

# SQL

# Τι είδαμε μέχρι τώρα

Δύο γλώσσες ερωτήσεων που αποτελούν το θεωρητικό υπόβαθρο

**Σχεσιακή άλγεβρα:** μια άλγεβρα συνόλων που αφορά πράξεις πάνω σε σχέσεις

**Σχεσιακό λογισμό (πλειάδων):** δηλωτικό τρόπο έκφρασης ερωτήσεων

# SQL

- *Ειδικού σκοπού γλώσσα προγραμματισμού για βάσεις δεδομένων*
- Η “standard” γλώσσα για σχεσιακές βάσεις δεδομένων.
- Δηλωτική (declarative) (αν και έχει κάποια στοιχεία διαδικαστικού προγραμματισμού)
- αρχικά Sequel (Structured English Query language) στην IBM ως μέρος του System R,
  - τώρα SQL (Stuctured Query Language)
- SQL-89, SQL-92, SQL-99, ...

# SQL

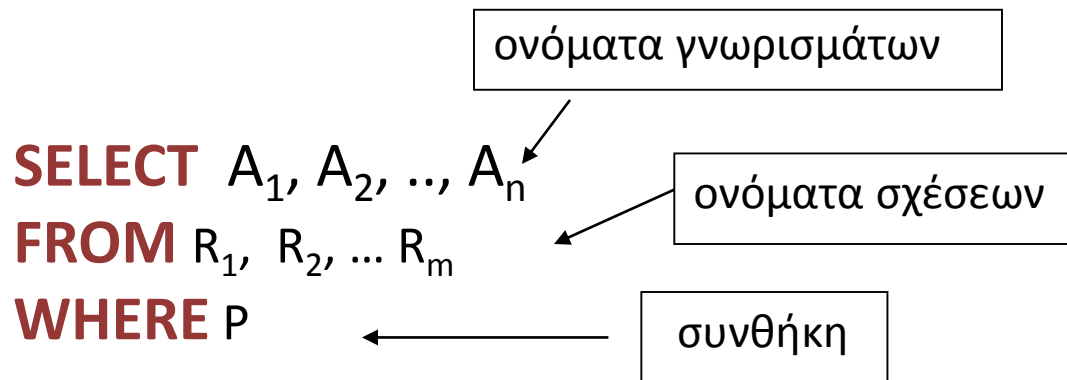
- **DDL (Data Definition Language)** Γλώσσα Ορισμού Δεδομένων (ΓΟΔ): ορισμός, δημιουργία, τροποποίηση και διαγραφή σχήματος – *την είδαμε σε προηγούμενο μάθημα*
- **DML (Data Manipulation Language)** Γλώσσα Χειρισμού Δεδομένων (ΓΟΔ)
  - εισαγωγή, τροποποίηση, διαγραφή δεδομένων - *την είδαμε σε προηγούμενο μάθημα*
  - επιλογή δεδομένων (γλώσσα ερωτήσεων)

Προδιαγραφές ασφάλειας - χρήστες και δικαιώματα.

# Βασική Δομή Ερώτησης

# Βασική Δομή

Η βασική δομή μιας ερώτησης σε SQL έχει την εξής μορφή:



Ισοδύναμο του:  $\pi_{A_1, A_2, \dots, A_n} (\sigma_P (R_1 \times R_2 \times \dots \times R_m))$

# select

**SELECT** A1, A2, ..., An

**FROM** R<sub>1</sub>, R<sub>2</sub>, ... R<sub>m</sub>

**WHERE** P

$$\pi_{A_1, A_2, \dots, A_n} (\sigma_P (R_1 \times R_2 \times \dots \times R_m))$$

**select** αντιστοιχεί στην πράξη της προβολής της σχεσιακής άλγεβρας

*Ποια γνωρίσματα θέλουμε να υπάρχουν στο αποτέλεσμα της ερώτησης.*

# from

**SELECT** A1, A2, ..., An  
**FROM** R<sub>1</sub>, R<sub>2</sub>, ..., R<sub>m</sub>  
**WHERE** P

$$\pi_{A_1, A_2, \dots, A_n} (\sigma_P (R_1 \times R_2 \times \dots \times R_m))$$

**from** αντιστοιχεί στην πράξη του καρτεσιανού γινομένου της σχεσιακής άλγεβρας.

*Ποιες σχέσεις θα χρησιμοποιηθούν για τον υπολογισμό του αποτελέσματος.*



# where

```
SELECT A1, A2, ..., An  
FROM R1, R2, ... Rm  
WHERE P
```

$$\pi_{A_1, A_2, \dots, A_n} (\sigma_P (R_1 \times R_2 \times \dots \times R_m))$$

**where** αντιστοιχεί στη συνθήκη της πράξης της επιλογής στη σχεσιακή άλγεβρα.

Το κατηγορημα **P** έχει γνωρίσματα των σχέσεων που εμφανίζονται στο from.

# Παράδειγμα

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)

Παίζει(Όνομα, Τίτλος, Έτος)

Ηθοποιός(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

Ονόματα ηθοποιών που παίζουν στην ταινία Gone by the Wind

```
SELECT Όνομα  
FROM Παίζει  
WHERE Τίτλος = "Gone by the Wind"
```

# select

- Όταν δεν υπάρχει το where, το P θεωρείται ότι ισχύει.

Παράδειγμα: Ονόματα όλων των ηθοποιών που έχουν παίξει σε ταινίες

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)

Παίζει(Όνομα, Τίτλος, Έτος)

Ηθοποιός(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

```
SELECT Όνομα  
FROM Παίζει
```

# select distinct

ΠΡΟΣΟΧΗ: Δε γίνεται απαλοιφή των διπλών εμφανίσεων.

- Η SQL επιτρέπει πολλαπλές εμφανίσεις της ίδιας πλειιάδας σε μια σχέση. Μια σχέση στην SQL είναι ένα πολυσύνολο (multiset) ή θύλακας (bag).

Απαλοιφή διπλών εμφανίσεων

```
SELECT DISTINCT Όνομα  
FROM Παίζει
```

# select \*

Επιλογή όλων των γνωρισμάτων

```
SELECT *  
FROM Παίζει
```

*Η «μικρότερη» SQL ερώτηση (μας δίνει το περιεχόμενο του αντίστοιχου πίνακα)*

# select

Αριθμητικές πράξεις (+, -, \*, /) ανάμεσα σε σταθερές ή γνωρίσματα πλειάδων

```
SELECT Τίτλος, Έτος, Διάρκεια/60, Είδος  
FROM Ταινία
```

Επιστρέφει μια σχέση ίδια με τη σχέση Ταινία μόνο που το γνώρισμα διάρκεια μας δίνει τις ώρες (έχει διαιρεθεί με το 60)

# where

## Συνθήκη του where

Λογικοί τελεστές: **and, or, not**

Τελεστές σύγκρισης: **<, <=, >, >=, =, <>, between, not between**

ανάμεσα σε αριθμητικές εκφράσεις, συμβολοσειρές (strings), και ειδικούς τύπους.

# Παράδειγμα

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)

Παίζει(Όνομα, Τίτλος, Έτος)

Ηθοποιός(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

Τον τίτλο όλων των ταινιών που γυρίστηκαν μετά το 1995 και είναι ασπρόμαυρες

```
SELECT Τίτλος  
FROM Ταινία  
WHERE Έτος > 1995 AND Είδος = "Ασπρόμαυρη"
```



# Παράδειγμα

Χρήση του between :

```
SELECT Τίτλος  
FROM Ταινία  
WHERE Έτος BETWEEN 1990 AND 1995
```

αντί του

```
SELECT Τίτλος  
FROM Ταινία  
WHERE Έτος >= 1990 AND Έτος <= 1995
```

# Βασική Δομή

- Όταν το ίδιο γνώρισμα εμφανίζεται στο σχήμα περισσότερων από μια σχέσεων, τότε διάκριση βάση του συμβολισμού:

<όνομα-σχέσης>. <όνομα-γνωρίσματος>

# Παράδειγμα

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)

Παίζει(Όνομα, Τίτλος, Έτος)

Ηθοποιός(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

Παράδειγμα *φυσικής συνένωσης*:

Τους ηθοποιούς (το όνομα τους) που γεννήθηκαν πριν το 1950 και έπαιξαν σε ταινίες μετά το 2010

```
SELECT DISTINCT Όνομα  
FROM Παίζει, Ηθοποιός  
WHERE Ηθοποιός.Έτος-Γέννησης < 1950 AND  
Παίζει.Έτος > 2010 AND  
Ηθοποιός.Όνομα = Παίζει.Όνομα
```

Προσοχή στις συνθήκες συνένωσης

# Παράδειγμα

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)

Παίζει(Όνομα, Τίτλος, Έτος)

Ηθοποιός(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

Παράδειγμα φυσικής συνένωσης:

Τους ηθοποιούς που παίζουν σε ασπρόμαυρες ταινίες

**SELECT DISTINCT** Όνομα

**FROM** Παίζει, Ταινία

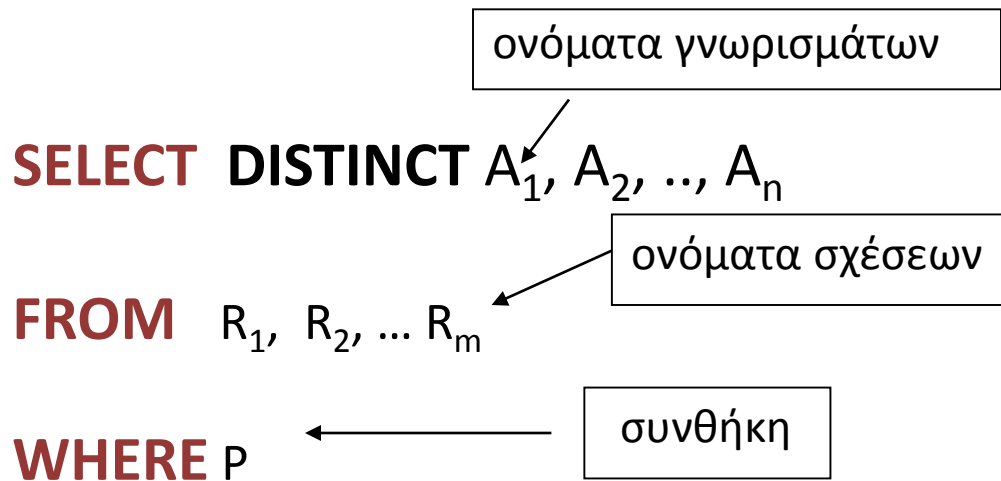
**WHERE** Είδος = "Ασπρόμαυρη" **AND**

Παίζει.Τίτλος = Ταινία.Τίτλος **AND** Παίζει.Έτος = Ταινία.Έτος

Συνθήκη συνένωσης

# Βασική Δομή (επανάληψη)

Η βασική δομή μιας ερώτησης σε SQL έχει την εξής μορφή:



Ισοδύναμο του:  $\pi_{A_1, A_2, \dots, A_n} (\sigma_P (R_1 \times R_2 \times \dots \times R_m))$

# Βασική Δομή (επανάληψη)

## Select

- ✓ Διαγραφή διπλότιμων: `select distinct`
- ✓ `select *` (όλα τα γνωρίσματα)

## Συνθήκη του `where`

Λογικοί τελεστές: `and`, `or`, `not`

Τελεστές σύγκρισης: `<`, `<=`, `>`, `>=`, `=`, `<>`, `between`, `not between`

ανάμεσα σε αριθμητικές εκφράσεις, συμβολοσειρές (`strings`), και ειδικούς τύπους.

Τα αποτελέσματα μιας ερώτησης ΔΕΝ αποθηκεύονται

# Παραδείγματα

ΠΙΤΣΑ(ΟΝΟΜΑ, ΣΥΣΤΑΤΙΚΟ)

ΑΡΕΣΕΙ(ΦΟΙΤΗΤΗΣ, ΣΥΣΤΑΤΙΚΟ)

1. Όλα τα συστατικά που αρέσουν σε φοιτητές
2. Τα συστατικά που αρέσουν στον φοιτητή Δημήτρη
3. Τα συστατικά της πίτσας Σπέσιαλ
4. Τις πίτσες που έχουν συστατικά που αρέσουν στον φοιτητή Δημήτρη

# Παράδειγμα

## ΠΙΤΣΑ

ΟΝΟΜΑ	ΣΥΣΤΑΤΙΚΟ
Vegetarian	μανιτάρι
Vegetarian	ελιά
Χαβάρη	ανανάς
Χαβάρη	ζαμπόν
Σπέσιαλ	ζαμπόν
Σπέσιαλ	μπέικον
Σπέσιαλ	μανιτάρι
Ελληνική	ελιά

## ΑΡΕΣΕΙ

ΦΟΙΤΗΤΗΣ	ΣΥΣΤΑΤΙΚΟ
Δημήτρης	μανιτάρι
Κώστας	ζαμπόν
Μαρία	ελιά
Κατερίνα	μανιτάρι
Μαρία	ζαμπόν
Δημήτρης	μπέικον
Μαρία	ανανάς



# SQL

## Περισσότερα για τη γλώσσα ερωτήσεων

- Πράξεις με Συμβολοσειρές
- Διάταξη Πλειάδων
- Αλλαγή Ονόματος
- Μεταβλητές Πλειάδων
- Η τιμή null

# Πράξεις με συμβολοσειρές

Η πιο συνηθισμένη πράξη είναι ταίριασμα προτύπων:

% ταιριάζει οποιαδήποτε συμβολοσειρά

\_ ταιριάζει οποιοδήποτε χαρακτήρα

Γίνεται διάκριση ανάμεσα σε κεφαλαία και μικρά

Σύγκριση χρησιμοποιώντας το like, not like

# Πράξεις με συμβολοσειρές

*Παράδειγμα:*

*Οι τίτλοι όλων των ταινιών που περιέχουν τη λέξη Θάλασσα*

```
SELECT DISTINCT Τίτλος  
FROM Ταινία  
WHERE Τίτλος LIKE '%Θάλασσα%'
```

Πολλές ακόμα πράξεις διαθέσιμες.

# Διάταξη Πλειάδων

Χρήση του `order by` ώστε οι πλειάδες στο αποτέλεσμα να είναι ταξινομημένες με βάση το αντίστοιχο γνώρισμα

```
SELECT DISTINCT Τίτλος, Έτος  
FROM Παίζει  
WHERE Όνομα = 'Robert De Niro'  
ORDER BY Έτος
```

# Διάταξη Πλειάδων

Default: αύξουσα διάταξη

Αλλά και άμεσος προσδιορισμός χρησιμοποιώντας το asc (αύξουσα) ή το desc (φθίνουσα). Επίσης, ταξινόμηση με βάση **πολλά** γνωρίσματα.

Παράδειγμα:

```
SELECT *  
FROM Ταινία  
ORDER BY Έτος DESC, Τίτλος ASC
```

*Η ταξινόμηση είναι δαπανηρή λειτουργία.*

# Περιορισμός μεγέθους αποτελέσματος

Περιορισμό του μεγέθους του αποτελέσματος με χρήση του `limit <k>`

*Σε συνδυασμό ή όχι με το `order by`:*

αν δεν υπάρχει το `order by` το `limit k` μας δίνει κάποιες τυχαίες `k` πλειάδες από το αποτέλεσμα – αν υπάρχει το `order by` μας δίνει τις πρώτες `k`

```
SELECT Τίτλος, Έτος  
FROM Παίζει  
WHERE Όνομα = 'Robert De Niro'  
ORDER BY Έτος DESC  
LIMIT 8
```

8 από τις πιο πρόσφατες -- αν δεν υπάρχει το `order by`, δίνει 8 τυχαίες

# Αλλαγή Ονόματος

Τα ονόματα των γνωρισμάτων στο αποτέλεσμα είναι αυτά των σχέσεων στην ερώτηση.

Δυνατότητα αλλαγής του ονόματος τόσο μιας σχέσης όσο και ενός γνωρίσματος:

<παλιό-όνομα> **as** <νέο-όνομα>

Το as μπορεί να εμφανίζεται στο select ή στο from

# Αλλαγή Ονόματος

Για παράδειγμα:

```
SELECT Τίτλος, Έτος, Διάρκεια/60 AS Ώρες-Διάρκεια, Είδος  
FROM Ταινία
```



# Αλλαγή Ονόματος

Χρήσιμο όταν

(α) όταν έχουμε αριθμητικές εκφράσεις στο select και δεν έχουν όνομα

(β) όταν θέλουμε να αλλάξουμε το όνομα του γνωρίσματος στο αποτέλεσμα

(γ) δυο σχέσεις του from έχουν γνωρίσματα με το ίδιο όνομα

# Μεταβλητές πλειάδων

Μια μεταβλητή πλειάδα μπορεί να οριστεί στο from χρησιμοποιώντας το as:

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)

Παίζει(Όνομα, Τίτλος, Έτος)

Ηθοποιός(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

**SELECT DISTINCT** Όνομα

**FROM** Παίζει **AS** Π, Ταινία **AS** Τ

**WHERE** Π.Τίτλος = Τ.Τίτλος **AND** Π.Έτος = Τ.Έτος **AND** Είδος = 'Ασπρόμαυρη'

# Μεταβλητές πλειάδων

- ✓ Οι μεταβλητές πλειάδων είναι ιδιαίτερα χρήσιμες όταν θέλουμε να συγκρίνουμε δυο πλειάδες της ίδιας σχέσης (με συνένωση - self-join).

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)

Παίζει(Όνομα, Τίτλος, Έτος)

Ηθοποιός(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

*Παράδειγμα: Τα ονόματα όλων των ταινιών που έχουν διάρκεια μεγαλύτερη τουλάχιστον από μία ταινία που γυρίστηκε το 1995*

```
SELECT DISTINCT T.Τίτλος  
FROM Ταινία AS S, Ταινία AS T  
WHERE T.Διάρκεια > S. Διάρκεια AND S.Έτος = 1995
```

# Λογική Τριών Τιμών

Η SQL **λογική τριών τιμών** με τιμές TRUE, FALSE, και ΑΓΝΩΣΤΟ (null)

Στο αποτέλεσμα του select-from-where ανήκουν μόνο οι πλειάδες που ικανοποιούν την συνθήκη του where (**η έκφραση έχει την τιμή TRUE**)

**NOT**

TRUE

FALSE

FALSE

TRUE

ΑΓΝΩΣΤΟ (NULL)

ΑΓΝΩΣΤΟ (NULL)

# Λογική Τριών Τιμών

<b>AND</b>	<b>TRUE</b>	<b>FALSE</b>	<b>UNKNOWN</b>
<b>TRUE</b>	TRUE	FALSE	UNKNOWN
<b>FALSE</b>	FALSE	FALSE	FALSE
<b>UNKOWN</b>	UNKNOWN	FALSE	UNKNOWN

<b>OR</b>	<b>TRUE</b>	<b>FALSE</b>	<b>UNKNOWN</b>
<b>TRUE</b>	TRUE	TRUE	TRUE
<b>FALSE</b>	TRUE	FALSE	UNKNOWN
<b>UNKOWN</b>	TRUE	UNKNOWN	UNKNOWN

$P = Q$ , αν ένα από τα δύο είναι UNKNOWN δίνει UNKNOWN

# Η τιμή null

Χρήση της λέξης κλειδί is null (is not null) σε μια συνθήκη για να ελέγξουμε αν μια τιμή είναι null.

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)

Παίζει(Όνομα, Τίτλος, Έτος)

Ηθοποιός(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

```
SELECT Ονόμα  
FROM Ηθοποιός  
WHERE Διεύθυνση IS NULL
```

# Η τιμή null

## Εμφάνιση null

- Σε αριθμητικές πράξεις: το αποτέλεσμα είναι null όταν οποιαδήποτε τιμή είναι null
  
- Σε συναθροιστικές συναρτήσεις: αγνοείται πλην από το `count(*)`

# Επανάληψη

## Πράξεις με Συμβολοσειρές

Η πιο συνηθισμένη πράξη είναι ταίριασμα προτύπων:

% ταιριάζει οποιαδήποτε συμβολοσειρά

\_ ταιριάζει οποιοδήποτε χαρακτήρα

Σύγκριση χρησιμοποιώντας το like, not like

## Διάταξη των Πλειάδων

Χρήση του order by ώστε οι πλειάδες στο αποτέλεσμα να είναι ταξινομημένες με βάση το αντίστοιχο γνώρισμα

Default: αύξουσα διάταξη, αλλά και άμεσα χρησιμοποιώντας το asc (αύξουσα) ή το desc (φθίνουσα).



# Επανάληψη

- Χρήση του συμβολισμού:

<όνομα-σχέσης>.<όνομα-γνωρίσματος>

- Δυνατότητα **αλλαγής του ονόματος** τόσο μιας σχέσης όσο και ενός γνωρίσματος:

<παλιό-όνομα> as <νέο-όνομα>

Το as μπορεί να εμφανίζεται στο select ή στο from

- ✓ Οι **μεταβλητές πλειάδων** (as στο from) είναι ιδιαίτερα χρήσιμες

# Βασική Δομή Ερώτησης

```
SELECT [DISTINCT] A1, A2, ..., An  
FROM R1, R2, ... Rm  
WHERE P  
ORDER BY Ai  
LIMIT k
```

# Πράξεις Συνόλου

# Πράξεις Συνόλου

Πράξεις:

- union (ένωση)
- intersect (τομή)
- except (διαφορά)

εφαρμόζονται σε συμβατές σχέσεις.

# Γενική Σύνταξη

(SELECT  
FROM  
WHERE )

union/intersect/except

(SELECT  
FROM  
WHERE )

# Τομή - Παράδειγμα

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)

Παίζει(Όνομα, Τίτλος, Έτος)

Ηθοποιός(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

*Τα ονόματα των ηθοποιών που έπαιξαν σε ταινίες του 2006 και του 2007*

```
(SELECT Όνομα  
FROM Παίζει  
WHERE Έτος = 2006)  
INTERSECT  
(SELECT Όνομα  
FROM Παίζει  
WHERE Έτος = 2007)
```

# Τομή

Απαλοιφή διπλών εμφανίσεων,

εκτός αν χρησιμοποιηθεί το **intersect all**

Μέγιστος αριθμός πολλαπλών εμφανίσεων;

# Ένωση

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)

Παίζει(Όνομα, Τίτλος, Έτος)

Ηθοποιός(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

```
(SELECT Όνομα  
FROM Παίζει  
WHERE Έτος = 2006)
```

**UNION**

```
(SELECT Όνομα  
FROM Παίζει  
WHERE Έτος = 2007)
```

Αντίστοιχα:

**UNION ALL**

Μέγιστος αριθμός πολλαπλών εμφανίσεων;



# Διαφορά

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)

Παίζει(Όνομα, Τίτλος, Έτος)

Ηθοποιός(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

```
(SELECT Όνομα  
FROM Παίζει  
WHERE Έτος = 2006)
```

**EXCEPT**

```
(SELECT Όνομα  
FROM Παίζει  
WHERE Έτος = 2007)
```

Αντίστοιχα:

**EXCEPT ALL**

Μέγιστος αριθμός πολλαπλών εμφανίσεων;

# Παραδείγματα

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)

Παίζει(Όνομα, Τίτλος, Έτος)

Ηθοποιός(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

1. Ηθοποιούς που δεν έπαιξαν σε έγχρωμη ταινία
2. Τις ταινίες (τίτλο) με τον ίδιο τίτλο που γυρίστηκαν το 2005 και το 2006 (δώστε δυο ερωτήσεις μια με πράξη συνόλου και μια χωρίς)

# Επανάληψη

Πράξεις:

- union
- intersect
- except (minus)

εφαρμόζονται σε συμβατές σχέσεις (ΠΡΟΣΟΧΗ: πρακτικά τα ΙΔΙΑ ΓΝΩΡΙΣΜΑΤΑ (ίδιο αριθμό και τύπο γνωρισμάτων) στα δύο select)

Σύνταξη,

(select-from-where) union/intersect/except (select-from-where)

**Απαλοιφή διπλών εμφανίσεων**, εκτός αν χρησιμοποιηθεί το union {intersect, except} all

# Υποερωτήσεις

# Υποερωτήσεις

Η SQL επιτρέπει το φώλιασμα υπο-ερωτήσεων.

Μια **υπο-ερώτηση** είναι μια έκφραση `select-from-where` που χρησιμοποιείται μέσα σε μια άλλη `select-from-where` ερώτηση ως συνθήκη στο `where`.

# Σύνταξη

SELECT ...

FROM ...

WHERE

<τελεστής> (SELECT ...  
FROM ...  
WHERE ... );

Υπο-ερώτηση



Η εσωτερική (φωλιασμένη) υπο-ερώτηση υπολογίζεται για κάθε γραμμή (πλειάδα) της εξωτερικής ερώτησης

Στη συνέχεια θα δούμε τι μπορεί να είναι ο **τελεστής**

# Ο τελεστής in (not in)

Ελέγχει αν μια **πλειάδα ανήκει (δεν ανήκει)** σε ένα σύνολο από πλειάδες που έχουν προκύψει από μια έκφραση select-from-where.

Γενική δομή:

SELECT ...

FROM ...

WHERE

T **IN (NOT IN)** (SELECT...  
FROM ...  
WHERE ... );

T: πλειάδα

# Ο τελεστής in (not in)

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)

Παίζει(Όνομα, Τίτλος, Έτος)

Ηθοποιός(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

*Παράδειγμα: Τα ονόματα όλων των ηθοποιών που δεν έχουν παίξει σε ταινία*

```
SELECT distinct Ηθοποιός.Όνομα  
FROM Ηθοποιός  
WHERE Ηθοποιός.Όνομα not in
```

```
(SELECT Όνομα  
FROM Παίζει)
```



# Ο τελεστής in (not in)

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)

Παίζει(Όνομα, Τίτλος, Έτος)

Ηθοποιός(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

*Παράδειγμα: Τα ονόματα όλων των ηθοποιών που έπαιξαν σε ασπρόμαυρη ταινία*

```
SELECT DISTINCT Παίζει.Όνομα  
FROM Παίζει  
WHERE (Παίζει.Τίτλος, Παίζει.Έτος) IN
```

```
(SELECT Ταινία.Τίτλος, Ταινία.Έτος  
FROM Ταινία  
WHERE Είδος = 'Ασπρόμαυρη')
```

# Ο τελεστής in (not in)

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)

Παίζει(Όνομα, Τίτλος, Έτος)

Ηθοποιός(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

*Παράδειγμα: Τον τίτλο όλων των ταινιών με διάρκεια πάνω από 100 λεπτά για τις οποίες υπάρχει ταινία με το ίδιο τίτλο και διάρκεια μικρότερη από 60 λεπτά*

```
SELECT DISTINCT Τίτλος  
FROM Ταινία  
WHERE Διάρκεια > 100  
AND Τίτλος IN
```

```
(SELECT Τίτλος  
FROM Ταινία  
WHERE Διάρκεια < 60)
```

Τροποποίηση ώστε η ταινία με διάρκεια < 60 να είναι διαφορετική

*Παράδειγμα: Η ίδια ερώτηση με πράξη συνόλου και με συνένωση*

# Ο τελεστής in (not in)

Μπορεί να χρησιμοποιηθεί και με *enumerated* σύνολα

*Παράδειγμα: Τους τίτλους όλων των ταινιών που δεν γυρίστηκαν το 2006 και το 2007.*

```
SELECT Τίτλος  
FROM Ταινία  
WHERE Έτος NOT IN (2006, 2007)
```

# Σύγκριση με (τιμές) Συνόλου: some

Ο τελεστής **some (any)** έχει τη σημασία του *τουλάχιστον ένα* από ένα σύνολο

Γενική δομή:

SELECT ...

FROM ...

WHERE

T > **SOME** (SELECT ...  
FROM ...  
WHERE ... );

T: πλειάδα

# some

*Παράδειγμα: Τους τίτλους όλων των ταινιών που γυρίστηκαν αργότερα από τουλάχιστον μια ασπρόμαυρη ταινία*

```
SELECT DISTINCT Τίτλος  
FROM Ταινία  
WHERE Έτος >SOME (SELECT Έτος  
FROM Ταινία  
WHERE Είδος = 'Ασπρόμαυρη')
```

# some

Επίσης:

<some,

<=some,

>=some,

=some (ισοδ. του in)

<>some (όχι ισοδ. του not in)

# Σύγκριση με Σύνολο: all

Ο τελεστής **all** έχει τη σημασία από όλα τα στοιχεία ενός συνόλου

*Παράδειγμα: Τους τίτλους όλων των ταινιών που γυρίστηκαν αργότερα από όλες τις ασπρόμαυρες ταινίες*

```
SELECT DISTINCT Τίτλος  
FROM Ταινία  
WHERE Έτος >ALL (SELECT Έτος  
                 FROM Ταινία  
                 WHERE Είδος = 'Ασπρόμαυρη')
```

# all

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)

Παίζει(Όνομα, Τίτλος, Έτος)

Ηθοποιός(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

*Παράδειγμα:* Τι υπολογίζει το παρακάτω;

```
SELECT Όνομα
FROM Ηθοποιός
WHERE Έτος-Γέννησης <=all (SELECT Έτος-Γέννησης
                             FROM Παίζει, Ηθοποιός
                             WHERE Παίζει.Όνομα = Ηθοποιός.Όνομα
                             AND Τίτλος = 'Μανταλένα')
```



# all

Επίσης:

<all,

<=all,

>=all,

=all,

<>all (ισοδ. του not in)

# Ο τελεστής exists (not exists)

Έλεγχος για άδεια σχέση:

Ο τελεστής **exists** (**not exists**): επιστρέφει true αν η υποερώτηση δεν είναι κενή (είναι κενή)

Γενική δομή:

SELECT ...

FROM ...

WHERE

**EXISTS (NOT EXISTS)** (SELECT ...  
FROM ...  
WHERE ... );

# Ο τελεστής exists (not exists)

*Παράδειγμα: Οι ασπρόμαυρες ταινίες με τουλάχιστον ένα ηθοποιό*

```
SELECT T.Τίτλος, T.Έτος
FROM   Ταινία as T
WHERE  T.είδος = 'Ασπρόμαυρη' AND
       EXISTS (SELECT *
              FROM Παίζει
              WHERE Παίζει.Τίτλος = T.Τίτλος AND Παίζει.Έτος = T.Έτος)
```

# Ο τελεστής exists (not exists)

Ο τελεστής not exists μπορεί να χρησιμοποιηθεί για έλεγχο αν η σχέση  $A$  περιέχει τη σχέση  $B$  (σχέση υπερσυνόλου/υποσυνόλου)

not exists (B except A)

*True if and only if  $A \supseteq B$*

# Ο τελεστής exists (not exists)

## *Ερώτηση*

Πως μπορεί να χρησιμοποιηθεί για να υπολογίσουμε τη «διαίρεση»;

# Παράδειγμα: Διαίρεση

Τις πίτσες που έχουν όλα τα συστατικά που αρέσουν στον Δημήτρη

ΠΙΤΣΑ(ΟΝΟΜΑ, ΣΥΣΤΑΤΙΚΟ)

ΑΡΕΣΕΙ(ΦΟΙΤΗΤΗΣ, ΣΥΣΤΑΤΙΚΟ)

*ΙΔΕΑ*

Θέλουμε τις πίτσες που τα συστατικά τους είναι υπερσύνολο των συστατικών που αρέσουν στο Δημήτρη

A: Συστατικά πίτσας Π

B: Συστατικά που αρέσουν στο Δημήτρη

not exists (B except A)

# Παράδειγμα Διαίρεσης

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)

Παίζει(Όνομα, Τίτλος, Έτος)

Ηθοποιός(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

Παράδειγμα: Οι ηθοποιοί που έχουν παίξει σε όλες τις ταινίες του George Clooney

B: όλες οι ταινίες του George Clooney

not exists (B except A)

A: όλες οι ταινίες του συγκεκριμένου ηθοποιού

```
SELECT distinct S.Όνομα
```

```
FROM Παίζει AS S
```

```
WHERE NOT EXISTS ((SELECT Τίτλος, Έτος  
FROM Παίζει  
WHERE Όνομα = 'George Clooney')  
EXCEPT  
(SELECT Τίτλος, Έτος  
FROM Παίζει as R  
WHERE R.Όνομα = S.Όνομα))
```

υπολογισμός για  
κάθε S

Τέτοιου είδους μεταβλητές δεν  
υπάρχουν στη σχεσιακή άλγεβρα

# Ο τελεστής unique (not unique)

Έλεγχος για Διπλές Εμφανίσεις

Ο τελεστής **unique**: επιστρέφει true αν η υποερώτηση δεν έχει πολλαπλές όμοιες πλειάδες – not unique

Γενική δομή:

SELECT ...

FROM ...

WHERE

**UNIQUE (NOT UNIQUE)** (SELECT ...  
FROM ...  
WHERE ... );

*Μπορεί να χρησιμοποιηθεί για να ελεγχθεί αν το αποτέλεσμα είναι σύνολο ή πολυσύνολο*



# Ο τελεστής unique (not unique)

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)  
Παίζει(Όνομα, Τίτλος, Έτος)  
Ηθοποιός(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

*Παράδειγμα: Οι ηθοποιοί που έχουν παίξει ακριβώς σε μια ταινία*

```
SELECT Όνομα  
FROM Παίζει AS T  
WHERE UNIQUE (SELECT R.Όνομα  
                FROM Παίζει AS R  
                WHERE T.Όνομα = R.Όνομα)
```

```
SELECT Όνομα      (θα το δούμε στη συνέχεια)  
FROM Παίζει  
GROUP BY Όνομα  
HAVING COUNT(*) = 1
```

# Ο τελεστής unique (not unique)

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)

Παίζει(Όνομα, Τίτλος, Έτος)

Ηθοποιός(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

*Παράδειγμα: Οι ηθοποιοί που έχουν παίξει τουλάχιστον σε δύο ταινίες*

```
SELECT Όνομα
FROM Παίζει AS T
WHERE NOT UNIQUE (SELECT R.Όνομα
                  FROM Παίζει AS R
                  WHERE T.Όνομα = R.Όνομα)
```

```
SELECT Όνομα      (θα το δούμε στη συνέχεια)
FROM Παίζει
GROUP BY Όνομα
HAVING COUNT(*) > 1
```

# Επανάληψη

Ο τελεστής μπορεί να είναι:

- T in/not in (συμμετοχή σε σύνολο)
- T (>, =, κλπ) some-any/all (σύγκριση με στοιχεία συνόλου)
- exists/not exists (έλεγχος για κενά σύνολα)
- unique/not unique (έλεγχος για διπλότιμα)

# Επανάληψη

Η SQL επιτρέπει το φώλιασμα υπο-ερωτήσεων.

Μια υπο-ερώτηση είναι μια έκφραση select-from-where που χρησιμοποιείται μέσα σε μια άλλη ερώτηση.

## Γενική δομή:

SELECT ...

FROM ...

WHERE **<x>**

(SELECT ...

FROM ...

WHERE ... );

**<x>** μπορεί να είναι

$T$  {=, <, <=, >, >=, <>} any(some), all

$T$  in

exists, unique

(όπου  $T$  πλειάδα)

Δηλαδή διατυπώνονται ως συνθήκες στο where

Υπολογισμός της υπο-ερώτησης για κάθε γραμμή (πλειάδα) της εξωτερικής ερώτησης

# Επανάληψη

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)  
Παίζει(Όνομα, Τίτλος, Έτος)  
Ηθοποιός(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

## Παραδείγματα

```
SELECT Ταινία.Τίτλος
FROM Ταινία
WHERE Διάρκεια >SOME (SELECT Διάρκεια
                        FROM Ταινία
                        WHERE Είδος = 'Εγχρωμη')
```

```
SELECT Ταινία.Τίτλος
FROM Ταινία
WHERE Διάρκεια IN (SELECT Διάρκεια
                    FROM Ταινία
                    WHERE Είδος = 'Εγχρωμη')
```

```
SELECT Ταινία.Τίτλος
FROM Ταινία
WHERE Διάρκεια >ALL (SELECT Διάρκεια
                       FROM Ταινία
                       WHERE Είδος = 'Εγχρωμη')
```

# Επανάληψη

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)

Παίζει(Όνομα, Τίτλος, Έτος)

Ηθοποιός(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

## Παραδείγματα

```
SELECT T.Τίτλος
FROM Ταινία AS T
WHERE EXISTS (SELECT *
                FROM Ταινία AS S
                WHERE T.Τίτλος = S.Τίτλος AND
                S.Διάρκεια > T.Διάρκεια)
```

```
SELECT T.Τίτλος
FROM Ταινία AS T
WHERE UNIQUE (SELECT *
                FROM Ταινία AS S
                WHERE T.Τίτλος = S.Τίτλος AND
                S.Διάρκεια > T.Διάρκεια)
```

# Συναθροιστικές Συναρτήσεις

# Συναθροιστικές Συναρτήσεις

Η SQL έχει 5 built-in συναθροιστικές συναρτήσεις:

Μέσος όρος: **avg**(A) (μόνο σε αριθμούς) A γνώρισμα

Ελάχιστο: **min**(A)

Μέγιστο: **max**(A)

Άθροισμα: **sum**(A) (μόνο σε αριθμούς)

Πλήθος: **count**(A)



# Παράδειγμα

*Παράδειγμα: Μέση διάρκεια όλων των έγχρωμων ταινιών*

```
SELECT AVG(Διάρκεια)  
FROM Ταινία  
WHERE Είδος = 'Έγχρωμη'
```

*Το αποτέλεσμα είναι μια σχέση με ένα γνώρισμα και μια γραμμή [μπορούμε να δώσουμε όνομα στο γνώρισμα χρησιμοποιώντας το AS]*

# Συναθροιστικές Συναρτήσεις

*Παράδειγμα: Μέγιστη διάρκεια όλων των έγχρωμων ταινιών και την ταινία με τη μεγαλύτερη διάρκεια!!*

```
SELECT Τίτλος, Έτος, MAX(Διάρκεια)  
FROM Ταινία  
WHERE Είδος = 'Έγχρωμη'
```

Αν το select συναθροιστική, τότε μόνο συναθροιστικές,

- εκτός αν υπάρχει group by - δηλαδή δεν μπορούμε να προβάσουμε και άλλα γνωρίσματα σχέσεων (θα το δούμε στη συνέχεια)

# Συναθροιστικές Συναρτήσεις

Αν θέλουμε να απαλείψουμε διπλές εμφανίσεις χρησιμοποιούμε τη λέξη-κλειδί **distinct** στην αντίστοιχη έκφραση.

```
SELECT SUM(DISTINCT Διάρκεια)  
FROM Ταινία
```

# Συναθροιστικές Συναρτήσεις

Για να μετρήσουμε πόσες πλειάδες έχει μια σχέση:

```
SELECT COUNT(*)  
FROM Ταινία
```

Δε μπορούμε να χρησιμοποιήσουμε το distinct με το count(\*).

# Συναθροιστικές Συναρτήσεις: group by

Μπορούμε να εφαρμόσουμε τις συναρτήσεις όχι μόνο σε ένα σύνολο από πλειάδες, αλλά σε ομάδες από σύνολα πλειάδων. Οι ομάδες προσδιορίζονται χρησιμοποιώντας το **group by**

*Παράδειγμα: Μέση διάρκεια ταινίας ανά είδος*

```
SELECT Είδος, avg(Διάρκεια)  
FROM Ταινία  
GROUP BY Είδος
```

Στο select και το όνομα του group by

# Συναθροιστικές Συναρτήσεις: group by

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)

Παίζει(Όνομα, Τίτλος, Έτος)

Ηθοποιός(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

Η ομαδοποίηση μπορεί να γίνει ως προς περισσότερα του ενός πεδία.

```
SELECT Τίτλος, Έτος, COUNT(Όνομα)  
FROM Παίζει  
GROUP BY Τίτλος, Έτος
```

# Συναθροιστικές Συναρτήσεις: having

Μπορούμε να εφαρμόσουμε μια συνθήκη σε μια συγκεκριμένη ομάδα από πλειάδες χρησιμοποιώντας το **having**

```
SELECT Έτος, COUNT(Τίτλος)
FROM Ταινία
GROUP BY Έτος
HAVING AVG(Διάρκεια) > 100
```

Η συνθήκη του **having** εφαρμόζεται **αφού** σχηματιστούν οι ομάδες και υπολογιστούν οι συναθροιστικές συναρτήσεις.

# Συναθροιστικές Συναρτήσεις

Όταν εμφανίζονται και το `where` και το `having`:

- η συνθήκη του `where` εφαρμόζεται πρώτα,
- οι πλειάδες που ικανοποιούν αυτή τη συνθήκη τοποθετούνται σε ομάδες με βάση το `group by`
- και μετά αν υπάρχει συνθήκη στο `having` εφαρμόζεται στις ομάδες και επιλέγονται όσες ικανοποιούν τη συνθήκη



## Συναθροιστικές Συναρτήσεις

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)

Παίζει(Όνομα, Τίτλος, Έτος)

Ηθοποιός(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

*Παράδειγμα: Αριθμό ταινιών που έπαιξε κάθε ηθοποιός που γεννήθηκε μετά το 1987 αν αυτός είναι μεγαλύτερος του 5*

```
SELECT  Ηθοποιός, count(*)4
FROM  Παίζει, Ηθοποιός
1 WHERE  Παίζει.Όνομα = Ηθοποιός.Όνομα  AND  Έτος-Γέννησης > 1987
2 GROUP BY Ηθοποιός.Όνομα
3 HAVING COUNT(*) >= 5
```

# Επανάληψη

Μέσος όρος: **avg** (μόνο σε αριθμούς)

Ελάχιστο: **min**

Μέγιστο: **max**

Άθροισμα: **sum** (μόνο σε αριθμούς)

Πλήθος: **count**

- ✓ Αν θέλουμε να απαλείψουμε διπλές εμφανίσεις χρησιμοποιούμε τη λέξη-κλειδί **distinct** στην αντίστοιχη έκφραση.
- ✓ Μπορούμε να εφαρμόσουμε τις συναρτήσεις όχι μόνο σε ένα σύνολο από πλειάδες, αλλά σε ομάδες από σύνολα πλειάδων. Οι ομάδες προσδιορίζονται χρησιμοποιώντας το **group by**
- ✓ Μπορούμε να εφαρμόσουμε μια συνθήκη σε μια συγκεκριμένη ομάδα από πλειάδες χρησιμοποιώντας το **having**. Η συνθήκη του **having** εφαρμόζεται αφού σχηματιστούν οι ομάδες και υπολογιστούν οι συναθροιστικές συναρτήσεις
- ✓ Οι null τιμές αγνοούνται πλην του **count(\*)**

# Βασική Δομή Ερώτησης

```
SELECT Ai1, Ai2, ..., Ain, ..., avg, ...  
FROM R1, R2, ... Rm  
WHERE P  
GROUP BY Ai1, Ai2, ..., Ain  
HAVING P  
ORDER BY Aj1, Aj2, ..., Ajk
```

# ΣΥΝΕΝΩΣΕΙΣ

# Συνένωση (join)

Η SQL-92 υποστηρίζει διάφορους τύπους συνενώσεων που συνήθως χρησιμοποιούνται στο from, αλλά μπορούν να χρησιμοποιηθούν οπουδήποτε μπορεί να χρησιμοποιηθεί μια σχέση.

Γενική σύνταξη:

```
<όνομα-σχέσης1> <τύπος-συνένωσης> <όνομα-σχέσης2>  
on <συνθήκη-συνένωσης>
```

## Τύποι Συνένωσης:

- [INNER] JOIN
- LEFT [OUTER] JOIN: αριστερή εξωτερική συνένωση
- RIGHT [OUTER] JOIN: δεξιά εξωτερική συνένωση
- FULL [OUTER] JOIN: πλήρης εξωτερική συνένωση

# Παράδειγμα

ΠΙΤΣΑ(ΟΝΟΜΑ, ΣΥΣΤΑΤΙΚΟ)

ΑΡΕΣΕΙ(ΦΟΙΤΗΤΗΣ, ΣΥΣΤΑΤΙΚΟ)

```
SELECT DISTINCT ΑΡΕΣΕΙ. ΦΟΙΤΗΤΗΣ, ΠΙΤΣΑ.ΟΝΟΜΑ  
FROM (ΑΡΕΣΕΙ INNER JOIN ΠΙΤΣΑ  
ON ΠΙΤΣΑ.ΣΥΣΤΑΤΙΚΟ = ΑΡΕΣΕΙ.ΣΥΣΤΑΤΙΚΟ)
```

```
SELECT DISTINCT ΑΡΕΣΕΙ. ΦΟΙΤΗΤΗΣ, ΠΙΤΣΑ.ΟΝΟΜΑ  
FROM ΑΡΕΣΕΙ, ΠΙΤΣΑ  
WHERE ΠΙΤΣΑ.ΣΥΣΤΑΤΙΚΟ = ΑΡΕΣΕΙ.ΣΥΣΤΑΤΙΚΟ
```

# Παράδειγμα

ΠΙΤΣΑ(ΟΝΟΜΑ, ΣΥΣΤΑΤΙΚΟ)

ΑΡΕΣΕΙ(ΦΟΙΤΗΤΗΣ, ΣΥΣΤΑΤΙΚΟ)

```
SELECT DISTINCT ΑΡΕΣΕΙ. ΦΟΙΤΗΤΗΣ, ΠΙΤΣΑ.ΟΝΟΜΑ  
FROM (ΑΡΕΣΕΙ LEFT OUTER JOIN ΠΙΤΣΑ  
      ON ΠΙΤΣΑ.ΣΥΣΤΑΤΙΚΟ = ΑΡΕΣΕΙ.ΣΥΣΤΑΤΙΚΟ)
```

```
SELECT DISTINCT ΑΡΕΣΕΙ. ΦΟΙΤΗΤΗΣ, ΠΙΤΣΑ.ΟΝΟΜΑ  
FROM (ΑΡΕΣΕΙ RIGHT OUTER JOIN ΠΙΤΣΑ  
      ON ΠΙΤΣΑ.ΣΥΣΤΑΤΙΚΟ = ΑΡΕΣΕΙ.ΣΥΣΤΑΤΙΚΟ)
```

# Παράδειγμα

## ΑΡΕΣΕΙ

ΦΟΙΤΗΤΗΣ	ΣΥΣΤΑΤΙΚΟ
Δημήτρης	μανιτάρι
Κώστας	ζαμπόν
Μαρία	ελιά
Κατερίνα	μανιτάρι
Μαρία	ζαμπόν
Δημήτρης	μπέικον
Μαρία	ανανάς
Ανδρόνικος	αντσούγια

## ΠΙΤΣΑ

ΟΝΟΜΑ	ΣΥΣΤΑΤΙΚΟ
Vegetarian	μανιτάρι
Vegetarian	ελιά
Χαβάρη	ανανάς
Χαβάρη	ζαμπόν
Σπέσιαλ	ζαμπόν
Σπέσιαλ	μπέικον
Σπέσιαλ	μανιτάρι
Ελληνική	ελιά
Γιαννιώτικη	μετσοβόνε



# Παράδειγμα

```
SELECT DISTINCT Όνομα  
FROM Παίζει, Ταινία  
WHERE Παίζει.Τίτλος = Ταινία.Τίτλος AND Παίζει.Έτος = Ταινία.Έτος  
and Είδος = "Ασπρόμαυρη"
```

```
SELECT Όνομα  
FROM (Παίζει JOIN Ταινία ON Παίζει.Τίτλος = Ταινία.Τίτλος AND  
      Παίζει.Έτος = Ταινία.Έτος)  
WHERE Είδος = 'Ασπρόμαυρη'
```

# Φυσική Συνένωση (natural join)

τα γνωρίσματα εμφανίζονται στο αποτέλεσμα με την εξής διάταξη: πρώτα αυτά με τα οποία έγινε η συνένωση (δηλ., αυτά που είναι κοινά (έχουν το ίδιο όνομα) και στις δύο σχέσεις), μετά τα υπόλοιπα της πρώτης σχέσης, και τέλος τα υπόλοιπα της δεύτερης σχέσης.

ΠΙΤΣΑ(ΟΝΟΜΑ, ΣΥΣΤΑΤΙΚΟ)

ΑΡΕΣΕΙ(ΦΟΙΤΗΤΗΣ, ΣΥΣΤΑΤΙΚΟ)

```
SELECT ΑΡΕΣΕΙ. ΦΟΙΤΗΤΗΣ, ΠΙΤΣΑ.ΟΝΟΜΑ  
FROM ΠΙΤΣΑ NATURAL JOIN ΑΡΕΣΕΙ
```

```
SELECT ΑΡΕΣΕΙ. ΦΟΙΤΗΤΗΣ, ΠΙΤΣΑ.ΟΝΟΜΑ  
FROM ΠΙΤΣΑ, ΑΡΕΣΕΙ  
WHERE ΠΙΤΣΑ.ΣΥΣΤΑΤΙΚΟ = ΑΡΕΣΕΙ.ΣΥΣΤΑΤΙΚΟ
```

# SFW στο FOR

Μπορούμε να έχουμε μια SFW ερώτηση στο FOR

ΠΙΤΣΑ(ΟΝΟΜΑ, ΣΥΣΤΑΤΙΚΟ)

ΑΡΕΣΕΙ(ΦΟΙΤΗΤΗΣ, ΣΥΣΤΑΤΙΚΟ)

```
SELECT DISTINCT Π.ΟΝΟΜΑ
FROM ΠΙΤΣΑ AS Π,
    ((SELECT ΑΡΕΣΕΙ.ΣΥΣΤΑΤΙΚΟ
    FROM ΑΡΕΣΕΙ
    WHERE ΦΟΙΤΗΤΗΣ = 'ΔΗΜΗΤΡΗΣ')
    EXCEPT
    ((SELECT ΑΡΕΣΕΙ.ΣΥΣΤΑΤΙΚΟ
    FROM ΑΡΕΣΕΙ
    WHERE ΦΟΙΤΗΤΗΣ = 'ΜΑΡΙΑ') AS Τ
WHERE Π.ΣΥΣΤΑΤΙΚΟ = Τ.ΣΥΣΤΑΤΙΚΟ;
```

# Γλώσσα Ενημερώσεις Δεδομένων

# Εισαγωγή

- Γλώσσα Ορισμού (του σχήματος)
- Γλώσσα Χειρισμού Δεδομένων (ΓΧΔ)
  - Γλώσσα Τροποποίησης Δεδομένων (εισαγωγή, διαγραφή, ενημέρωση πλειάδων)
  - Γλώσσα Ερωτήσεων (Query Languages)

# Τροποποίηση ΒΔ

## Τροποποιήσεις

1. Διαγραφή
2. Εισαγωγή
3. Ενημέρωση

Οι εντολές αυτές μεταβάλλουν το στιγμιότυπο της βάσης δεδομένων (δηλαδή, το περιεχόμενο των πινάκων)

*Δείτε και τις σχετικές διαφάνειες προηγούμενου μαθήματος*

# Εισαγωγή δεδομένων

Για να εισάγουμε δεδομένα σε μια σχέση είτε

(α) προσδιορίζουμε την πλειάδα,

```
INSERT INTO R(A1, ..., An) VALUES (v1, ..., vn)
```

είτε

(β) γράφουμε μια ερώτηση που το αποτέλεσμα της εισάγεται στη σχέση.

```
INSERT INTO R(A1, ..., An) SELECT-FROM-WHERE
```

# Εισαγωγή δεδομένων

ΠΙΤΣΑ(ΟΝΟΜΑ, ΣΥΣΤΑΤΙΚΟ)

ΑΡΕΣΕΙ(ΦΟΙΤΗΤΗΣ, ΣΥΣΤΑΤΙΚΟ)

Παράδειγμα

Εισαγωγή μιας πίτσας στη ΠΙΤΣΑ με όνομα «Κατερίνας-special» με συστατικά τα συστατικά που αρέσουν στη φοιτήτρια Κατερίνα

```
INSERT INTO ΠΙΤΣΑ(ΠΙΤΣΑ.ΟΝΟΜΑ, ΠΙΤΣΑ.ΣΥΣΤΑΤΙΚΟ)
SELECT Κατερίνας-Special, ΑΡΕΣΕΙ.ΣΥΣΤΑΤΙΚΟ
FROM ΑΡΕΣΕΙ
WHERE ΑΡΕΣΕΙ.ΦΟΙΤΗΤΗΣ = 'Κατερίνα'
```



# Διαγραφή δεδομένων

Μπορούμε να σβήσουμε μόνο *ολόκληρες* πλειάδες και όχι συγκεκριμένα γνωρίσματα.

```
DELETE FROM R WHERE P
```

Σβήνει όλες τις πλειάδες της R για τις οποίες ισχύει το P.

Όταν λείπει το `where` σβήνονται όλες οι πλειάδες μιας σχέσης.

# Διαγραφή δεδομένων

- Στο from μόνο μια σχέση, αλλά στη συνθήκη του where μπορεί να εμφανίζονται και άλλες
- Σβήνονται «ολόκληρες» πλειάδες
- Αν υπάρχουν παραπάνω από μια πλειάδες που ικανοποιούν τη συνθήκη, δεν υπάρχει τρόπος να διακρίνουμε τις πλειάδες, δηλαδή να σβήσουμε κάποιες
- Πρώτα, υπολογίζεται η συνθήκη του where και μετά διαγράφονται οι πλειάδες που ικανοποιούν τη συνθήκη

```
DELETE FROM Παίζει  
WHERE Τίτλος IN (SELECT Τίτλος  
                FROM Ταινία  
                WHERE Είδος = 'Έγχρωμη')
```

# Διαγραφή δεδομένων

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)

Παίζει(Όνομα, Τίτλος, Έτος)

Ηθοποιός(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

*Παράδειγμα: διαγραφή της ταινίας “The Big Blue” που γυρίστηκε το 1988*

DELETE FROM Ταινία

WHERE Τίτλος = ‘The Big Blue’ AND Έτος = 1988

*Το αποτέλεσμα εξαρτάται από το είδος περιορισμού αναφοράς που έχουμε ορίσει*

Αν δεν έχουμε ορίσει κάποια ειδική ενέργεια “on delete” πρέπει πρώτα να διαγράψουμε και τις εγγραφές του πίνακα Παίζει που σχετίζονται με την ταινία “The Big Blue”:

DELETE FROM Παίζει

WHERE Τίτλος = ‘The Big Blue’ AND Έτος = 1988

# Ενημέρωση

```
UPDATE R
SET Attr = New_Value
WHERE P
```

Παράδειγμα: Αύξηση τις διάρκειας κάθε ταινίας κατά 10 λεπτά για όλες τις ταινίες με διάρκεια < 100

```
UPDATE Ταινία
SET Διάρκεια = Διάρκεια + 10
WHERE Διάρκεια < 100
```

# Ενημέρωση

Όπως και για τη διαγραφή:

- Στο update μόνο μια σχέση, αλλά στη συνθήκη του where μπορεί να εμφανίζονται και άλλες
- Αν υπάρχουν παραπάνω από μια πλειάδες που ικανοποιούν τη συνθήκη, δεν υπάρχει τρόπος να διακρίνουμε τις πλειάδες, δηλαδή να ενημερώσουμε κάποιες
- Πρώτα, υπολογίζεται η συνθήκη του where και μετά ενημερώνονται οι πλειάδες που ικανοποιούν τη συνθήκη – δηλαδή, η συνθήκη υπολογίζεται στο τρέχων στιγμιότυπο – όχι στο τροποποιημένο

# Επανάληψη

## 1. Εισαγωγές

```
insert into R(A1, ..., An) values (v1, ..., vn)
```

```
insert into R(A1, ..., An) select-from-where
```

## 2. Διαγραφές

```
delete from R where P
```

## 3. Ενημερώσεις/Τροποποιήσεις

```
update R  
set Attr = New_Value  
where P
```

# Όψεις

# Ορισμός Όψεων (εικονικών πινάκων)

Μπορούμε να ορίσουμε μια όψη χρησιμοποιώντας την εντολή:

Ορισμός  
Όψης



```
CREATE VIEW <όνομα--όψης> AS <SELECT-FROM-WHERE ερώτηση>
```

Επίσης, μπορούν να προσδιοριστούν τα ονόματα των γνωρισμάτων άμεσα

```
CREATE VIEW <όνομα--όψης> (<λίστα ονομάτων-γνωρισμάτων>)  
AS <SELECT-FROM-WHERE ερώτηση>
```



# Διαφορά από create table

- **Αποθηκεύετε** ο ορισμός
- Μπορεί να χρησιμοποιηθεί όπου ένας πίνακας, αλλά η όψη (δηλαδή, το περιεχόμενο του πίνακα) *υπολογίζεται εκ νέου* κάθε φορά
- Χρήση: Σε ερωτήματα που υπολογίζονται συχνά ή για έλεγχο πρόσβασης

# Παράδειγμα

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)

Παίζει(Όνομα, Τίτλος, Έτος)

Ηθοποιός(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

```
CREATE VIEW Ασπρόμαυρη AS  
SELECT Τίτλος, Έτος  
FROM Ταινία  
WHERE Είδος = 'Ασπρόμαυρη'
```

Base relations/tables

Βασική Σχέση

# Ενημερώσιμες Όψεις

- Για ενημερώσεις ισχύουν περιορισμοί -- Τροποποιήσεις μέσω όψεων
  - **Ενημερώσιμες** όψεις - updatable
    - ένα μόνο πίνακα, πρωτεύον κλειδί της βασικής σχέσης και τιμές για όλα τα *not null* γνωρίσματα χωρίς default τιμή (select, project)
  - Υλοποιημένη (materialized) όψη

# Παράδειγμα

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)

Παίζει(Όνομα, Τίτλος, Έτος)

Ηθοποιός(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

```
CREATE VIEW Στατιστικά-Ηθοποιού (Όνομα-Ηθοποιού, Πλήθος-Ταινιών) AS
SELECT Παίζει.Όνομα, COUNT(*)
FROM Παίζει
GROUP BY Παίζει.Όνομα
```

Μη ενημερώσιμη!

# Διαγραφή όψης

- Ο ορισμός της όψης παραμένει στην βάση δεδομένων, εκτός αν σβηστεί:

`DROP VIEW <όνομα-όψης>`

# Ερωτήσεις;