

## Κεφάλαιο 2. Δημιουργία Βάσης Δεδομένων και Πινάκων

### Σύνοψη

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

### 2.1. Ορισμός και Δημιουργία μιας Βάσης Δεδομένων

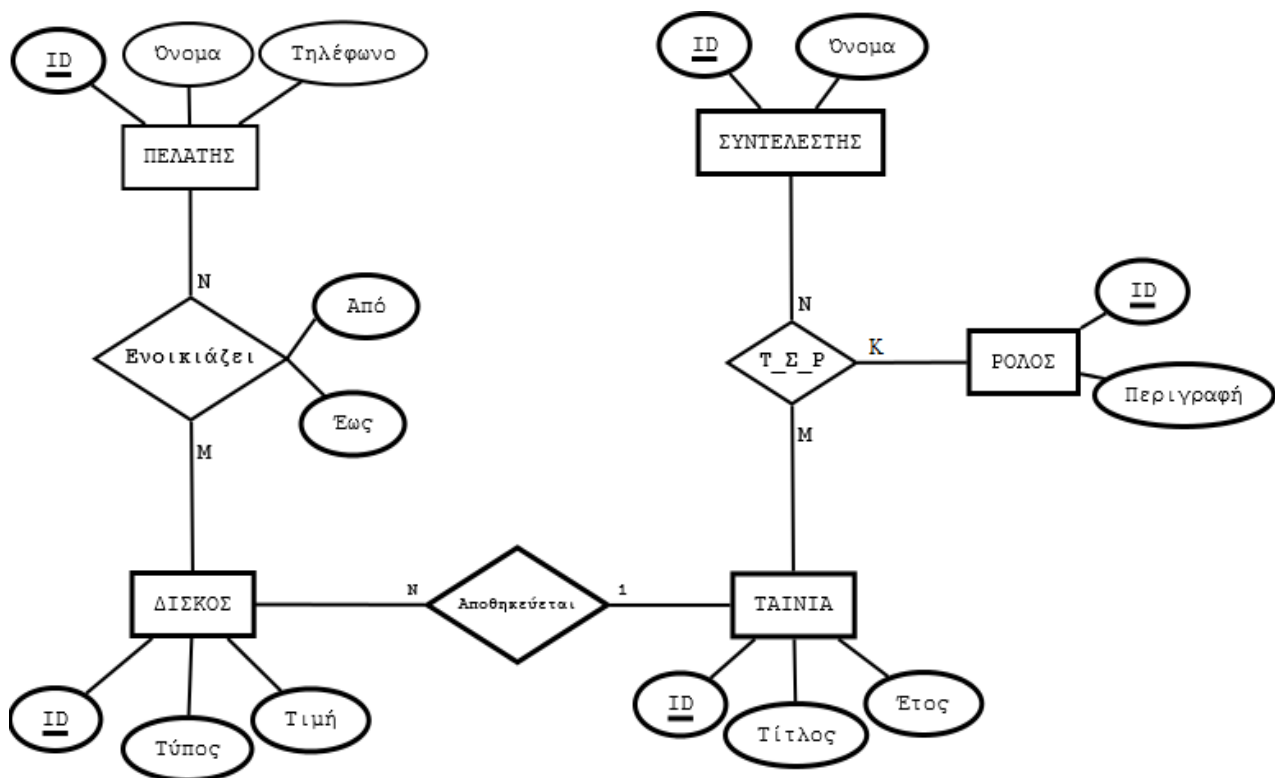
Ο SQL Server είναι ένα σύστημα διαχείρισης βάσεων δεδομένων (database management system). Ως εκ τούτου, υποστηρίζει όλες τις λειτουργίες που πρέπει να προσφέρει μια βάση δεδομένων, όπως είναι η αναζήτηση, η εισαγωγή, η διαγραφή, η ενημέρωση εγγραφών κλπ. Η βάση δεδομένων είναι μια συλλογή στοιχείων που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

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

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

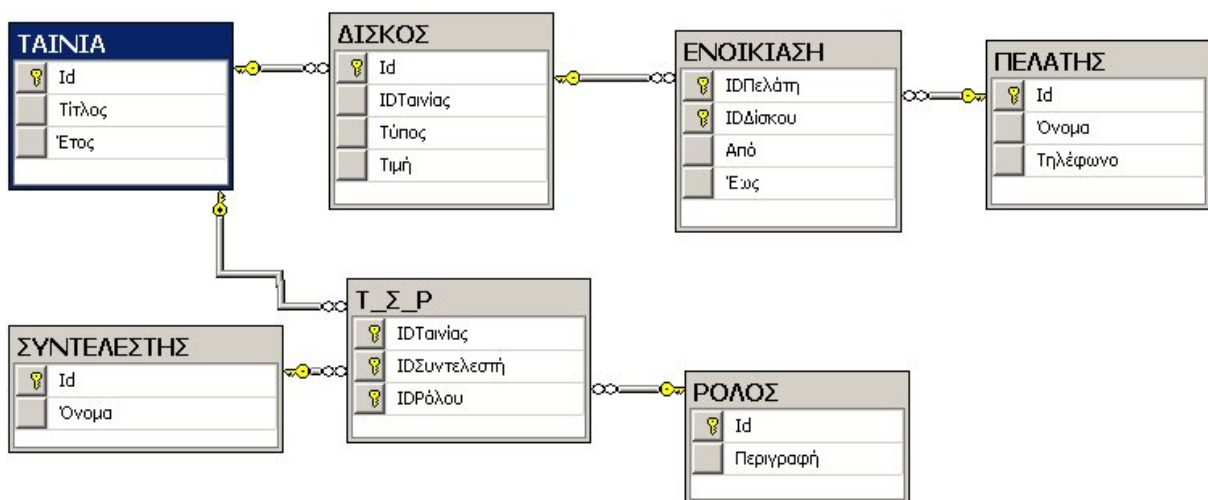
- Το DVDclub έχει ένα σύνολο από πελάτες για τους οποίους καταχωρεί ένα μοναδικό κωδικό, το όνομά τους και το τηλέφωνό τους.
- Ο κάθε πελάτης μπορεί να νοικιάσει ταινίες που αποθηκεύονται σε δίσκους dvd από μία ημερομηνία μέχρι μία άλλη ημερομηνία.
- Η κάθε ταινία χαρακτηρίζεται από ένα μοναδικό κωδικό,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

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



Εικόνα 2.1 Διάγραμμα Οντοτήτων – Συσχετίσεων του DVD club

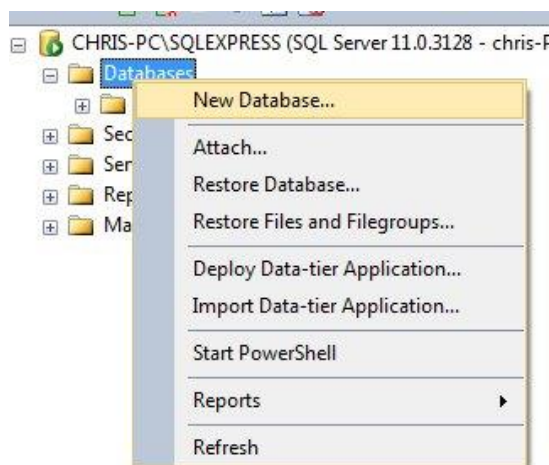
Συνολικά, το διάγραμμα Οντοτήτων-Συσχετίσεων (Ο-Σ) αποτελείται από πέντε οντότητες (ΠΕΛΑΤΗΣ, ΔΙΣΚΟΣ, ΤΑΙΝΙΑ, ΣΥΝΤΕΛΕΣΤΗΣ, ΡΟΛΟΣ) και τρεις συσχετίσεις (Ενοικιάζει, Αποθηκεύεται, Τ\_Σ\_Ρ). Από αυτές τις συσχετίσεις, οι δύο (Ενοικιάζει και Τ\_Σ\_Ρ) συσχετίζουν οντότητες με συσχέτιση πολλά προς πολλά και,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



Εικόνα 2.2 Σχεσιακό Σχήμα του DVDclub

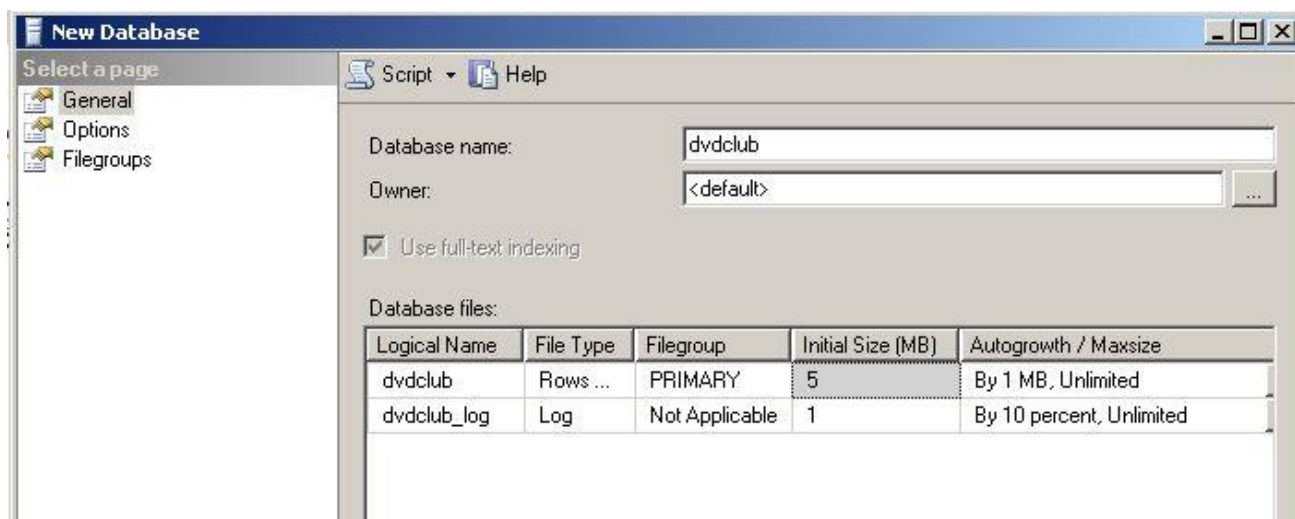
### 2.1.1. Δημιουργία νέας βάσης σε γραφικό περιβάλλον

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



Εικόνα 2.3

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

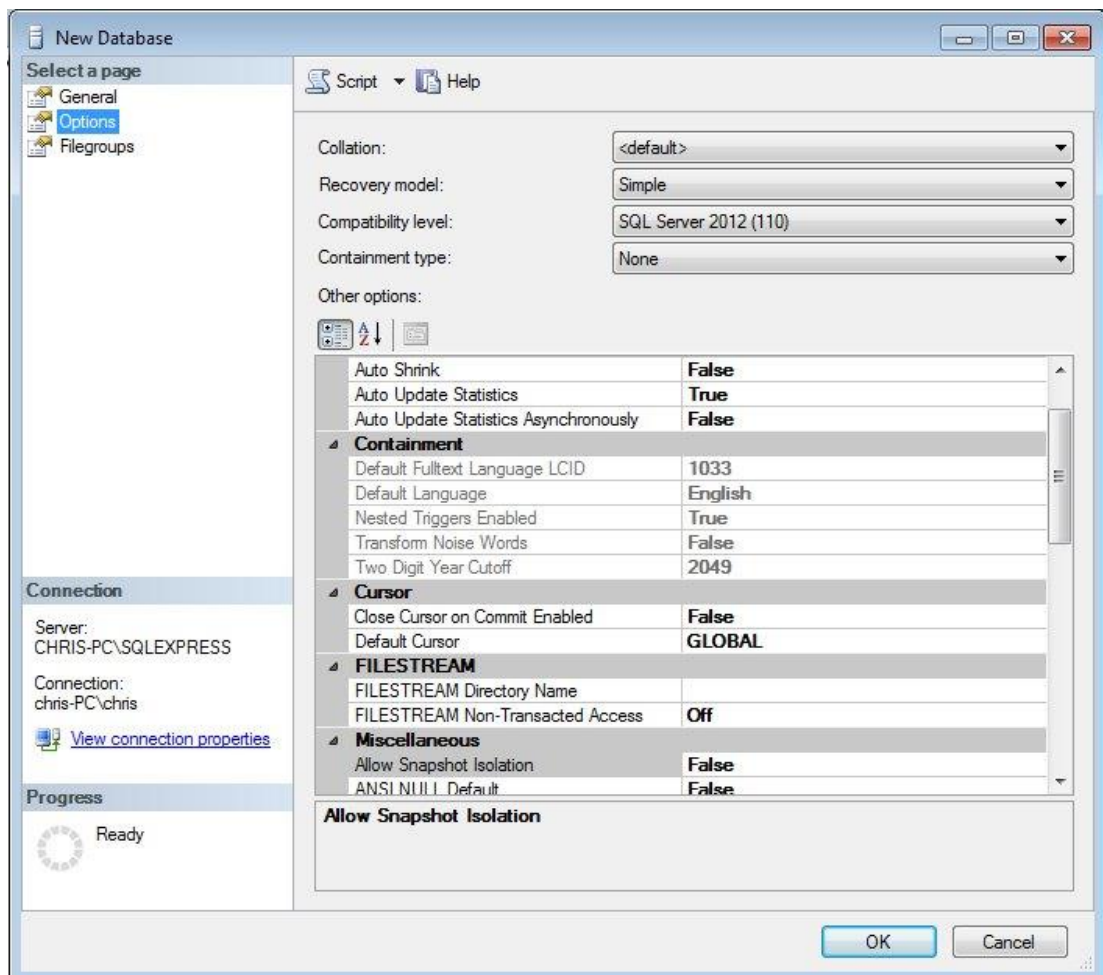


Εικόνα 2.4

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

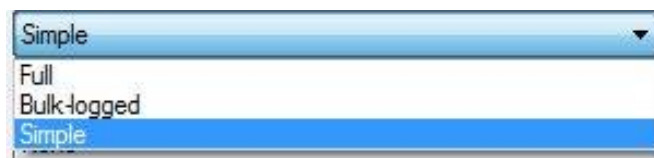
Οι πιο σημαντικές είναι οι παρακάτω:

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



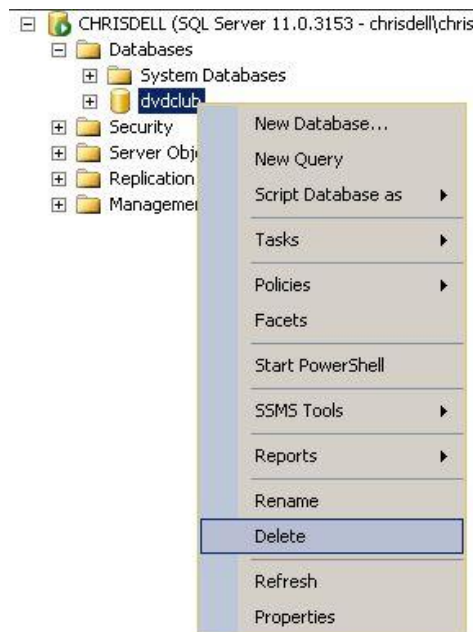
Εικόνα 2.5

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



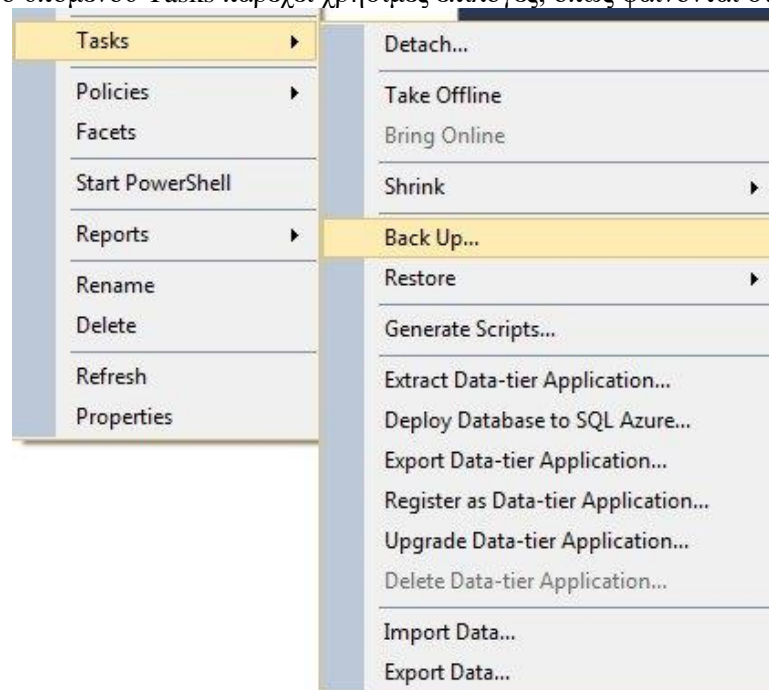
Εικόνα 2.6

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



Εικόνα 2.7

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



Εικόνα 2.8

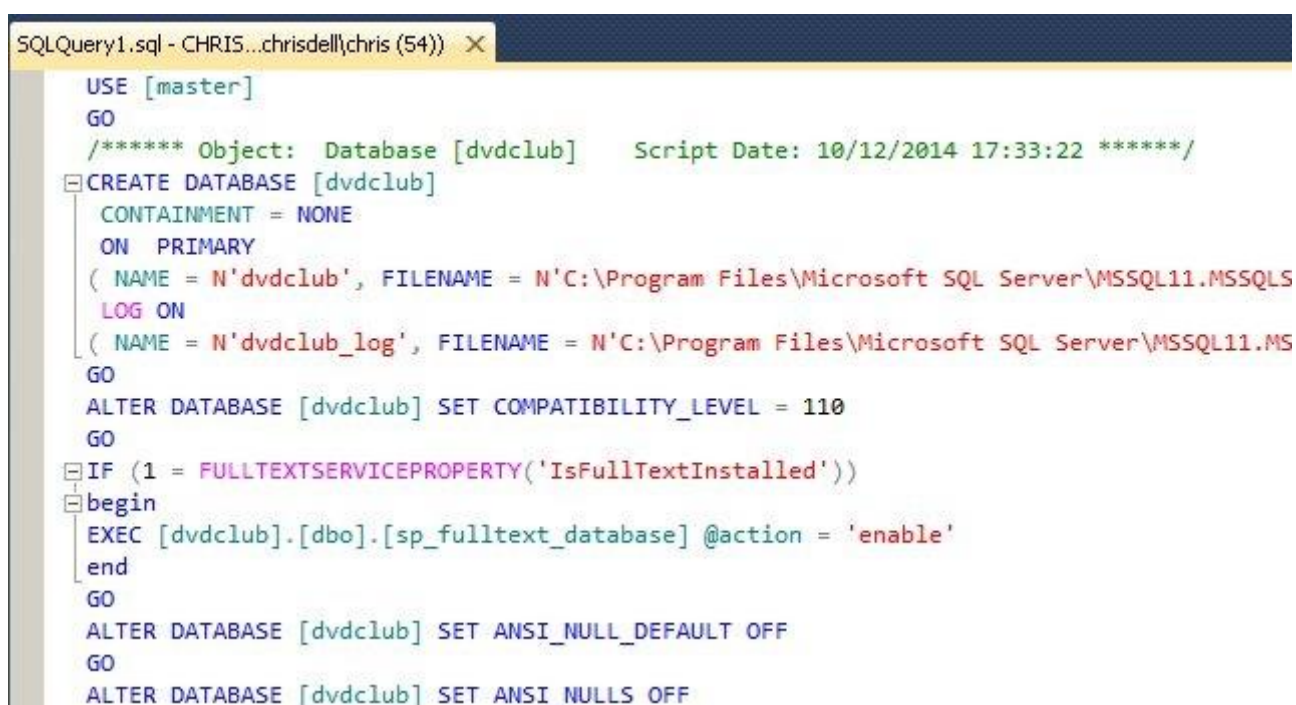
Πιο σημαντική είναι η διαδικασία του «Back Up», κατά την οποία θα επιλέξουμε το όνομα του αρχείου, προκειμένου να αποθηκεύεται κάθε φορά ένα αρχείο με κατάληξη «bak» που θα περιέχει όλα τα δεδομένα της βάσης μας. Μια άλλη χρήσιμη επιλογή είναι το «Restore», όταν θέλουμε να επαναφέρουμε μια βάση δεδομένων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

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

### 2.1.2. Δημιουργία νέας βάσης με κώδικα SQL

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

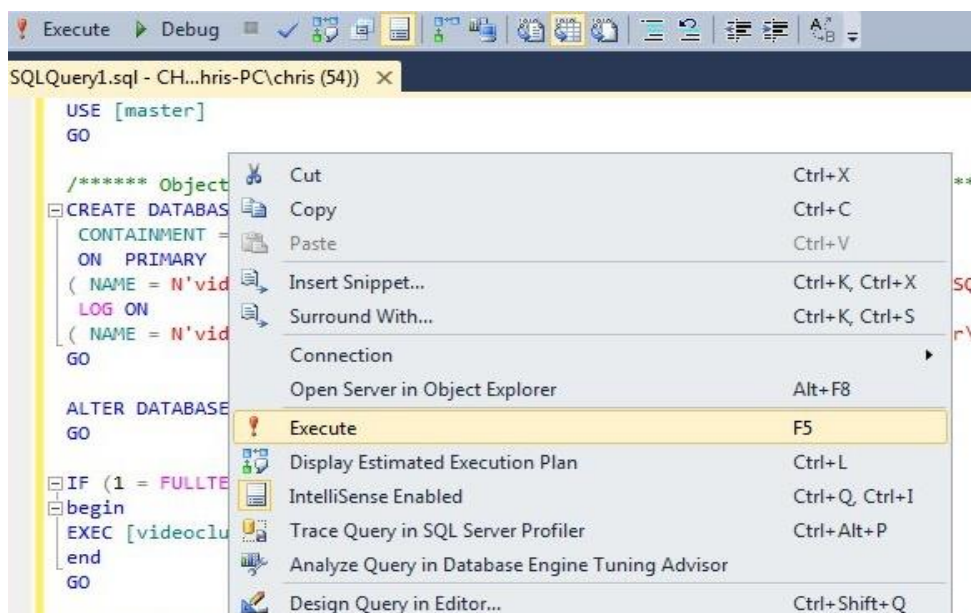
Αφού δημιουργήσουμε τη βάση δεδομένων, επιλέγουμε Generate Scripts από το αναδυόμενο μενού της Εικόνας 2.8 και, έτσι, φανερώνεται ένας οδηγός όπου αυτόματα παράγονται οι ανάλογες εντολές SQL, οι οποίες μπορούν να δημιουργήσουν τα υπόλοιπα στοιχεία της βάσης που φτιάξαμε με γραφικό τρόπο, όπως φαίνεται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



```
SQLQuery1.sql - CHRIS...chrisdell\chris (54) X
USE [master]
GO
/***** Object: Database [dvdclub]    Script Date: 10/12/2014 17:33:22 *****/
CREATE DATABASE [dvdclub]
    CONTAINMENT = NONE
    ON PRIMARY
    ( NAME = N'dvdclub', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLS
    LOG ON
    ( NAME = N'dvdclub_log', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL11.MS
GO
ALTER DATABASE [dvdclub] SET COMPATIBILITY_LEVEL = 110
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [dvdclub].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO
ALTER DATABASE [dvdclub] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [dvdclub] SET ANSI_NULLS OFF
```

Εικόνα 2.9





Εικόνα 2.10

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

### 2.1.3. Διαγραφή μίας βάσης με κώδικα SQL

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

## 2.2. Βασικές Έννοιες και Δημιουργία Πινάκων

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

### 2.2.1. Τύποι Δεδομένων

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

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

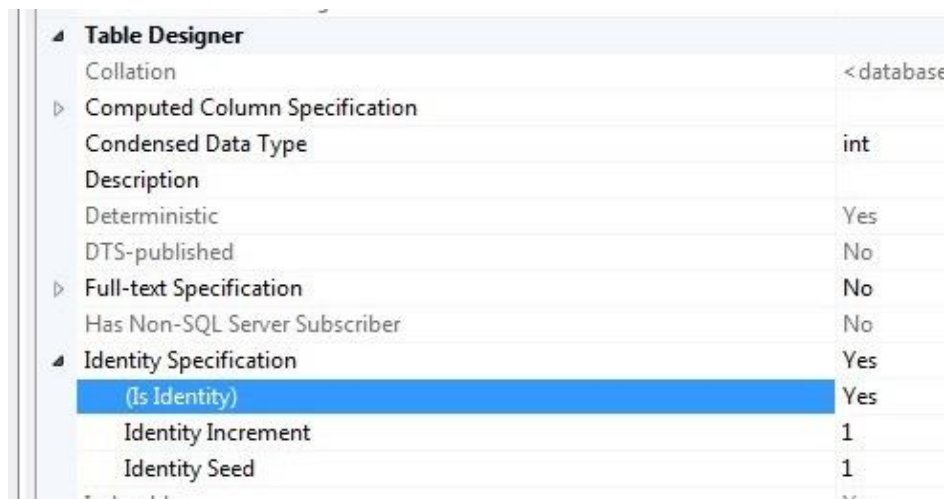
Πίνακας 2.1



### 2.2.2. Χρήσιμες συμβουλές για τους τύπους δεδομένων

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

- Εφόσον το μέγεθος ενός πεδίου συμβολοσειράς είναι μεταβλητό και όχι σταθερό (π.χ όλες οι καταχωρήσεις αποτελούνται από ένα κωδικό σταθερού μήκους 5 χαρακτήρων) προτιμήστε τους τύπους `varchar`, `nvarchar` και όχι `char`, `nchar`.
- Εφόσον θέλουμε η εφαρμογή μας να υποστηρίζει τη δυνατότητα του χρήστη να καταχωρεί Unicode χαρακτήρες (π.χ. γαλλικά ή ρωσικά ονόματα), πρέπει να χρησιμοποιήσουμε τον τύπο `nvarchar` και όχι `varchar`.
- Εφόσον θέλουμε να αποθηκεύσουμε μόνο την ημερομηνία μιας συναλλαγής, χρησιμοποιούμε τον τύπο `Date` και όχι τον τύπο `Datetime`, ο οποίος αποθηκεύει και την ώρα της συναλλαγής. Ο δεύτερος τύπος δεδομένων όχι μόνο σπαταλά αποθηκευτικό χώρο αλλά μπορεί και να δημιουργήσει προβλήματα σε ερωτήματα SQL. Αν για παράδειγμα, αποθηκεύσουμε στη βάση δεδομένων την ημερομηνία και την ακριβή ώρα στην οποία αγοράζεται ένα προϊόν με τη χρήση του τύπου δεδομένων `Datetime` (π.χ. '01/31/2012 13:47'), τότε, στην περίπτωση που θα θέλαμε να ανακτήσουμε όλες τις αγορές προϊόντων που έγιναν μέχρι και τις 31 Ιανουρίου του 2012, θα πρέπει να διαμορφώσουμε ένα ερώτημα με τέτοιο τρόπο ώστε να συμπεριλαμβάνεται ολόκληρη η υπό εξέταση μέρα, δηλαδή (`<= '01/31/2012 23:59'`). Διαφορετικά, αν το ερώτημα μας είχε απλά την μορφή (`<= '01/31/2012'`), τότε δεν θα εμφανίζονταν η αγορά που έγινε στις 31 Ιανουρίου του 2012, διότι το σύστημα εσφαλμένα θα εκλάμβανε το ερώτημα μας ως (`<= '01/31/2012 00:00'`).
- Εφόσον ένα πεδίο χρησιμοποιείται σαν κύριο κλειδί ή σαν ξένο κλειδί (τμήμα συσχέτισης), καλό είναι να χρησιμοποιούμε τον αριθμητικό τύπο δεδομένων `int` και όχι `smallint`, `tinyint`.
- Γενικότερα οι αριθμητικοί τύποι δεδομένων χωρίζονται σε προσεγγιστικούς (`Real`, `Float`) και ακριβείς (`decimal`, `numeric`). Στους τελευταίους δηλώνουμε πόσα ακριβώς ψηφία επιθυμούμε πριν και μετά την υποδιαστολή. Το σημείο αυτό χρήζει προσοχής σε σχέση με την διαχείριση αθροισμάτων και ποσών. Στους προσεγγιστικούς αριθμούς, ανάλογα με τον τρόπο υπολογισμού αθροισμάτων, μπορούμε να έχουμε προβλήματα απώλειας – ακρίβειας δεκαδικών ψηφίων. Για την κλασική ανάγκη αποθήκευσης χρηματικών ποσών προτείνεται ο, απόλυτα ακριβής, τύπος `decimal (9,2)` ή `decimal (18,2)` για μεγαλύτερα ποσά. .
- Πολύ πρακτική και χρήσιμη είναι η λειτουργία ενός αυτόματου κλειδιού. Αυτό, όπως φαίνεται στην Εικόνα 2.11, υλοποιείται με ένα πεδίο τύπου `int`, στις ιδιότητες του οποίου ρυθμίζουμε σε `yes` την επιλογή `Identity Specification`. Με τον τρόπο αυτό είμαστε σίγουροι ότι θα έχουμε, χωρίς να ασχολούμαστε στα ερωτήματα εισαγωγής, ένα σωστό μοναδικό κλειδί αυτόματα.



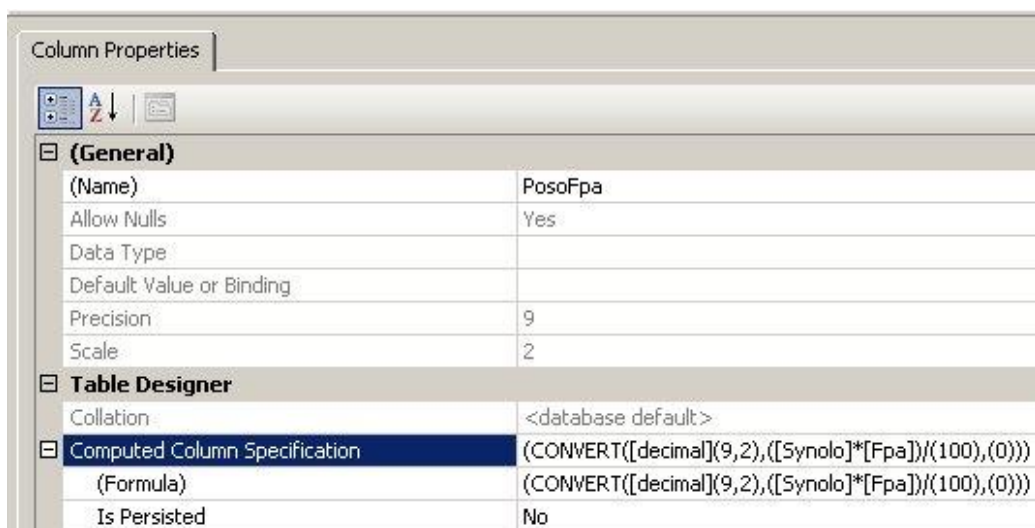
Εικόνα 2.11

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

|         |               |                                     |
|---------|---------------|-------------------------------------|
| Synolo  | decimal(9, 2) | <input type="checkbox"/>            |
| Fpa     | decimal(5, 2) | <input 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 τμήμα formula του πεδίου Computed Column Specification γράφουμε τον τύπο υπολογισμού  $[\text{Synolo}] * [\text{Fpa}]$ . Αν θέλουμε να στρογγυλοποιούμε στα δύο δεκαδικά, τότε γράφουμε:  $\text{CONVERT}([\text{decimal}](9,2), ([\text{Synolo}] * [\text{Fpa}]) / (100), (0))$ .



Εικόνα 2.13

### 2.2.3. Δημιουργία πινάκων με τον Management Studio

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

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

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

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

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

Πίνακας 2.2 TAINIA

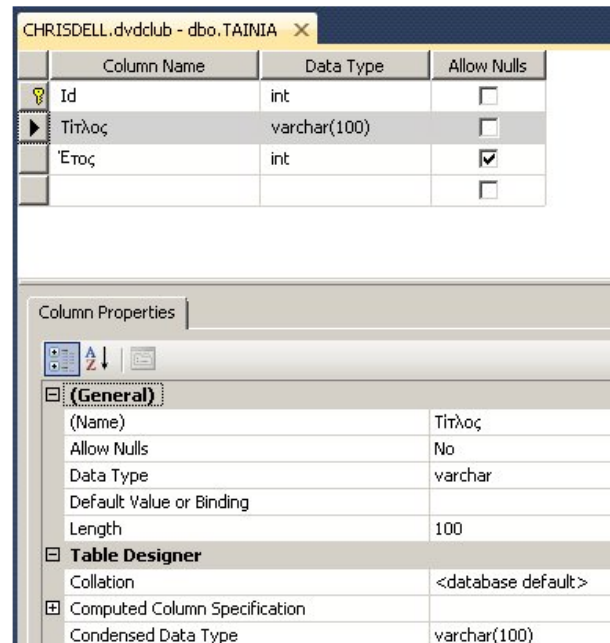
Προκειμένου να δημιουργήσουμε αυτόν το νέο πίνακα στην βάση μας με γραφικό τρόπο, ανοίγουμε τη δένδρική δομή. Στο αριστερό pane του Management Studio επεκτείνουμε τη βάση δεδομένων DVDclub, όπως φαίνεται στην Εικόνα 2.14. Στη συνέχεια, επεκτείνοντας το φάκελο Tables, βλέπουμε τους πίνακες που δημιουργεί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



Εικόνα 2.14

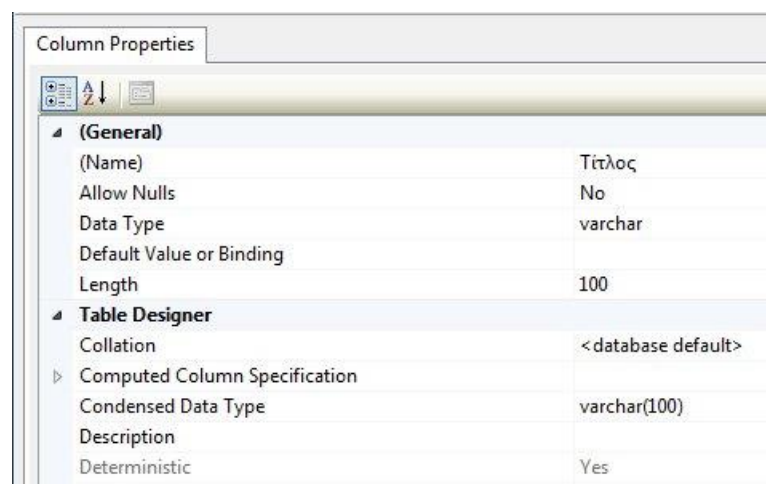
Ανοίγει ο Table Designer, όπως φαίνεται στην Εικόνα 2.15, ο οποίος μας επιτρέπει να σχεδιάσουμε το νέο πίνακα με γραφικό τρόπο. Συμπληρώνουμε τα στοιχεία του Πίνακα 2.2, όπως φαίνεται στην Εικόνα 2.15. Για

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



Εικόνα 2.15

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



Εικόνα 2.16

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

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

Πίνακας 2.3 ΠΕΛΑΤΗΣ

#### 2.2.4. Δημιουργία πινάκων με κώδικα της SQL

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

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

Πίνακας 2.4 ΣΥΝΤΕΛΕΣΤΗΣ

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

1<sup>ος</sup> τρόπος:

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

2<sup>ος</sup> τρόπος:

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

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



Εικόνα 2.17

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

Ο πίνακας ΔΙΣΚΟΣ έχει την ακόλουθη μορφή:

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

Πίνακας 2.5 ΔΙΣΚΟΣ

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

1<sup>ος</sup> τρόπος:

**CREATE TABLE ΔΙΣΚΟΣ**

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

2<sup>ος</sup> τρόπος:

**CREATE TABLE ΔΙΣΚΟΣ**

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



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

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

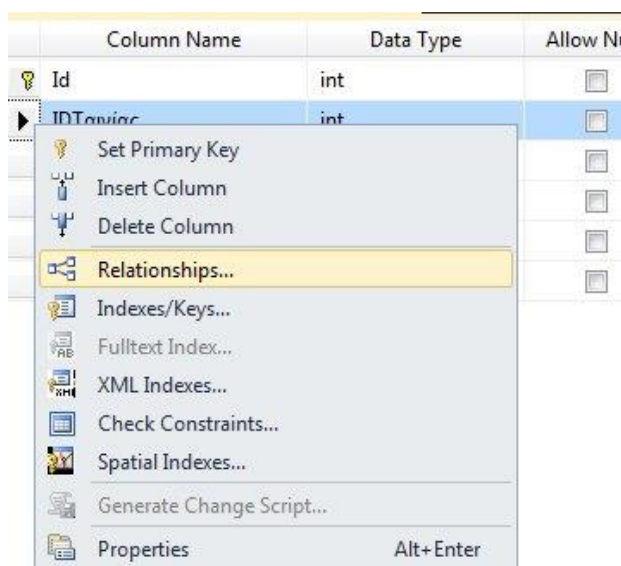
Η χρήση του ON DELETE ελέγχει την περίπτωση διαγραφής μιας εγγραφής του πίνακα ΤΑΙΝΙΑ. Υπάρχουν οι εξής επιλογές:

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

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

## 2.2.5. Συσχετίσεις/Relationships Πινάκων

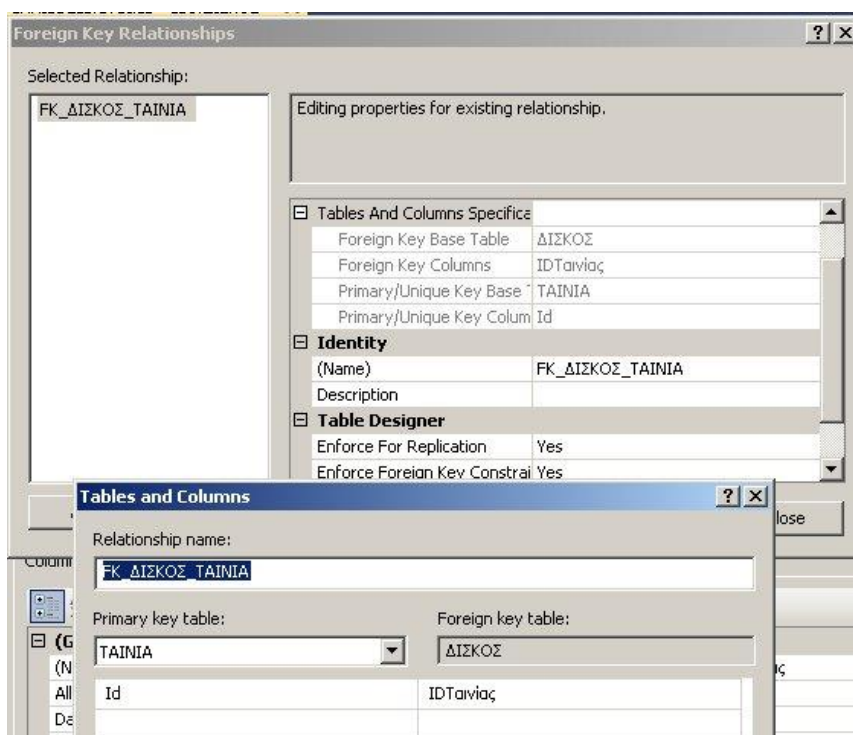
Στην Ενότητα 2.2.4 ο πίνακας ΔΙΣΚΟΣ συσχετίστηκε με τον πίνακα ΤΑΙΝΙΑ μέσω του πεδίου IDΤαινίας. Αν δεν είχαμε κάνει τη συσχέτιση των δύο πινάκων (της προηγούμενης ενότητας) αλλά είχαμε δημιουργήσει μόνο τους πίνακες, τότε, προκειμένου να δημιουργήσουμε την ίδια συσχέτιση με γραφικό τρόπο, πρέπει να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



Εικόνα 2.18

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

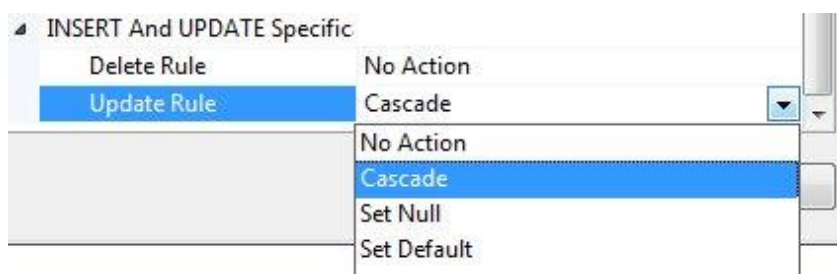
ΔΙΣΚΟΣ\_TAINIA. Στη συνέχεια κάνουμε κλικ στην επιλογή Tables and Columns Specification. Εμφανίζεται το αναδυόμενο παράθυρο της Εικόνας 2.19, στο οποίο δηλώνουμε τον κύριο πίνακα με το κύριο κλειδί του (TAINIA.ID), καθώς και τον εξαρτώμενο πίνακα με το ξένο κλειδί (ΔΙΣΚΟΣ.IDΤαινίας).



Εικόνα 2.19

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

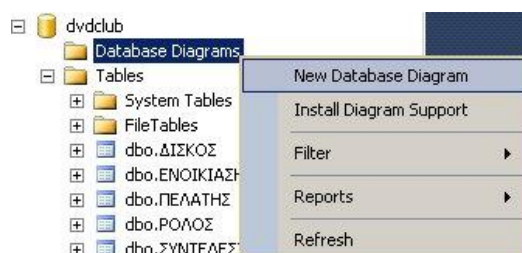
Τέλος, όπως φαίνεται στην Εικόνα 2.20, μπορούμε και με γραφικό τρόπο να κάνουμε χρήση του DELETE Rule, με επιλογές παρόμοιες με αυτές που αναφέρθηκαν στην Ενότητα 2.2.4 (π.χ. No Action, Cascade, Set Null, Set Default). Αντίστοιχα, η χρήση της ρύθμισης Update Rule αφορά την περίπτωση που γίνεται update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



Εικόνα 2.20

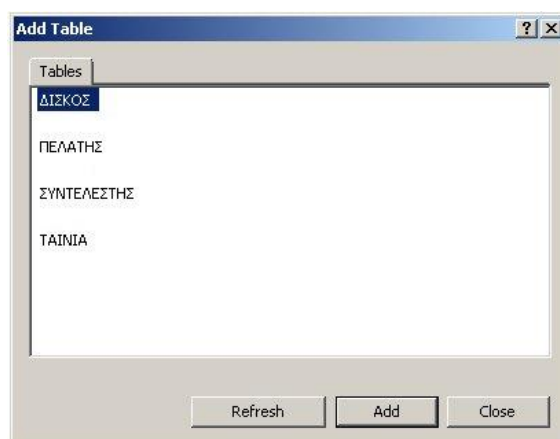
## 2.2.6. Δημιουργία πινάκων με τον Database Diagrams

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



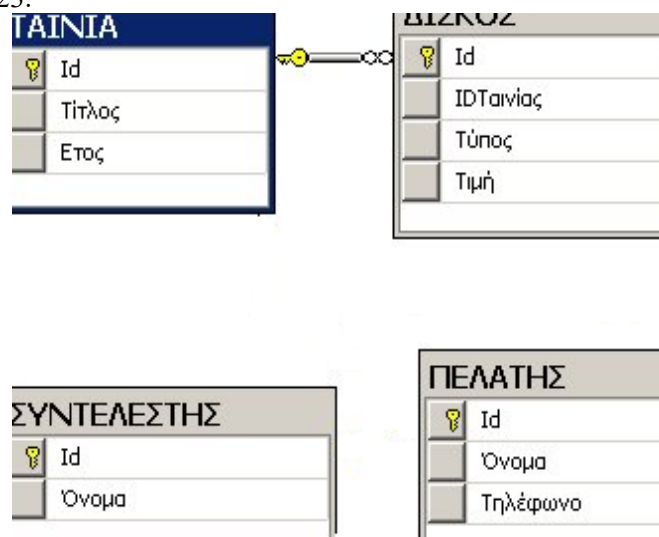
Εικόνα 2.21

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



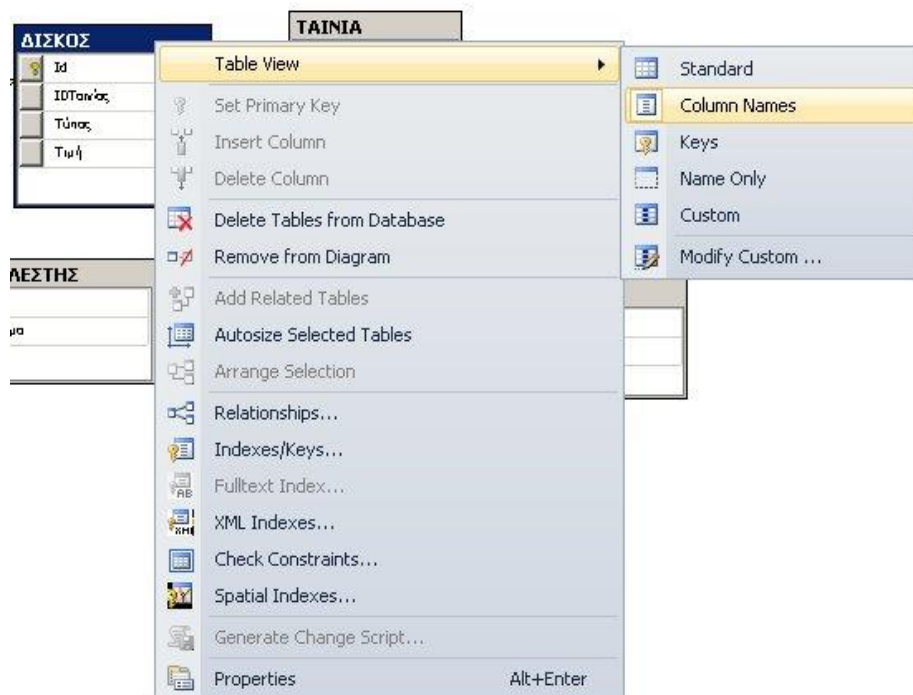
Εικόνα 2.22

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



Εικόνα 2.23

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



Εικόνα 2.24



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

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

Πίνακας 2.6 ΕΝΟΙΚΙΑΣΗ

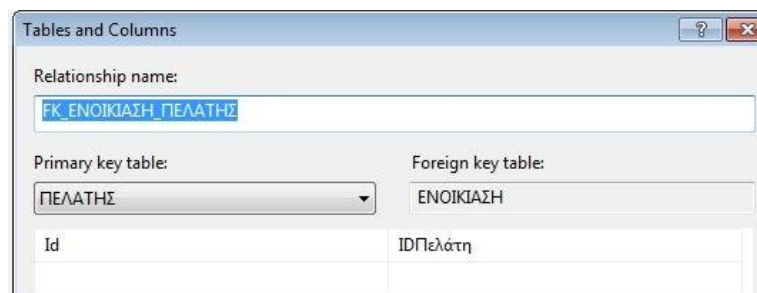
Στον πίνακα ΕΝΟΙΚΙΑΣΗ το κύριο κλειδί αποτελείται από περισσότερα του ενός πεδία (IDΠελάτη, IDΔίσκου). Σε αυτήν την περίπτωση, επιτρέπεται να υπάρχουν διπλές τιμές στο ίδιο πεδίο. Όμως, κάθε συνδυασμός των τιμών των δύο πεδίων που αποτελούν το κύριο κλειδί πρέπει να είναι μοναδικός. Μ' αυτόν τον τρόπο,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

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

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

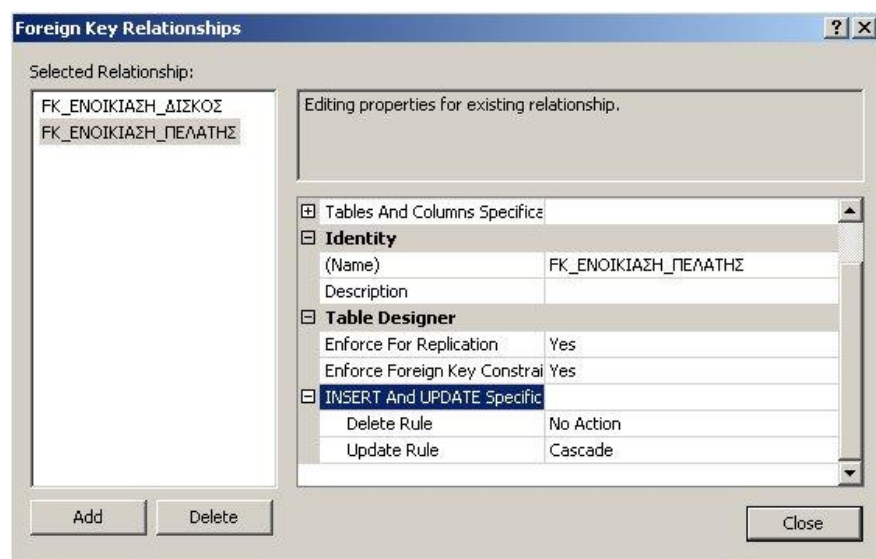
Εικόνα 2.25

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



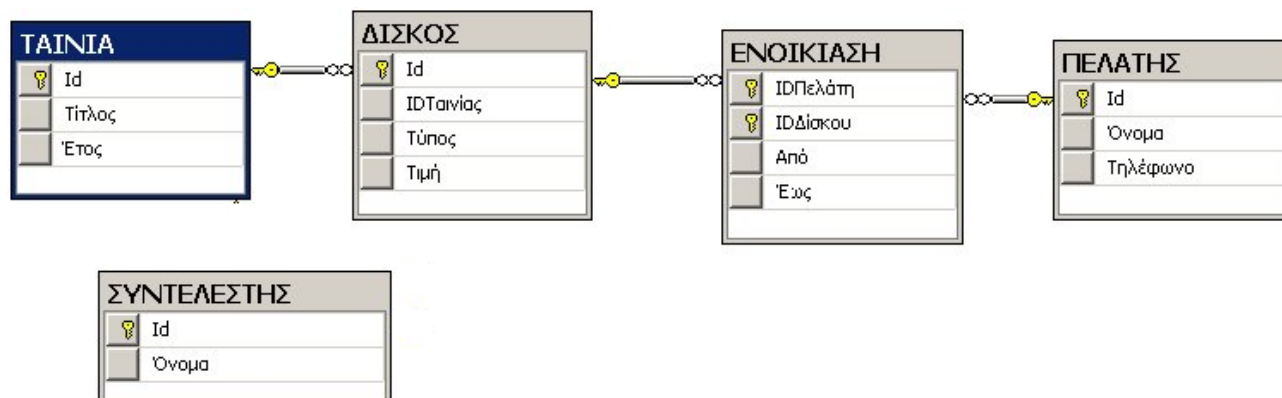
Εικόνα 2.26

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



Εικόνα 2.27

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



Εικόνα 2.28

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

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

Πίνακας 2.8 ΡΟΛΟΣ

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

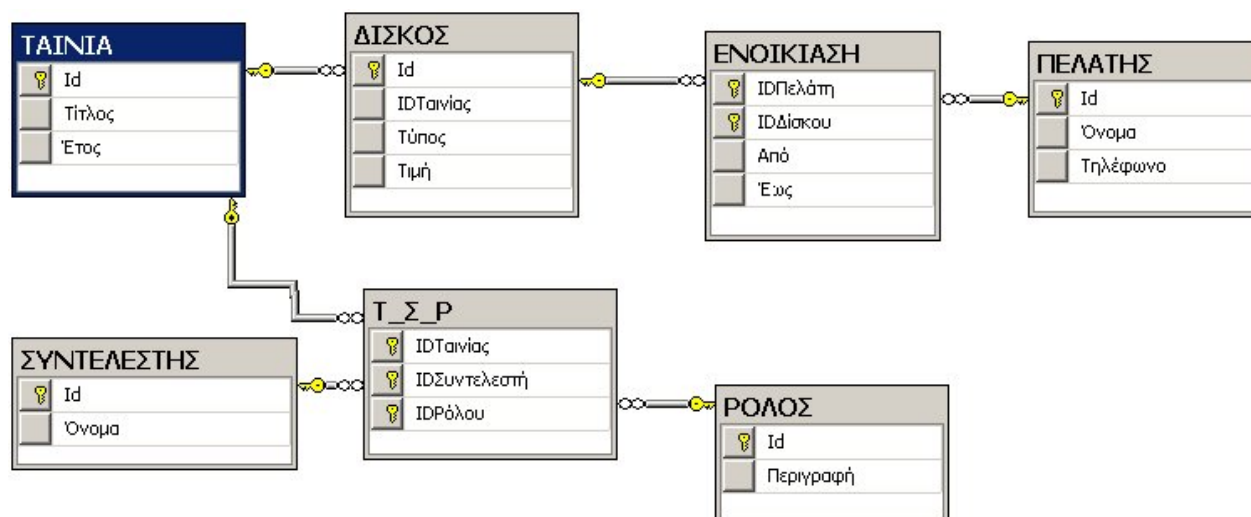
| Field        | Type | Null | Key |
|--------------|------|------|-----|
| IDΤαινίας    | Int  |      | PRI |
| IDΣυντελεστή | Int  |      | PRI |
| IDΡόλου      | Int  |      | PRI |

Πίνακας 2.9 T\_Σ\_P

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



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



Εικόνα 2.29

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

**ALTER TABLE ΠΕΛΑΤΗΣ**  
**ADD COLUMN** Ημερομηνία\_Γέννησης **DATE NOT NULL**

**ALTER TABLE ΠΕΛΑΤΗΣ**  
**DROP COLUMN** Ημερομηνία\_Γέννησης

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

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

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

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

## 2.3. Εισαγωγή εγγραφών στους πίνακες

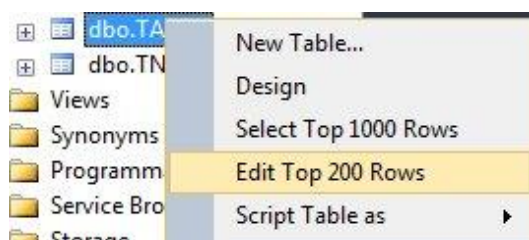
### 2.3.1. Εισαγωγή εγγραφών στους πίνακες με γραφικό τρόπο

Θα επιχειρήσουμε τώρα να εισάγουμε τις ακόλουθες εγγραφές στον πίνακα 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 DVDclub. Κάνουμε δεξί κλικ στον πίνακα TAINIA. Επιλέγουμε Edit Top 200 Rows, όπως φαίνεται στην Εικόνα 2.30.



Εικόνα 2.3

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

|   | 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 εγγραφές στον πίνακα ΣΥΝΤΕΛΕΣΤΗΣ:

| ID | Όνομα            |
|----|------------------|
| 1  | Alfred Hitchcock |
| 2  | Grace Kelly      |
| 3  | Anthony Perkins  |

Πίνακας 2.9 ΣΥΝΤΕΛΕΣΤΗΣ

Επίσης, για τον πίνακα ΡΟΛΟΣ εισάγουμε τις ακόλουθες εγγραφές:

| ID | Περιγραφή  |
|----|------------|
| 1  | Σκηνοθέτης |
| 2  | Ηθοποιός   |

Πίνακας 2.9 ΡΟΛΟΣ

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

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

Πίνακας 2.10 ΠΕΛΑΤΗΣ

Τέλος, επαναλαμβάνουμε τα προηγούμενα βήματα για να εισάγουμε τις ακόλουθες εγγραφές στον πίνακα T\_Σ\_P:

| IDΤαινίας | IDΣυντελεστή | IDΡόλου |
|-----------|--------------|---------|
| 1         | 1            | 1       |
| 2         | 1            | 2       |
| 1         | 2            | 1       |
| 2         | 3            | 2       |

Πίνακας 2.11 T\_Σ\_P

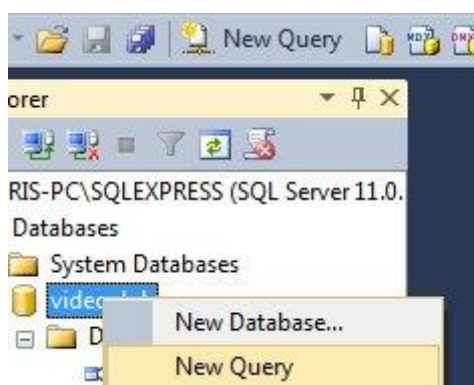
### 2.3.2. Εισαγωγή εγγραφών στους πίνακες με εντολές SQL

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

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

Πίνακας 2.12 ΔΙΣΚΟΣ

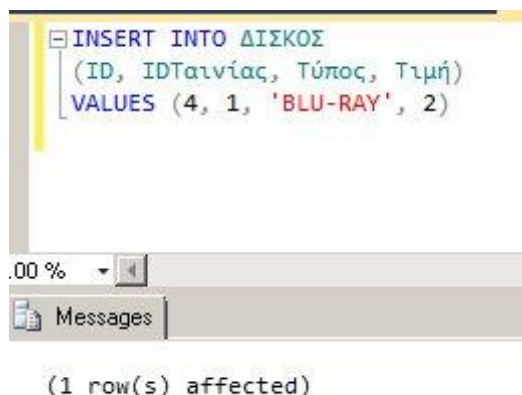
Μεταβαίνουμε στο Management Studio και επιλέγουμε από την μπάρα εργαλείων New Query. Η ίδια επιλογή μπορεί να γίνει με δεξί κλικ στην database DVDclub και κλικ στο New Query, όπως φαίνεται στην Εικόνα 2.32.



Εικόνα 2.32

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

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



Εικόνα 2.33

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

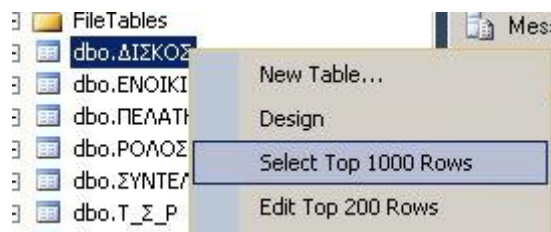
Η χρήση των ονομάτων των στηλών είναι προαιρετική. Πρέπει, όμως, να βάλουμε όλα τα ορίσματα με τη σωστή σειρά. Για παράδειγμα, η δεύτερη εγγραφή μπορεί να εισαχθεί ως εξής:

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

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

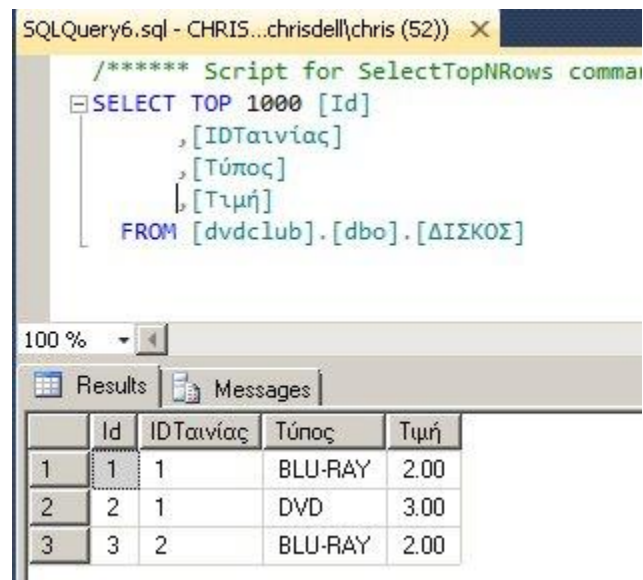
```
INSERT INTO ΔΙΣΚΟΣ  
(ID, IDΤαινίας, Τιμή, Τύπος)  
VALUES (3, 2, 2, 'BLU-RAY')  
GO
```

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



Εικόνα 2.34

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



Εικόνα 2.35

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

| IDΠελάτη | IDΔίσκου | Από        | Έως        |
|----------|----------|------------|------------|
| 1        | 1        | 07/10/2006 | 09/10/2006 |
| 1        | 2        | 09/20/2006 | 11/20/2006 |
| 2        | 1        | 09/10/2006 | Null       |
| 2        | 2        | 09/30/2006 | 09/30/2006 |

Πίνακας 2.13 ΕΝΟΙΚΙΑΣΗ

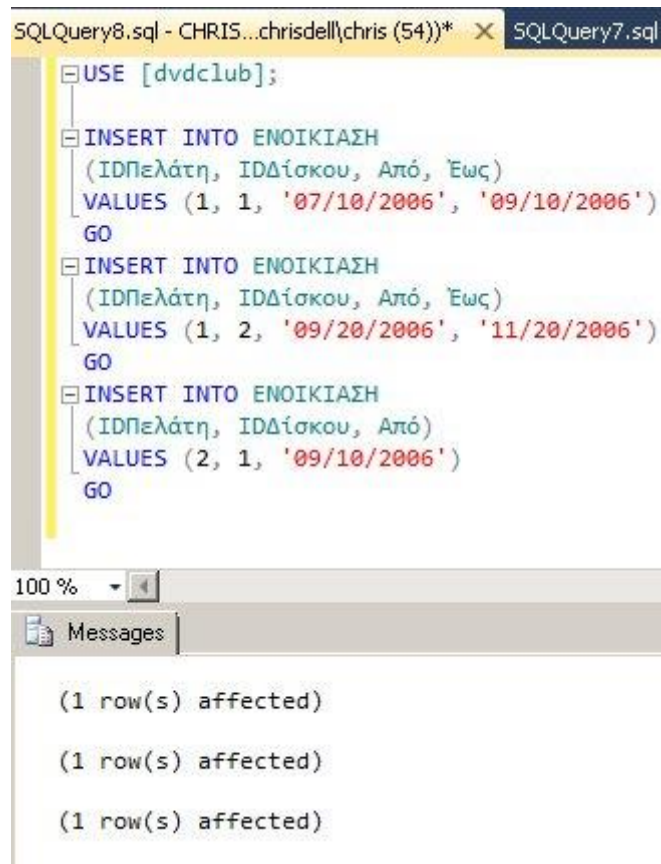
```
INSERT INTO ΕΝΟΙΚΙΑΣΗ
(IDΠελάτη, IDΔίσκου, Από, Έως)
VALUES (1, 1, '07/10/2006', '09/10/2006')
GO
```

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

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

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

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



Εικόνα 2.36

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

```
INSERT INTO ENOIKIASH  
(IDΠελάτη, IDΔίσκου, Από, Έως)  
VALUES (2, 2, '09/30/2006', '9/30/2006')  
GO
```

## 2.4. Αλλαγή σε δεδομένα πινάκων

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

### 2.4.1. Ενημέρωση δεδομένων

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

```
UPDATE όνομα_πίνακα  
SET πεδίο=νέα_τιμή  
WHERE κριτήρια
```



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

```
UPDATE ΕΝΟΙΚΙΑΣΗ  
SET Έως = '09/30/2006'  
WHERE IDΔίσκου = 2 and IDΠελάτη = 2
```

#### 2.4.2. Διαγραφή δεδομένων

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

```
DELETE FROM όνομα_πίνακα  
WHERE κριτήρια
```

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

```
DELETE FROM ΕΝΟΙΚΙΑΣΗ  
WHERE IDΔίσκου = 2 and IDΠελάτη = 2
```

## 2.5. Κώδικας SQL για τη δημιουργία της βάσης δεδομένων DVDclub

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

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

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

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

```

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

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

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

```

```

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[T_Σ_P]      Script Date: 11/7/2015 5:23:16 μμ
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[T_Σ_P](
    [IDΤαινίας] [int] NOT NULL,
    [IDΣυντελεστή] [int] NOT NULL,
    [IDΡόλου] [int] NOT NULL,
    CONSTRAINT [PK_TΣ2] PRIMARY KEY CLUSTERED
(
    [IDΤαινίας] ASC,
    [IDΣυντελεστή] ASC,
    [IDΡόλου] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

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

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

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

```

```

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

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

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

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

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

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

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

```

```

ALTER TABLE [dbo].[ΕΝΟΙΚΙΑΣΗ] WITH CHECK ADD CONSTRAINT
[FK_ΕΝΟΙΚΙΑΣΗ_ΔΙΣΚΟΣ] FOREIGN KEY ([IDΔίσκου])
REFERENCES [dbo].[ΔΙΣΚΟΣ] ([Id])
ON UPDATE CASCADE
GO
ALTER TABLE [dbo].[ΕΝΟΙΚΙΑΣΗ] CHECK CONSTRAINT [FK_ΕΝΟΙΚΙΑΣΗ_ΔΙΣΚΟΣ]
GO
ALTER TABLE [dbo].[ΕΝΟΙΚΙΑΣΗ] WITH CHECK ADD CONSTRAINT
[FK_ΕΝΟΙΚΙΑΣΗ_ΠΕΛΑΤΗΣ] FOREIGN KEY ([IDΠελάτη])
REFERENCES [dbo].[ΠΕΛΑΤΗΣ] ([Id])
ON UPDATE CASCADE
GO
ALTER TABLE [dbo].[ΕΝΟΙΚΙΑΣΗ] CHECK CONSTRAINT [FK_ΕΝΟΙΚΙΑΣΗ_ΠΕΛΑΤΗΣ]
GO
ALTER TABLE [dbo].[Τ_Σ_P] WITH CHECK ADD CONSTRAINT [FK_Τ_Σ_P_ΡΟΛΟΣ]
FOREIGN KEY ([IDΤαινίας])
REFERENCES [dbo].[ΡΟΛΟΣ] ([Id])
GO
ALTER TABLE [dbo].[Τ_Σ_P] CHECK CONSTRAINT [FK_Τ_Σ_P_ΡΟΛΟΣ]
GO
ALTER TABLE [dbo].[Τ_Σ_P] WITH CHECK ADD CONSTRAINT [FK_Τ_Σ_P_ΣΥΝΤΕΛΕΣΤΗΣ]
FOREIGN KEY ([IDΣυντελεστή])
REFERENCES [dbo].[ΣΥΝΤΕΛΕΣΤΗΣ] ([Id])
ON UPDATE CASCADE
GO
ALTER TABLE [dbo].[Τ_Σ_P] CHECK CONSTRAINT [FK_Τ_Σ_P_ΣΥΝΤΕΛΕΣΤΗΣ]
GO
ALTER TABLE [dbo].[Τ_Σ_P] WITH CHECK ADD CONSTRAINT [FK_Τ_Σ_P_ΤΑΙΝΙΑ]
FOREIGN KEY ([IDΤαινίας])
REFERENCES [dbo].[ΤΑΙΝΙΑ] ([Id])
ON UPDATE CASCADE
GO
ALTER TABLE [dbo].[Τ_Σ_P] CHECK CONSTRAINT [FK_Τ_Σ_P_ΤΑΙΝΙΑ]
GO

```



## 2.6. Ασκήσεις

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

**ΠΕΛΑΤΕΣ (Πελάτης\_Id, Όνομα, Περιοχή\_Id)**

**ΠΕΡΙΟΧΕΣ (Περιοχή\_Id, Περιγραφή)**

**ΑΚΙΝΗΤΑ(Ακίνητο\_Id, Περιοχή\_Id, Τύπος, Τετραγωνικά, τιμή\_εκκίνησης)**

**ΑΓΟΡΑΠΩΛΗΣΙΕΣ(Πελάτης\_id, Ακίνητο\_id, Ημερομηνία, Ποσό\_αγοράς)**

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

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

Πίνακας 2.14 ΠΕΛΑΤΕΣ

| Περιοχή_id | Περιγραφή |
|------------|-----------|
| 1          | KALAMARIA |
| 2          | KRINI     |
| 3          | PILEA     |

Πίνακας 2.15 ΠΕΡΙΟΧΕΣ

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

| Ακίνητο_id | Περιοχή_id | Τύπος      | Τετραγωνικά | Τιμή εκκίνησης |
|------------|------------|------------|-------------|----------------|
| 1          | 1          | DIAMERISMA | 100         | 9000           |
| 2          | 2          | KATASTIMA  | 150         | 14000          |
| 3          | 3          | MEZONETA   | 200         | 19000          |

Πίνακας 2.16 ΑΚΙΝΗΤΑ

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

Πίνακας 2.16 ΑΓΟΡΑΠΩΛΗΣΙΕΣ

4. Χρησιμοποιώντας την εντολή UPDATE της SQL να αλλάξετε την ημερομηνία αγοράς από 1/17/1990 σε 7/11/2015 στην εγγραφή ου πίνακα ΑΓΟΡΑΠΩΛΗΣΙΕΣ που αφορά το Ακίνητο με κωδικό (Ακίνητο\_id = 1), το οποίο αγοράστηκε από τον πελάτη με κωδικό (Πελάτης\_id = 2).
5. Χρησιμοποιώντας της εντολή DELETE της SQL να διαγράψετε την εγγραφή του πίνακα ΑΓΟΡΑΠΩΛΗΣΙΕΣ που αφορά το ακίνητο με κωδικό (Ακίνητο\_id = 2), το οποίο αγοράστηκε από τον πελάτη με κωδικό (IDΠελάτη=2).

## 2.7. Βιβλιογραφία/Αναφορές

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

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

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