

Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Σχολή Θετικών Επιστημών
Τμήμα Πληροφορικής

ΕΡΓΑΣΤΗΡΙΟ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

Νανόπουλος Αλέξανδρος

Θεσσαλονίκη 2007

Πίνακας Περιεχομένων

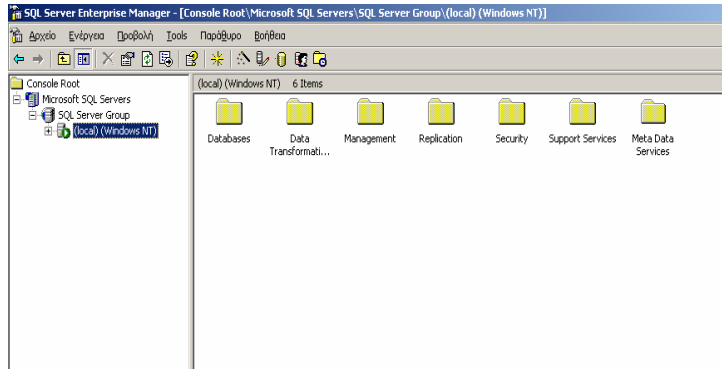
1. ΒΑΣΙΚΕΣ ΛΕΙΤΟΥΡΓΙΕΣ ΚΑΙ ΕΡΓΑΛΕΙΑ	2
1.1. Εκκίνηση του SQL Enterprise Manager	2
1.2. SQL Query Analyzer	3
2. ΔΗΜΙΟΥΡΓΙΑ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ	5
2.1. Δημιουργία νέας βάσης δεδομένων με τον wizard	6
2.2. Δημιουργία νέας βάσης με τον Enterprise Manager	8
2.3. Δημιουργία νέας βάσης δεδομένων με την SQL	10
3. ΔΗΜΙΟΥΡΓΙΑ ΠΙΝΑΚΩΝ.....	11
3.1. Τύποι Δεδομένων	11
3.2. Δημιουργία πινάκων με τον Enterprise Manager	12
3.3. Δημιουργία πινάκων με την SQL	14
3.4. Δημιουργία πινάκων με τον Database Designer	16
4. ΕΙΣΑΓΩΓΗ ΤΙΜΩΝ ΣΤΟΥΣ ΠΙΝΑΚΕΣ.....	23
4.1. Εισαγωγή τιμών στους πίνακες με τον Enterprise Manager	23
4.2. Εισαγωγή τιμών στους πίνακες με την SQL.....	24
5. ΒΑΣΙΚΑ ΕΡΩΤΗΜΑΤΑ ΣΕ TRANSACT-SQL	26
5.1. Διαχείριση του Results Pane	26
5.2. Εντολές επιλογής από έναν πίνακα	27
5.3. Ταξινόμηση αποτελεσμάτων	30
6. ΣΥΝΔΕΣΗ ΠΙΝΑΚΩΝ.....	32
6.1. Εσωτερική, Εξωτερική Σύνδεση	32
6.2. Μετονομασία και Αυτο-σύνδεση	34
7. ΣΥΝΘΕΤΑ ΕΡΩΤΗΜΑΤΑ: ΟΜΑΔΟΠΟΙΗΣΗ	36
7.1. Συναρτήσεις ομαδοποίησης.....	36
7.2. Ο όρος Group By.....	37
7.3. Ο όρος Having.....	39
8. ΣΥΝΘΕΤΑ ΕΡΩΤΗΜΑΤΑ: ΠΡΑΞΕΙΣ ΣΥΝΟΛΩΝ	41
8.1. Ένωση συνόλων	41
8.2. Εμφωλιασμένες εντολές	42
6.5.1. Έλεγχος περιεκτικότητας σε σύνολο	42
6.5.1. Σύγκριση συνόλων.....	43
6.5.2. Έλεγχος ύπαρξης και Διαφορά συνόλων	44
9. ΟΨΕΙΣ	45
10. ΜΕΤΑΒΟΛΗ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ.....	46
10.1. Εισαγωγή πολλών γραμμών σε πίνακα	47
10.2. Ενημέρωση.....	47
10.3. Διαγραφή.....	48
10.4. Μεταβολή σχήματος πίνακα.....	48
10.5. Μετονομασία.....	49
10.6. Κατάργηση πίνακα και βάσης.....	49
11. ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΕΡΩΤΗΜΑΤΩΝ.....	50
11.1. Παρακολούθηση Πλάνου Εκτέλεσης.....	50
11.2. Κατάλογοι	51
12. SCRIPTING ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ	53

1. ΒΑΣΙΚΕΣ ΛΕΙΤΟΥΡΓΙΕΣ ΚΑΙ ΕΡΓΑΛΕΙΑ

1.1. Εκκίνηση του SQL Enterprise Manager

Ο SQL Enterprise Manager αποτελεί το πρωτεύων διαχειριστικό εργαλείο του SQL Server 2000. Η εκκίνησή του γίνεται ως εξής:

Start->Programs->Microsoft SQL Server->Enterprise Manager.



Κάνοντας κλικ στον επιθυμητό εγγεγραμμένο server (στην περίπτωση μας είναι ο local), εμφανίζονται τα αντικείμενα από τα οποία αποτελείται ο server, που μας δίνουν την δυνατότητα ελέγχου και διαχείρισης όλων των λειτουργιών του.

Τα διάφορα αντικείμενα εμφανίζονται τόσο στο δένδρο με τους φακέλους και τους υποφακέλους στο αριστερό μέρος του Enterprise Manager, όσο και στο δεξί του με την μορφή εικονιδίων.

Το κύριο αντικείμενο είναι το εξής:

- **Databases:** περιέχει όλες τις βάσεις δεδομένων του SQL Server και τα αντικείμενα τους.

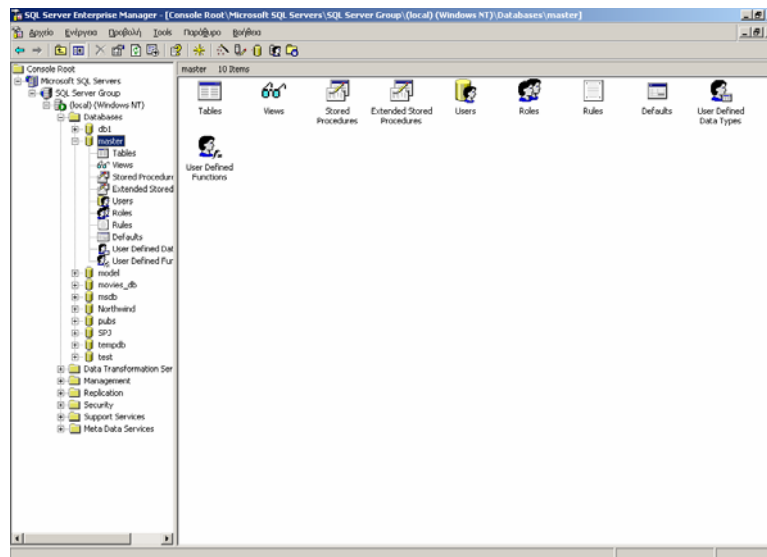
Κατά την εγκατάσταση ο SQL Server δημιουργεί κάποιες βάσεις δεδομένων, μεταξύ των οποίων είναι και οι εξής:

Βάση δεδομένων	Περιέχει
Master	Πληροφορίες για την λειτουργία του SQL Server, όπως λογαριασμούς χρηστών, άλλους SQL servers, μεταβλητές περιβάλλοντος, μηνύματα λάθους, βάσεις δεδομένων, χώρος δίσκου που κατανέμεται στις βάσεις κ.λ.π.
Northwind	Μια απλή βάση δεδομένων για να μάθετε τον SQL Server.
pubs	Μια απλή βάση δεδομένων για να μάθετε τον SQL Server.

Σε αυτές τις βάσεις καθώς και σε κάθε νέα που δημιουργείται από τον χρήστη υπάρχουν διάφορα αντικείμενα :

- Πίνακες
- Views
- Stored Procedures
- Extended Stored Procedures
- Χρήστες
- Ρόλοι
- Κανόνες
- Defaults
- Τύποι δεδομένων ορισμένοι από τον χρήστη
- Συναρτήσεις ορισμένες από τον χρήστη
- Full-text κατάλογοι

Enterprise Manager δίνει τη δυνατότητα είτε κάνοντας δεξί κλικ πάνω σε αυτά τα αντικείμενα είτε επιλέγοντας από το μενού να δημιουργήσει νέα, και μεταβάλλει τις ιδιότητες τους.





1.2. SQL Query Analyzer

Ο SQL Query Analyzer είναι ένα εργαλείο με γραφικό περιβάλλον (GUI) που σας επιτρέπει τον σχεδιασμό, έλεγχο, και την εκτέλεση Transact-SQL δηλώσεων, stored procedures, batches, και scripts αλληλεπιδραστικά. Μπορείτε να τρέξετε τον SQL Query Analyzer μέσα από τον Enterprise Manager (είτε κατευθείαν από το Start menu → All programs → Microsoft SQL Server → Query Analyzer). Εάν είστε συνδεδεμένοι σε έναν server, ο Query Analyzer θα συνδεθεί σε αυτόν τον server, ενώ συγχρόνως αν έχετε επιλέξει μια συγκεκριμένη βάση δεδομένων, ο Query Analyzer θα επιλέξει αυτήν την βάση δεδομένων.

Ο Editor pane είναι ένα text-editing παράθυρο που χρησιμοποιείται για την εισαγωγή και την εκτέλεση Transact-SQL δηλώσεων. Ο κώδικας που εισέρχεται μέσα στον Editor pane χρωματίζεται με βάση την κατηγορία στην οποία ανήκει. Ο πίνακας που ακολουθεί περικλείει τα defaults χρώματα και το τι σημαίνουν:

Χρώμα	Κατηγορία
Κόκκινο	String χαρακτήρων
Βαθύ κόκκινο	Stored procedure
Πράσινο	πίνακας συστήματος
Βαθύ πράσινο	σχόλιο
Πορφυρό	μεταβλητή συστήματος
Μπλε	δεσμευμένη λέξη
Γκρι	operator

Μπορείτε είτε να εκτελέσετε ολόκληρα script ή μόνο επιλεγμένες SQL δηλώσεις στο SQL Query Analyzer:

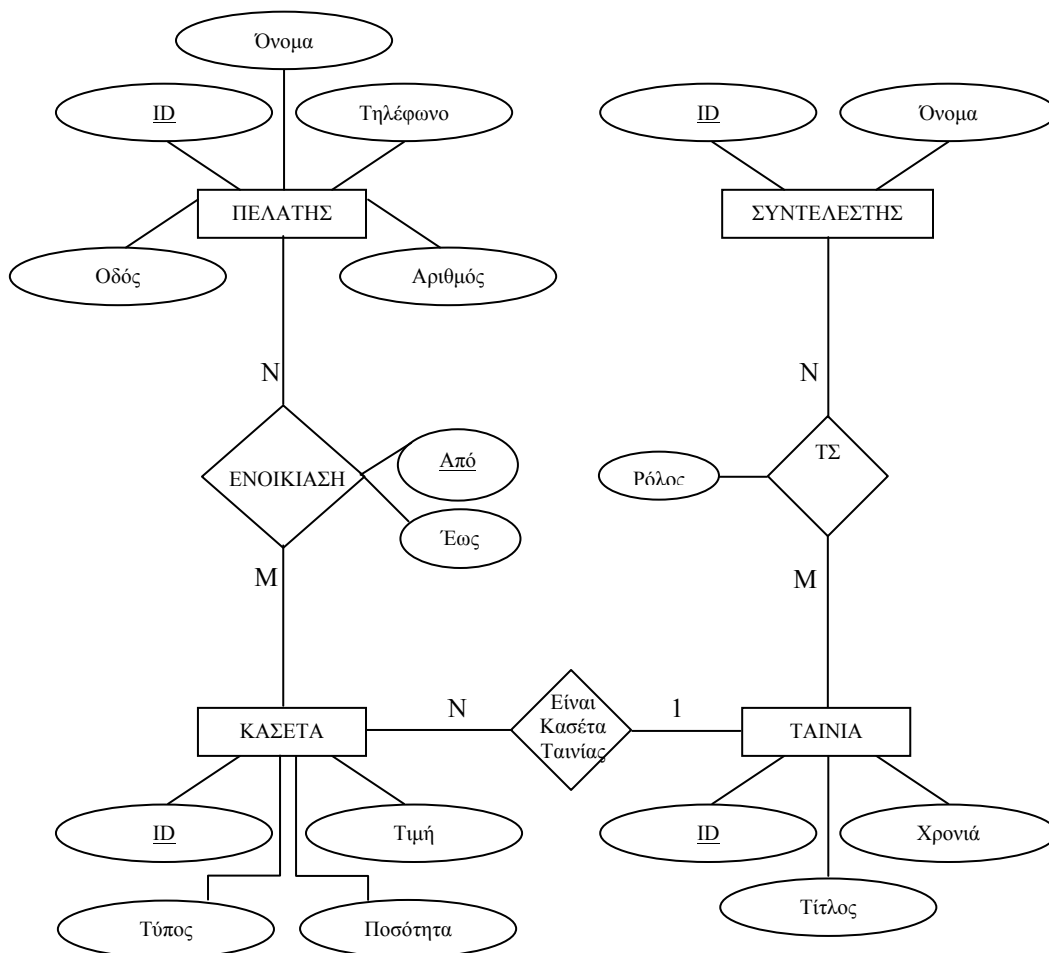
- Εκτελείτε ένα ολοκληρωμένο script δημιουργώντας το ή ανοίγοντας το, στον Editor pane και πατώντας το κουμπί F5 ή το κουμπί  στην γραμμή εργαλείων.
- Εκτελείτε μόνο επιλεγμένες SQL δηλώσεις, τονίζοντας τις γραμμές του κώδικα στον Editor pane και πατώντας το κουμπί F5 ή το κουμπί  στην γραμμή εργαλείων.

2. ΔΗΜΙΟΥΡΓΙΑ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

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

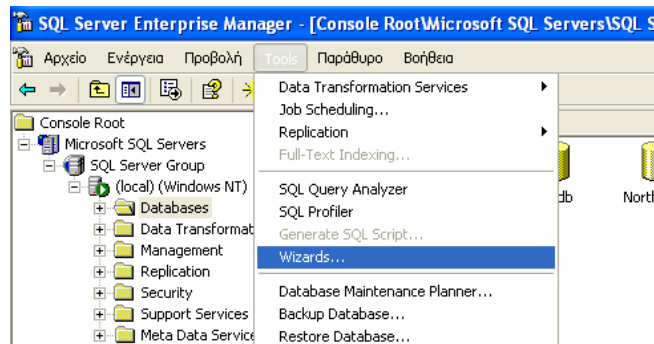
- Wizard δημιουργίας βάσης δεδομένων του Enterprise Manager
- SQL Server Enterprise Manager
- Transact-SQL εντολές

Στη συνέχεια θα χρησιμοποιήσουμε και τους τρεις τρόπους, για να δημιουργήσουμε τη βάση δεδομένων **videoclub**, το διάγραμμα οντοτήτων της οποίας δίνεται στο ακόλουθο σχήμα:

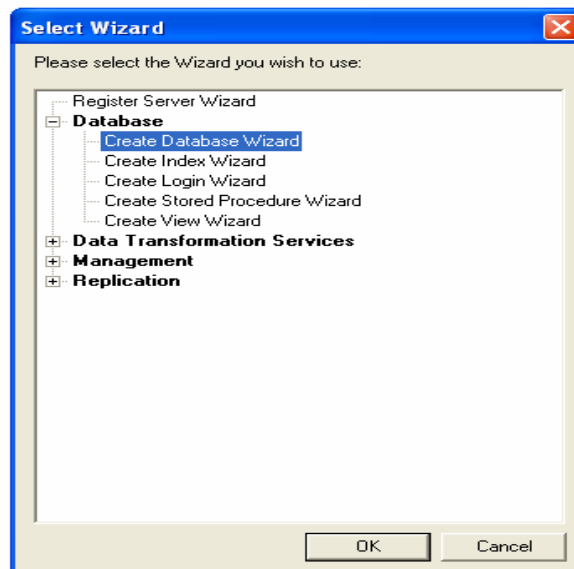


2.1. Δημιουργία νέας βάσης δεδομένων με τον wizard

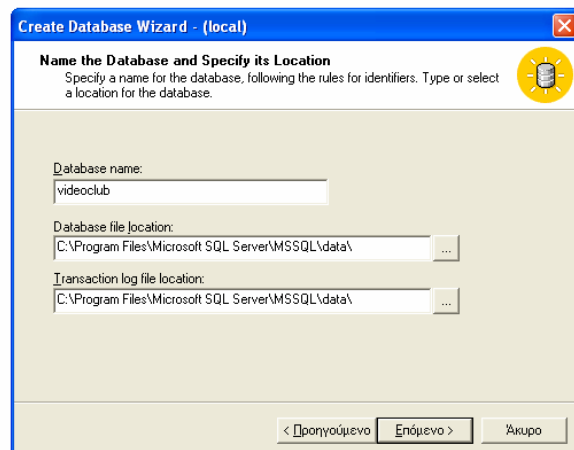
Επιλογή “Tools”→“Wizards”



Ανοίγουμε την επιλογή “Database” (κλικ στο σταυρό) και κλικ στο “Create Database Wizard” και “OK”.



Ακολουθούμε τις οδηγίες του Wizard. Πρώτα κλικ στο “Next”. Στη συνέχεια, δίνουμε το όνομα της βάσης.

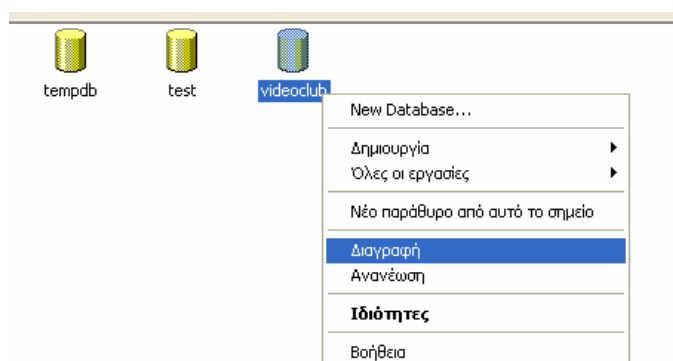


Σε όλα τα επόμενα παράθυρα, δεχόμαστε τις επιλογές πατώντας “Next”, μέχρι να τερματίσει ο Wizard, οπότε και πατάμε το “Finish”.

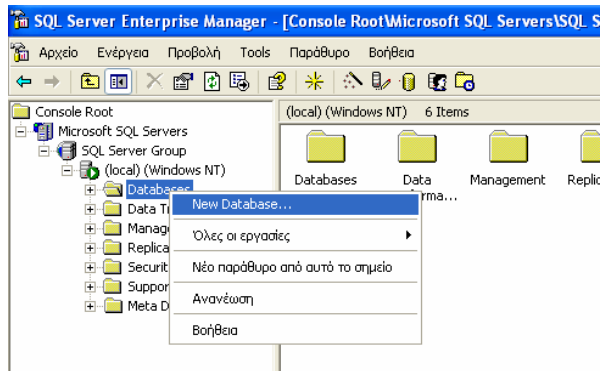


Ακόμα ένα message box εμφανίζεται, ρωτώντας αν θα θέλατε να δημιουργήσετε ένα πλάνο συντήρησης. Η δημιουργία ενός πλάνου συντήρησης συνιστάται για να διασφαλίζει πως η βάση λειτουργεί καλά, έχει γίνει σωστό backup σε περίπτωση βλάβης, και ελέγχει για ασυνέπειες. Κάντε κλικ στο No για να τελειώσετε.

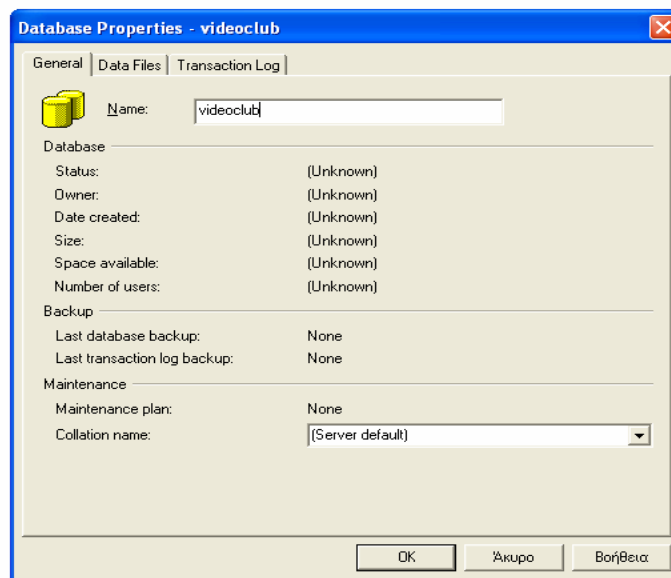
Διαγραφή της βάσης: Δεξί κλικ στο εικονίδιο της βάσης (κύλινδρος) και “Delete”.



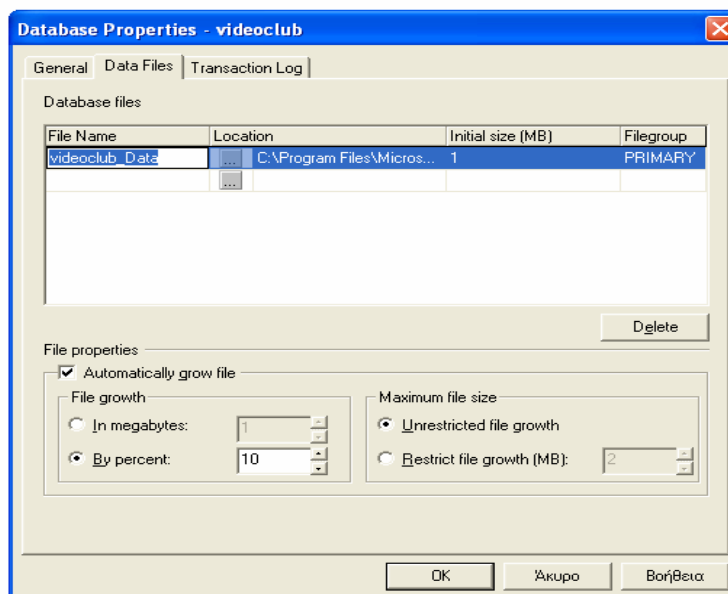
2.2. Δημιουργία νέας βάσης με τον Enterprise Manager



Δεξί κλικ στο “Databases”, επιλογή “New Database...”, δίνουμε το όνομα



Κλικ στο tab Data “Files”:



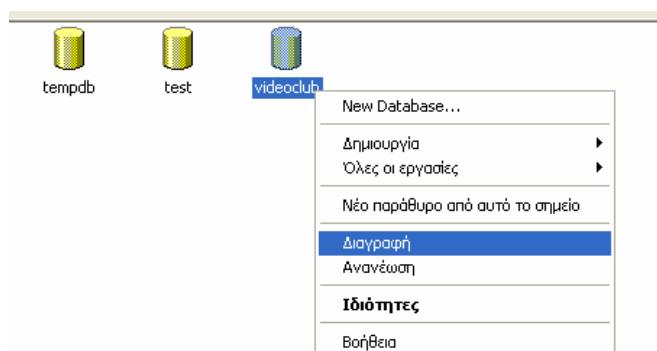
Το πρωτεύον αρχείο δημιουργείται αυτόματα και του δίνεται το όνομα της βάσης δεδομένων σας ως πρόθεμα. Το πρωτεύον αρχείο δεδομένων έχει .mdf επέκταση. Εάν έχετε ιδέα για το πόσο μεγάλη θα είναι τα αρχεία της βάσης δεδομένων σας, πληκτρολογήστε μια τιμή αρχικού μεγέθους. Διαφορετικά, αφήστε το προκαθορισμένο μέγεθος (μπορείτε να το μεταβάλλετε αργότερα). Οποιαδήποτε άλλα αρχεία δημιουργείτε επιπρόσθετα στο πρωτεύον αρχείο, θα είναι δευτερεύοντα αρχεία και θα τους δοθούν αυτομάτως η επέκταση .ndf. Όλα τα αρχεία που θα δημιουργηθούν εδώ θα τοποθετηθούν στο πρωτεύον filegroup.

Ο SQL Server μπορεί αυτόματα να αυξάνει το μέγεθος της βάσης δεδομένων όταν χρειαστεί, που σας βοηθά την επιβάρυνση της συντήρησης. Γενικά, προτείνεται να επιλέξετε το χαρακτηριστικό της αυτόματης αύξησης του αρχείου (Automatically Grow The Database Files) διότι απαιτεί μικρή επιβάρυνση στην επίδοση. Διαφορετικά, θα χρειαστεί να ρυθμίσετε χειροκίνητα το μέγεθος της βάσης δεδομένων: είτε με καθορισμένα blocks MB είτε με ποσοστό του τρέχοντος μεγέθους. Θυμηθείτε πως το αρχείο της βάσης θα μεγαλώσει μόνο όταν χρειάζεται. Μπορείτε επίσης να περιορίσετε την βάση δεδομένων σε ένα maximum μέγεθος ή να την αφήσετε να μεγαλώνει χωρίς περιορισμό.

Κλικ στο tab Data “Transaction Log”:

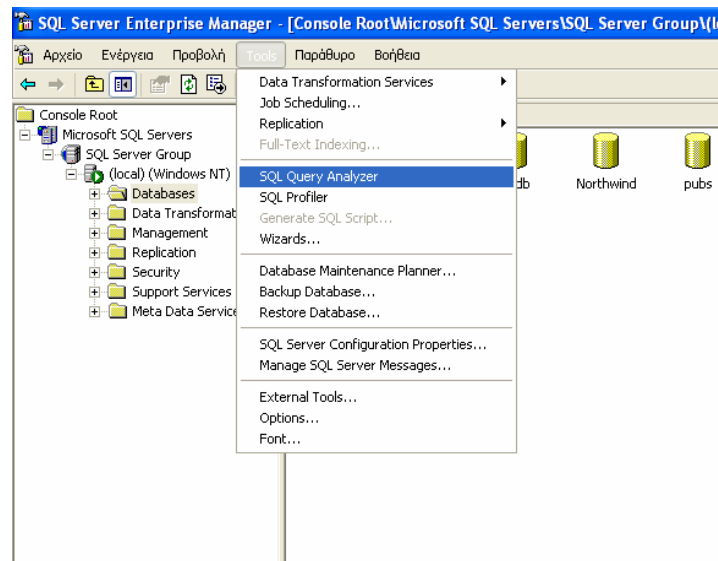
Όπως κάνατε για τα αρχεία της βάσης δεδομένων σας, πληκτρολογήστε το όνομα και το αρχικό μέγεθος για το transaction log αρχείο (θυμηθείτε πως το transaction log περιέχει μια εγγραφή με όλες τις μεταβολές της βάσης δεδομένων για δυνατότητα ανάνηψης σε περίπτωση βλάβης του συστήματος). Το πρώτο transaction log αρχείο δημιουργείται αυτόματα και παίρνει το όνομα της βάσης για πρόθεμα. Μπορείτε να αποδεχθείτε το όνομα ή να πληκτρολογήσετε ένα διαφορετικό. Τα transaction log δεδομένα αποθηκεύονται σε ένα αρχείο με επέκταση .ldf. Μπορείτε να προσθέσετε επιπλέον αρχεία αν χρειάζεται. Αν έχετε ιδέα για το μέγεθος που θα έχει το transaction log αρχείο, πληκτρολογήστε μια τιμή. Διαφορετικά κρατήστε το προκαθορισμένο μέγεθος.

Διαγραφή της βάσης: Δεξί κλικ στο εικονίδιο της βάσης (κύλινδρος) και “Delete”



2.3. Δημιουργία νέας βάσης δεδομένων με την SQL

Ανοίγουμε το εργαλείο “SQL Query Analyzer...”



Εντός του editor, γράφουμε την εντολή:
create database videoclub;

Στη συνέχεια πατάμε το πράσινο play κουμπί στην μπάρα εργαλείων, είτε το F5.

(Σημείωση: αν δώσουμε την εντολή **drop database** videoclub, γίνεται διαγραφή της βάσης. ΔΕΝ τη δίνουμε, ώστε να συνεχίσουμε στη δημιουργία πινάκων.)

3. ΔΗΜΙΟΥΡΓΙΑ ΠΙΝΑΚΩΝ

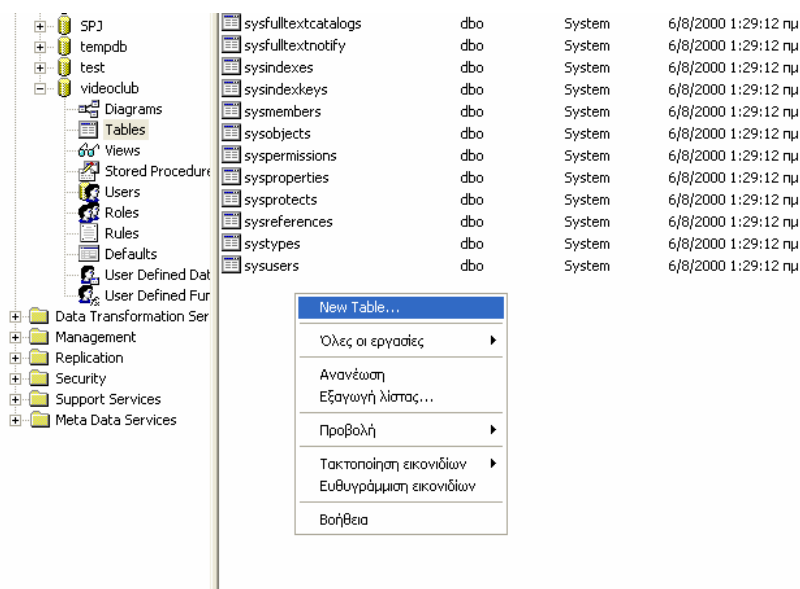
3.1. Τύποι Δεδομένων

Στον SQL Server, κάθε στήλη σχετίζεται με ένα τύπο δεδομένων, ο οποίος αποτελεί ένα χαρακτηριστικό που προσδιορίζει το είδος των δεδομένων (integer, character, monetary κ.λπ) που μπορεί να κρατήσει το αντικείμενο. Ο ακόλουθος πίνακας παρέχει τις περιγραφές των βασικών κατηγοριών των τύπων δεδομένων που υποστηρίζει ο SQL Server και τις περιγραφές των βασικών τύπων δεδομένων που περιέχει κάθε κατηγορία:

Κατηγορία	Περιγραφή	Τύπος δεδομένων	Περιγραφή
Character	Τα δεδομένα χαρακτήρων αποτελούνται από οποιοδήποτε συνδυασμό γραμμάτων, συμβόλων, και αριθμητικών δεδομένων. Για παράδειγμα, έγκυρα δεδομένα χαρακτήρων είναι οι συνδυασμοί "John123" και "(0\$%b99"	Char	Τα δεδομένα έχουν το ίδιο σταθερό μήκος(μέχρι 8KB)
		varchar	Τα δεδομένα μπορούν να ποικίλουν στον αριθμό των χαρακτήρων, αλλά το μήκος δεν μπορεί να υπερβαίνει τα 8KB
Unicode	Περιλαμβάνει όλους τους χαρακτήρες που ορίζονται στα διάφορα σύνολα χαρακτήρων. Οι Unicode τύποι δεδομένων καταλαμβάνουν διπλάσιο χώρο αποθήκευσης συγκριτικά με τους μη Unicode τύπους δεδομένων.	nchar	Τα δεδομένα πρέπει να έχουν το ίδιο σταθερό μήκος(μέχρι 4000 Unicode χαρακτήρες)
		nvarchar	Τα δεδομένα μπορούν να ποικίλουν στον αριθμό των Unicode χαρακτήρων τους (μέχρι 4000)
Date and time	Τα δεδομένα ημερομηνίας και ώρας αποτελούνται από έγκυρα δεδομένα από συνδυασμούς ημερομηνίας και ώρας. Δεν υπάρχουν ξεχωριστοί τύποι δεδομένων είτε μόνο για ημερομηνίες είτε μόνο για ώρες.	datetime	Τα δεδομένα κυμαίνονται από την 1 ^η Ιανουαρίου 1753 μέχρι την 31 ^η Δεκεμβρίου 9999.
		smalldatetime	Τα δεδομένα κυμαίνονται από την 1 ^η Ιανουαρίου 1900 μέχρι την 6 ^η Ιουνίου 2079
Integer	Αποτελούνται από αρνητικούς και θετικούς αριθμούς όπως τους -15, 0, 5 και 2.507.	bigint	Τα δεδομένα είναι αριθμοί από το -2^{63} μέχρι το $2^{63}-1$.
		int	Τα δεδομένα είναι αριθμοί από το -2.147.483.648 μέχρι το 2.147.483.647
		smallint	Τα δεδομένα είναι αριθμοί από το -32.768 μέχρι το 32.767.
Floating point	Τα προσεγγιστικά αριθμητικά δεδομένα αποτελούνται από δεδομένα preserved με την ακρίβεια που προσφέρει το δυαδικό αριθμητικό σύστημα	float	Τα δεδομένα είναι floating-point αριθμοί από $-1,79E + 308$ μέχρι $1,79E + 308$.
		real	Τα δεδομένα είναι floating-point αριθμοί από $-3,40E + 308$ μέχρι $3,40E + 308$.

3.2. Δημιουργία πινάκων με τον Enterprise Manager

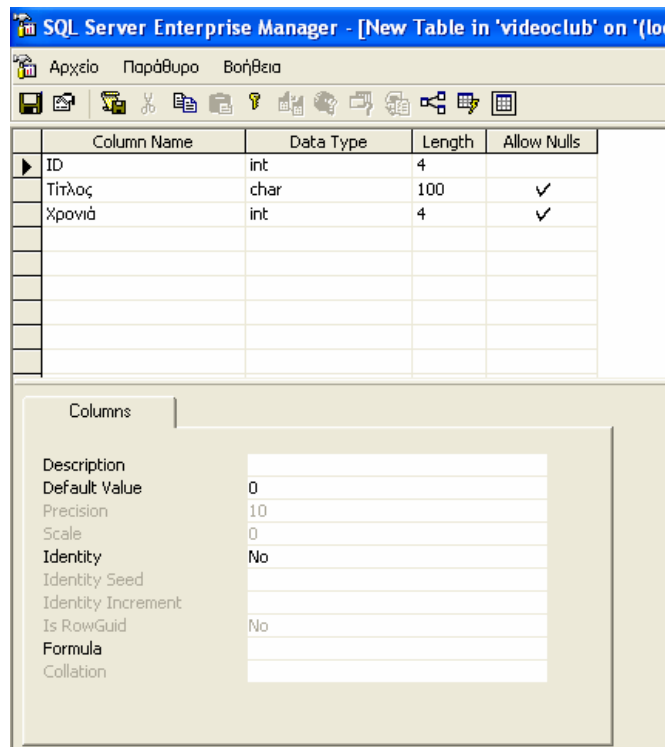
Στη δενδρική δομή, αριστερά, επεκτείνουμε τη βάση videoclub. Στη συνέχεια, επεκτείνουμε το εικονίδιο Tables. Βλέπουμε τους πίνακες που δημιουργεί ο SQL Server για κάθε νέα βάση. Δεξί κλικ στο δεξιό τμήμα και επιλέγουμε “New Table...”



Θα δημιουργήσουμε τον πίνακα TAINIA με τα ακόλουθα πεδία:

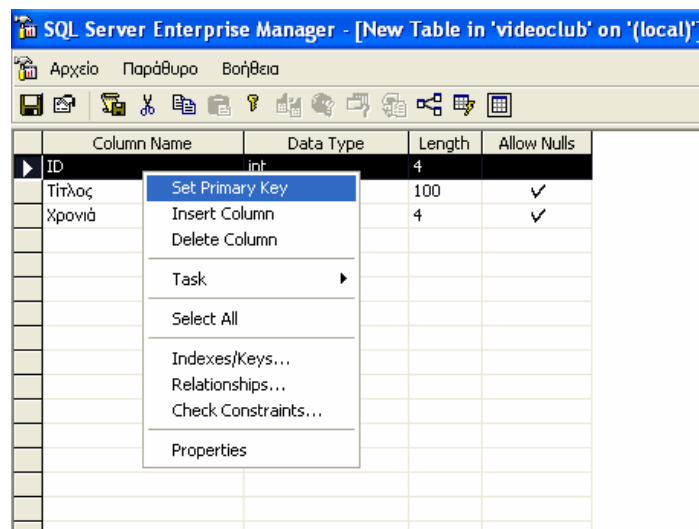
TAINIA			
Field	Type	Null	Key
ID	int(4)		PRI
Τίτλος	char(100)	YES	
Χρονιά	int(4)	YES	

Ανοίγει ο Table Designer, ένα γραφικό εργαλείο που σας επιτρέπει να σχεδιάσετε έναν πίνακα μέσα στη βάση δεδομένων στην οποία είστε συνδεδεμένοι.. Συμπληρώνουμε τα ακόλουθα:



Για την επιλογή τύπων δεδομένων, είτε πληκτρολογούμε το όνομά τους, είτε διαλέγουμε κάποιον από την drop-down λίστα. Προσοχή στον ορισμό του μήκους (στήλη Length). Η τιμή αυτή δίνεται σε bytes. Επίσης, δίνουμε προσοχή για κάθε πεδίο στο αν θα επιτρέψουμε NULL τιμές (το tick στην τελευταία στήλη) και στην Default Value (δεύτερη γραμμή στο κάτω τμήμα).

Στη συνέχεια, ορίζουμε το κύριο κλειδί. Δεξί κλικ στη γραμμή του πεδίου ID και επιλέγουμε “Set Primary Key”. Δίπλα στη γραμμή, εμφανίζεται ένα εικονίδιο που αναπαριστά ένα κίτρινο κλειδί.



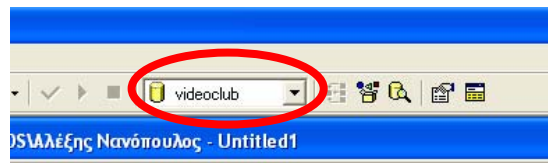
Από τη γραμμή εργαλείων (εικονίδιο με δισκέτα) σώζουμε τον πίνακα με όνομα TAINIA και κλείνουμε το παράθυρο δημιουργίας πίνακα (OXI τον enterprise manager). Στη λίστα με τους πίνακες της βάσης videoclub, μπορούμε να δούμε τον πίνακα που δημιουργήσαμε.

Επαναλάβετε το ίδιο για τη δημιουργία του πίνακα ΠΕΛΑΤΗΣ, με την ακόλουθη μορφή:

ΠΕΛΑΤΗΣ			
Field	Type	Null	Key
ID	int(4)		PRI
Όνομα	char(30)		
Τηλέφωνο	char(10)	YES	

3.3. Δημιουργία πινάκων με την SQL

Ανοίγουμε το εργαλείο SQL Query Analyzer. Προσέχουμε η επιλεγμένη βάση να είναι η videoclub.



Θα δημιουργήσουμε τον πίνακα ΣΥΝΤΕΛΕΣΤΗΣ, με την ακόλουθη μορφή:

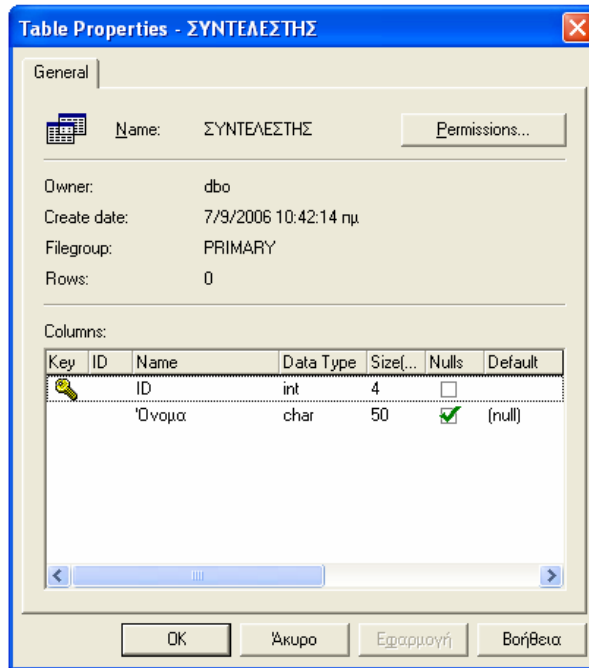
ΣΥΝΤΕΛΕΣΤΗΣ			
Field	Type	Null	Key
ID	int(4)		PRI
Όνομα	char(50)	YES	



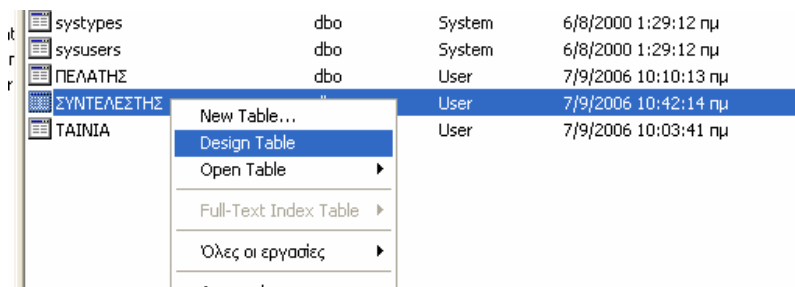
Στον editor γράφουμε την εξής εντολή:

```
CREATE TABLE ΣΥΝΤΕΛΕΣΤΗΣ (  
    ID int,  
    Όνομα char(50),  
    PRIMARY KEY (ID)  
)
```

Επιστρέφουμε στον Enterprise Manager. Στη λίστα με τους πίνακες, δεξί κλικ και “Refresh” (εναλλακτικά πατάμε F5). Μπορούμε να δούμε το νέο πίνακα που δημιουργήσαμε. Δεξί κλικ πάνω του και “Properties”. Βλέπουμε για τον πίνακα όσα εμφανίζονται στο ακόλουθο παράθυρο. Πατάμε OK.



Με δεξί κλικ στο όνομα του πίνακα στη λίστα πινάκων, μπορούμε να επιλέξουμε “Design Table”



Ανοίγει ένα παράθυρο σχεδιασμού του πίνακα, που είναι ίδιο με αυτά που χρησιμοποιήσαμε για τη δημιουργία του πίνακα. Μπορούμε να κάνουμε οποιαδήποτε τροποποίηση θέλουμε. Για την ώρα, ΔΕΝ μεταβάλλουμε κάποιο στοιχείο.

Στη συνέχεια, θα δημιουργήσουμε τον πίνακα ΚΑΣΕΤΑ. Επιστρέφουμε στο εργαλείο SQL Query Analyzer. Προσέχουμε η επιλεγμένη βάση να είναι η videoclub. Σβήνουμε από τον editor όλα τα περιεχόμενα. Ο πίνακας ΚΑΣΕΤΑ έχει την ακόλουθη μορφή:

ΚΑΣΕΤΑ

Field	Type	Null	Key
ID	int (4)		PRI
IDΤαινίας	int (4)		
Τύπος	char (4)	YES	
Ποσότητα	int (4)	YES	
Τιμή	int (4)	YES	



create table ΚΑΣΕΤΑ

```
(  
    ID int,  
    IDΤαινίας int NOT NULL,  
    Τύπος char(4),  
    Ποσότητα int,  
    Τιμή int,  
    PRIMARY KEY (ID),  
    FOREIGN KEY (IDΤαινίας) REFERENCES TAINIA(ID) on delete cascade  
)
```

Ο περιορισμός Ξένου Κλειδιού (FOREIGN KEY) καθορίζει συσχέτιση μεταξύ πινάκων. Ένα ξένο κλειδί σε έναν πίνακα, δείχνει σε ένα κύριο κλειδί σε έναν άλλο πίνακα. Οι περιορισμοί ξένου κλειδιού αποτρέπουν ενέργειες που αφήνουν «εκκρεμείς» γραμμές με τιμές ξένων κλειδιών σε κύρια κλειδιά που δεν υπάρχουν σε άλλους πίνακες (*ακεραιότητα αναφορών*). Στον πίνακα ΚΑΣΕΤΑ, ορίζουμε ότι το IDΤαινίας είναι ξένο κλειδί για το ID στον πίνακα ΤΑΙΝΙΑ. Κατ'αυτόν τον τρόπο, δεν θα μπορεί να υπάρξει τιμή για το IDΤαινίας, αν δεν υπάρχει και αντίστοιχη τιμή ID στον πίνακα ΤΑΙΝΙΑ. Η χρήση του ON DELETE ελέγχει την ενέργεια που θα πραγματοποιηθεί κατά την απόπειρα διαγραφής γραμμής στην οποία υπάρχει στήλη που δείχνεται από ξένο κλειδί σε άλλο πίνακα. Υπάρχουν οι εξής επιλογές:

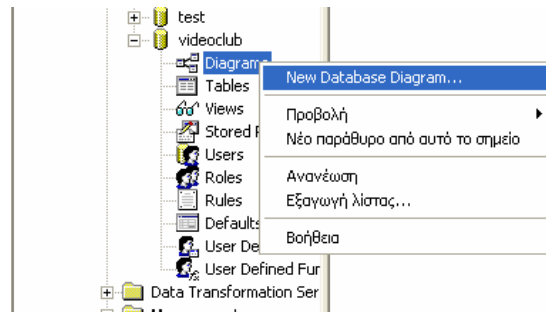
- NO ACTION: αποτρέπεται η διαγραφή και προβάλλεται μήνυμα λάθους.
- CASCADE: διαγράφει τη γραμμή και προκαλεί διαγραφή όλων των γραμμών σε άλλους πίνακες, οι οποίες περιέχουν ξένα κλειδιά προς την τιμή κλειδιού της διαγραμμένης στήλης.

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

3.4. Δημιουργία πινάκων με τον Database Designer

Ο Database Designer

Ο Database Designer είναι ένα γραφικό εργαλείο που σας επιτρέπει να σχεδιάσετε και να απεικονίσετε τη βάση δεδομένων στην οποία είστε συνδεδεμένοι. Για οποιαδήποτε βάση δεδομένων, μπορείτε να δημιουργήσετε όσα διαγράμματα επιθυμείτε. Κάθε διάγραμμα που δημιουργείτε αποθηκεύεται στην σχετιζόμενη βάση δεδομένων. Για να δημιουργήσετε ένα διάγραμμα από την βάση δεδομένων videoclub, επεκτείνετε την videoclub στο αριστερό pane του Enterprise Manager, και κάντε δεξί κλικ στο Diagrams. Επιλέξτε το New Database Diagram από το μενού επιλογών.



Εμφανίζεται το Create Database Diagram Wizard, όπως φαίνεται και στο παρακάτω σχήμα.



Κάντε κλικ στο Next, για να εμφανίσετε το παράθυρο διαλόγου Select Tables to be Added. Επιλέξτε τους πίνακες που θέλετε να συμπεριλαμβάνονται στο διάγραμμα από τη λίστα Available Tables, και κάντε κλικ στο Add. Εισάγουμε τους πίνακες: ΚΑΣΕΤΑ, ΠΕΛΑΤΗΣ, ΣΥΝΤΕΛΕΣΤΗΣ, ΤΑΙΝΙΑ. Πατάμε Next και Finish.

Θα δείτε το διάγραμμα της βάσης δεδομένων σας, όπως φαίνεται και στο παρακάτω σχήμα.

ΠΕΛΑΤΗΣ	
ID	
Όνομα	
Τηλέφωνο	

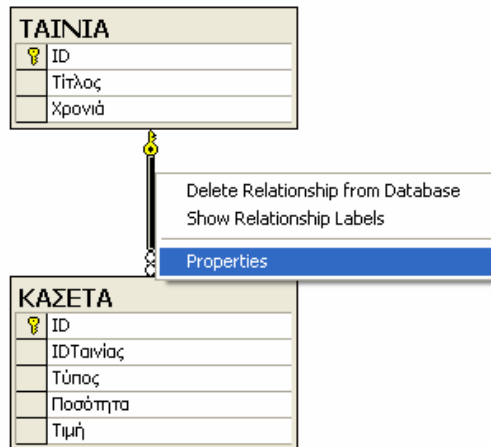
ΣΥΝΤΕΛΕΣΤΗΣ	
ID	
Όνομα	

ΚΑΣΕΤΑ	
ID	
IDΤαινίας	
Τύπος	
Ποσότητα	
Τιμή	

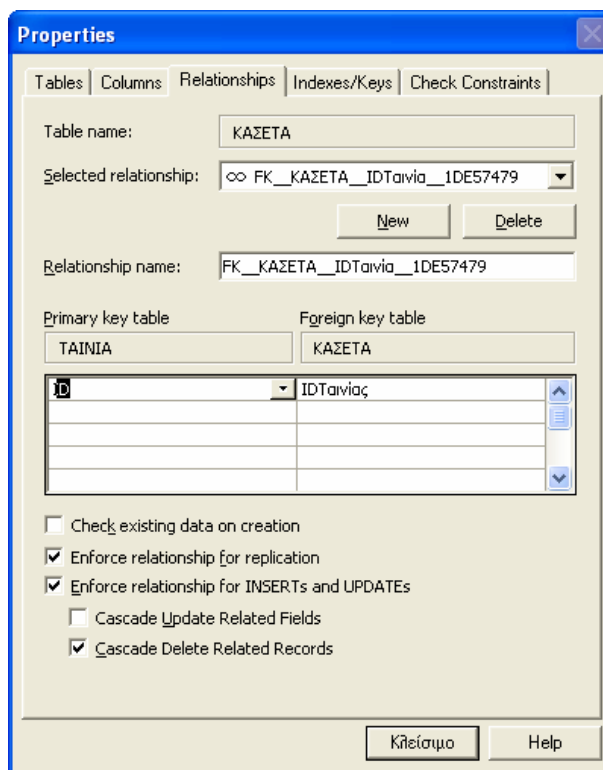
ΤΑΙΝΙΑ	
ID	
Τίτλος	
Χρονιά	



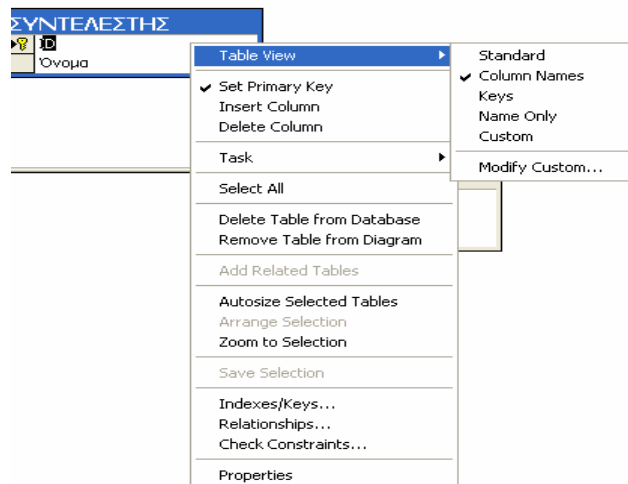
Σώζουμε το διάγραμμα πατώντας το εικονίδιο με τη δισκέτα, δίνοντας το όνομα VC_DIAGRAM. Στο διάγραμμα εμφανίζονται τα ονόματα των ιδιοτήτων κάθε πίνακα, τα κύρια κλειδιά σημειώνονται με το εικονίδιο του κίτρινου κλειδιού, ενώ οι σχέσεις ξένου κλειδιού αναπαριστώνται με τις γραμμές που συνδέουν τους πίνακες. Κάνουμε δεξί κλικ πάνω σε στη γραμμή μεταξύ ΤΑΙΝΙΑ και ΚΑΣΕΤΑ, και επιλέγουμε Properties.



Εμφανίζεται το ακόλουθο παράθυρο, στο οποίο φαίνεται ότι το ξένο κλειδί είναι το IDΤαινίας στον πίνακα ΚΑΣΕΤΑ, ενώ το κύριο κλειδί είναι το ID στον πίνακα ΤΑΙΝΙΑ. Επίσης φαίνεται η δήλωση για ON DELETE CASCADE. Αυτά είναι σύμφωνα με τα όσα δηλώσαμε στην εντολή δημιουργίας του πίνακα ΚΑΣΕΤΑ μέσω της SQL. Πατάμε κλείσιμο.



Στο διάγραμμα, μπορούμε να επιλέξουμε διάφορους τρόπους προβολής των πινάκων. Για παράδειγμα, κάνουμε δεξί κλικ πάνω στον πίνακα ΣΥΝΤΕΛΕΣΤΗΣ και επιλέγουμε το Table View. Εκεί υπάρχουν οι εναλλακτικές που φαίνονται στο σχήμα:



- Standard: εμφανίζονται όλες οι ιδιότητες των στηλών, όπως φαίνεται και στο παρακάτω σχήμα. Η προβολή αυτή είναι παρόμοια με το τον Table Designer, τον οποίο μελετήσαμε προηγουμένως, και κάθε πίνακας μπορεί να μεταβληθεί χρησιμοποιώντας τις ίδιες μεθόδους που χρησιμοποιήθηκαν στον Table Designer.
- Column Names: εμφανίζονται μόνο τα ονόματα των στηλών, καθώς και ενδείξεις για τα κύρια κλειδιά.
- Keys: εμφανίζονται μόνο οι στήλες που συμμετέχουν στο κύριο κλειδί.
- Name Only: εμφανίζεται μόνο το όνομα του ΠΙΝΑΚΑ.

Επιλέγουμε το είδος προβολής αναλόγως της πολυπλοκότητας του διαγράμματος (π.χ., σε ένα περίπλοκο διάγραμμα με πολλούς πίνακες, η Name Only επιλογή είναι βολική). Στο παράδειγμά μας, επιλέγουμε την Column Names.


Εισαγωγή ενός νέου πίνακα στο διάγραμμα




Θα δημιουργήσουμε τον πίνακα ΕΝΟΙΚΙΑΣΗ, που έχει την ακόλουθη μορφή

ΕΝΟΙΚΙΑΣΗ			
Field	Type	Null	Key
IDΠελάτη	int(4)		PRI
IDΚασέτας	int(4)		PRI
Από	datetime		PRI
Έως	datetime	YES	

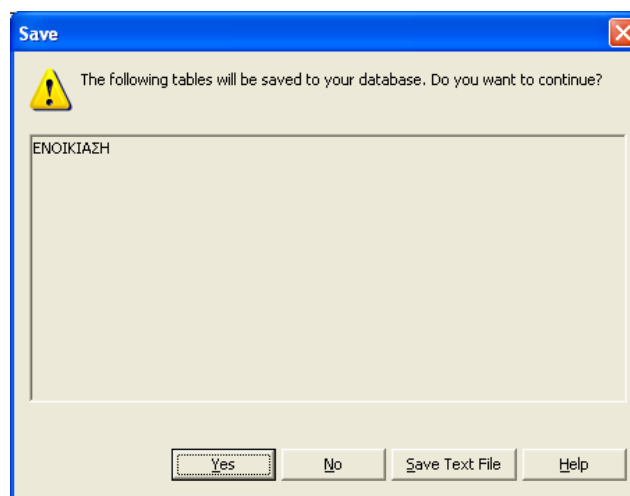
Στον πίνακα ΕΝΟΙΚΙΑΣΗ, το κύριο κλειδί αποτελείται από περισσότερες της μίας στήλης (IDΠελάτη, IDΚασέτας, Από). Σε αυτήν την περίπτωση, οι διπλές τιμές επιτρέπονται μόνο σε μία στήλη, αλλά κάθε συνδυασμός των τιμών από όλες τις στήλες που αποτελούν το κύριο κλειδί πρέπει να είναι μοναδικές.

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 ΚΑΣΕΤΑ.

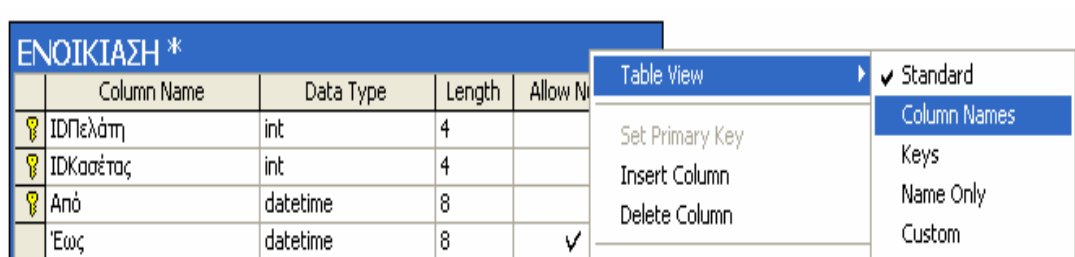
5.1. Κάντε κλικ στο **New Table** κουμπί  στην γραμμή εργαλείων ή κάντε δεξί κλικ στο διάγραμμα και επιλέξτε το **New Table**. Στο παράθυρο διαλόγου Choose Name επιλέξτε το όνομα ΕΝΟΙΚΙΑΣΗ και πατήστε OK. Ανοίγει ένα παράθυρο που μοιάζει με αυτό του Database Designer. Προσθέτουμε τα στοιχεία για τις στήλες και ορίζουμε το κύριο κλειδί (επιλογή με Ctrl των τριών πεδίων, δεξί κλικ πάνω τους και Set Primary Key), όπως φαίνεται στο σχήμα:

	Column Name	Data Type	Length	Allow Nulls
	IDΠελάτη	int	4	
	IDΚασέτας	int	4	
	Από	datetime	8	
	Έως	datetime	8	✓
▶				

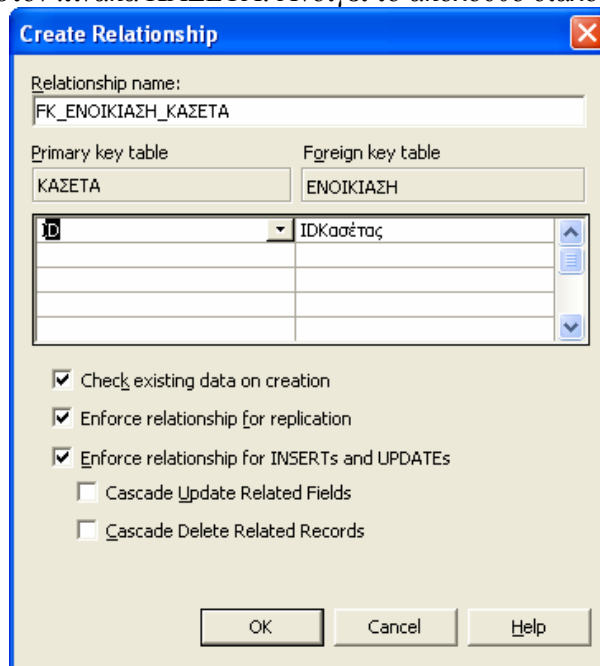
5.2. Πατήστε το εικονίδιο της δισκέτας για να σώσετε τον πίνακα. Ανοίγει το ακόλουθο διαλογικό παράθυρο, στο οποίο πατάμε το Yes.



5.3. Στη συνέχεια, κάνουμε δεξί κλικ πάνω στον πίνακα και επιλέγουμε Table View → Column Names.



- 5.4. Θα ορίσουμε τους περιορισμούς ξένου κλειδιού. Φέρνουμε τον πίνακα ΕΝΟΙΚΙΑΣΗ στα αριστερά, ανάμεσα στους πίνακες ΠΕΛΑΤΗΣ και ΚΑΣΕΤΑ. Στον πίνακα ΕΝΟΙΚΙΑΣΗ, κάνουμε (αριστερό κλικ) στο πεδίο IDΚασέτας και κρατώντας πατημένο το αριστερό πλήκτρο του ποντικιού, σέρνουμε και αφήνουμε (drag and drop) πάνω στον πίνακα ΚΑΣΕΤΑ. Ανοίγει το ακόλουθο διαλογικό παράθυρο.



- 5.5. Στο παράθυρο δηλώνεται ότι θα δημιουργηθεί περιορισμός ξένου κλειδιού μεταξύ του ΕΝΟΙΚΙΑΣΗ.IDΚασέτας και ΚΑΣΕΤΑ.ID. Επίσης, τσεκάρουμε τις δύο επιλογές στο τέλος για Cascade Update και Delete, και πατάμε OK.
- 5.6. Επαναλαμβάνουμε αντίστοιχες ενέργειες για τον περιορισμό ξένου κλειδιού μεταξύ ΕΝΟΙΚΙΑΣΗ.IDΠελάτη και ΠΕΛΑΤΗΣ.ID.

Στη συνέχεια, θα εισάγουμε πίνακα που υλοποιεί τη δυαδική συσχέτιση ΤΣ, μεταξύ του πίνακα ΤΑΙΝΙΑ και του πίνακα Συντελεστής.

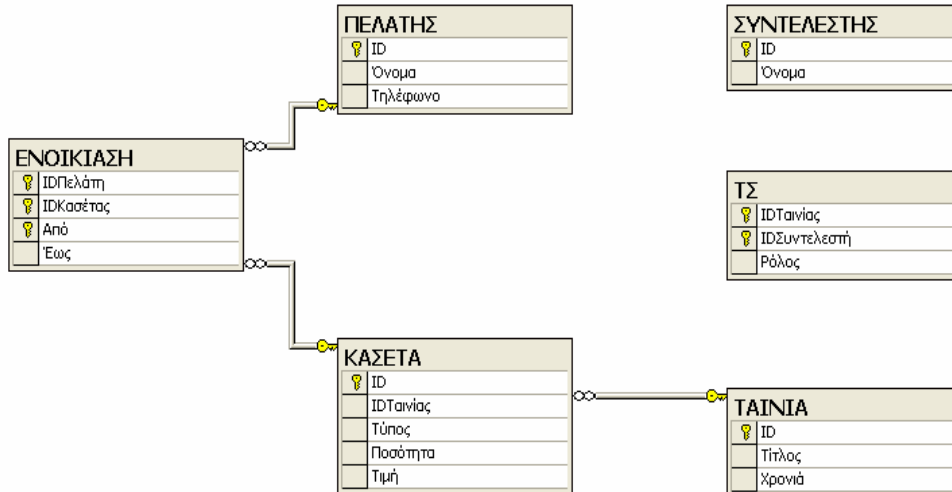
ΤΣ

Field	Type	Null	Key
IDΤαινίας	int (4)		PRI
IDΣυντελεστή	int (4)		PRI
Ρόλος	char (30)	YES	

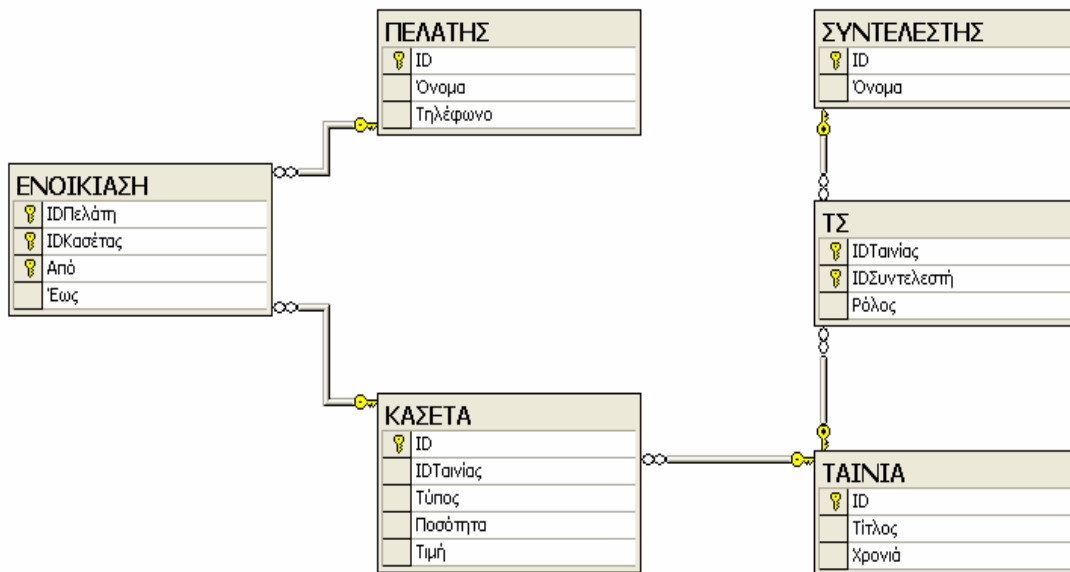
(Σημείωση: Λόγω του κύριου κλειδιού, ο ίδιος συντελεστής δεν μπορεί να συμμετέχει σε μία ταινία με περισσότερους από ένα ρόλους. Αυτό αποτελεί απλούστευση.)



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.

Εκτελούμε βήματα αντίστοιχα με αυτά στα βήματα 1—4 της προηγούμενης περίπτωσης. Τοποθετούμε τους πίνακες όπως φαίνεται στο επόμενο σχήμα:



Στη συνέχεια, θα ορίσουμε τους περιορισμούς ξένου κλειδιού για τον πίνακα ΤΣ. Κάνουμε κλικ και σέρνουμε το πεδίο ΤΣ.IDΤαινίας πάνω στον πίνακα ΤΑΙΝΙΑ. Στο διαλογικό παράθυρο που ανοίγει, κάνουμε τικ στις επιλογές για τα Cascades, και πατάμε ΟΚ. Όμοια, σέρνουμε το πεδίο ΤΣ.IDΣυντελεστή στον πίνακα ΣΥΝΤΕΛΕΣΤΗΣ, κάνουμε τικ στις επιλογές για τα Cascades, και πατάμε ΟΚ. Το τελικό διάγραμμα είναι το εξής:



Πατάμε το εικονίδιο της δισκέτας, για να σώσουμε το διάγραμμα. Τέλος, κλείνουμε την επεξεργασία του διαγράμματος από το κουμπί  (Προσοχή: όχι το )


4. ΕΙΣΑΓΩΓΗ ΤΙΜΩΝ ΣΤΟΥΣ ΠΙΝΑΚΕΣ

4.1. Εισαγωγή τιμών στους πίνακες με τον Enterprise Manager

Θα εισάγουμε τις ακόλουθες τιμές στον πίνακα ΤΑΙΝΙΑ:

<i>ID</i> Ταινίας	<i>Τίτλος</i>	<i>Χρονιά</i>
1	Rear Window	1954
2	Psycho	1960
3	Ben-Hur	1959

Είμαστε στην κονσόλα του Enterprise Manager και βλέπουμε τους πίνακες της βάσης videoclub.

1. Δεξί κλικ στον πίνακα ΤΑΙΝΙΑ.
2. Επιλέξτε την επιλογή Open Table.
3. Επιλέγουμε το Return all rows.
4. Εισάγετε τα δεδομένα με τρόπο που θα εισάγατε τιμές σε ένα φύλλο του Excel.
5. Αποθηκεύστε τις αλλαγές στον πίνακα πατώντας το κουμπί 

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

<i>ID</i>	<i>Όνομα</i>
1	Alfred Hitchcock
2	Grace Kelly
3	Anthony Perkins

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

<i>ID</i>	<i>Όνομα</i>	<i>Τηλέφωνο</i>
1	Perkins	246801
2	Καντακουζηνός	246801
3	Παλαιολόγος	987654

Τέλος, επαναλαμβάνουμε τα προηγούμενα βήματα για να εισάγουμε τις ακόλουθες τιμές στον πίνακα ΤΣ:

<i>ID</i> Ταινίας	<i>ID</i> Συντελεστή	<i>Ρόλος</i>
1	1	Σκηνοθέτης
2	1	Σκηνοθέτης
1	2	Ηθοποιός
2	3	Ηθοποιός

<

4.2. Εισαγωγή τιμών στους πίνακες με την SQL

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

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

Μεταβαίνουμε στον SQL Query Analyzer και σβήνουμε από τον editor πιθανά προηγούμενα επερωτήματα.



Δίνουμε την εξής εντολή:

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



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 ΚΑΣΕΤΑ  
VALUES (2, 1, 'DVD ', 1, 3)  
GO
```



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

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

Για να ελέγξουμε αν εισήχθησαν σωστά οι τιμές, επιστρέφουμε στον Enterprise Manager. Δεξί κλικ στον πίνακα ΚΑΣΕΤΑ → Open Table → Return All Rows. Κλείνουμε από το

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	Έως
1	1	07/10/2006	09/10/2006
1	2	09/20/2006	11/20/2006
2	1	09/10/2006	NULL



```
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
```



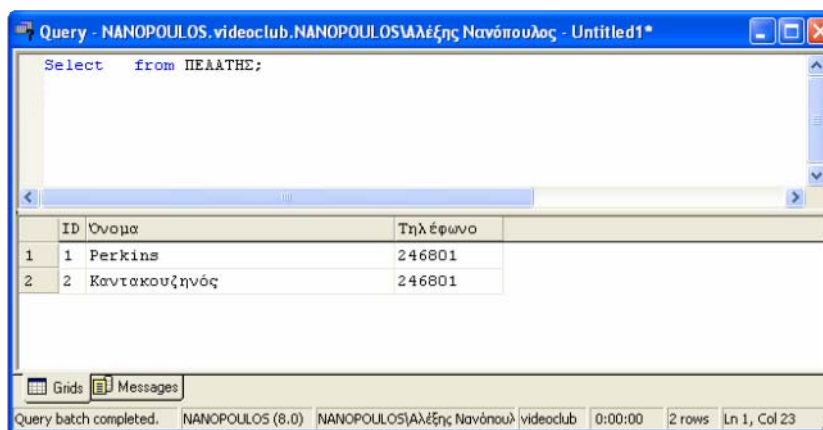
Αν παραλείψουμε ένα όρισμα, τότε στο αντίστοιχο πεδίο εισάγεται η τιμή NULL, αρκεί, βέβαια, να μην υπάρχει περιορισμός NOT NULL για αυτό το πεδίο. Για παράδειγμα:

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

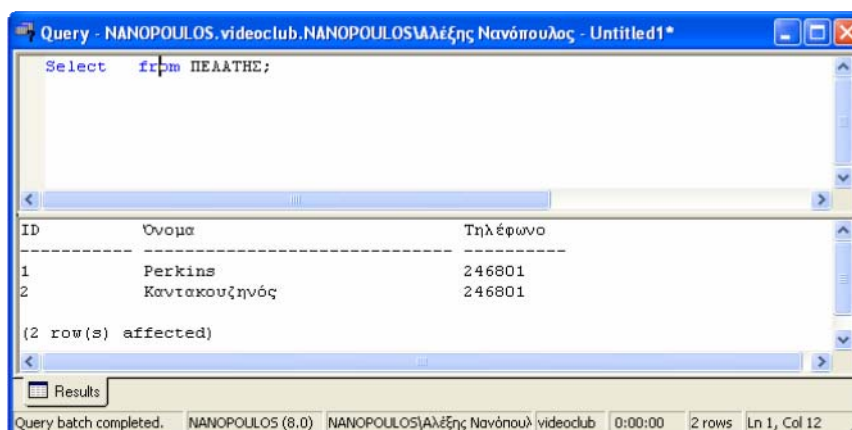
5. ΒΑΣΙΚΑ ΕΡΩΤΗΜΑΤΑ: ΕΠΙΛΟΓΗ ΚΑΙ ΠΡΟΒΟΛΗ


5.1. Διαχείριση του Results Pane

Όταν εκτελείτε μία Transact-SQL δήλωση, τα αποτελέσματα της ερώτησης εμφανίζονται στο Results Pane. Το Grids tab εμφανίζει το σύνολο των αποτελεσμάτων σε πλέγμα, όπως διακρίνεται και στην προηγούμενη εικόνα. Το πλέγμα εμφανίζεται παρόμοια με έναν πίνακα και σας επιτρέπει να επιλέγετε ξεχωριστά κελιά, στήλες ή γραμμές από το σύνολο των αποτελεσμάτων. Παράδειγμα δίνεται στο ακόλουθο σχήμα.



Το Results tab, όπως και το Grids tab, εμφανίζει το σύνολο των αποτελεσμάτων που παράγονται κατά την εκτέλεση μιας Transact-SQL δήλωσης. Στο Results tab, όμως, τα αποτελέσματα εμφανίζονται με την μορφή κειμένου, όπως φαίνεται στην παρακάτω εικόνα, και όχι σε μορφή πλέγματος.



Για να εμφανίζει ο SQL Server το Results tab, επιλέγουμε το “Execute Mode”  κουμπί από την γραμμή εργαλείων και επιλέγουμε το “Results in Text”. Το σύνολο των αποτελεσμάτων από το επόμενο ερώτημα που θα τρέξετε θα εμφανιστεί στο Results tab. Μπορείτε επίσης να έχετε το σύνολο των αποτελεσμάτων σε ένα αρχείο, αντί στο Grids tab ή στο Results tab. Για να στείλετε ένα σύνολο αποτελεσμάτων σε ένα αρχείο, επιλέγουμε το Execute Mode κουμπί στην γραμμή εργαλείων και μετά επιλέγουμε το Results to File. Θα σας ζητηθεί το όνομα του αρχείου και ο φάκελος στον οποίο θα αποθηκευτεί το αρχείο.

5.2. Εντολές επιλογής από έναν πίνακα

Η βασική λειτουργία της αναζήτησης πληροφοριών στη βάση δεδομένων γίνεται με την ακόλουθη επερώτηση:

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

Η σύνταξη της εντολής περιλαμβάνει τρεις *όρους*¹:

- Ο όρος **select** χρησιμοποιείται για τη δήλωση των n πεδίων A_1, A_2, \dots, A_n που θα περιέχονται στο αποτέλεσμα (αντιστοιχεί στην πράξη της προβολής (projection)).
- Ο όρος **from** δηλώνει το σύνολο των m σχέσεων (πινάκων) R_1, R_2, \dots, R_m 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 βιντεοκλάμπ. Ένα παράδειγμα ερώτησης είναι:



“Να δοθούν τα ονόματα όλων των πελατών”. Η αντίστοιχη εντολή σε SQL είναι

Q1

```
Select      Όνομα  
From       ΠΕΛΑΤΗΣ;
```

Το αποτέλεσμα είναι η σχέση:

Όνομα

Perkins
Καντακουζηνός
Παλαιολόγος

Ο όρος **select** περιέχει το πεδίο ΟΝΟΜΑ, το οποίο θα περιέχεται στο αποτέλεσμα. Τα υπόλοιπα πεδία της σχέσης ΠΕΛΑΤΗΣ, όπως για παράδειγμα το πεδίο ΤΗΛΕΦΩΝΟ, 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.

Η σχέση του αποτελέσματος που προκύπτει από την εντολή της SQL μπορεί να περιέχει δύο ίδιες πλειάδες. Για παράδειγμα, έστω η επερώτηση:



“Να δοθούν τα τηλέφωνα όλων των πελατών”. Η αντίστοιχη εντολή σε SQL είναι

Q2

```
select      Τηλέφωνο  
from       ΠΕΛΑΤΗΣ;
```

¹ Υπενθυμίζεται ότι η χρήση κεφαλαίων και μικρών γραμμάτων κατά την σύνταξη της εντολής είναι ισοδύναμη.

Το αποτέλεσμα είναι η σχέση:

```
Τηλέφωνο
-----
246801
246801
987654
```

Αν δύο πελάτες έχουν το ίδιο τηλέφωνο (π.χ., είναι συγγάτοικοι), 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.



Q3

Η αντίστοιχη εντολή είναι
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.



Q4

Για παράδειγμα, για την ερώτηση: “Να δοθούν όλα τα στοιχεία των πελατών”.
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 ότι για παράδειγμα ότι στο πεδίο ΤΗΛΕΦΩΝΟ θέλουμε να εμφανίζεται το πρόθεμα 2310. Θα χρησιμοποιήσουμε τη σύνδεση συμβολοσειρών.

Q5

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

```
Όνομα
-----
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        ΚΑΣΕΤΑ
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)

Σαν 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, με χρήση παρενθέσεων.



Π.χ., “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 είναι

Q7

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

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

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

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



Π.χ., “Να βρεθούν ποια ονόματα πελατών αρχίζουν από K”

Q8

```
Select      Όνομα
From        ΠΕΛΑΤΗΣ
Where       Όνομα like 'K%';
```

```
Όνομα
-----
Καντακουζηνός
```

Για την τιμή NULL ορίζονται μόνο οι τελεστές IS NULL και IS NOT NULL.



Π.χ., “Να βρεθούν τα στοιχεία των ενοικιάσεων που δεν έχει οριστεί ημερομηνία επιστροφής”

Q9

```
Select *
From ENOΙΚΙΑΣΗ
Where Έως IS NULL;
```

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

5.3. Ταξινόμηση αποτελεσμάτων

Η διάταξη των πλειάδων δεν έχει σημασία αφού τα στοιχεία ενός συνόλου δεν είναι διατεταγμένα. Πολλές φορές προκύπτει η ανάγκη ταξινόμησης των αποτελεσμάτων με βάση την τιμή ενός πεδίου. Η SQL επιτρέπει την ταξινόμηση με τον όρο **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 ΚΑΣΕΤΑ
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.



Π.χ.: “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 φθίνοντα τρόπο ως προς την ποσότητα”:

Q11

```
Select ID, Τιμή, Ποσότητα
From ΚΑΣΕΤΑ
Order by Τιμή desc, Ποσότητα asc;
```

ID	Τιμή	Ποσότητα
-----	-----	-----
2	3	1
3	2	2
1	2	3

Εκτός των αριθμητικών, η ταξινόμηση μπορεί να γίνει και σε διαφορετικούς τύπους δεδομένων. Επίσης, αξ σημειωθεί ότι η τιμή NULL θεωρείται μικρότερη από κάθε άλλη.



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

Q12

```

Select      ΕΝΟΙΚΙΑΣΗ.IDΚασέτας,   DAY(Έως)   as   Ημερομηνία,
              MONTH(Έως) as Μήνας,   YEAR(Έως) as Έτος
From        ΕΝΟΙΚΙΑΣΗ
Order by    Έως;
  
```

(**Σημείωση:** Οι συναρτήσεις DAY, MONTH, YEAR επιστρέφουν την ημερομηνία, μήνα και χρόνο, αντιστοίχως, ενός τύπου δεδομένων datetime.)

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

Όταν στο αποτέλεσμα της διάταξης 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        ΚΑΣΕΤΑ
Order by    Ποσότητα;
  
```

ID
2
3

6. ΒΑΣΙΚΑ ΕΡΩΤΗΜΑΤΑ: ΣΥΝΔΕΣΗ ΠΙΝΑΚΩΝ

6.1. Εσωτερική, Εξωτερική Σύνδεση

Με τον όρο **from** δηλώνεται το σύνολο των σχέσεων στις οποίες γίνεται η αναζήτηση. Στα προηγούμενα παραδείγματα οι εντολές αναφερόταν σε μόνο μία σχέση. Η χρήση περισσότερων σχέσεων αντιστοιχεί στην πράξη της *σύνδεσης* (join).



Εστω η ερώτηση: “Να δοθεί για κάθε συντελεστή (όνομα), οι ρόλοι με τους οποίους έχει συμμετάσχει σε ταινίες”. Η αντίστοιχη εντολή σε SQL είναι:

Q1

```
Select Όνομα, Ρόλος
From ΤΣ, ΣΥΝΤΕΛΕΣΤΗΣ
Where ΤΣ.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, όμως, είναι δυνατό να περιέχονται και άλλες που πρέπει να πληρούν οι πλειάδες του αποτελέσματος.



Για παράδειγμα, το ερώτημα: “Να δοθούν οι κωδικοί των ταινιών στις οποίες έχει συμμετάσχει ο Alfred Hitchcock”:

Q2

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

```
IDΤαινίας
-----
1
2
```

Σε αυτό το παράδειγμα, εκτός από την συνθήκη σύνδεσης, όπως προηγουμένως, 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.



Το ερώτημα του προηγούμενου παραδείγματος γίνεται:

Q3

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

```
IDΤαινίας
-----
1
2
```

Σύνδεση μπορεί να γίνει και σε περισσότερους από 2 πίνακες.



Π.χ., “Να βρεθούν για κάθε πελάτη (όνομα), ο κωδικός και η τιμή των κασετών που έχει ενοικιάσει”:

Q4

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

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

Στο αποτέλεσμα του inner join συμμετέχουν μόνο οι πλειάδες 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.



Π.χ., “Να βρεθούν για κάθε πελάτη (όνομα), ο κωδικός και η τιμή των κασετών που έχει ενοικιάσει. Να εμφανίζονται και οι πελάτες που δεν έχουν ενοικιάσει κάποια κασέτα”:

Q5

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

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

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.



Π.χ., “Να βρεθούν για κάθε πελάτη (όνομα), ο κωδικός και η τιμή των κασετών που έχει ενοικιάσει. Να εμφανίζονται οι κωδικοί και τιμές των κασετών που δεν έχουν ενοικιαστεί από κάποιον πελάτη”:

Q6

```

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



Π.χ., “ Να βρεθούν για κάθε πελάτη (όνομα), 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 ”:

Q7

```

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

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

6.2. Μετονομασία και Αυτο-σύνδεση

Σε ορισμένα ερωτήματα προκύπτει η ανάγκη μετονομασίας πεδίων ή σχέσεων. Αυτό γίνεται με την χρήση ενός επιπλέον όρου που ονομάζεται **as**. Έστω το παράδειγμα της επερώτησης Q5, (προηγούμενο κεφάλαιο), 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 ΤΗΛΕΦΩΝΟ 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

Ο όρος **as** χρησιμοποιείται και με τον όρο **from** για τη μετονομασία σχέσεων.



Έστω για παράδειγμα η ερώτηση: “Να βρεθεί το όνομα κάθε πελάτη που έχει το ίδιο τηλέφωνο με τον πελάτη με το όνομα 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.



Π.χ., “Να βρεθούν ο κωδικός κάθε ταινίας για την οποία η κασέτα τύπου VHS είναι σε μεγαλύτερη διαθέσιμη ποσότητα από την αντίστοιχη κασέτα τύπου DVD”.

Q10

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

IDΤαινίας
1

7. ΣΥΝΘΕΤΑ ΕΡΩΤΗΜΑΤΑ: ΟΜΑΔΟΠΟΙΗΣΗ

7.1. Συναρτήσεις ομαδοποίησης

Η SQL περιέχει τις εξής κυριότερες συναρτήσεις ομαδοποίησης (aggregate functions):

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

Οι συναρτήσεις *αθροίσματος* και *μέσου όρου* δέχονται σαν είσοδο μόνο αριθμητικές τιμές, ενώ οι υπόλοιπες μπορούν να δεχθούν τιμές και άλλων τύπων, όπως αλφαριθμητικά.



Σαν παράδειγμα, για την ερώτηση: “Να βρεθεί η μεγαλύτερη τιμή ενοικίασης κασέτας”, η εντολή της SQL είναι

Q1

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

Μέγιστη Τιμή Κασέτας

3



Σαν ένα άλλο παράδειγμα, έστω η ερώτηση: “Να βρεθεί ο συνολικός αριθμός των κασετών”. Η αντίστοιχη εντολή είναι:

Q2

```
Select      count(*)  
From        ΚΑΣΕΤΑ;
```

3

Μπορούν να εφαρμοσθούν και αλγεβρικές πράξεις μεταξύ των συναρτήσεων.



“Να βρεθεί ο η διαφορά μεταξύ της ακριβότερης και της φθηνότερης κασέτας”:

Q3

```
Select      max(Τιμή) - min(Τιμή)  
From        ΚΑΣΕΤΑ;
```

1

Η ύπαρξη ορισμάτων εντός 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*.



“Να βρεθεί ο αριθμός των πελατών που έχουν κάνει ενοικιάσεις”:

Q4

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

2

7.2. 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        ΚΑΣΕΤΑ
Group by    Τύπος;
```

Τύπος	Μέση Τιμή
DVD	3
VHS	2

Μπορείτε να τοποθετηθείτε περισσότερες από μία στήλη στον όρος **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.



“Για κάθε πελάτη (κωδικός) να βρεθεί ο αριθμός των φορών που ενοικίασε κάθε κασέτα (κωδικός), χωρίς να εμφανίζονται στοιχεία για κασέτες που δεν έχουν ενοικιαστεί”:

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.

Ομαδοποίηση μπορεί να γίνει και στο αποτέλεσμα σύνδεσης.



“Για κάθε συντελεστή (όνομα) να βρεθεί ο αριθμός των διακριτών ρόλων του σε όλες τις ταινίες”:

Q7

```
Select      ΣΥΝΤΕΛΕΣΤΗΣ.Όνομα, count (distinct ΤΣ.Ρόλος) as
            'Αριθμός Διακριτών Ρόλων'
From        ΣΥΝΤΕΛΕΣΤΗΣ inner join ΤΣ
            on ΣΥΝΤΕΛΕΣΤΗΣ.ID = ΤΣ.IDΣυντελεστή
Group by    ΣΥΝΤΕΛΕΣΤΗΣ.Όνομα;
```

Όνομα	Αριθμός Διακριτών Ρόλων
Alfred Hitchcock	1
Anthony Perkins	1
Grace Kelly	1

Η group by παρέχει μια προαιρετική λέξη κλειδί, ALL, η οποία διευκρινίζει ότι όλες οι ομάδες πρέπει να περιληφθούν στο σύνολο αποτελέσματος, ακόμα κι αν δεν ικανοποιούν την συνθήκη αναζήτησης. Οι ομάδες που δεν έχουν σειρές που ικανοποιούν την συνθήκη αναζήτησης περιέχουν NULL στη συνοπτική στήλη έτσι ώστε να μπορούν να προσδιοριστούν εύκολα.



“Να βρεθεί η μέση ποσότητα ανά τύπο κασέτας, για τις κασέτες των οποίων η τιμή είναι μικρότερη από 2. Να παρουσιάζονται επίσης οι κασέτες που δεν ικανοποιούν την συνθήκη”:

```

Select      Τύπος, avg (Ποσότητα) as 'Μέση Ποσότητα'
From        ΚΑΣΕΤΑ
Where       Τιμή <= 2
Group by    all Τύπος;

```

Q8

Τύπος	Μέση Ποσότητα
DVD	NULL
VHS	2

Τα αποτελέσματα του group by μπορούν να ταξινομηθούν ως προς τις συναρτήσεις ομαδοποίησης.



Π.χ., “Να βρεθεί η μέση τιμή ανά τύπο κασέτας. Τα αποτελέσματα να είναι ταξινομημένα κατά αύξουσα μέση τιμή”:

```

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

```

Q9

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.



Π.χ., “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”:

Q10

```

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.

7.3. Ο όρος Having

Είναι δυνατό ο διαχωρισμός μιας σχέσεις σε τμήματα με τον όρο group by, να συνδυαστεί και με κάποια συνθήκη που πρέπει να ικανοποιεί το κάθε τμήμα (**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 η ερώτηση: “Να βρεθούν οι τύποι κασέτας για τους οποίους ο μέσος όρος τιμής είναι μεγαλύτερος από 2”.

Q11

```

Select      Τύπος, avg (Τιμή)
From        ΚΑΣΕΤΑ
Group by    Τύπος
Having      avg (Τιμή) > 2;

```

```

Τύπος
-----
DVD      3

```



Ακόμα ένα παράδειγμα: “Να βρεθούν τα ονόματα των συντελεστών που έχουν συμμετοχή σε περισσότερες από μία ταινίες”:

Q12

```

Select      Όνομα
From        ΤΣ inner join ΣΥΝΤΕΛΕΣΤΗΣ on
                ΤΣ.IDΣυντελεστή= ΣΥΝΤΕΛΕΣΤΗΣ.ID
Group by    ΤΣ.IDΣυντελεστή, Όνομα
Having      count (*) > 1;

```

```

Όνομα
-----
Alfred Hitchcock

```

Η στήλη Όνομα προστίθεται στο group by μόνο για να μπορεί να προβληθεί στο αποτέλεσμα, εφόσον στον όρο select δεν μπορεί να υπάρχει στήλη που δεν υπάρχει και στον όρο group by. Παρατηρήστε ότι, επειδή ο πρώτος όρος του group by είναι ο IDΣυντελεστή, η προσθήκη της στήλης Όνομα δεν δημιουργεί περαιτέρω διαχωρισμό (γιατί;)

Σε μία επερώτηση SQL μπορεί να υπάρχει ταυτόχρονα συνθήκη για τον όρο **where** και για τον όρο **having**. Σε αυτήν την περίπτωση, πρώτα εκτελείτε η επιλογή του where, στη συνέχεια, γίνεται το group by, και τέλος το having.



Για παράδειγμα, η ερώτηση: “Να βρεθούν οι κωδικοί των συντελεστών που έχουν σκηνοθετήσει περισσότερες από μία ταινίες”:

Q13

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

```
IDΣυντελεστή
-----
1
```

8. ΣΥΝΘΕΤΑ ΕΡΩΤΗΜΑΤΑ: ΠΡΑΞΕΙΣ ΣΥΝΟΛΩΝ

8.1. Ένωση συνόλων

Ο όρος **union** πραγματοποιεί την πράξη της *ένωσης* σχέσεων. Για την εφαρμογή αυτού του όρου μεταξύ δύο σχέσεων πρέπει αυτές να είναι *συμβατές*, δηλαδή να έχουν τα ίδια πεδία.



Έστω η ερώτηση: “Να βρεθούν οι τίτλοι των ταινιών που γυρίστηκαν το 1959 ή αυτών στις οποίες συμμετέχει η Grace Kelly”. Η αντίστοιχη εντολή είναι

```
( Select   Τίτλος
From     ΤΑΙΝΙΑ
Where    Χρονιά = 1959 )
union
( Select   Τίτλος
From     ΤΑΙΝΙΑ, ΣΥΝΤΕΛΕΣΤΗΣ, ΤΣ
Where    ΤΣ.IDΤαινίας = ΤΑΙΝΙΑ.ID and
           ΤΣ.IDΣυντελεστή= ΣΥΝΤΕΛΕΣΤΗΣ.ID and
           ΣΥΝΤΕΛΕΣΤΗΣ.Όνομα = 'Grace Kelly' );
```

Q1

Τίτλος

Ben-Hur
Rear Window

Αν χρησιμοποιήσετε την προαιρετική λέξη **ALL** με την **union**, οι διπλές σειρές θα περιληφθούν στο αποτέλεσμα. Επίσης, μπορείτε να χρησιμοποιήσετε την **order by** μόνο στην τελευταία δήλωση μιας ένωσης, όχι σε κάθε **select** δήλωση. Αυτός ο περιορισμός εξασφαλίζει ότι το τελικό σύνολο αποτελέσματος θα είναι ταξινομηθεί μόνο μια φορά για όλα τα αποτελέσματα.



“Να βρεθούν οι τίτλοι των ταινιών που γυρίστηκαν πριν το 1960 ή αυτές στις οποίες συμμετέχει η Grace Kelly. Να γίνει προβολή των διπλοεγγραφών, καθώς και ταξινόμηση ως προς τον τίτλο της ταινίας”:

```
( Select   Τίτλος
From     ΤΑΙΝΙΑ
Where    Χρονιά < 1960 )
union all
( Select   Τίτλος
From     ΤΑΙΝΙΑ, ΣΥΝΤΕΛΕΣΤΗΣ, ΤΣ
Where    ΤΣ.IDΤαινίας = ΤΑΙΝΙΑ.ID and
           ΤΣ.IDΣυντελεστή= ΣΥΝΤΕΛΕΣΤΗΣ.ID and
           ΣΥΝΤΕΛΕΣΤΗΣ.Όνομα = 'Grace Kelly' )
Order by Τίτλος;
```

Q2

Τίτλος

Ben-Hur
Rear Window
Rear Window

8.2. Εμφωλιασμένες εντολές

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.

8.2.1. Έλεγχος περιεκτικότητας σε σύνολο

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 \in .



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. Θέλουμε να εξεταστεί η απάντηση με χρήση του όρου **in**.

Q3

```
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”:

Q4

```
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.



Να εξετασθεί το ερώτημα “Να βρεθούν οι κωδικοί των ταινιών στις οποίες έχει συμμετάσχει ο Alfred Hitchcock και έχουν ενοικιασθεί περισσότερες από δύο φορές”:

Q5

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

```
IDΤαινίας
-----
1
```

8.2.2. Σύγκριση συνόλων

Σε κάποιες περιπτώσεις προκύπτουν ερωτήματα όπου ένα πεδίο πρέπει να συγκριθεί με την τιμή του πεδίου *μίας τουλάχιστον* πλειάδας μίας σχέσης, η οποία είναι αποτέλεσμα φωλιασμένης εντολής. Σε αυτή την περίπτωση χρησιμοποιείται ο όρος της SQL **some** (ισοδύναμος είναι ο όρος **any**). Αν πρέπει να συγκριθεί με την τιμή του πεδίου *όλων των* πλειάδων της σχέσης, τότε χρησιμοποιείται ο όρος **all**.



Έστω η ερώτηση: “Να βρεθεί ο κωδικός του πελάτη που έκανε την πρώτη ενοικίαση”:

Q6

```
Select IDΠελάτη
From   ΕΝΟΙΚΙΑΣΗ
Where  Από <= all
      (Select Από
       From ΕΝΟΙΚΙΑΣΗ);
```

```
IDΠελάτη
-----
1
```

(Σημείωση: γιατί όχι **select IDΠελάτη, min(Από) From ΕΝΟΙΚΙΑΣΗ;**)



Άλλο παράδειγμα: “Να βρεθούν οι πελάτες (Όνομα) για τους οποίους υπάρχει κάποια ενοικιασμένη κασέτα για την οποία δεν έχει ορισθεί ημερομηνία επιστροφής”:

Q7

```
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 φωλιασμένα, δηλαδή **max(count(*))**, είναι δυνατό να χρησιμοποιηθεί ο όρος **all** για να δηλώσει ότι μέγιστη τιμή είναι αυτή που είναι μεγαλύτερη από όλες.



“Να βρεθεί ο κωδικός πελάτη με το μεγαλύτερο αριθμό ενοικιάσεων”.

Q8

```

Select      IDΠελάτη
From        ΕΝΟΙΚΙΑΣΗ
Group by   IDΠελάτη
Having      count (*) >= all ( Select count (*)
                                From      ΕΝΟΙΚΙΑΣΗ
                                Group by  IDΠελάτη );

```

IDΠελάτη

1

8.2.3. Έλεγχος ύπαρξης και Διαφορά συνόλων

Η SQL περιέχει τους όρους **exists** και **not exists** η οποίοι ελέγχουν αν μία εντολή παράγει σαν αποτέλεσμα μία σχέση που έχει πλειάδες ή είναι άδεια, αντίστοιχα.



“Να βρεθούν τα ονόματα των πελατών που έχουν κάνει μία τουλάχιστον ενοικίαση”:
(Σημείωση: Εναλλακτικά θα μπορούσε να γίνει και με **inner join**.)

Q9

```

Select      Όνομα
From        ΠΕΛΑΤΗΣ
where       exists (select IDΠελάτη
                    from      ΕΝΟΙΚΙΑΣΗ
                    where     IDΠελάτη = ΠΕΛΑΤΗΣ.ID);

```

Όνομα

Perkins
Καντακουζηνός

Ο όρος **not exists** μπορεί να χρησιμοποιηθεί για να ελεγχθεί αν μία σχέση Y περιέχει μία άλλη 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 .²



“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)”:

Q10

```

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

```

IDΠελάτη

1

² Ο SQLServer2000 δεν υποστηρίζει την πράξη της διαφοράς (**except**).

9. ΟΨΕΙΣ

Ο ορισμός όψης στην SQL γίνεται με τη δήλωση **create view**.



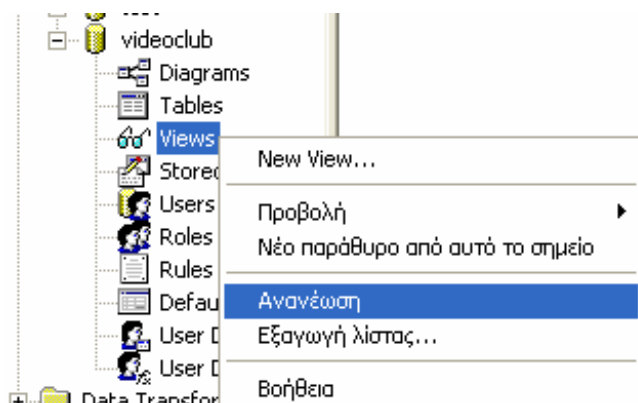
“Να ορισθεί όψη με όνομα H, που περιέχει όλα τους κωδικούς των ταινιών που συμμετείχε ο Alfred Hitchcock”:

Q11

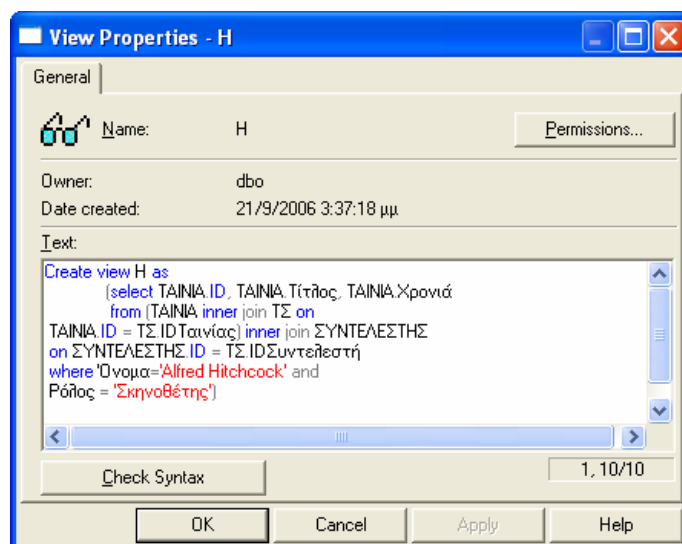
Create view H as

```
(select IDΤαινίας
From ΤΣ inner join ΣΥΝΤΕΛΕΣΤΗΣ on
    ΤΣ.IDΣυντελεστή = ΣΥΝΤΕΛΕΣΤΗΣ.ID
where ΣΥΝΤΕΛΕΣΤΗΣ.Όνομα = 'Alfred Hitchcock')
```

Επιστρέφουμε στον Enterprise Manager και κάτω από τη βάση videoclub, επεκτείνουμε την επιλογή Views κάνουμε δεξί κλικ πάνω της και επιλέγουμε Refresh.



Εμφανίζεται η όψη H. Με διπλό κλικ πάνω της, βλέπουμε τη σύνταξή της:



Μέσω των όψεων μπορούμε να απλοποιήσουμε τη σύνταξη ερωτημάτων.



Π.χ., μέσω της όψης H, απλοποιήστε το ερώτημα του προηγούμενου κεφαλαίου: “Να βρεθούν οι κωδικοί των ταινιών στις οποίες έχει συμμετάσχει ο Alfred Hitchcock και έχουν ενοικιασθεί περισσότερες από δύο φορές”:

Q12

```

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

```

```

IDΤαινίας
-----
1

```

Η διαγραφή μίας όψης γίνεται με τη δήλωση **drop view**.



Διαγράψτε την όψη H.

Q13

```
drop view H;
```

Άλλο παράδειγμα χρήσης όψης για απλοποίηση ερωτήματος: “Να βρεθεί ο κωδικός πελάτη με το μεγαλύτερο αριθμό ενοικιάσεων”, το οποίο εξετάσαμε στο Κεφάλαιο 8.2.1 με χρήση των όρων **having** και **all**.

Q14

```

Create view ΑρΕνοικ as
    (select IDΠελάτη, count(*) as A
     From ΕΝΟΙΚΙΑΣΗ
     Group by IDΠελάτη)

```

```

Select      IDΠελάτη
From        ΑρΕνοικ
Where       A = (select max(A)
                from ΑρΕνοικ);

```

```

IDΠελάτη
-----
1

```

10. ΜΕΤΑΒΟΛΗ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

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.

10.1. Εισαγωγή πολλών γραμμών σε πίνακα

Στην SQL, είναι δυνατή η εισαγωγή γραμμών οι οποίες ανήκουν σε σχέση-αποτέλεσμα ενός ερωτήματος.



Για παράδειγμα, “Άς δημιουργηθεί ένας πίνακας για ταινίες πριν από το 1960, με το όνομα ΠΑΛΙΑ_ΤΑΙΝΙΑ”:

Q1

```
CREATE TABLE ΠΑΛΙΑ_ΤΑΙΝΙΑ(  
    ID int NOT NULL ,  
    Τίτλος char(100) DEFAULT NULL,  
    Χρονιά int DEFAULT 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”:

Q2

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



“Δώστε όλα τα περιεχόμενα του πίνακα ΠΑΛΙΑ_ΤΑΙΝΙΑ”:

10.2. Ενημέρωση

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: “Γίνεται ενοικίαση της κασέτας με κωδικό 1. Να μειωθεί κατά 1 η ένδειξη της διαθέσιμης ποσότητας της”:

Q3

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


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 (μπορούν να είναι και περισσότερα από ένα).



Π.χ., “Γίνεται ανακαίνιση και πλέον όλες οι κασέτες είναι τύπου DVD και σε διαθέσιμη ποσότητα ίση με 5. Να ενημερωθεί ο πίνακας ΚΑΣΕΤΑ”:

Q4

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

10.3. Διαγραφή

Στην SQL επιτρέπεται η διαγραφή ολόκληρων πλειάδων και όχι η μερική διαγραφή για ορισμένα μόνο πεδία. Η διαγραφή γίνεται με την εντολή **delete**.



Για παράδειγμα, “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, “διαγράψτε όλα τα περιεχόμενα του πίνακα ΠΕΛΑΤΗΣ”

Q6

```
Delete from ΠΕΛΑΤΗΣ;
```

(Προσοχή: πριν εκτελεστεί η εντολή, δεν ζητείται επιβεβαίωση.)

10.4. Μεταβολή σχήματος πίνακα

Για την προσθήκη, διαγραφή ή μετονομασία πεδίων σε έναν πίνακα, υπάρχει η εντολή της SQL **alter table**.



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

Q7

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

(Σημείωση: οι τιμές του πεδίου Γλώσσα είναι αρχικά **null** για όλες τις γραμμές.)



“Να διαγραφεί το πεδίο Γλώσσα από τον πίνακα ΤΑΙΝΙΑ”:

Q8

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

Είναι δυνατή και η αλλαγή τύπου δεδομένων για ένα πεδίο.



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

Q9

```
Alter table ΤΑ
```

```
Alter column Ρόλος varchar(100);
```

10.5. Μετονομασία

Η μετονομασία αντικειμένων στην T-SQL γίνεται με τη χρήση της ενσωματωμένης συνάρτησης `sp_rename`.



Π.χ., “Να μετονομαστεί ο πίνακας ΚΑΣΕΤΑ σε DVD”:

Q10

```
Exec sp_rename 'ΚΑΣΕΤΑ', 'DVD';
```



Π.χ., “Να μετονομαστεί πεδίο Χρονιά του πίνακα ΤΑΙΝΙΑ σε Έτος”:

Q11

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

10.6. Κατάργηση πίνακα και βάσης

Για την κατάργηση ενός πίνακα από τη βάση δεδομένων υπάρχει η εντολή `drop table`.



Για παράδειγμα “Να διαγραφεί ο πίνακας ΠΑΛΙΑ_ΤΑΙΝΙΑ”:

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.

Η διαγραφή μίας βάσης γίνεται με την εντολή:

```
Drop database Όνομα_Βάσης;
```

(Σημείωση: ΜΗΝ διαγράφετε την τρέχουσα βάση!)

11. ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΕΡΩΤΗΜΑΤΩΝ

11.1. Παρακολούθηση Πλάνου Εκτέλεσης

Ο Query Optimizer είναι ένα εσωτερικό module το οποίο προσπαθεί να βρει το βέλτιστο πλάνο εκτέλεσης για κάθε T-SQL δήλωση. Ο Query Optimizer αναλύει κάθε T-SQL δήλωση, ψάχνει για τον αριθμό των πιθανών πλάνων εκτέλεσης και υπολογίζει το κόστος του κάθε πλάνου, σε σχέση με τους πόρους που είναι απαραίτητοι και το χρόνο επεξεργασίας, και επιλέγει το λιγότερο ακριβό πλάνο εκτέλεσης.

Χρησιμοποιώντας τον Query Analyzer, μπορείτε να δείτε το πλάνο εκτέλεσης το οποίο ο SQL Server query optimizer επέλεξε για το SQL ερώτημα. Πληκτρολογήστε το ερώτημα και πατήστε Ctrl+L.



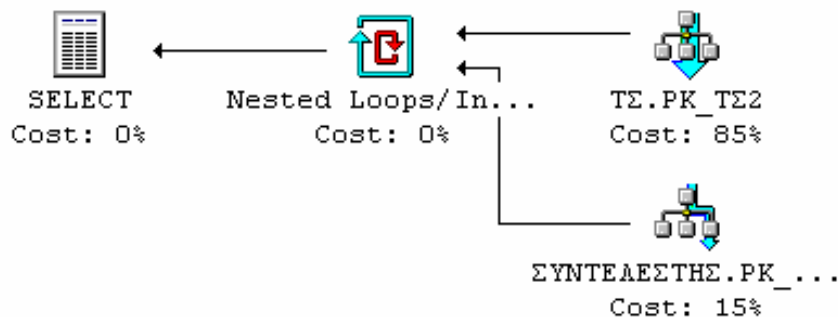
Π.χ., να απεικονισθεί το πλάνο εκτέλεσης για το ερώτημα “Προβάλετε τους κωδικούς των ταινιών στις οποίες συμμετείχε η Grace Kelly”:

Q13

```
select IDΤαινίας
from ΤΣ, ΣΥΝΤΕΛΕΣΤΗΣ
where ΣΥΝΤΕΛΕΣΤΗΣ.ID = ΤΣ.IDΣυντελεστή and Όνομα='Grace Kelly'
```

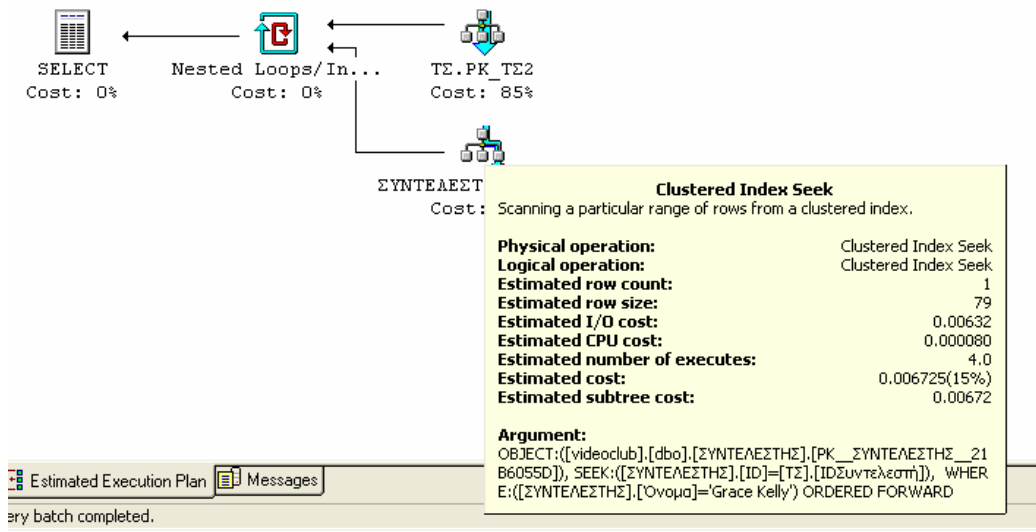
```
+-----+
| IDΤαινίας |
+-----+
|      1    |
+-----+
```

Query 1: Query cost (relative to the batch): 100.00%
Query text: select IDΤαινίας from ΤΣ, ΣΥΝΤΕΛΕΣΤΗΣ where



Το Εκτιμώμενο Πλάνο Εκτέλεσης παρέχει πρόσβαση σε επιπρόσθετα δεδομένα. Για να τις δείτε, κρατήστε το ποντίκι πάνω στο εικονίδιο της λειτουργίας. Εμφανίζεται ένα pop-up παράθυρο που περιέχει τις πρόσθετες πληροφορίες, όπως φαίνεται και στο επόμενο σχήμα.

Query 1: Query cost (relative to the batch): 100.00%
 Query text: select IDΤαινίας from ΤΣ, ΣΥΝΤΕΛΕΣΤΗΣ where ΣΥΝΤΕΛΕΣΤΗΣ.ID = ΤΣ.IDΣυντελεσ



Το pop-up παράθυρο περιέχει τις παρακάτω πληροφορίες:

- **Physical Operation/Logical Operation** Οι λειτουργίες που εκτελούνται από το ερώτημα, όπως index scans, joins, aggregations κ.ά.
- **Estimated Row Count** Ο αριθμός των σειρών που ο query optimizer υπολογίζει ότι θα ανακτηθούν από την λειτουργία.
- **Estimated Row Size** Το εκτιμώμενο μέγεθος των ανακατωμένων σειρών σε bytes.
- **Estimated I/O Cost/Estimated CPU Cost** Οι εκτιμώμενοι I/O πόροι και χρόνος CPU τα οποία θα χρησιμοποιηθούν από αυτήν την λειτουργία.
- **Estimated Number of Executes** Ο εκτιμώμενος αριθμός φορών που θα εκτελεστεί αυτή η λειτουργία κατά τη διάρκεια της εκτέλεσης.
- **Estimated Cost** Το κόστος της λειτουργίας όπως αυτό υπολογίζεται από τον query optimizer. Αυτό το κόστος εμφανίζεται ως ποσοστό του συνολικού κόστους του συνολικού κόστους.
- **Estimated Subtree Cost** Το εκτιμώμενο κόστος εκτέλεσης των προηγούμενων και αυτού του συγκεκριμένου τμήματος.

11.2. Κατάλογοι

Ένας κατάλογος 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.



“Να δημιουργηθεί κατάλογος για το όνομα του συντελεστή”:

```

Create index      IND_ΣΥΝΤΕΛΕΣΤΗΣ
on               ΣΥΝΤΕΛΕΣΤΗΣ
(
                Όνομα
)
  
```

Q14

Η ύπαρξη ενός καταλόγου, π.χ., για το όνομα συντελεστή, βοηθά το πλάνο εκτέλεσης, ώστε να γρήγορα γίνει πρώτα η επιλογή βάσει της συνθήκης για το όνομα, και στη συνέχεια η σύνδεση.



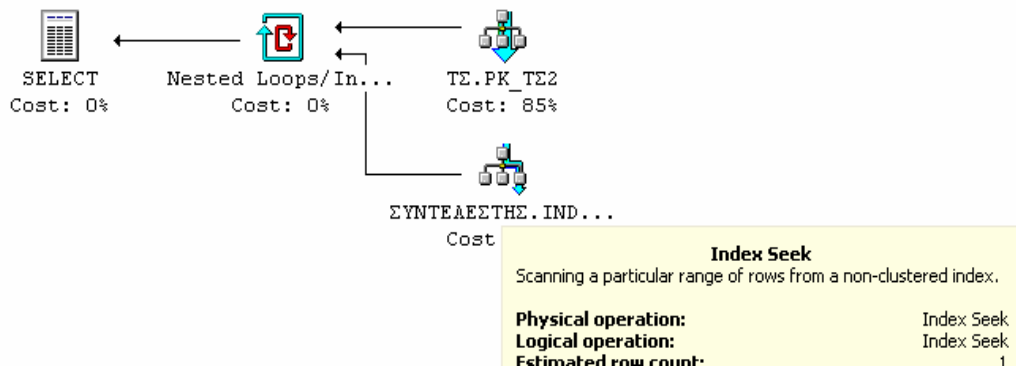
Π.χ., να βελτιωθεί το πλάνο εκτέλεσης για το ερώτημα “Προβάλετε τους κωδικούς των ταινιών στις οποίες συμμετείχε η Grace Kelly”, με χρήση καταλόγου για το όνομα του συντελεστή:

Q15

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

Η χρήση του όρου WITH επιβάλλει τη χρήση του καταλόγου. Ο όρος WITH είναι προαιρετικός. Το πλάνο εκτέλεσης είναι το ακόλουθο:

Query 1: Query cost (relative to the batch): 100.00%
 Query text: select IDΤαινίας from ΤΣ, ΣΥΝΤΕΛΕΣΤΗΣ where ΣΥΝΤΕΛΕΣΤΗΣ.ID = ΤΣ.IDΣυντε



Στο πλάνο αυτό φαίνεται ότι η συνθήκη για το όνομα συντελεστή γίνεται μέσω του non-clustered καταλόγου που δημιουργήσαμε (clustered κατάλογος δημιουργείται by default από τον SQL Server επί του κύριου κλειδιού).

Στην T-SQL επιτρέπεται η διαγραφή ενός καταλόγου.



Π.χ., “Να διαγραφεί ο κατάλογος IND_ΣΥΝΤΕΛΕΣΤΗΣ”:

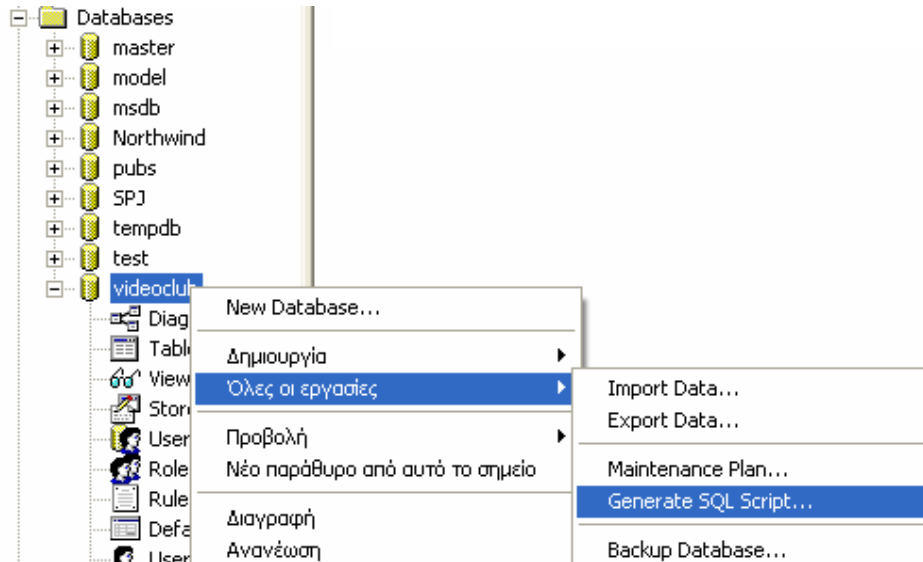
Q16

```
Drop index ΣΥΝΤΕΛΕΣΤΗΣ.IND_ΣΥΝΤΕΛΕΣΤΗΣ;
```

12. SCRIPTING ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Με τη λειτουργία του scripting μπορούμε να δημιουργήσουμε ένα αρχείο με εντολές SQL, οι οποίες θα κατασκευάζουν τη βάση δεδομένων. Το αρχείο αυτό μπορεί να λειτουργήσει και ως αντίγραφο ασφαλείας για τη βάση δεδομένων, καθώς με την εκτέλεσή του ανακτούμε το σχήμα της βάσης μας.

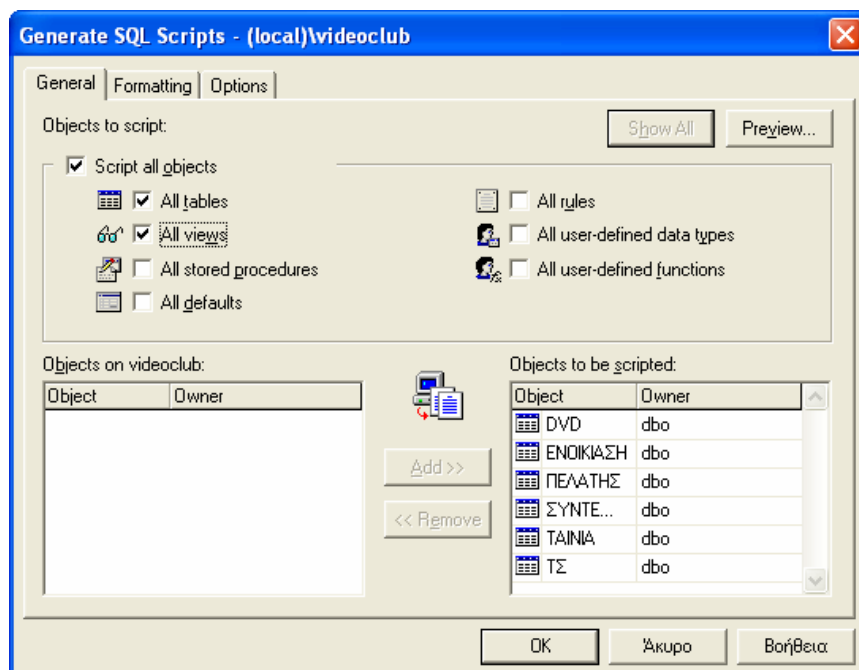
Δεξί κλικ στη βάση δεδομένων videoclub → Όλες οι εργασίες → Generate SQL Script...



Ανοίγει διαλογικό παράθυρο στο οποίο πατάμε το κουμπί Show All. Επιλέγουμε (με τικ) τα αντικείμενα που θέλουμε να καταγραφούν, συγκεκριμένα, τα ακόλουθα:

- All tables, και
- All views

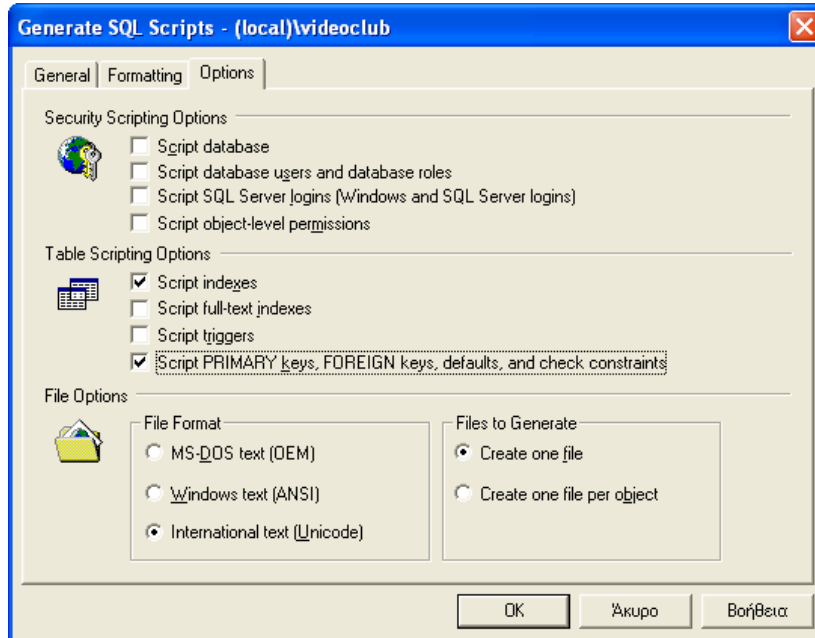
όπως φαίνεται στο ακόλουθο σχήμα:



Στο tab με όνομα Options, επιλέγουμε (με tick) τα εξής:

- Script indexes,
- Script PRIMARY keys, FOREIGN keys, defaults, and check constraints

όπως φαίνεται στο ακόλουθο σχήμα:



Πατάμε το OK και δίνουμε το όνομα του αρχείου. Ανοίγουμε το αρχείο και εξετάζουμε τα περιεχόμενά του.