

# Βάσεις Δεδομένων

## Εργαστήριο V

Τμήμα Πληροφορικής ΑΠΘ  
2013-2014

## Σκοπός του 5<sup>ου</sup> εργαστηρίου

- Σκοπός αυτού του εργαστηρίου είναι:
  - η μελέτη ερωτημάτων τροποποίησης δομής / δεδομένων
  - η μελέτη σύνθετων ερωτημάτων σύνδεσης σε δύο ή περισσότερες σχέσεις
  - η μελέτη σύνθετων εμφωλευμένων ερωτημάτων

## Τροποποίηση Δομής Πινάκων μιας Βάσης Δεδομένων (1 / 3)

- Τρόποι τροποποίησης δομής πίνακα:
  - Προσθήκης νέου πεδίου / στήλης σε ένα πίνακα
  - Αφαίρεση κάποιου πεδίου / στήλης ενός πίνακα
  - Αλλαγή κάποιου πεδίου / στήλης ενός πίνακα
  - Διαγραφή κάποιου πίνακα

# Τροποποίηση Δομής Πινάκων μιας Βάσης Δεδομένων (2/3)

- Βασική Σύνταξη Εντολών:
  - **ALTER TABLE** Πίνακας  
**ADD COLUMN** Πεδίο Τύπος
  - **ALTER TABLE** Πίνακας  
**DROP COLUMN** Πεδίο Τύπος
- Παραδείγματα:
  - **ALTER TABLE ΠΕΛΑΤΗΣ**  
**DROP COLUMN** Τηλέφωνο
  - **ALTER TABLE ΠΕΛΑΤΗΣ**  
**ADD COLUMN** email **varchar**(200)

# Τροποποίηση Δομής Πινάκων μιας Βάσης Δεδομένων (3/3)

- Βασική Σύνταξη Εντολών:
  - **ALTER TABLE** Πίνακας
  - ALTER COLUMN** Πεδίο Τύπος
  
- Παράδειγμα:
  - **ALTER TABLE ΠΕΛΑΤΗΣ**
  - ALTER COLUMN** Επίθετο char(100) not null

# Τροποποίηση Δεδομένων Πινάκων μιας Βάσης Δεδομένων

- Βασική Σύνταξη Εντολών:
  - **UPDATE** Πίνακας **SET** Πεδίο $i$  = Τιμή $i$ , Πεδίο $j$  = Τιμή $j$ , ...  
**WHERE** Συνθήκη;
- Παράδειγμα:
  - **UPDATE** TAINIA **SET** Χρονιά = 1960 **WHERE** Τίτλος = 'Ben-Hur';

## Όψεις (Views)

- Όψη: ένας εικονικός πίνακας που προκύπτει από άλλους πίνακες
- Μπορούμε να χρησιμοποιήσουμε μία όψη σαν να είναι ένας πίνακας (για εντολές select)
- Η όψη δεν απαιτείται να είναι αποθηκευμένη σε φυσική μορφή.

## Όψεις (Views)

- Ορισμός της όψης στην SQL:
  - `Create view < name of view > as ( sql query )`
- Διαγραφή όψης στην SQL:
  - `Drop view < name of view >`



## Όψεις (Views)

- Παράδειγμα δημιουργίας όψης
  - Να ορισθεί όψη με όνομα H, που περιέχει όλους τους κωδικούς των ταινιών που συμμετέχει ο Alfred Hitchcock:

**CREATE VIEW H AS**

**(SELECT IDΤαινίας**

**FROM ΤΣ INNER JOIN ΣΥΝΤΕΛΕΣΤΗΣ ON**

**ΤΣ.IDΣυντελεστή = ΣΥΝΤΕΛΕΣΤΗΣ.ID**

**WHERE ΣΥΝΤΕΛΕΣΤΗΣ. Όνομα**

**= 'Alfred Hitchcock')**

## Όψεις (Views)

- Παράδειγμα χρήσης όψης σε ερωτήματα επιλογής:
  - Να βρεθεί ο κωδικός πελάτη με το μεγαλύτερο αριθμό ενοικιάσεων:

```
CREATE VIEW numRent AS
    (SELECT IDΠελάτη, COUNT (*) as A
     FROM ΕΝΟΙΚΙΑΣΗ
     GROUP BY IDΠελάτη)
SELECT IDΠελάτη
FROM numRent
WHERE A = (SELECT MAX(A)
           FROM numRent)
```

## Σύνθετα SQL ερωτήματα με δύο ή περισσότερες σχέσεις

- ❖ Να βρεθούν οι τίτλοι των ταινιών που γυρίστηκαν πριν το 1960 ή αυτών στις οποίες έχει συμμετάσχει η Grace Kelly. Να γίνει προβολή των διπλοεγγραφών καθώς και να δωθούν με φθίνουσα σειρά ταξινόμησης οι πλειάδες εξόδου.

```
(SELECT Τίτλος
FROM TAINIA
WHERE Χρονιά < 1960)
```

**UNION ALL**

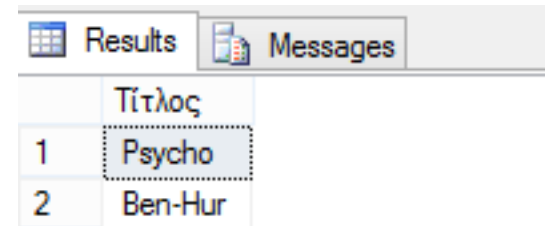
```
(SELECT Τίτλος
FROM TAINIA, ΣΥΝΤΕΛΕΣΤΗΣ, ΤΣ
WHERE ΤΣ.IDΤαινίας = TAINIA.ID AND
      ΤΣ.IDΣυντελεστής = ΣΥΝΤΕΛΕΣΤΗΣ.ID AND
      ΣΥΝΤΕΛΕΣΤΗΣ.Όνομα = 'Grace Kelly');
ORDER BY Τίτλος
```

Results		Messages
	Τίτλος	
1	Ben-Hur	
2	Rear Window	
3	Rear Window	

## Σύνθετα SQL ερωτήματα με δύο ή περισσότερες σχέσεις

- ❖ Να βρεθούν οι τίτλοι των ταινιών για τις οποίες είτε δεν υπάρχει DVD, είτε υπάρχει DVD και δεν έχει ενοικιαστεί ποτέ. (να μην χρησιμοποιηθεί ο τελεστής NOT IN).

```
SELECT Τίτλος  
FROM TAINIA LEFT OUTER JOIN DVD ON  
    DVD.IDΤαινίας = TAINIA.ID LEFT OUTER JOIN  
    ΕΝΟΙΚΙΑΣΗ ON ΕΝΟΙΚΙΑΣΗ.IDdvd = DVD.ID  
WHERE ΕΝΟΙΚΙΑΣΗ.IDdvd IS NULL
```



	Τίτλος
1	Psycho
2	Ben-Hur

## Σύνθετα SQL ερωτήματα με δύο ή περισσότερες σχέσεις

- ❖ 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)

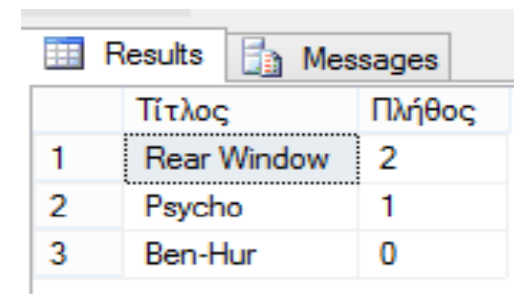
```
SELECT A.IDΤαινίας  
FROM DVD AS A, DVD AS B  
WHERE A.IDΤαινίας = B.IDΤαινίας AND  
A.Τύπος = '3D' AND B.Τύπος= 'SD' AND  
A.Ποσότητα > B.Ποσότητα
```

Results		Messages	
	IDΤαινίας		
1	1		

## Σύνθετα SQL ερωτήματα με δύο ή περισσότερες σχέσεις

- ❖ Για κάθε ταινία (τίτλος), να βρεθεί ο αριθμός διαφορετικών τύπων DVD που περιέχει την ταινία. Στο αποτέλεσμα να εμφανίζονται και οι ταινίες για τις οποίες δεν υπάρχει DVD:

```
SELECT TAINIA.Τίτλος, count(DVD.Τύπος) as 'πλήθος'  
FROM TAINIA LEFT OUTER JOIN DVD ON  
DVD.IDΤαινίας = TAINIA.ID  
GROUP BY TAINIA.ID, TAINIA.Τίτλος
```

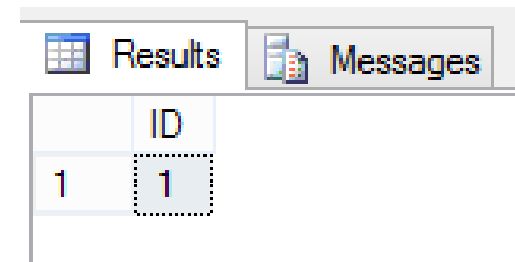


	Τίτλος	Πλήθος
1	Rear Window	2
2	Psycho	1
3	Ben-Hur	0

## Σύνθετα SQL ερωτήματα με δύο ή περισσότερες σχέσεις

- ❖ Να βρεθούν οι κωδικοί των DVD που είναι τύπου 3D και έχουν ενοικιασθεί περισσότερες από μία φορές:

```
SELECT DVD.ID  
FROM DVD INNER JOIN ΕΝΟΙΚΙΑΣΗ ON  
    DVD.ID = ΕΝΟΙΚΙΑΣΗ.IDdvd  
WHERE DVD.Τύπος = '3D'  
GROUP BY DVD.ID  
HAVING COUNT(*) > 1;
```



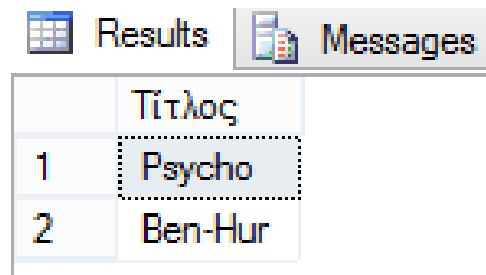
The screenshot shows a database interface with two tabs: 'Results' and 'Messages'. The 'Results' tab is active and displays a table with one column labeled 'ID' and one row containing the value '1'.

ID
1

# Σύνθετα Εμφωλευμένα SQL ερωτήματα

- ❖ Να βρεθούν οι τίτλοι των ταινιών που δεν έχουν ενοικιασθεί ποτέ (Να μην γίνει χρήση του outer join):

```
SELECT Τίτλος  
FROM TAINIA  
WHERE NOT EXISTS  
    (SELECT DVD.IDΤαινίας  
    FROM ΕΝΟΙΚΙΑΣΗ INNER JOIN DVD ON  
    ΕΝΟΙΚΙΑΣΗ.IDdvd = DVD.ID  
    WHERE IDΤαινίας = TAINIA.ID);
```



The screenshot shows a SQL query results window with two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with two columns: 'ID' and 'Τίτλος'. The table contains two rows: row 1 with ID '1' and title 'Psycho', and row 2 with ID '2' and title 'Ben-Hur'.

ID	Τίτλος
1	Psycho
2	Ben-Hur



## Σύνθετα Εμφωλευμένα SQL ερωτήματα

- ❖ Για κάθε πελάτη (Επίθετο) να δοθεί το πλήθος των DVD που έχει ενοικιάσει του οποίου ο τύπος είναι '3D' καθώς και ο τίτλος της ταινίας ξεκινά από το γράμμα 'R':

```

SELECT ΠΕΛΑΤΗΣ.Επίθετο, COUNT(DVD.ID) AS 'Πλήθος'
FROM ΤΑΙΝΙΑ INNER JOIN DVD ON DVD.IDΤαινίας = ΤΑΙΝΙΑ.ID
INNER JOIN ΕΝΟΙΚΙΑΣΗ ON ΕΝΟΙΚΙΑΣΗ.IDdvd =
DVD.ID
INNER JOIN ΠΕΛΑΤΗΣ ON ΠΕΛΑΤΗΣ.ID =
ΕΝΟΙΚΙΑΣΗ.IDΠελάτη
WHERE ΤΑΙΝΙΑ.Τίτλος LIKE 'R%' AND DVD.Τύπος='3D'
GROUP BY ΠΕΛΑΤΗΣ.ID, ΠΕΛΑΤΗΣ.Επίθετο
  
```

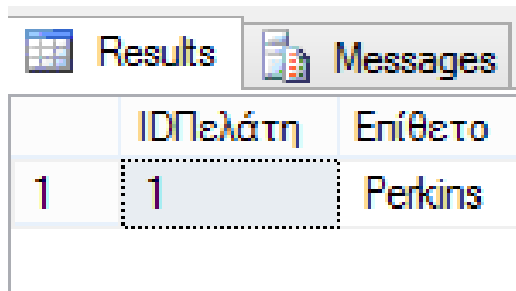
	Επίθετο	πλήθος
1	Perkins	1
2	Παπαδόπουλος	1

## Σύνθετα Εμφωλευμένα SQL ερωτήματα

- ❖ Να βρεθεί ο κωδικός και το επίθετο του πελάτη με το μεγαλύτερο αριθμό ενοικιάσεων:

```
SELECT IDΠελάτη, ΠΕΛΑΤΗΣ.Επίθετο  
FROM ΠΕΛΑΤΗΣ INNER JOIN ΕΝΟΙΚΙΑΣΗ ON  
ΠΕΛΑΤΗΣ.ID = ΕΝΟΙΚΙΑΣΗ.IDΠελάτη  
GROUP BY IDΠελάτη, ΠΕΛΑΤΗΣ.Επίθετο  
HAVING COUNT(*) >= ALL
```

```
( SELECT COUNT(*)  
FROM ΕΝΟΙΚΙΑΣΗ  
GROUP BY IDΠελάτη )
```

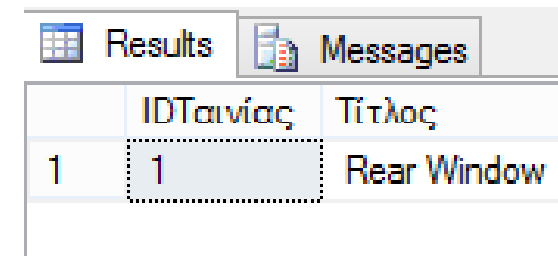


	IDΠελάτη	Επίθετο
1	1	Perkins

# Σύνθετα Εμφωλευμένα SQL ερωτήματα

- ❖ Να βρεθούν οι κωδικοί και οι τίτλοι των ταινιών στις οποίες έχει συμμετάσχει ο Alfred Hitchcock και έχουν ενοικιαστεί περισσότερες από δύο φορές:

```
SELECT DVD.IDΤαινίας, TAINIA.Τίτλος
FROM TAINIA INNER JOIN DVD ON TAINIA.ID = DVD.IDΤαινίας
      INNER JOIN ΕΝΟΙΚΙΑΣΗ ON ΕΝΟΙΚΙΑΣΗ.IDdvd = DVD.ID
WHERE DVD.IDΤαινίας IN
      (SELECT IDΤαινίας
      FROM ΤΣ INNER JOIN ΣΥΝΤΕΛΕΣΤΗΣ ON
      ΤΣ.IDΣυντελεστή = ΣΥΝΤΕΛΕΣΤΗΣ.ID
      WHERE ΣΥΝΤΕΛΕΣΤΗΣ.Όνομα = 'Alfred Hitchcock')
GROUP BY DVD.IDΤαινίας, TAINIA.Τίτλος
HAVING COUNT(*) > 2
```



	IDΤαινίας	Τίτλος
1	1	Rear Window

# Σύνθετα Εμφωλευμένα SQL ερωτήματα

- ❖ Να βρεθεί το όνομα και ο κωδικός του συντελεστή που έχει τις περισσότερες συμμετοχές σε ταινίες συνολικά:

```
SELECT ID,Όνομα
FROM ΣΥΝΤΕΛΕΣΤΗΣ
WHERE ID IN
```

```
(SELECT IDΣυντελεστή
FROM ΤΣ
```

```
GROUP BY IDΣυντελεστή
```

```
HAVING COUNT(IDΤαινίας) >= ALL
```

```
(SELECT COUNT(*)
```

```
FROM ΤΣ
```

```
GROUP BY IDΣυντελεστή)
```

Βρίσκουμε το όνομα  
του συντελεστή

Βρίσκουμε το συντελεστή  
με το μεγαλύτερο πλήθος  
συμμετοχών

Βρίσκουμε το πλήθος  
των συμμετοχών ανά  
συντελεστή

Results		Messages
ID	Όνομα	
1	Alfred Hitchcock	

# Σύνθετα Εμφωλευμένα SQL ερωτήματα

- ❖ Να βρεθεί ο κωδικός του DVD με τη δεύτερη μεγαλύτερη ποσότητα. Στο αποτέλεσμα να εμφανίζεται μόνο ο κωδικός αυτός. (Υπόδειξη: το DVD με τη δεύτερη μεγαλύτερη ποσότητα είναι αυτή που συμμετέχει στο top2 της κατάταξης ως προς την ποσότητα, αλλά δεν συμμετέχει στο top1.)

```
CREATE VIEW T1 as  
(SELECT TOP 2 ID  
FROM DVD  
ORDER BY Ποσότητα desc)
```

```
SELECT ID  
FROM T2  
WHERE ID NOT IN  
(SELECT ID FROM T1);
```

```
CREATE VIEW T2 as  
(SELECT TOP 1 ID  
From DVD  
ORDER BY Ποσότητα desc)
```

```
DROP VIEW T 1  
T 2
```