

# 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

Σχεσιακή άλγεβρα (στην «καθαρή» μορφή της, μόνο γλώσσα ερωτήσεων)

Σχεσιακός λογισμός (πλειάδων και πεδίου)

### Εμπορικές Γλώσσες Προγραμματισμού

SQL

QBE

## 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.

## Εισαγωγή

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

*Πριν μια σύντομη επανάληψη της σχεσιακής άλγεβρας*

**Σχεσιακή άλγεβρα:** έναν απλό τρόπο δημιουργίας νέων σχέσεων από υπάρχουσες.

Ένα σύνολο από **πράξεις** που όταν εφαρμοστούν σε **σχέσεις** μας δίνουν **νέες σχέσεις**

Οι πράξεις της σχεσιακής άλγεβρας:

1. Πράξεις που αφαιρούν κομμάτια από μια σχέση είτε **επιλέγοντας γραμμές** είτε **προβάλλοντας στήλες**
2. Οι συνηθισμένες **πράξεις συνόλου** - ένωση, τομή, διαφορά
3. Πράξεις που **συνδυάζουν πλειάδες από δύο σχέσεις**
4. **Μετονομασία γνωρισμάτων**

Πλήρες σύνολο πράξεων

επιλογή ( $\sigma$ )  
 προβολή ( $\pi$ )  
 ένωση ( $\cup$ )  
 διαφορά ( $-$ )  
 καρτεσιανό γινόμενο ( $\times$ )

Επίσης  
 τομή ( $\cap$ )  
 συνένωση  $\bowtie$   
 συνένωση ισότητας  
 φυσική συνένωση ( $*$ )  
 διαίρεση

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

## Η Γλώσσα SQL

*What men or gods are these? What maidens loth?*

*What mad pursuit? What struggle to escape?*

*What pipes and timbrels? What wild ecstasy?*

John Keats, Ode on a Grecian Urn

*What is the average salary in the Toy department?*

Anonymous SQL user

- 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 Βάσεων Δεδομένων SQL

(Μέρος 1: Βασική Δομή, Πράξεις Συνόλου)

### Βασική Δομή

Μια χαρακτηριστική ερώτηση σε SQL έχει την εξής μορφή:

```

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

ονόματα γνωρισμάτων  
 ονόματα σχέσεων  
 συνθήκη

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

```

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

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

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

Ποια γνωρίσματα θέλουμε να υπάρχουν στο αποτέλεσμα της ερώτησης.

```

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

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

Ποιες σχέσεις θα χρησιμοποιηθούν για τον υπολογισμό του αποτελέσματος.

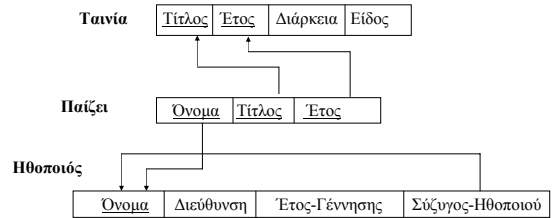
```

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

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

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

- Όταν δεν υπάρχει το **where**, το P θεωρείται ότι ισχύει.
- **ΠΡΟΣΟΧΗ:** Δε γίνεται απαλοιφή των διπλών εμφανίσεων.



Παράδειγμα:

Ονόματα ηθοποιών που παίζουν στην ταινία Gone by the Wind

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

### Select

Παράδειγμα: Ονόματα όλων των ηθοποιών που έχουν παίξει σε ταινίες

```
select Όνομα
from Παίζει
```

- 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
```

## 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

### Where

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

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

## Where

### συνθήκη του where

Λογικοί τελεστές: **and**, **or**, **not**

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

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

## 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
```

## Βασική Δομή

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

<όνομα-σχέσης>. <όνομα-γνωρίσματος>

Παράδειγμα φυσικής συνένωσης:

Τους ηθοποιούς που παίζουν σε ασπρόμαυρες ταινίες

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

## Βασική Δομή

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)

Παιζει(Όνομα, Τίτλος, Έτος)

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

Παραδείγματα

- Εγχρωμες ταινίες με διάρκεια μέχρι 80 λεπτά
- Οι ηθοποιοί που γεννήθηκαν μετά το 1935 και έπαιξαν σε ασπρόμαυρες ταινίες πριν το 1945

• 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

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 των Πλειάδων

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 άμεσα 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.*

Αλλαγή Ονόματος

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**

Για παράδειγμα:

```
select Τίτλος, Έτος, Διάρκεια / 60 as Ωρες-Διάρκεια, Είδος
from Ταινία
```

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**:

```
select distinct Όνομα
from Παίζει as Π1, Ταινία as T
where Π1.Τίτλος = T.Τίτλος and Π1.Έτος = T.Έτος and Είδος =
"Ασπρόμαυρη"
```

- Οι μεταβλητές πλειάδων είναι ιδιαίτερα χρήσιμες όταν θέλουμε να συγκρίνουμε δυο πλειάδες της ίδιας σχέσης.

*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

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

```
select Αριθμός-Δανείου
from Δάνειο
where Ποσό is 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(\*)

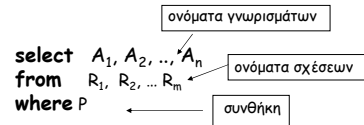
Παράδειγμα:  

```
select sum(Ποσό)
from Δάνειο
```

# Η Γλώσσα Βάσεων Δεδομένων SQL

(Μέρος 1: Βασική Δομή, Πράξεις Συνόλου)

Μια χαρακτηριστική ερώτηση σε SQL έχει την εξής μορφή:



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

## 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.

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

<όνομα-σχέσης>. <όνομα-γνωρισματος>

- 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**

- Οι **μεταβλητές πλειάδων** είναι ιδιαίτερα χρήσιμες όταν θέλουμε να συγκρίνουμε δυο πλειάδες τις ίδιες σχέσης.

## Πράξεις με Συμβολοσειρές

Η πιο συνηθισμένη πράξη είναι ταίριασμα προτύπων:  
**%** ταίριαζει οποιαδήποτε συμβολοσειρά  
**\_** ταίριαζει 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).

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



Η τιμή null

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

Πράξεις Συνόλων

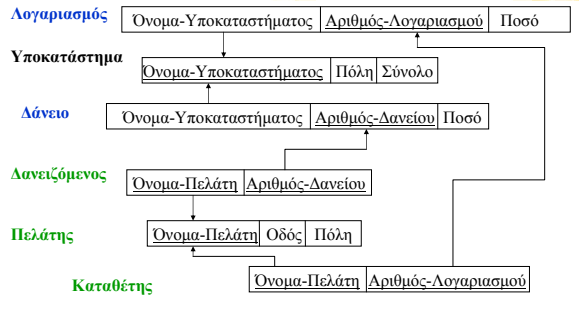
Πράξεις:

- **union**
- **intersection**
- **except**

εφαρμόζονται σε συμβατές σχέσεις.

Γενική Σύνταξη:

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



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)
```

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

Μέγιστος αριθμός πολλαπλών εμφανίσεων;

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 )
```

Αντίστοιχα υπάρχει το **intersect all**

Μέγιστος αριθμός πολλαπλών εμφανίσεων;

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)
```

Αντίστοιχα υπάρχει το **except all**

Μέγιστος αριθμός πολλαπλών εμφανίσεων;

**Select**

- Διαγραφή διπλότυπων: **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.

- Όταν το ίδιο γνωρίσμα 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** είναι ιδιαίτερα χρήσιμες όταν θέλουμε να συγκρίνουμε δυο πλειάδες τις ίδιες σχέσης.

**Πράξεις με Συμβολοσειρές**

Η πιο συνηθισμένη πράξη είναι ταίριασμα προτύπων:

**%** ταίριαζει οποιαδήποτε συμβολοσειρά

**\_** ταίριαζει 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 γνωρίσμα

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

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

Πράξεις Συνόλων

Πράξεις:

- union
- intersection
- except

εφαρμόζονται σε συμβατές σχέσεις.

- Σύνταξη

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

- Απαλοιφή διπλών εμφανίσεων, εκτός αν χρησιμοποιηθεί το union all

# Η Γλώσσα SQL

(Μέρος 2: Φωλιασμένες Υπο-ερωτήσεις, Γλώσσα Ορισμού)

## Φωλιασμένες 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.

Γενική δομή:

select ...  
from ...  
where

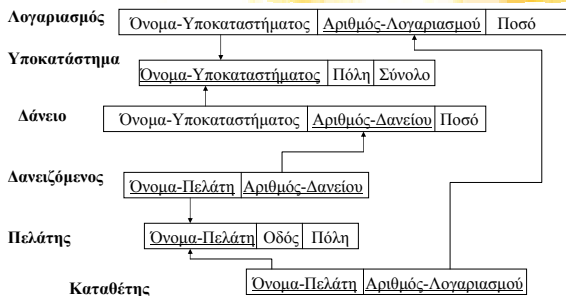
<ΤΕΛΕΣΤΗΣ>

(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 ΤΕΛΕΣΤΗΣ



## 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.

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)
```

• Παραπάνω από δύο γνωρίσματα

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

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

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

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

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

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

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

Μπορεί να χρησιμοποιηθεί και 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")
```

### Σύγκριση Συνόλων

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

Παράδειγμα: Τα ονόματα όλων των υποκαταστημάτων που έχουν καταθέσεις μεγαλύτερες από τις καταθέσεις ενός τουλάχιστον υποκαταστήματος των Ιωαννίνων.

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

• επίσης:

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

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

Παράδειγμα: Τα ονόματα όλων των υποκαταστημάτων που έχουν καταθέσεις μεγαλύτερες από τις καταθέσεις όλων των υποκαταστημάτων των Ιωαννίνων.

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

Φωλιασμένες Υπο-ερωτήσεις

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

Παράδειγμα: ;

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

Φωλιασμένες Υπο-ερωτήσεις

• επίσης:

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

Φωλιασμένες Υπο-ερωτήσεις

Παράδειγμα: Το υποκατάστημα με το μεγαλύτερο μέσο ποσό καταθέσεων.

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

Φωλιασμένες Υπο-ερωτήσεις

3. Έλεγχος για άδεια σχέση

Ο τελεστής **exists**: επιστρέφει true αν η υποερώτηση δεν είναι κενή

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)
```

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 ⊇ B
```

• Ποια πράξη της σχεσιακής άλγεβρας;

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 σε όλες τις ταινίες της Βουγιουκλάκη

B: όλες οι ταινίες της Βουγιουκλάκη

A: όλες οι ταινίες του συγκεκριμένου ηθοποιού

```
not exists (B except A)
```

```
select distinct S.Όνομα
from Παίζει as S
where not exists
```

```
((select Τίτλος, Έτος
from Παίζει
where Όνομα = "Βουγιουκλάκη")
except
(select Τίτλος, Έτος
from Παίζει as R
where R.Όνομα = S.Όνομα))
```

υπολογισμός για κάθε S

## 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 ))
not exists (B except A)
```

## Φωλιασμένες Υπο-ερωτήσεις

### 4. Έλεγχος για Διπλές Εμφανίσεις

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

## Φωλιασμένες Υπο-ερωτήσεις

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)

Παίξει(Όνομα, Τίτλος, Έτος)

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

Παράδειγμα: Οι ηθοποιοί που έχουν παίξει το πολύ σε μια ταινία

```
select Όνομα
from Παίξει as T
where unique (select T.Όνομα
              from Παίξει as R
              where T.Όνομα = R.Όνομα)
```

**select** Όνομα (θα το δούμε στη συνέχεια)

```
from Παίξει
group by Όνομα
having count(*) < 1
```

## 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")
```

## Φωλιασμένες Υπο-ερωτήσεις

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

```
select T.Όνομα-Πελάτη
from Καταθέτης as T
where not 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")
```

## Φωλιασμένες Υπο-ερωτήσεις (επανάληψη)

Ο τελεστής μπορεί να είναι:

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

## Η Γλώσσα SQL

(Μέρος 2: Ξηλωτισμένες Υπο-ερωτήσεις, Γλώσσα Ορισμού)

Η SQL έχει διάφορα τμήματα:

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

### Γλώσσα Ορισμού Δεδομένων

#### Γλώσσα Ορισμού Δεδομένων (ΓΟΔ)

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

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

### Γλώσσα Ορισμού Δεδομένων

```
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.

### Πεδία Τιμών

#### Τύποι Πεδίου Ορισμού

Για τον ορισμό του πεδίου ορισμού, οι διαθέσιμοι built-in τύποι περιλαμβάνουν:

**char**(n) (σταθερού μήκους)  
**varchar**(n)  
**int**  
**smallint**  
**numeric**(p, d) (d από τα p ψηφία είναι στα δεξιά της υποδιαστολής)  
**real**, **double precision**  
**float**(n)  
**date** (ημερομηνία)  
**time** (ώρα)

### Πεδία Τιμών

Ο ορισμός πεδίου μπορεί να περιέχει τον προσδιορισμό **not null**

Επίσης, επιτρέπεται δημιουργία πεδίου:

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

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

## Ορισμός Σχήματος

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

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<sub>1j</sub>, A<sub>2j</sub>, ..., A<sub>ij</sub>, (δεν επιτρέπονται επαναλαμβανόμενες τιμές και NULL τιμές)
- **unique** A<sub>1j</sub>, A<sub>2j</sub>, ..., A<sub>ij</sub>, (δεν επιτρέπονται επαναλαμβανόμενες τιμές; NULL τιμές επιτρέπονται)
- **check** P
- **foreign key** (A<sub>i</sub>) **references** A<sub>j</sub>

Παραδείγματα

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

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

Επίσης, πιο περίπλοκες συνθήκες, π.χ., για ξένα κλειδιά:

```
check (Όνομα-Υποκαταστήματος in select Όνομα-Υποκαταστήματος  

from Υποκατάστημα)
```

## Περιορισμοί Αναφοράς

Σύνταξη:

```
foreign key (Ai) references Aj
```

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

```
on delete cascade  

on update cascade
```



Παράδειγμα

**create table**

..

**foreign key** (Όνομα-Υποκαταστήματος) **references** Υποκατάστημα  
**on delete cascade**  
**on update cascade**

...

## Περιορισμοί Ακεραιότητας

## Πεδίου ορισμού

Χρησιμοποιώντας την εντολή **check**:

Παράδειγματα

(1) Ελάχιστος ωρομίσθιο

**create domain** Ωρομίσθιο **numeric**(5, 2)  
**constraint** Έλεγχος-Ωρομισθίου **check**(Ποσό >= 4.00)

(2) Να μην περιέχει την τιμή null

**create domain** Πεδίο-Αριθμός-Λογαριασμού **char**(10)  
**constraint** Έλεγχος-Αριθμός-Λογαριασμού **check**(value not null)

(3) Να παίρνει συγκεκριμένες τιμές

**create domain** Τύπος-Λογαριασμού **char**(10)  
**constraint** Έλεγχος-Τύπος-Λογαριασμού **check value in** (' Όψεως',  
'Ταμειευτήριου')

## 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

## Τροποποίηση Σχήματος

`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 3: Γλώσσα Τροποποίησης, Συναθροιστικές Συναρτήσεις, Συνενώσεις, Όψεις Ενσωματωμένη SQL)

## Φωλιασμένες Υπο-ερωτήσεις (επανάληψη)

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

Παράδειγμα:

```
select Τίτλος
from Ταινία
where Διάρκεια > any (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)

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>
```

<x> μπορεί να είναι  
T {=, <, <=, >, >=, <>} any(some), all  
T in  
exists, unique  
(όπου T πλειάδα)

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

## Φωλιασμένες Υπο-ερωτήσεις (επανάληψη)

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

Παράδειγμα:

```
select Τίτλος
from Ταινία
where exists (select Διάρκεια
             from Ταινία
             where Είδος = έγχρωμη)

select Τίτλος
from Ταινία
where unique (select Διάρκεια
              from Ταινία
              where Είδος = έγχρωμη)
```

Μία συνθήκη του where

## 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

## Γλώσσα Ορισμού Δεδομένων (ΓΟΔ)

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

- του σχήματος κάθε σχέσης
- του πεδίου τιμών κάθε γνωρίσματος
- των περιορισμών ακεραιότητας

- Τροποποίηση Βάσης Δεδομένων  
Γλώσσα Χειρισμού Δεδομένων (ΓΧΔ)

## Τροποποιήσεις

1. Διαγραφή
2. Εισαγωγή
3. Ενημέρωση

## Εισαγωγή

Για να εισάγουμε δεδομένα σε μια σχέση είτε

(α) προσδιορίζουμε την πλειάδα, είτε

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

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

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

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 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"
```

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
```

Επίσης, εισαγωγή null τιμών:

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

## 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.

Παραδείγματα

(1) Όλους τους λογαριασμούς του Παπαδόπουλου

```
delete from Καταθέτης
where Όνομα-Πελάτη = "Παπαδόπουλος"
```

(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*

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: 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")
```

Πρέπει να διαγράψουμε και όλους τους λογαριασμούς:

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

ΠΡΟΣΟΧΗ: 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.

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.

## Ενημερώσεις

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

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

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;

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
```

# Η Γλώσσα SQL

(Μέρος 3: Γλώσσα Τροποποίησης, **Συναθροιστικές Συναρτήσεις**, Συνενώσεις, Όψεις Ενσωματωμένη SQL)

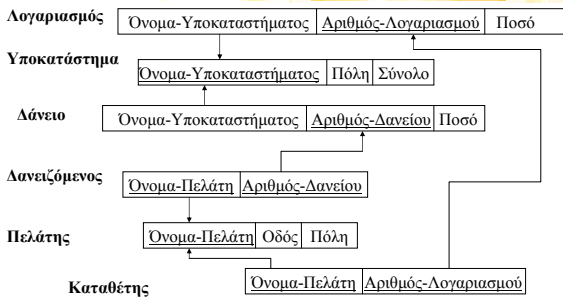
## Συναθροιστικές Συναρτήσεις

### 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)**

## Συναθροιστικές Συναρτήσεις



## Συναθροιστικές Συναρτήσεις

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

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

Το αποτέλεσμα είναι μια σχέση με ένα γνώρισμα και μια γραμμή, μπορούμε να δώσουμε όνομα στο γνώρισμα χρησιμοποιώντας το **as**

## Συναθροιστικές Συναρτήσεις

*Παράδειγμα: Μέγιστο ποσό όλων των λογαριασμών στο υποκατάστημα Καλούτσανη και τον αριθμό του λογαριασμού!!*

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

Αν το select συναθροιστική, τότε **μόνο συναθροιστικές**, εκτός αν υπάρχει group by

## Συναθροιστικές Συναρτήσεις

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: Μέσο ποσό των λογαριασμών σε κάθε υποκατάστημα*

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

## Συναθροιστικές Συναρτήσεις

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

Παράδειγμα: Αριθμός καταθετών σε κάθε υποκατάστημα

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

## Συναθροιστικές Συναρτήσεις

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

Μέσος όρος καταθέσεων ανά πελάτη και ανά υποκατάστημα

```
SELECT Όνομα-Υποκαταστήματος, Όνομα-Πελάτη, avg(balance)
FROM account
GROUP BY Όνομα-Υποκαταστήματος, Όνομα-Πελάτη
```

Ομαδοποίηση γίνεται πρώτα ως προς το branch\_name. Στην συνέχεια δημιουργούνται υποομάδες ως προς το customer\_name

## Συναθροιστικές Συναρτήσεις

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

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
```

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.

## Συναθροιστικές Συναρτήσεις

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

3

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

## Συναθροιστικές Συναρτήσεις

Για να μετρήσουμε πόσες πλειάδες έχει μια σχέση:

```
select count (*)
from Πελάτης
```

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

**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

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

Περαιτέρω για τη γλώσσα ερωτήσεων  
 - Συνενώσεις Συνόλων

- Ορισμός Όψευων

Συνενώσεις Συνόλων

Η SQL--92 υποστηρίζει διάφορους τύπους συνενώσεων που συνήθως χρησιμοποιούνται στο **for**, 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>

Τύποι Συνένωσης:

**inner join**: εσωτερική (θήτα) συνένωση  
**left outer join**: αριστερή εξωτερική συνένωση  
**right outer join**  
**full outer join**

Συνθήκες Συνένωσης:

**on P**  
**using** (A<sub>1</sub>, A<sub>2</sub>, ..., A<sub>n</sub>): γνωρίσματα που πρέπει να ταιριάζουν στη συνένωση είναι τα A<sub>i</sub>. Τα A<sub>i</sub> πρέπει να είναι γνωρίσματα κοινά και στις δυο σχέσεις και εμφανίζονται στο αποτέλεσμα μόνο μια φορά.

- Για την εσωτερική συνένωση η συνθήκη δεν είναι υποχρεωτική, όταν λείπει ισοδυναμεί με καρτεσιανό γινόμενο.

```
select A1, A2, .. An
from R1 outer join R2 on R1.Ai=R2.Ai
```



Οι λέξεις κλειδιά **inner** και **outer** είναι προαιρετικές.

**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
```

## Παραγόμενες Σχέσεις

- Η SQL-92 δίνει τη δυνατότητα μια υπο-ερώτηση να χρησιμοποιηθεί στο **from**
- Τότε πρέπει να της δοθεί ένα όνομα και τα γνωρίσματα της να μετονομαστούν
- Αυτό γίνεται χρησιμοποιώντας το **as**

Η SQL-92 δίνει τη δυνατότητα χρησιμοποιώντας το **as** να δοθεί ένα προσωρινό όνομα σε μία προσωρινή σχέση που προκύπτει από μια υποερώτηση.

*Παράδειγμα: Το μέσο υπόλοιπο για όλα τα υποκαταστήματα για τα οποία το μέσο ποσό είναι μεγαλύτερο των \$1200*

```
select Όνομα-Υποκαταστήματος, Μέσο-υπόλοιπο
from (select Όνομα-Υποκαταστήματος, avg(Ποσό)
      from Καταθέτης
      group by Όνομα-Υποκαταστήματος
      as Αποτέλεσμα(Όνομα-Υποκαταστήματος, Μέσο-υπόλοιπο)
where Μέσο-Υπόλοιπο > 1200
```

## Ορισμός Όψεων

view definition

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>
```

*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
```

- 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>
```

## Ενσωματωμένη SQL

Προσπέλαση μιας ΒΔ μέσα από μια γλώσσα προγραμματισμού γενικού σκοπού απαιτείται τουλάχιστον γιατί:

- υπάρχουν ερωτήσεις που δε μπορούν να διατυπωθούν σε SQL, γιατί η SQL δεν έχει όλες τις δυνατότητες μιας γλώσσας προγραμματισμού γενικού σκοπού
- μη-δηλωτικές εντολές (π.χ., εκτύπωση, επικοινωνία με το χρήστη) δε μπορούν να γίνουν μέσω της SQL

Ενσωμάτωση της SQL σε μια γλώσσα που καλείται *φιλόξενη* (host)

Σε αυτήν την περίπτωση, η επεξεργασία των ερωτήσεων γίνεται από τη ΒΔ, και το αποτέλεσμα γίνεται διαθέσιμο στο πρόγραμμα *μια πλειάδα τη φορά*

Ένας ειδικός προ-επεξεργαστής (preprocessor) αντικαθιστά τον ενσωματωμένο κώδικα της SQL με δηλώσεις και κλήσεις συναρτήσεων στη host γλώσσα και μεταφράζεται το πρόγραμμα

Σύνταξη της μορφής:

```
EXEC SQL < embedded SQL statement > END-EXEC
```

Η ακριβής σύνταξη εξαρτάται από τη host γλώσσα

Χρησιμοποιούμε την εντολή: **SQL INCLUDE**, για να δηλώσουμε στον προ-επεξεργαστή που πρέπει να εισάγει τις δηλώσεις των μεταβλητών της SQL

Μεταβλητές της γλώσσας μπορεί να χρησιμοποιηθούν στην εντολή της SQL αν το σύμβολο : προηγείται του ονόματός τους

Για να γράψουμε μια ερώτηση σε SQL:

**EXEC SQL**

```
declare c cursor for
select Όνομα-Πτελάτη, Πόλη
from Κατάθεση, Πτελάτης
where Κατάθεση.Όνομα-Πτελάτη = Πτελάτης.Όνομα-Πτελάτη
and Κατάθεση.Ποσό > :amount
```

**END-EXEC**

Η παραπάνω εντολή δεν προκαλεί την εκτέλεση της ερώτησης, για να εκτελεστεί:

```
EXEC SQL open c END-EXEC
```

Το αποτέλεσμα σώζεται σε μια προσωρινή σχέση, αν υπάρχει λάθος το διαγνωστικό μήνυμα σώζεται σε μια ειδική μεταβλητή (SQLCA)

Οι πλειάδες του αποτελέσματος γίνονται διαθέσιμες στο πρόγραμμα μέσω μιας σειράς από **fetch** εντολές, χρειάζεται μια μεταβλητή της host γλώσσας για κάθε γνώρισμα

```
EXEC SQL fetch c into :cn, :cc END-EXEC
```

Για να σβήσουμε τη προσωρινή σχέση:

```
EXEC SQL close c END-EXEC
```

Επίσης:

```
EXEC SQL <update, insert ή delete έκφραση> END-EXEC
```

## Δυναμική SQL

Τα προγράμματα μπορούν να δημιουργούν ερωτήσεις σε SQL ως συμβολοσειρές δυναμικά κατά την εκτέλεση και είτε να τα εκτελούν αμέσως είτε να τα προετοιμάζουν (τα μεταφράζουν) για να χρησιμοποιηθούν αργότερα

Παράδειγμα:

```
char * sqlprog = "update Λογαριασμός set Ποσό = Ποσό * 1.05
where Αριθμός-Λογαριασμού = ?"
EXEC SQL prepare dynprog from :sqlprog;
char account[10] = "A-101";
EXEC SQL execute dynprog using :account;
```

## Άλλα Χαρακτηριστικά

- Γλώσσες 4ης Γενιάς
- Έννοια του *session* μεταξύ ενός *client* και του *server* του ΣΒΔ
- Δημιουργία σχήματος: `create schema` και
- Σβήσιμο σχήματος : `drop schema`