

Κεφάλαιο 3. Ερωτήματα SQL

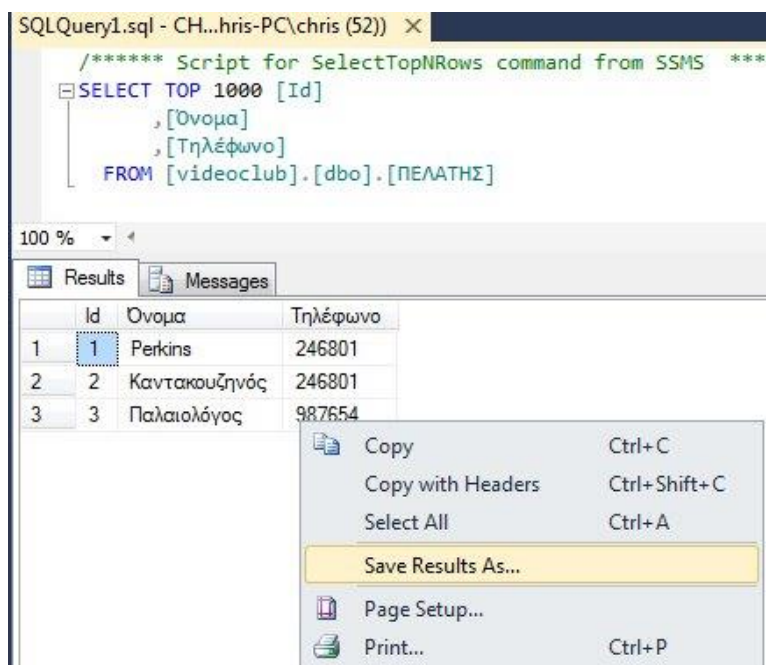
Σύνοψη

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

3.1. Βασικά Ερωτήματα

3.1.1. Διαχείριση του Results Pane

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



Εικόνα 3.1

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

3.1.2. Ερωτήματα επιλογής εγγραφών από έναν πίνακα.

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

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

Η σύνταξη της εντολής περιλαμβάνει τρεις όρους:

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

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

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

```


|    |                                            |
|----|--------------------------------------------|
| Q1 | <b>Select</b> Όνομα<br><b>From</b> ΠΕΛΑΤΗΣ |
|----|--------------------------------------------|


```

Το αποτέλεσμα του ερωτήματος είναι το παρακάτω:

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

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

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

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

Q2

```
Select Τηλέφωνο  
From ΠΕΛΑΤΗΣ
```

Το αποτέλεσμα του ερωτήματος είναι το παρακάτω:

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

Αν δύο πελάτες έχουν το ίδιο τηλέφωνο (π.χ., είναι 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 εγγραφών, όπως δηλαδή συμβαίνει με την εφαρμογή της πράξης της προβολής στη σχεσιακή άλγεβρα. Για παράδειγμα, έστω το ερώτημα:

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

Q3

```
Select distinct Τηλέφωνο  
From ΠΕΛΑΤΗΣ
```

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

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

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

Q4

```
Select *  
From ΠΕΛΑΤΗΣ
```

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

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

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

Q5

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

Όνομα	Τηλέφωνο
Perkins	2310246801
Καντακουζηνός	2310246801
Παλαιολόγος	2310987654

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

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

Q6

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

ID
1
3

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

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

Q7

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

ID
1
2
3

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

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

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

Q8 **Select** Όνομα
From ΠΕΛΑΤΗΣ
Where Όνομα like 'Κ%'

Όνομα

 Καντακουζηνός

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

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

Q9 **Select** *
From ΕΝΟΙΚΙΑΣΗ
Where Έως IS NULL

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

3.1.3. Ταξινόμηση αποτελεσμάτων

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

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

Q10 **Select** ID, Τιμή
From ΔΙΣΚΟΣ
Order by Τιμή

ID	Τιμή
-----	-----
1	2.00
3	2.00
2	3.00

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

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

Q11 **Select** Τιμή, ID
From ΔΙΣΚΟΣ
Order by Τιμή desc, ID asc

Τιμή	ID
3.00	2
2.00	1
2.00	3

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

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

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

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

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

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

Q13 **Select top 2** ID
From ΔΙΣΚΟΣ
Order by Τιμή

ID
1
3

3.2. Ερωτήματα επιλογής εγγραφών από πολλούς πίνακες

3.2.1. Εσωτερική και εξωτερική σύνδεση πινάκων

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

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

Q14

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

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

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

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

Q15

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

IDΤαινίας
1
2

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

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

```

Q16 Select IDΤαινίας
From T_Σ_P inner join ΣΥΝΤΕΛΕΣΤΗΣ on
T_Σ_P.IDΣυντελεστή = ΣΥΝΤΕΛΕΣΤΗΣ.ID
Where ΣΥΝΤΕΛΕΣΤΗΣ.Όνομα = 'Alfred Hitchcock'

```

IDΤαινίας
1
2

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

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

```

Q17 Select ΠΕΛΑΤΗΣ.Όνομα, ΔΙΣΚΟΣ.ID, ΔΙΣΚΟΣ.Τιμή
From ΠΕΛΑΤΗΣ inner join ΕΝΟΙΚΙΑΣΗ on
ΠΕΛΑΤΗΣ.ID = ΕΝΟΙΚΙΑΣΗ.IDΠελάτη inner join ΔΙΣΚΟΣ on
ΕΝΟΙΚΙΑΣΗ.IDΔίσκου = ΔΙΣΚΟΣ.ID

```

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

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

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

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

Q18

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

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

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

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

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

Q19

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

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

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

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

Q20

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

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

3.2.2. Μετονομασία και αυτό-σύνδεση

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

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

Q21

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

Όνομα	Τηλέφωνο
Perkins	2310246801
Καντακουζηνός	2310246801
Παλαιολόγος	2310987654

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

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

Q22

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

Όνομα

Καντακουζηνός

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

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

Q23

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

IDΤαινίας

1

3.3. Ερωτήματα συνάθροισης και ομαδοποίησης

3.3.1. Ερωτήματα με συναρτήσεις συνάθροισης

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

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

Πίνακας 3.14

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

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

```
Q24  Select max(Τιμή) as 'Μέγιστη Τιμή Δίσκου'  
      From ΔΙΣΚΟΣ
```

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

Ένα δεύτερο παράδειγμα ερωτήματος είναι το παρακάτω :

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

```
Q25  Select count(*)  
      From ΔΙΣΚΟΣ
```

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

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

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

```
Q26  Select max(Τιμή) - min(Τιμή)  
      From ΔΙΣΚΟΣ
```

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

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

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

```
Q27 Select count(distinct IDΠελάτη)
      From ΕΝΟΙΚΙΑΣΗ
```

2

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

3.3.2 Ομαδοποίηση των δεδομένων - Ο όρος `Group by`

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

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

```
Q28 Select Τύπος, avg(Τιμή) as 'Μέση Τιμή'
      From ΔΙΣΚΟΣ
      Group by Τύπος
```

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

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

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

```
Q29 Select IDΠελάτη, IDΔίσκου, count(IDΔίσκου) as 'Αριθμός Ενοικιάσεων'
      From ΕΝΟΙΚΙΑΣΗ
      Group by IDΠελάτη, IDΔίσκου
```

IDΠελάτη	IDΔίσκου	Αριθμός Ενοικιάσεων
-----	-----	-----
1	1	1
1	2	1
2	1	1

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

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

Q30

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

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

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

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

Q31

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

ID	Αριθμός Ενοικιάσεων
1	2
2	1
3	0

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

3.3.3. Ο όρος Having

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

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

Q32

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

Τύπος	Μέση Τιμή Ενοικίασης
DVD	3,000000

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

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

Q33

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

IDΣυντελεστή
1

3.4. Ερωτήματα με πράξεις συνόλων και εμφωλευμένα ερωτήματα

3.4.1. Βασικές πράξεις

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

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

Q34

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

union

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

Τίτλος

Ben-Hur

Psycho

Rear Window

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

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

Q35

```
Select Τίτλος  
From TAINIA  
Where Τίτλος like ‘_e%’
```

intersect

```
Select Τίτλος  
From TAINIA  
Where Χρονιά = ‘1954’
```

Τίτλος

Rear Window

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

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

Q36

```
Select Τίτλος
From TAINIA
Where Τίτλος like ‘_e%’
```

except

```
Select Τίτλος
From TAINIA
Where Χρονιά = ‘1954’
```

Τίτλος

Ben-Hur

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

3.4.2. Εμφωλευμένα ερωτήματα

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

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

Q37

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

Όνομα

Perkins
Καντακουζηνός

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

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

Q38 **Select** Όνομα
From ΣΥΝΤΕΛΕΣΤΗΣ
Where Όνομα **not in** ('Alfred Hitchcock', 'Grace Kelly')

Όνομα

 Anthony Perkins

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

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

Q39 **Select** ΔΙΣΚΟΣ.IDΤαινίας
From ΔΙΣΚΟΣ **inner join** ΕΝΟΙΚΙΑΣΗ **on**
 ΕΝΟΙΚΙΑΣΗ.IDΔίσκου = ΔΙΣΚΟΣ.ID **Where** ΔΙΣΚΟΣ.IDΤαινίας **IN**
 (**Select** IDΤαινίας **From** T_Σ_P **inner join** ΣΥΝΤΕΛΕΣΤΗΣ **on**
 T_Σ_P.IDΣυντελεστή = ΣΥΝΤΕΛΕΣΤΗΣ.ID
where ΣΥΝΤΕΛΕΣΤΗΣ.Όνομα = 'Alfred Hitchcock')
Group by ΔΙΣΚΟΣ.IDΤαινίας
Having count(*) > 2

IDΤαινίας

 1

3.4.3. Σύγκριση μεταξύ συνόλων

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

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

Q40 **Select** Όνομα
From ΠΕΛΑΤΗΣ
Where ID = **some**
 (**Select** IDΠελάτη
From ΕΝΟΙΚΙΑΣΗ
where Έως IS NULL)

Όνομα

 Καντακουζηνός

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

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

Q41

```

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

```

```

IDΠελάτη
-----
1

```

3.4.4. Έλεγχος κενότητας

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

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

Q42

```

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

```

```

Όνομα
-----
Perkins
Καντακουζηνός

```

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

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

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

Q43

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

IDΠελάτη

1

3.5. Ερωτήματα SQL για όψεις

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

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

```
Q44 Create View MyView as
      (Select T_Σ_P.IDΤαινίας
       from ΣΥΝΤΕΛΕΣΤΗΣ inner join T_Σ_P ON
        ΣΥΝΤΕΛΕΣΤΗΣ.Id = T_Σ_P.IDΣυντελεστή
        where ΣΥΝΤΕΛΕΣΤΗΣ.Όνομα = 'Alfred Hitchcock')
```

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

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

```
Q45 drop view MyView
```

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

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

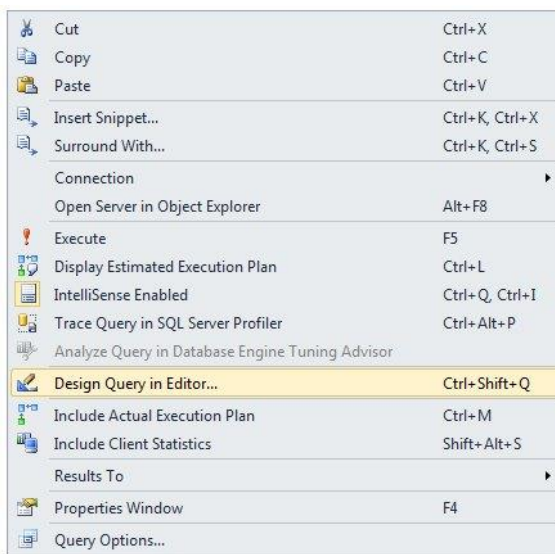
```
Q46 Select ΔΙΣΚΟΣ.IDΤαινίας
      From ΔΙΣΚΟΣ inner join ΕΝΟΙΚΙΑΣΗ on ΕΝΟΙΚΙΑΣΗ.IDΔίσκου = ΔΙΣΚΟΣ.ID Where
      ΔΙΣΚΟΣ.IDΤαινίας IN (select * from MyView)
      Group by ΔΙΣΚΟΣ.IDΤαινίας
      Having count(*) > 2
```

IDΤαινίας

1

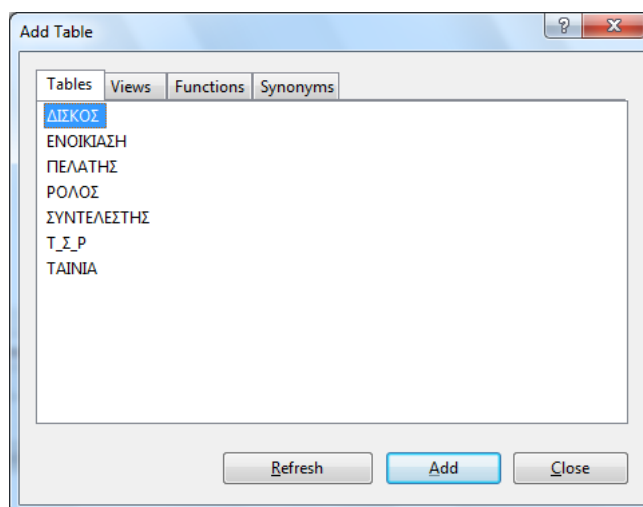
3.6. Το εργαλείο Query Designer για δημιουργία ερωτημάτων Query by Example

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



Εικόνα 3.2

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



Εικόνα 3.3

Στο παράδειγμά μας θα υλοποιήσουμε το παρακάτω ερώτημα:

Q47

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

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

Query Designer

ΣΥΝΤΕΛΕΣΤΗΣ

- * (All Columns)
- Id
- Όνομα

Τ_Σ_Ρ

- * (All Columns)
- IDΤαινίας
- IDΣυντελεστή
- IDΡόλου

ΤΑΙΝΙΑ

- * (All Columns)
- Id
- Τίτλος
- Έτος

Column	Alias	Table	Output	Sort Type	Sort Order	Filter	Or...
Id		TAINIA	<input checked="" type="checkbox"/>				
Τίτλος		TAINIA	<input checked="" type="checkbox"/>				
Έτος		TAINIA	<input checked="" type="checkbox"/>				
Όνομα		ΣΥΝΤΕΛΕΣ...	<input type="checkbox"/>			= 'Alfred Hitchcock'	

```
SELECT TAINIA.Id, TAINIA.Τίτλος, TAINIA.Έτος
FROM ΣΥΝΤΕΛΕΣΤΗΣ INNER JOIN
  Τ_Σ_Ρ ON ΣΥΝΤΕΛΕΣΤΗΣ.Id = Τ_Σ_Ρ.IDΣυντελεστή INNER JOIN
  ΤΑΙΝΙΑ ON Τ_Σ_Ρ.IDΤαινίας = ΤΑΙΝΙΑ.Id
WHERE (ΣΥΝΤΕΛΕΣΤΗΣ.Όνομα = 'Alfred Hitchcock')
```

11:53 10/12/2010

Εικόνα 3.4

Παρακάτω περιγράφουμε τις βασικές στήλες του grid:

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

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

ID	Τίτλος	Έτος
1	Rear Window	1954
2	Psycho	1960

(2 row(s) affected)

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

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

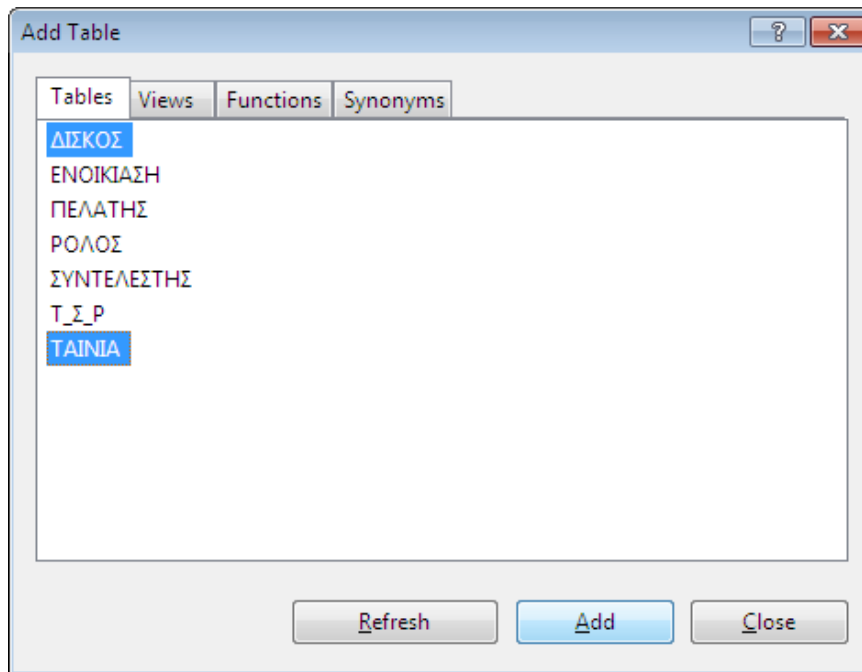
Τέλος, παρακάτω παρουσιάζουμε ένα παράδειγμα δημιουργίας ερωτήματος ομαδοποίησης με QBE:

Q48

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

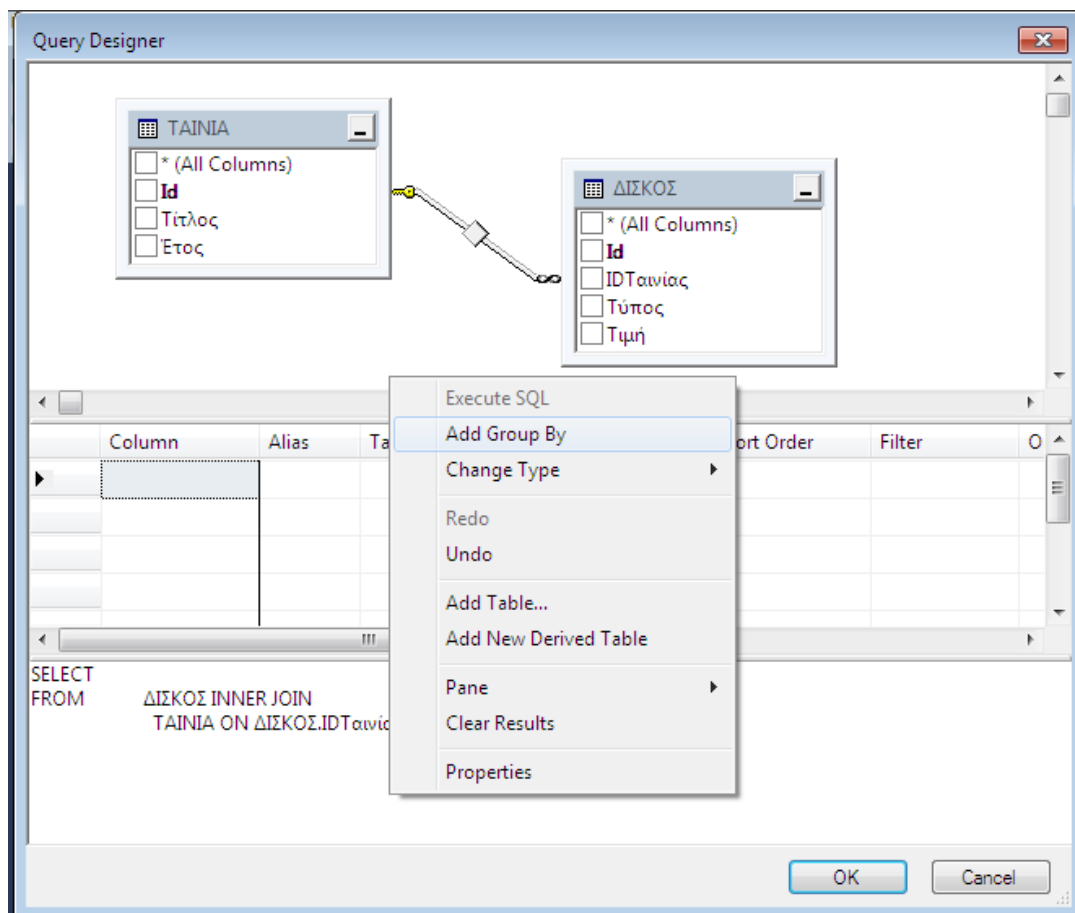
Τίτλος	Τύπος	Αριθμός
Psycho	BLU-RAY	1
Rear Window	BLU-RAY	1
Rear Window	DVD	1

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



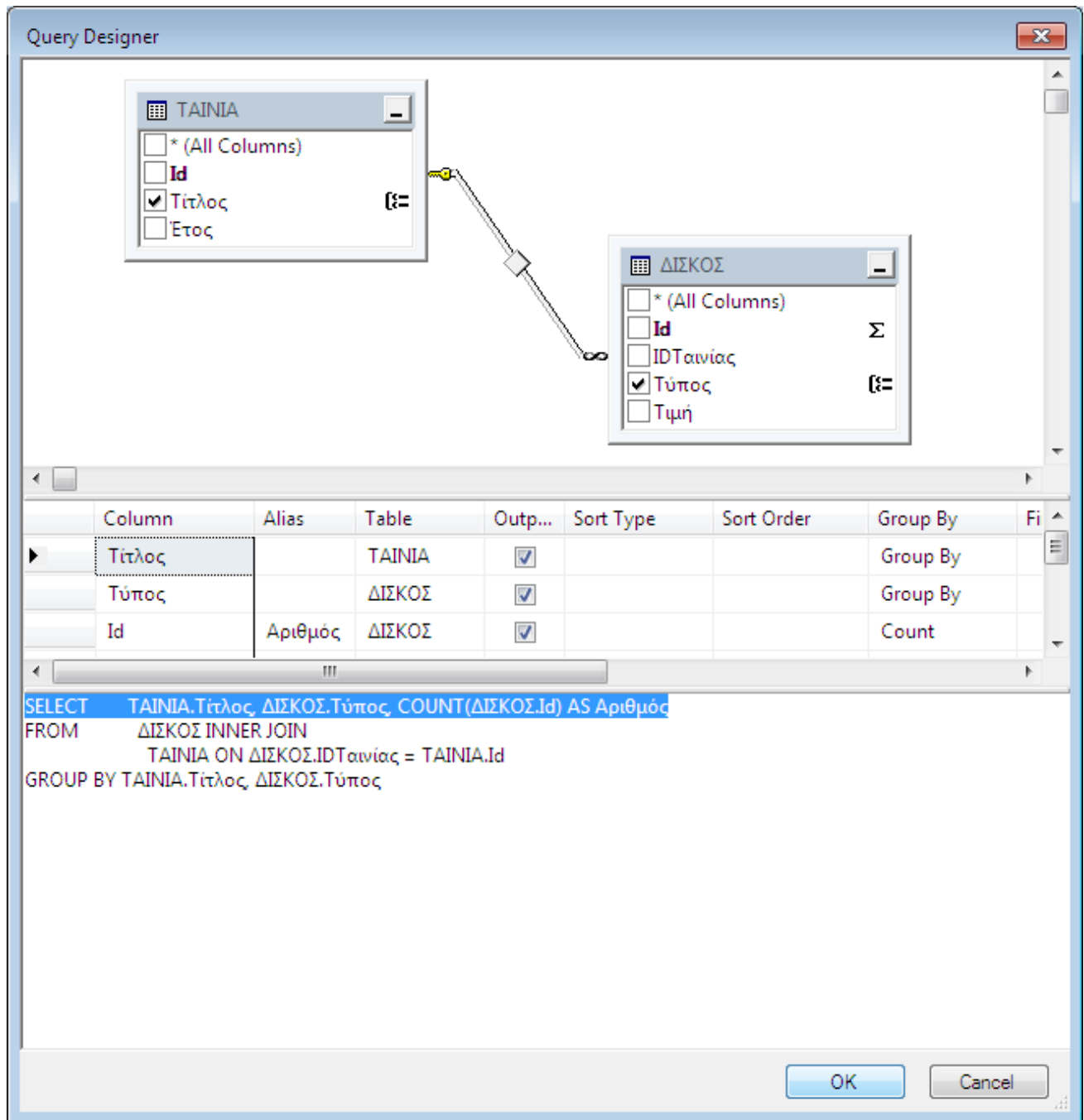
Εικόνα 3.5

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



Εικόνα 3.6

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



Εικόνα 3.7

3.7. Ασκήσεις με ερωτήματα SQL

3.7.1. Ασκήσεις με ερωτήματα επιλογής γραμμών από ένα πίνακα

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

3.7.2. Ασκήσεις με ερωτήματα επιλογής γραμμών από πολλούς πίνακες

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

3.7.3. Ασκήσεις με Ερωτήματα ομαδοποίησης/συνάθροισης δεδομένων

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

3.7.4. Ασκήσεις με ερωτήματα με φωλιασμένες εντολές SQL

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

3.8. Λύσεις ασκήσεων με ερωτήματα SQL

3.8.1. Λύσεις ασκήσεων με ερωτήματα επιλογής γραμμών από ένα πίνακα

1. Για κάθε δίσκο, να προβληθεί ο κωδικός και η τιμή. Η τιμή να εμφανίζεται χωρίς το ΦΠΑ, δηλαδή μειωμένη κατά $0.23 \cdot \text{Τιμή}$.

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

ID	Αποφορολογημένη Τιμή
1	1.626016
2	2.439024
3	1.626016

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

```
Select distinct IDΣυντελεστή  
from Τ_Σ_Ρ
```

IDΣυντελεστή
1
2
3

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

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

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

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

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

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

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

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

```
IDΔίσκου  
-----  
1  
2
```

3.8.2. Λύσεις ασκήσεων με ερωτήματα επιλογής γραμμών από πολλούς πίνακες

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

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

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

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

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

```
Όνομα  
-----  
Παλαιολόγος
```

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

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

```
IDΣυντελεστή  
-----  
1
```

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

```

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

```

Τίτλος
-----
Psycho
Ben-Hur
    
```

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

```

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

```

Όνομα
-----
Perkins
    
```

3.8.3. Λύσεις ασκήσεων με ερωτήματα ομαδοποίησης/συνάθροισης δεδομένων

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

```

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

```

Αριθμός Ταινιών
-----
2
    
```

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

```

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

```

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

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

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

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

3.8.4. Λύσεις ασκήσεων με ερωτήματα με φωλιασμένες εντολές SQL

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

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

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

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

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

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

3.9. Βιβλιογραφία/Αναφορές

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

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