

# Η Γλώσσα SQL

## Εισαγωγή

Πως θα υλοποιήσουμε (προγραμματίσουμε) την εφαρμογή μας χρησιμοποιώντας ένα σχεσιακό ΣΔΒΔ:

- Γλώσσα Ορισμού (του σχήματος)
- Γλώσσα Χειρισμού Δεδομένων
  - Γλώσσα Τροποποίησης (εισαγωγή, διαγραφή πλειάδων)
  - Γλώσσα Ερωτήσεων

## Εισαγωγή

### Τυπικές (Formal) Γλώσσες (Επε)-Ερωτήσεων

Σχεσιακή άλγεβρα (στην «καθαρή» μορφή της, μόνο γλώσσα ερωτήσεων) - την είδαμε στα προηγούμενα μαθήματα

Σχεσιακός λογισμός (πλειάδων και πεδίου) - Θα τη δούμε σύντομα

### Εμπορικές Γλώσσες Προγραμματισμού

SQL - ΣΗΜΕΡΑ!

QBE

## Η γλώσσα SQL

- Η "standard" γλώσσα για σχεσιακές βάσεις δεδομένων.
- αρχικά *Sequel* στην IBM ως μέρος του **System R**, τώρα *SQL* (Structured Query Language)
- SQL--89, SQL--92, SQL-99

## Η γλώσσα SQL

Η SQL έχει διάφορα τμήματα:

- Γλώσσα Ορισμού Δεδομένων (ΓΟΔ)
- Γλώσσα Χειρισμού Δεδομένων (ΓΧΔ)
- Ενσωματωμένη Γλώσσα Χειρισμού Δεδομένων
- Ορισμό Όψεων
- Εξουσιοδότηση (authentication)
- Ακεραιότητα
- Έλεγχο Συναλλαγών

## Η γλώσσα SQL

SQL αποτελείται από:

**DDL** (Data Definition Language) - ορισμός, δημιουργία, τροποποίηση και διαγραφή *σχήματος*.

**DML** (Data Manipulation Language) - ορισμός, δημιουργία, τροποποίηση, διαγραφή και *επιλογή δεδομένων (γλώσσα ερωτήσεων)*.

**Προδιαγραφές ασφάλειας** - χρήστες και δικαιώματα.

Θα αρχίσουμε από τη γλώσσα ερωτήσεων  
(ερωτήσεις πάνω στο τρέχον στιγμιότυπο της  
βάσης δεδομένων)

## Η Γλώσσα Βάσεων Δεδομένων SQL

Μέρος 1 - Γλώσσα ερωτήσεων: (Βασική Δομή, Πράξεις  
Συνόλου, Ξηλωσμένες Υπο-ερωτήσεις, Συνάθροιση)

Μια χαρακτηριστική ερώτηση σε SQL έχει την εξής μορφή:

$$\text{select } A_1, A_2, \dots, A_n$$

$$\text{from } R_1, R_2, \dots, R_m$$

$$\text{where } P$$

ονόματα γνωρισμάτων  
 ονόματα σχέσεων  
 συνθήκη

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

$$\text{select } A_1, A_2, \dots, A_n$$

$$\text{from } R_1, R_2, \dots, R_m$$

$$\text{where } P$$

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

select αντιστοιχεί στην πράξη της προβολής της  
σχεσιακής άλγεβρας

Ποια γνωρίσματα θέλουμε να υπάρχουν στο  
αποτέλεσμα της ερώτησης.

$$\text{select } A_1, A_2, \dots, A_n$$

$$\text{from } R_1, R_2, \dots, R_m$$

$$\text{where } P$$

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

from αντιστοιχεί στην πράξη του καρτεσιανού  
γινομένου της σχεσιακής άλγεβρας.

Ποιες σχέσεις θα χρησιμοποιηθούν για τον  
υπολογισμό του αποτελέσματος.

$$\text{select } A_1, A_2, \dots, A_n$$

$$\text{from } R_1, R_2, \dots, R_m$$

$$\text{where } P$$

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

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

Το κατηγορήμα P έχει γνωρίσματα των σχέσεων που  
εμφανίζονται στο from.

```

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

Παράδειγμα:

Ονόματα ηθοποιών που παίζουν στην ταινία Gone by the Wind

```

select Όνομα
from Παίζει
where Τίτλος = 'Gone by the Wind'
    
```

• Όταν δεν υπάρχει το **where**, το P θεωρείται ότι ισχύει.

```

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

Παράδειγμα: Ονόματα όλων των ηθοποιών που έχουν παίξει σε ταινίες

```

select Όνομα
from Παίζει
    
```

• **ΠΡΟΣΟΧΗ:** Δε γίνεται απαλοιφή των διπλών εμφανίσεων.

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

Απαλοιφή διπλών εμφανίσεων

```

select distinct Όνομα
from Παίζει
    
```

Επιλογή όλων των γνωρισμάτων

```

select *
from Παίζει
    
```

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

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 διααιρεθεί με το 60)

Συνθήκη του where

Λογικοί τελεστές: **and, or, not**

Τελεστές σύγκρισης: <, <=, >, >=, =, <>

**between, not between**

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

Where

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)  
 Παιζεί(Όνομα, Τίτλος, Έτος)  
 Ηθοποιοί(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

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

```
select Τίτλος
from Ταινία
where Έτος > 1995 and Είδος = "Ασπρόμαυρη"
```

Βάσεις Δεδομένων 2008-2009
Ευαγγελία Πιτουρά
19

Where

Παράδειγμα χρήσης του between :

```
select Τίτλος
from Ταινία
where Έτος between 1990 and 1995
```

αντί του

```
select Τίτλος
from Ταινία
where Έτος >= 1990 and Έτος <= 1995
```

Βάσεις Δεδομένων 2008-2009
Ευαγγελία Πιτουρά
20

Βασική Δομή

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

<όνομα-σχέσης>. <όνομα-γνωρίσματος>

Βάσεις Δεδομένων 2008-2009
Ευαγγελία Πιτουρά
21

Βασική Δομή

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)  
 Παιζεί(Όνομα, Τίτλος, Έτος)  
 Ηθοποιοί(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

Παράδειγμα φυσικής συνένωσης:  
Τους ηθοποιούς που παίζουν σε ασπρόμαυρες ταινίες

```
select distinct Όνομα
from Παιζεί, Ταινία
where Παιζεί.Τίτλος = Ταινία.Τίτλος and Παιζεί.Έτος =
Ταινία.Έτος and Είδος = "Ασπρόμαυρη"
```

Προσοχή στις συνθήκες

Βάσεις Δεδομένων 2008-2009
Ευαγγελία Πιτουρά
22

Παράδειγμα

ΠΡΟΤΙΜΑ(ΠΟΤΗΣ, ΜΠΥΡΑ)  
 ΣΥΧΝΑΖΕΙ(ΠΟΤΗΣ, ΜΑΓΑΖΙ)  
 ΣΕΡΒΙΡΕΙ(ΜΑΓΑΖΙ, ΜΠΥΡΑ)

- Μαγαζί που σερβίρει μπίρες Heineken
- Τα ονόματα όσων συχνάζουν σε μαγαζιά που σερβίρουν μπίρες Heineken

Βάσεις Δεδομένων 2008-2009
Ευαγγελία Πιτουρά
23

Βασική Δομή

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)  
 Παιζεί(Όνομα, Τίτλος, Έτος)  
 Ηθοποιοί(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

Παράδειγμα

*Οι ηθοποιοί που γεννήθηκαν μετά το 1935 και έπαιξαν σε ασπρόμαυρες ταινίες πριν το 1945*

Βάσεις Δεδομένων 2008-2009
Ευαγγελία Πιτουρά
24



• Περισσότερα για τη γλώσσα ερωτήσεων

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



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



Παράδειγμα:

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

```
select distinct Τίτλος
from Ταινία
where Τίτλος like "%Θάλασσα%"
```

Πολλές ακόμα πράξεις διαθέσιμες.



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

```
select distinct Ταινία, Έτος
from Παίζει
where Όνομα = "Robert De Niro"
order by Έτος
```



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.*



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

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

<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 Ωρες-Διάρκεια, Είδος
from Ταινία
```

Σημείωση: τα αποτελέσματα μιας ερώτησης δεν «αποθηκεύονται»

## 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 πλειάδας 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
```

## Η τιμή null

### Η τιμή null

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

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

Παράδειγμα (NOT)

TRUE	FALSE
FALSE	TRUE
ΑΓΝΩΣΤΟ	ΑΓΝΩΣΤΟ

## Η τιμή null

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

```
select Αριθμός-Δανείου
from Δάνειο
where Ποσό is null
```

## Η τιμή null

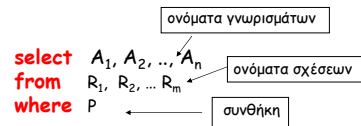
### Εμφάνιση null

- Σε αριθμητικές πράξεις: το αποτέλεσμα είναι null όταν οποιαδήποτε τιμή είναι null
- Σε συγκρίσεις: σύγκριση με null συνήθως δίνει αποτέλεσμα false
- Σε συναθροιστικές συναρτήσεις: αγνοείται πλην από το count(\*)

Παράδειγμα:  
`select sum(Ποσό)  
from Δάνειο`

## Βασική Δομή (επανάληψη)

Μια χαρακτηριστική ερώτηση 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.

## Βασική Δομή (επανάληψη)

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

`<όνομα-σχέσης>.<όνομα-γνωρίματος>`

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

`<παλιό-όνομα> as <νέο-όνομα>`

Το `as` μπορεί να εμφανίζεται στο `select` ή στο `from`

- Οι **μεταβλητές πλειάδων** (`as` στο `from`) είναι ιδιαίτερα χρήσιμες όταν θέλουμε να συγκρίνουμε δυο πλειάδες τις ίδιες σχέσης.

## Βασική Δομή (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).

## Βασική Δομή (επανάληψη)

```
select A1, A2, ..., An  
from R1, R2, ..., Rm  
where P  
order by
```

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

ΠΡΟΤΙΜΑ(ΠΟΤΗΣ, ΜΠΥΡΑ)  
ΣΥΧΝΑΖΕΙ(ΠΟΤΗΣ, ΜΑΓΑΖΙ)  
ΣΕΡΒΙΡΕΙ(ΜΑΓΑΖΙ, ΜΠΥΡΑ)

Παράδειγμα



1. Τους πότες που συχνάζουν σε μαγαζιά που σερβίρουν μπίρα «Guinness»
2. Τα μαγαζιά που σερβίρουν μπίρα «Guinness» ή μπίρα «Leffe Brune» ή και τα δύο
3. Τα μαγαζιά που σερβίρουν μπίρα «Guinness» και μπίρα «Leffe Brune»
4. Τα μαγαζιά που σερβίρουν μόνο μπίρα «Guinness»
5. Μαγαζιά που σερβίρουν **τουλάχιστον δύο** διαφορετικές μπίρες. (μόνο μία:)
6. Μαγαζιά που σερβίρουν ακριβώς δύο διαφορετικές μπίρες.
7. Τα μαγαζιά που σερβίρουν μπίρες που προτιμά ο πότης «Δημήτρης».
8. Τα μαγαζιά που σερβίρουν όλες τις μπίρες που προτιμά ο «Δημήτρης».

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

43

## Η Γλώσσα Βάσεων Δεδομένων SQL

Μέρος 1 - Γλώσσα ερωτήσεων: (Βασική Δομή, Πράξεις Συνόλου, Φωλιασμένες Υπο-ερωτήσεις, Συνάθροιση)

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

44

Πράξεις Συνόλων

Πράξεις:

- **union** (ένωση)
- **intersection** (τομή)
- **except** (διαφορά)

εφαρμόζονται σε συμβατές σχέσεις.

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

45

Πράξεις Συνόλων

Γενική Σύνταξη:

```
(select  
from  
where )  
union/intersection/except  
(select  
from  
where )
```

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

46

Πράξεις Συνόλων

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)  
Παίξει(Όνομα, Τίτλος, Έτος)  
Ηθοποίος(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

Παράδειγμα intersect:

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

```
(select Όνομα  
from Παίξει  
where Έτος = 2006)  
intersect  
(select Όνομα  
from Παίξει  
where Έτος = 2007)
```

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

47

Πράξεις Συνόλων

**Απαλοιφή διπλών εμφανίσεων, εκτός αν χρησιμοποιηθεί το intersect all**

Μέγιστος αριθμός πολλαπλών εμφανίσεων:

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

48



Πράξεις Συνόλων

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

```
(select Όνομα
from Παίζει
where Έτος = 2006)
union
(select Όνομα
from Παίζει
where Έτος = 2007)
```

Ποιο είναι το αποτέλεσμα;

Αντίστοιχα:  
**union all**  
 Μέγιστος αριθμός πολλαπλών εμφανίσεων;

Πράξεις Συνόλων

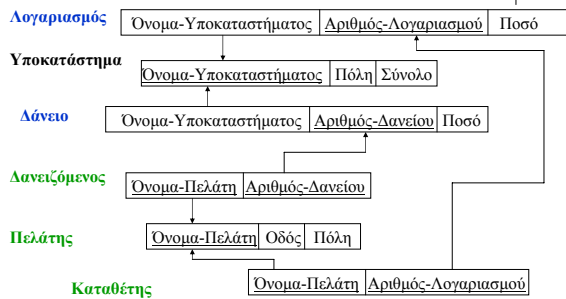
Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)  
 Παίζει(Όνομα, Τίτλος, Έτος)  
 Ηθοποιοί(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

```
(select Όνομα
from Παίζει
where Έτος = 2006)
except
(select Όνομα
from Παίζει
where Έτος = 2007)
```

Ποιο είναι το αποτέλεσμα;

Αντίστοιχα:  
**except all**  
 Μέγιστος αριθμός πολλαπλών εμφανίσεων;

Παράδειγμα Τράπεζα



Παράδειγμα Τράπεζα: Πράξεις Συνόλων

Παράδειγμα **intersect**:

Τα ονόματα όλων των πελατών που έχουν καταθέσεις και έχουν πάρει δάνειο

```
(select Όνομα-Πελάτη
from Καταθέτης )
intersect
(select Όνομα-Πελάτη
from Δανειζόμενος )
```

Παράδειγμα Τράπεζα: Πράξεις Συνόλων

Παράδειγμα **union**:

Τα ονόματα όλων των πελατών που έχουν καταθέσεις ή/και έχουν πάρει δάνειο

```
(select Όνομα-Πελάτη
from Καταθέτης)
union
(select Όνομα-Πελάτη
from Δανειζόμενος)
```

Παράδειγμα Τράπεζα: Πράξεις Συνόλων

Παράδειγμα **except**:

Τα ονόματα όλων των πελατών που έχουν καταθέσεις και δεν έχουν πάρει δάνειο

```
(select Όνομα-Πελάτη
from Καταθέτης)
except
(select Όνομα-Πελάτη
from Δανειζόμενος)
```

## Πράξεις Συνόλων

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)  
Παίξει(Όνομα, Τίτλος, Έτος)  
Ηθοποιάς(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

### Παραδείγματα

- Ηθοποιούς που δεν έπαιξαν σε έγχρωμη ταινία
- Τις ταινίες (τίτλο) με τον ίδιο τίτλο που γυρίστηκαν το 2005 και το 2006

## Πράξεις συνόλων (επανάληψη)

Πράξεις:

- union
- intersection
- except

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

Σύνταξη,

(select-from-where) union (select-from-where)

Απαλοιφή διπλών εμφανίσεων, εκτός αν χρησιμοποιηθεί το union (intersection, except) all

ΠΡΟΤΙΜΑ(ΠΟΤΗΣ, ΜΠΥΡΑ)  
ΣΥΧΝΑΖΕΙ(ΠΟΤΗΣ, ΜΑΓΑΖΙ)  
ΣΕΡΒΙΡΕΙ(ΜΑΓΑΖΙ, ΜΠΥΡΑ)

### Παράδειγμα

1. Τους πότες που συχνάζουν σε μαγαζιά που σερβίρουν μπίρα «Guinness»
2. Τα μαγαζιά που σερβίρουν μπίρα «Guinness» ή μπίρα «Leffe Brune» ή και τα δύο
3. Τα μαγαζιά που σερβίρουν μπίρα «Guinness» και μπίρα «Leffe Brune»
4. Τα μαγαζιά που σερβίρουν μόνο μπίρα «Guinness»
5. Τα μαγαζιά που δε σερβίρουν μπίρα «Guinness»
6. Μαγαζιά που σερβίρουν τουλάχιστον δύο διαφορετικές μπίρες. (μόνο μία.)
7. Μαγαζιά που σερβίρουν ακριβώς δύο διαφορετικές μπίρες.
8. Τα μαγαζιά που σερβίρουν μπίρες που προτιμά ο πότης «Δημήτρης».
9. Τα μαγαζιά που σερβίρουν όλες τις μπίρες που προτιμά ο «Δημήτρης».

## Η Γλώσσα Βάσεων Δεδομένων SQL

Μέρος 1 - Γλώσσα ερωτήσεων: (Βασική Δομή, Πράξεις Συνόλου, Φωλιασμένες Υπο-ερωτήσεις, Συνάθροιση)

## Φωλιασμένες Υπο-ερωτήσεις

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 (συγκεκριμένα ως συνθήκη στο **where**).

## Φωλιασμένες Υπο-ερωτήσεις

Γενική δομή:

select ...  
from ...  
where

<ΤΕΛΕΣΤΗΣ>

(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

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

Παράδειγμα: Τα ονόματα όλων των ηθοποιών που δεν έπαιξαν σε καμία ταινία

```
select distinct Ηθοποιός.Όνομα
from Ηθοποιός
where Ηθοποιός.Όνομα not in
```

```
(select Όνομα
from Παιζει)
```

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

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

```
select distinct Παιζει.Όνομα
from Παιζει
where (Παιζει.Τίτλος, Παιζει.Έτος) in
```

```
(select Ταινία.Τίτλος, Ταινία.Έτος
from Ταινία
where Είδος = «Ασπρόμαυρη»)
```

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)  
 Παιζει(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 όνομα και διάρκεια μικρότερη από 60 λεπτά

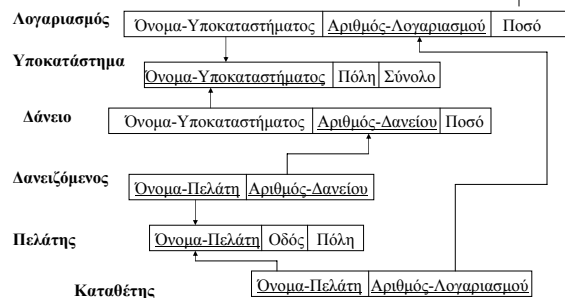
```
select distinct Τίτλος
from Ταινία
where Διάρκεια > 100
and Τίτλος in
```

```
(select Τίτλος
from Ταινία
where Διάρκεια < 60)
```

Μπορεί να χρησιμοποιηθεί και με enumerated σύνολα

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

```
select distinct Τίτλος
from Ταινία
where Έτος not in (2006, 2007)
```



## Παράδειγμα Τράπεζα-Φωλιασμένες Υπο-ερωτήσεις

Παράδειγμα: Τα ονόματα όλων των πελατών που έχουν πάρει δάνειο και έχουν καταθέσεις

```
select distinct Όνομα-Πελάτη
from Δανειζόμενος
where Όνομα-Πελάτη in (select Όνομα-Πελάτη
from Καταθέτης)
```

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

67

## Παράδειγμα Τράπεζα-Φωλιασμένες Υπο-ερωτήσεις

Παράδειγμα: Τα ονόματα όλων των πελατών που έχουν πάρει δάνειο και έχουν καταθέσεις στο υποκατάστημα Ψηλά-Αλιώνια

```
select distinct Όνομα-Πελάτη
from Δανειζόμενος, Δάνειο
where Δανειζόμενος.Αριθμός-Δανείου = Δάνειο.Αριθμός.Δανείου
and Όνομα-Υποκαταστήματος = "Ψηλά-Αλιώνια"
and (Όνομα-Υποκαταστήματος, Όνομα-Πελάτη) in
```

```
(select Όνομα-Υποκαταστήματος, Όνομα-Πελάτη
from Καταθέτης, Λογαριασμός
where Καταθέτης.Αριθμός-Λογαριασμού =
Λογαριασμός.Αριθμός-Λογαριασμού)
```

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

68

## Παράδειγμα Τράπεζα-Φωλιασμένες Υπο-ερωτήσεις

Με *enumerated* σύνολα

Παράδειγμα: Τα ονόματα όλων των πελατών που έχουν πάρει δάνειο και δε λέγονται "Παπαδόπουλος" ή "Πέτρου".

```
select distinct Όνομα-Πελάτη
from Δανειζόμενος
where Όνομα-Πελάτη not in ("Παπαδόπουλος", "Πέτρου")
```

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

69

## Φωλιασμένες Υπο-ερωτήσεις

### Σύγκριση Συνόλων

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

Γενική δομή:

```
select ...
from ...
where
  T >some (select ...
           from ...
           where ... );
```

T: κλειάδα

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

70

## Φωλιασμένες Υπο-ερωτήσεις

### Σύγκριση Συνόλων

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

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")
```

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

71

## Φωλιασμένες Υπο-ερωτήσεις

• επίσης:

```
<some,
<=some,
>=some,
=some (ισοδ. του in)
<>some (όχι ισοδ. του not in)
```

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

72

Φωλιασμένες Υπο-ερωτήσεις

2. 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")
```

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

73

Φωλιασμένες Υπο-ερωτήσεις

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

Παράδειγμα: Τι υπολογίζει το παρακάτω;

```
select distinct Όνομα
from Ηθοποιοί
where Έτος-Γέννησης <= all (select Έτος-Γέννησης
                             from Παίζει, Ηθοποιοί
                             where Παίζει.Όνομα = Ηθοποιοί.Όνομα
                             and Τίτλος = «Μανταλένα»)
```

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

74

Φωλιασμένες Υπο-ερωτήσεις

• επίσης:

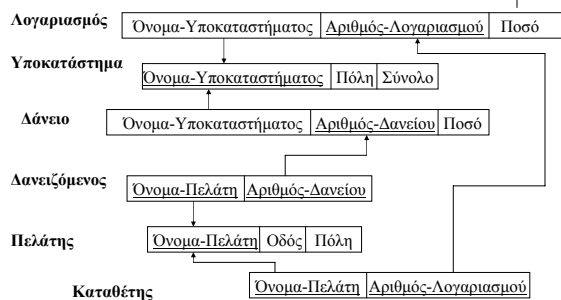
- <all,
- <=all,
- >=all,
- =all,
- <>all (ισοδ. του not in)

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

75

Παράδειγμα Τράπεζα



Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

76

Παράδειγμα Τράπεζα-Φωλιασμένες Υπο-ερωτήσεις

Παράδειγμα: Τα ονόματα όλων των υποκαταστημάτων που έχουν λογαριασμό με ποσό καταθέσεων μεγαλύτερο από το ποσό καταθέσεων όλων των υποκαταστημάτων των Ιωαννίνων.

```
select distinct Όνομα-Υποκαταστήματος
from Υποκατάστημα
where Ποσό > all (select Ποσό
                 from Υποκατάστημα
                 where Πόλη = "Ιωάννινα")
```

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

77

Παράδειγμα Τράπεζα-Φωλιασμένες Υπο-ερωτήσεις

Παράδειγμα: Τα ονόματα όλων των υποκαταστημάτων που έχουν λογαριασμό με ποσό καταθέσεων μεγαλύτερο από το ποσό καταθέσεων ενός τουλάχιστον υποκαταστήματος των Ιωαννίνων.

```
select distinct Όνομα-Υποκαταστήματος
from Υποκατάστημα
where Ποσό > some (select Ποσό
                  from Υποκατάστημα
                  where Πόλη = "Ιωάννινα")
```

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

78

### Παράδειγμα Τράπεζα-Φωλιασμένες Υπο-ερωτήσεις

Παράδειγμα: Το υποκατάστημα με το μεγαλύτερο μέσο ποσό καταθέσεων. (θα τη δούμε στη συνέχεια)

```
select distinct Όνομα-Υποκαταστήματος
from Λογαριασμός
group by Όνομα-Υποκαταστήματος
having avg (Ποσό) >=all (select avg(Ποσό)
                        from Λογαριασμός
                        group by Όνομα-Υποκαταστήματος)
```

Το avg θα το δούμε στη συνέχεια

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

79

### Φωλιασμένες Υπο-ερωτήσεις

#### 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 ... );
```

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

80

### Φωλιασμένες Υπο-ερωτήσεις

Παράδειγμα: Οι ασπρόμαυρες ταινίες με τουλάχιστον ένα ηθοποιό

```
select T.Τίτλος, T.Έτος
from Tαινία as T
where T.είδος = «Ασπρόμαυρη» and
exists (select *
        from Παίζει
        where Παίζει.Τίτλος = T.Τίτλος and Παίζει.Έτος = T.Έτος)
```

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

81

### Φωλιασμένες Υπο-ερωτήσεις

Ο τελεστής not exists μπορεί να χρησιμοποιηθεί για έλεγχο αν η σχέση A περιέχει τη σχέση B

**not exists (B except A)**  
True if and only if  $A \supseteq B$

• Ποια πράξη της σχεσιακής άλγεβρας;

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

82

### Φωλιασμένες Υπο-ερωτήσεις

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)  
Παίζει(Όνομα, Τίτλος, Έτος)  
Ηθοποιοί(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

A: όλες οι ταινίες του συγκεκριμένου ηθοποιού

```
select distinct S.Όνομα
from Παίζει as S
where not exists
  ((select Τίτλος, Έτος B
    from Παίζει
    where Όνομα = "George Clooney")
  except
  (select Τίτλος, Έτος A
    from Παίζει as R
    where R.Όνομα = S.Όνομα))
```

υπολογισμός για κάθε S

Τέτοιου είδους μεταβλητές δεν υπάρχουν στη σχεσιακή άλγεβρα

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

83

### Παράδειγμα Τράπεζα-Φωλιασμένες Υπο-ερωτήσεις

Παράδειγμα: Οι πελάτες που έχουν καταθέσεις και έχουν πάρει δάνειο.

```
select Όνομα-Πελάτη
from Δανειζόμενος
where exists (select *
            from Καταθέτης
            where Καταθέτης.Όνομα-Πελάτη = Δανειζόμενος.Όνομα-Πελάτη)
```

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

84

### Παράδειγμα Τράπεζα-Φωλιασμένες Υπο-ερωτήσεις

Παράδειγμα: Οι πελάτες που έχουν καταθέσεις σε όλα τα υποκαταστήματα της Πάτρας.

B: όλα τα υποκαταστήματα της Πάτρας

A: όλα τα υποκαταστήματα στα οποία έχει κατάθεση ο συγκεκριμένος πελάτης

```
select distinct S.Όνομα-Πελάτη
from Καταθέτης as S
where not exists ((select Όνομα-Υποκαταστήματος
from Υποκατάστημα
where Πόλη = "Πάτρα")
except
(select R.Όνομα-Υποκαταστήματος
from Καταθέτης as T, Λογαριασμός as R
where T.Όνομα-Πελάτη = S.Όνομα-Πελάτη and
T.Αριθμός-Λογαριασμού = R.Αριθμός-Λογαριασμού))
```

### Φωλιασμένες Υπο-ερωτήσεις

#### 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

### Φωλιασμένες Υπο-ερωτήσεις

```
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
```

### Φωλιασμένες Υπο-ερωτήσεις

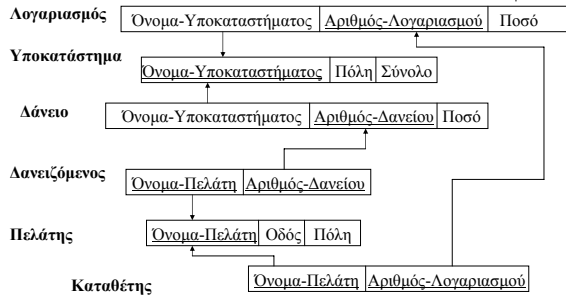
```
Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)
Παίζει(Όνομα, Τίτλος, Έτος)
Ηθοποιός(Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)
```

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
```

### Παράδειγμα Τράπεζα



### Παράδειγμα Τράπεζα-Φωλιασμένες Υπο-ερωτήσεις

Παράδειγμα: Οι πελάτες που έχουν ακριβώς μια κατάθεση στο υποκατάστημα "Ψηλά Αλώνια"

```
select T.Όνομα-Πελάτη
from Καταθέτης as T
where unique (select R.Όνομα-Πελάτη
from Λογαριασμός, Καταθέτης as R
where T.Όνομα-Πελάτη = R.Όνομα-Πελάτη
and R.Αριθμός-Λογαριασμού = Λογαριασμός.Αριθμός-Λογαριασμού
and Λογαριασμός.Όνομα-Υποκαταστήματος = "Ψηλά Αλώνια")
```

## Παράδειγμα Τράπεζα-Φωλιασμένες Υπο-ερωτήσεις

Παράδειγμα: Οι πελάτες που έχουν τουλάχιστον δύο καταθέσεις στο υποκατάστημα "Ψηλά Αλώνια"

```
select T.Όνομα-Πτελάτη
from Καταθέτης as T
where not unique (select R.Όνομα-Πτελάτη
from Λογαριασμός, Καταθέτης as R
where T.Όνομα-Πτελάτη = R.Όνομα-Πτελάτη
and R.Αριθμός-Λογαριασμού =
Λογαριασμός.Αριθμός-Λογαριασμού
and Λογαριασμός.Όνομα-Υποκαταστήματος = "Ψηλά
Αλώνια")
```

## Φωλιασμένες Υπο-ερωτήσεις (επανάληψη)

Ο τελεστής μπορεί να είναι:

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

## Φωλιασμένες Υπο-ερωτήσεις (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

## Φωλιασμένες Υπο-ερωτήσεις (επανάληψη)

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)  
Παιζεί (Όνομα, Τίτλος, Έτος)  
Ηθοποιοί (Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

### Παράδειγματα

```
select Τίτλος
from Ταινία
where Διάρκεια > some (select Διάρκεια
from Ταινία
where Είδος = «Έγχρωμη»)

select Τίτλος
from Ταινία
where Διάρκεια > all (select Διάρκεια
from Ταινία
where Είδος = «Έγχρωμη»)

select Τίτλος
from Ταινία
where Διάρκεια in (select Διάρκεια
from Ταινία
where Είδος = «Έγχρωμη»)

Μία συνθήκη του where
```

## Φωλιασμένες Υπο-ερωτήσεις (επανάληψη)

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)  
Παιζεί (Όνομα, Τίτλος, Έτος)  
Ηθοποιοί (Όνομα, Διεύθυνση, Έτος-Γέννησης, Σύζυγος-Ηθοποιού)

### Παράδειγματα

```
select Τίτλος
from Ταινία as T
where exists (select *
from Ταινία as S
where T.Τίτλος = S.Τίτλος and
S.Διάρκεια > T.Διάρκεια)

select Τίτλος
from Ταινία as T
where unique (select *
from Ταινία as S
where T.Τίτλος = S.Τίτλος and
S.Διάρκεια > T.Διάρκεια)
```

# Η Γλώσσα Βάσεων Δεδομένων SQL

Μέρος 1 - Γλώσσα ερωτήσεων: (Βασική Δομή, Πράξεις Συνόλου, Φωλιασμένες Υπο-ερωτήσεις, Συνάθροιση)





## 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: 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**



*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 προβάλουμε και άλλα γνωρίσματα σχέσεων



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

```
select sum(distinct Διάρκεια)
from Ταινία
```



Για να μετρήσουμε πόσες πλειάδες έχει μια σχέση:

```
select count(*)
from Ταινία
```

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



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

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

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

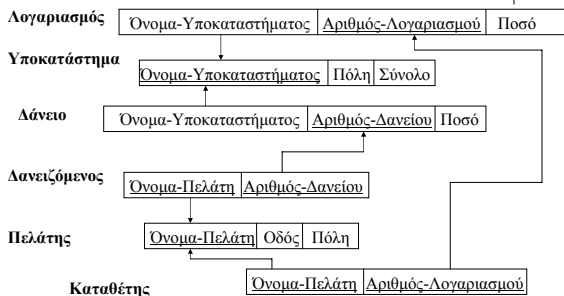
### Συναθροιστικές Συναρτήσεις

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(*)
from Παίζει, Ηθοποιός
1 where Παίζει.Όνομα = Ηθοποιός.Όνομα and Έτος-Γέννησης > 1987
2 group by Ηθοποιός.Όνομα
3 having count (*) >= 5
```

### Παράδειγμα Τράπεζα



### Παράδειγμα Τράπεζα: Συναθροιστικές Συναρτήσεις

*Παράδειγμα: Μέσο ποσό όλων των λογαριασμών στο υποκατάστημα Καλούτσανη*

```
select avg(Ποσό)
from Λογαριασμός
where Όνομα-Υποκαταστήματος = "Καλούτσανη"
```

Παράδειγμα Τράπεζα: Συναθροιστικές Συναρτήσεις

Παράδειγμα: Μέγιστο ποσό όλων των λογαριασμών στο υποκατάστημα Καλούτσανη

```
select max(Ποσό)
from Λογαριασμός
where Όνομα-Υποκαταστήματος = "Καλούτσανη"
```

*Αν θέλουμε και τον αριθμό του λογαριασμού;*

Παράδειγμα Τράπεζα: Συναθροιστικές Συναρτήσεις

Παράδειγμα: Μέσο ποσό των λογαριασμών σε κάθε υποκατάστημα

```
select Όνομα-Υποκαταστήματος, avg(Ποσό)
from Λογαριασμός
group by Όνομα-Υποκαταστήματος
```

Παράδειγμα Τράπεζα: Συναθροιστικές Συναρτήσεις

Παράδειγμα: Αριθμός καταθετών σε κάθε υποκατάστημα

```
select Όνομα-Υποκαταστήματος, count(distinct Όνομα-Πελάτη)
from Καταθέτης, Λογαριασμός
where ...
group by Όνομα-Υποκαταστήματος
```

Παράδειγμα Τράπεζα: Συναθροιστικές Συναρτήσεις

Μέσος όρος καταθέσεων ανά πελάτη και ανά υποκατάστημα

```
select Όνομα-Υποκαταστήματος, Όνομα-Πελάτη, avg(balance)
from account
group by Όνομα-Υποκαταστήματος, Όνομα-Πελάτη
```

Παράδειγμα Τράπεζα: Συναθροιστικές Συναρτήσεις

Παράδειγμα: Ονόματα υποκαταστημάτων με μέσο ποσό καταθέσεων μεγαλύτερο των \$1200

```
select Όνομα-Υποκαταστήματος, avg(Ποσό)
from Λογαριασμός
group by Όνομα-Υποκαταστήματος
having avg(Ποσό) > 1200
```

Συναθροιστικές Συναρτήσεις

Παράδειγμα: Μέσο ποσό για κάθε πελάτη που ζει στα Ιωάννινα και έχει τουλάχιστον τρεις λογαριασμούς

```
select Καταθέτης.Όνομα-Πελάτη, avg(Ποσό)
from Καταθέτης, Λογαριασμός, Πελάτης
where Καταθέτης.Αριθμός-Λογαριασμού = Λογαριασμός.Αριθμός-Λογαριασμού
and Καταθέτης.Όνομα-Πελάτη = Πελάτης.Όνομα-Πελάτη and Πόλη = "Ιωάννινα"
group by Καταθέτης.Όνομα-Πελάτη
having count (distinct Καταθέτης.Αριθμός-Λογαριασμού) >= 3
```

## Συναθροιστικές Συναρτήσεις

### Περίληψη

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

## Βασική Δομή (επανάληψη)

```
select Ai1, Ai2, ..., Ain, ..., avg, ...  
from R1, R2, ..., Rm  
where P  
group by Aj1, Aj2, ..., Ajn  
having P  
order by Ak1, Ak2, ..., Ajk
```

## Η Γλώσσα SQL

(Μέρος 2: Γλώσσα Ορισμού, Γλώσσα Τροποποίησης)

## Η γλώσσα SQL

Η SQL είναι η γλώσσα για όλα τα εμπορικά σχεσιακά συστήματα διαχείρισης βάσεων δεδομένων

Η SQL έχει διάφορα τμήματα:

- Γλώσσα Ορισμού Δεδομένων (ΓΟΔ)
- Γλώσσα Χειρισμού Δεδομένων (ΓΧΔ)
- Ενσωματωμένη Γλώσσα Χειρισμού Δεδομένων
- Ορισμό Όψεων
- Εξουσιοδότηση (authentication)
- Ακεραιότητα
- Έλεγχο Συναλλαγών

## Εισαγωγή

### Βήματα Δημιουργίας και Χρήσης μιας (Σχεσιακής) Βάσης Δεδομένων

Σχεδιασμός Σχήματος

Δημιουργία Σχήματος χρησιμοποιώντας τη ΓΟΔ (DDL)

Μαζική Φόρτωση των αρχικών δεδομένων

⇒ Η βάση δεδομένων έχει δεδομένα

Repeat: εκτέλεση ερωτήσεων (select-from-where) και τροποποιήσεων (insert-delete-update) στη βάση δεδομένων

## Η γλώσσα SQL

### Μερικές Γενικές Παρατηρήσεις

Oracle SQL και η MySQL μερικές φορές δεν ακολουθούν ακριβώς τα standards - μερικές εντολές στις διαφάνειες μπορεί να μη «τρέχουν»

*Κάποιες αποκλίσεις περιγράφονται στη web σελίδα του μαθήματος*

Επίσης, "interactive" SQL - εντολές που πληκτρολογούνται μετά από το prompt και οι απαντήσεις εμφανίζονται στην οθόνη ως πίνακες

"Embedded" και "dynamic" SQL: θα τη δούμε στην (επόμενη) προγραμματιστική άσκηση



## Γλώσσα Ορισμού Δεδομένων (ΓΟΔ)

Σχετικά με το λογικό σχήμα, η ΓΟΔ SQL υποστηρίζει τους ορισμούς:

- του *σχήματος* κάθε σχέσης
- του *πεδίου τιμών* κάθε γνωρισματος
- των *περιορισμών ακεραιότητας*

Δείτε και τις σχετικές διαφάνειες προηγούμενου μαθήματος



## Γενική Δομή Ορισμού

```
create table R(A1 D1, A2 D2, ..., An Dn),
<περιορισμός-ακεραιότητας1>,
...,
<περιορισμός-ακεραιότηταςk>
```

όπου **R** είναι το όνομα της σχέσης, **A<sub>i</sub>** τα ονόματα των γνωρισμάτων, και **D<sub>i</sub>** οι τύποι των αντίστοιχων πεδίων τιμών.



Επιτρεπτοί περιορισμοί ακεραιότητας είναι της μορφής:

- **primary key**  $A_{j_1}, A_{j_2}, \dots, A_{j_n}$ , (δεν επιτρέπονται επαναλαμβανόμενες τιμές και NULL τιμές)  
*για τον ορισμό του πρωτεύοντος κλειδιού*
- **unique**  $A_{j_1}, A_{j_2}, \dots, A_{j_n}$ , (δεν επιτρέπονται επαναλαμβανόμενες τιμές; NULL τιμές επιτρέπονται (μόνο μία))  
*για τον ορισμό υποψηφίων κλειδίων*
- **check P**  
*για τον ορισμό σημασιολογικών περιορισμών*
- **foreign key (A<sub>i</sub>) references A<sub>j</sub>**  
*για τον ορισμό ξένου κλειδιού*



Παραδείγματα

(1)  
**create table** Πτελάτης  
 (Όνομα-Πτελάτη **char(20) not null**,  
 Οδός **char(30)**,  
 Πόλη **char(30)**,  
**primary key** (Όνομα-Πτελάτη))



(2)  
**create table** Λογαριασμός  
 (Αριθμός-Λογαριασμού **char(10) not null**,  
 Όνομα-Υποκαταστήματος **char(15)**,  
 Ποσό **int default 0**,  
**primary key** (Αριθμός-Λογαριασμού)  
**check** (Ποσό >= 0))



Επίσης, πιο περίπλοκες συνθήκες:

**check** (Όνομα-Υποκαταστήματος **in select** Όνομα-Υποκαταστήματος  
**from** Υποκατάστημα)



Περισσότερα για τους Περιορισμούς Αναφοράς

Σύνταξη:

**foreign key (A<sub>i</sub>) references A<sub>j</sub>**

Όταν μια πράξη παραβιάζει έναν περιορισμό αναφοράς απορρίπτεται εκτός αν έχει οριστεί:

**cascade, set null, set default  
on delete  
on update**



Παράδειγμα σύνταξης

**create table**  
..  
**foreign key (Όνομα-Υποκατάστημα) references Υποκατάστημα  
on delete cascade  
on update cascade**  
...



Ο ορισμός πεδίου μπορεί να περιέχει τον προσδιορισμό **not null** και **default** τιμή

Επίσης, επιτρέπεται δημιουργία πεδίου:

**create domain <name> as <type-description>**

**create domain** Όνομα-Προσώπου **char(20)**



Διαγραφή Σχήματος

Μια καινούργια σχέση είναι αρχικά άδεια.

Για να σβηστεί ένα σχήμα:

**drop table R**

Διαφορά από

**delete from R**



Τροποποίηση Σχήματος

**ALTER TABLE** όνομα πίνακα

- **ADD** - προσθέτει καινούργια στήλη
- **DROP** - διαγράφει μια στήλη
- **MODIFY** - τροποποιεί μια στήλη



Προσθήκη νέου γνωρίσματος:

**alter table R add A D**

προσθήκη σε μια σχέση R που ήδη υπάρχει του γνωρίσματος A με πεδίο τιμών D, η τιμή των πλειάδων της R στο καινούργιο γνώρισμα είναι null.

Διαγραφή γνωρίσματος:

**alter table R drop A**



`alter table R modify` (όνομα\_στήλης new\_datatype)

`modify` μπορεί να τροποποιήσει μόνο τον τύπο δεδομένων, όχι το όνομα της στήλης

## Η Γλώσσα SQL

(Μέρος 2: Γλώσσα Ορισμού, Γλώσσα Τροποποίησης)



Τροποποίηση Βάσης Δεδομένων: Γλώσσα Χειρισμού Δεδομένων (ΓΧΔ)

### Τροποποιήσεις

1. Διαγραφή
2. Εισαγωγή
3. Ενημέρωση

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

Δείτε και τις σχετικές διαφάνειες προηγούμενου μαθήματος

## Εισαγωγή



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`



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

Παράδειγμα για το (α)

`insert into Ταινία`  
`values ('The Big Blue', 1988, 132, 'Έγχρωμη')`

Όταν με οποιαδήποτε σειρά, π.χ.:

`insert into Ταινία (Τίτλος, Είδος, Διάρκεια, Έτος)`  
`values ('The Big Blue', 'Έγχρωμη', 132, 1988)`



Παράδειγμα για το (α)

`insert into Λογαριασμός`  
`values ("Ψηλά-Αλώνια", "A--9732", 1200)`

Όταν με οποιαδήποτε σειρά, π.χ.:

`insert into Λογαριασμός (Αριθμός-Λογαριασμού, Όνομα-Υποκαταστήματος, Ποσό)`  
`values ("A--9732", "Ψηλά-Αλώνια", 1200)`

## Παράδειγμα Τράπεζα: Εισαγωγή



Παράδειγμα για το (β):

Για κάθε πελάτη που έχει πάρει δάνειο από το υποκατάστημα Ψηλά Αλώνια προστίθεται ως δώρο ένας λογαριασμός των \$200

```
insert into Λογαριασμός
select Όνομα-Υποκαταστήματος, Αριθμός-Δανείου, 200
from Δάνειο
where Όνομα-Υποκαταστήματος = "Ψηλά Αλώνια"
```

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

139

## Εισαγωγή



Επίσης, εισαγωγή *null* τιμών:

```
insert into Λογαριασμός
values (null, "A--9732", 1200)
```

```
insert into Ταινία
values ('The Big Blue', 1988, null, 'Έγχρωμη')
```

ή αν *δε δίνω τιμές για όλα τα γνωρίσματα*

```
insert into Ταινία (Τίτλος, Έτος, Είδος)
values ('The Big Blue', 1988, 'Έγχρωμη')
```

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

140

## Εισαγωγή



Πρέπει πρώτα να υπολογιστεί το **select** πλήρως και μετά να γίνει η εισαγωγή.

Τι αποτέλεσμα έχει η παρακάτω εντολή αν αυτό δε συμβαίνει;

```
insert into Λογαριασμός
select *
from Λογαριασμός
```

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

141

## Διαγραφή



### Διαγραφή

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.

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

142

## Διαγραφή



Παράδειγματα

(1) Όλες οι ηθοποιοί με το όνομα Kildman

```
delete from Ηθοποιοί
where Όνομα = 'Kildman'
```

(2) Όλες τις έγχρωμες ταινίες

```
delete from Ταινία
where Τίτλος, Έτος in (select Τίτλος, Έτος
from Ταινία
where Είδος = «Έγχρωμη»)
```

(3) Όλες τις ταινίες που έχουν γυριστεί πριν το 1950

```
delete from Ταινία
where Έτος < 1950
```

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

143

## Διαγραφή



▪ 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 Παιζει
where Τίτλος, Έτος in (select Τίτλος, Έτος
from Ταινία
where Είδος = «Έγχρωμη»)
```

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

144



### Διαγραφή



Παράδειγμα: διαγραφή της ταινίας "The Big Blue" που γυρίστηκε το 1988

**delete from** Ταϊνιά

**where** Τίτλος = 'The Big Blue' and Έτος = 1988

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

Πρέπει πρώτα να διαγράψουμε και τις εγγραφές του πίνακα Παιζει που σχετίζονται με την ταινία "The Big Blue":

**delete from** Παιζει

**where** Τίτλος = 'The Big Blue' and Έτος = 1988

*ΠΡΟΣΟΧΗ: όταν θέλουμε να διαγράψουμε κάποια δεδομένα, πρέπει να διαγράψουμε όλα τα δεδομένα που συσχετίζονται με αυτά (γενικά). Επίσης πρέπει να προσέξουμε την σειρά με την οποία θα γίνουν οι διαγραφές.*

### Παράδειγμα Τράπεζα: Διαγραφή



Παραδείγματα

(1) Όλους τους λογαριασμούς του Παπαδόπουλου

**delete from** Καταθέτης

**where** Όνομα-Πελάτη = "Παπαδόπουλος"

### Παράδειγμα Τράπεζα: Διαγραφή



(2) Όλους τους λογαριασμούς στα υποκαταστήματα της Πάτρας

**delete from** Λογαριασμός

**where** Όνομα-Υποκαταστήματος in (**select** Όνομα-Υποκαταστήματος  
**from** Υποκατάστημα  
**where** Πόλη = "Πάτρα")

*Παρατήρηση: δεν υπάρχει τρόπος να διαγράψουμε τη μία από δυο ίδιες πλειάδες που ικανοποιούν το where*

### Παράδειγμα Τράπεζα: Διαγραφή



Αν και μπορούμε να σβήσουμε πλειάδες μόνο από μία σχέση τη φορά μπορούμε να αναφερθούμε σε περισσότερες από μια σχέσεις στην υποερώτηση του **where**

(3) Όλους τους λογαριασμούς μιας τράπεζας με ποσό μικρότερο από το μέσο ποσό στην τράπεζα.

**delete from** Λογαριασμός  
**where** Ποσό > (**select avg**(Ποσό)  
**from** Λογαριασμός)

Πρώτα γίνεται ο έλεγχος σε όλες τις πλειάδες και μετά αυτές που ικανοποιούν τη συνθήκη διαγράφονται.

### Παράδειγμα Τράπεζα: Διαγραφή



Παράδειγμα: μια τράπεζα θέλει να κλείσει όλα τα υποκαταστήματά της που βρίσκονται στην Καστοριά

**delete from** Υποκατάστημα

**where** Όνομα-Υποκαταστήματος in (**select** Όνομα-Υποκαταστήματος  
**from** Υποκατάστημα  
**where** Πόλη = "Καστοριά")

### Παράδειγμα Τράπεζα: Διαγραφή



Πρέπει να διαγράψουμε και όλους τους λογαριασμούς:

**delete from** Λογαριασμός  
**where** Όνομα-Υποκαταστήματος in (**select** Όνομα-Υποκαταστήματος  
**from** Υποκατάστημα  
**where** Πόλη = "Καστοριά")

Υπενθύμιση: όταν θέλουμε να διαγράψουμε κάποια δεδομένα, πρέπει να διαγράψουμε όλα τα δεδομένα που συσχετίζονται με αυτά. Επίσης πρέπει να προσέξουμε την σειρά με την οποία θα γίνουν οι διαγραφές.

### Παράδειγμα Τράπεζα: Διαγραφή

υποκατάστημα		Λογαριασμός		
Πόλη	Όνομα_Υποκ.	Όνομα_Υποκ.	Όνομα-Πελάτη	Υπόλοιπο
Καστοριά	K1	K1	ΚΩΤΣΗΣ	350.000
Καστοριά	K3	K2	ΑΠΟΣΤΟΛΙΔΗΣ	230.000
Θεσσαλονίκη	Θ1	Θ1	ΣΤΕΦΑΝΟΥ	670.000
Θεσσαλονίκη	Θ2	Θ2	ΠΑΠΑΝΙΚΟΛΑΟΥ	256.000
Αθήνα	A1	K3	ΧΑΤΖΟΠΟΥΛΟΣ	410.000
...	...	...	...	...

- αν διαγράψουμε από τον πίνακα υποκατάστημα όλα τα υποκαταστήματα της Καστοριάς, θα έχουμε πρόβλημα ορθότητας στον πίνακα λογαριασμός.
- πρώτα πρέπει να διαγράψουμε τους λογαριασμούς και μετά τα υποκαταστήματα.

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

151

### Ενημερώσεις

### Ενημερώσεις

```
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
```

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

152

### 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 διαγράφονται οι πλειάδες που ικανοποιούν τη συνθήκη - δηλαδή, η συνθήκη υπολογίζεται στο τρέχων στιγμιότυπο - όχι στο τροποποιημένο

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

153

### Παράδειγμα Τράπεζα: Ενημέρωση

Παράδειγμα: Αύξηση όλων των καταθέσεων που είναι μεγαλύτερες των 100€ κατά 5% λόγω τοκισμού

```
update Λογαριασμός
set Ποσό = Ποσό * 1.05
where Ποσό > 100
```

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

154

### Παράδειγμα Τράπεζα: Ενημέρωση

Παράδειγμα:

στους πελάτες που έχουν υπόλοιπο < 1.000.000 η τράπεζα δίνει 5% και στους πελάτες που έχουν υπόλοιπο > 1.000.000 δίνει 9%:

```
update Λογαριασμός
set Ποσό = Ποσό * 1.05
where Ποσό < 1.000.000
```

```
update Λογαριασμός
set Ποσό = Ποσό * 1.09
where Ποσό > 1.000.000
```

Ποιο update πρέπει να τρέξουμε πρώτα;

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

155

### Παράδειγμα Τράπεζα: Ενημέρωση

Παράδειγμα: Αύξηση όλων των υπολοίπων που είναι μεγαλύτερα από τον μέσο όρο κατά 5%

```
update Λογαριασμός
set Υπόλοιπο = Υπόλοιπο * 1.05
where Υπόλοιπο > select avg(Υπόλοιπο)
from Λογαριασμός
```

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

156

## Γλώσσα Χειρισμού Δεδομένων (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
```

## Η Γλώσσα SQL

(Μέρος 3: Όψεις, Συνενώσεις)

## Ορισμός Όψεων

### Ορισμός Όψεων (εικονικών πινάκων)

Μπορούμε να ορίσουμε μια όψη χρησιμοποιώντας την εντολή:

```
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>
```

## Ορισμός Όψεων

*Παράδειγμα: Μια όψη που περιλαμβάνει τα ονόματα όλων των υποκαταστημάτων και το άθροισμα του ποσού των δανείων που έχουν γίνει από αυτά*

```
create view Υποκατάστημα-Σύνολο-Δανείων (Όνομα-  
Υποκαταστήματος, Σύνολο-Δανείων) as  
select Όνομα-Υποκαταστήματος, sum(Ποσό)  
from Δάνειο  
group by Όνομα-Υποκαταστήματος
```

## Όψεις

Διαφορά από τον πίνακα που ορίζεται με create table:

- Η όψη υπολογίζεται εκ νέου
- Αποθηκεύουμε τον ορισμό
- Τροποποιήσεις μέσω όψεων
- Υλοποιημένη (materialized) όψη

## Ορισμός Όψεων

- Τα ονόματα όψεων μπορεί να χρησιμοποιηθούν οπουδήποτε μπορεί να χρησιμοποιηθεί το όνομα μιας σχέσης
- Υπολογίζεται **εκ νέου** κάθε φορά
- Ο ορισμός της όψης παραμένει στην βάση δεδομένων, εκτός αν σβηστεί:

```
drop view <όνομα--όψης>
```

## Η Γλώσσα SQL

(Μέρος 3: Όψεις, Συνενώσεις)

### Συνενώσεις Συνόλων

#### Συνενώσεις Συνόλων

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> <συνθήκη-συνένωσης>
```

ή

```
<όνομα-σχέσης1> natural <τύπος-συνένωσης> <όνομα-σχέσης2>
```

### Συνενώσεις Συνόλων

Τύποι Συνένωσης:

**inner join** - το default

**left outer join**: αριστερή εξωτερική συνένωση

**right outer join**

**full outer join**

### Συνενώσεις Συνόλων

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

*Παράδειγμα: Τα ονόματα των πελατών που είτε έχουν καταθέσεις είτε έχουν πάρει δάνεια (αλλά όχι και τα δυο)*

```
select Όνομα-Πελάτη  
from Καταθέτης natural full outer join Δανειζόμενος  
where Αριθμός-Λογαριασμού is null or Αριθμός-Δανείου is null
```

### Συνενώσεις Συνόλων

Η συνθήκη της συνένωσης στο **from** με χρήση του **ON**

```
select distinct Όνομα  
from Παιζει, Ταινία  
where Παιζει.Τίτλος = Ταινία.Τίτλος and Παιζει.Έτος =  
Ταινία.Έτος and Είδος = "Ασπρόμαυρη"
```

```
select distinct Όνομα  
from (Παιζει Join Ταινία on Παιζει.Τίτλος = Ταινία.Τίτλος and  
Παιζει.Έτος = Ταινία.Έτος)  
where Είδος = "Ασπρόμαυρη"
```