

# Η Γλώσσα SQL

(Μέρος 2: Φωλιασμένες Υπο-ερωτήσεις, Συνενώσεις, Όψεις)

## Η γλώσσα SQL

### Επανάληψη

- Βασική Σύνταξη Γλώσσας Χειρισμού Δεδομένων (ΓΧΔ)  
-- `select-from-where`
- Περισσότερα για τη γλώσσα ερωτήσεων
  - Αλλαγή Ονόματος
  - Μεταβλητές Πλειάδων
  - Πράξεις με Συμβολοσειρές
  - Διάταξη Πλειάδων
  - Η τιμή null
- Πράξεις Συνόλων
- Συναθροιστικές Συναρτήσεις

## Βασική Δομή

Μια χαρακτηριστική ερώτηση σε SQL έχει την εξής μορφή:

$$\begin{array}{l} \text{select } A_1, A_2, \dots, A_n \\ \text{from } R_1, R_2, \dots, R_m \\ \text{where } P \end{array}$$

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

- Αριθμητικές πράξεις (+, -, \*, /) ανάμεσα σε σταθερές ή γνωρίσματα πλειάδων
- Διαγραφή διπλότιμων: **select distinct**

## Συνθήκη του where

Λογικοί τελεστές: **and, or, not**

Τελεστές σύγκρισης: <, <=, >, >=, =, <>, **between, not between**  
 ανάμεσα σε αριθμητικές εκφράσεις, συμβολοσειρές (strings),  
 και ειδικούς τύπους.

## Βασική Δομή

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

<όνομα-σχέσης>.<όνομα-γνωρίσματος>

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

<παλιό-όνομα> **as** <νέο-όνομα>

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

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

## Συμβολοσειρές. Διάταξη

### 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
```

### 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 τιμή null

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

## Συναθροιστικές Συναρτήσεις

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

Ελάχιστο: **min(A)**

Μέγιστο: **max(A)**

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

Πλήθος: **count(A)**, όπου **A** γνώρισμα

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

**group by:** για να εφαρμόσουμε τις συναρτήσεις όχι μόνο σε ένα σύνολο από πλειάδες, αλλά σε **ομάδες από σύνολα πλειάδων**.

**having:** για να εφαρμόσουμε μια συνθήκη σε μια συγκεκριμένη ομάδα από πλειάδες. 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
```

## Φωλιασμένες Υποερωτήσεις

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:

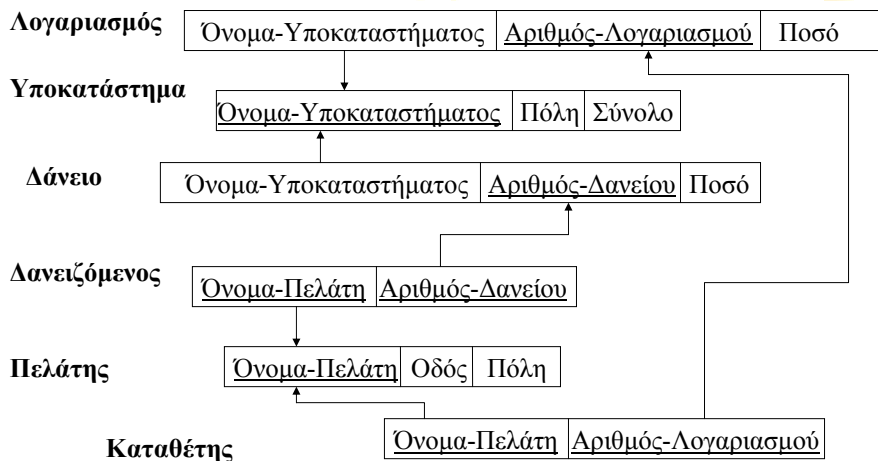
**select ...**  
**from ...**  
**where**

**? (select ...**  
**from ...**  
**where ... );**

Υπο-ερώτηση



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



## 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)
```



## Φωλιασμένες Υπο-ερωτήσεις

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)

Παίζει(Όνομα, Τίτλος, Έτος)

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

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

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

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

## Φωλιασμένες Υπο-ερωτήσεις

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**)

## Φωλιασμένες Υπο-ερωτήσεις

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)
```

### 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)
```

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;

## 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

not exists (B except A)

**A:** όλες οι ταινίες του συγκεκριμένου ηθοποιού

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

```
((select Τίτλος, Έτος
from Παίζει
where Όνομα = "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
```

not exists (B except A)

```
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 ))
```

#### 4. Έλεγχος για Διπλές Εμφανίσεις

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

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)

Παίζει(Όνομα, Τίτλος, Έτος)

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

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

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

```
select Όνομα
from Παίζει
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")
```



- 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)

Περισσότερα για τη γλώσσα ερωτήσεων  
- Συνενώσεις Συνόλων

- Ορισμός Όψεων

## 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** ( $A_1, A_2, \dots, A_n$ ): γνωρίσματα που πρέπει να ταιριάζουν στη συνένωση είναι τα  $A_i$ . Τα  $A_i$  πρέπει να είναι γνωρίσματα κοινά και στις δύο 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.Aj
```

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

Μπορούμε να ορίσουμε μια όψη χρησιμοποιώντας την εντολή:

```
create view <όνομα--όψης> as <select-from-where ερώτηση>
```

view  
definition

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 (Σύνολο-Δανείων, Όνομα-  
Υποκαταστήματος) 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>