οθόνη.



Εικόνα 11.8

2. Με drag and drop προσθέτουμε τα attributes που μας ενδιαφέρουν (The Year, Quarter, The Month, Week of Year, The Day), μεταφέροντας τα από την περιοχή Data Source View στην περιοχή attributes. Στην συνέχεια, για να δημιουργήσουμε την ιεραρχία της διάστασης Time by Day μεταφέρουμε ξανά με drag and drop τα attributes στην περιοχή Hierarchies, όπως φαίνεται στην Εικόνα 11.9.

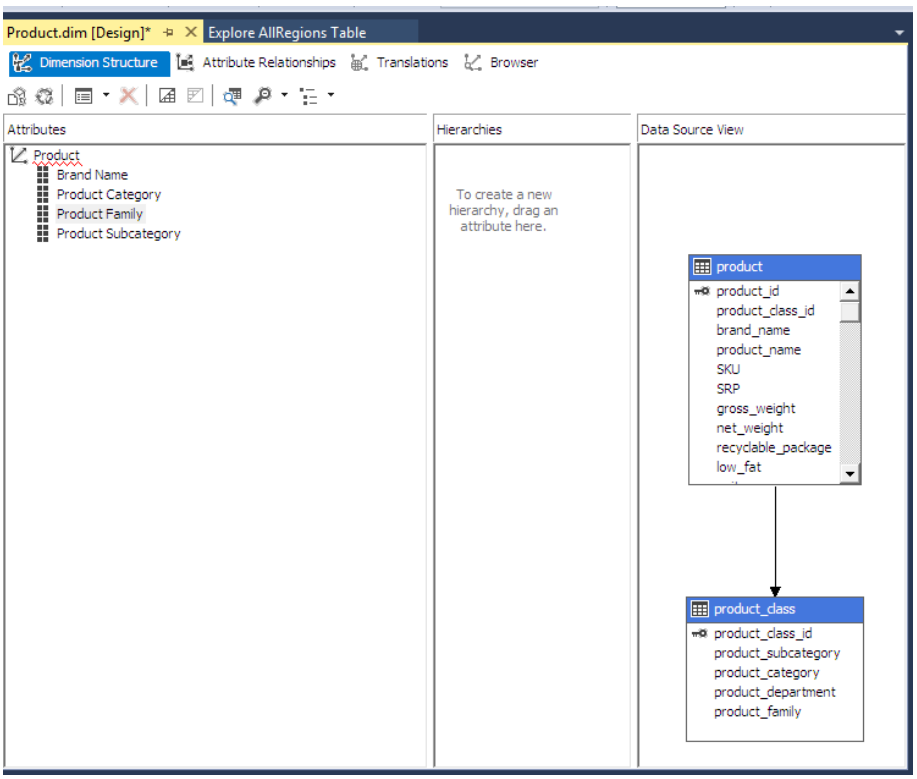


Εικόνα 11.9

3. Το ίδιο θα κάνουμε και με τις άλλες δύο διάστασεις (Store, Product). Για παράδειγμα, για τη διάσταση «Store», η ιεραρχία που θα δημιουργήσουμε φαίνεται στην Εικόνα 11.10.

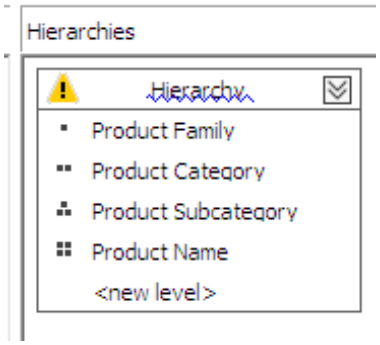
Εικόνα 11.10

4. Για να δημιουργήσουμε τη διάσταση Product, κάνουμε δεξί κλικ πάνω στο Dimensions του Solution Explorer και, στην συνέχεια, κάνουμε κλικ στο View Designer, οπότε θα εμφανιστεί η οθόνη της Εικόνας 11.11. Παρατηρούμε ότι ο πίνακας Product συνδέεται με έναν δεύτερο πίνακα, τον πίνακα product_class.



Εικόνα 11.11

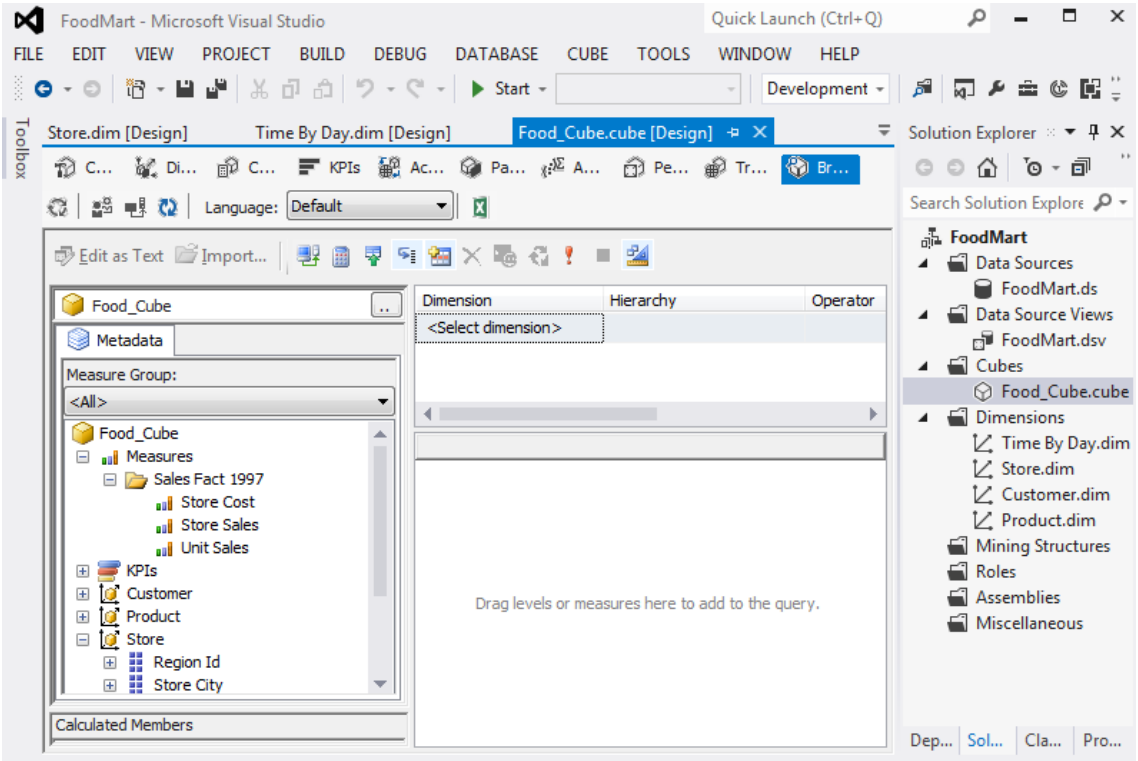
5. Προκειμένου να φτιάξουμε την ιεραρχία για τη διάσταση Product, θα πάρουμε πεδία και από τους δύο συσχετιζόμενους πίνακες, όπως φαίνεται στην Εικόνα 11.12.



Εικόνα 11.12

11.4. Υποβολή ερωτημάτων στον κύβο δεδομένων

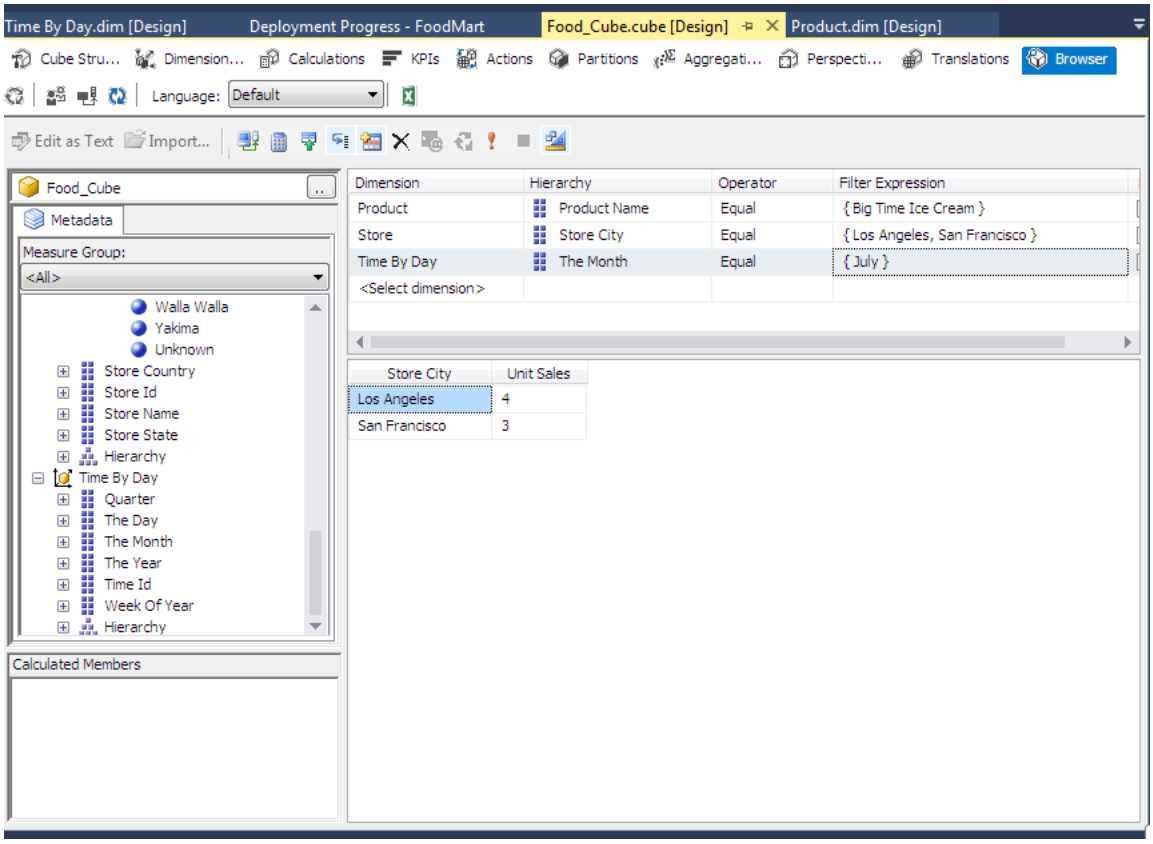
Αφού ολοκληρώσαμε τη διαδικασία δημιουργίας ιεραρχίας για καθεμία απ' τις διαστάσεις του κύβου, κάνουμε process τον κύβο μας, ώστε να ενημερωθεί με τις καινούργιες ιεραρχίες που δημιουργήσαμε στις διαστάσεις του χρόνου, των καταστημάτων και των προϊόντων. Πηγαίνουμε στο solution explorer, κάνουμε διπλό κλικ πάνω στο Food_Cube.cube και επιλέγουμε την καρτέλα Browser. Εμφανίζεται το παράθυρο της Εικόνας 11.13, στο οποίο με drag and drop τοποθετούμε τις διαστάσεις (product, store κτλ.) και το μετρικό (Store cost, Store sales, Unit Sales) που επιθυμούμε.



Εικόνα 11.113

Ας υποθέσουμε ότι μας ζητείται να υπολογίσουμε της ποσότητες που πουλήθηκαν στα καταστήματα της εταιρίας για το προϊόν "Big Time Ice Cream" στις πόλεις Los Angeles και San Francisco για τον μήνα Ιούλιο. Για να ικανοποιήσουμε το παραπάνω ερώτημα, τοποθετούμε το μετρικό Unit Sales με drag and drop στην περιοχή Drag levels or measures to add to the query, όπως φαίνεται στην Εικόνα 11.14. Επίσης, τοποθετούμε με drag and drop στην ίδια περιοχή το πεδίο Store City από τη διάσταση Store.

Προκειμένου να φιλτράρουμε τα δεδομένα μας μόνο για τις πόλεις Los Angeles και San Fransisco για το προϊόν "Big Time Ice Cream" για τον μήνα Ιούλιο, διαμορφώνουμε την περιοχή πάνω από το grid, που αναγράφει <Select dimension>. Συγκεκριμένα, επιλέγουμε Product Name = {Big Time Ice Cream}, Store City = {Los Angeles, San Fransisco} και The Month ={July} για τις διαστάσεις Product, Store και Time by day αντίστοιχα. Είμαστε πλέον έτοιμοι να κάνουμε τις πράξεις drill through ή roll up στον κύβο μας.

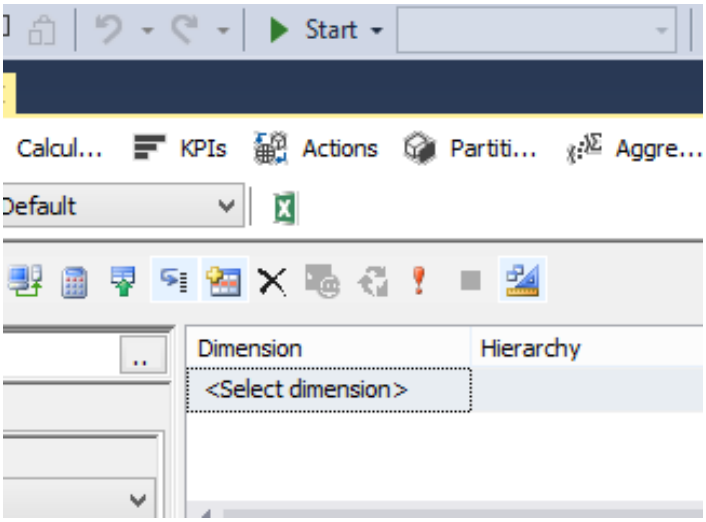


Εικόνα 11.14

11.5. Υποβολή ερωτημάτων μέσω Pivot table του Excel

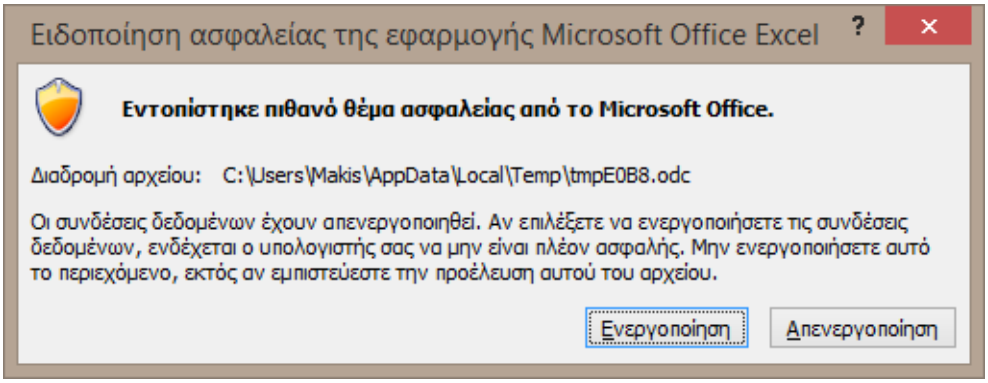
Ένας εναλλακτικός τρόπος υποβολής ερωτημάτων στον κύβο δεδομένων μας είναι η χρήση του Pivot table του MS Excel. Το κύριο πλεονέκτημα αυτής της επιλογής είναι ότι το Excel επιτρέπει να φτιάχνουμε διαγράμματα που περιγράφουν τα δεδομένα μας με γραφικό τρόπο. Το Visual Studio προσφέρει τη δυνατότητα της άμεσης μεταφοράς στο περιβάλλον του MS Excel (εφόσον αυτό έχει προεγκατασταθεί) σύμφωνα με τα παρακάτω βήματα:

1. Επιλέγουμε το κουμπί με το λογότυπο του MS Excel (πράσινο X) στη μέση περίπου της οθόνης μας, όπως φαίνεται στην Εικόνα 11.15.



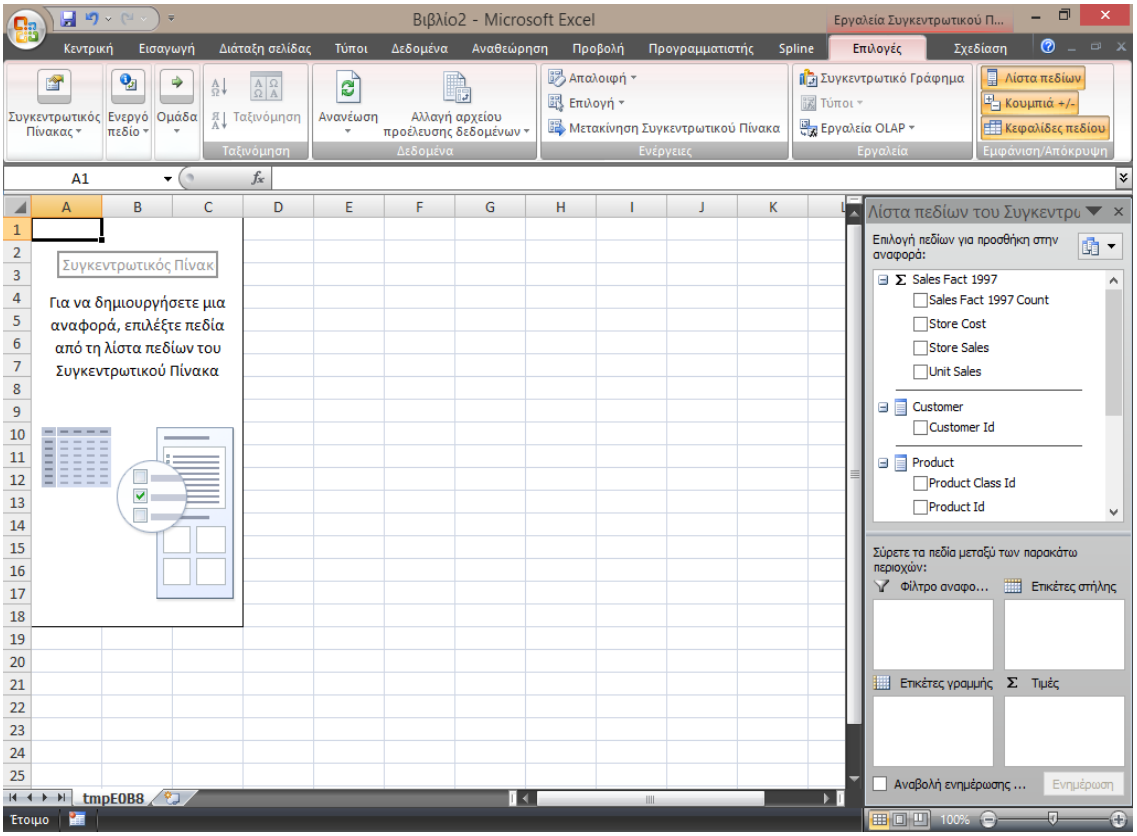
Εικόνα 11.15

2. Μεταφερόμαστε στο MS Excel (στο παράδειγμά μας, στην έκδοση 2007), όπου εμφανίζεται το μήνυμα της Εικόνας 11.16. Επιλέγουμε Ενεργοποίηση.



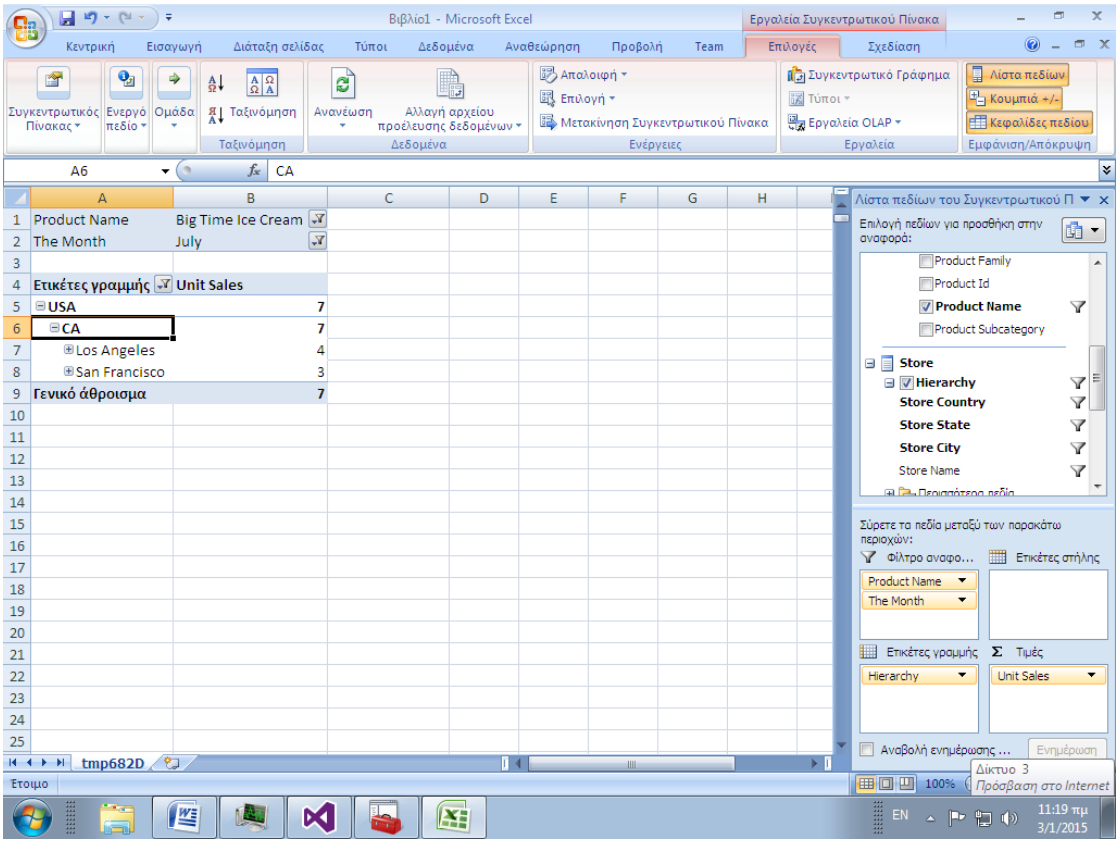
Εικόνα 11.16

3. Εμφανίζεται η οθόνη που φαίνεται στην Εικόνα 11.17. Στα πάνω αριστερά κελιά (Συγκεντρωτικός πίνακας) παρατηρούμε το pivot table. Ακόμη, στο δεξί μέρος βλέπουμε τη λίστα με τα διαθέσιμα πεδία που δημιουργήθηκαν και μεταφέρθηκαν απ' το Visual Studio. Κάτω απ' αυτήν τη λίστα υπάρχουν τέσσερα πεδία: Φίλτρα, Ετικέτες στήλης, Ετικέτες γραμμής, Τιμές.



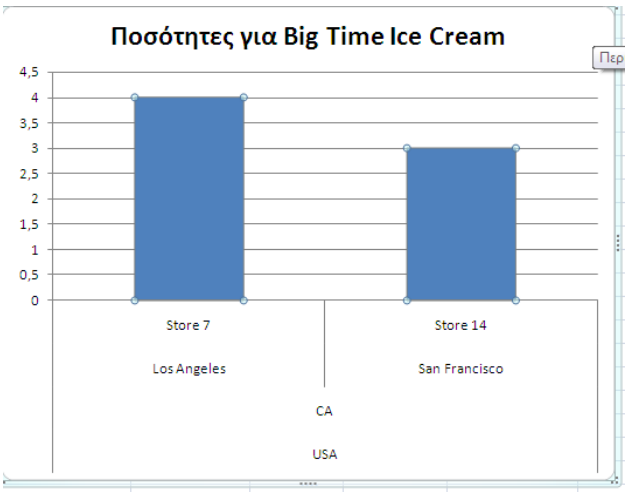
Εικόνα 11.17

4. Προκειμένου να απαντήσουμε το ερώτημα της Ενότητας 11.4., μεταφέρουμε το Unit Sales απ' τη λίστα πεδίων του Συγκεντρωτικού Πίνακα στην αριστερή περιοχή, όπως φαίνεται στην Εικόνα 11.18. Επίσης, μεταφέρουμε την ιεραρχία της διάστασης Store στις Ετικέτες γραμμής. Παράλληλα, στην περιοχή πεδίων του Συγκεντρωτικού Πίνακα, εφαρμόζουμε όλα τα φίλτρα του ερωτήματός μας, δηλαδή: (Product Name = {Big Time Ice Cream}, Store City = {Los Angeles, San Fransisco} και The Month ={July}).



Εικόνα 11.18

5. Τέλος, όπως φαίνεται στην Εικόνα 11.19, στο excel μπορούμε να δημιουργήσουμε ενημερωτικά γραφήματα με εύκολο τρόπο.



Εικόνα 11.19

11.6. Ασκήσεις για κύβους δεδομένων

1. Να προσδιορίσετε τις συνολικές ποσότητες που πωλήθηκαν στην Αμερική ανά πολιτεία (California, Oregon, Washington) για κάθε τρίμηνο του έτους 1997.

2. Ποια τρία καταστήματα σημείωσαν τις μεγαλύτερες πωλήσεις για το 1997; Να φτιάξετε σχετική γραφική παράσταση στο excel.

3. Δημιουργήστε ένα γράφημα που να παρουσιάζει ποιες είναι οι έξι μεγαλύτερες ποσότητες (σε μονάδες/τεμάχια) ανά προϊόν που πωλήθηκαν τον μήνα Δεκέμβριο του 1997.

4. Δημιουργήστε ένα γράφημα που να παρουσιάζει ποια πέντε προϊόντα σημείωσαν τις μεγαλύτερες πωλήσεις (σε αξία) τον μήνα Δεκέμβριο του 1997.

5. Να προσδιορίσετε τις συνολικές πωλήσεις σε αξία των καταστημάτων για τα δύο τελευταία τρίμηνα του 1997 ανά πολιτεία των ΗΠΑ.

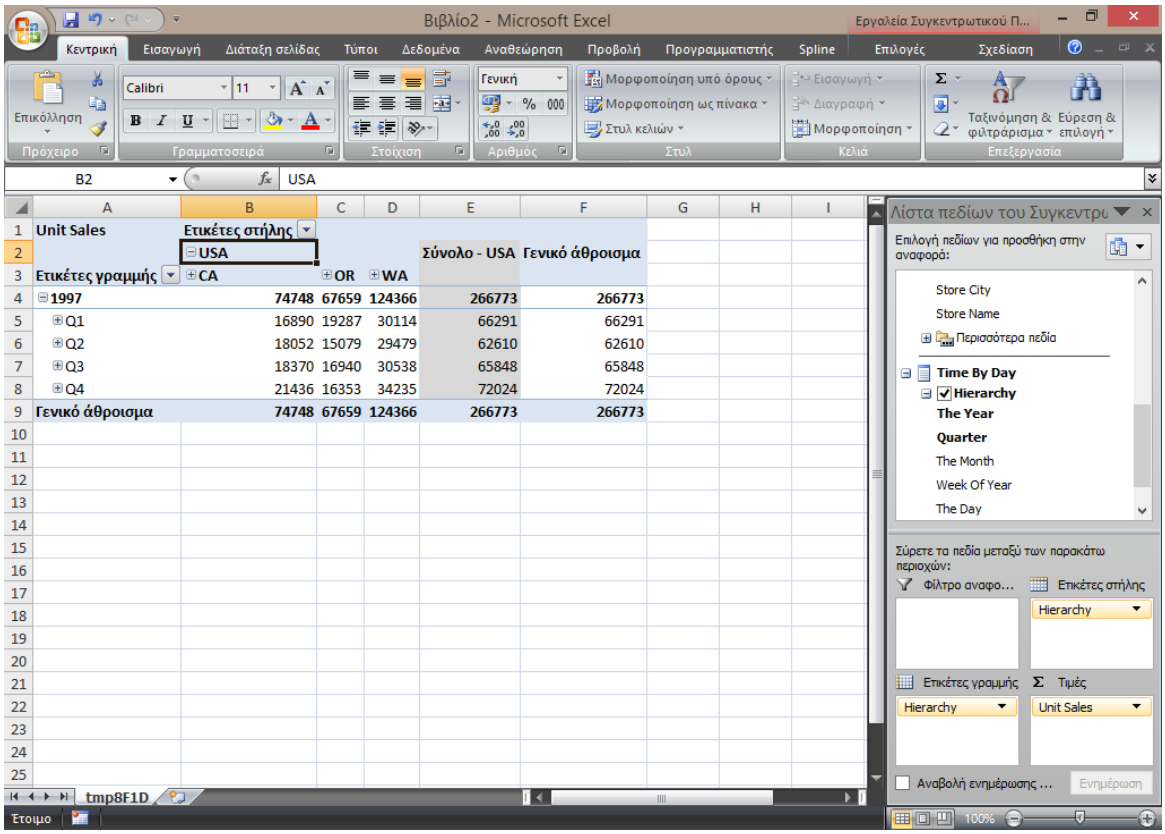
11.7. Λύσεις Ασκήσεων για κύβους δεδομένων

Άσκηση 1

Να προσδιορίσετε τις συνολικές ποσότητες που πωλήθηκαν στην Αμερική ανά πολιτεία (California, Oregon, Washington) για κάθε τρίμηνο του έτους 1997.

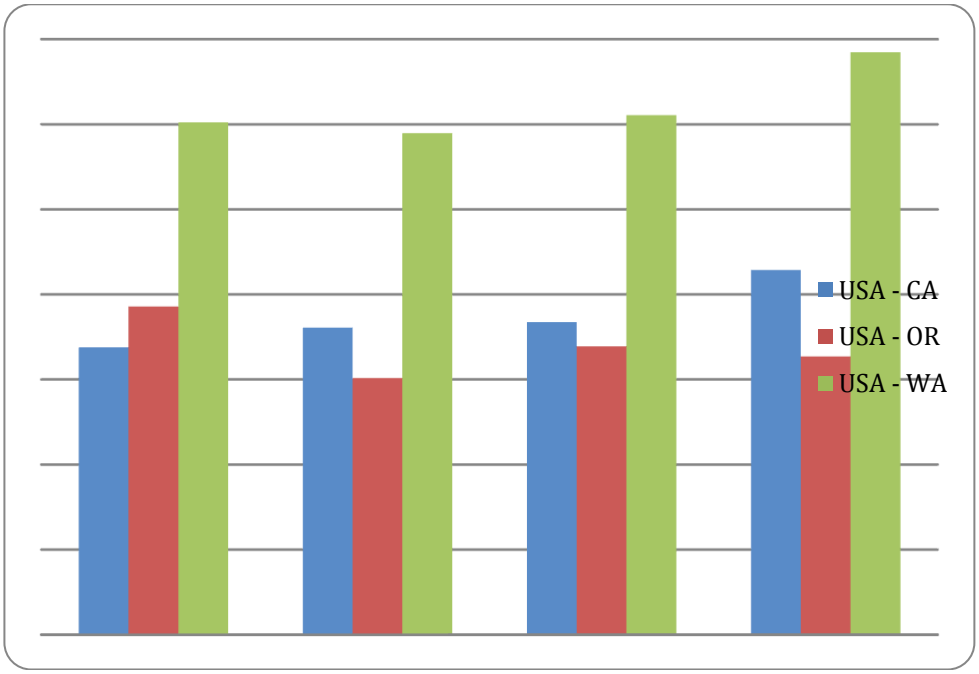
Λύση

1. Πηγαίνουμε στη λίστα πεδίων του Συγκεντρωτικού Πίνακα στο Excel και επιλέγουμε Unit Sales, το οποίο εμφανίζεται πλέον στο κάτω δεξί παράθυρο με τις τιμές που συναθροίζονται (Σ τιμές). Στη συνέχεια, επιλέγουμε ολόκληρη την ιεραρχία της διάστασης Time by day και την τοποθετούμε στην Ετικέτα Γραμμής, η οποία εμφανίζεται στο κάτω αριστερό παράθυρο στη λίστα πεδίων του Συγκεντρωτικού Πίνακα. Τέλος, επιλέγουμε ολόκληρη την ιεραρχία της διάστασης Store και την τοποθετούμε στην περιοχή Ετικέτες στήλης, η οποία εμφανίζεται πλέον στο πάνω δεξί παράθυρο στη λίστα πεδίων του Συγκεντρωτικού Πίνακα. Τονίζεται ότι οι Ετικέτες γραμμής είναι οι γραμμές του Pivot Table, ενώ οι Ετικέτες στήλης αποτελούν τις στήλες του Pivot table. Στη συνέχεια, επιλέγοντας τα σύμβολα + και – , μπορούμε να κάνουμε drill through/roll up τόσο στην ιεραρχία της διάστασης του χρόνου (time) όσο και στη διάσταση των καταστημάτων (stores), όπως φαίνεται στην Εικόνα 11.21.



Εικόνα 11.21

2. Παρατηρούμε, όπως φαίνεται στην Εικόνα 11.22, ένα ραβδόγραμμα με την αποτίμηση των πωλήσεων ανά περιοχή και ανά τρίμηνο του 1997 για τη χώρα της Αμερικής (USA).



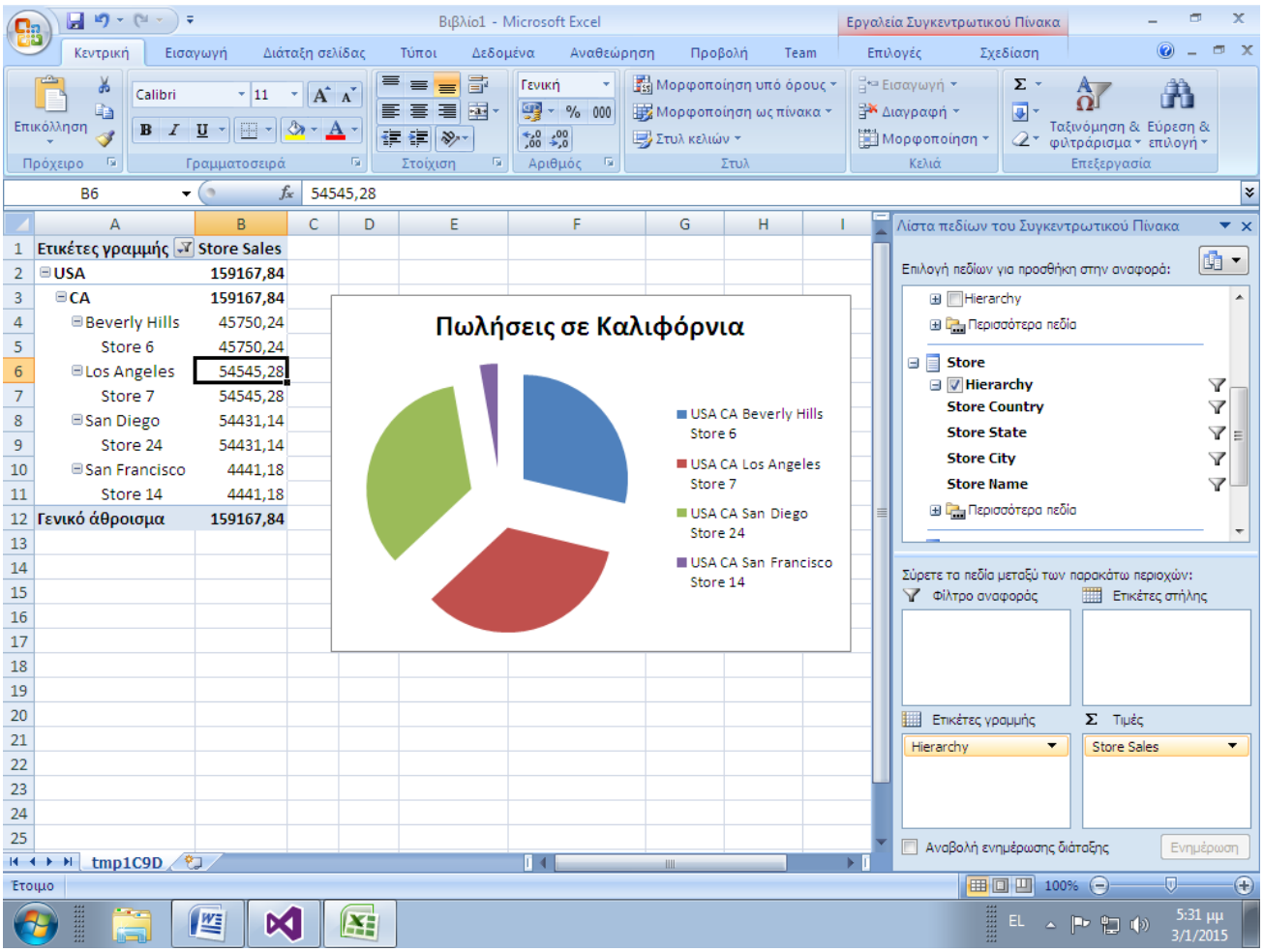
Εικόνα 11.22

Άσκηση 2

Ποιο κατάστημα σημείωσε τις μεγαλύτερες πωλήσεις για το 1997 στη πολιτεία της Καλιφόρνια; Να φτιάξετε σχετική γραφική παράσταση στο excel που να εμπεριέχει όλα τα καταστήματα της Καλιφόρνια.

Λύση

Στη λίστα πεδίων του Συγκεντρωτικού Πίνακα στο Excel, επιλέγουμε Store Sales, το οποίο εμφανίζεται πλέον στο κάτω δεξί παράθυρο με τις τιμές που συναθροίζονται (Σ τιμές). Στη συνέχεια, επιλέγουμε ολόκληρη την ιεραρχία της διάστασης Store και την τοποθετούμε στην περιοχή Ετικέτες Γραμμής, η οποία εμφανίζεται στο κάτω αριστερό παράθυρο στη λίστα πεδίων του Συγκεντρωτικού Πίνακα. Όπως φαίνεται στην Εικόνα 11.23, διαπιστώνουμε ότι το κατάστημα του Los Angeles είναι αυτό που έχει τις μεγαλύτερες πωλήσεις, με 54545,28 δολάρια Αμερικής. Το ίδιο αποτέλεσμα, εξάλλου, επιβεβαιώνεται και με το συγκεντρωτικό γράφημα πίτας.



Ετικέτες γραμμής	Store Sales
USA	159167,84
CA	159167,84
Beverly Hills	45750,24
Store 6	45750,24
Los Angeles	54545,28
Store 7	54545,28
San Diego	54431,14
Store 24	54431,14
San Francisco	4441,18
Store 14	4441,18
Γενικό άθροισμα	159167,84

Εικόνα 11.23

Άσκηση 3

Δημιουργήστε ένα γράφημα που να παρουσιάζει ποιες είναι οι έξι μεγαλύτερες ποσότητες (σε μονάδες/τεμάχια) ανά προϊόν που πωλήθηκαν τον μήνα Δεκέμβριο του 1997.

Λύση

Στο γράφημα της Εικόνας 11.24 παρουσιάζονται τα έξι προϊόντα της εταιρίας με τις μεγαλύτερες πωλήσεις για το χρονικό διάστημα αναφοράς. Τα προϊόντα αναφέρονται στο σύνολο της εμπορικής δραστηριότητας της εταιρίας και στο σύνολο της γεωγραφικής κάλυψης των προϊόντων της. Όπως φαίνεται, το προϊόν "Fast Beef Jerky" παρουσίασε τις υψηλότερες πωλήσεις στη συγκεκριμένη περίοδο.

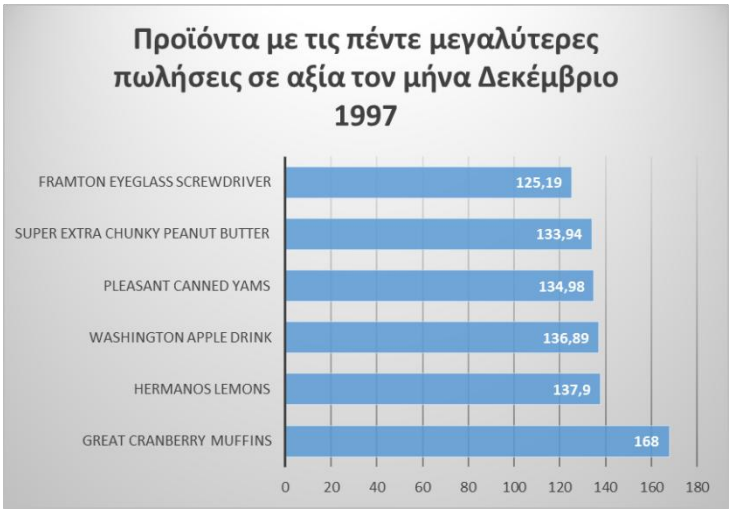
Εικόνα 11.24

Άσκηση 4

Δημιουργήστε ένα γράφημα που να παρουσιάζει ποια πέντε προϊόντα σημείωσαν τις μεγαλύτερες πωλήσεις (σε αξία) τον μήνα Δεκέμβριο του 1997.

Λύση

Στο γράφημα της Εικόνας 11.25 παρουσιάζεται το διάγραμμα των προϊόντων με τις πέντε μεγαλύτερες πωλήσεις σε αξία για τον μήνα Δεκέμβριο του 1997.



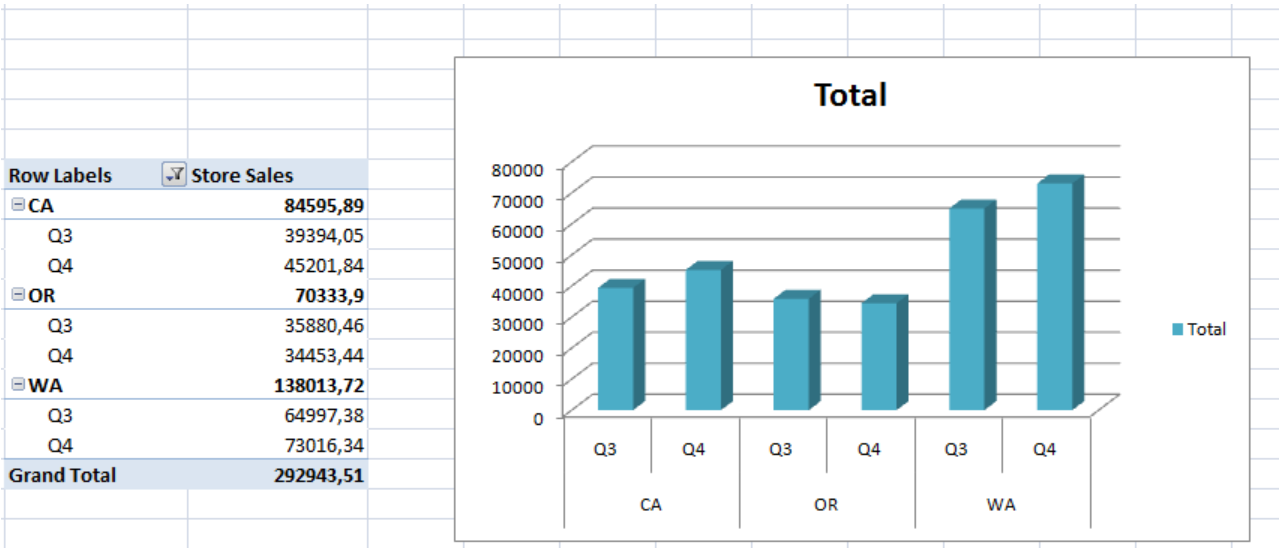
Εικόνα 11.25

Άσκηση 5

Να προσδιορίσετε τις συνολικές πωλήσεις σε αξία των καταστημάτων για τα δύο τελευταία τριμήνων του 1997 ανά πολιτεία των ΗΠΑ.

Λύση

Όπως φαίνεται στην Εικόνα 11.26, υπάρχει ανοδική πορεία πωλήσεων μεταξύ του τρίτου και του τέταρτου τριμήνου στα καταστήματα που βρίσκονται στις πολιτείες CA και WA. Αντίθετα, τα καταστήματα της πολιτείας του OR φαίνεται πως έχουν μικρή μείωση των πωλήσεών τους απ' το τρίτο στο τέταρτο τρίμηνο.



Row Labels	Store Sales
CA	84595,89
Q3	39394,05
Q4	45201,84
OR	70333,9
Q3	35880,46
Q4	34453,44
WA	138013,72
Q3	64997,38
Q4	73016,34
Grand Total	292943,51

Εικόνα 11.26

11.8. Βιβλιογραφία/Αναφορές

Νανόπουλος, Α., & Μανωλόπουλος, Ι. (2008). *Εισαγωγή στην Εξόρυξη και τις Αποθήκες Δεδομένων*, Αθήνα, Εκδόσεις Νέων Τεχνολογιών.

Χαλκίδη, Μ., & Βεζυργιάννης, Μ. (2005). *Εξόρυξη Γνώσης από Βάσεις Δεδομένων και τον Παγκόσμιο Ιστό*, Αθήνα, Τυπωθήτω.

Βιβλιογραφία

Aggarwal, C. C. (2015). *Data Mining: The Textbook*, Springer.

Chakrabarti, S. (2003). *Mining the Web: Discovering Knowledge from Hypertext Data*, Morgan – Kauffman.

Dunham, M. H. (2003). *Data Mining: Introductory and Advanced Topics*, New Jersey, Prentice Hall.

Han, J., & Kamber, M. (2001). *Data Mining: Concepts and Techniques*, Academic Press.

Hoffer, J. A., Venkatarama, R., & Topi, H. (2013). *Modern Database Management*, Prentice Hall.

Liu, B. (2007). *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*, Springer.

Μανωλόπουλος, Ι., & Παπαδόπουλος, Α. Ν. (2006). *Συστήματα Βάσεων Δεδομένων: Θεωρία & Πρακτική Εφαρμογή*, Αθήνα, Εκδόσεις Νέων Τεχνολογιών.

Νανόπουλος, Α., & Μανωλόπουλος, Ι. (2008). *Εισαγωγή στην Εξόρυξη και τις Αποθήκες Δεδομένων*, Αθήνα, Εκδόσεις Νέων Τεχνολογιών.

Rajaraman, A., Leskovec, J., & Ullman, J.D. (2015). *Mining of Massive Datasets*, Cambridge University Press.

Ramakrishnan, R., & Gehrke, J. (2003). Database Management Systems, McGraw-Hill.

Roiger, R., & Geatz, M. (2003). *Data Mining: A tutorial-based Primer*, Addison Wesley.

Tan, P - N., Steinbach, M., & Kumar, V. (2006). *Introduction to Data Mining*, Addison Wesley.

Χαλκίδη, Μ., & Βεζυργιάννης, Μ. (2005). *Εξόρυξη Γνώσης από Βάσεις Δεδομένων και τον Παγκόσμιο Ιστό*, Αθήνα, Τυπωθήτω.

Ευρετήριο όρων