

ΠΑΝΑΓΙΩΤΗΣ ΣΥΜΕΩΝΙΔΗΣ
Διδάκτωρ Τμήματος Πληροφορικής
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Βάσεις, Αποθήκες και Εξόρυξη Δεδομένων με τον SQL Server

Εργαστηριακός Οδηγός

ΕΛΛΗΝΙΚΑ ΑΚΑΔΗΜΑΪΚΑ ΗΛΕΚΤΡΟΝΙΚΑ
ΣΥΓΓΡΑΜΜΑΤΑ ΚΑΙ ΒΟΗΘΗΜΑΤΑ

Τίτλος Ηλεκτρονικού Συγγράμματος

Συγγραφή

Παναγιώτης Συμεωνίδης

Κριτικός αναγνώστης

Όνομα

Συντελεστές έκδοσης

ΓΛΩΣΣΙΚΗ ΕΠΙΜΕΛΕΙΑ: Παναγιώτης Συμεωνίδης

ΓΡΑΦΙΣΤΙΚΗ ΕΠΙΜΕΛΕΙΑ: Χρήστος Άνδρας

ΤΕΧΝΙΚΗ ΕΠΕΞΕΡΓΑΣΙΑ: Χρήστος Άνδρας

@ Copyright Θεσσαλονίκη 2013

Πνευματικά δικαιώματα

ΣΥΝΔΕΣΜΟΣ ΕΛΛΗΝΙΚΩΝ ΑΚΑΔΗΜΑΪΚΩΝ ΒΙΒΛΙΟΘΗΚΩΝ

Εθνικό Μετσόβιο Πολυτεχνείο

Ηρώων Πολυτεχνείου 9, 15780 Ζωγράφου

www.kallipos.gr

ISBN:

Πίνακας περιεχομένων

Εισαγωγή	5
Κεφάλαιο 1. Εγκατάσταση και Περιβάλλον του SQL Server 2012.....	6
1.1. Εγκατάσταση του SQL Server.....	6
1.2. Έλεγχος καλής λειτουργίας της εγκατάστασης.....	9
1.3. Το περιβάλλον του SQL Server Management Studio	11
Κεφάλαιο 2. Δημιουργία Βάσης Δεδομένων και Πινάκων.....	13
2.1. Δημιουργία Βάσης Δεδομένων.....	13
2.1.1. Δημιουργία νέας βάσης σε γραφικό περιβάλλον.....	14
2.1.2. Δημιουργία νέας βάσης με κώδικα SQL.....	17
2.2. Δημιουργία Πινάκων	18
2.2.1. Τύποι Δεδομένων	18
2.2.2. Χρήσιμες συμβουλές για τους τύπους δεδομένων	20
2.2.3. Δημιουργία πινάκων με τον Management Studio	22
2.2.4. Δημιουργία πινάκων με κώδικα της SQL.....	24
2.2.5. Συσχετίσεις/Relationships Πινάκων.....	26
2.2.6 Δημιουργία πινάκων με τον Database Diagrams	28
2.3. Εισαγωγή τιμών στους πίνακες	33
2.3.1. Εισαγωγή τιμών στους πίνακες με το Management Studio	33
2.3.1. Εισαγωγή τιμών στους πίνακες με εντολές SQL.....	34
Κεφάλαιο 3. Ερωτήματα SQL	39
3.1. Βασικά Ερωτήματα	39
3.1.1 Διαχείριση του Results Pane.....	39
3.1.2 Ερωτήματα επιλογής γραμμών από ένα πίνακα.....	40
3.1.3 Ταξινόμηση αποτελεσμάτων.....	43
3.2. Ερωτήματα επιλογής γραμμών από πολλούς πίνακες.....	45
3.2.1 Εσωτερική και εξωτερική σύνδεση πινάκων	45
3.2.2 Μετονομασία και αυτό-σύνδεση.....	48
3.3. Ερωτήματα ομαδοποίησης.....	50
3.3.1 Συναρτήσεις ομαδοποίησης	50
3.3.2 Ο όρος Group by.....	51
3.3.3 Ο όρος Having.....	52
3.4. Ερωτήματα για πράξεις συνόλων με φωλιασμένες εντολές SQL.....	54
3.4.1 Η πράξη της Ένωσης (Union).....	54
3.4.2 Έλεγχος περιεκτικότητας σε σύνολο	54
3.4.3 Σύγκριση μεταξύ συνόλων	55
3.4.4 Σύγκριση μεταξύ συνόλων	56
3.5. Ερωτήματα SQL για όψεις	58

3.6. Query Designer και Query by Example.....	59
3.7. Ασκήσεις με ερωτήματα SQL.....	62
3.7.1 Ασκήσεις με ερωτήματα επιλογής γραμμών από ένα πίνακα.....	62
3.7.2 Ασκήσεις με ερωτήματα επιλογής γραμμών από πολλούς πίνακες.	62
3.7.3 Ασκήσεις με Ερωτήματα ομαδοποίησης/συνάθροισης δεδομένων.	62
3.7.4 Ασκήσεις με ερωτήματα με φωλιασμένες εντολές SQL και όψεις.	62
Κεφάλαιο 4. Προχωρημένες λειτουργίες στον SQL Server	63
4.1. Ερωτήματα ορισμού δεδομένων.....	63
4.1.1 Εισαγωγή πολλών γραμμών σε πίνακα	63
4.1.2 Ενημέρωση τιμής των πεδίων ενός πίνακα.....	64
4.1.3 Διαγραφή των γραμμών ενός πίνακα	64
4.1.4 Μεταβολή της δομής ενός πίνακα	65
4.1.5 Μετονομασία πίνακα και πεδίου πίνακα.....	65
4.1.6 Διαγραφή πίνακα και βάσης δεδομένων	66
4.2. Αποθηκευμένες Διαδικασίες, Εναύσματα, Συναλλαγές.....	67
4.2.1 Αποθηκευμένες διαδικασίες/Stored Procedures	67
4.2.2 Εναύσματα/Triggers	70
4.2.3 Συναλλαγές/Transactions	71
4.3. Βελτιστοποίηση Ερωτημάτων	73
4.3.1 Παρακολούθηση του πλάνου εκτέλεσης ερωτήματος SQL.....	73
4.3.2 Ευρετήρια/Indexes.....	75
4.3.3 Παράδειγμα βελτίωσης απόδοσης ερωτήματος με την χρήση ευρετηρίου.....	76
4.4. Εξαγωγή του κώδικα της βάσης δεδομένων.....	78

Εισαγωγή

Το βιβλίο αυτό περιγράφει τις βασικές λειτουργίες και δυνατότητες που προσφέρονται από ένα σύστημα διαχείρισης βάσεων δεδομένων. Συγκεκριμένα, στο βιβλίο αυτό περιγράφονται οι δυνατότητες του SQL Server 2012. Ο SQL Server είναι ένα ισχυρό εργαλείο που περιέχει ένα πλήθος δυνατοτήτων, όπως οι εντολές που αφορούν την γλώσσα ορισμού δεδομένων (Data Definition Language), εντολές της γλώσσας χειρισμού δεδομένων (Data Manipulation Language), και την γλώσσα προγραμματισμού Transact-SQL για την δημιουργία αποθηκευμένων διαδικασιών, ενανυσμάτων και συναλλαγών. Κάθε κεφάλαιο αυτού του βιβλίου περιγράφει λεπτομερώς μία από τις παραπάνω δυνατότητες του SQL Server.

Το πρώτο κεφάλαιο είναι εισαγωγικό και ο αναγνώστης θα δει πώς μπορεί να εγκαταστήσει τον SQL Server 2012. Συγκεκριμένα, στο παρόν 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.

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

Στο τρίτο κεφάλαιο θα παρουσιάσουμε βασικά και σύνθετα ερωτήματα της SQL. Τα ερωτήματα θα υποβληθούν στην βάση δεδομένων Video Club, που δημιουργήθηκε στο προηγούμενο κεφάλαιο. Συγκεκριμένα, θα μελετηθούν εντολές της SQL που αφορούν την διαχείριση δεδομένων (DML). Ενδεικτικά αναφέρεται ότι θα παρουσιαστούν ερωτήματα σύνδεσης πινάκων, ομαδοποίησης, πράξεων συνόλων 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 μελετηθούν εντολές της SQL που αφορούν την γλώσσα DDL, όπως την μεταβολή της δομής των πινάκων, την διαγραφή τους κτλ. Άλλα προχωρημένα θέματα που θα παρουσιαστούν είναι η βελτιστοποίηση ερωτημάτων με την χρήση indexes, οι αποθηκευμένες διαδικασίες, τα ενανύσματα και οι συναλλαγές με την βοήθεια της γλώσσας προγραμματισμού Transact-SQL.

Σε αυτό το σημείο θέλω να ευχαριστήσω ιδιαίτερα τον συνάδελφο και φίλο μου Χρήστο Άνδρα που βοήθησε στην επικαιροποίηση των κεφαλαίων αυτού του βιβλίου, προκειμένου αυτά να προσαρμοστούν στο νέο περιβάλλον του SQL Server 2012, καθώς επίσης και για τις χρήσιμες παρατηρήσεις του που οδήγησαν στην περαιτέρω βελτίωση του βιβλίου.

Κεφάλαιο 1. Εγκατάσταση και Περιβάλλον του SQL Server 2012

Σύνοψη

Στο παρόν 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

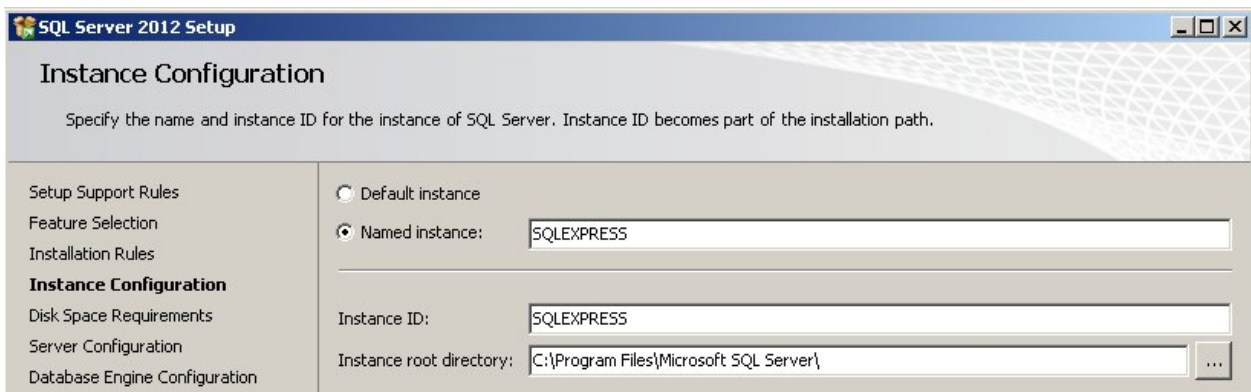
Ο SQL Server 2012 διατίθεται σε διαφορετικές εκδόσεις (Express, Standard, Enterprise), οι οποίες καλύπτουν είτε βασικές είτε πιο εξειδικευμένες ανάγκες μιας επιχείρησης. Εμείς θα εγκαταστήσουμε την έκδοση Express with Tools μιας και είναι freeware και διατίθεται δωρεάν. Συγκεκριμένα, θα εγκαταστήσουμε την έκδοση Express with Tools (όνομα αρχείου: SQLEXPRTW_x86_ENU.exe, μέγεθος αρχείου: 1.091.146 KB), η οποία περιλαμβάνει την μηχανή του SQL Server για την διαχείριση των δεδομένων, αλλά και ένα γραφικό περιβάλλον διαχείρισης (Management Studio). Συνεπώς, κατεβάζουμε την έκδοση Express with Tools του SQL Server 2012 με ενσωματωμένο το service pack 1 από το ακόλουθο link: <http://www.microsoft.com/en-us/download/details.aspx?id=35579>.

Εκτελούμε με διπλό κλικ στο αρχείο SQLEXPRTW_x86_ENU.exe την εγκατάσταση. Ανοίγει 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

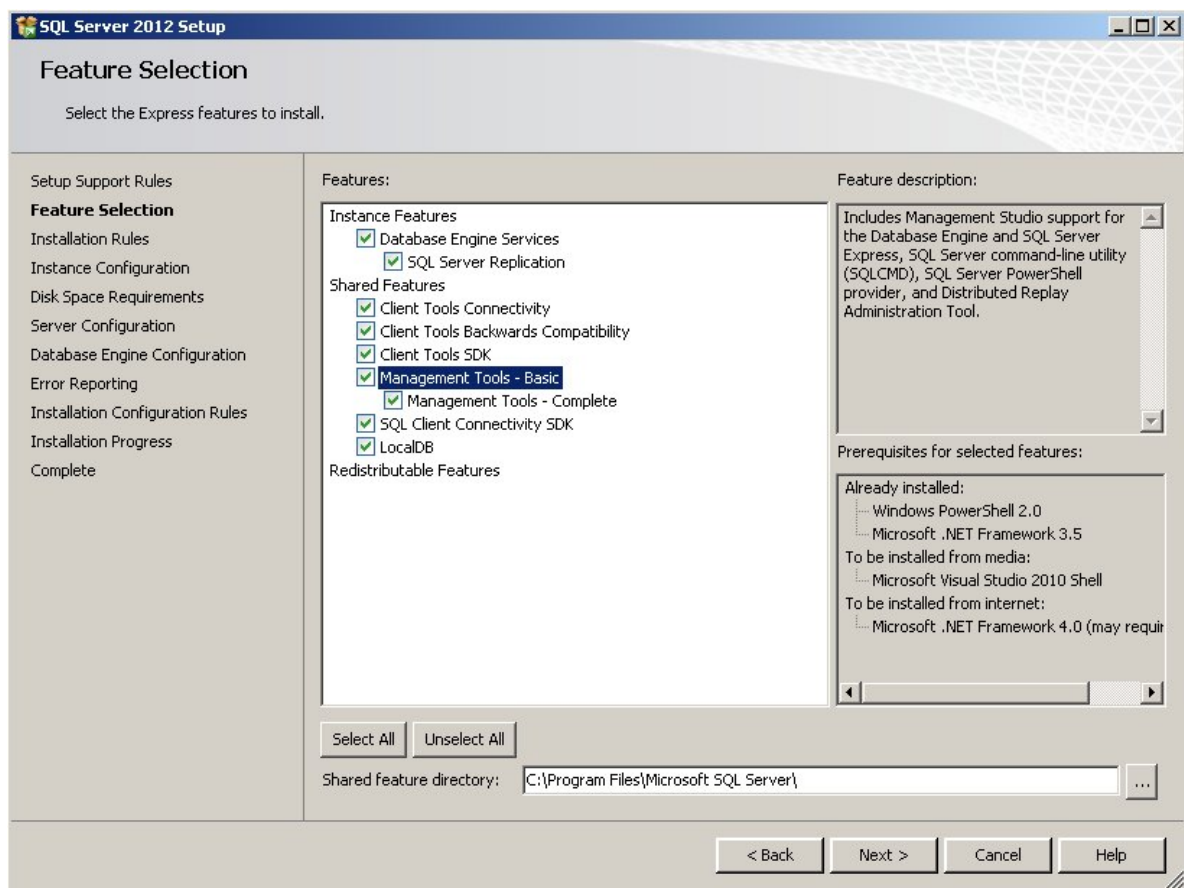
Καταρχήν εκτελούνται κάποιοι έλεγχοι για την ύπαρξη ή μη των ελάχιστων υπολογιστικών προδιαγραφών που πρέπει να έχει ο υπολογιστής, στον οποίο γίνεται η εγκατάσταση. Σε επόμενο στάδιο μας ζητείται το όνομα του instance της εγκατάστασης, όπως φαίνεται στην Εικόνα 1.2. Εφόσον επιλέξουμε «Named instance»,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



Εικόνα 1.2

ΠΡΟΣΟΧΗ! Σε περίπτωση που αλλάξετε το όνομα του υπολογιστή σας, θα πρέπει να αλλάξετε και το όνομα του instance για την σύνδεσή σας στον SQL Server.

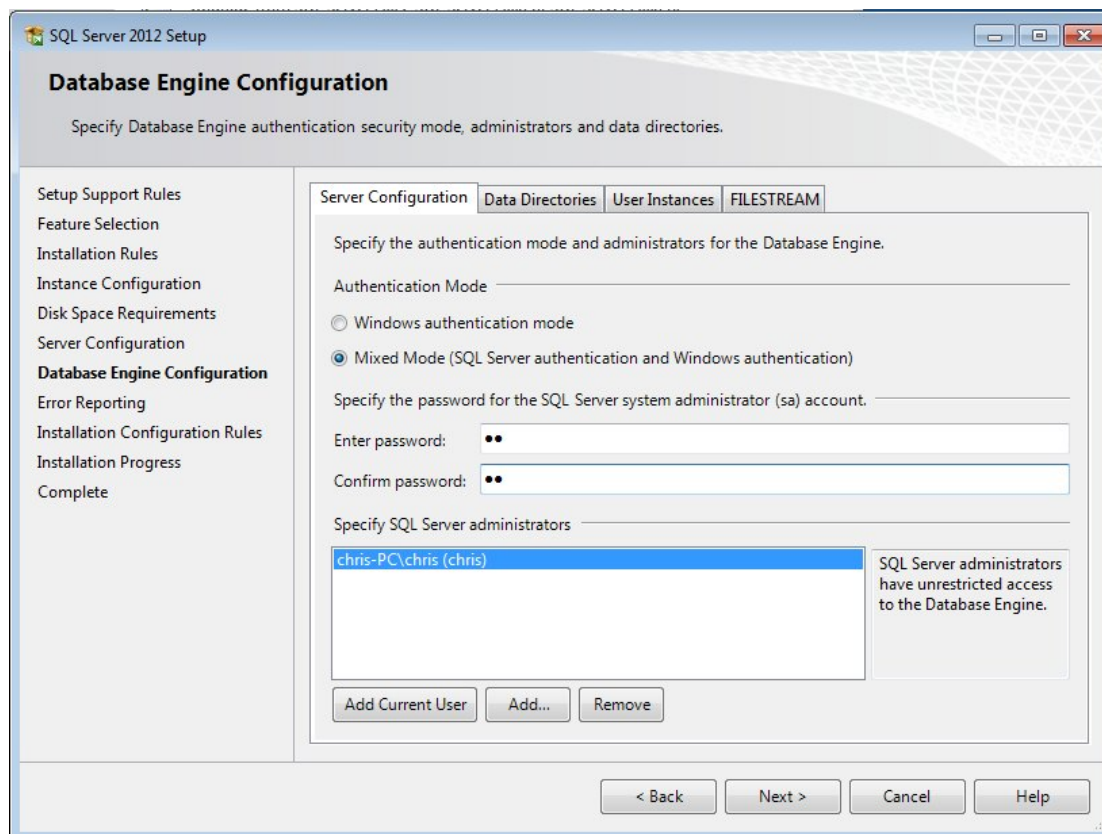
Στο παράθυρο της Εικόνας 1.3 μπορούμε να επιλέξουμε ποιά χαρακτηριστικά θέλουμε να εγκατασταθούν. Στο παράδειγμα μας, εμείς θα επιλέξουμε να εγκατασταθούν όλα τα χαρακτηριστικά του SQL Server.



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 θα προσδιορίσουμε και ένα ανεξάρτητο σύστημα πιστοποίησης του SQL Server (Mixed Mode authentication). Στην δεύτερη περίπτωση, η οποία και προτείνεται μιας και

είναι απαραίτητη για να συνδεόμαστε μέσω δικτύου στον SQL Server, υπάρχει ένας predefined default user με όνομα «sa» για τον οποίο προσδιορίζουμε σε αυτό το σημείο της εγκατάστασης το password που επιθυμούμε. Αργότερα μπορείτε να φτιάξετε και άλλους χρήστες ή Logins ανάλογα με τις ανάγκες σας. 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 chris και η εγκατάσταση γίνεται στον υπολογιστή chris – PC.

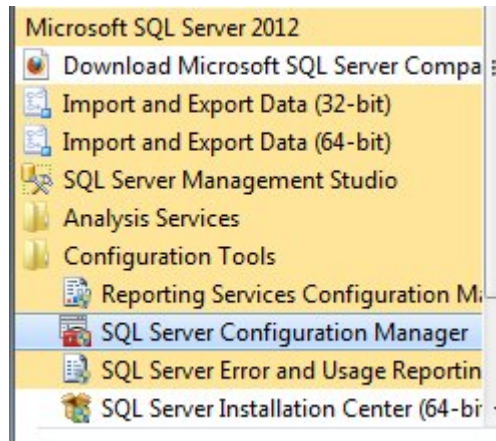


Εικόνα 1.4

Μετά από λίγο θα τερματιστεί η εγκατάσταση. **ΠΡΟΣΟΧΗ!** Σε περίπτωση προβλήματος κατά την διάρκεια της εγκατάστασης, μην επιχειρήσετε να εκτελέσετε νέες εγκαταστάσεις την μία πάνω στην άλλη. Ο σωστός τρόπος είναι να απεγκατασταθεί οποιοδήποτε προηγούμενο ίχνος μίας προβληματικής εγκατάστασης του SQL Server από την προσθαφαίρεση προγραμμάτων και όχι με χειρωνακτικό σβήσιμο φακέλων.

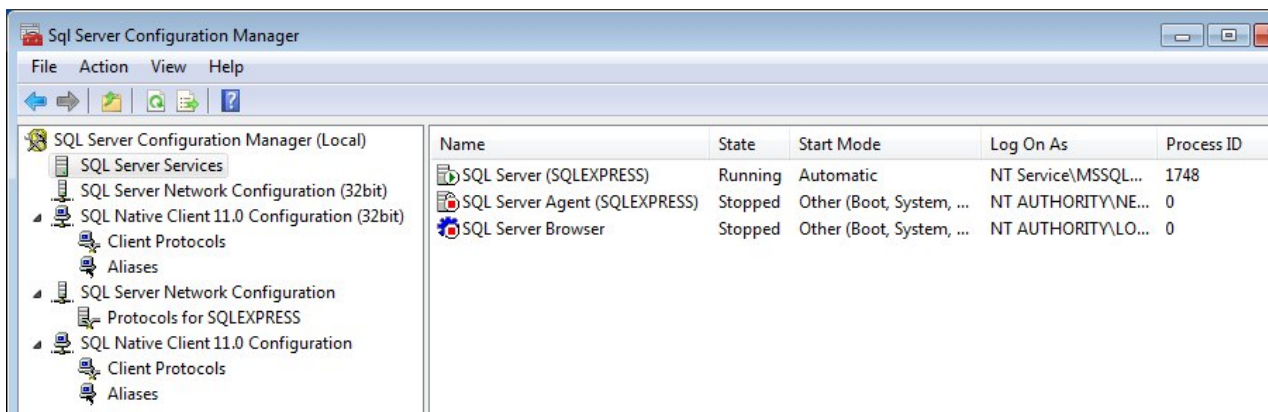
1.2. Έλεγχος καλής λειτουργίας της εγκατάστασης

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



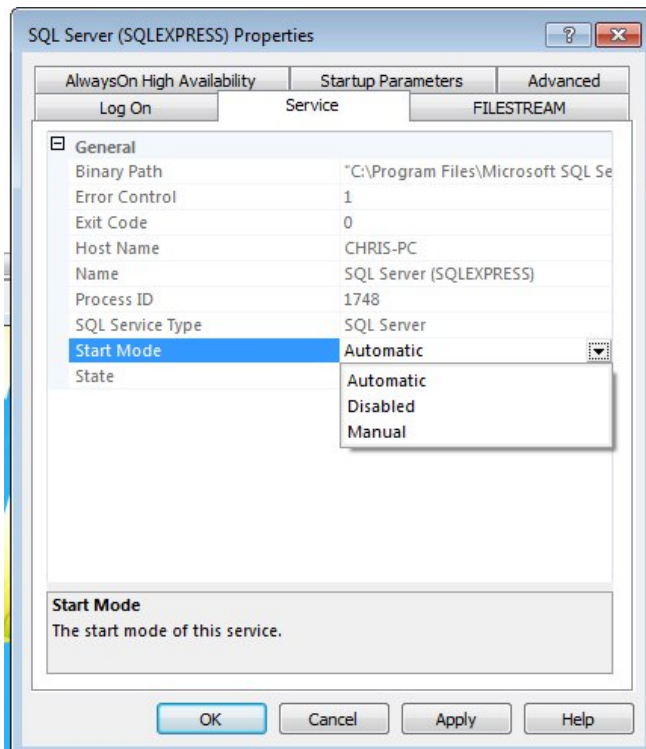
Εικόνα 1.5

Το πρώτο και βασικό service που πρέπει να έχει κατάσταση (state) running είναι αυτό του SQL Server. Διαφορετικά, δεν θα μπορούμε να συνδεθούμε στην βάση δεδομένων μας, να αποθηκεύουμε δεδομένα, ή να εκτελούμε ερωτήματα σε αυτήν. Είναι και το πρώτο που ελέγχουμε όπως φαίνεται στην Εικόνα 1.6. Το δεύτερο σημαντικότερο service είναι αυτό του SQL Server Browser, η δουλειά του οποίου είναι να βοηθά να συνδεθεί κάποιος σε περιβάλλον δικτύου με το όνομα του instance του SQL Server και όχι μόνο τη διεύθυνση IP του υπολογιστή.



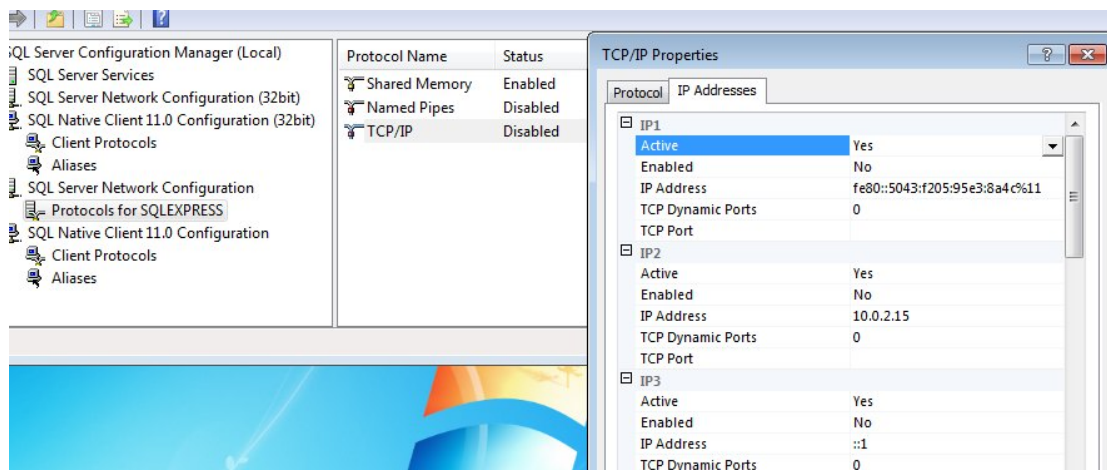
Εικόνα 1.6

Κάθε υπηρεσία μπορούμε ανάλογα με τις ανάγκες μας να την ρυθμίσουμε να ξεκινά (σε σχέση με τον υπολογιστή) αυτόματα, χειρωνακτικά ή να την απενεργοποιήσουμε πλήρως. Η τελευταία επιλογή είναι χρήσιμη όταν για μεγάλο χρονικό διάστημα δεν σκοπεύουμε να χρησιμοποιήσουμε τον SQL Server και έτσι δεν θέλουμε να δεσμεύουμε την 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.

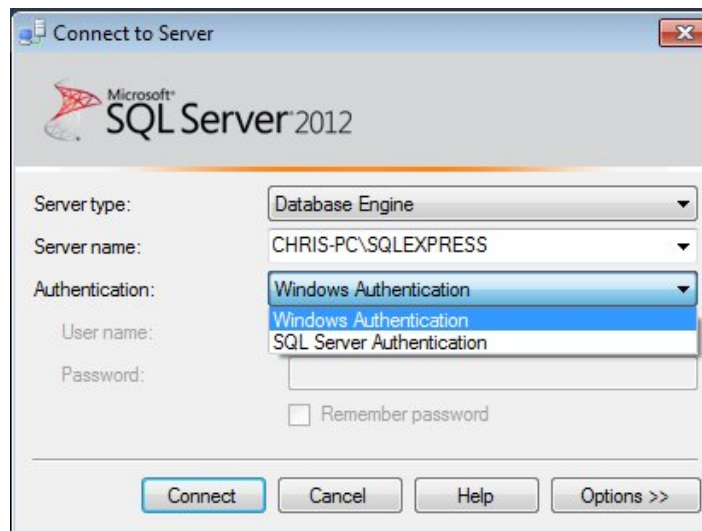


Εικόνα 1.8

1.3. Το περιβάλλον του SQL Server Management Studio

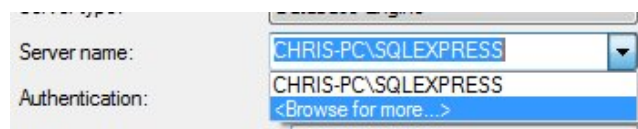
Το πρόγραμμα αυτό αποτελεί ένα γραφικό περιβάλλον σύνδεσης και διαχείρισης του SQL Server. Επισημαίνεται ότι δεν αφορά μόνο την σύνδεση με τον SQL Server που εγκαταστάθηκε στον υπολογιστή σας αλλά και για οποιοδήποτε άλλον απομακρυσμένο SQL Server μέσω είτε τοπικού δικτύου είτε διαδικτύου, Internet,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

Από το μενού του SQL Server στα προγράμματα θα βρούμε την επιλογή (Start→Programs→>Microsoft SQL Server 2012→ SQL Server Management Studio) και κάνοντας κλικ θα ξεκινήσει η εφαρμογή. Η πρώτη οθόνη μας ζητά να συνδεθούμε σε κάποιο instance, όπως φαίνεται στην Εικόνα 1.9.



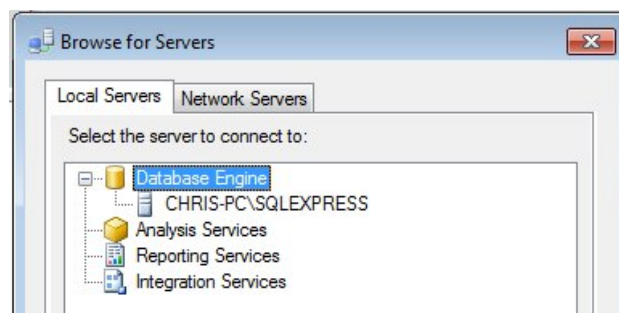
Εικόνα 1.9

Προκειμένου να συνδεθούμε, αρκεί να επιλέξουμε το Server name και το Authentication mode. Στην περίπτωση που επιλέξουμε ως τρόπο πρόσβασης το SQL Server authentication mode θα πρέπει να εισάγουμε ένα username και password. Σε περίπτωση προβλημάτων σύνδεσης και εφόσον δεν μπορείτε να γράψετε χειρωνακτικά το όνομα του Server μπορείτε να επιλέξετε browse for more, όπως φαίνεται στην Εικόνα 1.10.



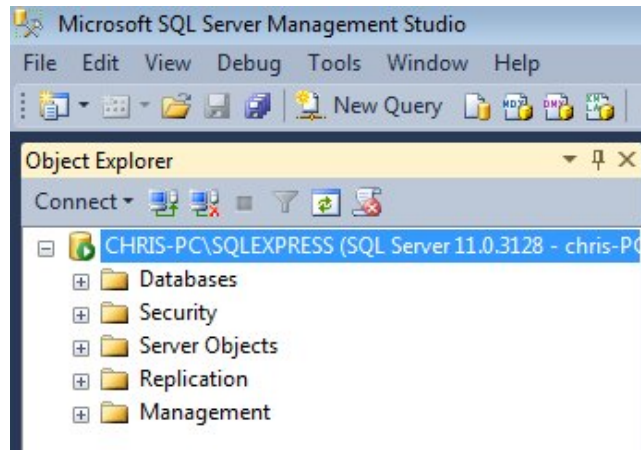
Εικόνα 1.10

Τότε το πρόγραμμα θα σας εμφανίσει τα instances που βρίσκει διαθέσιμα στο τοπικό δίκτυο, όπως φαίνεται στην Εικόνα 1.11.



Εικόνα 1.11

Εφόσον συνδεθούμε επιτυχώς, όπως φαίνεται στην Εικόνα 1.12, εμφανίζεται το instance στο οποίο συνδεθήκαμε (CHRIS-PC\SQLEXPRESS) και η έκδοσή του (11.0.3128). Αμέσως μετά, βλέπουμε σε δενδροειδή μορφή επιλογές διαχείρισης του SQL Server. Η βασική επιλογή που θα ασχοληθούμε στο βιβλίο μας είναι η επιλογή **Databases**. Σε αυτό τον φάκελο μπορούμε να διαχειριστούμε μία ή περισσότερες βάσεις δεδομένων και τα αντικείμενα τους που εμφανίζονται κάνοντας κλικ πάνω σε μία βάση δεδομένων στον φάκελο databases.



Εικόνα 1.12

Κεφάλαιο 2. Δημιουργία Βάσης Δεδομένων και Πινάκων

Σύνοψη

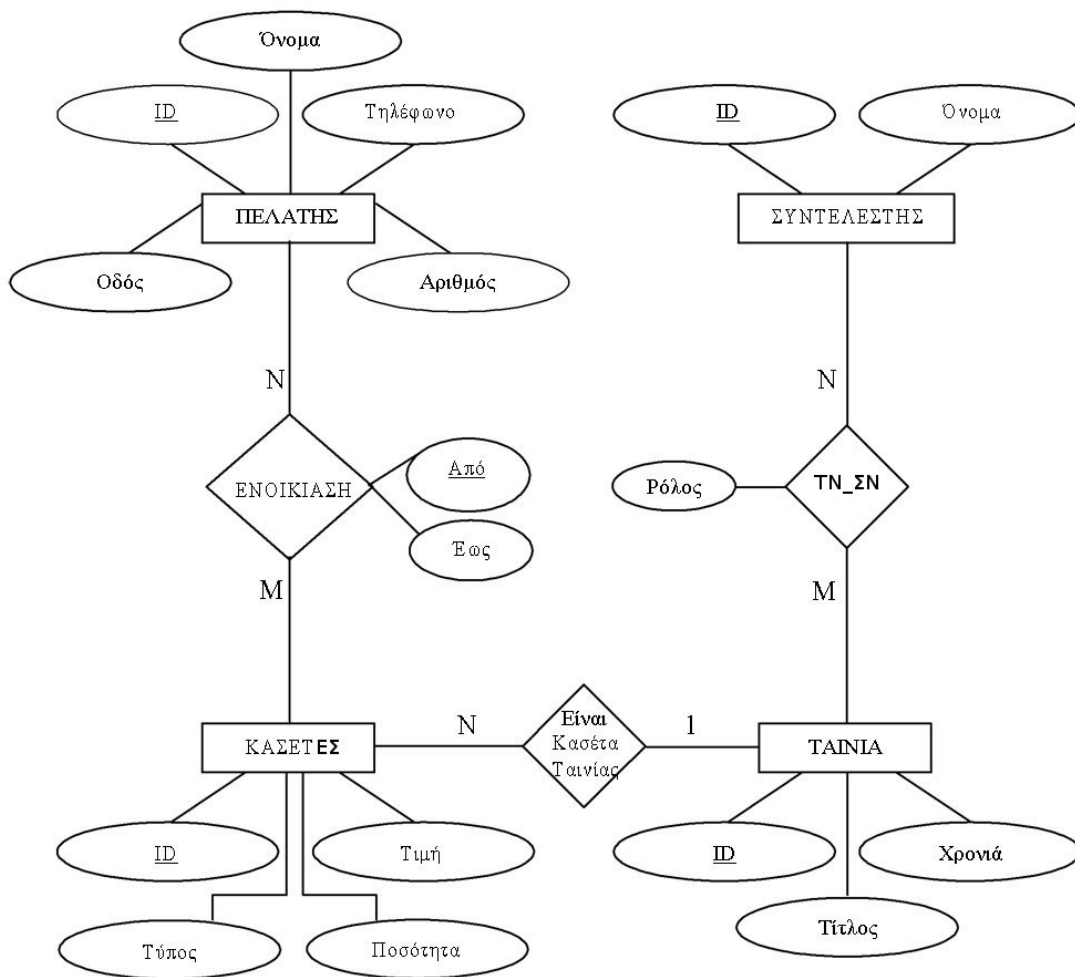
Στο παρόν κεφάλαιο θα δημιουργήσουμε μια βάση δεδομένων που αφορά ένα Video Club. Θα εργαστούμε κυρίως με εντολές από την γλώσσα ορισμού δεδομένων (Data Definition Language). Θα συζητήσουμε για τους διαφορετικούς τύπους δεδομένων, τον τρόπο δημιουργίας πινάκων με γραφικό τρόπο και με κώδικα. Τέλος, θα δημιουργήσουμε συσχετίσεις μεταξύ των πινάκων στο Database diagram και θα εισάγουμε τιμές στους πίνακες.

2.1. Δημιουργία Βάσης Δεδομένων

Ο SQL Server παρέχει δύο μεθόδους για να δημιουργήσετε μια βάση δεδομένων:

- Το SQL Server Management Studio (Γραφικό περιβάλλον)
- Εντολές της SQL

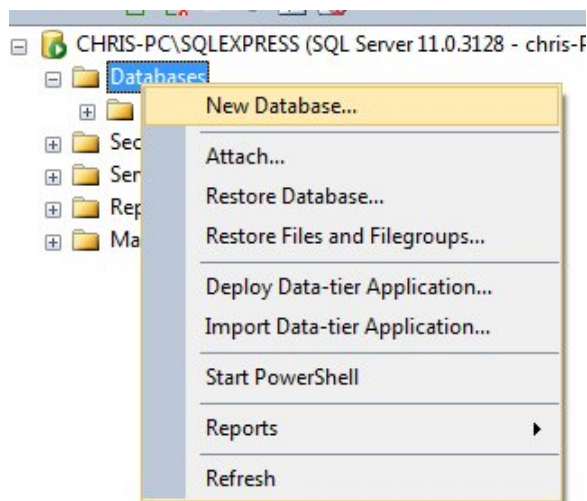
Στη συνέχεια θα χρησιμοποιήσουμε και τις δύο μεθόδους για να δημιουργήσουμε τη βάση δεδομένων και τους πίνακες ενός Video Club, το διάγραμμα οντοτήτων-συσχετίσεων του οποίου δίνεται στην Εικόνα 2.1.



Εικόνα 2.1

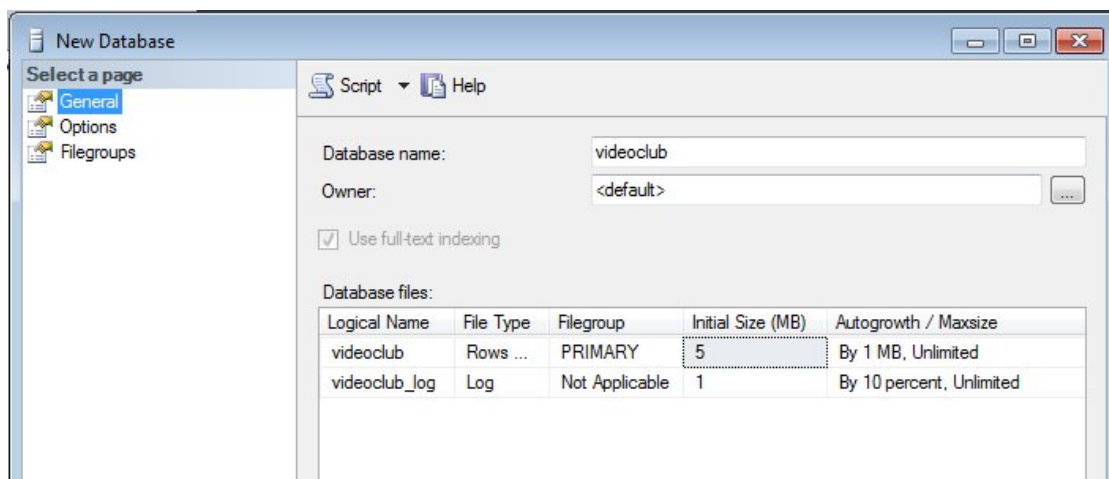
2.1.1. Δημιουργία νέας βάσης σε γραφικό περιβάλλον

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



Εικόνα 2.2

δίνουμε το όνομα της βάσης δεδομένων μας (videoclub), όπως φαίνεται στην Εικόνα 2.3.



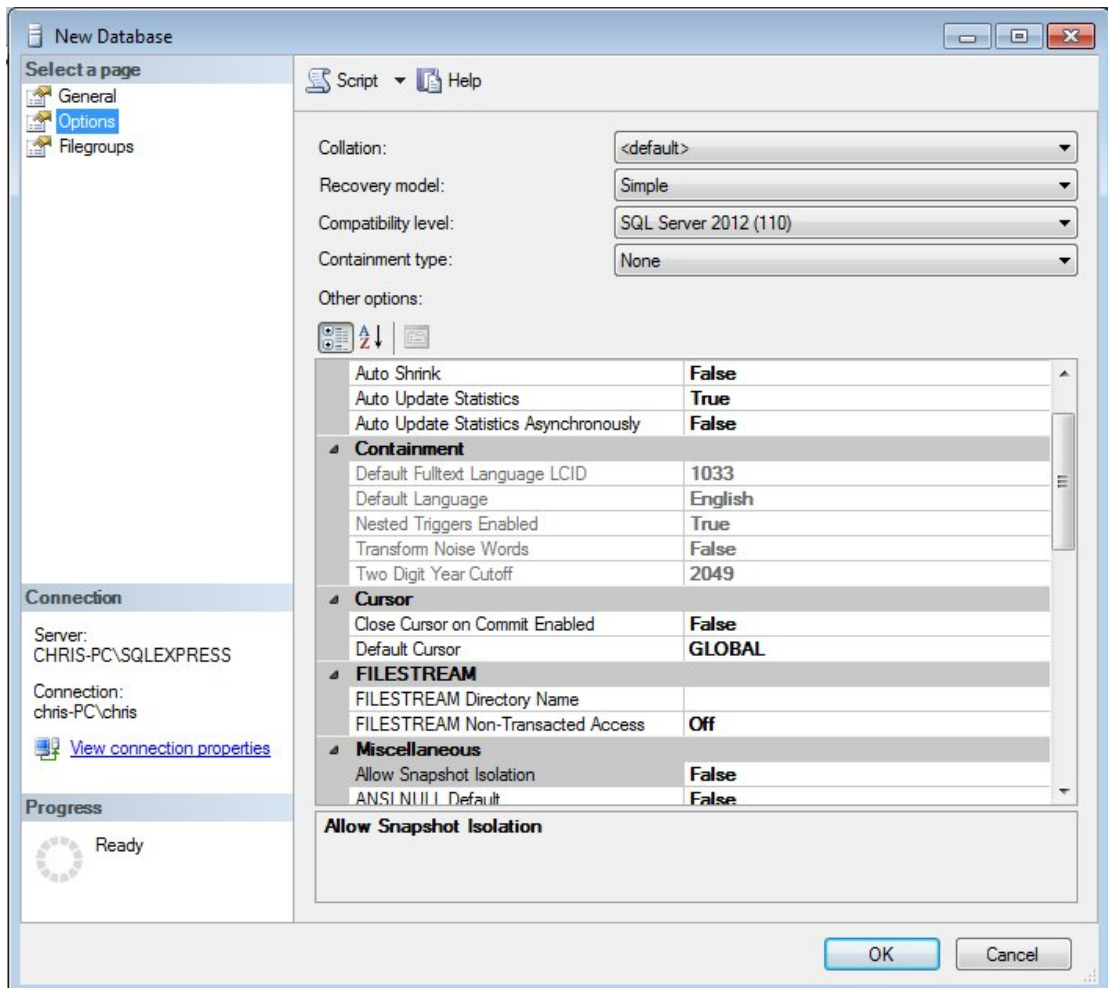
Εικόνα 2.3

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. Μπορείτε να αποδεχθείτε το όνομα ή να πληκτρολογήσετε 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.

Στην Εικόνα 2.3, εφόσον κάνετε κλικ στην επιλογή «Options», μπορείτε να καθορίσετε πιο εξειδικευμένες ρυθμίσεις που εμφανίζονται στην Εικόνα 2.4.

Οι πιο σημαντικές είναι οι παρακάτω:

Compatibility level: Αν θέλετε να μεταφέρετε την βάση δεδομένων που θα φτιάξετε ή θέλετε να είναι προσβάσιμη από άλλες εφαρμογές που δεν έχουν την δυνατότητα να συνδεθούν στο SQL Server 2012 μπορείτε να επιλέξετε συμβατότητα με προηγούμενες εκδόσεις όπως δείχνουν οι επιλογές της Εικόνας 2.5.

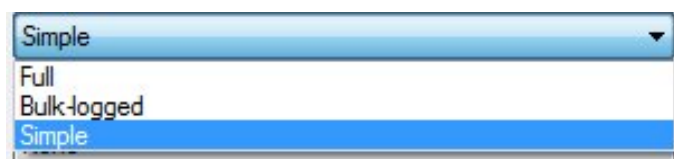


Εικόνα 2.4



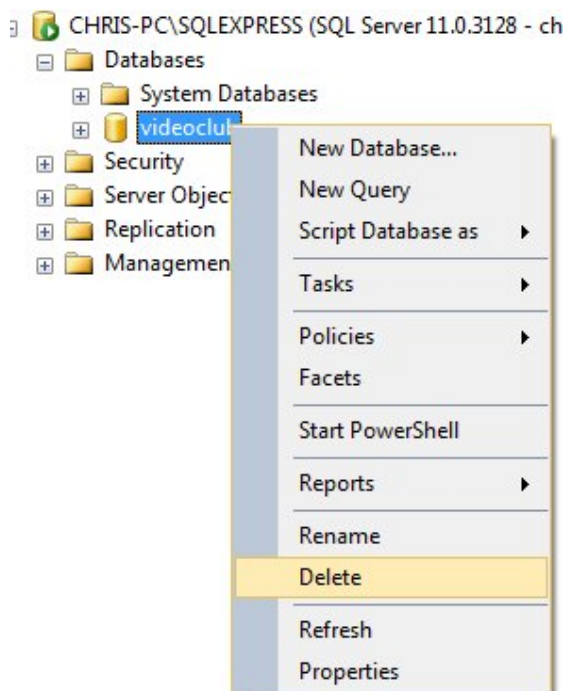
Εικόνα 2.5

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 έχουμε και τα ανάλογα back-ups του log file και καταλαμβάνει περισσότερο χώρο στο δίσκο.



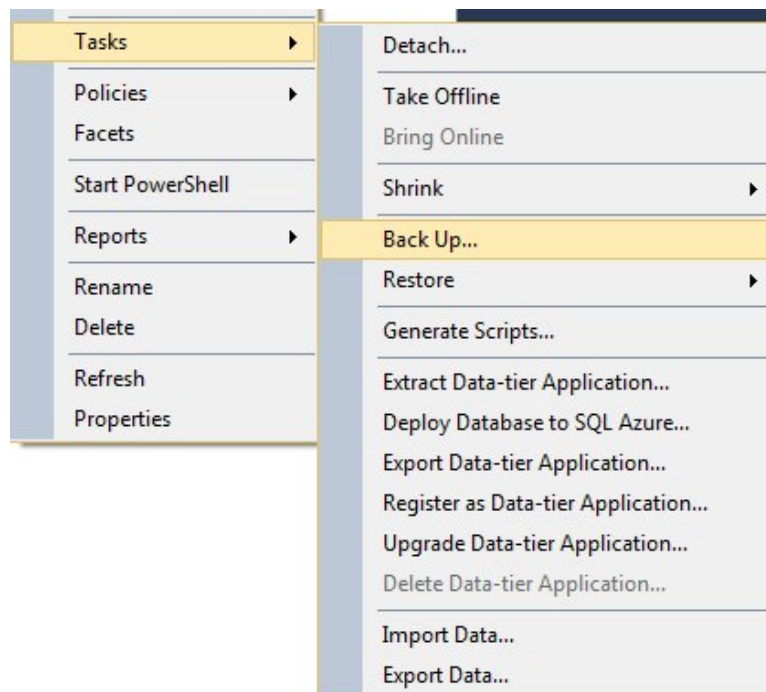
Εικόνα 2.6

Για να αποθηκευτεί η βάση δεδομένων μας (videoclub), κάνουμε κλικ στο κουμπί ΟΚ(βλέπε Εικόνας 2.4). Τώρα, κάνοντας δεξί κλικ πάνω στη βάση δεδομένων μας βλέπουμε τις διαθέσιμες ενέργειες που εμφανίζονται στην Εικόνα 2.7.



Εικόνα 2.7

Στην επιλογή «Properties» μπορούμε να αλλάξουμε τις βασικές ρυθμίσεις που μόλις ορίσαμε κατά την δημιουργία της. Επιπλέον, το υπομενού Tasks παρέχει πολλές χρήσιμες επιλογές που εμφανίζονται στην Εικόνα 2.8.



Εικόνα 2.8

Πιο σημαντική είναι η διαδικασία του «Back Up» όπου θα επιλέξουμε το όνομα του αρχείου προκειμένου να αποθηκευτεί κάθε φορά ένα αρχείο με κατάληξη «bak» που θα περιέχει όλα τα δεδομένα

της βάσης μας. Η αντίστοιχη επιλογή του “restore” χρειάζεται για να επαναφέρουμε μία 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.

2.1.2. Δημιουργία νέας βάσης με κώδικα SQL

Επιλέγοντας Generate Scripts, όπως φαίνεται στην Εικόνα 2.8, ανοίγει ένας οδηγός όπου αυτόματα παράγονται οι ανάλογες εντολές SQL που μπορούν να δημιουργήσουν την βάση που μόλις φτιάξαμε με γραφικό τρόπο.



```
SQLQuery1.sql - CH...hris-PC\chris (54) X
USE [master]
GO

/***** Object: Database [videoclub]    Script Date: 22/5/2013 8:51:
CREATE DATABASE [videoclub]
    CONTAINMENT = NONE
    ON PRIMARY
    ( NAME = N'videoclub', FILENAME = N'C:\Program Files\Microsoft SQL Ser
    LOG ON
    ( NAME = N'videoclub_log', FILENAME = N'C:\Program Files\Microsoft SQ
GO

ALTER DATABASE [videoclub] SET COMPATIBILITY_LEVEL = 110
GO

IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [videoclub].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO

ALTER DATABASE [videoclub] SET ANSI_NULL_DEFAULT OFF
GO

ALTER DATABASE [videoclub] SET ANSI_NULLS OFF
GO

ALTER DATABASE [videoclub] SET ANSI_PADDING OFF
GO

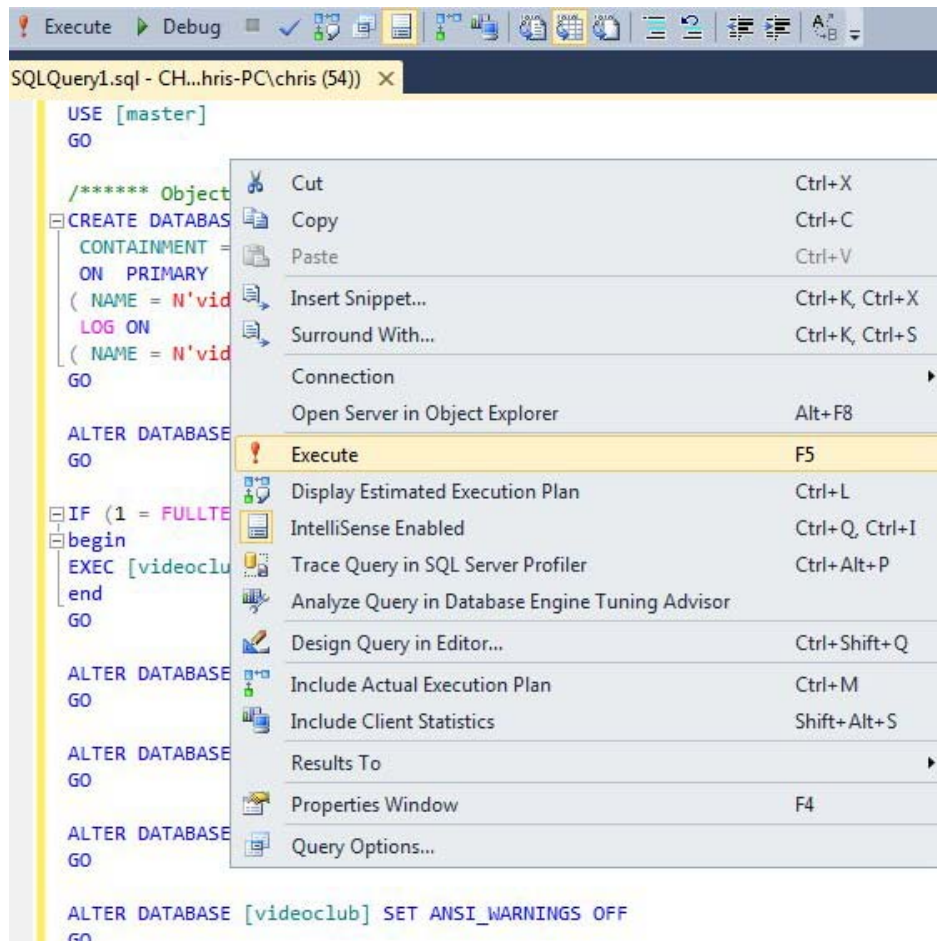
ALTER DATABASE [videoclub] SET ANSI_WARNINGS OFF
GO

ALTER DATABASE [videoclub] SET ARITHABORT OFF
GO

ALTER DATABASE [videoclub] SET AUTO_CLOSE OFF
GO
```

Εικόνα 2.9

Προκειμένου να εκτελέσουμε τον κώδικα SQL και να δημιουργήσουμε την βάση δεδομένων videoclub, θα πρέπει να κάνουμε κλικ στην εργαλειοθήκη στο κουμπί «New Query» για να ανοίξει ένας Query Editor. Στην συνέχεια, θα επικολλήσουμε τον κώδικα SQL και θα εκτελέσουμε τις εντολές είτε με F5 στο πληκτρολόγιο, είτε κάνοντας κλικ στο κουμπί “EXECUTE” όπως φαίνεται στην Εικόνα 2.10.



Εικόνα 2.10

2.2. Δημιουργία Πινάκων

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:

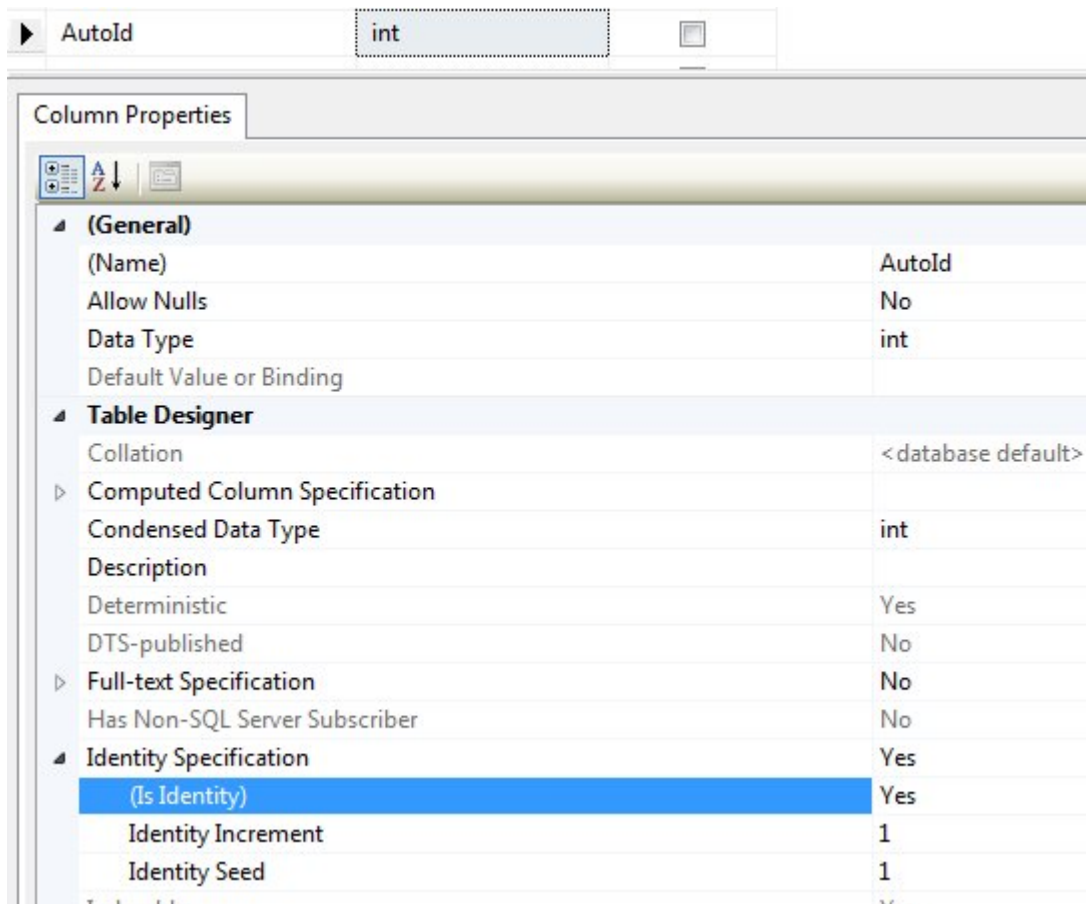
Κατηγορία	Περιγραφή	Τύπος δεδομένων	Περιγραφή
Character	Τα δεδομένα χαρακτήρων αποτελούνται από οποιοδήποτε συνδυασμό γραμμάτων, συμβόλων, και αριθμητικών δεδομένων. Για παράδειγμα, έγκυρα δεδομένα χαρακτήρων είναι οι συνδυασμοί "John123" και "(0\$%b99"	<i>Char</i>	Τα δεδομένα έχουν το ίδιο σταθερό μήκος (μέχρι 8KB)
		<i>varchar</i>	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.	<i>nchar</i>	Τα δεδομένα πρέπει να έχουν το ίδιο σταθερό μήκος(μέχρι 4000 Unicode χαρακτήρες)
		<i>nvarchar</i>	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.	<i>datetime</i>	Τα δεδομένα κυμαίνονται από 1 ^η Ιανουαρίου 1753 μέχρι την 31 ^η Δεκεμβρίου 9999. Συμπεριλαμβάνεται και η ώρα. Π.χ: 2007-05-08 12:35:29
		<i>smalldatetime</i>	Τα δεδομένα κυμαίνονται από 1 Ιανουαρίου 1900 μέχρι την 6 ^η Ιουνίου 2079
		<i>Date</i>	Αποθηκεύεται μόνο η ημερομηνία χωρίς την ώρα.π.χ: 2007-05-08
Integer	Αποτελούνται από αρνητικούς και θετικούς αριθμούς όπως τους -15, 0, 5 και 2.507.	<i>bigint</i>	Τα δεδομένα είναι αριθμοί από -2 ⁶³ μέχρι το 2 ⁶³ -1. (8 bytes)
		<i>int</i>	Τα δεδομένα είναι αριθμοί από το -2.147.483.648 μέχρι το 2.147.483.647 (4 bytes)
		<i>smallint</i>	Τα δεδομένα είναι αριθμοί από το -32.768 μέχρι το 32.767. (2 bytes)
		<i>tinyint</i>	Τα δεδομένα είναι αριθμοί από το 0 μέχρι το 255. (1 byte)
Decimal	Δεκαδικός αριθμός με απόλυτη ακρίβεια.	<i>decimal - numeric</i>	Παράμετροι 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	<i>float</i>	Τα δεδομένα είναι floating-point αριθμοί από -1,79E + 308 μέχρι 1,79E + 308.
		<i>real</i>	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 με ελάχιστες διαφορές που κάνουν την ίδια δουλειά και μόνο σε πολύ εξειδικευμένο στάδιο μπορούν να μας προκαλέσουν προβλήματα ή ζητήματα απόδοσης. Κάποιες χρήσιμες 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 θέλετε η εφαρμογή σας να υποστηρίζει την δυνατότητα του χρήστη να καταχωρεί Unicode χαρακτήρες (π.χ γαλλικά ή ρωσικά ονόματα) πρέπει να χρησιμοποιήσετε τον τύπο `nvarchar` και όχι `varchar`.
- Εφόσον θέλετε να αποθηκεύσετε μόνο την ημερομηνία, χρησιμοποιήστε τον τύπο `Date` και όχι τον `datetime` που αποθηκεύει και την ώρα. Ο δεύτερος τύπος δεδομένων σπαταλά χώρο αλλά μπορεί να σας δημιουργήσει και προβλήματα σε ερωτήματα SQL. Για παράδειγμα, χρησιμοποιώντας τους τελεστές μεγαλύτερο ή μικρότερο χάνετε μία μέρα αν δεν βάλετε και την ώρα (π.χ. `MydateField <= '01/31/2012 23:59'`).
- Εφόσον ένα πεδίο χρησιμοποιείται σαν κύριο κλειδί ή σαν ξένο κλειδί (τμήμα συσχέτισης) χρησιμοποιείται 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 τύπος `decimal (9,2)` ή `decimal (18,2)` για μεγαλύτερα ποσά που είναι απόλυτα ακριβής.
- Πολύ πρακτική και χρήσιμη είναι η λειτουργία ενός αυτόματου κλειδιού. Αυτό όπως φαίνεται στην Εικόνα 2.11 παρακάτω υλοποιείται με ένα πεδίο τύπου `int`, όπου στις ιδιότητές του 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.



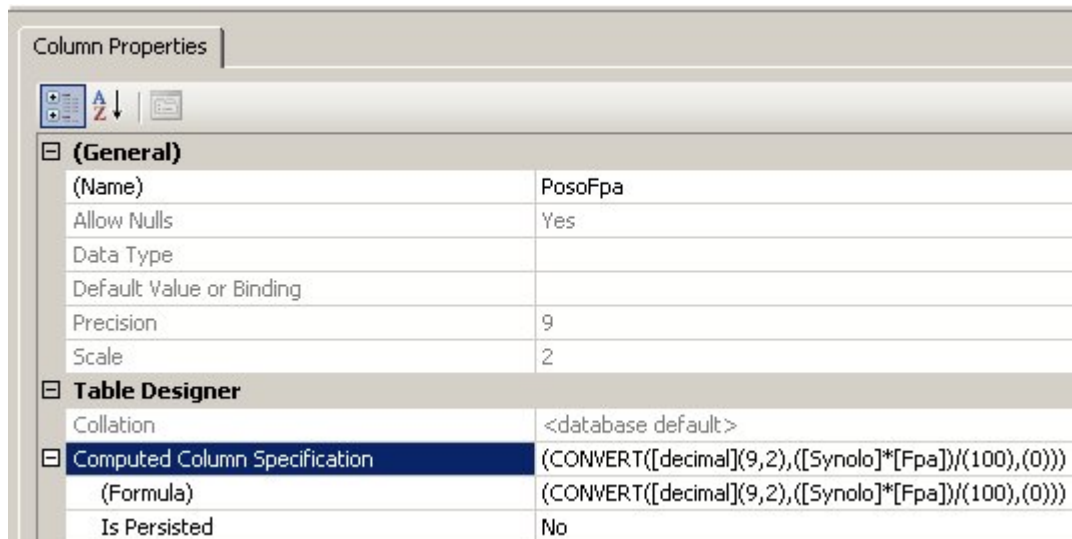
Εικόνα 2.11

- Πολύ χρήσιμος είναι ο τύπος πεδίου (υπολογιζόμενο πεδίο). Πρόκειται για ένα πεδίο που δεν μπορεί να το καταχωρεί – τροποποιεί ο χρήστης γιατί απλά αποτελεί το αποτέλεσμα μίας πράξης μεταξύ άλλων πεδίων. Για παράδειγμα έστω ότι έχουμε 2 αριθμητικά πεδία το σύνολο αξίας (Synolo) και τον συντελεστή ΦΠΑ (Fpa) όπως φαίνεται παρακάτω.

<input type="checkbox"/>	Synolo	decimal(9, 2)	<input type="checkbox"/>
<input type="checkbox"/>	Fpa	decimal(5, 2)	<input type="checkbox"/>
<input checked="" type="checkbox"/>	PosoFpa		<input checked="" type="checkbox"/>

Εικόνα 2.12

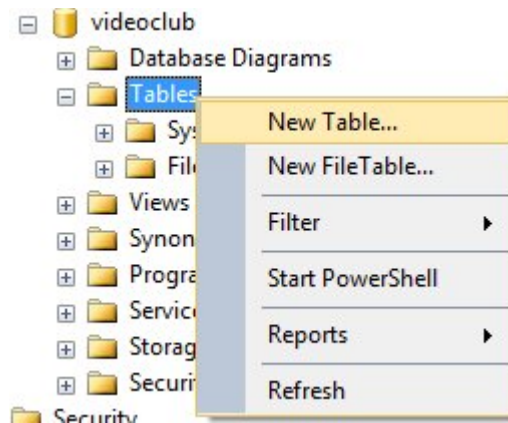
- Τέλος, μπορούμε στο επίπεδο της βάσης δεδομένων (για απόλυτη αξιοπιστία) και όχι στην εφαρμογή που θα φτιάξουμε να έχουμε αυτόματα σαν πεδίο το ποσό ΦΠΑ. Όπως φαίνεται στην Εικόνα 2.13, στο πεδίο Computed Column Specification στο τμήμα formula γράφουμε το τύπο υπολογισμού : $[Synolo]*[Fpa]$ ή αν θέλουμε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



Εικόνα 2.13

2.2.3. Δημιουργία πινάκων με τον Management Studio

Προκειμένου να δημιουργήσουμε ένα νέο πίνακα στην βάση μας με γραφικό τρόπο, πηγαίνουμε στη δενδρική δομή, στο αριστερό pane του Management Studio και επεκτείνουμε τη βάση δεδομένων videoclub, όπως φαίνεται στην Εικόνα 2.14. Στη συνέχεια, επεκτείνουμε το φάκελο Tables. Βλέπουμε τους πίνακες που δημιουργεί ο SQL Server για κάθε νέα βάση. Δεξί κλικ στο δεξιό τμήμα και επιλέγουμε “New Table...”



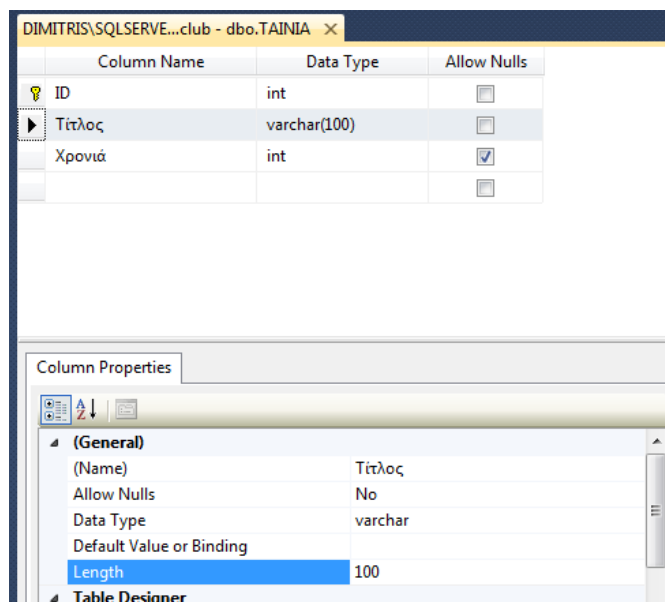
Εικόνα 2.14

Θα δημιουργήσουμε τον πίνακα TAINIA με τα ακόλουθα πεδία:

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

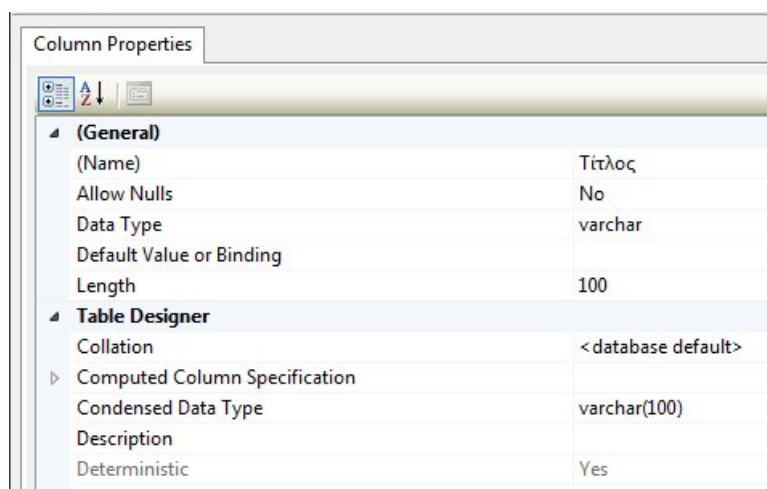
Πίνακας 2.2 TAINIA

Ανοίγει ο Table Designer, όπως φαίνεται στην Εικόνα 2.15, ο οποίος σας επιτρέπει να σχεδιάσετε έναν πίνακα με γραφικό τρόπο. Συμπληρώνουμε τα στοιχεία του Πίνακα 2.2, όπως φαίνεται ακολούθως 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. Επίσης, δίνουμε προσοχή σε κάθε πεδίο του πίνακα για το αν θα επιτρέπουμε NULL τιμές (π.χ. δείτε το tick στην τελευταία στήλη της Εικόνας 2.15).



Εικόνα 2.15

Παρατηρήστε στην Εικόνα 2.16, τις ιδιότητες του υπό εξέταση κάθε φορά πεδίου του πίνακα TAINIA (π.χ. 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 TAINIA στον πίνακα και κλείνουμε το παράθυρο δημιουργίας πίνακα. Στη λίστα με τους πίνακες της βάσης videoclub, μπορούμε να δούμε πλέον τον πίνακα που δημιουργήσαμε.



Εικόνα 2.16

Επαναλάβετε 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

Θα δημιουργήσουμε τον πίνακα ΣΥΝΤΕΛΕΣΤΗΣ, με την ακόλουθη μορφή:

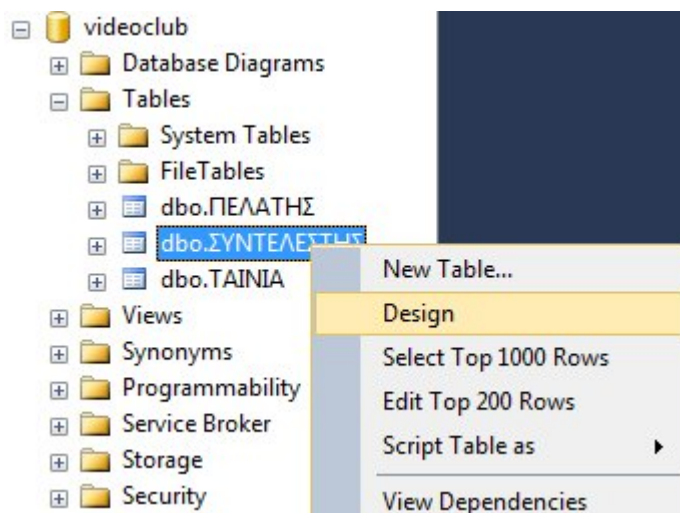
Field	Type	Null	Key
ID	Int		PRI
Όνομα	Varchar(50)		

Πίνακας 2.4 ΣΥΝΤΕΛΕΣΤΗΣ

Ανοίγουμε τον Query Editor και προσέχουμε η ενεργή/επιλεγμένη βάση να είναι η videoclub. Συνεπώς, μπορούμε να πληκτρολογήσουμε τον παρακάτω κώδικα για να δημιουργήσουμε τον πίνακα ΣΥΝΤΕΛΕΣΤΗΣ:

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

Για να εκτελέσουμε τον παραπάνω κώδικα πατάμε είτε στην εργαλειοθήκη το κουμπί Execute είτε στο πληκτρολόγιο F5. Κάνοντας δεξί κλικ – Refresh πάνω από τον φάκελο Tables, βλέπουμε τους 3 πίνακες που έχουμε δημιουργήσει. Αν θέλουμε να αλλάξουμε οτιδήποτε στην σχεδίασή ενός πίνακα τον επιλέγουμε με δεξί κλικ και κάνουμε κλικ στην επιλογή Design.



Εικόνα 2.17

Στη συνέχεια, θα δημιουργήσουμε τον πίνακα ΚΑΣΕΤΕΣ. Επιστρέφουμε στον Query Editor. Προσέχουμε η επιλεγμένη βάση να είναι η videoclub. Σβήνουμε από τον editor όλα τα περιεχόμενα. Ο πίνακας ΚΑΣΕΤΕΣ έχει την ακόλουθη μορφή:

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

Πίνακας 2.5 ΚΑΣΕΤΕΣ

Πληκτρολογούμε τον παρακάτω κώδικα για να δημιουργήσουμε τον πίνακα ΚΑΣΕΤΕΣ:

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

Ο περιορισμός ξένου κλειδιού (FOREIGN KEY constraint) ορίζει μια συσχέτιση μεταξύ δύο πινάκων. 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 ΚΑΣΕΤΕΣ, 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 ΚΑΣΕΤΕΣ, 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:

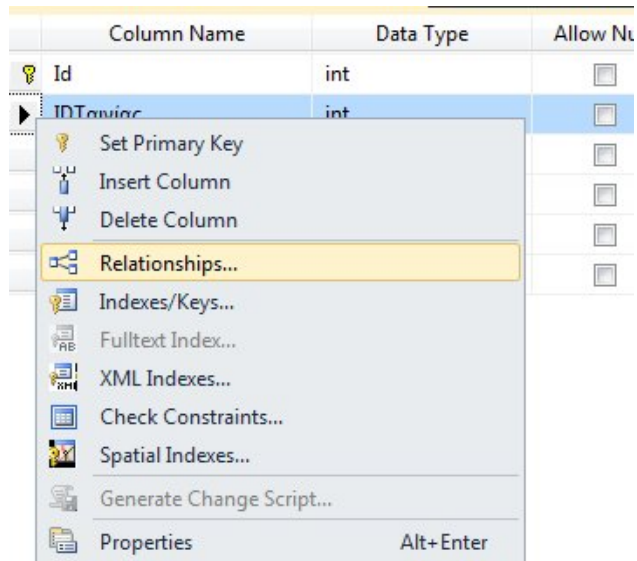
- **NO ACTION:** αποτρέπεται η διαγραφή και προβάλλεται μήνυμα λάθους.
- **CASCADE:** διαγράφει την εγγραφή και προκαλεί διαγραφή όλων των εγγραφών με την ίδια τιμή στον πίνακα ΚΑΣΕΤΕΣ.
- **Set Null:** διαγράφει την εγγραφή και στις εξαρτώμενες εγγραφές θα εισάγει την τιμή Null.
- **Set Default:** διαγράφει την εγγραφή και στις εξαρτώμενες εγγραφές θα εισάγει την default τιμή που ορίσουμε κατά την δημιουργία του πεδίου.

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

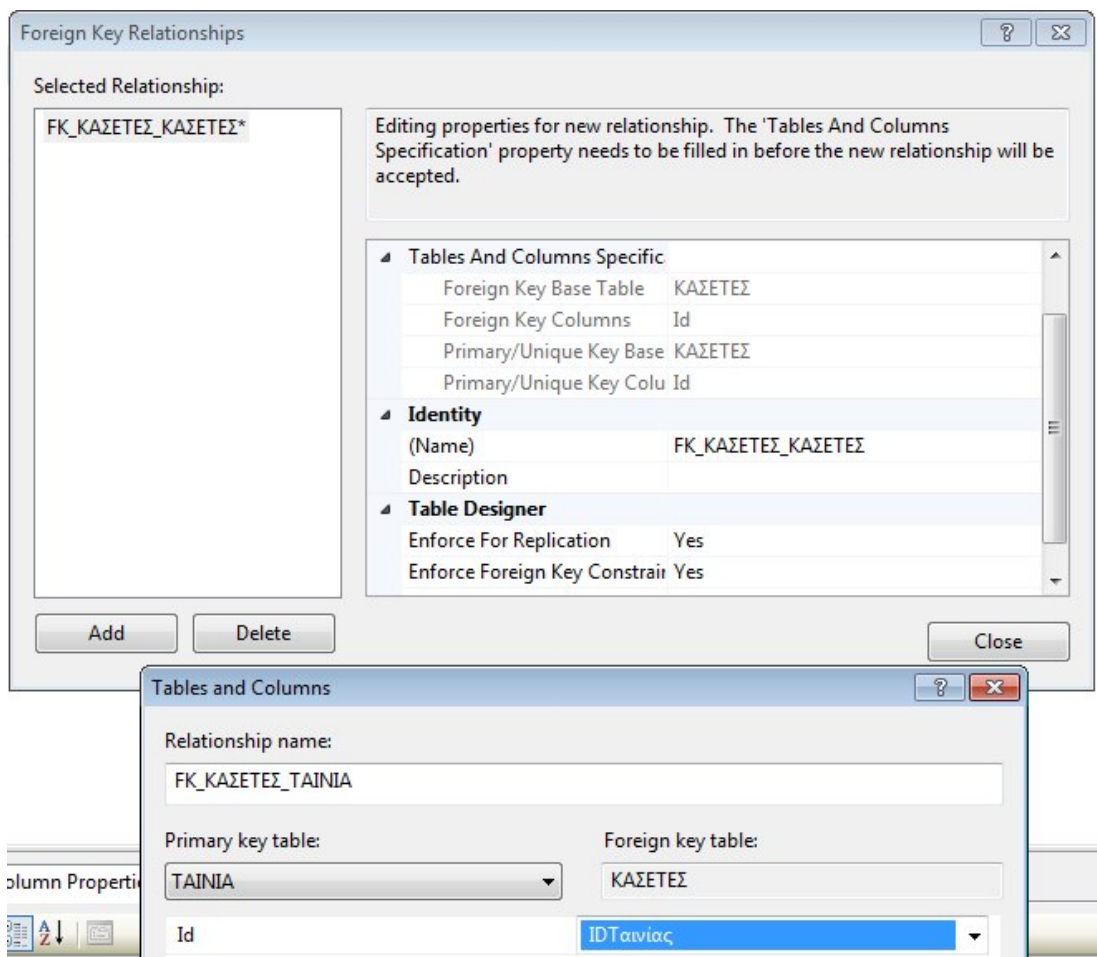
2.2.5. Συσχετίσεις/Relationships Πινάκων

Όπως συζητήσαμε στη Ενότητα 2.2.4, ο πίνακας ΚΑΣΕΤΕΣ σχετίζεται με τον πίνακα ΤΑΙΝΙΑ μέσω του πεδίου IDΤαινίας. Προκειμένου να δημιουργήσουμε αυτή την συσχέτιση με γραφικό τρόπο, επεκτείνουμε το φάκελο Tables και κάνουμε δεξί κλικ πάνω στον πίνακα ΚΑΣΕΤΕΣ και κλικ στην επιλογή Design. Ανοίγει ο Table Designer και με δεξί κλικ οπουδήποτε πάνω στην σχεδίαση επιλέγουμε Relationships όπως φαίνεται στην Εικόνα 2.18.

Στην συνέχεια εμφανίζεται το αρχικό παράθυρο της Εικόνας 2.19 όπου πατώντας στο κουμπί ADD μπορούμε να δημιουργήσουμε την συσχέτιση με γραφικό τρόπο. Καταρχήν μετονομάζουμε την συσχέτιση σε FK_ΚΑΣΕΤΕΣ_ΤΑΙΝΙΑ και στην 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 (ΚΑΣΕΤΕΣ.IDΤαινίας).



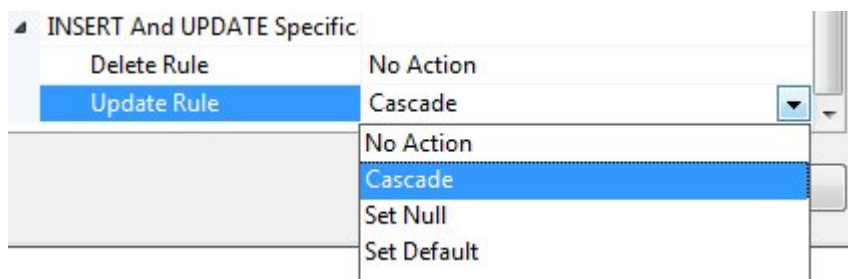
Εικόνα 2.18



Εικόνα 2.19

Με αυτόν τον τρόπο, στον πίνακα ΚΑΣΕΤΕΣ, 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 ΚΑΣΕΤΕΣ, αν αυτή προηγουμένως δεν έχει εισαχθεί στο πεδίο ID του πίνακα ΤΑΙΝΙΑ.

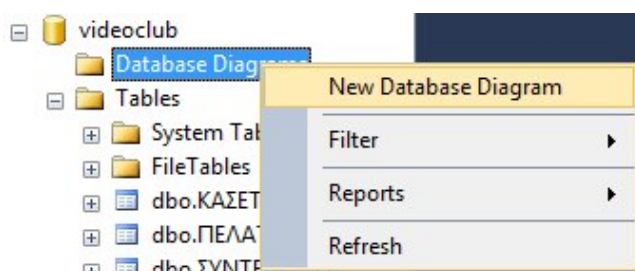
Τέλος, όπως φαίνεται στην Εικόνα 2.20, μπορούμε και με γραφικό τρόπο να κάνουμε χρήση του DELETE Rule με επιλογές παρόμοιες με αυτές που αναφέρθηκαν στην Ενότητα 2.2.4 (π.χ. No Action, Cascade, Set Null, Set Default). Αντίστοιχα, η χρήση της ρύθμισης Update Rule αφορά την περίπτωση που γίνεται update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



Εικόνα 2.20

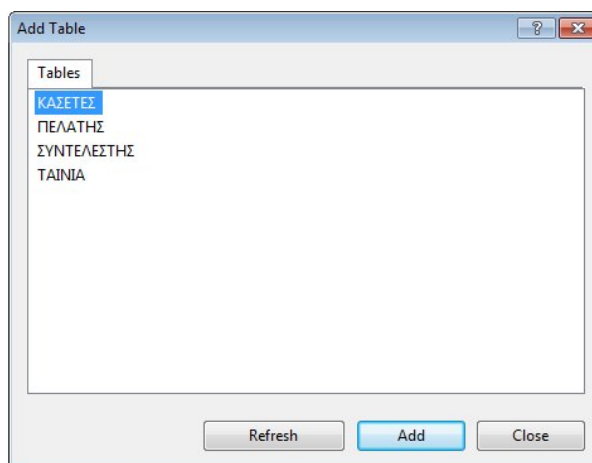
2.2.6 Δημιουργία πινάκων με τον Database Diagrams

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



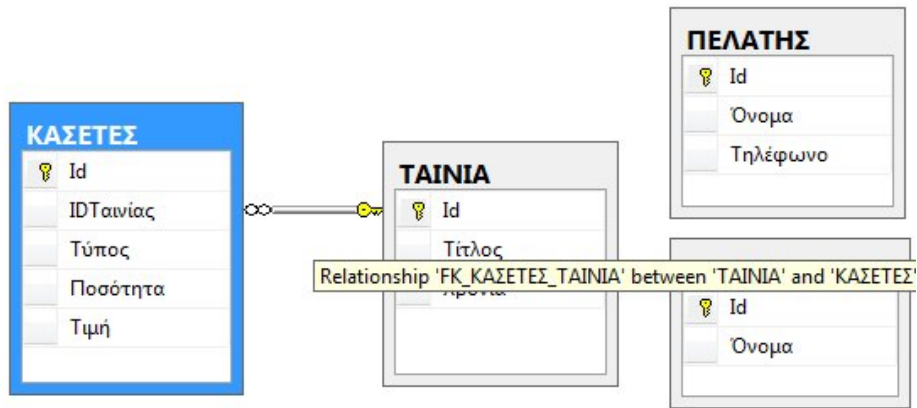
Εικόνα 2.21

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



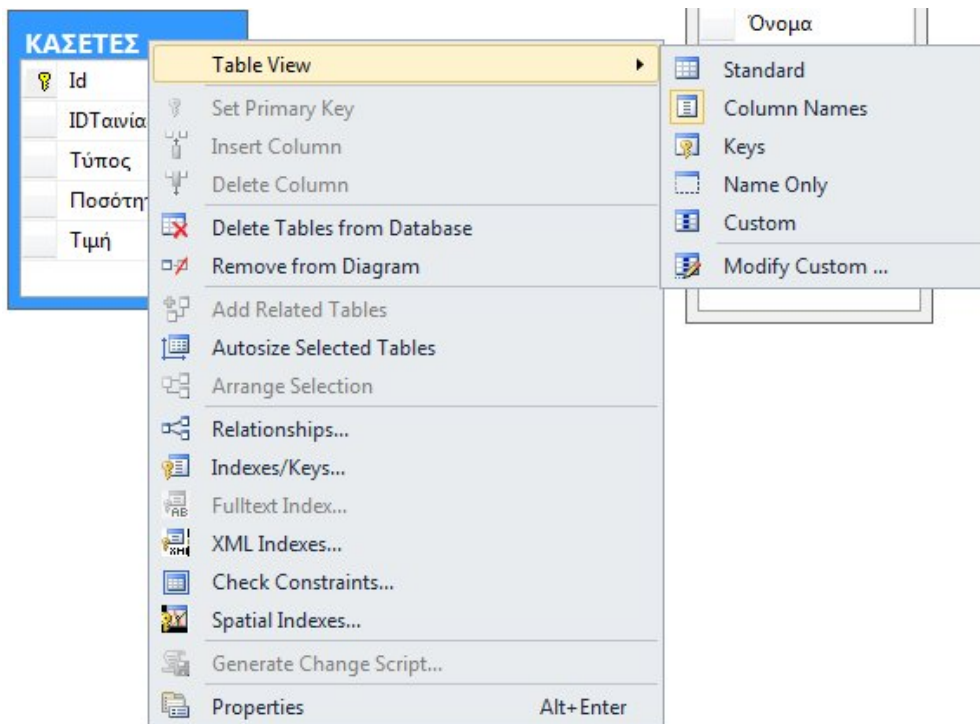
Εικόνα 2.22

Με το ποντίκι 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 μίας 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

Προχωρούμε στην δημιουργία του πίνακα ΕΝΟΙΚΙΑΣΗ, που έχει την ακόλουθη μορφή:

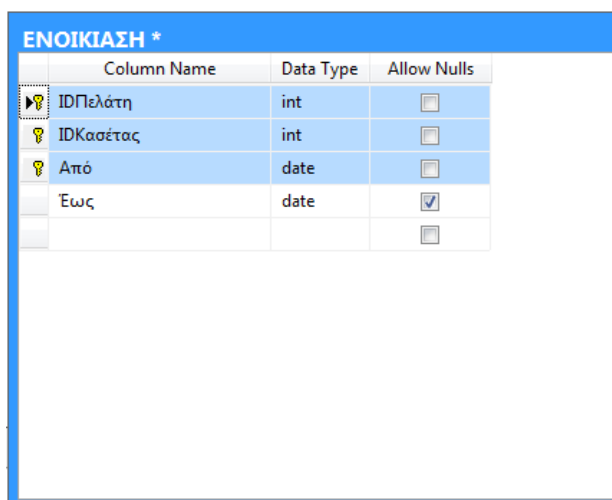
Field	Type	Null	Key
IDΠελάτη	Int		PRI
IDΚασέτας	int		PRI
Από	date		PRI
Έως	date	YES	

Πίνακας 2.6 ΕΝΟΙΚΙΑΣΗ

Στον πίνακα ΕΝΟΙΚΙΑΣΗ, το κύριο κλειδί αποτελείται από περισσότερα του ενός πεδία (IDΠελάτη, IDΚασέτας, Από). Σε αυτήν την περίπτωση, επιτρέπεται να υπάρχουν διπλές τιμές στο ίδιο πεδίο. Κάθε όμως συνδυασμός των τιμών των τριών πεδίων που αποτελούν το κύριο κλειδί πρέπει να είναι μοναδικός.

Σημειώνεται ότι η στήλη IDΠελάτη είναι ξένο κλειδί προς τη στήλη ID του πίνακα ΠΕΛΑΤΗΣ, ενώ η στήλη IDΚασέτας είναι ξένο κλειδί προς τη στήλη ID του πίνακα ΚΑΣΕΤΕΣ.

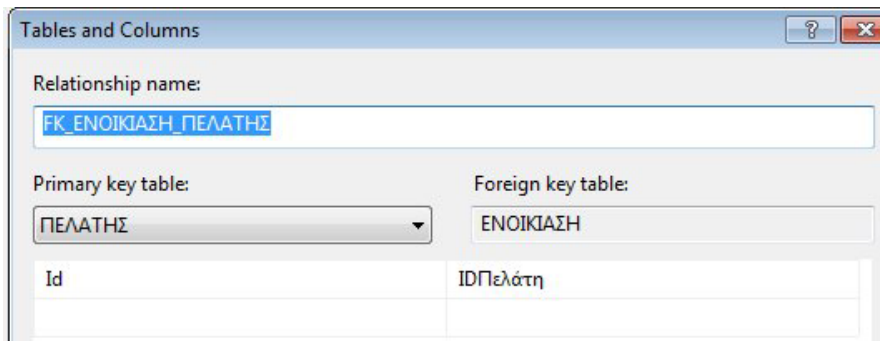
Προκειμένου να φτιάξετε τον πίνακα ΕΝΟΙΚΙΑΣΗ μέσα από το περιβάλλον του Database Diagrams, κάντε κλικ στο κουμπί **New Table** στην γραμμή εργαλείων ή κάντε δεξί κλικ στο διάγραμμα και επιλέξτε **New Table**. Στο παράθυρο διαλόγου Choose Name επιλέξτε το όνομα ΕΝΟΙΚΙΑΣΗ και πατήστε OK. Ανοίγει ένα παράθυρο που μοιάζει με αυτό του Table Designer. Προσθέτουμε τα στοιχεία για τα πεδία και ορίζουμε το κύριο κλειδί (επιλογή των τριών πεδίων κρατώντας πατημένος το πλήκτρο Ctrl, δεξί κλικ πάνω τους και κλικ στο Set Primary Key), όπως φαίνεται στην Εικόνα 2.25:



Column Name	Data Type	Allow Nulls
IDΠελάτη	int	<input type="checkbox"/>
IDΚασέτας	int	<input type="checkbox"/>
Από	date	<input type="checkbox"/>
Έως	date	<input checked="" type="checkbox"/>

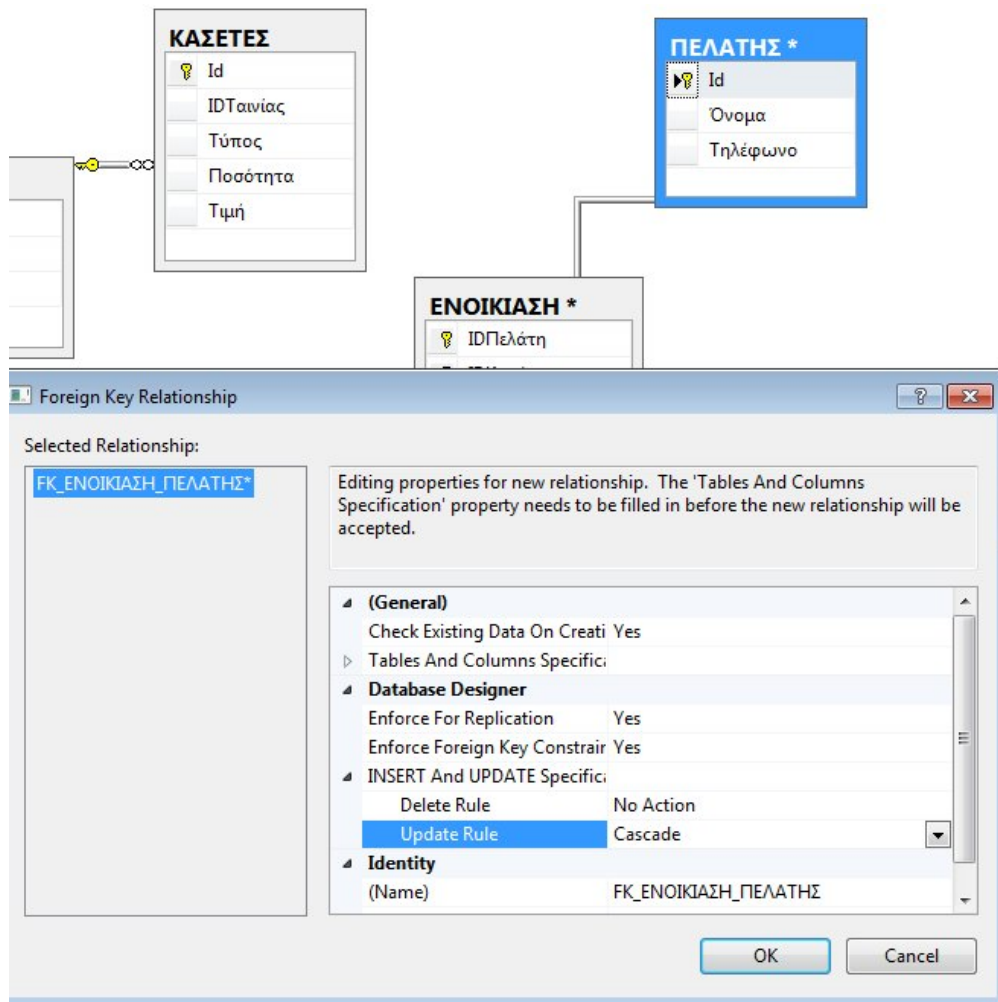
Εικόνα 2.25

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



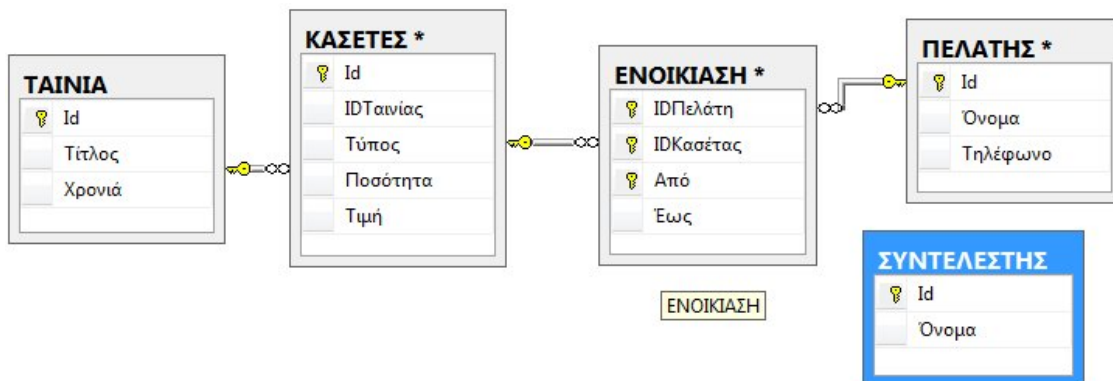
Εικόνα 2.26

Επιλέγουμε OK και στη συνέχεια επεκτείνοντας 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 ΚΑΣΕΤΕΣ με τον πίνακα ΕΝΟΙΚΙΑΣΗ με βάση τα πεδία (ΚΑΣΕΤΕΣ.Id) και (ΕΝΟΙΚΙΑΣΗ.IDΚασέτας). Η τρέχουσα κατάσταση του Database Diagram φαίνεται στην Εικόνα 2.28.



Εικόνα 2.28

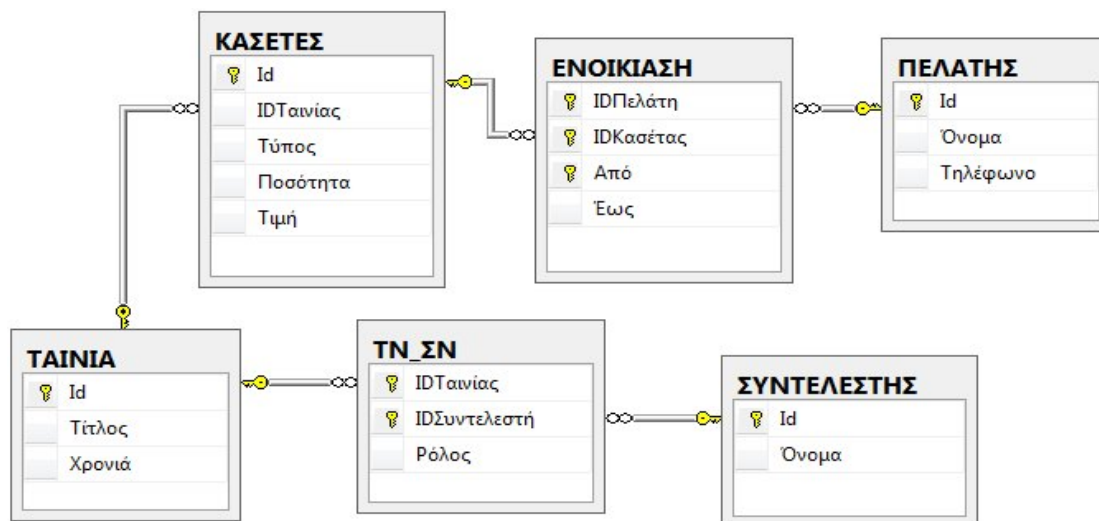
Τέλος θα εισάγουμε τον πίνακα που υλοποιεί τη δυαδική συσχέτιση TN_ΣΝ, μεταξύ του πίνακα ΤΑΙΝΙΑ και του πίνακα ΣΥΝΤΕΛΕΣΤΗΣ, που έχει την ακόλουθη μορφή.

Field	Type	Null	Key
IDΤαινίας	Int		PRI
IDΣυντελεστή	Int		PRI
Ρόλος	Varchar(30)	YES	

Πίνακας 2.7 TN_ΣΝ

ΠΡΟΣΟΧΗ! Λόγω του κύριου κλειδιού, ο ίδιος συντελεστής δεν μπορεί να συμμετέχει σε μία ταινία με περισσότερους από ένα ρόλους (π.χ. ηθοποιός και σκηνοθέτης ταυτόχρονα) κάτι που αποτελεί απλούστευση για της ανάγκες του παραδείγματος μας.

Σημειώνεται ότι η στήλη IDΤαινίας είναι ξένο κλειδί προς τη στήλη ID του πίνακα ΤΑΙΝΙΑ και η στήλη IDΣυντελεστή είναι ξένο κλειδί προς τη στήλη ID του πίνακα ΣΥΝΤΕΛΕΣΤΗΣ. Όπως περιγράφηκε παραπάνω ολοκληρώνουμε τις συσχετίσεις του πίνακα TN_ΣΝ. Το τελικό σχήμα της βάσης δεδομένων μας φαίνεται στην Εικόνα 2.29.



Εικόνα 2.29

2.3. Εισαγωγή τιμών στους πίνακες

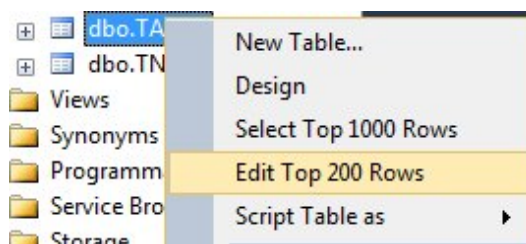
2.3.1. Εισαγωγή τιμών στους πίνακες με το Management Studio

Θα εισάγουμε τις ακόλουθες τιμές στον πίνακα TAINIA:

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

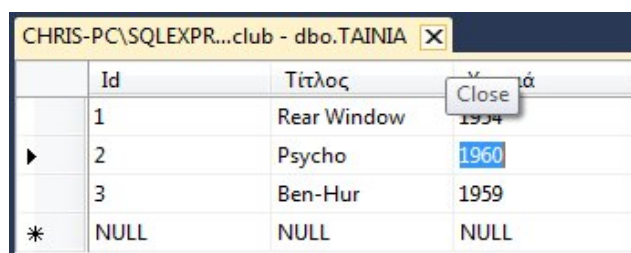
Πίνακας 2.8 TAINIA

Είμαστε στην κονσόλα του Management Studio και βλέπουμε τους πίνακες της βάσης videoclub. Κάνουμε δεξί κλικ στον πίνακα TAINIA. Επιλέξτε την επιλογή Edit Top 200 Rows, όπως φαίνεται στην Εικόνα 2.30.



Εικόνα 2.30

Εισάγετε τα δεδομένα με τρόπο που θα εισάγατε τιμές σε ένα φύλλο του Excel, όπως φαίνεται στην Εικόνα 2.31. Εφόσον καταχωρείτε μία εγγραφή και δείκτης πηγαίνει στην επόμενη η εγγραφή σας έχει αποθηκευτεί.



	Id	Τίτλος	Χρονιά
	1	Rear Window	1954
▶	2	Psycho	1960
	3	Ben-Hur	1959
*	NULL	NULL	NULL

Εικόνα 2.31

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

ID	Όνομα
1	Alfred Hitchcock
2	Grace Kelly
3	Anthony Perkins

Πίνακας 2.9 ΣΥΝΤΕΛΕΣΤΗΣ

Επαναλαμβάνουμε τα προηγούμενα βήματα για να εισάγουμε τις ακόλουθες τιμές 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 τιμές στον πίνακα TN_ΣΝ:

IDΤαινίας	IDΣυντελεστή	Ρόλος
1	1	Σκηνοθέτης
2	1	Σκηνοθέτης
1	2	Ηθοποιός
2	3	Ηθοποιός

Πίνακας 2.11 TN_ΣΝ

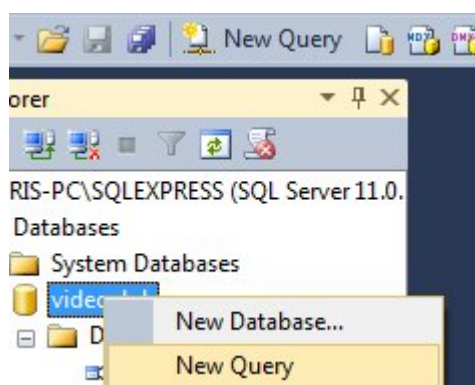
2.3.1. Εισαγωγή τιμών στους πίνακες με εντολές SQL

Η δήλωση INSERT, χρησιμοποιείται για να προσθέσει μια νέα γραμμή (ή γραμμές) σε έναν πίνακα. Θα εισάγουμε τις ακόλουθες τιμές στον πίνακα ΚΑΣΕΤΕΣ:

ID	IDΤαινίας	Τύπος	Ποσότητα	Τιμή
1	1	VHS	3	2
2	1	DVD	1	3
3	2	VHS	2	2

Πίνακας 2.12 ΚΑΣΕΤΕΣ

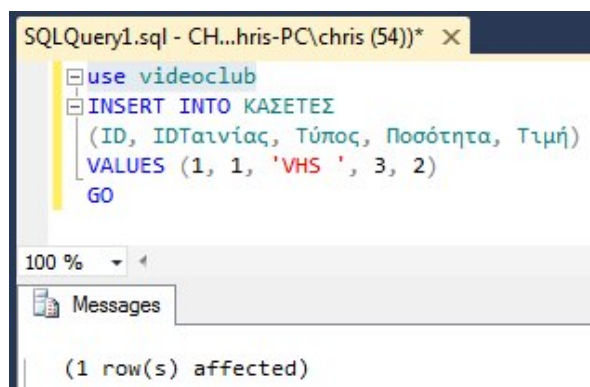
Μεταβαίνουμε στο Management Studio και επιλέγουμε από την μπάρα εργαλείων New Query ή δεξί κλικ στην database videoclub και New Query, όπως φαίνεται στην Εικόνα 2.32.



Εικόνα 2.32

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

```
INSERT INTO ΚΑΣΕΤΕΣ  
(ID, IDΤαινίας, Τύπος, Ποσότητα, Τιμή)  
VALUES (1, 1, 'VHS', 3, 2)  
GO
```



Εικόνα 2.33

Επιλέγουμε EXECUTE/F5 για εκτέλεση και στο παράθυρο Messages αν όλα πάνε καλά και εισαχθεί η εγγραφή θα δούμε το μήνυμα (1 row(s) affected), όπως φαίνεται στην Εικόνα 2.33.

ΠΡΟΣΟΧΗ! Προκειμένου να ξέρει το Management studio σε ποια βάση δεδομένων να εκτελέσει το ερώτημα πρέπει ή να δηλώσετε στην αρχή του ερωτήματος ότι χρησιμοποιείτε/εργάζεστε στην βάση δεδομένων videoclub (με την χρήση της δήλωσης «use videoclub») ή να δηλώνετε κάθε φορά την πλήρη διαδρομή ονόματος του κάθε πίνακα (π.χ videoclub.dbo.ΚΑΣΕΤΕΣ). Σημειώνεται ότι η δήλωση dbo είναι μία προκαθορισμένη συντόμευση του Management Studio για το ιδιοκτήτη της βάσης δεδομένων (DataBase Owner), ο οποίος έχει και τα πλήρη δικαιώματα διαχείρισης της υπό εξέτασης βάσης δεδομένων.

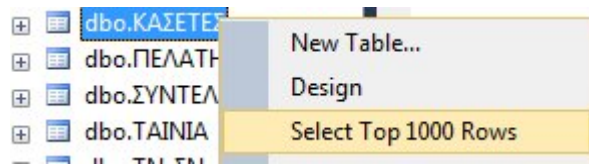
Η χρήση των ονομάτων των στηλών είναι προαιρετική, αρκεί να βάλουμε όλα τα ορίσματα με τη σωστή σειρά. Για παράδειγμα, η δεύτερη γραμμή μπορεί να εισαχθεί ως εξής:

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

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

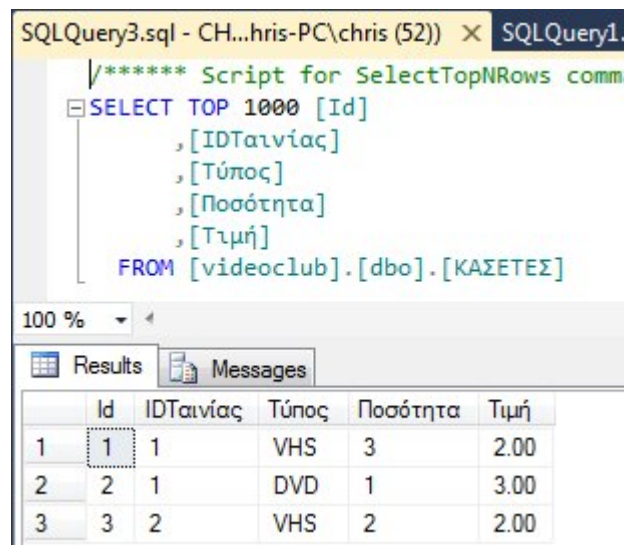
```
INSERT INTO ΚΑΣΕΤΕΣ  
(ID, IDΤαινίας, Τιμή, Ποσότητα, Τύπος)  
VALUES (3, 2, 2, 2, 'VHS')  
GO
```

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



Εικόνα 2.34

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



Εικόνα 2.35

Επαναλαμβάνουμε τα προηγούμενα βήματα για την εισαγωγή των ακόλουθων τιμών στον πίνακα ΕΝΟΙΚΙΑΣΗ (Προσοχή! πρώτα γράφουμε το μήνα και μετά την ημερομηνία.):

IDΠελάτη	IDΚασέτας	Από	Εώς
1	1	07/10/2006	09/10/2006
1	2	09/20/2006	11/20/2006
2	1	09/10/2006	Null

Πίνακας 2.13 ΕΝΟΙΚΙΑΣΗ

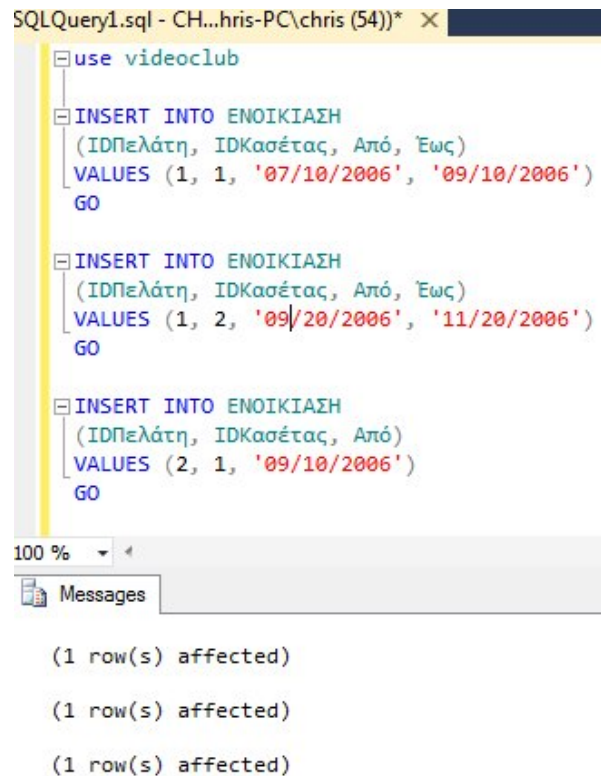
```
INSERT INTO ΕΝΟΙΚΙΑΣΗ
(IDΠελάτη, IDΚασέτας, Από, Έως)
VALUES (1, 1, '07/10/2006', '09/10/2006')
GO
```

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

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

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

Μπορείτε να δοκιμάσετε να εκτελέσετε όλες μαζί τις εντολές και όχι μία-μία. Το αποτέλεσμα πρέπει να φαίνεται όπως αυτό της Εικόνας 2.36:



```
SQLQuery1.sql - CH...hris-PC\chris (54))* X  
use videoclub  
INSERT INTO ΕΝΟΙΚΙΑΣΗ  
(IDΠελάτη, IDΚασέτας, Από, Έως)  
VALUES (1, 1, '07/10/2006', '09/10/2006')  
GO  
INSERT INTO ΕΝΟΙΚΙΑΣΗ  
(IDΠελάτη, IDΚασέτας, Από, Έως)  
VALUES (1, 2, '09/20/2006', '11/20/2006')  
GO  
INSERT INTO ΕΝΟΙΚΙΑΣΗ  
(IDΠελάτη, IDΚασέτας, Από)  
VALUES (2, 1, '09/10/2006')  
GO  
100 %  
Messages  
(1 row(s) affected)  
(1 row(s) affected)  
(1 row(s) affected)
```

Εικόνα 2.36

Κεφάλαιο 3. Ερωτήματα SQL

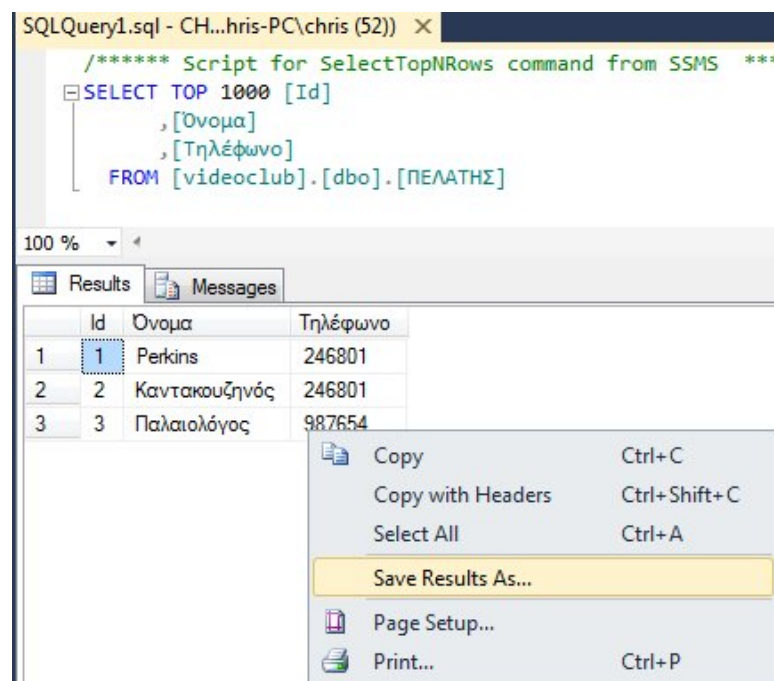
Σύνοψη

Στο παρόν κεφάλαιο θα παρουσιάσουμε βασικά και σύνθετα ερωτήματα της SQL. Τα ερωτήματα θα υποβληθούν στην βάση δεδομένων video club, που δημιουργήθηκε στο προηγούμενο κεφάλαιο. Συγκεκριμένα, θα μελετηθούν εντολές της SQL που αφορούν την διαχείριση δεδομένων (Data Manipulation Language). Ενδεικτικά αναφέρεται ότι θα παρουσιαστούν ερωτήματα σύνδεσης πινάκων, ομαδοποίησης, πράξεων συνόλων και η δημιουργία ερωτημάτων με γραφικό τρόπο (Query by Example) μέσα από το περιβάλλον του Query Designer.

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 σας επιτρέπει να επιλέγετε 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 ή απλά να τα εκτυπώσουμε, όπως φαίνεται στο αναδυόμενο παράθυρο της Εικόνας 3.1.

3.1.2 Ερωτήματα επιλογής γραμμών από ένα πίνακα.

Η βασική λειτουργία της αναζήτησης πληροφοριών στη βάση δεδομένων γίνεται με το ακόλουθο ερώτημα:

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

Η σύνταξη της εντολής περιλαμβάνει τρεις όρους:

- Ο όρος **select** χρησιμοποιείται για τη δήλωση των n πεδίων $A1, A2, \dots, An$ που θα περιέχονται στο αποτέλεσμα (αντιστοιχεί στην πράξη της προβολής/projection).
- Ο όρος **from** δηλώνει το σύνολο των m σχέσεων (πινάκων) $R1, R2, \dots, Rm$ στους οποίους γίνεται η αναζήτηση (αντιστοιχεί στην πράξη του Καρτεσιανού γινομένου).
- Ο όρος **where** περιέχει την *συνθήκη* που πρέπει να πληρούν οι πλειάδες 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 πλειάδες, ανεξαρτήτως συνθήκης.

Έστω ότι αναφερόμαστε στον πίνακα ΠΕΛΑΤΗΣ, ο οποίος περιέχει πληροφορίες για τους πελάτες του videoclub. 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 πλειάδων του αποτελέσματος. Στην συγκεκριμένη εντολή, γίνεται επιλογή από μία μόνο σχέση. Η εντολή του παραδείγματος δεν έχει τον όρο **where**, με αποτέλεσμα να επιλεγθούν όλοι οι πελάτες ανεξαρτήτως συνθήκης, όπως απαιτούσε η ερώτηση.

Τα αποτελέσματα που προκύπτουν από μια εντολή της SQL μπορεί να περιέχουν δύο ίδιες πλειάδες. 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 συγγάμοι), 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 πλειάδων. 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 μία συνθήκη που πρέπει να ικανοποιούν οι πλειάδες του αποτελέσματος. Έστω για παράδειγμα η ερώτηση:

“Να βρεθούν οι κωδικοί από όσες κασέτες έχουν διαθέσιμη ποσότητα από ένα έως δύο αντίτυπα, ή όσες η τιμή τους είναι μεγαλύτερη του 2”. Η αντίστοιχη εντολή της SQL είναι:

Q6

```
Select ID
From ΚΑΣΕΤΕΣ
Where (Ποσότητα between 1 and 2) or Τιμή > 2;
```

ID
2
3

Ο όρος “between 1 and 2” είναι ισοδύναμος με τον όρο “>= 1 and <= 2”. 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 την χρήση παρενθέσεων. Για παράδειγμα, για την ερώτηση:

“Να βρεθούν οι κωδικοί από τις κασέτες που είναι τύπου VHS και, επιπλέον, η διαθέσιμη ποσότητα τους είναι μεγαλύτερη του 2 ή η τιμή τους είναι μεγαλύτερη του 2”. Η αντίστοιχη εντολή SQL είναι:

Q7

```
Select ID
From ΚΑΣΕΤΕΣ
Where Τύπος = 'VHS' and (Ποσότητα > 2 or Τιμή > 2);
```

ID
1

Για αλφαριθμητικά δεδομένα, χρήσιμος είναι 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 'K%';
```

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	IDΚασέτα	Από	Έως
2	1	2006-09-10	NULL

3.1.3 Ταξινόμηση αποτελεσμάτων

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

“**Να δοθούν οι κωδικοί των κασετών ταξινομημένοι ως προς τη διαθέσιμη ποσότητά τους, κατά αύξοντα τρόπο**”:

```
Q10 Select ID, Ποσότητα
From ΚΑΣΕΤΕΣ
Order by Ποσότητα;
```

ID	Ποσότητα
2	1
3	2
1	3

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 κασετών ταξινομημένοι κατά φθίνοντα τρόπο ως προς την τιμή ενοικίασης. Στην περίπτωση ίσων τιμών ενοικίασης, η ταξινόμηση να γίνει κατά αύξουσα ταξινόμηση ως προς την ποσότητα”:

Q11 **Select** ID, Τιμή, Ποσότητα
From ΚΑΣΕΤΕΣ
Order by Τιμή **desc**, Ποσότητα **asc**;

ID	Τιμή	Ποσότητα
2	3	1
3	2	2
1	2	3

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

“Να δοθούν οι κωδικοί των κασετών που έχουν ενοικιαστεί, καθώς και οι ημερομηνίες επιστροφής τους, ταξινομημένες ως προς τις ημερομηνίες επιστροφής”:

Q12 **Select** IDΚασέτας, DAY(Έως) as Ημέρα, MONTH(Έως) as Μήνας, YEAR(Έως) as Έτος
From ΕΝΟΙΚΙΑΣΗ
Order by Έως;

IDΚασέτας	Ημερομηνία	Μήνας	Έτος
1	NULL	NULL	NULL
1	10	9	2006
2	20	11	2006

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

“Να δοθούν οι κωδικοί των 2 κασετών με τη μεγαλύτερη διαθέσιμη ποσότητα”.

Q13 **Select top 2** ID
From ΚΑΣΕΤΕΣ
Order by Ποσότητα;

ID
2
3

3.2. Ερωτήματα επιλογής γραμμών από πολλούς πίνακες

3.2.1 Εσωτερική και εξωτερική σύνδεση πινάκων

Με τον όρο **from** δηλώνεται το σύνολο των πινάκων στους οποίους γίνεται η αναζήτηση. Στα προηγούμενα παραδείγματα, οι αναζητήσεις δεδομένων αφορούσαν ένα μόνο πίνακα. Η αναζήτηση δεδομένων σε περισσότερους πίνακες γίνεται με την βοήθεια της πράξης της *σύνδεσης* (join). Για παράδειγμα. έστω η ερώτηση:

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

Q1

```
Select Όνομα, Ρόλος
From TN_ΣΝ, ΣΥΝΤΕΛΕΣΤΗΣ
Where TN_ΣΝ.IDΣυντελεστή = ΣΥΝΤΕΛΕΣΤΗΣ.ID;
```

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

Στο προηγούμενο παράδειγμα σύνδεσης, ο όρος **where** περιείχε μόνο την συνθήκη που ήταν απαραίτητη για τη δημιουργία της σύνδεσης. Εκτός αυτής της συνθήκης είναι δυνατό να περιέχονται και άλλες που πρέπει να πληρούν οι 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:

Q2

```
Select IDΤαινίας
From TN_ΣΝ, ΣΥΝΤΕΛΕΣΤΗΣ
Where TN_ΣΝ.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 την χρήση του όρου inner join είναι:

```

Q3      Select IDΤαινίας
From TN_ΣΝ inner join ΣΥΝΤΕΛΕΣΤΗΣ on
TN_ΣΝ.IDΣυντελεστή = ΣΥΝΤΕΛΕΣΤΗΣ.ID
Where ΣΥΝΤΕΛΕΣΤΗΣ.Όνομα = 'Alfred Hitchcock';
    
```

IDΤαινίας
1
2

ΠΡΟΣΟΧΗ! Στο ερώτημα ποιον από τους δύο τρόπους να χρησιμοποιούμε θα λέγαμε ότι στην περίπτωση όπου η αναζήτηση αφορά μερικούς πίνακες τότε η σύνδεση με την βοήθεια του όρου where είναι πιο συμφέρουσα. Όμως στην περίπτωση που η αναζήτηση μας αφορά πολλούς πίνακες τότε η χρήση του inner join είναι πιο βολική επειδή με αυτή κάθε φορά βλέπουμε με ποια πεδία συνδέονται οι υπό σύνδεση πίνακες. Συνεπώς, γίνεται πιο εύκολα η ανάγνωση ενός πολύπλοκου ερωτήματος σύνδεσης πινάκων. Επιπροσθέτως, 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 κασετών που έχει ενοικιάσει**”. Η αντίστοιχη εντολή SQL είναι:

```

Q4      Select ΠΕΛΑΤΗΣ.Όνομα, ΚΑΣΕΤΕΣ.ID, ΚΑΣΕΤΕΣ.Τιμή
From ΠΕΛΑΤΗΣ inner join ΕΝΟΙΚΙΑΣΗ on
ΠΕΛΑΤΗΣ.ID = ΕΝΟΙΚΙΑΣΗ.IDΠελάτη inner join ΚΑΣΕΤΕΣ on
ΕΝΟΙΚΙΑΣΗ.IDΚασέτας = ΚΑΣΕΤΕΣ.ID;
    
```

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

Στο αποτέλεσμα του **inner join** συμμετέχουν μόνο οι εγγραφές των σχέσεων, 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 μία ειδικότερη πράξη σύνδεσης (join) σχέσεων, που ονομάζεται **left outer join**. Με την πράξη του *left outer join*, το αποτέλεσμα περιέχει όλες αυτές τις πλειάδες, και επιπλέον όλες τις πλειάδες της αριστερής σχέσης για τις οποίες δεν έγινε ταύτιση με καμία στη δεξιά σχέση. Σε αυτού του είδους τις πλειάδες, οι στήλες που αντιστοιχούν στη δεύτερη σχέση έχουν τιμή ίση με null. Συνεπώς, *κάθε* πλειάδα της αριστερής σχέσης συμμετέχει στο αποτέλεσμα. Για παράδειγμα, έστω το ερώτημα:

“Να βρεθούν για κάθε πελάτη το όνομα του, ο κωδικός και η τιμή των κασετών που έχει ενοικιάσει. Να εμφανίζονται και οι πελάτες που δεν έχουν ενοικιάσει κάποια κασέτα”. Η αντίστοιχη εντολή SQL είναι:

Q5

```
Select ΠΕΛΑΤΗΣ.Όνομα, ΚΑΣΕΤΕΣ.ID, ΚΑΣΕΤΕΣ.Τιμή
From ΠΕΛΑΤΗΣ left outer join ΕΝΟΙΚΙΑΣΗ on
ΠΕΛΑΤΗΣ.ID = ΕΝΟΙΚΙΑΣΗ.IDΠελάτη left outer join ΚΑΣΕΤΕΣ on
ΕΝΟΙΚΙΑΣΗ.IDΚασέτας = ΚΑΣΕΤΕΣ.ID;
```

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

Όπως φαίνεται στα αποτελέσματα του παραδείγματος 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 πλειάδες, και επιπλέον όλες τις πλειάδες της δεξιάς σχέσης για τις οποίες δεν έγινε ταύτιση με καμία στην αριστερή σχέση. Σε αυτού του είδους τις πλειάδες, οι στήλες που αντιστοιχούν στη δεύτερη σχέση έχουν τιμή ίση με null. Συνεπώς, *κάθε* πλειάδα της δεξιάς σχέσης συμμετέχει στο αποτέλεσμα. Για παράδειγμα, έστω το ερώτημα:

“Να βρεθούν για κάθε πελάτη το όνομα του, ο κωδικός και η τιμή των κασετών που έχει ενοικιάσει. Να εμφανίζονται οι κωδικοί και οι τιμές των κασετών που δεν έχουν ενοικιαστεί από κάποιον πελάτη”. Η αντίστοιχη εντολή SQL είναι:

Q6

```
Select ΠΕΛΑΤΗΣ.Όνομα, ΚΑΣΕΤΕΣ.ID, ΚΑΣΕΤΕΣ.Τιμή
From ΠΕΛΑΤΗΣ right outer join ΕΝΟΙΚΙΑΣΗ on ΠΕΛΑΤΗΣ.ID = ΕΝΟΙΚΙΑΣΗ.IDΠελάτη
right outer join ΚΑΣΕΤΕΣ on ΕΝΟΙΚΙΑΣΗ.IDΚασέτας = ΚΑΣΕΤΕΣ.ID;
```

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

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

“**Να βρεθούν για κάθε πελάτη το όνομα του, ο κωδικός και η τιμή των κασετών που έχει ενοικιάσει. Να εμφανίζονται και οι πελάτες που δεν έχουν ενοικιάσει κάποια κασέτα, αλλά και οι κωδικοί και τιμές των κασετών που δεν έχουν ενοικιαστεί από κάποιον πελάτη**”. Η αντίστοιχη εντολή SQL είναι:

Q7

```
Select ΠΕΛΑΤΗΣ.Όνομα, ΚΑΣΕΤΕΣ.ID, ΚΑΣΕΤΕΣ.Τιμή
From ΠΕΛΑΤΗΣ full outer join ΕΝΟΙΚΙΑΣΗ on ΠΕΛΑΤΗΣ.ID = ΕΝΟΙΚΙΑΣΗ.IDΠελάτη
full outer join ΚΑΣΕΤΕΣ on ΕΝΟΙΚΙΑΣΗ.IDΚασέτας = ΚΑΣΕΤΕΣ.ID;
```

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

3.2.2 Μετονομασία και αυτό-σύνδεση

Σε ορισμένα ερωτήματα προκύπτει η ανάγκη μετονομασίας πεδίων ή πινάκων. Αυτό γίνεται με την 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:

Στο πεδίο Τηλέφωνο θέλουμε να εμφανίζεται το πρόθεμα 2310 και στο αποτέλεσμα η στήλη θα έχει όνομα Τηλέφωνο.

Q8

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

Όνομα	Τηλέφωνο
Perkins	2310246801
Καντακουζηνός	2310246801
Παλαιολόγος	2310987654

Σημειώστε 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:

Q9

```
Select B.Όνομα
From ΠΕΛΑΤΗΣ as A, ΠΕΛΑΤΗΣ as B
Where A.Τηλέφωνο = B.Τηλέφωνο and A.Όνομα = 'Perkins' and B.Όνομα <> 'Perkins';
```

Όνομα

Καντακουζηνός

Χωρίς μετονομασία, δεν μπορούσε να γραφεί στη συνθήκη ότι, π.χ., ΠΕΛΑΤΗΣ.Τηλέφωνο = ΠΕΛΑΤΗΣ.Τηλέφωνο, γιατί δεν γίνεται 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 η κασέτα τύπου VHS είναι σε μεγαλύτερη ποσότητα από την αντίστοιχη κασέτα τύπου DVD**”. Η αντίστοιχη εντολή SQL είναι:

Q10

```
Select A.IDΤαινίας
From ΚΑΣΕΤΕΣ as A, ΚΑΣΕΤΕΣ as B
Where A.IDΤαινίας = B.IDΤαινίας and A.Τύπος = 'VHS' and B.Τύπος = 'DVD' and
A.Ποσότητα > B.Ποσότητα;
```

IDΤαινίας

1

3.3. Ερωτήματα ομαδοποίησης

3.3.1 Συναρτήσεις ομαδοποίησης

Η SQL περιέχει τις εξής κυριότερες συναρτήσεις ομαδοποίησης (aggregate functions) όπως φαίνονται στον Πίνακα 3.14:

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

Πίνακας 3.14

Οι συναρτήσεις *αθροίσματος* και *μέσων όρων* 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 μια κασέτας**”. Η εντολή της SQL είναι:

```
Q1  Select max(Τιμή) as 'Μέγιστη Τιμή Κασέτας'  
     From ΚΑΣΕΤΕΣ;
```

Μέγιστη Τιμή Κασέτας

3

Ένα δεύτερο παράδειγμα ερωτήματος είναι το παρακάτω :

“**Να βρεθεί ο συνολικός αριθμός των κασετών**”. Η αντίστοιχη εντολή SQL είναι:

```
Q2  Select count(*)  
     From ΚΑΣΕΤΕΣ;
```

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 μιας κασέτας**”. Η εντολή της SQL είναι:

```
Q3  Select max(Τιμή) - min(Τιμή)  
     From ΚΑΣΕΤΕΣ;
```

1

Η ύπαρξη ορισμάτων εντός μιας συνάρτησης ομαδοποίησης δηλώνει ότι αυτή εφαρμόζεται στο αποτέλεσμα προβολής ως προς τα ορίσματα αυτά. 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”. Η αντίστοιχη εντολή SQL είναι:

```
Q4 Select count(distinct IDΠελάτη)
    From ΕΝΟΙΚΙΑΣΗ;
```

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

3.3.2 Ο όρος `Group by`

Σε αρκετές περιπτώσεις προκύπτει η ανάγκη διαμερισμού των πλειάδων 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 μέσος όρος τιμής ενοικίασης ανά τύπο κασέτας (VHS ή DVD)”. Η εντολή της SQL είναι:

```
Q5 Select Τύπος, avg(Τιμή) as 'Μέση Τιμή'
    From ΚΑΣΕΤΕΣ
    Group by Τύπος;
```

Τύπος	Μέση Τιμή
DVD	3
VHS	2

Μπορείτε να τοποθετηθείτε περισσότερες από μία στήλες στον όρο `group by`. Η τοποθέτηση περισσότερων από μια 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 κασέτα (κωδικός)”. Η εντολή SQL είναι:

```
Q6 Select IDΠελάτη, IDΚασέτας, count(IDΚασέτας) as 'Αριθμός Ενοικιάσεων'
    From ΕΝΟΙΚΙΑΣΗ
    Group by IDΠελάτη, IDΚασέτας;
```

IDΠελάτη	IDΚασέτας	Αριθμός Ενοικιάσεων
1	1	1
1	2	1
2	1	1

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

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

Q7

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

Τύπος	Μέση Τιμή
VHS	2
DVD	3

Αν θέλουμε να έχουμε ομαδοποίηση με τον τελεστή count και να εμφανίζονται (με count ίσο με 0) οι εγγραφές που δεν συμμετέχουν, τότε χρησιμοποιούμε μία 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:

Q8

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

Η χρήση της προβολής ως προς IDΚασέτας 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 τμήμα εφόσον ομαδοποιηθεί και μετά (προσοχή! όχι η κάθε γραμμή όπως στον όρο where). Σε αυτήν την περίπτωση χρησιμοποιείται ο όρος **having** μετά από τον group by, ώστε μία συνθήκη αναζήτησης να εξεταστεί αφότου ομαδοποιηθούν τα αποτελέσματα. Έστω για παράδειγμα η ερώτηση:

“Να βρεθεί ο τύπος κασέτας για τον οποίο ο μέσος όρος τιμής ενοικίασης είναι μεγαλύτερος από 2”. Η εντολή SQL είναι:

Q9

```
Select Τύπος, avg(Τιμή) as 'Μέσος Όρος Τιμής Ενοικίασης'  
From ΚΑΣΕΤΕΣ  
Group by Τύπος  
Having avg(Τιμή) > 2;
```

Τύπος	Μέσος Όρος Τιμής Ενοικίασης
DVD	3

Τέλος, ένα παράδειγμα, στο οποίο εμφανίζεται τόσο ο όρος where όσο και ο όρος Having εμφανίζεται στο παρακάτω ερώτημα:

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

Q10

```
Select IDΣυντελεστή
From TN_ΣΝ
where Ρόλος = 'Σκηνοθέτης'
Group by IDΣυντελεστή
Having count(distinct IDΤαινίας) > 1;
```

IDΣυντελεστή

1

3.4. Ερωτήματα για πράξεις συνόλων με φωλιασμένες εντολές SQL

3.4.1 Η πράξη της Ένωσης (Union)

Ο όρος **Union** πραγματοποιεί την πράξη της ένωσης σχέσεων. Για την εφαρμογή αυτού του όρου μεταξύ δύο σχέσεων πρέπει αυτές 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 αυτών στις οποίες συμμετέχει η Grace Kelly**”. Η αντίστοιχη εντολή SQL είναι:

Q1

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

union

```
(Select Τίτλος  
From TAINIA,ΣΥΝΤΕΛΕΣΤΗΣ,TN_ΣΝ  
Where TN_ΣΝ.IDΤαινίας = TAINIA.ID and  
TN_ΣΝ.IDΣυντελεστή= ΣΥΝΤΕΛΕΣΤΗΣ.ID and  
ΣΥΝΤΕΛΕΣΤΗΣ.Όνομα = 'Grace Kelly' );
```

Τίτλος

Ben-Hur

Rear Window

3.4.2 Έλεγχος περιεκτικότητας σε σύνολο

Οι εντολές της SQL έχουν την ιδιότητα της *κλειστότητας*. Με αυτό τον τρόπο είναι δυνατό να εμφωλιαστούν εντολές ώστε το αποτέλεσμα της μιας να είναι είσοδος μίας άλλης. 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 στο μαθηματικό σύμβολο. Έστω για παράδειγμα η ερώτηση:

“**Να βρεθούν οι πελάτες που έχουν κάνει ενοικίαση τουλάχιστον μίας κασέτας**”. 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**.

Q2

```
Select Όνομα  
From ΠΕΛΑΤΗΣ  
Where ID in (Select IDΠελάτη From ΕΝΟΙΚΙΑΣΗ);
```

Όνομα

Perkins

Καντακουζηνός

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

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

```

Q3  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 του. Για παράδειγμα, έστω η παρακάτω 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 :

```

Q4  Select ΚΑΣΕΤΕΣ.IDΤαινίας
      From ΚΑΣΕΤΕΣ inner join ΕΝΟΙΚΙΑΣΗ on
      ΕΝΟΙΚΙΑΣΗ.IDΚασέτας = ΚΑΣΕΤΕΣ.ID Where ΚΑΣΕΤΕΣ.IDΤαινίας IN
      (Select IDΤαινίας From TN_ΣΝ inner join ΣΥΝΤΕΛΕΣΤΗΣ on
      TN_ΣΝ.IDΣυντελεστή = ΣΥΝΤΕΛΕΣΤΗΣ.ID
      where ΣΥΝΤΕΛΕΣΤΗΣ.Όνομα = 'Alfred Hitchcock')
      Group by ΚΑΣΕΤΕΣ.IDΤαινίας
      Having count(*) > 2;

```

IDΤαινίας

 1

3.4.3 Σύγκριση μεταξύ συνόλων

Σε κάποιες περιπτώσεις προκύπτουν ερωτήματα όπου ένα πεδίο πρέπει να συγκριθεί με την τιμή του ίδιου πεδίου σε τουλάχιστον μία άλλη γραμμή μιας σχέσης. Σε αυτή 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**. Για παράδειγμα, έστω το παρακάτω ερώτημα:

“**Να βρεθούν οι πελάτες (Όνομα) για τους οποίους υπάρχει κάποια ενοικιασμένη κασέτα για την οποία δεν έχει οριστεί ημερομηνία επιστροφής**”. Η εντολή SQL είναι :

```

Q5  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. Για παράδειγμα, έστω το παρακάτω ερώτημα:

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

Q6

```
Select IDΠελάτη
from ENOΙΚΙΑΣΗ
group by IDΠελάτη
having count(*) >= all
( Select count(*)
from ENOΙΚΙΑΣΗ
group by IDΠελάτη );
```

```
IDΠελάτη
-----
1
```

3.4.4 Σύγκριση μεταξύ συνόλων

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

“Να βρεθούν τα ονόματα των πελατών που έχουν κάνει μία τουλάχιστον ενοικίαση”. Η εντολή SQL με τον όρο **exists** είναι: (Προσοχή: Εναλλακτικά θα μπορούσε να γίνει και με **inner join**.)

Q7

```
Select Όνομα
from ΠΕΛΑΤΗΣ
where exists
(select IDΠελάτη
from ENOΙΚΙΑΣΗ
where IDΠελάτη = ΠΕΛΑΤΗΣ.ID);
```

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


Ο όρος **not exists** μπορεί να χρησιμοποιηθεί για να ελεγχθεί αν μία σχέση Y περιέχει μία άλλη σχέση X , δηλαδή είναι υπερσύνολό της. Βάσει της συνολοθεωρίας: $X \subseteq Y \Leftrightarrow X - Y = \emptyset$. Σε αυτή την περίπτωση, δεν υπάρχει πλειάδα που να ανήκει στην X που να μην ανήκει στην Y . Για παράδειγμα, έστω το 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 :

Q8

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

IDΠελάτη

1

3.5. Ερωτήματα SQL για όψεις

Μία όψη (view) είναι ένα αφηρημένο υποσύνολο (ερώτημα) που αντιστοιχεί σε ένα τμήμα ενός πίνακα της βάσης δεδομένων ή 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 :

Q9

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

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

“Διαγράψτε την όψη MyView”. Η αντίστοιχη εντολή SQL είναι:

Q10

```
drop view MyView;
```

Μία από τις χρήσεις των όψεων, είναι ότι 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, απλοποιήστε 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 ανοικιασθεί περισσότερες από δύο φορές”. Η αντίστοιχη εντολή SQL είναι:

Q11

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

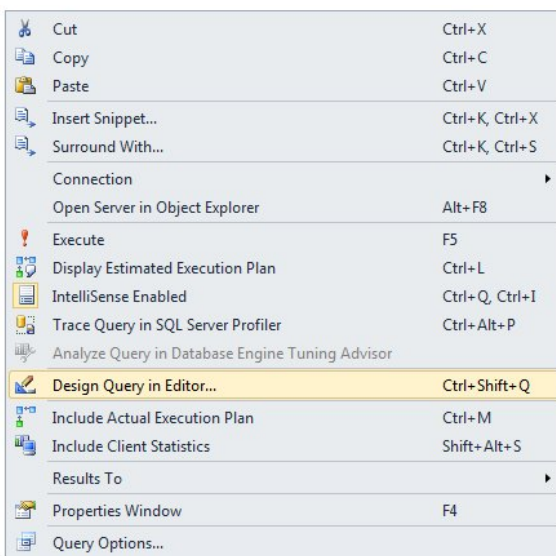
IDΤαινίας

1

3.6. Query Designer και Query by Example

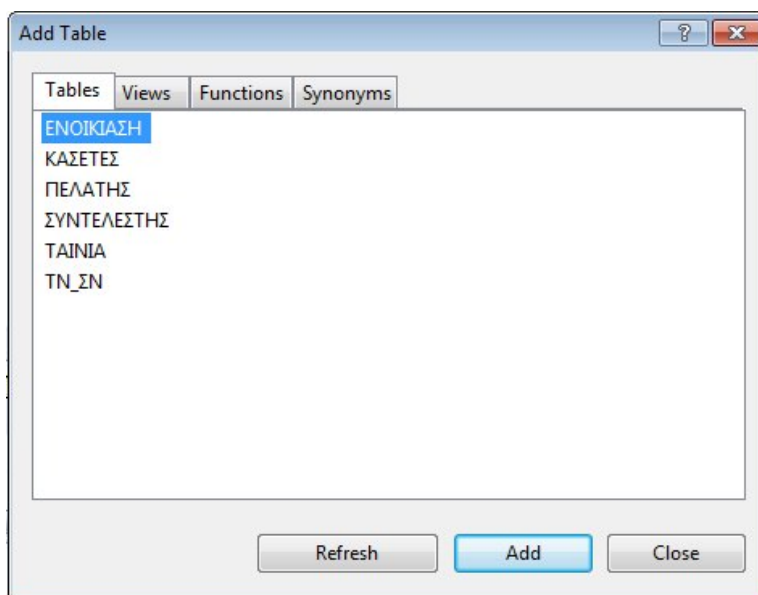
Η σύνταξη όλων των ερωτημάτων μέχρι τώρα έγινε στον Query Editor. Αυτός ο τρόπος σύνταξης SQL ερωτημάτων θεωρείται όμως αργός και δύσκολος, διότι πρέπει να πληκτρολογήσεις όλο το ερώτημα και να θυμάσαι ταυτόχρονα και τα ονόματα των πεδίων/πινάκων που θα συμμετάσχουν σ' αυτό.

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 videoclub, 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, όπου επιλέγουμε Design Query in Editor.



Εικόνα 3.2

Στο επόμενο βήμα, θα εμφανιστεί 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:

Q12

«Να εμφανίσετε τα στοιχεία των ταινιών που συμμετείχε ως συντελεστής ο Alfred Hitchcock».

Για να απαντηθεί το παραπάνω ερώτημα πρέπει να αντλήσουμε δεδομένα από 3 συσχετιζόμενους πίνακες. Συνεπώς, από τους διαθέσιμους πίνακες της Εικόνας 3.3 επιλέγουμε τους 3 πίνακες (ΤΑΙΝΙΑ, ΤΝ_ΤΣ, ΣΥΝΤΕΛΕΣΤΗΣ) που θα συμμετάσχουν στο ερώτημα μας. Όπως φαίνεται στην Εικόνα 3.4, οι συσχετίσεις τους εμφανίζονται αυτόματα και 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, Or, κτλ.). Σε αυτό το grid θα δημιουργήσουμε το ερώτημα μας απλά τσεκάροντας τα αντίστοιχα πεδία από τους συσχετιζόμενους πίνακες. Επιλέξτε λοιπόν τα πεδία (ID, Τίτλος, Έτος) από τον πίνακα ΤΑΙΝΙΑ και αυτά θα εμφανιστούν αυτόματα στο grid στις στήλες column και output. Στη συνέχεια, επιλέξτε το πεδίο Όνομα από τον πίνακα ΣΥΝΤΕΛΕΣΤΗΣ και στην στήλη Filter πληκτρολογήστε το όνομα του συντελεστή όπως φαίνεται στην Εικόνα 3.4. Το ερώτημα μας έχει πλέον δημιουργηθεί και εμφανίζεται σε εντολές SQL στο κάτω μέρος της Εικόνας 3.4.

The screenshot shows the Query Designer window with three tables: TAINIA, TN_ΣΝ, and ΣΥΝΤΕΛΕΣΤΗΣ. The TAINIA table has columns ID, Τίτλος, and ΤΑΙΝΙΑ.Έτος selected. The TN_ΣΝ table has columns IDΤαινίας and IDΣυντελεστή selected. The ΣΥΝΤΕΛΕΣΤΗΣ table has the column Όνομα selected and filtered with the value 'Alfred Hitchcock'. The grid below shows the resulting query structure, and the SQL query is displayed at the bottom.

Column	Alias	Table	Outp...	Sort Type	Sort Order	Filter
ID		TAINIA	<input checked="" type="checkbox"/>			
Τίτλος		TAINIA	<input checked="" type="checkbox"/>			
[ΤΑΙΝΙΑ.Έτος]		TAINIA	<input checked="" type="checkbox"/>			
Όνομα		ΣΥΝΤΕΛΕΣ...	<input type="checkbox"/>			= 'Alfred Hitchcock'

```
SELECT TAINIA.ID, TAINIA.Τίτλος, TAINIA.[ΤΑΙΝΙΑ.Έτος]
FROM TN_ΣΝ INNER JOIN
      ΣΥΝΤΕΛΕΣΤΗΣ ON TN_ΣΝ.IDΣυντελεστή = ΣΥΝΤΕΛΕΣΤΗΣ.ID INNER JOIN
      ΤΑΙΝΙΑ ON TN_ΣΝ.IDΤαινίας = ΤΑΙΝΙΑ.ID
WHERE (ΣΥΝΤΕΛΕΣΤΗΣ.Όνομα = 'Alfred Hitchcock')
```

Εικόνα 3.4

Παρακάτω περιγράφουμε τις βασικές στήλες του grid:

- Η στήλη Alias ορίζει προαιρετικά το τελικό όνομα του πεδίου όπως θα εμφανιστεί στα αποτελέσματα, εφόσον θέλουμε να αλλάξει.
- Η στήλη Output ορίζει αν το πεδίο θα εμφανιστεί στα τελικά αποτελέσματα ή 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 μας επιτρέπουν να ορίσουμε τα φίλτρα ενός ή περισσότερων πεδίων. Σε κάθε επόμενη στήλη μπορούμε να ορίσουμε μια νέα τιμή ενός πεδίου μέσω του τελεστή Or.

Για να εκτελέσουμε το ερώτημα μας πατάμε OK και στην συνέχεια επιλέγουμε F5. Παρακάτω εμφανίζονται τα αποτελέσματα, σύμφωνα με τα οποία ο Alfred Hitchcock έχει συμμετάσχει σε δύο ταινίες.

ID	Τίτλος	Έτος
1	Rear Window	1954
2	Psycho	1960

(2 row(s) affected)

Σημειώστε 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 την επιλεγμένη περιοχή επιλέγουμε ξανά Design Query in Editor... και ο Editor του Management Studio μας επανασχεδιάζει οπτικά το ερώτημά μας για να συνεχίσουμε.

Συμπερασματικά, 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.

3.7. Ασκήσεις με ερωτήματα SQL

3.7.1 Ασκήσεις με ερωτήματα επιλογής γραμμών από ένα πίνακα.

1. Για κάθε κασέτα, να προβληθεί ο κωδικός και η τιμή. Η τιμή να εμφανίζεται χωρίς το ΦΠΑ, δηλαδή μειωμένη κατά $0.19 * \text{Τιμή}$.
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 κασετών που έχουν ενοικιασθεί με ορισμένη ημερομηνία επιστροφής

3.7.2 Ασκήσεις με ερωτήματα επιλογής γραμμών από πολλούς πίνακες.

6. Να βρεθούν οι κωδικοί των 2 κασετών με τη μεγαλύτερη διαθέσιμη ποσότητα
7. Να βρεθούν τα ονόματα των πελατών που έχουν ενοικιάσει τουλάχιστον μία κασέτα
8. Να βρεθούν τα ονόματα των πελατών που δεν έχουν ενοικιάσει ούτε μία κασέτα (να μην χρησιμοποιηθεί ο τελεστής NOT IN).
9. Να βρεθούν οι κωδικοί των συντελεστών που έχουν συμμετάσχει σε τουλάχιστον 2 ταινίες (να μην χρησιμοποιηθεί ο τελεστής COUNT).
10. Να βρεθούν οι τίτλοι των ταινιών για τις οποίες είτε δεν υπάρχει κασέτα, είτε υπάρχει κασέτα και δεν έχει ενοικιαστεί ποτέ (να μην χρησιμοποιηθεί ο τελεστής NOT IN).
11. Να βρεθούν οι πελάτες με επίθετο ίδιο με αυτό κάποιου συντελεστή ταινίας

3.7.3 Ασκήσεις με Ερωτήματα ομαδοποίησης/συνάθροισης δεδομένων.

12. Να βρεθεί ο αριθμός των ταινιών που έχει συμμετάσχει ο Alfred Hitchcock. Σημείωση: Οι ταινίες στις οποίες έχει συμμετάσχει με περισσότερους από ένα ρόλους, να προσμετρούνται μία μόνο φορά.
13. Για κάθε ταινία (τίτλος), να βρεθεί ο αριθμός διαφορετικών τύπων κασέτας (VHS ή DVD) που περιέχει την ταινία. Στο αποτέλεσμα να εμφανίζονται και οι ταινίες για τις οποίες δεν υπάρχει κασέτα.
14. Να βρεθούν οι κωδικοί των κασετών που είναι τύπου VHS και έχουν ενοικιασθεί περισσότερες από μία φορές.

3.7.4 Ασκήσεις με ερωτήματα με φωλιασμένες εντολές SQL και όψεις.

15. Να βρεθούν οι τίτλοι των ταινιών που δεν έχουν ενοικιασθεί ποτέ. (Να μην χρησιμοποιηθεί outer join)
16. Να βρεθεί το όνομα του Συντελεστή που έχει συμμετάσχει στις περισσότερες ταινίες.
17. Να βρεθεί ο κωδικός της κασέτας με τη δεύτερη μεγαλύτερη ποσότητα. Στο αποτέλεσμα να εμφανίζεται μόνο ο κωδικός αυτός. (Υπόδειξη: η κασέτα με τη δεύτερη μεγαλύτερη ποσότητα είναι αυτή που συμμετέχει στο top2 της κατάταξης ως προς την ποσότητα, αλλά δεν συμμετέχει στο top1.)

Κεφάλαιο 4. Προχωρημένες λειτουργίες στον SQL Server

Σύνοψη

Στο παρόν κεφάλαιο θα παρουσιάσουμε προχωρημένες λειτουργίες που γίνονται στο περιβάλλον του SQL Server. Συγκεκριμένα, θα μελετηθούν εντολές της SQL που αφορούν τον ορισμό δεδομένων (Data Definition Language). Άλλα προχωρημένα θέματα που θα παρουσιαστούν 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 στην αναζήτηση πληροφορίας σε μία βάση δεδομένων. Η SQL περιέχει εντολές και για την εισαγωγή, διαγραφή και μεταβολή της πληροφορίας που βρίσκεται στη βάση δεδομένων.

4.1.1 Εισαγωγή πολλών γραμμών σε πίνακα

Στην SQL, είναι δυνατή η εισαγωγή πολλών γραμμών σε ένα πίνακα με την βοήθεια ενός ερωτήματος επιλογής από ένα άλλο πίνακα. Για παράδειγμα έστω τα παρακάτω 2 ερωτήματα:

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

Q1

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

“**Στον πίνακα ΠΑΛΙΑ_ΤΑΙΝΙΑ να προστεθούν οι γραμμές του πίνακα ΤΑΙΝΙΑ, οι οποίες αντιστοιχούν σε ταινίες με χρονιά μικρότερη του 1960**”. Η εντολή SQL είναι:

Q2

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

```
-----  
(3 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:

“Ενοικιάστηκε η κασέτα με κωδικό 1. Να μειωθεί κατά 1 η ένδειξη της διαθέσιμης ποσότητας της”. Η εντολή SQL είναι:

Q3

```
Update ΚΑΣΕΤΕΣ  
Set Ποσότητα = Ποσότητα - 1  
Where ID = 1;
```

(1 row(s) affected)

Ο όρος where είναι προαιρετικός. Αν δεν δίνεται συνθήκη, τότε ενημερώνονται τα πεδία όλων των γραμμών. Ο όρος set περιέχει τα πεδία που θα ενημερωθούν, τα οποία μπορεί να είναι και περισσότερα από ένα. Για παράδειγμα, έστω το ερώτημα:

“Έγινε ανακαίνιση στο video club και πλέον όλες οι κασέτες είναι τύπου DVD και σε διαθέσιμη ποσότητα ίση με 5. Να ενημερωθεί ο πίνακας ΚΑΣΕΤΕΣ”. Η εντολή SQL είναι:

Q4

```
Update ΚΑΣΕΤΕΣ  
Set Τύπος = 'DVD', Ποσότητα = 5;
```

(3 row(s) affected)

4.1.3 Διαγραφή των γραμμών ενός πίνακα

Στην SQL επιτρέπεται η διαγραφή ολόκληρων πλειάδων 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:

Q5

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

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

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

Q6

```
Delete from ΠΕΛΑΤΗΣ;
```


4.1.4 Μεταβολή της δομής ενός πίνακα

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

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

Q7

```
Alter table ΤΑΙΝΙΑ  
add Γλώσσα char(30);
```

ΠΡΟΣΟΧΗ! οι τιμές του πεδίου Γλώσσα είναι αρχικά null για όλες τις γραμμές.

Για την διαγραφή πεδίων σε έναν πίνακα, υπάρχει η εντολή της 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”:

Q8

```
Alter table ΤΑΙΝΙΑ drop column Γλώσσα;
```

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

“Στον πίνακα TN_ΣN, ο τύπος του πεδίου Ρόλος να γίνει varchar(100)”:

Q9

```
Alter table TN_ΣN  
Alter column Ρόλος 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 ΚΑΣΕΤΕΣ σε DVD”. Η εντολή SQL είναι:

Q10

```
Exec sp_rename 'ΚΑΣΕΤΕΣ', '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 πεδίο Χρονιά του πίνακα ΤΑΙΝΙΑ σε Έτος”. Η εντολή SQL είναι:

Q11

```
Exec sp_rename 'ΤΑΙΝΙΑ.Χρονιά', 'Έτος', '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:

Q12

Drop table ΠΑΛΙΑ_ΤΑΙΝΙΑ;

Είναι σημαντικό να σημειωθεί ότι η παραπάνω εντολή: (α) διαγράφει *όλες* τις γραμμές και (β) καταργεί τον ίδιο τον πίνακα. Δηλαδή, δεν είναι δυνατή η αναφορά, για παράδειγμα, 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 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:

Q13

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, οι οποίες 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:

Q1

```
use videoclub
go
Create procedure InsertRandomPelates @CustomersTotalNumber as int
as
begin
    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)

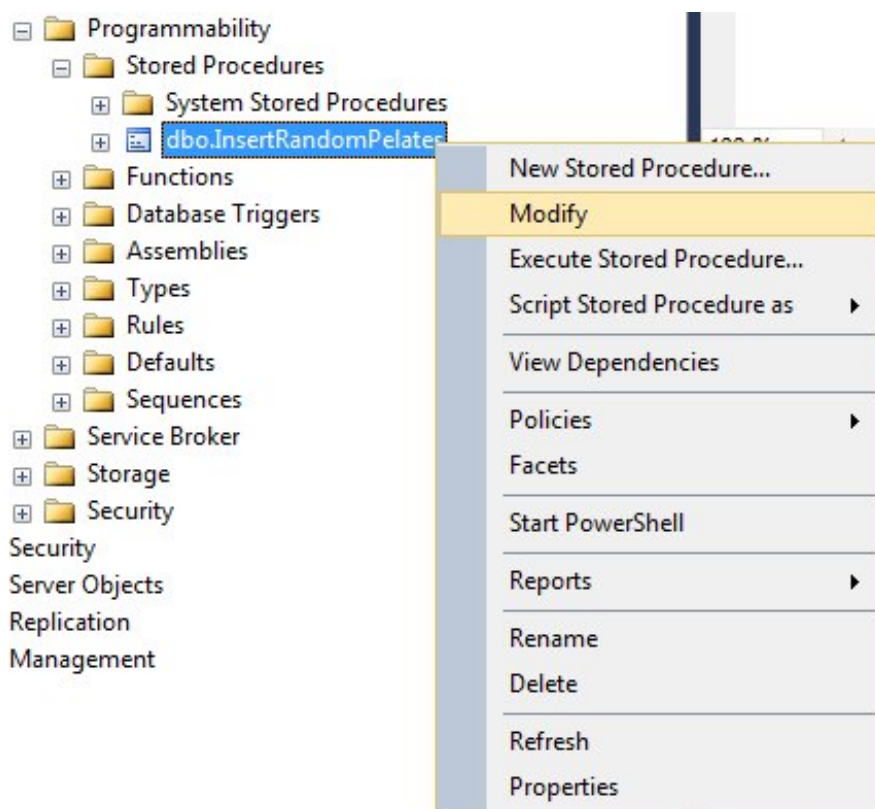
        Set @counter= @counter +1
    end
end
```

Ακολουθώς 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 και δηλώνεται μία 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 μία μεταβλητή με μήκος 30 χαρακτήρες, η οποία θα κρατά το όνομα πελάτη, που θα δημιουργείται κάθε φορά με σειριακό/άξοντα τρόπο.
- *Declare @counter as int:* Δηλώνουμε μία 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.

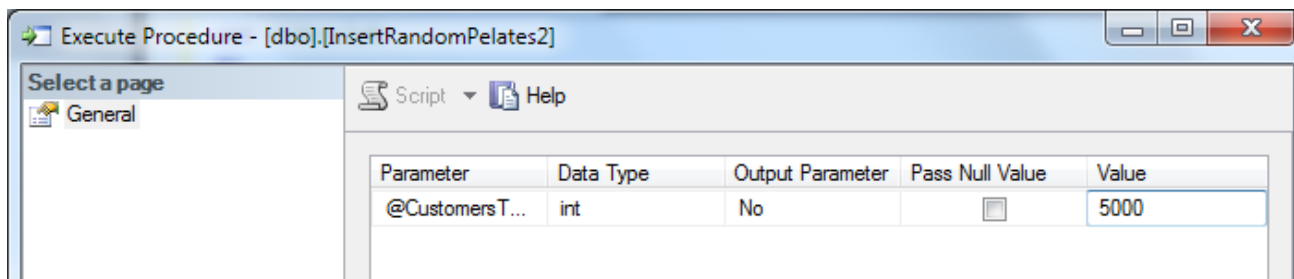
- *while @counter <= @CustomersTotalNumber*: Ξεκινά η επαναληπτική διαδικασία εισαγωγής γραμμών και διαρκεί όσο ο μετρητής μας είναι μικρότερος ή ίσος από τον συνολικό αριθμό εισαγωγής γραμμών που έχουμε δηλώσει με την μεταβλητή @CustomersTotalNumber.
- *Set @CustomersName=cast(@counter as varchar) + '-Name'*: Δημιουργώ το όνομα του πελάτη που αποτελείται από 2 κομμάτια: Τον τρέχοντα αύξων 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 τον μεταβλητή @counter τύπου int σε μεταβλητή τύπου varchar (χαρακτήρας).
- *INSERT INTO ΠΕΛΑΤΗΣ (ID, Όνομα) VALUES (@counter, @CustomersName)*: Η εντολή που εισάγει μία νέα εγγραφή στον πίνακα ΠΕΛΑΤΗΣ.
- *Set @m = @m + 1*: Η εντολή που αυξάνει τον μετρητή counter κατά 1.

Μετά την επιτυχημένη εκτέλεση της αποθηκευμένης διαδικασίας, την βρίσκουμε στο φάκελο Programmability – Stored Procedures, όπως φαίνεται στην Εικόνα 4.1. Αν θέλουμε να αλλάξουμε τον κώδικα της διαδικασίας επιλέγουμε δεξί κλικ πάνω της και 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 εισαγάγουμε 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 ξεκινά η εκτέλεσή της.



Εικόνα 4.2

Ανάλογα με τον όγκο της εργασίας της, η αποθηκευμένη διαδικασία εκτελείται για 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 πάμε και 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:

```

Q2 use videoclub
go
Create Procedure myphone1
as
Select Τηλέφωνα
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:

```

Q3 use videoclub
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 μπορείτε να δοκιμάσετε να δημιουργήσετε μια stored procedure που θα εμφανίζει όλους τους ηθοποιούς.

4.2.2 Εναύσματα/Triggers

Μία διαδικασία την οποία δηλώνει ο διαχειριστής της βάσης δεδομένων και ενεργοποιείται αυτόματα από τον SQL Server κάθε φορά που συμβαίνουν μεταβολές εισαγωγής, διαγραφής ή ενημέρωσης στα δεδομένα ονομάζεται έναυσμα/trigger. Για παράδειγμα, έστω τα παρακάτω ερωτήματα:

“Δημιουργήστε ένα νέο πεδίο στον πίνακα ΚΑΣΕΤΕΣ με όνομα ‘loaned’ και τύπο δεδομένων char(1). Το πεδίο θα έχει default αρχική τιμή (‘n’), που σημαίνει ότι η κασέτα δεν είναι ενοικιασμένη.”

Q4

```
use videoclub
go
Alter Table ΚΑΣΕΤΕΣ
add loaned char(1) default ‘n’;
```

Προσοχή! Σημειώνεται ότι οι ήδη υπάρχουσες εγγραφές του πίνακα ΚΑΣΕΤΕΣ θα έχουν τιμή NULL στο πεδίο loaned και μόνο οι νέες εγγραφές θα παίρνουν εξ αρχής αυτόματα την τιμή ‘n’. Συνεπώς θα πρέπει να ενημερώσουμε τις παλιές εγγραφές με το παρακάτω ερώτημα:

“Ενημερώστε τον πίνακα ΚΑΣΕΤΕΣ ώστε το πεδίο loaned στις παλιές εγγραφές να έχει την τιμή (‘n’).

Q5

```
use videoclub
go
UPDATE ΚΑΣΕΤΕΣ
SET loaned=‘n’;
```

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

Q6

```
use videoclub
go
Create Trigger loaned_updater ON ΕΝΟΙΚΙΑΣΗ For Insert
as
Update ΚΑΣΕΤΕΣ Set loaned = ‘y’
From ΚΑΣΕΤΕΣ, INSERTED i
Where ΚΑΣΕΤΕΣ.ID = i.IDΚασέτας
```

Εφόσον εκτελέσουμε επιτυχώς τον παραπάνω κώδικα στον Query Editor μπορούμε να πάμε και 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 νέα εγγραφής στον πίνακα ΕΝΟΙΚΙΑΣΗ. Τέλος, ως άσκηση δημιουργήστε αντίστοιχο trigger για την περίπτωση της επιστροφής μιας ΚΑΣΕΤΑΣ που είχε προηγουμένως ενοικιαστεί.

4.2.3 Συναλλαγές/Transactions

Συναλλαγή ή δοσοληψία (transaction) καλείται ένα σύνολο λειτουργιών/ενεργειών ενημέρωσης, διαγραφής ή εισαγωγής γραμμών, το οποίο αποτελεί μία 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 μια ενέργεια αποτύχει τότε καμία από 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). Δείτε για παράδειγμα, τα παρακάτω 2 ερωτήματα, εκ των οποίων το πρώτο σκοπίμως είναι εσφαλμένο και δεν θα εκτελεστεί:

“Εισάγετε στον πίνακα ΕΝΟΙΚΙΑΣΗ την ενοικίαση της κασέτας με IDΚασέτας 2 και στην συνέχεια ενημερώστε τον πίνακα ΚΑΣΕΤΕΣ ώστε το πεδίο loaned για την κασέτα με κωδικό Id 2 να είναι ‘y’”.

Q7

```
use videoclub
go
```

```
Insert into ΕΝΟΙΚΙΑΣΗ (IDΠελάτη,IDΚασέτας,Από,Έως)
Values (2,2,'07/24/2013',Null)
Go
```

```
Update ΚΑΣΕΤΕΣ 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 παρατηρείται από τα παραπάνω αποτελέσματα το ερώτημα εισαγωγής νέας εγγραφής δεν εκτελέστηκε λόγω λάθους ημερομηνίας ('07/24/2013'), ενώ το ερώτημα ενημέρωσης εκτελέστηκε επιτυχώς. Αποτέλεσμα της πρώτης μη επιτυχημένης εκτέλεσης είναι να έχουμε ασυνέπεια δεδομένων στην βάση μας. Δηλαδή, ενώ μια κασέτα δεν φαίνεται να έχει ενοικιαστεί στο πίνακα ΕΝΟΙΚΙΑΣΗ, φαίνεται νοικιασμένη στον πίνακα ΚΑΣΕΤΕΣ.

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

Q8

```
use videoclub  
go
```

```
Declare @myerror as int  
Select @myerror = 0;
```

```
Begin Transaction
```

```
Insert into ENOIKIASH (IDΠελάτη, IDΚασέτας, Από, Έως) Values (2,2,'07/24/2013',Null)
```

```
Update ΚΑΣΕΤΕΣ 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 υπάρξει 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.

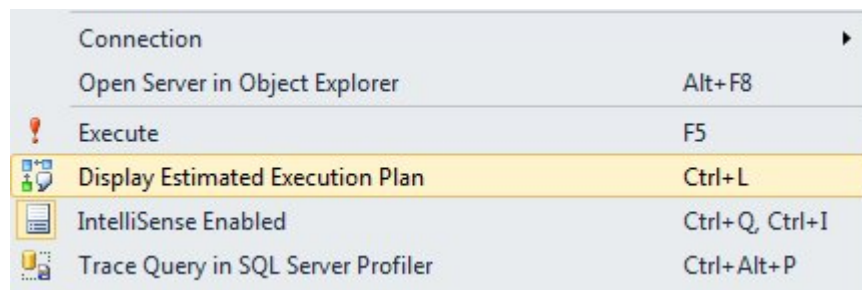
Rollback Transaction: Στην περίπτωση που υπάρξουν σφάλματα δεν αποδεχόμαστε τα αποτελέσματα των δύο ερωτημάτων μας και επιστρέφουμε στην προηγούμενη κατάσταση της βάση δεδομένων μας.

4.3. Βελτιστοποίηση Ερωτημάτων

Πριν την εκτέλεση ενός ερωτήματος από τον SQL Server προηγείται 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), που 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 το χρόνο επεξεργασίας που απαιτείται. Τελικά, επιλέγεται το λιγότερο ακριβό πλάνο εκτέλεσης. Μπορείτε να δείτε το πλάνο εκτέλεσης που ο SQL Server επέλεξε για το SQL ερώτημα σας ως εξής: Πληκτρολογήστε το ερώτημα και πατήστε Ctrl+L ή διαφορετικά με δεξί κλικ επιλέξτε Display Estimated Execution Plan, όπως φαίνεται στην Εικόνα 4.3.



Εικόνα 4.3

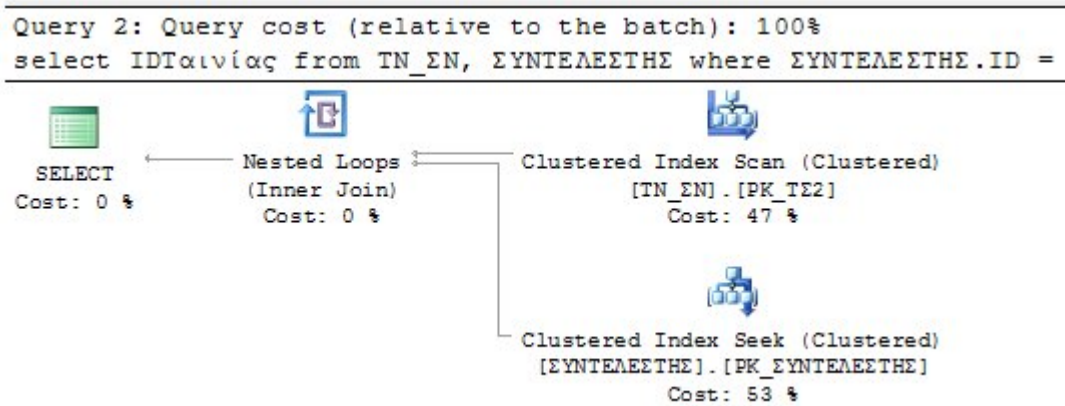
Για παράδειγμα, έστω ότι θέλουμε να απεικονισθεί το πλάνο εκτέλεσης για το παρακάτω ερώτημα:

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

```
Q1 Select IDΤαινίας
    from TN_ΣΝ, ΣΥΝΤΕΛΕΣΤΗΣ
    where ΣΥΝΤΕΛΕΣΤΗΣ.ID = TN_ΣΝ.IDΣυντελεστή and Όνομα='Grace Kelly'
```

```
IDΤαινίας
-----
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. Για να τις δείτε, κρατήστε το ποντίκι πάνω στο εικονίδιο της λειτουργίας. Εμφανίζεται ένα αναδυόμενο παράθυρο που περιέχει τις πρόσθετες πληροφορίες, που φαίνονται στην Εικόνα 4.5.

Query 2: Query cost (relative to the batch): 100%

```
select IDΤαινίας from TN_ΣΝ, ΣΥΝΤΕΛΕΣΤΗΣ wh
```

Clustered Index Scan (Clustered)	
Scanning a clustered index, entirely or only a range.	
Physical Operation	Clustered Index Scan
Logical Operation	Clustered Index Scan
Estimated Execution Mode	Row
Estimated Operator Cost	0,0032864 (47%)
Estimated I/O Cost	0,003125
Estimated CPU Cost	0,0001614
Estimated Subtree Cost	0,0032864
Estimated Number of Executions	1
Estimated Number of Rows	4
Estimated Row Size	15 B
Ordered	False
Node ID	1
Object	
[videoclub].[dbo].[TN_ΣΝ].[PK_ΤΣ2]	
Output List	
[videoclub].[dbo].[TN_ΣΝ].IDΤαινίας; [videoclub].[dbo].[TN_ΣΝ].IDΣυντελεστή	

Query executed successfully.

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/Indexes

Ένα ευρετήριο είναι μια βοηθητική δομή που σας επιτρέπει να βελτιώσετε την απόδοση των ερωτημάτων σας, μειώνοντας το μέγεθος της απαιτούμενης I/O δραστηριότητας για την ανάκτηση των ζητούμενων δεδομένων. Για παράδειγμα, έστω το παρακάτω ερώτημα:

“**Να δημιουργηθεί ευρετήριο για το όνομα του συντελεστή**”. Η εντολή SQL είναι:

Q2

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. Για παράδειγμα, στο παρακάτω ερώτημα βελτιώνουμε το πλάνο εκτέλεσης του ερωτήματος Q14 της προηγούμενης Ενότητας 4.2.1, με την χρήση ευρετηρίου για το όνομα του συντελεστή:

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

Q3

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

Η χρήση του όρου WITH επιβάλλει τη χρήση του ευρετηρίου. Στο νέο πλάνο εκτέλεσης φαίνεται ότι η επιλογή των γραμμών βάσει του ονόματος συντελεστή πλέον γίνεται μέσω του ευρετηρίου που δημιουργήσαμε. Σημειώστε ότι σε αυτό το απλοποιημένο παράδειγμα με τις λίγες γραμμές που ανακτώνται δεν υπάρχει διαφορά στους χρόνους εκτέλεσης. Για αυτό τον λόγο στην επόμενη ενότητα θα υλοποιήσουμε ένα παράδειγμα όπου θα είναι εμφανής η βελτίωση της απόδοσης του ερωτήματος με την χρήση ευρετηρίων. Τέλος, στην SQL επιτρέπεται η διαγραφή ενός ευρετηρίου όπως στο ακόλουθο ερώτημα:

“**Να διαγραφεί ο ευρετήριο IND_ΣΥΝΤΕΛΕΣΤΗΣ**”. Η εντολή SQL είναι:

Q4

Drop index ΣΥΝΤΕΛΕΣΤΗΣ.IND_ΣΥΝΤΕΛΕΣΤΗΣ;

4.3.3 Παράδειγμα βελτίωσης απόδοσης ερωτήματος με την χρήση ευρετηρίου

Τα Ευρετήρια/Indexes επιτρέπουν την σημαντική βελτίωση του χρόνου εκτέλεσης ερωτημάτων. Αυτό βέβαια είναι κάτι που γίνεται αντιληπτό όταν έχουμε μεγάλο όγκο δεδομένων και τα ερωτήματα μας απαιτούν αρκετό χρόνο για να εκτελεστούν.

Στην Ενότητα 4.2.1 με την βοήθεια του ερωτήματος Q1, εισαγάγαμε στον πίνακα ΠΕΛΑΤΗΣ 5000 εγγραφές. 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:

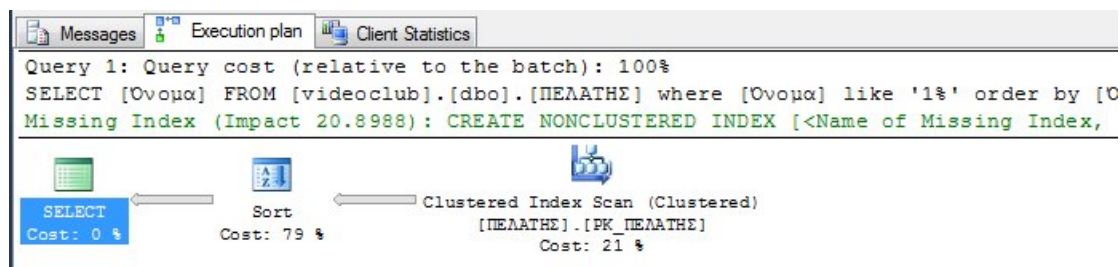
Q5

```
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 με ένα διάγραμμα, τα στάδια εκτέλεσης του ερωτήματος και το κόστος σε πόρους (CPU & I/O disks) που το κάθε ένα απαιτεί. Συγκεκριμένα, στην Εικόνα 4.4 φαίνεται το συνολικό κόστος του ερωτήματος μας να αφορά τρία στάδια:

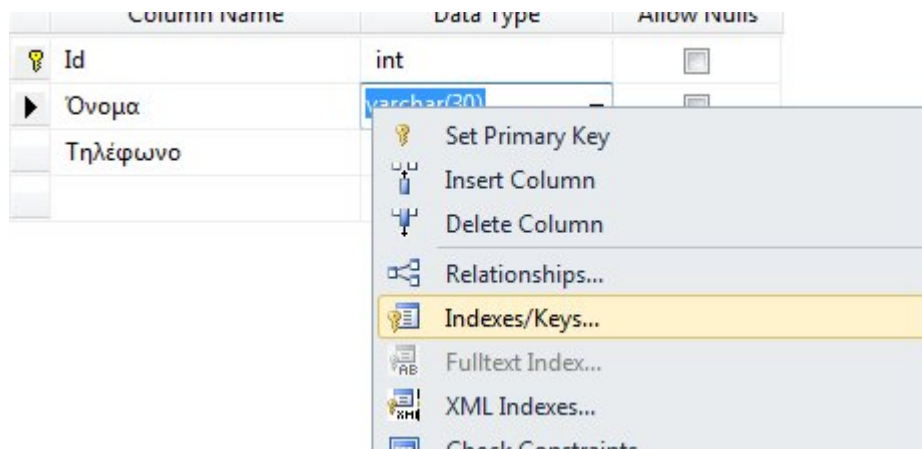
- Η αναζήτηση και επιλογή των γραμμών από τον πίνακα ΠΕΛΑΤΗΣ βάσει της συνθήκης WHERE και με την χρήση του κύριου κλειδιού του πίνακα έχει κόστος το 21% του συνολικού χρόνου. Σημειώστε ότι το ευρετήριο τύπου clustered με όνομα PK_ΠΕΛΑΤΗΣ δημιουργήθηκε εξ ορισμού από τον SQL Server στο κύριο κλειδί (στο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

Παρατηρήστε στην Εικόνα 4.6, την οδηγία που μας 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 ερωτήματος μας θα ήταν 20.8988 επί του συνολικού χρόνου.



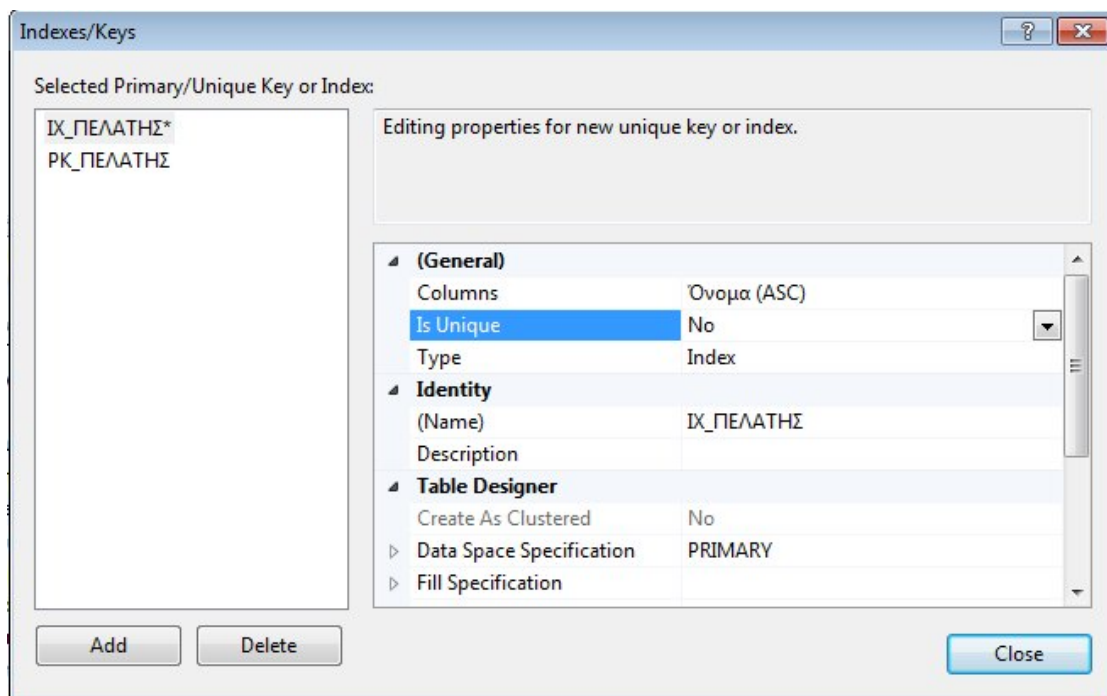
Εικόνα 4.6

Προκειμένου λοιπόν να βελτιστοποιήσουμε το αρχικό μας ερώτημα, επιλέγουμε τον πίνακα ΠΕΛΑΤΗΣ και δεξί κλικ στο Design για 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

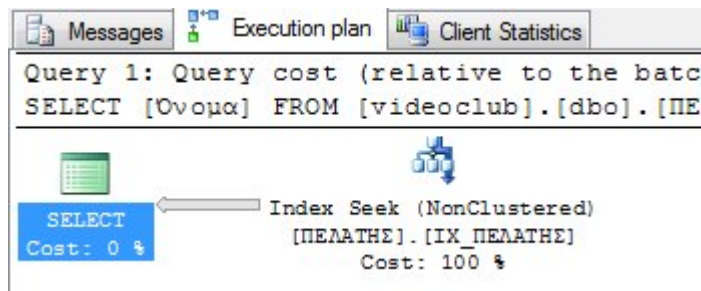
Βλέπουμε ότι υπάρχει ήδη ένα ευρετήριο με όνομα PK_ΠΕΛΑΤΗΣ (του κύριου κλειδιού) που δημιουργείται αυτόματα όταν δηλώνουμε μία στήλη ως κύριο κλειδί. Επιλέγουμε Add και στο πεδίο Columns επιλέγουμε το πεδίο Όνομα (με ASC ταξινόμηση), όπως φαίνεται στην Εικόνα 4.8. Αν θέλουμε το πεδίο να είναι μοναδικό 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 για να δούμε ότι το πλάνο εκτέλεσης άλλαξε και πλέον ο SQL Server χρησιμοποιεί το index που ορίσαμε για την αναζήτηση και ταξινόμηση του ερωτήματός μας χωρίς να σπαταλά χρόνο, όπως φαίνεται στην Εικόνα 4.9. Βέβαια τις διαφορές στους χρόνους εκτέλεσης ερωτημάτων δεν μπορούμε να τις δούμε εύκολα ακόμη και σε ένα πίνακα με 5000 εγγραφές. Αν έχετε

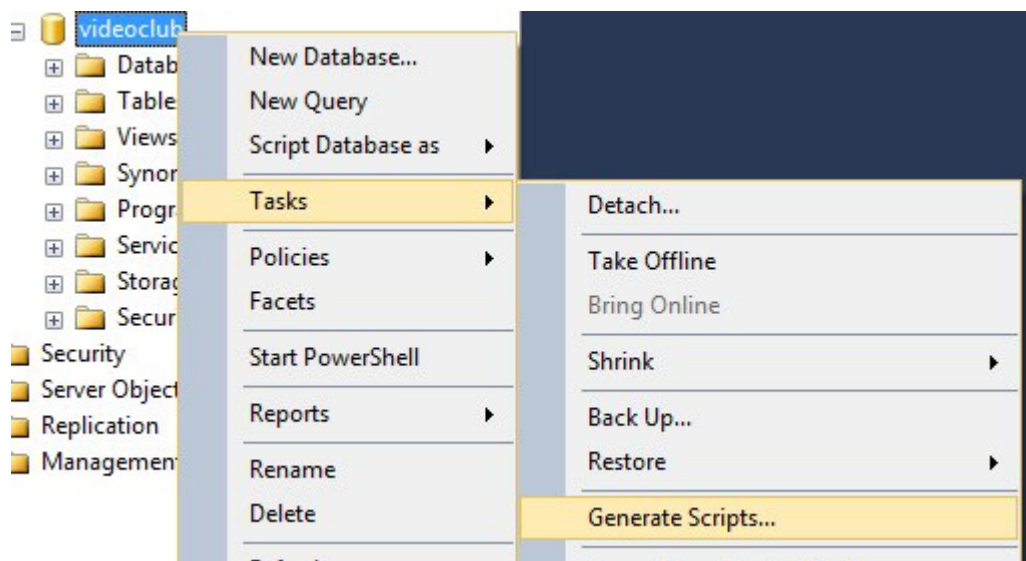
υπομονή μπορείτε να εισάγετε 2-3 εκατομμύρια πελάτες με την αποθηκευμένη διαδικασία της Ενότητας 4.2.1 και να πειραματιστείτε με τον χρόνο εκτέλεσης του ερωτήματος του παραδείγματος μας.



Εικόνα 4.9

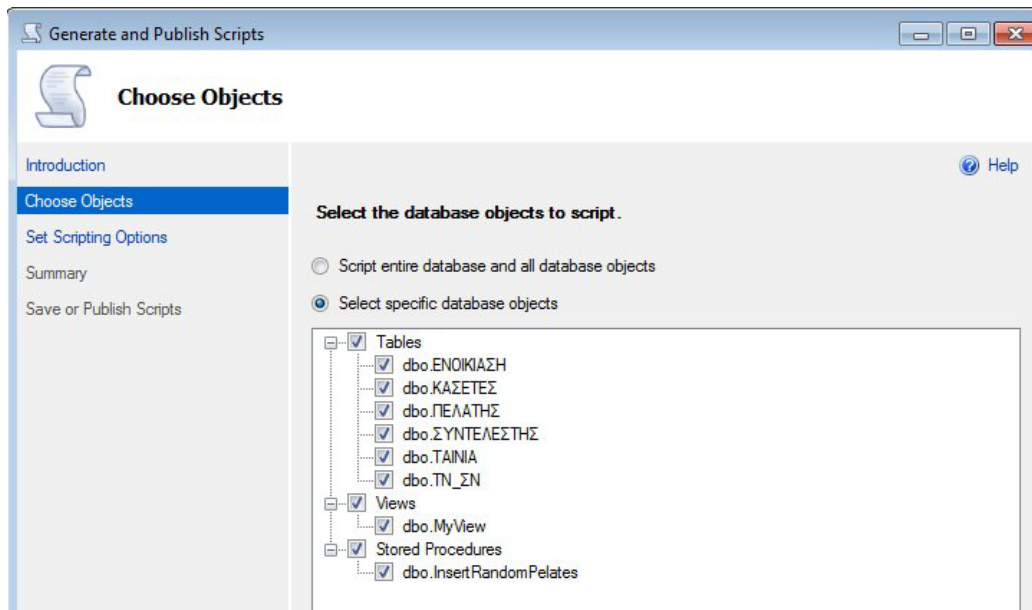
4.4. Εξαγωγή του κώδικα της βάσης δεδομένων

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, όπως φαίνεται στην Εικόνα 4.10, κάνουμε δεξί κλικ στη βάση δεδομένων videoclub→Tasks→Generate SQL Script.



Εικόνα 4.10

Ανοίγει το παράθυρο της Εικόνας 4.11, στο οποίο επιλέγουμε είτε την επιλογή «Script entire database and all database objects» είτε αν θέλουμε να επιλέξουμε κάποιο υποσύνολο των αντικειμένων της βάσης δεδομένων επιλέγουμε «select specific database objects». Στο παράδειγμά μας, επιλέξαμε την δεύτερη επιλογή όπως φαίνεται στην Εικόνα 4.11 και στην συνέχεια επιλέξαμε τα αντικείμενα που θέλουμε να εξαχθούν σε εντολές SQL.

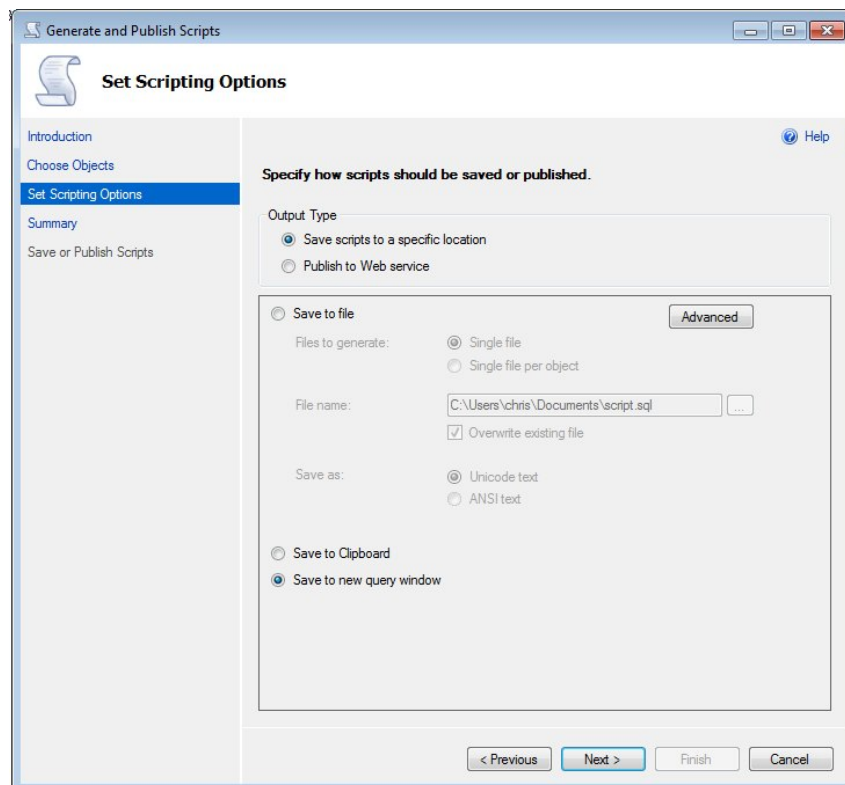


Εικόνα 4.11

Στη συνέχεια πρέπει να επιλέξουμε τον τρόπο που θα αποθηκεύσουμε τον κώδικα που θα παραχθεί. Μπορούμε είτε:

- 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 επιλέξαμε στο παράδειγμά μας την τρίτη επιλογή, όπως φαίνεται στην Εικόνα 4.12:



Εικόνα 4.12

Τέλος, πατάμε Next και σε ένα νέο παράθυρο ερωτήματος προβάλλεται ο κώδικας που δημιουργεί όλα τα αντικείμενα της βάσης δεδομένων όπως φαίνεται στην Εικόνα 4.13.

```

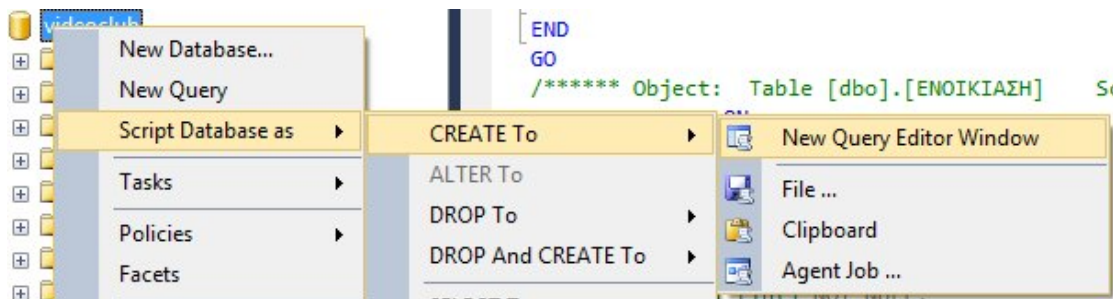
/***** Object: Table [dbo].[ΕΝΟΙΚΙΑΣΗ]   Script Date: 4/7/2013 9:10:10 πμ *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[ΕΝΟΙΚΙΑΣΗ](
    [IDΠελάτη] [int] NOT NULL,
    [IDΚασέτας] [int] NOT NULL,
    [Από] [date] NOT NULL,
    [Εως] [date] NULL,
    CONSTRAINT [PK_ΕΝΟΙΚΙΑΣΗ] PRIMARY KEY CLUSTERED
    (
        [IDΠελάτη] ASC,
        [IDΚασέτας] ASC,
        [Από] ASC
    )
    WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON) 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 να έχει δημιουργηθεί η βάση δεδομένων (create), για να μπορούν να φτιαχτούν όλα τα αντικείμενά της με τον κώδικα που μόλις παράχθηκε.

Θυμίζουμε ότι τον κώδικα δημιουργίας της βάσης δεδομένων μπορούμε να τον παράγουμε από την επιλογή Script Database as → Create To → New Query Editor Window, όπως φαίνεται στην Εικόνα 4.14 ή πιο απλά μπορούμε να την φτιάξουμε από την αρχή χειρωνακτικά όπως δείξαμε στην Ενότητα 2.1.1.



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 videoclub, βάσει του 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			
ΚΑΣΕΤΕΣ	Manager	Manager	Manager	Manager
	Employee	Employee	Employee	
	Customer			
ΤΝ_ΣΝ	Manager	Manager	Manager	Manager
	Employee	Employee	Employee	
	Customer			
ΤΑΙΝΙΑ	Manager	Manager	Manager	Manager
	Employee	Employee	Employee	
	Customer			

Παρακάτω θα δημιουργήσουμε ένα login με το οποίο θα έχουμε πρόσβαση στον sql server. Στη συνέχεια θα δημιουργήσουμε ένα χρήστη - employee, ο οποίος θα αναφέρεται σε 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 στη βάση videoclub.

```
create login employee with password = '123'
go
```

```
use videoclub
create user employee for login employee with default_schema=[dbo]
go
```

4.5.1 Εκχώρηση δικαιωμάτων χρήστη

Για την εκχώρηση ενός προνομίου πρέπει ο διαχειριστής της βάσης δεδομένων 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];
```

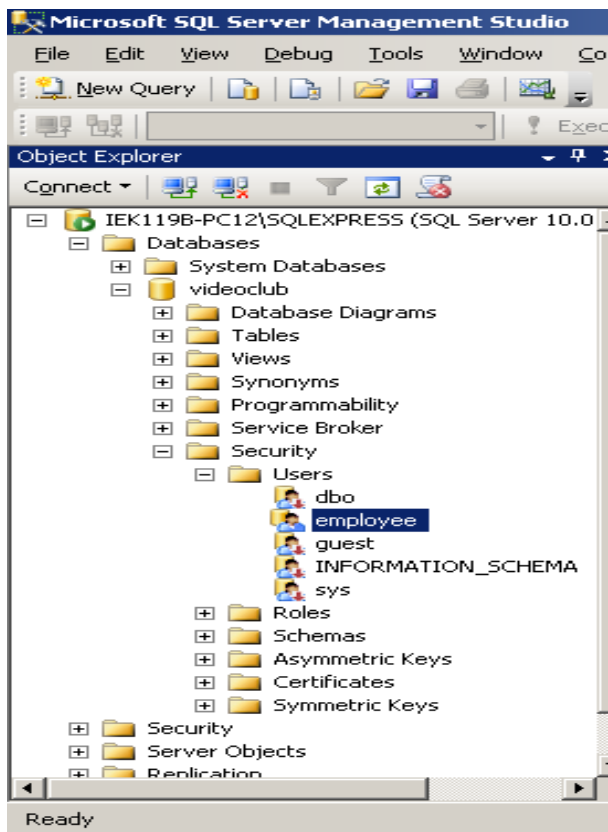
- privilege_name είναι το προνόμιο ή δικαίωμα πρόσβασης το οποίο εκχωρείται στο χρήστη. Μερικά από τα δικαιώματα πρόσβασης είναι ALL, EXECUTE και SELECT
- object_name είναι το όνομα ενός αντικειμένου της βάσης (πίνακας, όψη)
- user_name είναι το όνομα του χρήστη στον οποίο εκχωρείται ένα δικαίωμα πρόσβασης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

4.5.2 Εκχώρηση δικαιωμάτων στο χρήστη Employee με κώδικα SQL.

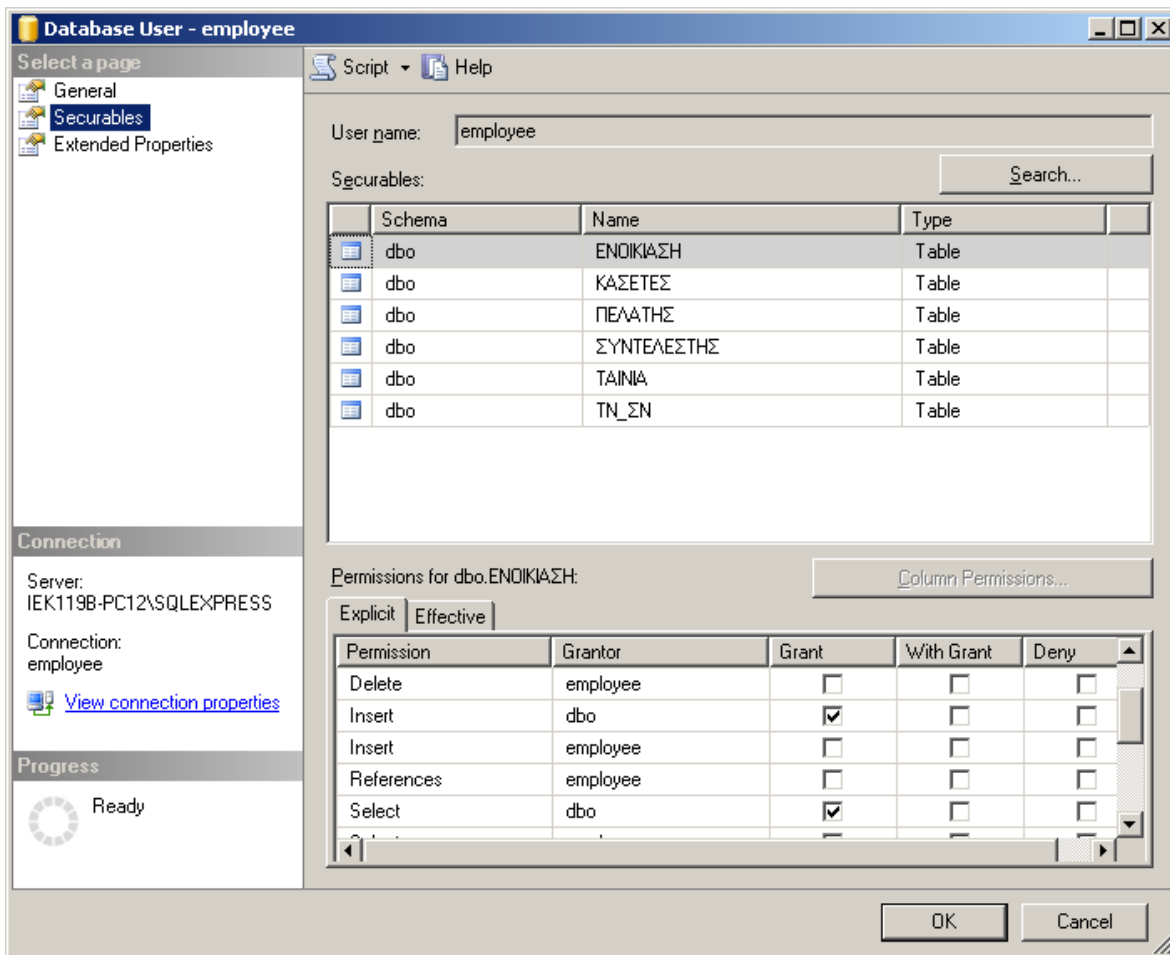
Σε αυτή την ενότητα θα περιγράψουμε την εκχώρηση δικαιωμάτων στον χρήστη employee. Ο employee θέλουμε να έχει πλήρη πρόσβαση στη βάση δεδομένων, ώστε να μπορεί να δημιουργεί καινούριες εγγραφές και να τροποποιεί τις υπάρχουσες.

```
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 ΚΑΣΕΤΕΣ to employee  
grant select, insert, update on ΤΝ_ΣΝ to employee  
grant select, insert, update on ΤΑΙΝΙΑ to employee
```

2) 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 κλικ στο Users όπου πρέπει να υπάρχει ο νέος χρήστης.



3) Αφού επαληθεύσουμε τη δημιουργία του χρήστη employee κάνουμε διπλό κλικ στον χρήστη και στο παράθυρο που θα εμφανιστεί επιλέγουμε το Securables για να επαληθεύσουμε ότι έχει πάρει τα δικαιώματα που θέλουμε.

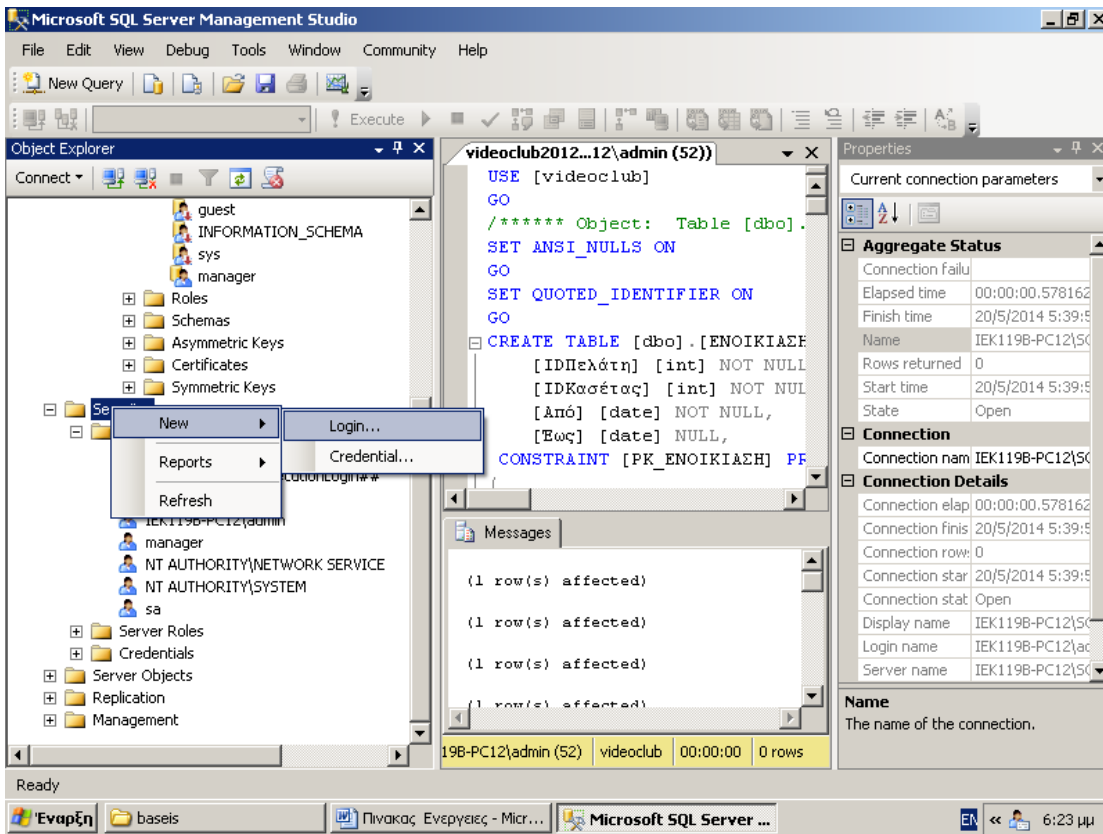


4.5.3 Εκχώρηση δικαιωμάτων στο χρήστη Manager με γραφικό τρόπο

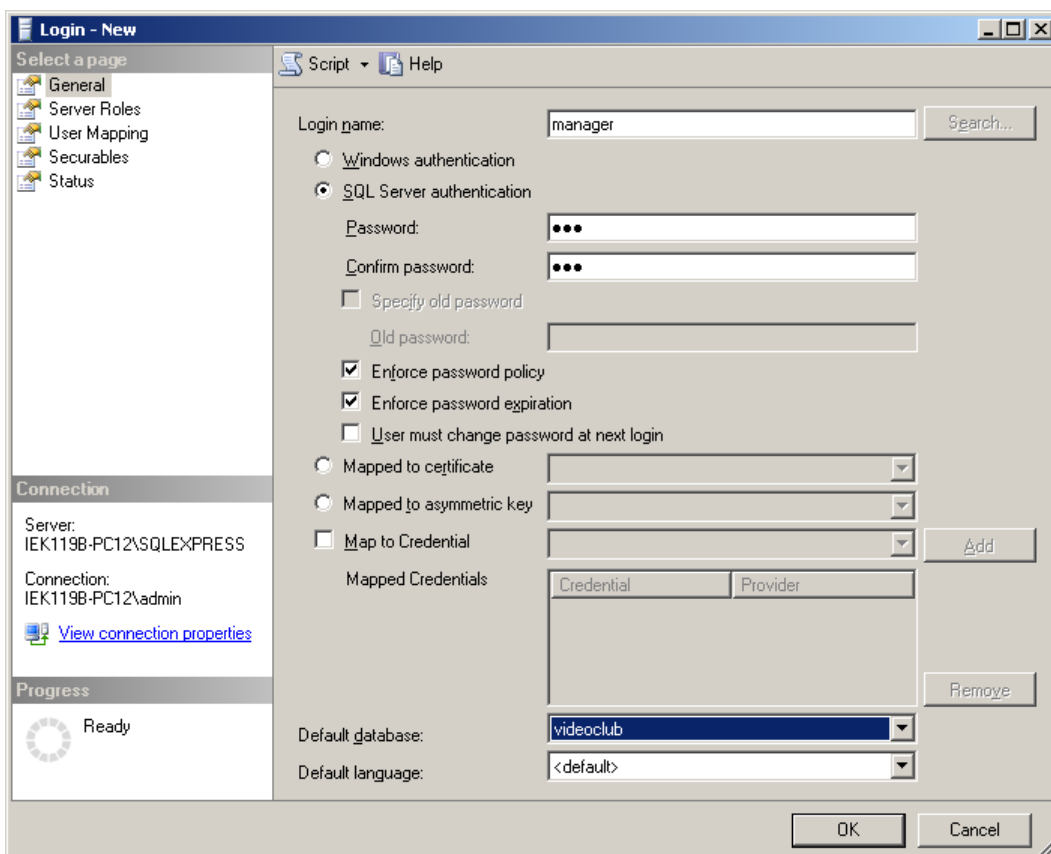
Σε αυτή την ενότητα θα δείξουμε πως γίνεται η εκχώρηση δικαιωμάτων στο χρήστη Manager με γραφικό τρόπο. Είναι η ίδια ακριβώς διαδικασία όπως στην προηγούμενη ενότητα μόνο που τώρα γίνεται με γραφικό τρόπο..

Για να δημιουργήσουμε τον χρήστη Manager ακολουθούμε τα εξής βήματα:

Βήμα 1 : Επιλέγουμε το Security -> New -> Login στο κεντρικό περιβάλλον του αριστερού pane στον sql server.

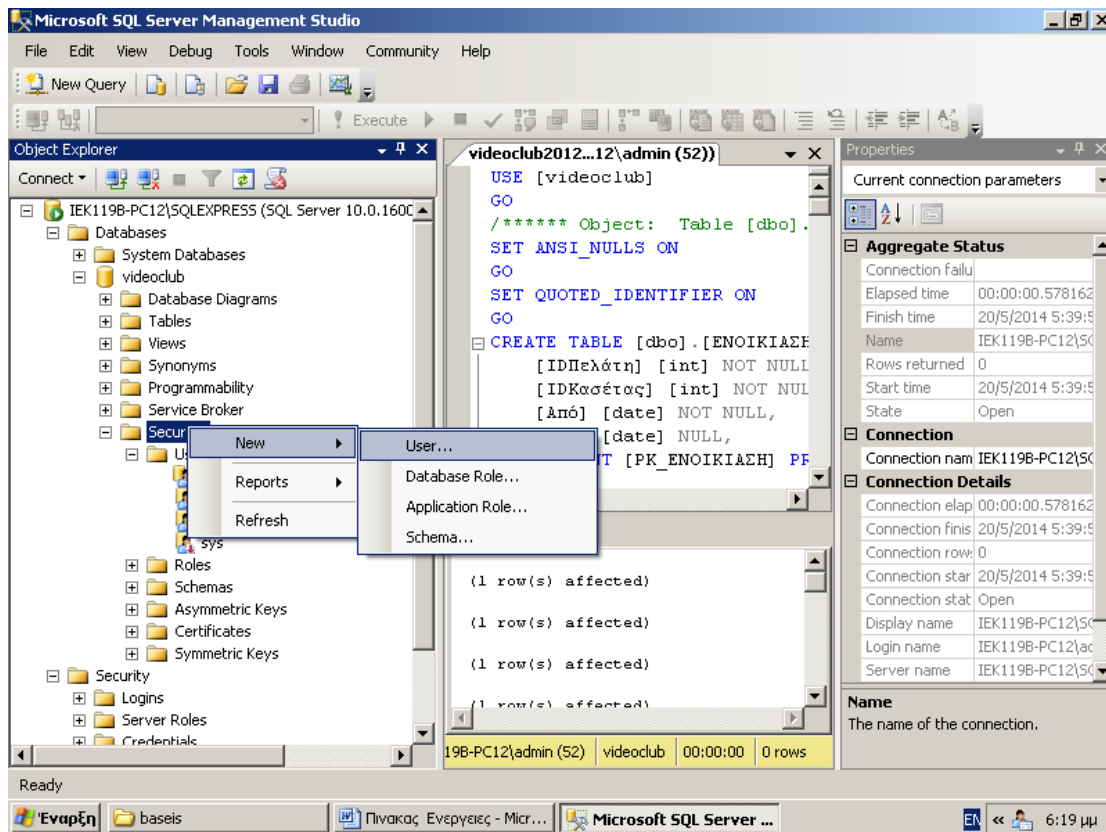


Βήμα 2: Εμφανίζεται το παρακάτω παράθυρο όπου πληκτρολογούμε στο Login name: manager. Κλικ στο Sql Server Authentication -> Password:123. Confirm password: 123. Αποεπιλέγουμε το User must change password at next login. Επιλέγουμε την βάση από το Default database: videoclub και πατάμε OK.

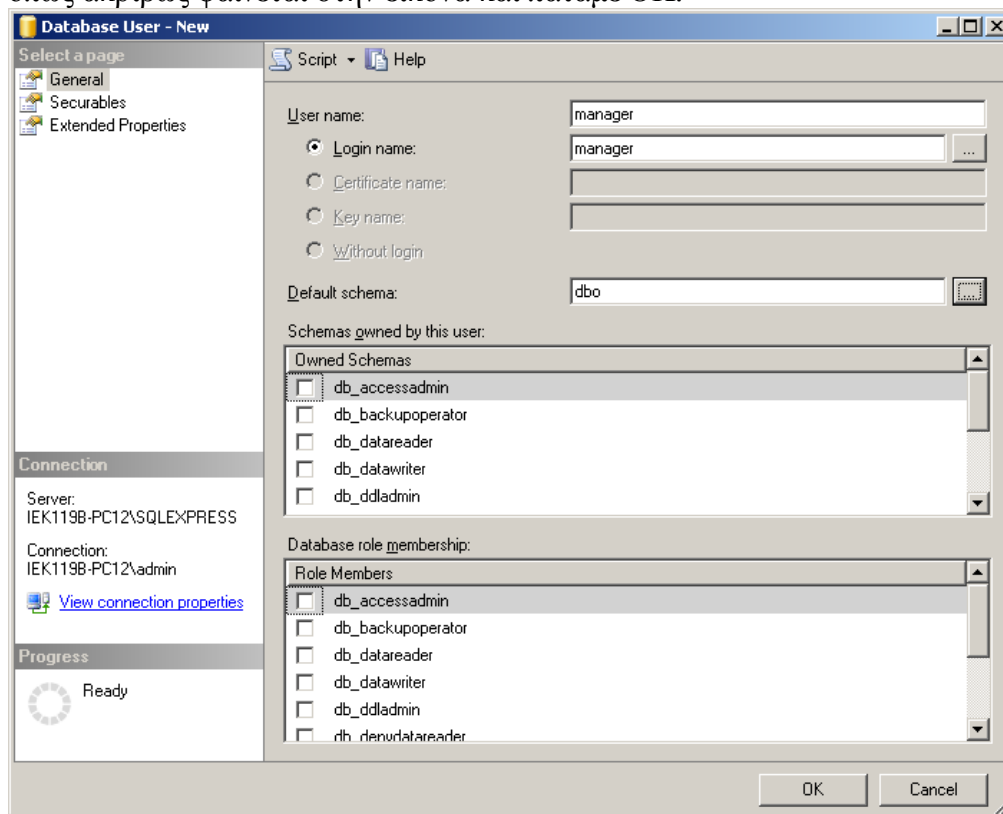


Βήμα 3:

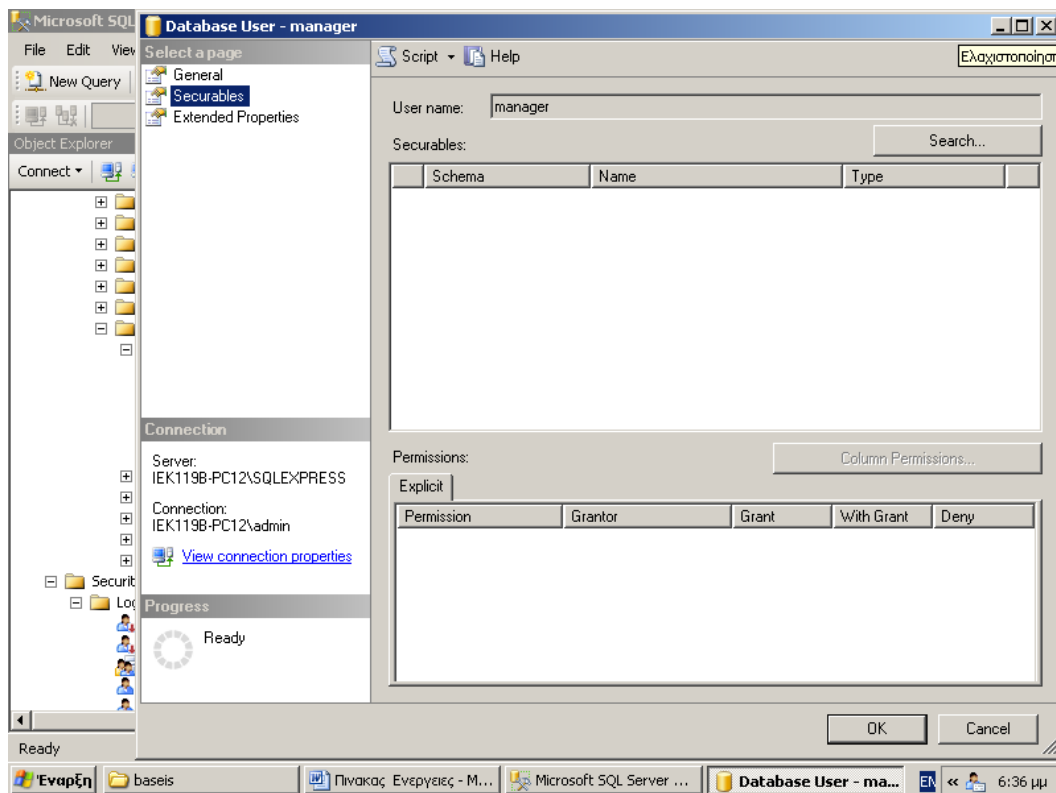
Δεξί κλικ στο Security -> New -> User στο περιβάλλον της βάσης δεδομένων videoclub



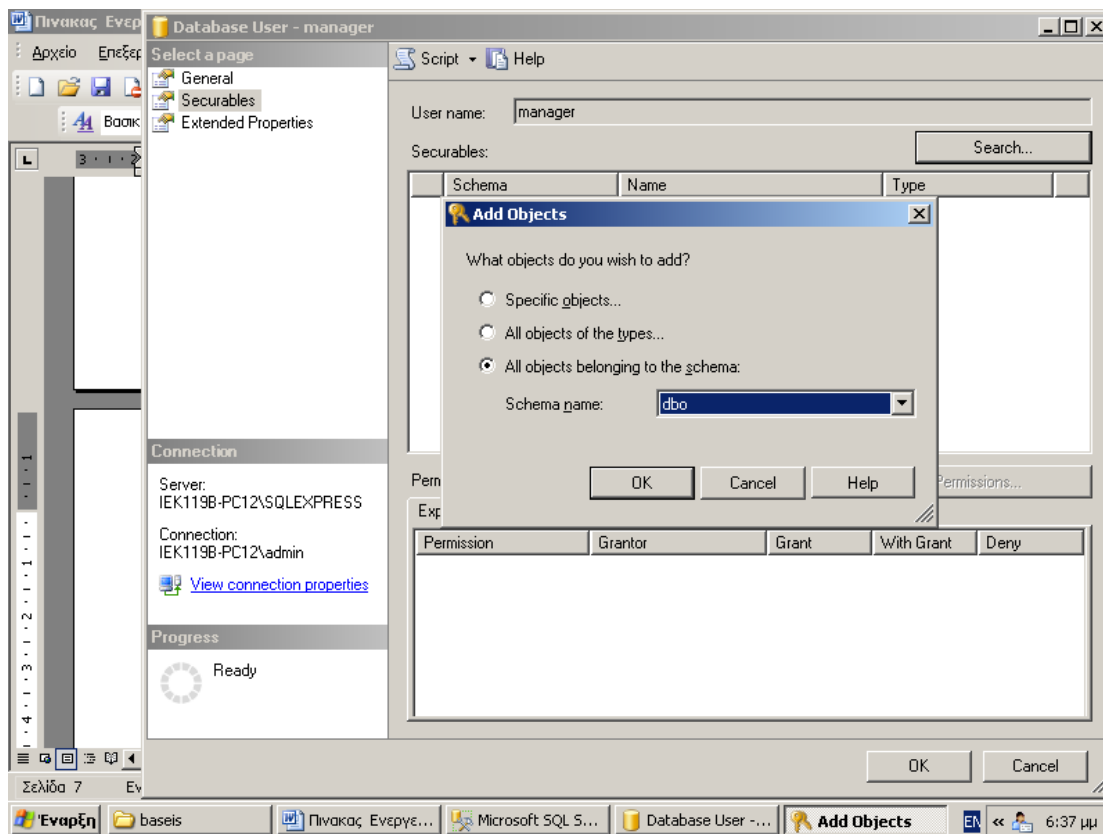
Βήμα 4: Πληκτρολογούμε στα πεδία User name και στο Login name την λέξη manager. Έτσι όπως ακριβώς φαίνεται στην εικόνα και πατάμε OK.



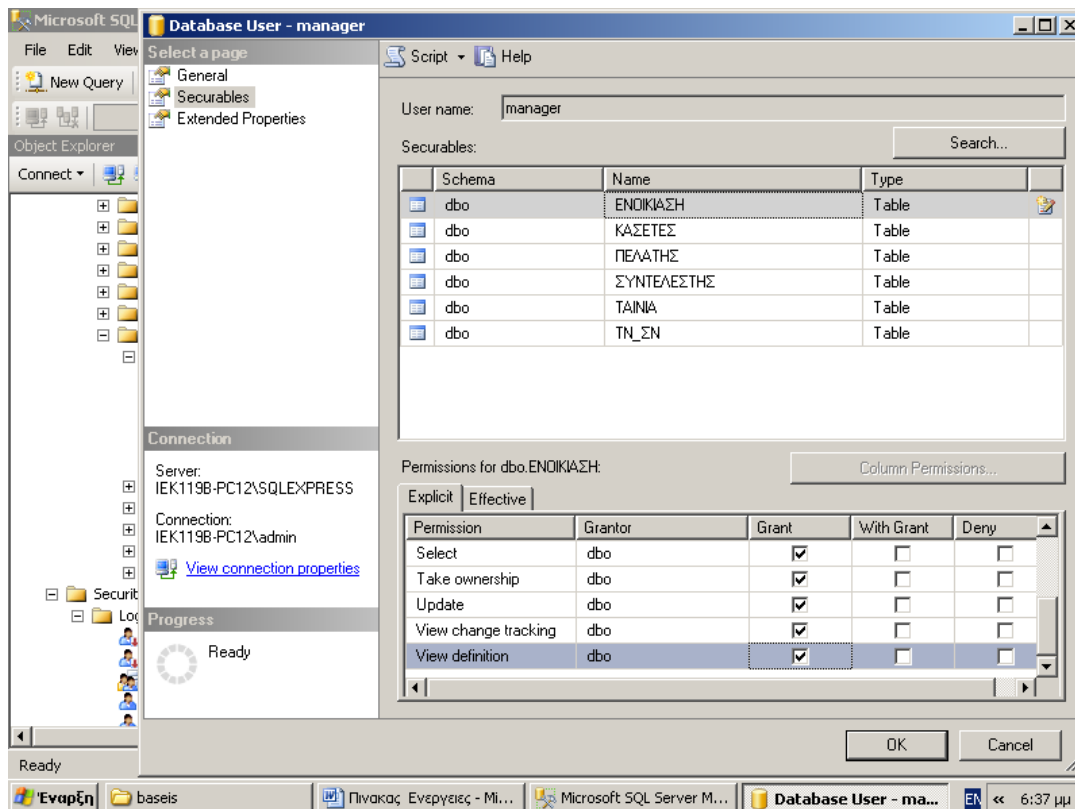
Βήμα 5: Στη συνέχεια κάνουμε ξανά δεξί κλικ στην επιλογή Security, επιλέγουμε το Users και μετά στο χρήστη manager που δημιουργήσαμε κάνουμε διπλό κλικ και επιλέγουμε το Securables.



Βήμα 6: Πατάμε το Search και στο παράθυρο που εμφανίζεται, επιλέγουμε το All objects belonging to the schema και στη συνέχεια στο schema name το dbo.



Με αυτόν τον τρόπο εμφανίζονται οι πίνακες της βάσης, ώστε να εκχωρήσουμε τα δικαιώματα σε κάθε ένα ξεχωριστά. Για παράδειγμα επιλέγουμε τον πίνακα Ενοικίαση και τσεκάρουμε όλα τα κουτάκια στη στήλη Grant. Η ίδια διαδικασία πρέπει να γίνει σε όλους τους πίνακες.



SOS: Η διαδικασία πρέπει να γίνει για όλους τους πίνακες πριν πατήσουμε OK, γιατί διαφορετικά θα πρέπει να ξαναμπούμε στο Securables και να ακολουθήσουμε τα βήματα από την αρχή. (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 ένα ή περισσότερους πίνακες. Η σύνταξη είναι η ακόλουθη:

Ανάλυση της εντολής **Revoke**.

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

Για παράδειγμα: Revoke select on employee FROM customer.

Αυτή η εντολή θα ανακαλέσει το δικαίωμα εμφάνισης που έχει ο χρήστης customer στον πίνακα employee. 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 περισσότερους από έναν

χρήστες, τότε θα μπορεί να εμφανίζει δεδομένα από αυτόν τον πίνακα μέχρι αυτοί που τα έχουν εκχωρήσει να κάνουν αναίρεση. Δεν μπορεί να γίνει αναίρεση δικαιωμάτων αν αυτά αρχικά δεν έχουν εκχωρηθεί από εσάς.

4.5.5 Άρνηση δικαιωμάτων σε χρήστη

Σε αυτή 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 employee To customer

Αυτή η εντολή απαγορεύει στο χρήστη customer να ενημερώσει τα περιεχόμενα του πίνακα employee.

4.6 Άσκηση:

1) Να γίνει η εκχώρηση δικαιωμάτων στο χρήστη Customer και με τους δυο τρόπους.

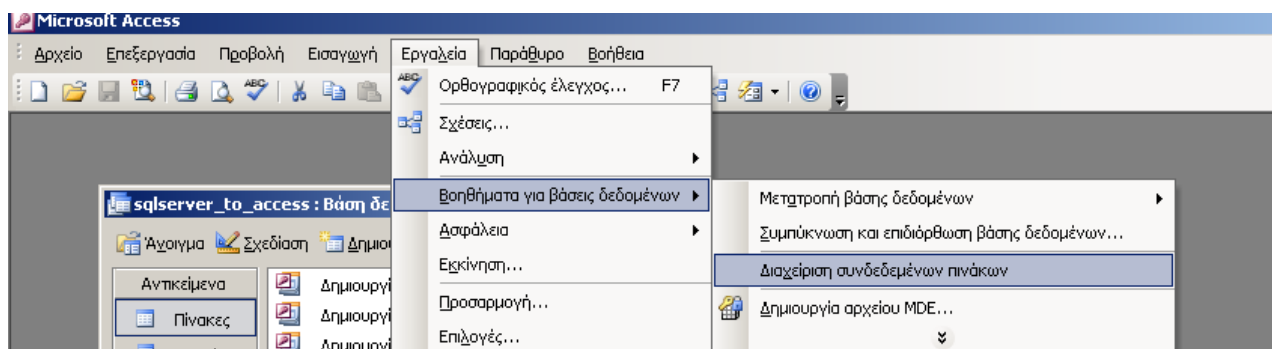
Παρατήρηση: Η διαδικασία είναι η ίδια με τα βήματα που χρησιμοποιήθηκαν στο χρήστη Employee και στο χρηστή Manager.

ΚΕΦΑΛΑΙΟ 5 – ΔΗΜΙΟΥΡΓΙΑ ΦΟΡΜΩΝ ΓΙΑ ΤΗΝ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ VIDEO CLUB

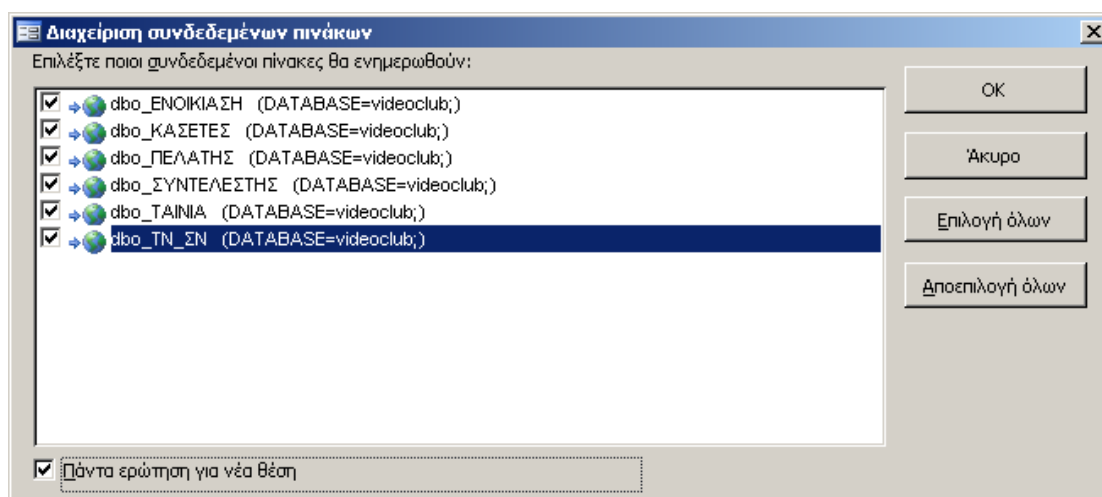
Προκειμένου να δημιουργήσουμε τις απαραίτητες φόρμες για την βάση δεδομένων Videoclub, θα πρέπει να κατεβάσουμε το σχετικό zip αρχείο από το site του μαθήματος: <http://delab.csd.auth.gr/courses/dpmsdb/index.html> (9η Διάλεξη)

5.1. Δημιουργία συνδεδεμένων πινάκων από τον SQL Server στην Access 2003 του Microsoft office.

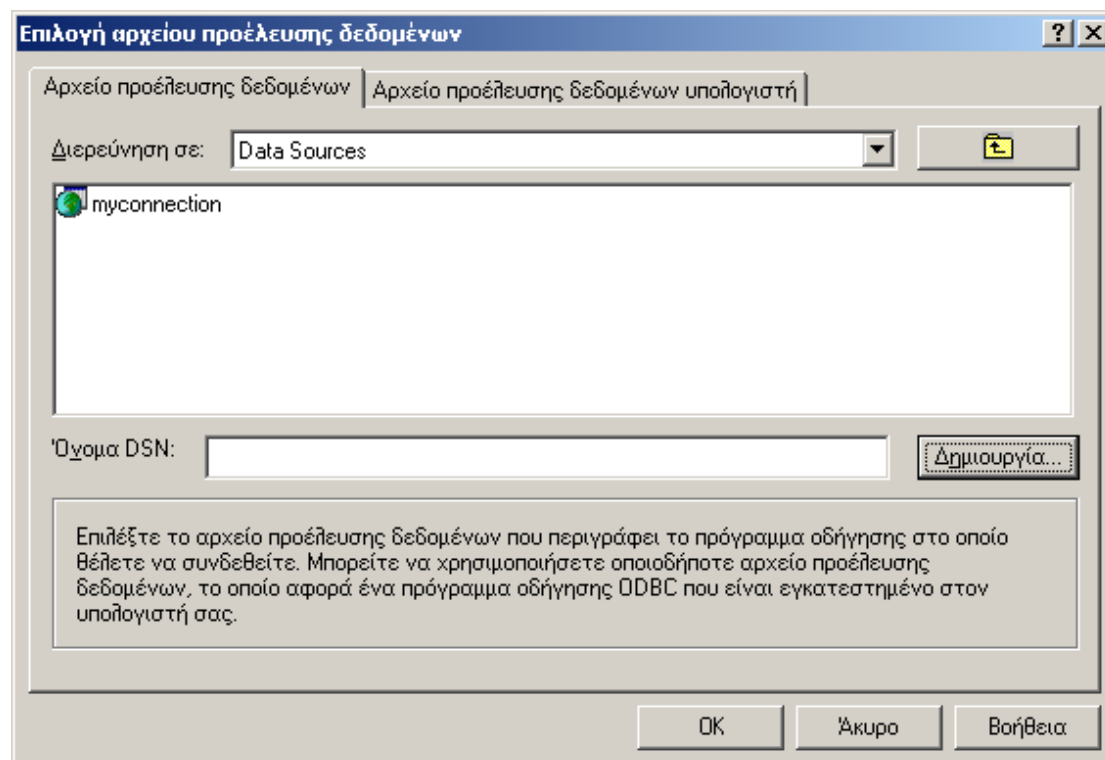
Βήμα 1: Ανοίγουμε το αρχείο της Access πάμε στην επιλογή **Εργαλεία -> Βοηθήματα για βάσεις δεδομένων -> Διαχείριση συνδεδεμένων πινάκων.**



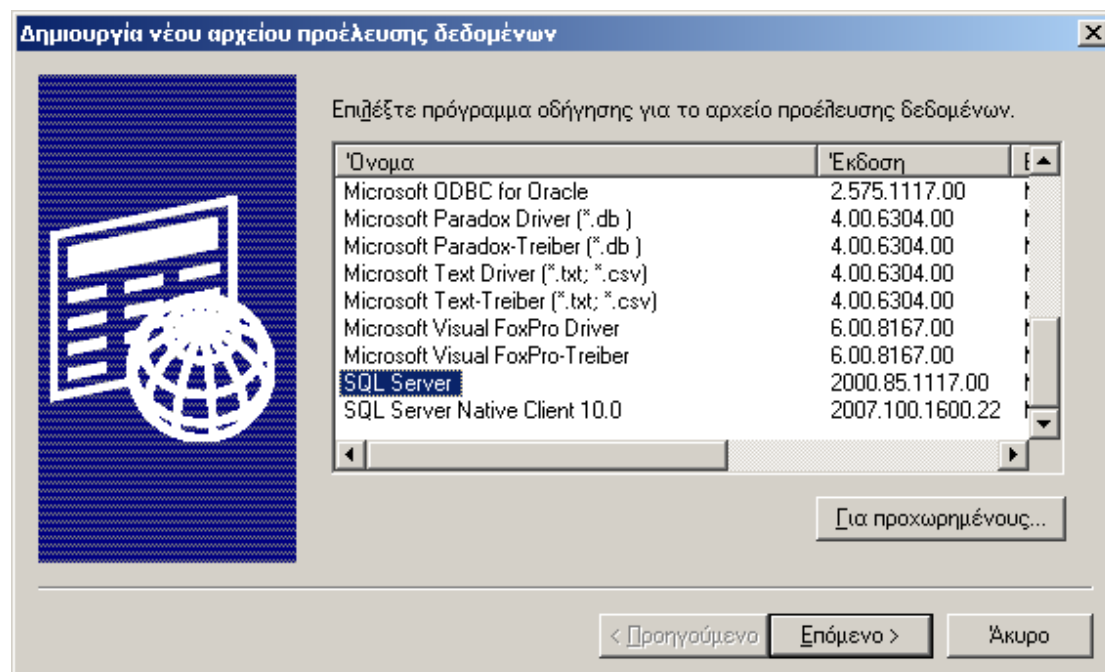
Βήμα 2: Επιλογή όλων των πινάκων καθώς και της επιλογής "Πάντα ερώτηση για νέα θέση"



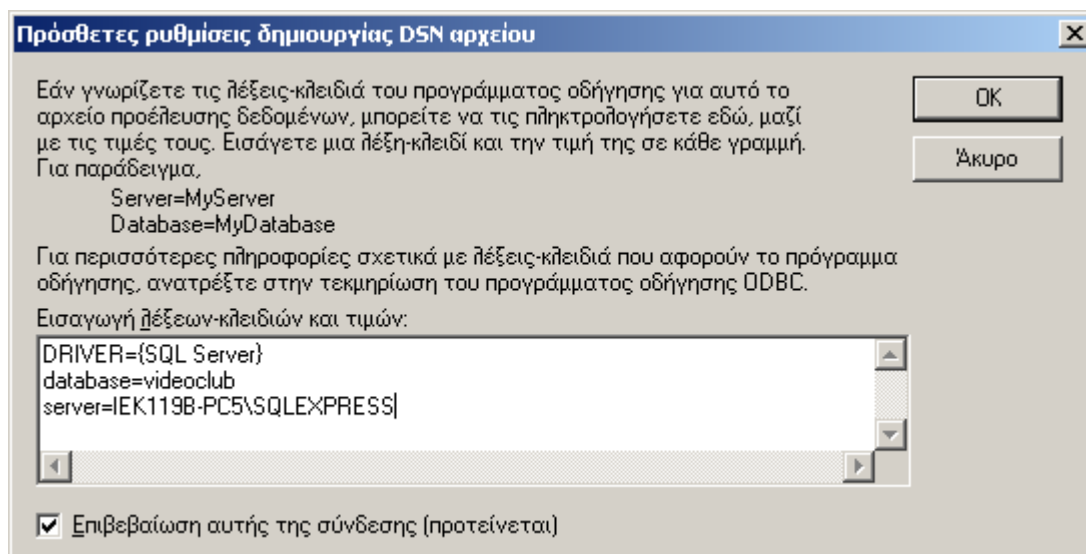
Βήμα 3: Πατάμε OK και βγαίνει το παρακάτω παράθυρο. Κάνουμε κλικ στο κουμπί «Δημιουργία»



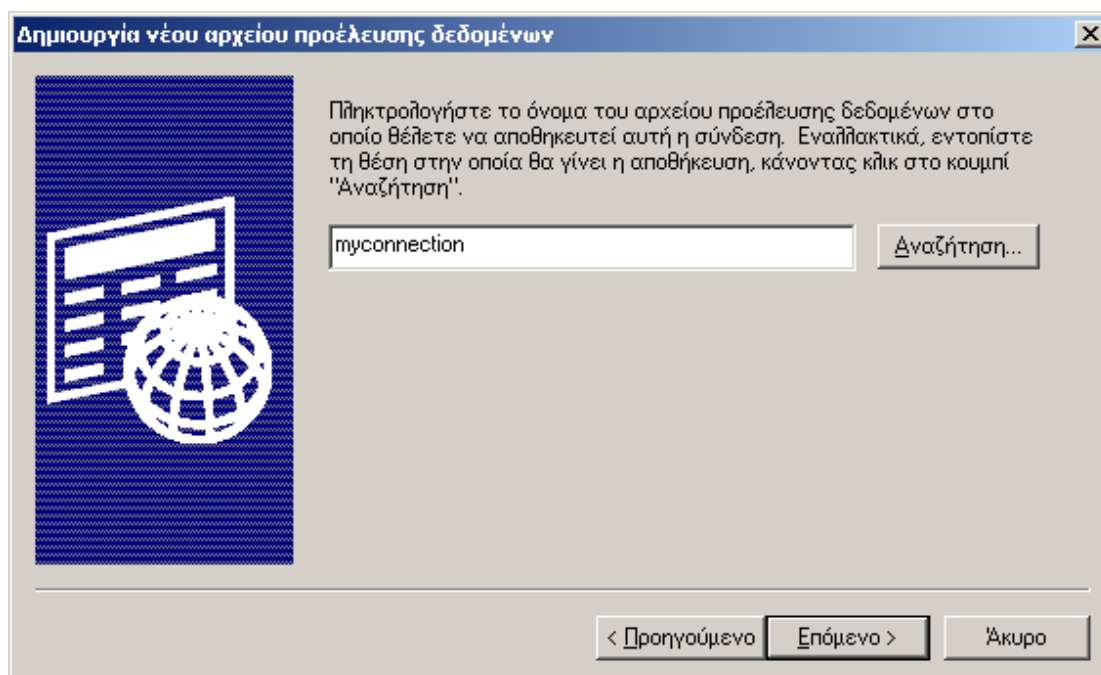
Βήμα 4: Επιλέγουμε την επιλογή SQL Server και στη συνέχεια το κουμπί Για προχωρημένους..



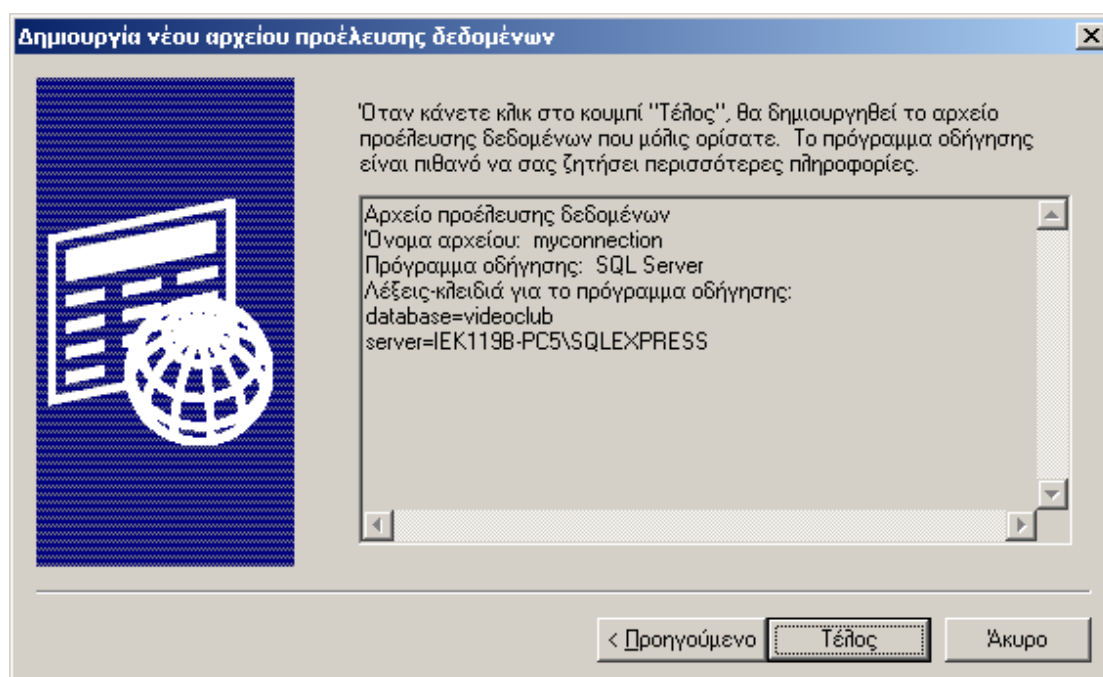
Βήμα 5: Συμπληρώνουμε όπως δείχνει το παρακάτω παράθυρο το όνομα της βάσης μας και το όνομα του server μας (όπου server βάζουμε το όνομα με το οποίο συνδέεται ο sql server στο συγκεκριμένο υπολογιστή που βρισκόμαστε)



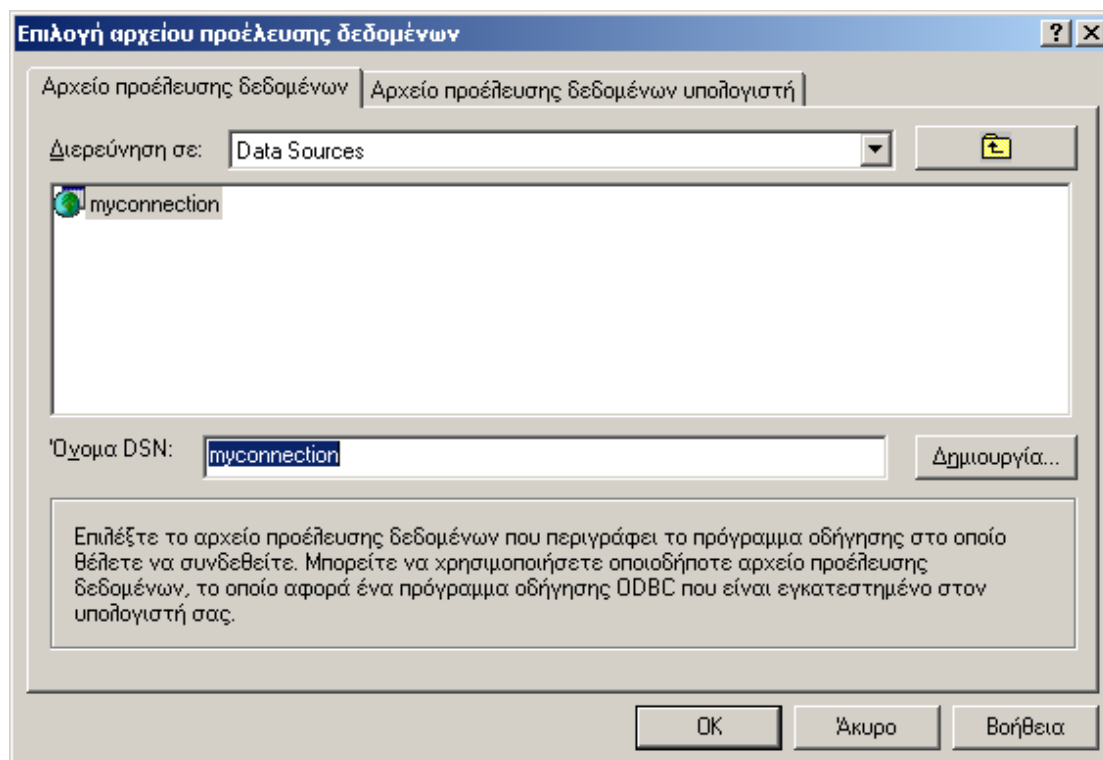
Βήμα 6: Στο πεδίο Αναζήτησης συμπληρώνουμε : myconnection και πατάμε Επόμενο



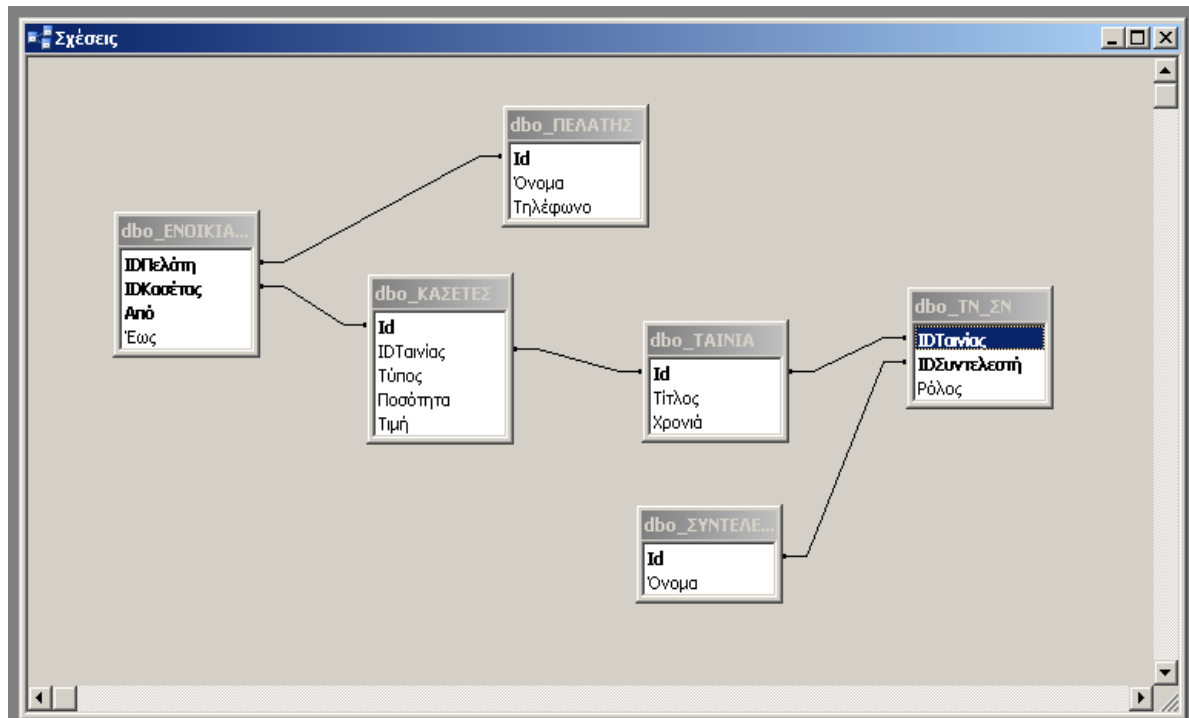
Βήμα 7: Πατάμε το κουμπί Τέλος



Βήμα 8: Επιλέγουμε το myconnection και πατάμε OK.



Βήμα 9: **Προσοχή!** Από την επιλογή Εργαλεία -> Σχέσεις θα πρέπει να συνδέσουμε ξανά τους πίνακες μεταξύ τους από την αρχή.



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

5.2.1. Δημιουργία μιας απλής φόρμας εισαγωγής στοιχείων Πελατών.

Στόχος είναι να δημιουργήσουμε μια φόρμα εισαγωγής για τους Πελάτες.

Επιλέγουμε: Δημιουργία Φόρμας με τη χρήση του οδηγού φορμών. Στο πεδίο Πίνακες/Ερωτήματα επιλέγουμε τον πίνακα που μας ενδιαφέρει και πατώντας το >> επιλέγουμε όλα τα πεδία του πίνακα για να περάσουν στη φόρμα.

Οδηγός φορμών

Ποια πεδία θέλετε στη φόρμα;
Έχετε τη δυνατότητα να επιλέξετε από περισσότερους από έναν πίνακες ή ερωτήματα.

Πίνακες/Ερωτήματα
Πίνακας: dbo_ΠΕΛΑΤΗΣ

Διαθέσιμα πεδία:

Επιλεγμένα πεδία:
Id
Όνομα
Τηλέφωνο

Άκυρο < Προηγούμενο Επόμενο > Τέλος

Πατάμε Επόμενο

Οδηγός φορμών

Τι διάταξη θέλετε να έχει η φόρμα;

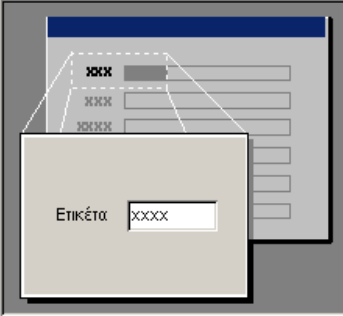
Στήλης
Πίνακα
Φύλλου δεδομένων
Στοιχισμένη
Συγκεντρωτικός Πίνακας
Συγκεντρωτικό Γράφημα

Άκυρο < Προηγούμενο Επόμενο > Τέλος

Πατάμε Επόμενο

Οδηγός φορμών

Ποιο στυλ προτιμάτε;

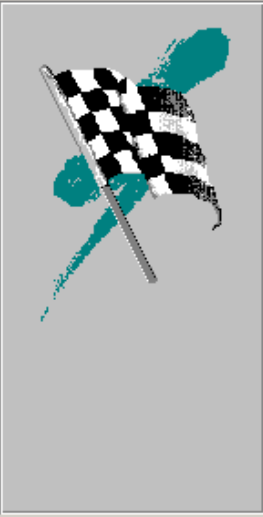


- Αμμόλιθος
- Αποστολή
- Βιομηχανικό
- Διεθνής
- Ζωγραφική Sumi
- Λίθος
- Μιλμετρέ
- Πολυχρωμία
- Ριζόχαρτο
- Τυπικό**

Άκυρο < Προηγούμενο Επόμενο > Τέλος

Πατάμε Επόμενο

Οδηγός φορμών



Τι τίτλο θέλετε να έχει η φόρμα;

φρμ_Εισαγωγή_ΠΕΛΑΤΗΣ

Αυτές είναι όλες οι πληροφορίες που χρειάζεται ο Οδηγός για να δημιουργήσει τη φόρμα.

Θέλετε να ανοίξει η φόρμα ή να τροποποιήσετε τη σχεδίασή της;

Ανοιγμα της φόρμας για την προβολή ή καταχώρηση πληροφοριών.

Τροποποίηση σχεδίασης της φόρμας.

Να εμφανιστεί βοήθεια για την εργασία με τη φόρμα;

Άκυρο < Προηγούμενο Επόμενο > Τέλος

Προκειμένου να θυμόμαστε ότι πρόκειται για φόρμα καλό είναι να μετονομάσουμε την φόρμα που μόλις φτιάξαμε σε φρμΕισαγωγή_ΠΕΛΑΤΗΣ

Και Τέλος.

5.2.2 Δημιουργία Κύριας και δευτερεύουσας φόρμας

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

Η ενότητα αυτή αφορά πίνακες που σχετίζονται ως οντότητες στο E-R με σχέσεις 1 προς πολλά.

Στόχος μας είναι η δημιουργία κύριας και δευτερεύουσας φόρμας προκειμένου να καταχωρούνται τα αντίτυπα κασετών της ίδιας ταινίας.

Επιλέγουμε: Δημιουργία Φόρμας με τη χρήση του οδηγού φορμών. Στο πεδίο Πίνακες/Ερωτήματα επιλέγουμε τον πίνακα **dbo_TAINIA** και πατώντας το >> επιλέγουμε όλα τα πεδία του πίνακα για να περάσουν στη φόρμα.

Οδηγός φορμών

Ποια πεδία θέλετε στη φόρμα;
Εχετε τη δυνατότητα να επιλέξετε από περισσότερους από έναν πίνακες ή ερωτήματα.

Πίνακες/Ερωτήματα
Πίνακας: dbo_TAINIA

Διαθέσιμα πεδία:

Επιλεγμένα πεδία:
Id
Τίτλος
Χρονιά

Ακυρο < Προηγούμενο Επόμενο > Τέλος

Στη συνέχεια και χωρίς να πατήσουμε Επόμενο επιλέγουμε τον πίνακα **dbo_Κασέτες** και παίρνουμε όλα τα πεδία του εκτός από το IDΤαινίας.

Οδηγός φορμών

Ποια πεδία θέλετε στη φόρμα;
Εχετε τη δυνατότητα να επιλέξετε από περισσότερους από έναν πίνακες ή ερωτήματα.

Πίνακες/Ερωτήματα
Πίνακας: dbo_ΚΑΣΕΤΕΣ

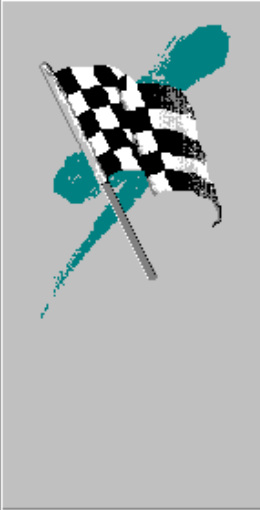
Διαθέσιμα πεδία:
IDΤαινίας

Επιλεγμένα πεδία:
dbo_TAINIA.Id
Τίτλος
Χρονιά
dbo_ΚΑΣΕΤΕΣ.Id
Τύπος
Ποσότητα
Τιμή

Ακυρο < Προηγούμενο Επόμενο > Τέλος

Στο σημείο αυτό και επειδή στο τέλος θα έχουμε πολλές κύριες – δευτερεύουσες φόρμες προκειμένου να ξέρουμε ποια αντιστοιχεί σε ποια, μετονομάζουμε την κύρια και την δευτερεύουσα σε φρμ_ΤΑΙΝΙΑ_κύρια_1. Αντίστοιχα μετονομάζουμε την δευτερεύουσα φόρμα σε φρμ_ΚΑΣΕΤΕΣ_Δευτερεύουσα_1 και πατάμε τέλος.

Οδηγός φορμών



Τι τίτλο θέλετε για τις φόρμες;

Φόρμα:

Δευτερεύουσα φόρμα:

Αυτές είναι όλες οι πληροφορίες που χρειάζεται ο Οδηγός για να δημιουργήσει τη φόρμα.

Θέλετε να ανοίξει η φόρμα ή να τροποποιήσετε τη σχεδίασή της;

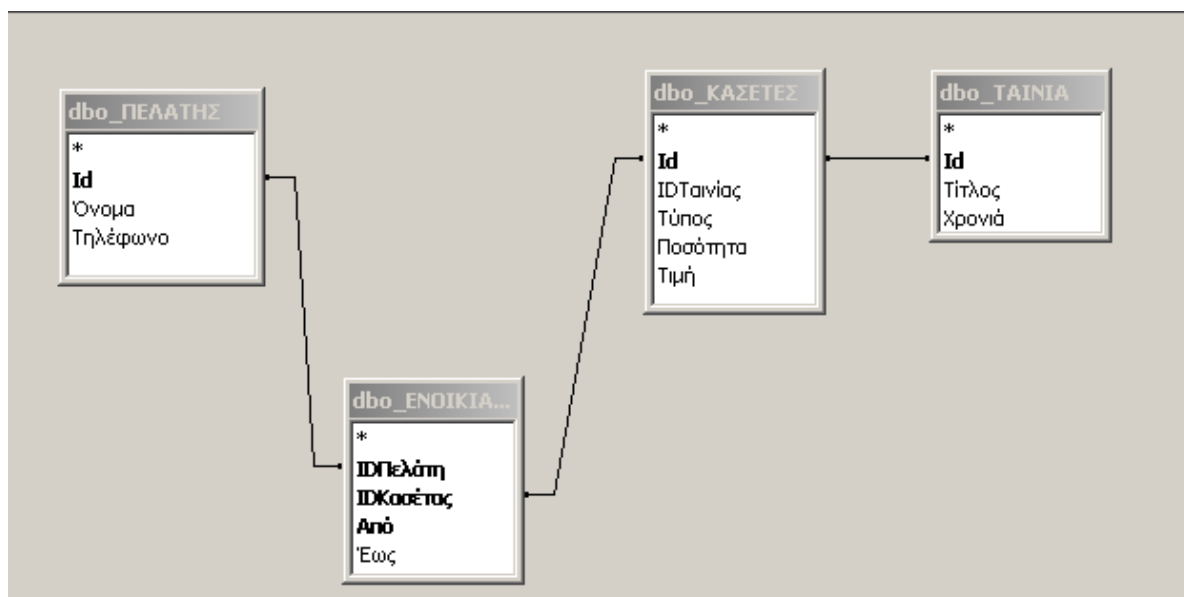
Ανοιγμα της φόρμας για την προβολή ή καταχώρηση πληροφοριών.

Τροποποίηση σχεδίασης της φόρμας.

Να εμφανιστεί βοήθεια για την εργασία με τη φόρμα;

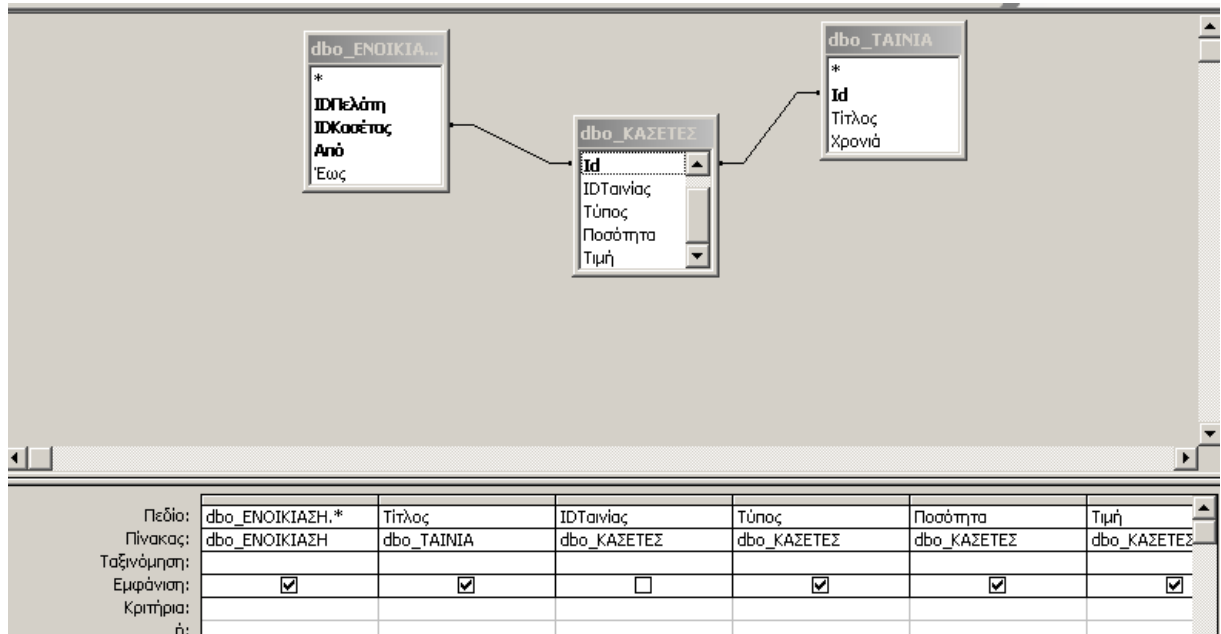
5.2.2.2. Δημιουργία φόρμας πελάτη-ενοικιαζόμενων κασετών με την χρήση ερωτήματος. (Αφορά πίνακες που σχετίζονται στο E-R με σχέσεις πολλά προς πολλά)

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



Για να φτιάξω μια κύρια φόρμα μαζί με μια δευτερεύουσα φόρμα στη σχέση πολλά προς πολλά όπου η κύρια να παίρνει τιμές από τον πίνακα dbo_ΠΕΛΑΤΗΣ, ενώ στη δευτερεύουσα να καταχωρώ τις κασέτες που ενοικίασε ένας πελάτης εργαζομαι ως εξής:

1. Χρησιμοποιώ τον ενδιάμεσο πίνακα dbo_ΕΝΟΙΚΙΑΣΗ που συνδέει τον πίνακα dbo_ΚΑΣΕΤΕΣ και τον πίνακα dbo_ΠΕΛΑΤΗΣ. Ο πίνακας ενοικίαση έχει ως πεδία τα πεδία κλειδιά των δύο πινάκων που συνδέει και όσα άλλα πεδία είναι απαραίτητα (π.χ. πεδίο Από, πεδίο Έως).
2. Δημιουργώ Ερώτημα που παίρνει τιμές από τον ενδιάμεσο πίνακα (dbo_ΕΝΟΙΚΙΑΣΗ), τον πίνακα (dbo_ΤΑΙΝΙΑ) και τον πίνακα (dbo_ΚΑΣΕΤΕΣ). Στο ερώτημα βάζω i) τα πεδία του ενδιάμεσου πίνακα, ii) από τον πίνακα dbo_ΚΑΣΕΤΕΣ βάζω τα πεδία του που θέλω να εμφανίζονται στην δευτερεύουσα φόρμα εκτός από το πεδίο κλειδί και iii) από τον πίνακα dbo_ΤΑΙΝΙΑ το πεδίο Τίτλος. Το ονομάζω ΕρΕνοικίαση_Κασέτες



Παρακάτω βλέπουμε τα αποτελέσματα όταν τρέχουμε το ερώτημα μας.

IDΠελάτη	IDΚασέτας	Από	Έως	Τίτλος	Τύπος	Ποσότητα	Τιμή
1	1	2006-07-10	2006-09-10	Rear Window	VHS	3	2
1	2	2006-09-20	2006-11-20	Psycho	DVD	1	3
2	1	2006-09-10		Rear Window	VHS	3	2

Στόχος μας είναι να δημιουργήσουμε μια κύρια φόρμα Πελάτη και δευτερεύουσα που να εισάγουμε τις ταινίες που νοικιάζει. Δηλαδή την φόρμα όπως εμφανίζεται στην παρακάτω Εικόνα.

IDΚασέτας	Τίτλος	Από	Έως
1	Rear Window	2006-07-10	2006-09-10
2	Rear Window	2006-09-20	2006-11-20

Δημιουργώ Φόρμα Κύρια Δευτερεύουσα βάζοντας σαν Κύριο πίνακα τον πίνακα από όπου πρέπει να παίρνει τιμές η Κύρια φόρμα (dbo_ΠΕΛΑΤΗΣ). Η Δευτερεύουσα φόρμα θα παίρνει τιμές από το ερώτημα που περιέχει τις Κασέτες δηλ το ΕρΕνοικίαση_Κασέτες.

Δημιουργία φόρμα με τη χρήση οδηγού
Επιλέγουμε τον πίνακα dbo_ΠΕΛΑΤΗΣ και παίρνουμε όλα τα πεδία του

Οδηγός φορμών

Ποια πεδία θέλετε στη φόρμα;
Έχετε τη δυνατότητα να επιλέξετε από περισσότερους από έναν πίνακες ή ερωτήματα.

Πίνακες/Ερωτήματα
Πίνακας: dbo_ΠΕΛΑΤΗΣ

Διαθέσιμα πεδία:

Επιλεγμένα πεδία:

	>	Id
	>>	Όνομα
	<	Τηλέφωνο
	<<	

Άκυρο < Προηγούμενο Επόμενο > Τέλος

Στη συνέχεια και χωρίς να πατήσουμε Επόμενο επιλέγουμε το ερώτημα ΕρΕνοικιασηΚασέτες και παίρνουμε όλα τα πεδία του εκτός από το IDΠελάτη. Τέλος στο ίδιο παράθυρο επιλέγουμε τον πίνακα dbo_ΤΑΙΝΙΑ και παίρνουμε το πεδίο Τίτλος.

Οδηγός φορμών

Ποια πεδία θέλετε στη φόρμα;
Έχετε τη δυνατότητα να επιλέξετε από περισσότερους από έναν πίνακες ή ερωτήματα.

Πίνακες/Ερωτήματα
Ερώτημα: ΕρΕνοικιαση_Κασέτες

Διαθέσιμα πεδία:

Επιλεγμένα πεδία:

IDΠελάτη	>	Id
IDΤαινίας	>>	Όνομα
Τύπος	<	Τηλέφωνο
Τιμή	<<	IDΚασέτας
Χρονιά		Τίτλος
		Από
		Εως

Άκυρο < Προηγούμενο Επόμενο > Τέλος

Οδηγός φορμών

Ποια πεδία θέλετε στη φόρμα;

Έχετε τη δυνατότητα να επιλέξετε από περισσότερους από έναν πίνακες ή ερωτήματα.

Πίνακες/Ερωτήματα
 Πίνακας: dbo_TAINIA

Διαθέσιμα πεδία:

Id
Χρονιά

Επιλεγμένα πεδία:

Id
Όνομα
Τηλέφωνο
IDΚασέτας
Τίτλος
Από
Έως

Άκυρο < Προηγούμενο Επόμενο > Τέλος

Πατάμε Επόμενο

Οδηγός φορμών

Πώς θέλετε να προβάλλονται τα δεδομένα;

κατά dbo_ΠΕΛΑΤΗΣ
 κατά dbo_ΕΝΟΙΚΙΑΣΗ
 κατά dbo_TAINIA

Id, Όνομα, Τηλέφωνο

IDΚασέτας, Τίτλος, Από, Έως

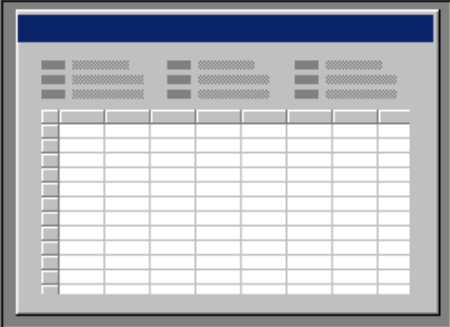
Φόρμα με δευτερεύουσες φόρμες Συνδεδεμένες φόρμες

Άκυρο < Προηγούμενο Επόμενο > Τέλος

Επόμενο..

Οδηγός μορμών

Τι διάταξη θέλετε να έχει η δευτερεύουσα φόρμα;



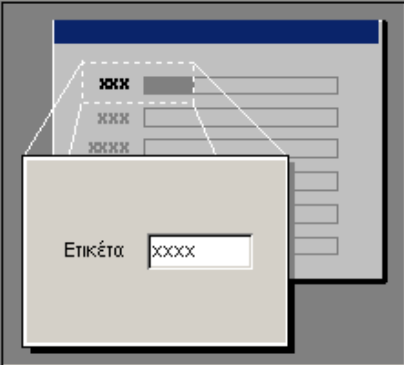
- Πίνακα
- Φύλλου δεδομένων
- Συγκεντρωτικός Πίνακας
- Συγκεντρωτικό Γράφημα

Άκυρο < Προηγούμενο Επόμενο > Τέλος

Επόμενο

Οδηγός μορμών

Ποιο στυλ προτιμάτε;



- Αμμόλιθος
- Αποστολή
- Βιομηχανικό
- Διεθνής
- Ζωγραφική Sumi
- Λίθος
- Μίλιμετρέ
- Πολυχρωμία
- Ριζόχαρτο
- Τυπικό**

Άκυρο < Προηγούμενο Επόμενο > Τέλος

Επόμενο

Οδηγός φορμών

Τι τίτλο θέλετε για τις φόρμες;

Φόρμα:

Δευτερεύουσα φόρμα:

Αυτές είναι όλες οι πληροφορίες που χρειάζεται ο Οδηγός για να δημιουργήσει τη φόρμα.

Θέλετε να ανοίξει η φόρμα ή να τροποποιήσετε τη σχεδίασή της;

Ανοιγμα της φόρμας για την προβολή ή καταχώρηση πληροφοριών.

Τροποποίηση σχεδίασης της φόρμας.

Να εμφανιστεί βοήθεια για την εργασία με τη φόρμα;

Ακυρο < Προηγούμενο Επόμενο > Τέλος

Στο σημείο αυτό και επειδή στο τέλος θα έχουμε πολλές κύριες – δευτερεύουσες φόρμες προκειμένου να ξέρουμε ποια αντιστοιχεί σε ποια, μετονομάζουμε την κύρια και την δευτερεύουσα σε φρμ_Πελάτης_κύρια_1. Αντίστοιχα μετονομάζουμε την δευτερεύουσα φόρμα σε φρμ_ΕρΕνοικίαση_Δευτερεύουσα_1 και πατάμε τέλος.

dbo_ΠΕΛΑΤΗΣ1

Id: 1
Όνομα: Perkins
Τηλέφωνο: 246801
dbo_ΕΝΟΙΚΙΑΣΗ

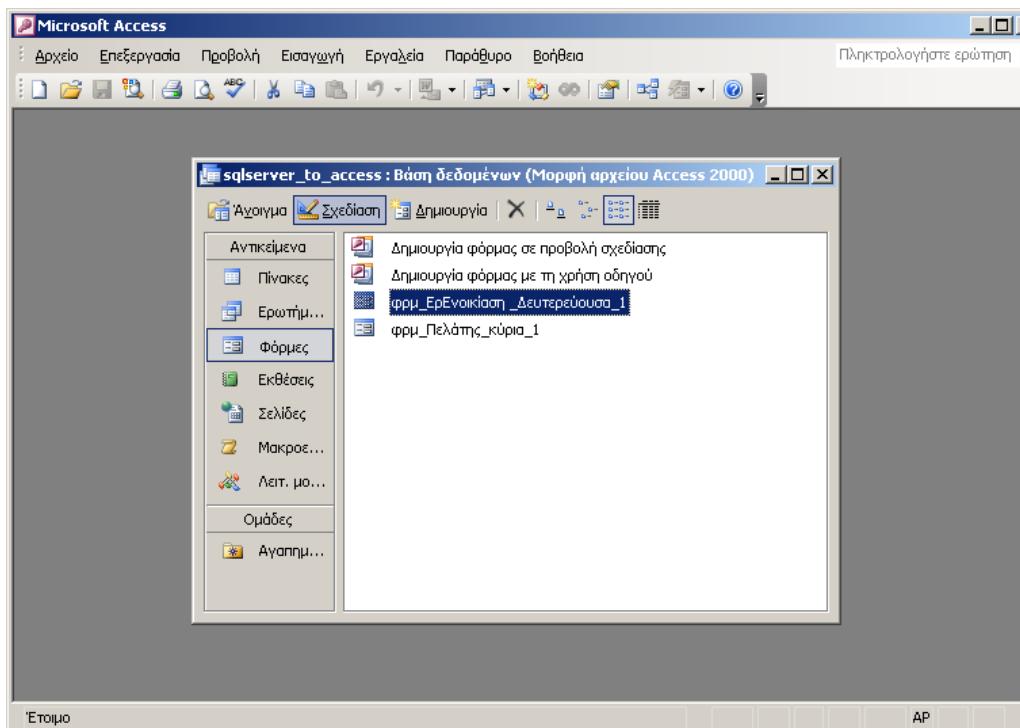
IDΚασέτας	Τίτλος	Από	Έως
1	Rear Window	2006-07-10	2006-09-10
2	Rear Window	2006-09-20	2006-11-20
*			

Εγγραφή: 1 από 2
Εγγραφή: 1 από 3

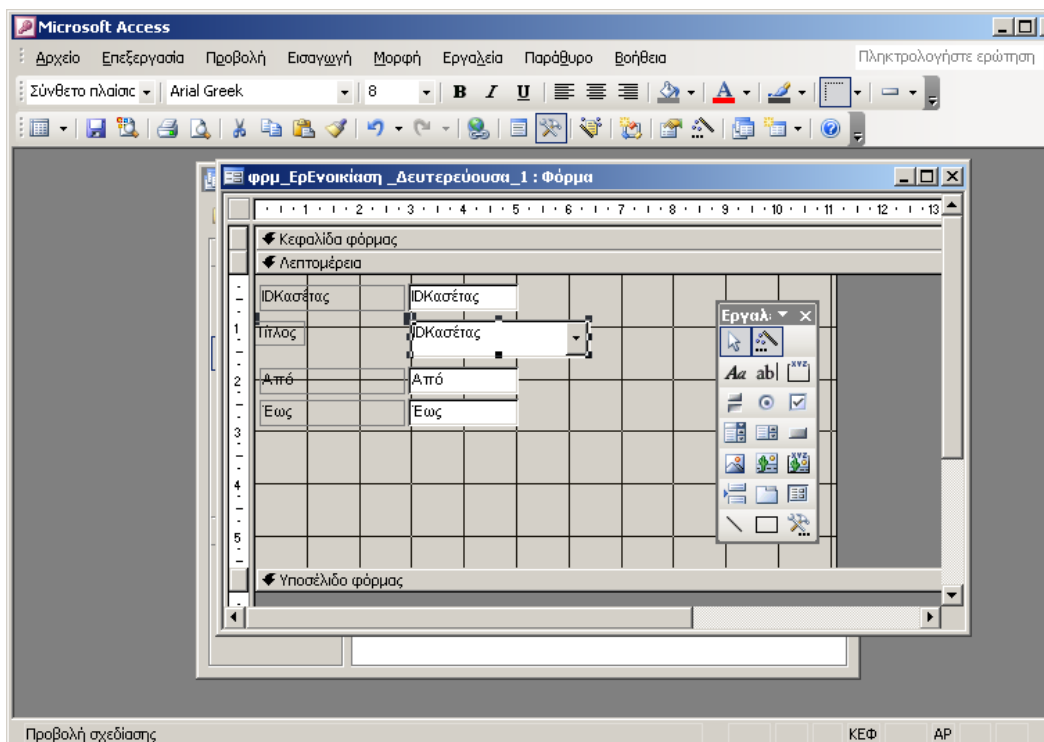
5.3 Δημιουργία Λίστας Αναζήτησης σε φόρμα

Σε αυτή την ενότητα θα παρουσιάσουμε τα 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 ονόματος τους και όχι του ID τους.

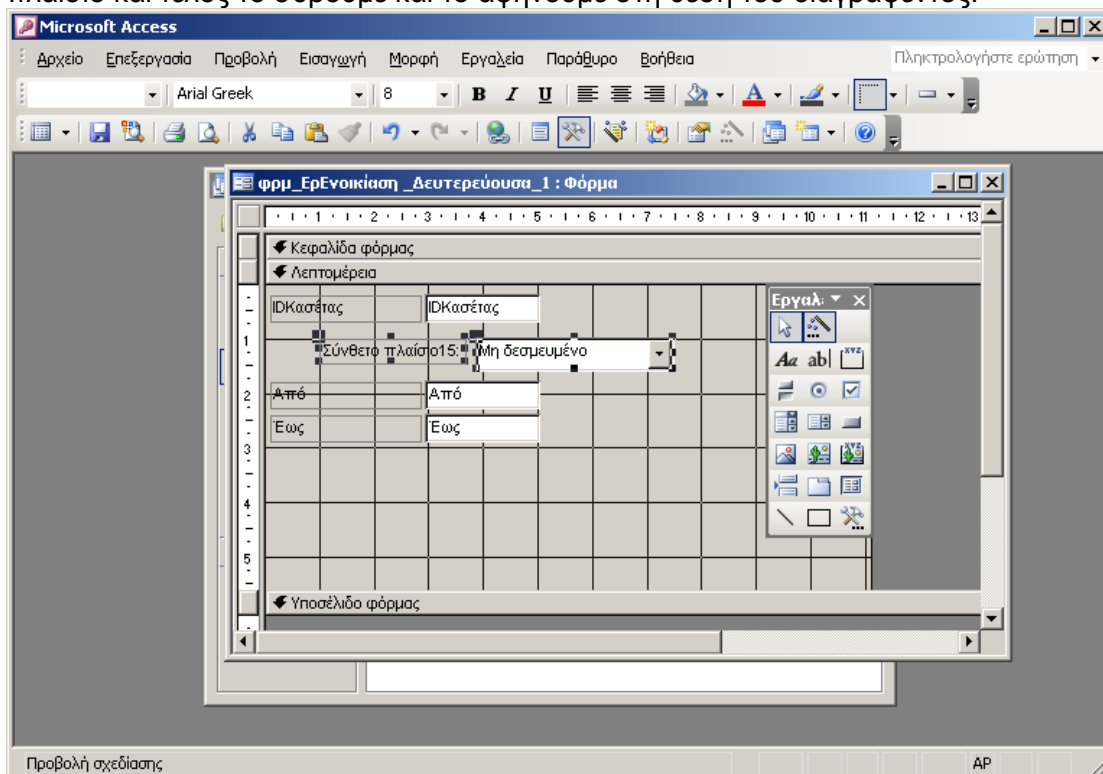
Βήμα 1: Επιλεγούμε την φόρμα με το όνομα : φρμ_ΕρΕνοικίαση_Δευτερεύουσα_1. Στην συνέχεια, επιλέγουμε σχεδίαση.



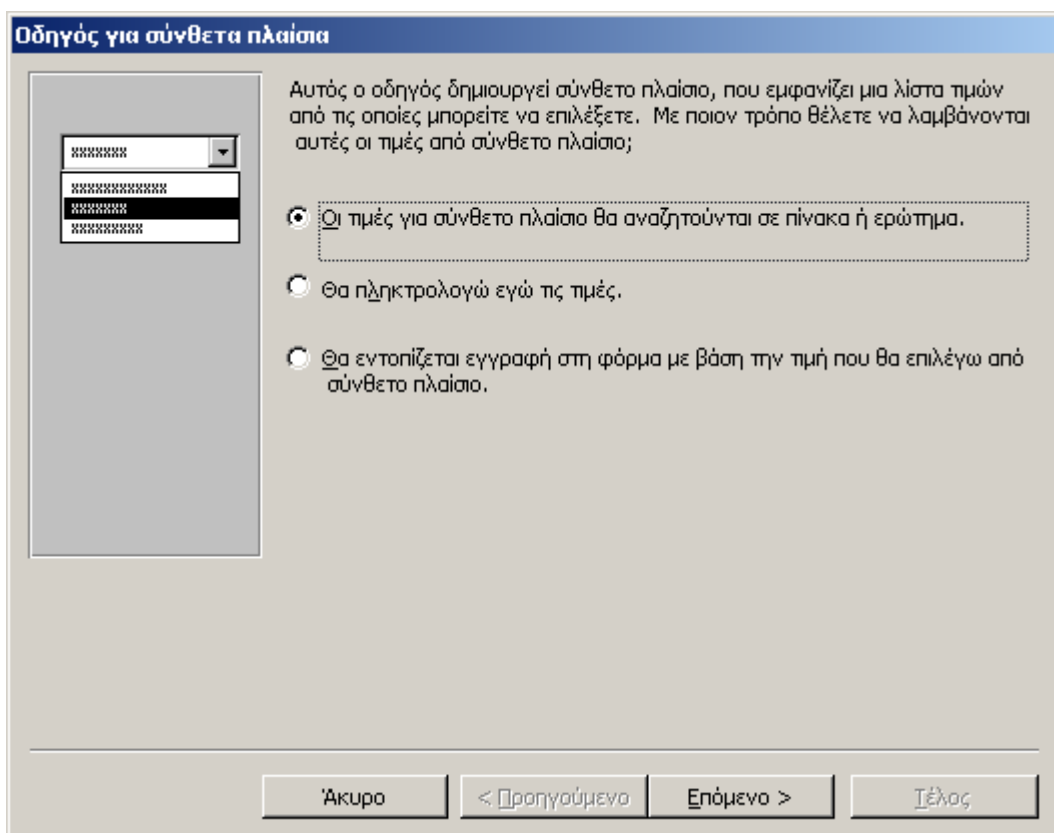
Βήμα 2: Επιλέγουμε την ετικέτα και το πεδίο Τίτλος και τα διαγράφουμε.



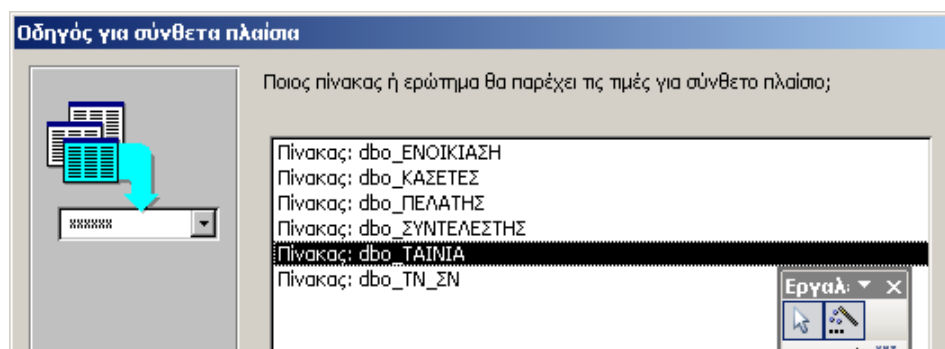
Βήμα 3 : Αφού διεγράφη ο Τίτλος, προσθέτουμε απ'τα Εργαλεία το εικονίδιο σύνθετο πλαίσιο και τέλος το σύρουμε και το αφήνουμε στη θέση του διαγραφέντος.



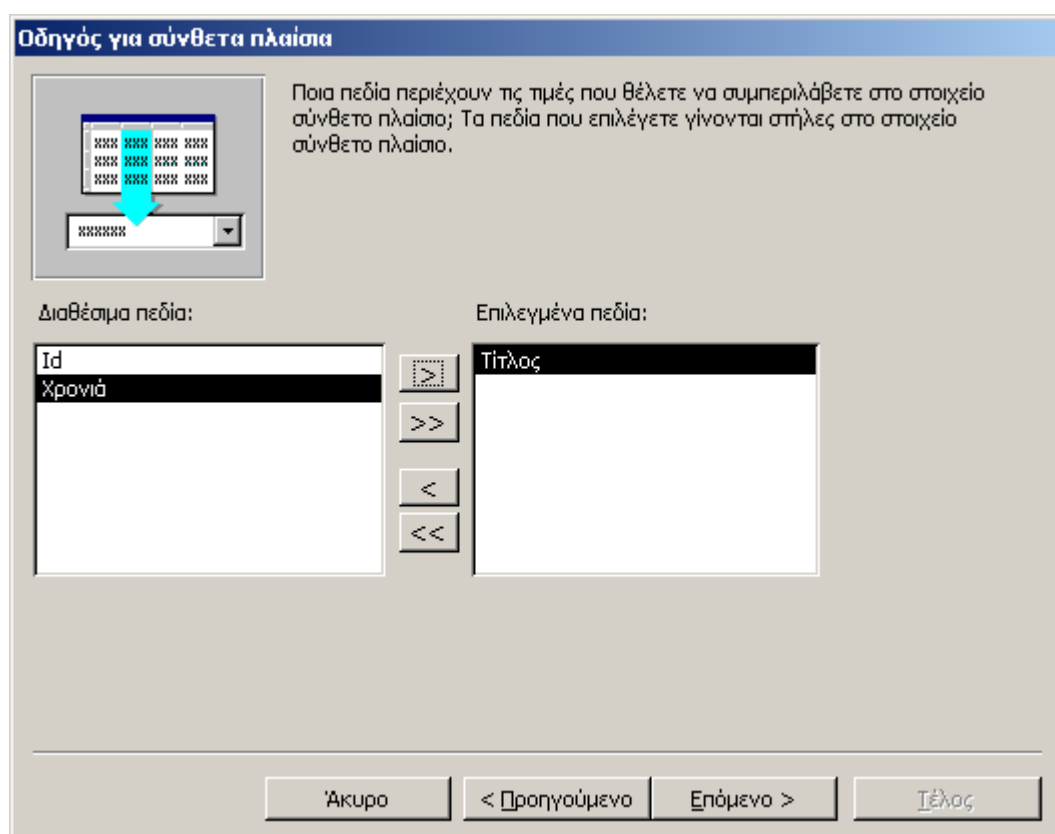
Βήμα 4: Πατάμε Επόμενο.



Βήμα 5 : Επιλέγουμε το πίνακα ΤΑΙΝΙΑ και πατάμε Επόμενο.

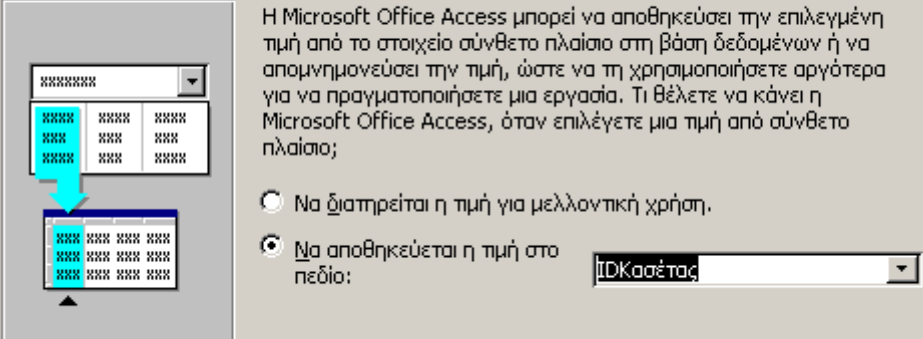


Βήμα 6 : Από τα διαθέσιμα πεδία επιλεγούμε Τίτλος και το μεταφέρουμε στα επιλεγμένα πεδία με το σύμβολο > και πατάμε Επόμενο.



Βήμα 7 : Και πατάμε Επόμενο μέχρι να φτάσουμε σε αυτό το σημείο και ακολουθούμε τις παρακάτω επιλογές.

Οδηγός για σύνθετα πλαίσια



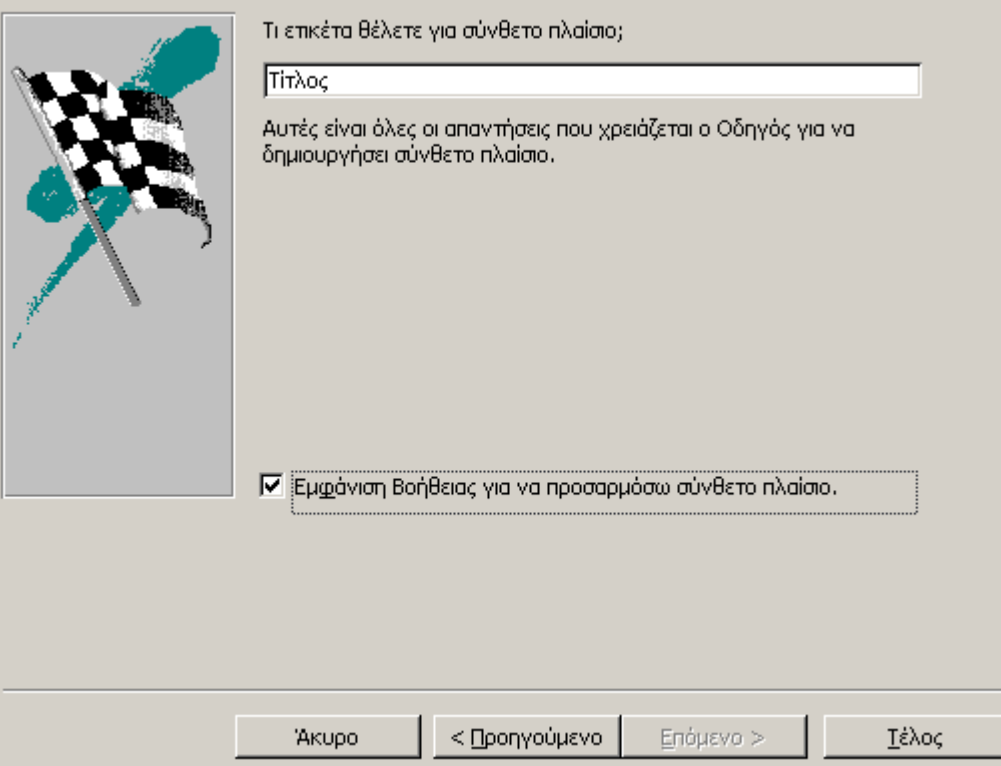
Η Microsoft Office Access μπορεί να αποθηκεύσει την επιλεγμένη τιμή από το στοιχείο σύνθετο πλαίσιο στη βάση δεδομένων ή να απομνημονεύσει την τιμή, ώστε να τη χρησιμοποιήσετε αργότερα για να πραγματοποιήσετε μια εργασία. Τι θέλετε να κάνει η Microsoft Office Access, όταν επιλέγετε μια τιμή από σύνθετο πλαίσιο;

Να διατηρείται η τιμή για μελλοντική χρήση.

Να αποθηκεύεται η τιμή στο πεδίο:

Βήμα 8 : Πατάμε Επόμενο και Τέλος.

Οδηγός για σύνθετα πλαίσια



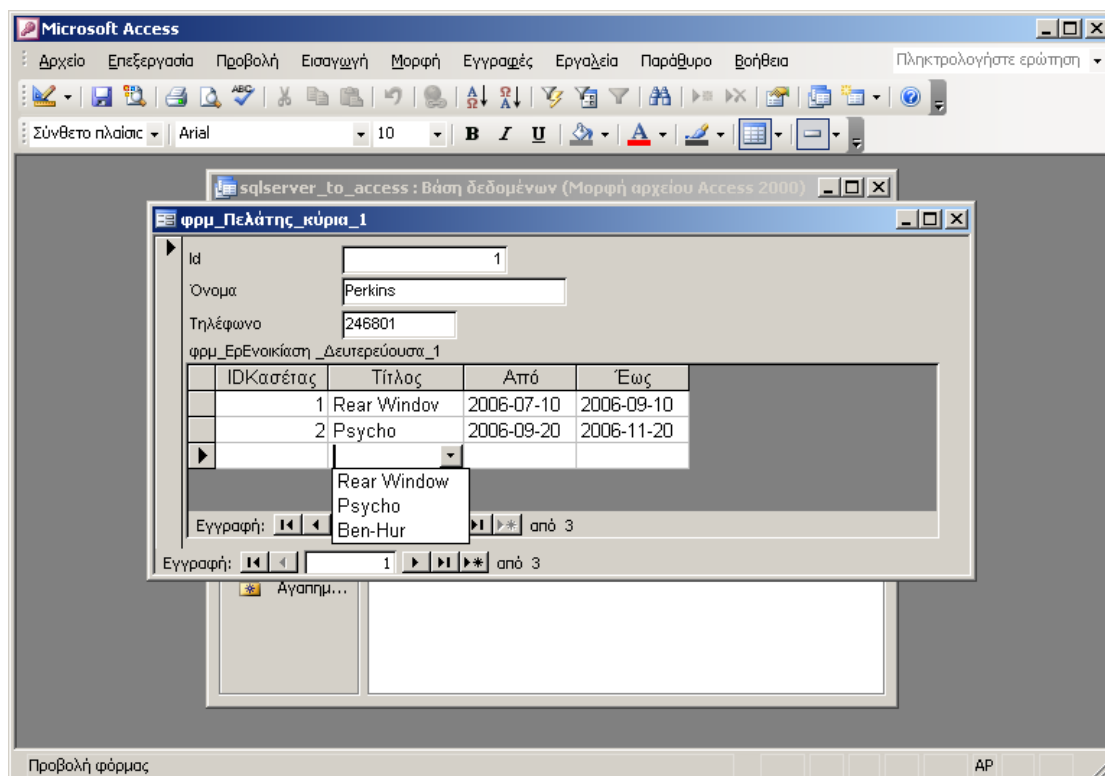
Τι επκέτα θέλετε για σύνθετο πλαίσιο;

Αυτές είναι όλες οι απαντήσεις που χρειάζεται ο Οδηγός για να δημιουργήσει σύνθετο πλαίσιο.

Εμφάνιση Βοήθειας για να προσαρμόσω σύνθετο πλαίσιο.

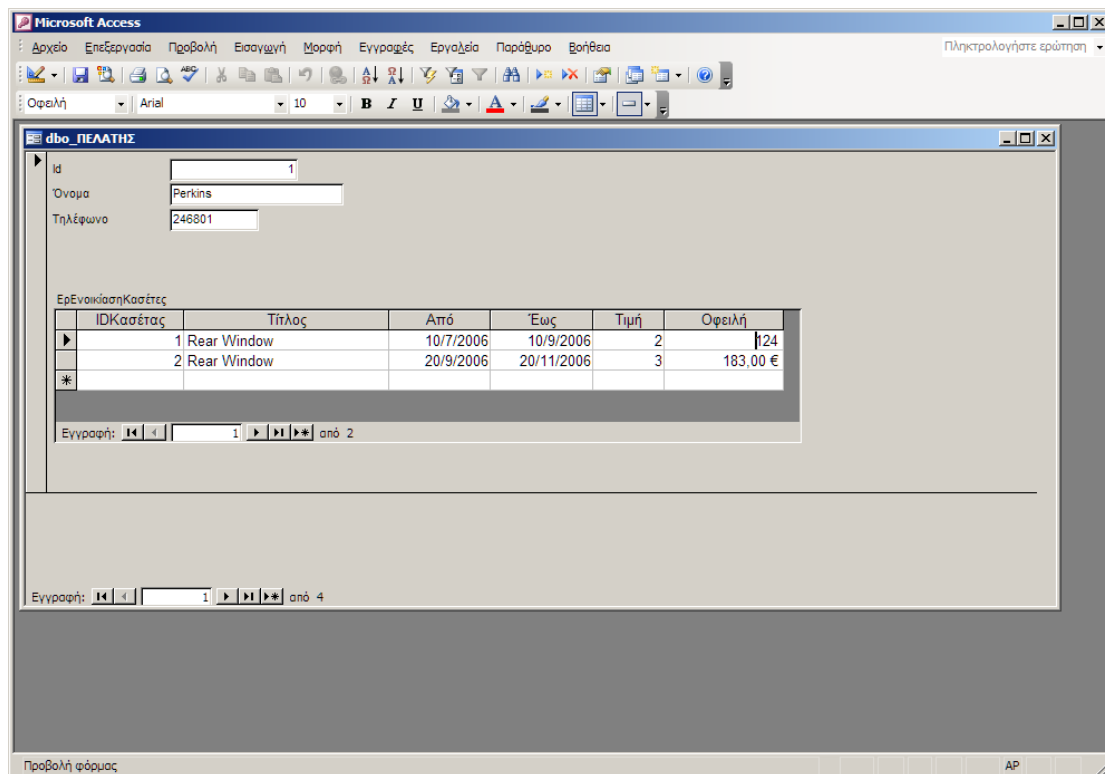
Άκυρο < Προηγούμενο Επόμενο > Τέλος

Βήμα 9 : Πάμε στην κύρια φόρμα και μας εμφανίζεται η παρακάτω εικόνα, όπου πλέον μπορούμε να αναζητήσουμε μια ταινία με το όνομα της χωρίς να πρέπει να γνωρίζουμε τον κωδικό της.



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 το ποσό σε ευρώ με το οποίο θα χρεώνεται ο πελάτης όταν επιστρέφει μία κασέτα, όπως εμφανίζεται στην παρακάτω Εικόνα.



Το υπολογιζόμενο πεδίο Οφειλή, προκύπτει από τον παρακάτω τύπο:

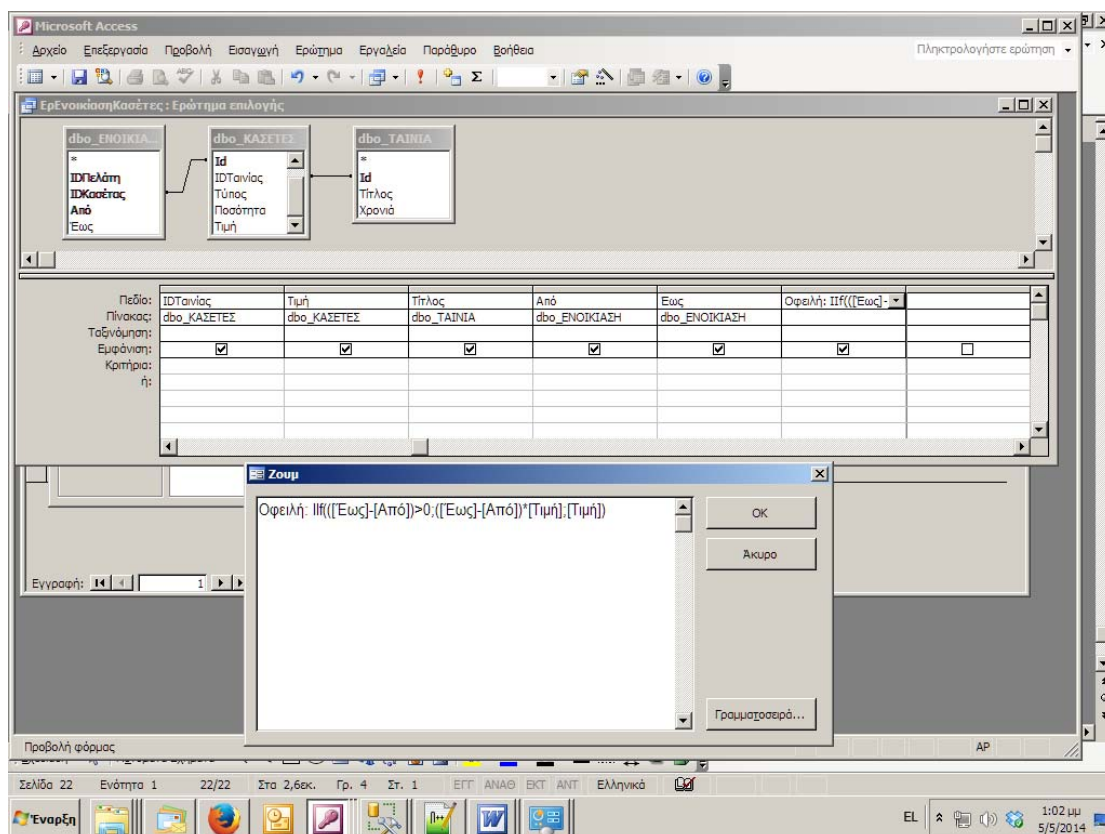
Οφειλή: $([Έως]-[Από])*[Τιμή]$

Σύμφωνα με τον παραπάνω τύπο, προκειμένου να βρούμε την χρέωση ενός πελάτη για μία κασέτα 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 της κασέτας.

Για την περίπτωση που ένας πελάτης επιστρέφει την κασέτα αυθημερόν θα πρέπει να διαμορφώσουμε τον παραπάνω τύπο ως εξής:

Οφειλή: $\text{If}(([Έως]-[Από])>0;([Έως]-[Από])*[Τιμή];[Τιμή])$

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



Ασκήσεις

1. Δημιουργήστε μια Κύρια/Δευτερεύουσα φόρμα, όπου στην Κύρια θα φαίνεται η Κασέτα και στην δευτερεύουσα φόρμα θα μπορεί κάποιος να καταχωρεί τον πελάτη που ενοικιάζει/επιστρέφει την κασέτα.
2. Δημιουργήστε μία λίστα αναζήτησης στην παραπάνω Δευτερεύουσα φόρμα για το πεδίο όνομα πελάτη.
3. Δημιουργήστε ένα υπολογιζόμενο πεδίο στο οποίο θα υπολογίζεται στην τελική οφειλή ενός πελάτη και φόρος ΦΠΑ 23%. Το νέο υπολογιζόμενο πεδίο θα ονομάζεται «Οφειλή με ΦΠΑ μαζί»