

# Βάσεις Δεδομένων

## Εργαστήριο IV

Τμήμα Πληροφορικής ΑΠΘ  
2013-2014

## Σκοπός του 4<sup>ου</sup> εργαστηρίου

- Σκοπός αυτού του εργαστηρίου είναι:
  - η μελέτη ερωτημάτων σύνδεσης
  - η μελέτη ερωτημάτων συνάθροισης

## Εκφράσεις Σύνδεσης (1/3)

- Οι εκφράσεις σύνδεσης δέχονται ως είσοδο δύο σχέσεις και επιστρέφουν ως αποτέλεσμα μία άλλη σχέση.
- Η συνθήκη σύνδεσης **JOIN**:
  - καθορίζει ποιες πλειάδες ανάμεσα στις σχέσεις σύνδεσης ταιριάζουν και βάσει ποιων χαρακτηριστικών (**natural, on, using**).
- Ο τύπος σύνδεσης **JOIN**:
  - Καθορίζει τον τρόπο με τον οποίο θα χειριστούμε τις πλειάδες της μίας σχέσης, οι οποίες δεν ταιριάζουν με κάποια πλειάδα της άλλης σχέσης σύμφωνα με τη συνθήκη JOIN (**INNER JOIN, LEFT OUTER JOIN, RIGHT OUTER JOIN, FULL OUTER JOIN**).

## Εκφράσεις Σύνδεσης(2/3)

- Οι εκφράσεις σύνδεσης χρησιμοποιούνται είτε ως υποερωτήματα στο τμήμα FROM.
  - Είτε έμμεσα (δηλαδή χωρίς τη λέξη JOIN) στο τμήμα WHERE, όπως είδαμε στο προηγούμενο μάθημα

```
SELECT ...  
FROM R1,R2  
WHERE R1.ID= R2.ID
```



```
SELECT ...  
FROM R1 INNER JOIN R2 ON  
R1.ID= R2.ID
```

## Εκφράσεις Σύνδεσης(3/3)

- Η εντολή **INNER JOIN** τοποθετείται μόνο μεταξύ δύο σχέσεων οι οποίες μπορούν να συνδεθούν βάσει κάποιου χαρακτηριστικού.
- Έστω ότι έχουμε τις σχέσεις  $\rightarrow R1(\underline{a},b), R2(\underline{b},c) R3(\underline{c},d)$

```
SELECT ...  
FROM R2 INNER JOIN R3 ON ...  
INNER JOIN R1 ON ....
```

ΛΑΘΟΣ



```
SELECT ...  
FROM R1 INNER JOIN R2 ON R1.b= R2.b  
INNER JOIN R3 ON R2.c= R3.c
```

ΣΩΣΤΟ

## Παράδειγμα σύνδεσης δύο σχέσεων

- Να δοθούν οι κωδικοί των ταινιών όπου συμμετείχε ο 'Alfred Hitchcock'.

```
SELECT IDΤαινίας  
FROM ΤΣ, ΣΥΝΤΕΛΕΣΤΗΣ  
WHERE ΤΣ.IDΣυντελεστή = ΣΥΝΤΕΛΕΣΤΗΣ.ID AND  
      ΣΥΝΤΕΛΕΣΤΗΣ.Όνομα = 'Alfred Hitchcock'
```



```
SELECT IDΤαινίας  
FROM ΤΣ INNER JOIN ΣΥΝΤΕΛΕΣΤΗΣ ON  
ΤΣ.IDΣυντελεστή = ΣΥΝΤΕΛΕΣΤΗΣ.ID  
WHERE ΣΥΝΤΕΛΕΣΤΗΣ.Όνομα = 'Alfred Hitchcock'
```

## Σύνδεση τριών ή περισσότερων σχέσεων

- Η σύνδεση γίνεται συνδέοντας διαδοχικά την 1<sup>η</sup> με τη 2<sup>η</sup> σχέση, τη 2<sup>η</sup> με την 3<sup>η</sup>, την 3<sup>η</sup> με την 4<sup>η</sup>, κτλ.

```
SELECT ...  
FROM R1,R2, R3,..., Rm  
WHERE R1.ID= R2.ID AND R2.ID= R3.ID ...  
AND Rm-1.ID = Rm.ID
```

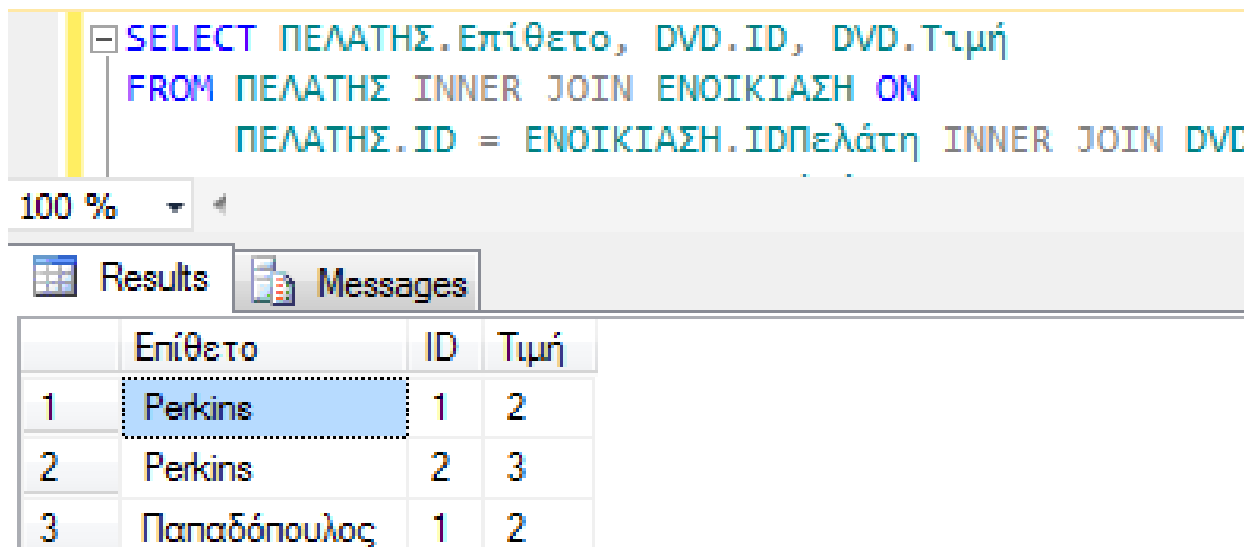


```
SELECT ...  
FROM R1 INNER JOIN R2 ON R1.ID= R2.ID INNER JOIN R3 ON R2.ID= R3.ID  
INNER JOIN .... ON Rm-1.ID = Rm.ID
```

## Ερώτημα με σύνδεση τριών ή περισσότερων σχέσεων στον SQL Server 2012

- Να βρεθούν για κάθε πελάτη (Επίθετο), ο κωδικός και η τιμή των dvd που έχει ενοικιάσει:

```
SELECT ΠΕΛΑΤΗΣ.Επίθετο, DVD.ID, DVD.Τιμή  
FROM ΠΕΛΑΤΗΣ INNER JOIN ΕΝΟΙΚΙΑΣΗ ON  
ΠΕΛΑΤΗΣ.ID = ΕΝΟΙΚΙΑΣΗ.IDΠελάτη INNER JOIN DVD  
ON ΕΝΟΙΚΙΑΣΗ.IDdvd = DVD.ID
```



The screenshot shows a SQL query window with the following text:

```
SELECT ΠΕΛΑΤΗΣ.Επίθετο, DVD.ID, DVD.Τιμή  
FROM ΠΕΛΑΤΗΣ INNER JOIN ΕΝΟΙΚΙΑΣΗ ON  
ΠΕΛΑΤΗΣ.ID = ΕΝΟΙΚΙΑΣΗ.IDΠελάτη INNER JOIN DVD  
ON ΕΝΟΙΚΙΑΣΗ.IDdvd = DVD.ID
```

Below the query window, there are tabs for 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with the following data:

	Επίθετο	ID	Τιμή
1	Perkins	1	2
2	Perkins	2	3
3	Παπαδόπουλος	1	2



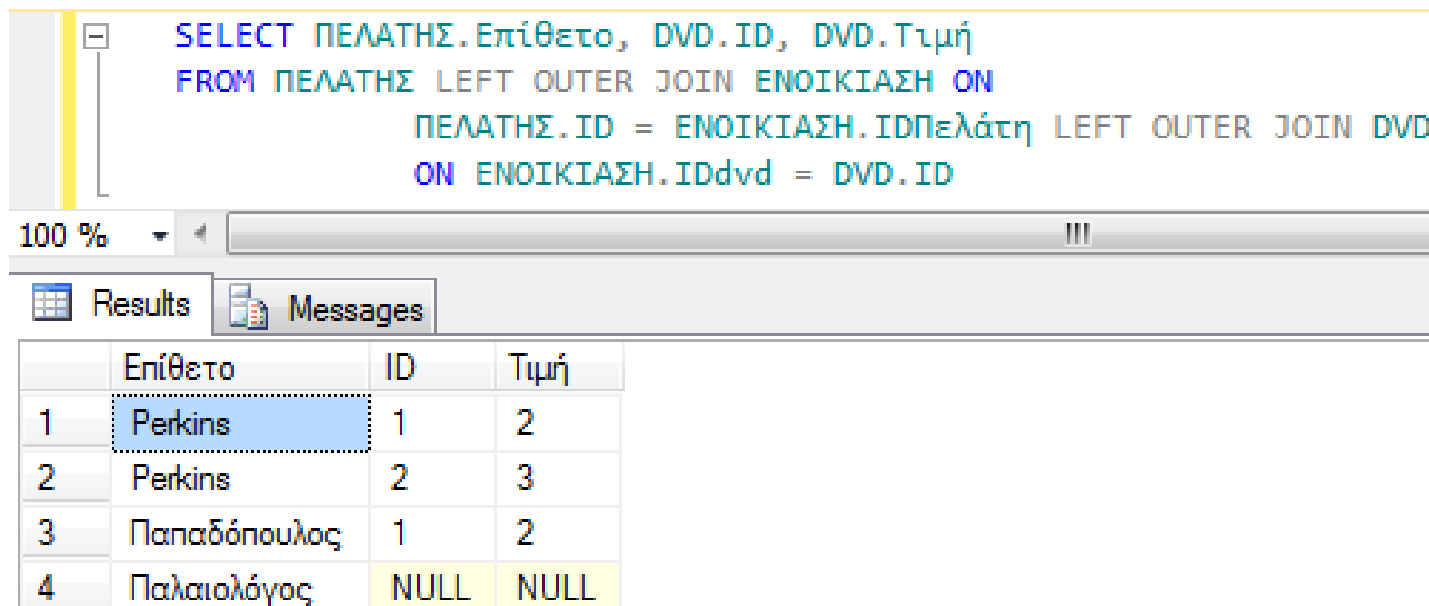
## Αριστερή εξωτερική σύνδεση μεταξύ τριών ή περισσότερων σχέσεων (LEFT OUTER JOIN)

- **LEFT OUTER JOIN** → περιέχει όλες τις πλειάδες που εμφανίζονται στο **INNER JOIN**, και επιπλέον **όλες τις πλειάδες της αριστερής σχέσης για τις οποίες δεν έγινε ταύτιση τιμών** με καμία πλειάδα της δεξιάς σχέσης.
- Οι στήλες που αντιστοιχούν στη δεξιά σχέση παίρνουν τιμή **NULL** για τις πλειάδες που δεν υπήρχε ταύτιση τιμών.
- Κάθε πλειάδα της αριστερής σχέσης συμμετέχει στο αποτέλεσμα.

# Ερώτημα με σύνδεση LEFT OUTER JOIN στον SQL Server 2012

- Να βρεθούν για κάθε πελάτη (Επίθετο), ο κωδικός και η τιμή των dvd που έχει ενοικιάσει και οι πελάτες που δεν έχουν νοικιάσει κάποιο dvd:

```
SELECT ΠΕΛΑΤΗΣ.Επίθετο, DVD.ID, DVD.Τιμή  
FROM ΠΕΛΑΤΗΣ LEFT OUTER JOIN ΕΝΟΙΚΙΑΣΗ ON  
ΠΕΛΑΤΗΣ.ID = ΕΝΟΙΚΙΑΣΗ.IDΠελάτη LEFT OUTER JOIN DVD  
ON ΕΝΟΙΚΙΑΣΗ.IDdvd = DVD.ID
```



The screenshot shows a SQL query window with the following text:

```
SELECT ΠΕΛΑΤΗΣ.Επίθετο, DVD.ID, DVD.Τιμή  
FROM ΠΕΛΑΤΗΣ LEFT OUTER JOIN ΕΝΟΙΚΙΑΣΗ ON  
ΠΕΛΑΤΗΣ.ID = ΕΝΟΙΚΙΑΣΗ.IDΠελάτη LEFT OUTER JOIN DVD  
ON ΕΝΟΙΚΙΑΣΗ.IDdvd = DVD.ID
```

Below the query window, the 'Results' tab is active, displaying a table with the following data:

	Επίθετο	ID	Τιμή
1	Perkins	1	2
2	Perkins	2	3
3	Παπαδόπουλος	1	2
4	Παλαιολόγος	NULL	NULL

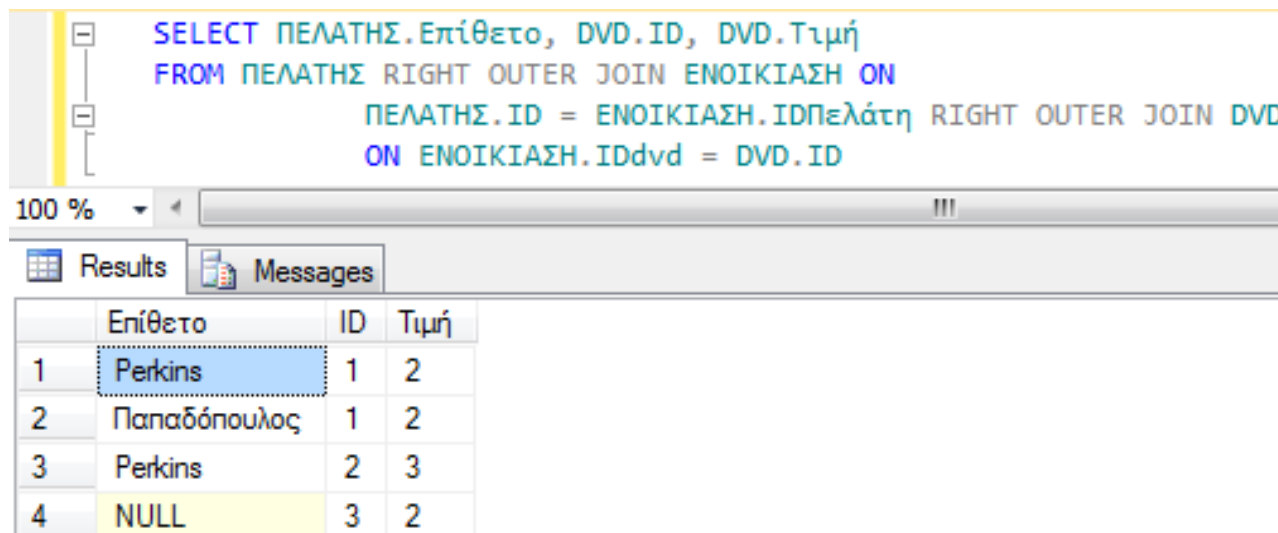
## Δεξιά εξωτερική σύνδεση μεταξύ τριών ή περισσότερων σχέσεων (RIGHT OUTER JOIN)

- **RIGHT OUTER JOIN** → περιέχει όλες τις πλειάδες που εμφανίζονται στο **INNER JOIN**, και επιπλέον **όλες τις πλειάδες της δεξιάς σχέσης για τις οποίες δεν έγινε ταύτιση** τιμών με καμία πλειάδα της αριστερής σχέσης.
- Οι στήλες που αντιστοιχούν στην αριστερή σχέση παίρνουν τιμή **NULL** για τις πλειάδες που δεν υπήρχε ταύτιση τιμών.
- Κάθε πλειάδα της δεξιάς σχέσης συμμετέχει στο αποτέλεσμα.

# Ερώτημα με σύνδεση RIGHT OUTER JOIN στον SQL Server 2012

- Να βρεθούν για κάθε πελάτη (Επίθετο), ο κωδικός και η τιμή των dvd που έχει ενοικιάσει και, επιπλέον, οι κωδικοί και οι τιμές των dvd που δεν έχουν ενοικιαστεί από κάποιον πελάτη:

```
SELECT ΠΕΛΑΤΗΣ.Επίθετο, DVD.ID, DVD.Τιμή
FROM ΠΕΛΑΤΗΣ RIGHT OUTER JOIN ΕΝΟΙΚΙΑΣΗ ON
ΠΕΛΑΤΗΣ.ID = ΕΝΟΙΚΙΑΣΗ.IDΠελάτη RIGHT OUTER JOIN DVD
ON ΕΝΟΙΚΙΑΣΗ.IDdvd = DVD.ID
```



The screenshot shows a SQL query window with the following text:

```
SELECT ΠΕΛΑΤΗΣ.Επίθετο, DVD.ID, DVD.Τιμή
FROM ΠΕΛΑΤΗΣ RIGHT OUTER JOIN ΕΝΟΙΚΙΑΣΗ ON
ΠΕΛΑΤΗΣ.ID = ΕΝΟΙΚΙΑΣΗ.IDΠελάτη RIGHT OUTER JOIN DVD
ON ΕΝΟΙΚΙΑΣΗ.IDdvd = DVD.ID
```

Below the query window, the 'Results' tab is active, displaying the following data:

	Επίθετο	ID	Τιμή
1	Perkins	1	2
2	Παπαδόπουλος	1	2
3	Perkins	2	3
4	NULL	3	2

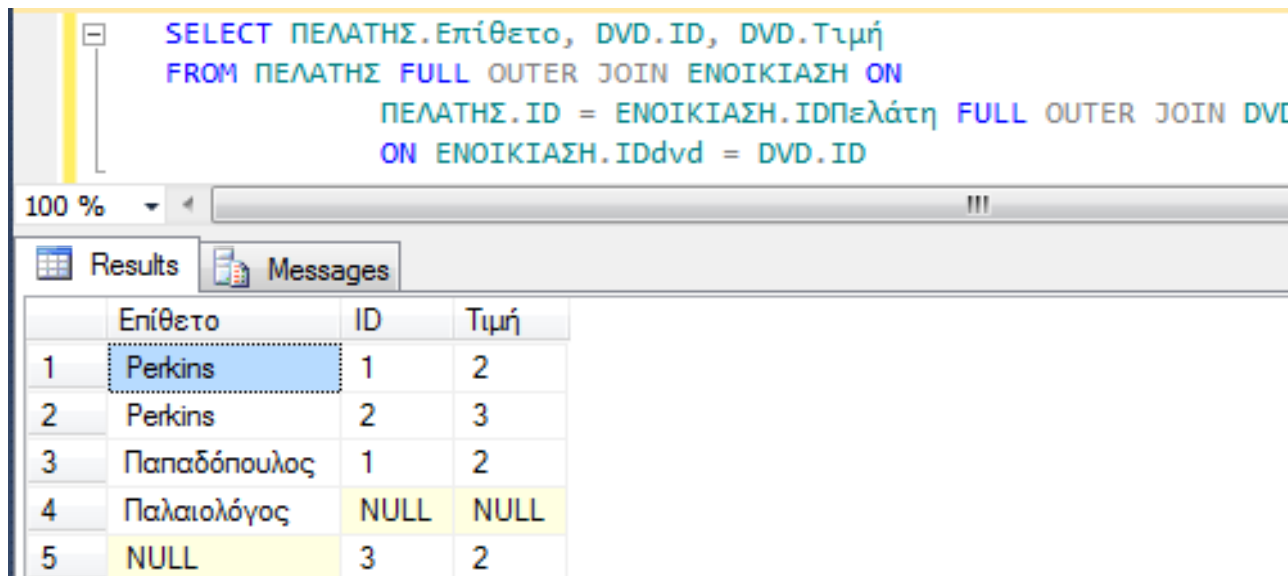
## Πλήρης εξωτερική σύνδεση μεταξύ τριών ή περισσότερων σχέσεων (FULL OUTER JOIN)

- **FULL OUTER JOIN** → περιέχει όλες τις πλειάδες της δεξιάς και της αριστερής σχέσης για τις οποίες δεν έγινε ταύτιση τιμών.
- Οι στήλες που αντιστοιχούν στην άλλη σχέση (αριστερή ή δεξιά) παίρνουν την τιμή **NULL** για τις πλειάδες που δεν υπήρχε ταύτιση τιμών.
- Κάθε πλειάδα και των δύο σχέσεων συμμετέχει στο αποτέλεσμα.

# Ερώτημα με σύνδεση FULL OUTER JOIN στον SQL Server 2012

- Να βρεθούν για κάθε πελάτη (Επίθετο), ο κωδικός και η τιμή των dvd που έχει ενοικιάσει, οι πελάτες που δεν έχουν ενοικιάσει κάποιο dvd, αλλά και οι κωδικοί και οι τιμές των dvd που δεν έχουν ενοικιαστεί από κάποιον πελάτη:

```
SELECT ΠΕΛΑΤΗΣ.Επίθετο, DVD.ID, DVD.Τιμή  
FROM ΠΕΛΑΤΗΣ FULL OUTER JOIN ΕΝΟΙΚΙΑΣΗ ON  
ΠΕΛΑΤΗΣ.ID = ΕΝΟΙΚΙΑΣΗ.IDΠελάτη FULL OUTER JOIN DVD  
ON ΕΝΟΙΚΙΑΣΗ.IDdvd = DVD.ID
```



The screenshot shows a SQL query window with the following text:

```
SELECT ΠΕΛΑΤΗΣ.Επίθετο, DVD.ID, DVD.Τιμή  
FROM ΠΕΛΑΤΗΣ FULL OUTER JOIN ΕΝΟΙΚΙΑΣΗ  
ON ΠΕΛΑΤΗΣ.ID = ΕΝΟΙΚΙΑΣΗ.IDΠελάτη FULL OUTER JOIN DVD  
ON ΕΝΟΙΚΙΑΣΗ.IDdvd = DVD.ID
```

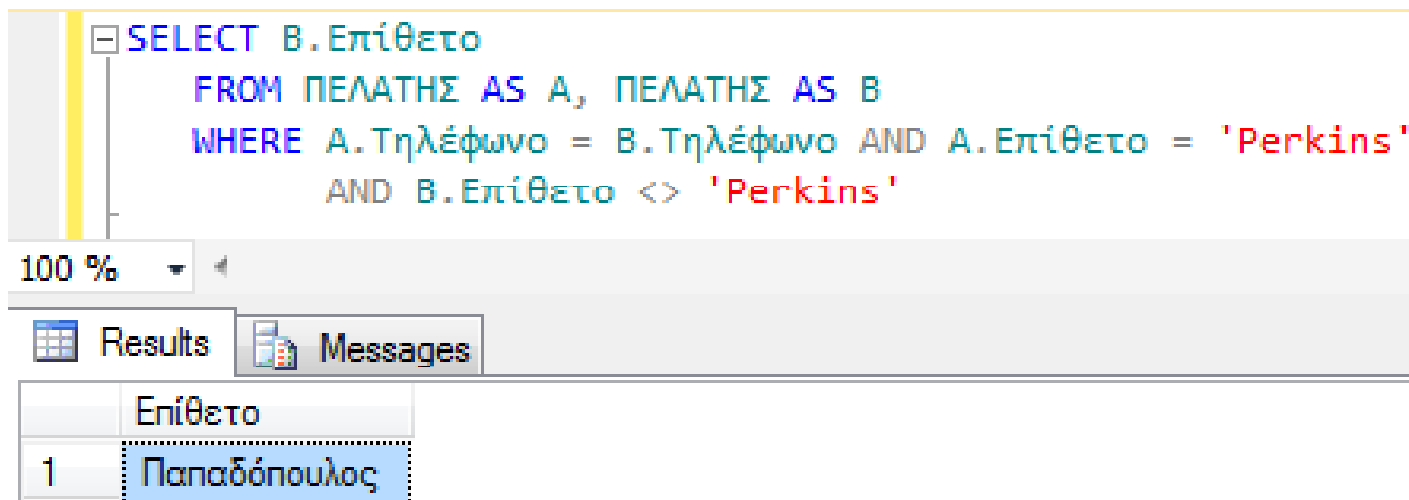
Below the query window, the 'Results' tab is active, displaying a table with 5 rows and 4 columns: 'Επίθετο', 'ID', and 'Τιμή'. The first three rows show data for customers 'Perkins' and 'Παπαδόπουλος'. The fourth row shows a customer 'Παλαιολόγος' with NULL values for ID and price. The fifth row shows a DVD with ID 3 and price 2, but with NULL for the customer name.

	Επίθετο	ID	Τιμή
1	Perkins	1	2
2	Perkins	2	3
3	Παπαδόπουλος	1	2
4	Παλαιολόγος	NULL	NULL
5	NULL	3	2

## Ερώτημα με αυτοσύνδεση στον SQL Server 2012

- Να βρεθεί το επίθετο κάθε πελάτη που έχει το ίδιο τηλέφωνο με τον πελάτη που έχει το επίθετο 'Perkins':

```
SELECT B.Επίθετο  
FROM ΠΕΛΑΤΗΣ AS A, ΠΕΛΑΤΗΣ AS B  
WHERE A.Τηλέφωνο = B.Τηλέφωνο AND A.Επίθετο = 'Perkins'  
AND B.Επίθετο <> 'Perkins'
```



The screenshot shows a SQL query window with the following text:

```
SELECT B.Επίθετο  
FROM ΠΕΛΑΤΗΣ AS A, ΠΕΛΑΤΗΣ AS B  
WHERE A.Τηλέφωνο = B.Τηλέφωνο AND A.Επίθετο = 'Perkins'  
AND B.Επίθετο <> 'Perkins'
```

Below the query window, there are tabs for "Results" and "Messages". The "Results" tab is active, showing a table with the following data:

	Επίθετο
1	Παπαδόπουλος

## SQL ερωτήματα στον SQL Server 2012

- Δοκιμάστε να εκφράσετε σε SQL τα ερωτήματα:
  - ❖ Να βρεθούν τα ονόματα (επίθετο) των πελατών που έχουν ενοικιάσει τουλάχιστον ένα dvd του οποίου η διαθέσιμη ποσότητα είναι πάνω από 2.
  - ❖ Να βρεθούν ο κωδικός κάθε ταινίας για την οποία το dvd τύπου '3D' είναι σε μεγαλύτερη διαθέσιμη ποσότητα από το αντίστοιχο dvd τύπου 'SD'. (χρήση αυτοσύνδεσης)



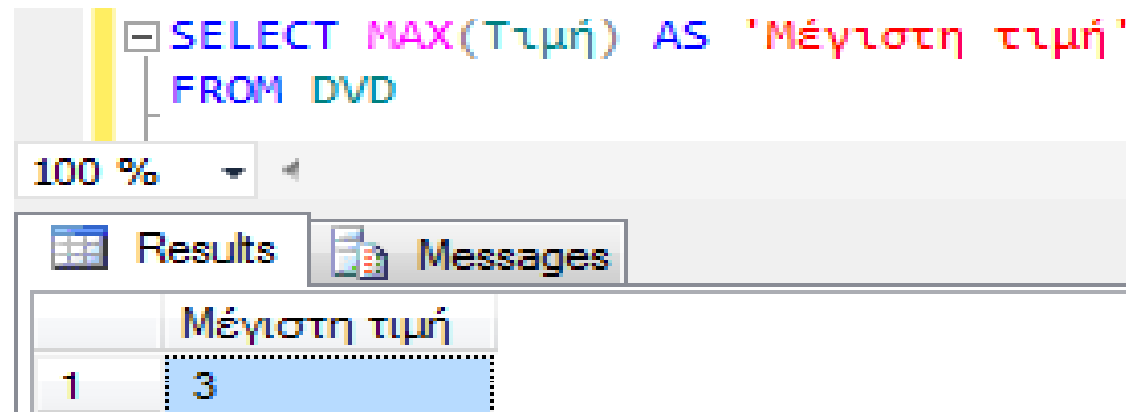
## Συγκεντρωτικοί τελεστές

- Οι συγκεντρωτικοί τελεστές ή αλλιώς τελεστές συνάθροισης είναι:
  - **SUM** : υπολογίζει το άθροισμα δύο τιμών
  - **AVG** : τη μέση τιμή των τιμών
  - **MIN** : το ελάχιστο των τιμών
  - **MAX** : το μέγιστο των τιμών
  - **COUNT**: πλήθος των τιμών
- **DISTINCT**: σε συνδυασμό με τους συγκεντρωτικούς τελεστές ώστε να απαλειφθούν τα διπλότυπα, π.χ. **COUNT(DISTINCT x)**.

## Ερώτημα με συγκεντρωτικό τελεστή στον SQL Server 2012 (1/2)

- Να βρεθεί η μεγαλύτερη τιμή ενοικίασης ενός dvd:

```
SELECT MAX(Τιμή) AS 'Μέγιστη τιμή'  
FROM DVD
```



The screenshot shows the SQL Server Enterprise Manager interface. The query editor displays the following SQL query:

```
SELECT MAX(Τιμή) AS 'Μέγιστη τιμή'  
FROM DVD
```

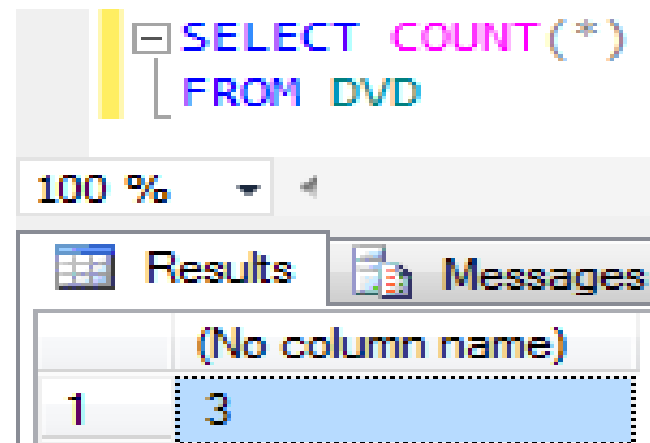
Below the query editor, the 'Results' tab is active, showing a single row of data:

	Μέγιστη τιμή
1	3

## Ερώτημα με συγκεντρωτικό τελεστή στον SQL Server 2012 (2/2)

- Να βρεθεί ο συνολικός αριθμός των dvd:

```
SELECT COUNT(*)  
FROM DVD
```



The screenshot shows the SQL Server Enterprise Manager interface. The query window displays the SQL statement: `SELECT COUNT(*) FROM DVD`. Below the query window, the 'Results' tab is active, showing a grid with one row and one column. The column header is '(No column name)' and the value in the row is '3'.

	(No column name)
1	3

# Οι φράσεις GROUP BY και HAVING

- **GROUP BY:** ομαδοποιεί τις πλειάδες μίας σχέσης με βάση τις τιμές κάποιου(-ων) χαρακτηριστικού(-ών) μίας σχέσης.
- **HAVING:** κάθε μία ομάδα πλειάδων μπορεί να ικανοποιεί ορισμένες συνθήκες.

## Η σύνταξη ενός SQL ερωτήματος με τις εντολές GROUP BY και HAVING

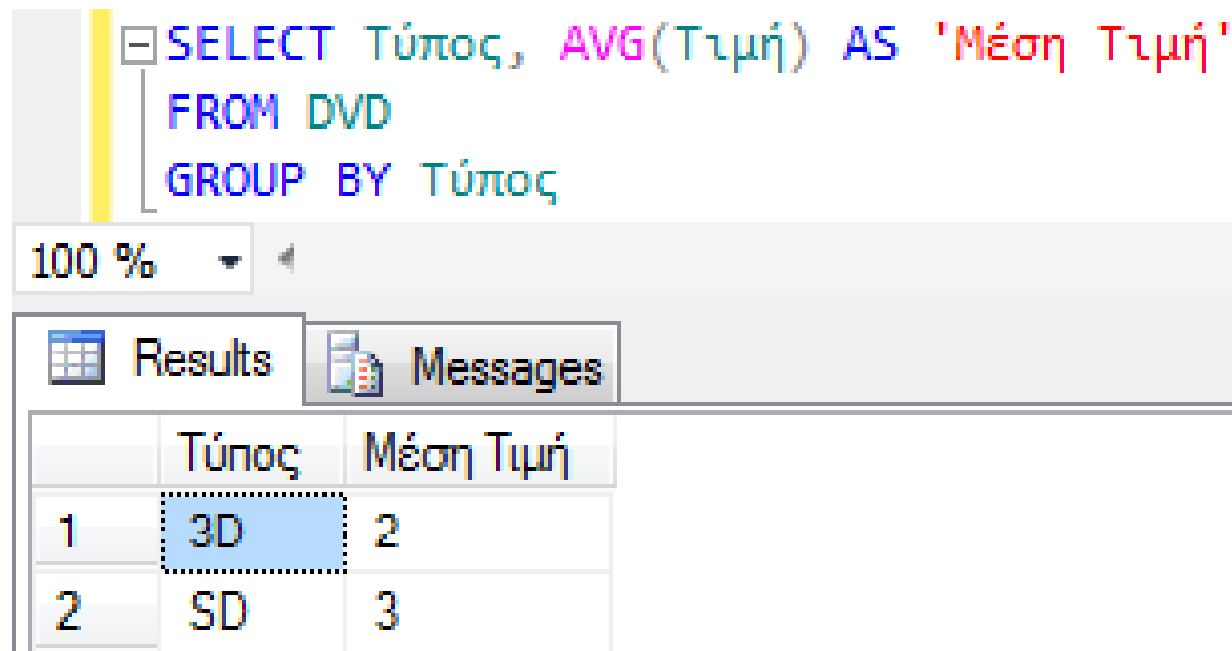
- **SELECT**  $A_1, A_2, \dots, A_k$   
**FROM**  $R_1, \dots, R_m$   
**WHERE** <συνθήκη>  
**GROUP BY**  $A_1, A_2, \dots, A_n$   
**HAVING** <συνθήκη> (υπολογίζει μία μόνο τιμή για την κάθε ομάδα πλειάδων που σχηματίζεται βάσει των τιμών των χαρακτηριστικών  $A_1, A_2, \dots, A_n$ )
- Οποιοδήποτε χαρακτηριστικό εμφανίζεται μετά το SELECT πρέπει να εμφανίζεται και μετά το GROUP BY (εξαιρούνται τα χαρακτηριστικά μέσα στους συγκεντρωτικούς τελεστές)

# Η σειρά εκτέλεσης ενός ερωτήματος ομαδοποίησης με GROUP BY και HAVING

1. Σχηματίζεται το καρτεσιανό γινόμενο των σχέσεων  $R_1, R_2, \dots, R_m$  του FROM.
2. Επιλέγονται πλειάδες από το καρτεσιανό γινόμενο που ικανοποιούν την συνθήκη του WHERE.
3. Διατηρούνται μόνο τα χαρακτηριστικά που βρίσκονται μετά τις εντολές SELECT, GROUP BY και HAVING.
4. Δημιουργούνται ομάδες πλειάδων σύμφωνα με την εντολή GROUP BY.
5. Εφαρμόζεται η συνθήκη καταλληλότητας της κάθε ομάδας σύμφωνα με την εντολή HAVING.
6. Υπολογίζονται οι πλειάδες εξόδου, πχ. υπολογίζονται όσοι τελεστές συνάθροισης υπάρχουν μετά την εντολή SELECT.

# Ερώτημα με GROUP BY στον SQL Server 2012 (1/3)

- Να βρεθεί ο μέσος όρος τιμής ενοικίασης ανά τύπο dvd('SD' ή '3D'):  
**SELECT** Τύπος, AVG(Τιμή) AS 'Μέση Τιμή'  
**FROM** DVD  
**GROUP BY** Τύπος



The screenshot shows the SQL Server Enterprise Manager interface. The query editor contains the following SQL query:

```
SELECT Τύπος, AVG(Τιμή) AS 'Μέση Τιμή'  
FROM DVD  
GROUP BY Τύπος
```

Below the query editor, the 'Results' tab is active, displaying the following table:

	Τύπος	Μέση Τιμή
1	3D	2
2	SD	3

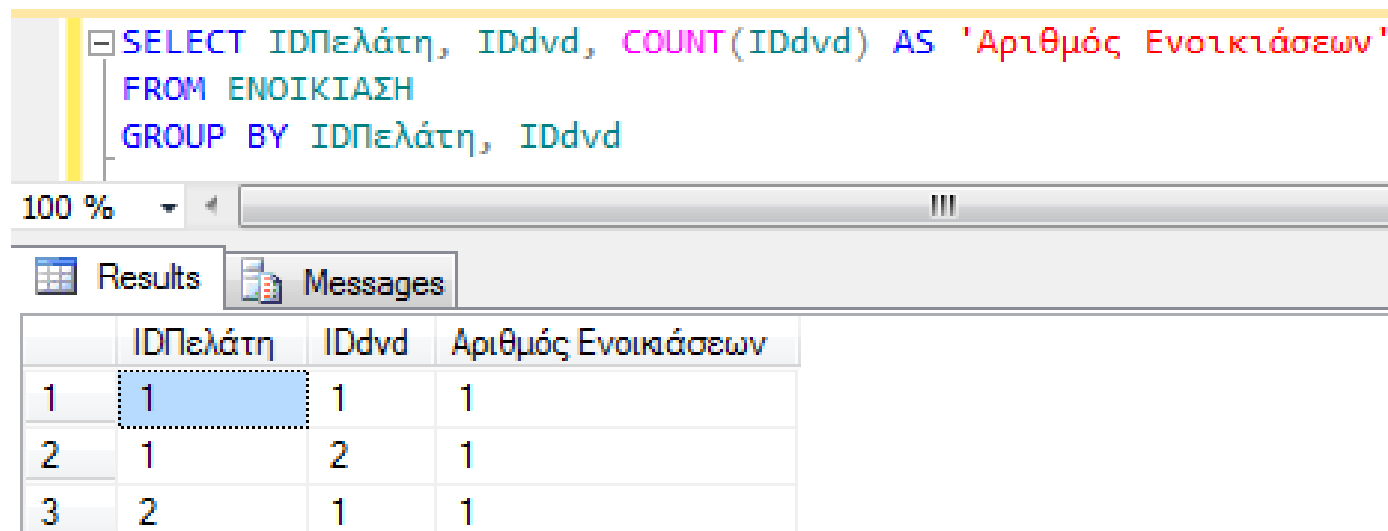
## Ερώτημα με GROUP BY στον SQL Server 2012 (2/3)

- Για κάθε πελάτη (κωδικός) να βρεθεί ο αριθμός των φορών που ενοικίασε κάθε dvd(κωδικός):

```
SELECT IDΠελάτη, IDdvd, COUNT(IDdvd) AS 'Αριθμός  
Ενοικιάσεων'
```

```
FROM ΕΝΟΙΚΙΑΣΗ
```

```
GROUP BY IDΠελάτη, IDdvd
```



The screenshot shows the SQL Server Enterprise Manager interface. The query editor at the top contains the following SQL query:

```
SELECT IDΠελάτη, IDdvd, COUNT(IDdvd) AS 'Αριθμός Ενοικιάσεων'  
FROM ΕΝΟΙΚΙΑΣΗ  
GROUP BY IDΠελάτη, IDdvd
```

Below the query editor, the 'Results' tab is active, displaying the following data table:

	IDΠελάτη	IDdvd	Αριθμός Ενοικιάσεων
1	1	1	1
2	1	2	1
3	2	1	1



# Ερώτημα με GROUP BY στον SQL Server 2012 (3/3)

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

```
SELECT ΣΥΝΤΕΛΕΣΤΗΣ.Όνομα, COUNT(DISTINCT ΤΣ.Ρόλος) AS 'Αριθμός  
Ρόλων'
```

```
FROM ΣΥΝΤΕΛΕΣΤΗΣ INNER JOIN ΤΣ ON ΣΥΝΤΕΛΕΣΤΗΣ.ID = ΤΣ.IDΣυντελεστή  
GROUP BY ΣΥΝΤΕΛΕΣΤΗΣ.Όνομα
```

The screenshot shows a SQL query window with the following text:

```
SELECT ΣΥΝΤΕΛΕΣΤΗΣ.Όνομα, COUNT(DISTINCT ΤΣ.Ρόλος) AS 'Αριθμός Ρόλων'  
FROM ΣΥΝΤΕΛΕΣΤΗΣ INNER JOIN ΤΣ ON ΣΥΝΤΕΛΕΣΤΗΣ.ID = ΤΣ.IDΣυντελεστή  
GROUP BY ΣΥΝΤΕΛΕΣΤΗΣ.Όνομα
```

Below the query window, the 'Results' tab is active, displaying a table with the following data:

	Όνομα	Αριθμός
1	Alfred Hitchcock	1
2	Anthony Perkins	1
3	Grace Kelly	1

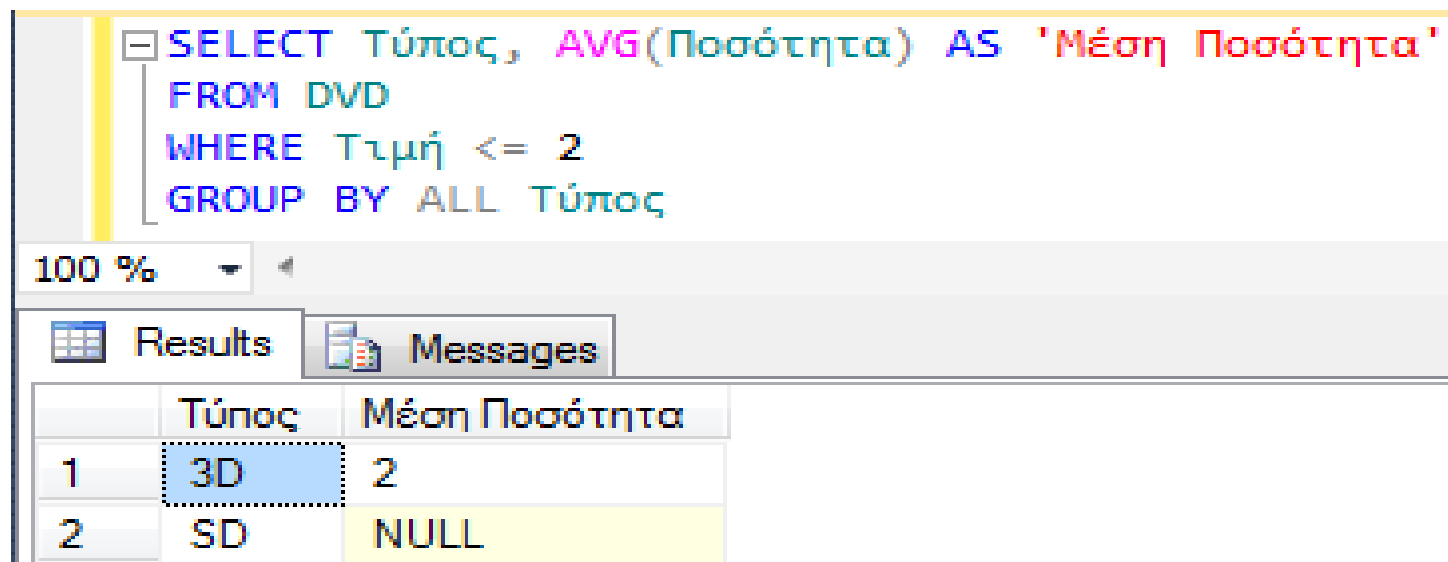
## Η φράση GROUP BY ALL

- **GROUP BY ALL:** δηλώνει ότι όλες οι ομάδες πρέπει να συμπεριληφθούν στο αποτέλεσμα, ακόμα κι αν καμία πλειάδα τους δεν ικανοποιεί την συνθήκη της εντολής WHERE.
- Οι ομάδες που δεν έχουν πλειάδες να ικανοποιούν τη συνθήκη της εντολής WHERE περιέχουν **NULL**.

# Ερώτημα με GROUP BY ALL στον SQL Server 2012

- Να βρεθεί η μέση ποσότητα ανά τύπο dvd, για τα dvd των οποίων η τιμή είναι μικρότερη από 2. Ακόμη, να παρουσιάζονται οι τύποι dvd που δεν ικανοποιούν την παραπάνω συνθήκη:

```
SELECT Τύπος, AVG(Ποσότητα) AS 'Μέση Ποσότητα'  
FROM DVD  
WHERE Τιμή <= 2  
GROUP BY ALL Τύπος
```



The screenshot shows a SQL query window with the following text:

```
SELECT Τύπος, AVG(Ποσότητα) AS 'Μέση Ποσότητα'  
FROM DVD  
WHERE Τιμή <= 2  
GROUP BY ALL Τύπος
```

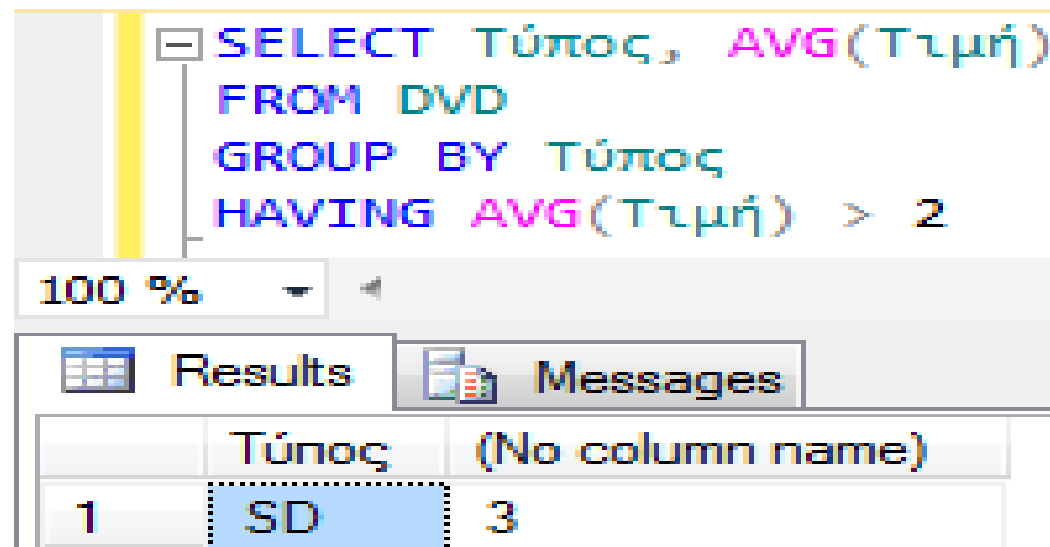
Below the query window, there is a 'Results' tab showing the following data:

	Τύπος	Μέση Ποσότητα
1	3D	2
2	SD	NULL

## Ερώτημα με HAVING στον SQL Server 2012

- Να βρεθούν οι τύποι dvd για τους οποίους ο μέσος όρος τιμής είναι μεγαλύτερος από 2:

```
SELECT Τύπος, AVG(Τιμή)  
FROM DVD  
GROUP BY Τύπος  
HAVING AVG(Τιμή) > 2
```



The screenshot shows a SQL query window with the following text:

```
SELECT Τύπος, AVG(Τιμή)  
FROM DVD  
GROUP BY Τύπος  
HAVING AVG(Τιμή) > 2
```

Below the query window, there is a zoom level of 100% and two tabs: "Results" and "Messages". The "Results" tab is active, displaying a table with the following data:

	Τύπος	(No column name)
1	SD	3

## SQL ερωτήματα στον SQL Server 2012

- Δοκιμάστε να εκφράσετε σε SQL τα ερωτήματα:
  - ❖ Να βρεθεί η μεγαλύτερη τιμή ενοικίασης ενός dvd που έχει ενοικιαστεί.
  - ❖ Να βρεθεί η διαφορά μεταξύ του ακριβότερου και του φθηνότερου DVD.
  - ❖ Να βρεθούν τα ονόματα των συντελεστών που έχουν συμμετάσχει σε περισσότερες από μία ταινίες.
  - ❖ Να βρεθούν οι κωδικοί των συντελεστών που έχουν σκηνοθετήσει περισσότερες από μία ταινίες.