

# Συστήματα Βάσεων Δεδομένων

Δρ. Κρηνίδης Στυλιανός  
*stkrini@immail.teikav.edu.gr*

*Τμήμα Διαχείρισης Πληροφοριών  
Τεχνολογικό Εκπαιδευτικό Ίδρυμα Καβάλας*

*Η ψηφιοποίηση του εκπαιδευτικού υλικού έγινε στο πλαίσιο υλοποίησης της πράξης με τίτλο «ΕΝΙΣΧΥΣΗ ΣΠΟΥΔΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ στο ΤΕΙ ΚΑΒΑΛΑΣ», του Μέτρου 2.2 «Αναμόρφωση Προγραμμάτων Σπουδών - Διεύρυνση Τριτοβάθμιας Εκπαίδευσης» του ΕΠΕΑΕΚ II, που συγχρηματοδοτείται από το Ευρωπαϊκό Κοινωνικό Ταμείο (Ε.Κ.Τ.) κατά 80% και Εθνικούς πόρους κατά 20%.*



*Ανάπτυξη παντού. Ανάπτυξη για όλους.*

ΥΠΟΥΡΓΕΙΟ ΕΘΝΙΚΗΣ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΕΑΕΚ  
ΕΥΡΩΠΑΪΚΗ ΕΝΩΣΗ  
ΣΥΓΧΡΗΜΑΤΟΔΟΤΗΣΗ  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ  
ΕΥΡΩΠΑΪΚΟ ΤΑΜΕΙΟ ΠΕΡΙΦΕΡΕΙΑΚΗΣ ΑΝΑΠΤΥΞΗΣ



Η ΠΑΙΔΕΙΑ ΣΤΗΝ ΚΟΡΥΦΗ  
Επιχειρησιακό Πρόγραμμα  
Εκπαίδευσης και Αρχικής  
Επαγγελματικής Κατάρτισης

# SQL (Structured Query Language)

- Η πιο ευρέως χρησιμοποιούμενη γλώσσα ερωτημάτων σε εμπορικές εφαρμογές
- Μια γλώσσα ερωτημάτων, με δυνατότητες πολύ περισσότερες του ενός απλού ερωτήματος σε μια βάση δεδομένων
- Η SQL μπορεί:
  - να ορίσει τη δομή των δεδομένων
  - να τροποποιήσει τα δεδομένα της ΒΔ
  - να καθορίσει περιορισμούς ασφάλειας



# SQL (Structured Query Language)

## ➤ Η SQL:

- Βασίζεται στη σχεσιακή άλγεβρα
- Είναι μία **δηλωτική** γλώσσα (μη-διαδικαστική) στην οποία περιγράφουμε τι θέλουμε και όχι πως θα το αποκτήσουμε
- Μπορεί να ενσωματωθεί σε άλλες γλώσσες προγραμματισμού
- Το λογισμικό του ΣΔΒΔ αναλαμβάνει την εύρεση του τρόπου ανάκτησης των δεδομένων από τη βάση



# SQL – DDL (1)

- Εντολές γλώσσας ορισμού δεδομένων (DDL):
  - CREATE TABLE (δημιουργία πίνακα)
  - CREATE INDEX (δημιουργία ευρετηρίου)
  - CREATE VIEW (δημιουργία όψης)
  - DROP TABLE (διαγραφή πίνακα)
  - DROP INDEX (διαγραφή ευρετηρίου)
  - DROP VIEW (διαγραφή όψης)
- ALTER (τροποποίηση της δομής των αντικειμένων της βάσης, δηλ., πινάκων, ευρετηρίων και όψεων)



# SQL – DDL (2)

- Τύποι πεδίου ορισμού στην SQL:
  - **char(n)**: σταθερού μήκους συμβολοσειρά, με μήκος  $n$
  - **varchar(n)**: μεταβλητού μήκους συμβολοσειρά, με μέγιστο μήκος  $n$
  - **int**: ακέραιος (το εύρος του εξαρτάται από τον υπολογιστή)
  - **smallint**: μικρός ακέραιος (υποσύνολο των ακεραίων, το εύρος του εξαρτάται από τον υπολογιστή)
  - **numeric(p,d)**: αριθμός σταθερής υποδιαστολής, με ακρίβεια  $p$  ψηφίων εκ των οποίων τα  $d$  θα είναι μετά την υποδιαστολή
  - **real, double precision**: αριθμός κινητής υποδιαστολής απλής και διπλής ακρίβειας αντίστοιχα
  - **float(n)**: αριθμός κινητής υποδιαστολής (το  $n$  καθορίζει την ελάχιστη ακρίβεια που μπορούμε να έχουμε)



# SQL – DDL (3)

- Τύποι πεδίου ορισμού στην SQL:
  - **date**: ημερομηνίες που περιέχουν τετραψήφιο έτος, μήνα και ημέρα
    - π.χ., *date* “2007-7-27”
  - **time**: ώρα της ημέρας (ώρες, λεπτά, δευτερόλεπτα)
    - π.χ., *time* “09:00:30.25”
  - **timestamp**: ημερομηνία + ώρα
    - π.χ., *timestamp* “2008-5-22 17:44:52.75”
  - **interval**: χρονική περίοδος
    - π.χ., *interval* “1” μέρα
    - Η αφαίρεση μία τιμής date/time/timestamp από μία άλλη δίνει interval τιμή
    - Οι interval τιμές μπορούν να προστεθούν σε τιμές date/time/timestamp



# SQL – DDL (4)

- Περιορισμοί ακεραιότητας:
  - **not null**
  - **primary key** ( $A_1, A_2, \dots, A_n$ )
  - **check (P)**: όπου P είναι ένα κατηγορημα

- Παράδειγμα:

```
create table branch(  
    branch_name    char(15),  
    branch_city    char(30),  
    assets          int not null,  
    primary key (branch_name),  
    check (assets >= 0)  
)
```



# SQL – DDL (5)

- Ένας πίνακας στην SQL ορίζεται με την εντολή **create table**:

```
create table r(A1 D1, A2, D2, ..., An Dn,  
              (περιορισμός ακεραιότητας1),  
              ...,  
              (περιορισμός ακεραιότηταςk))
```

- **r** είναι το όνομα του πίνακα
- Κάθε **A<sub>i</sub>** είναι ένα όνομα χαρακτηριστικού του πίνακα
- **D<sub>i</sub>** είναι ο τύπος δεδομένων των τιμών στον τύπο πεδίο ορισμού του χαρακτηριστικού **A<sub>i</sub>**





# SQL – DDL (6)

## ➤ Παράδειγμα δημιουργίας πίνακα:

```
create table ΥΠΑΛΛΗΛΟΣ(  
    Κωδικός          char(9) not null,  
    Επώνυμο          varchar(15) not null,  
    Πατρώνυμο        char,  
    Όνομα             varchar(15) not null,  
    ΗμερΓεν          date,  
    Διεύθυνση         varchar(30),  
    Φύλο              char,  
    Μισθός            numeric(10,2),  
    Κωδ_Τμημ         int not null,  
    primary key (Κωδικός)  
)
```



# SQL – DDL (7)

➤ Παράδειγμα δημιουργίας πίνακα:

```
create table ΠΡΟΣΤΑΤΕΥΟΜΕΝΟΣ (  
    Κωδ_Υπαλ      char(9) not null,  
    Όνομα_Προστ   varchar(15) not null,  
    Φύλο          char,  
    ΗμερΓεν       date,  
    Σχέση         varchar(8) not null,  
    primary key (Κωδ_Υπαλ, Όνομα_Προστ))
```

```
create table ΤΜΗΜΑ_ΤΟΠΟΘ(  
    Κωδ_Τμήμ      int not null,  
    Τοποθεσία     varchar(15) not null,  
    primary key (Κωδ_Τμήμ, Τοποθεσία))
```



# SQL – DDL (8)

➤ Παράδειγμα διαγραφής πίνακα:

➤ **drop table** ΠΡΟΣΤΑΤΕΥΟΜΕΝΟΣ

➤ **drop table** ΥΠΑΛΛΗΛΟΣ

➤ **drop table** ΤΜΗΜΑ



# SQL – DDL (9)

➤ Παράδειγμα τροποποίησης πίνακα:

➤ **alter table** ΥΠΑΛΛΗΛΟΣ **add** Δουλειά **varchar**(12);

➤ **alter table** ΤΜΗΜΑ **drop column** Κωδ\_Διευθυντή;



# SQL – DDL (10)

- Παράδειγμα δημιουργίας/διαγραφής ευρετηρίου:
  - **create index** Y\_EΠΩΝ\_INDEX **on** ΥΠΑΛΛΗΛΟΣ(Επώνυμο);
  - **create index** Y\_INDEX **on** ΥΠΑΛΛΗΛΟΣ(Επώνυμο asc, Όνομα desc);
  - **create unique index** Y\_ΚΩΔ\_INDEX **on** ΥΠΑΛΛΗΛΟΣ(Κωδικός);
  - **drop index** Y\_EΠΩΝ\_INDEX;
  - **drop index** Y\_ΚΩΔ\_INDEX;



# SQL – DML (1)

- Εντολές γλώσσας χειρισμού δεδομένων (DML):
  - INSERT (εισαγωγή εγγραφών σε πίνακα)
  - DELETE (διαγραφή εγγραφών από πίνακα)
  - UPDATE (ενημέρωση εγγραφών σε πίνακα)
  - SELECT (ανάκτηση δεδομένων από πίνακα)



# SQL – DML (2)

- Εισαγωγή εγγραφών σε πίνακα:
  - INSERT INTO πίνακας (πεδίο, ...) VALUES (τιμή, ...);
    - π.χ., **insert into** ΥΠΑΛΛΗΛΟΣ (Κωδικός, Επώνυμο, Όνομα) **values** (“346298653”, “Πετρίδης”, “Γεώργιος”)
    - π.χ., **insert into** ΥΠΑΛΛΗΛΟΣ **values** (“346298653”, “Πετρίδης”, “Δ”, “Γεώργιος”, “1970-10-20”, “Δελφών 12”, “Α”, 700, 4)



# SQL – DML (3)

- Διαγραφή εγγγραφών από πίνακα
  - DELETE FROM πίνακας WHERE συνθήκη;
    - π.χ., **delete from ΥΠΑΛΛΗΛΟΣ where Κωδικός="346298653";**
    - π.χ., **delete from ΥΠΑΛΛΗΛΟΣ where Επώνυμο="Παυλίδης";**
    - π.χ., **delete from ΥΠΑΛΛΗΛΟΣ where Μισθός>=1000;**





# SQL – DML (4)

- Ενημέρωση εγγγραφών σε πίνακα
  - UPDATE πίνακας SET πεδίο=τιμή WHERE συνθήκη;
    - π.χ., **update** ΤΜΗΜΑ **set** Τοποθεσία="Καβάλα" **where** Κωδ\_Τμήμ=4;
    - π.χ., **update** ΥΠΑΛΛΗΛΟΣ **set** Μισθός=Μισθός\*1.1 **where** Κωδ\_Τμήμ=1;



# SQL – Select (1)

- Ανάκτηση δεδομένων από πίνακα

**select**  $A_1, A_2, \dots, A_n$   
**from**  $r_1, r_2, \dots, r_m$   
**where**  $P$

- Τα  $A_i$  αναπαριστούν πεδία
- Τα  $r_i$  αναπαριστούν πίνακες
- Το  $P$  είναι συνθήκη

- Ισοδύναμη έκφραση σχεσιακής άλγεβρας

$\Pi_{A_1, A_2, \dots, A_n}(\sigma_P(r_1 \times r_2 \times \dots \times r_m))$



# SQL – Select (2)

- Τελεστές που συνδυάζονται με τη SELECT
  - \* όλα τα πεδία του πίνακα στον οποίο αναφέρεται η FROM
  - **DISTINCT** παραλείπει τις διπλές εγγραφές
    - ΠΡΟΣΟΧΗ! Η SQL, αντίθετα από την σχεσιακή άλγεβρα, δεν απαλείφει αυτόματα τις διπλοεγγραφές



# SQL – Select (3)

```
select A1, A2, ..., An  
from r1, r2, ..., rm  
where P
```

- Η πρόταση FROM

- Αντιστοιχεί στην πράξη του καρτεσιανού γινομένου της σχεσιακής άλγεβρας

- Η πρόταση WHERE

- Αντιστοιχεί στην πράξη επιλογής της σχεσιακής άλγεβρας



# SQL – Select (4)

## ➤ Τελεστές συνθήκης

➤ >, >=, <, <=, <>, =

αριθμητικοί

➤ BETWEEN ... AND

αριθμητικοί

➤ AND, OR, NOT

λογικοί

➤ LIKE

σύγκριση συμβολοσειρών

➤ IS [NOT] NULL

αναζήτησης [μη] κενών τιμών



# SQL – Select (5)

## ➤ Παράδειγμα:

➤ Βρες τα ονόματα όλων των υπαλλήλων της εταιρίας μας  
`select Επώνυμο, Όνομα`  
`from ΥΠΑΛΛΗΛΟΣ`

➤ Ο “\*” στο τμήμα select σημαίνει “όλα τα χαρακτηριστικά”  
`select *`  
`from ΥΠΑΛΛΗΛΟΣ`



# SQL – Select (6)

- Η SQL επιτρέπει πολλαπλές εμφανίσεις της ίδιας πλειάδας στο αποτέλεσμα μίας ερώτησης. Για την απαλοιφή των πολλαπλών εμφανίσεων, εισάγουμε τη δεσμευμένη λέξη **distinct** μετά το **select**
- Παράδειγμα: Βρείτε τα ονόματα των υπαλλήλων

```
select distinct Όνομα  
from ΥΠΑΛΛΗΛΟΣ
```



# SQL – Select (7)

- Το τμήμα `select` μπορεί να περιέχει αριθμητικές εκφράσεις περιλαμβάνοντας τις πράξεις `+`, `-`, `*` και `/` ανάμεσα σε σταθερές ή χαρακτηριστικά πλειάδων

- Παράδειγμα:

```
select Επώνυμο, Όνομα, Μισθός*1.1  
from ΥΠΑΛΛΗΛΟΣ
```

Θα επιστρέψει μία σχέση η οποία είναι ίδια με τη σχέση ΥΠΑΛΛΗΛΟΣ, εκτός από το χαρακτηριστικό *Μισθός*, το οποίο έχει πολλαπλασιαστεί επί 1.1





# SQL – Select (8)

- Το τμήμα **where** αντιστοιχεί στη συνθήκη της πράξης της επιλογής στη σχεσιακή άλγεβρα. Αποτελείται από ένα κατηγορημα που περιέχει χαρακτηριστικά των σχέσεων που εμφανίζονται στην πρόταση **from**
- Παράδειγμα: Βρείτε όλους τους υπαλλήλους που δουλεύουν στο τμήμα με κωδικό 4 και παίρνουν μισθό μεγαλύτερο από 1000€

```
select Επώνυμο, Όνομα  
from ΥΠΑΛΛΗΛΟΣ  
where Κωδ_Τμήμ=4 AND Μισθός>1000
```



# SQL – Select (9)

- Παράδειγμα: Βρείτε όλους τους υπαλλήλους που γεννήθηκαν μετά το 1970 και παίρνουν μισθό μεγαλύτερο από 1000€

```
select Επώνυμο, Όνομα  
from ΥΠΑΛΛΗΛΟΣ  
where ΗμερΓεν>=01/01/1970 AND Μισθός>1000
```

- Παράδειγμα: Βρείτε όλους τους υπαλλήλους που παίρνουν μισθό από 1000€ έως 2000€

```
select Επώνυμο, Όνομα  
from ΥΠΑΛΛΗΛΟΣ  
where Μισθός>=1000 AND Μισθός<=2000  
[where Μισθός between 1000 AND 2000]
```



# SQL – Select (10)

- Παράδειγμα: Βρείτε την ημερομηνία γέννησης και τη διεύθυνση του Παπαδόπουλου Γεώργιου του Δημητρίου

```
select Διεύθυνση, ΗμερΓεν  
from ΥΠΑΛΛΗΛΟΣ  
where Επώνυμο="Παπαδόπουλος" AND Όνομα="Γεώργιος"  
AND Πατρώνυμο="Δ"
```



# SQL – Select (11)

- Το τμήμα **from** αντιστοιχεί στο καρτεσιανό γινόμενο της σχεσιακής άλγεβρας. Παραθέτει τους πίνακες που πρέπει να σαρωθούν για την αποτίμηση της έκφρασης
- Παράδειγμα: Βρείτε όλους τους υπαλλήλους που δουλεύουν στο τμήμα Έρευνας

```
select ΥΠΑΛΛΗΛΟΣ.Επώνυμο, ΥΠΑΛΛΗΛΟΣ.Όνομα  
from ΥΠΑΛΛΗΛΟΣ, ΤΜΗΜΑ  
where ΥΠΑΛΛΗΛΟΣ.Κωδ_Τμήμ=ΤΜΗΜΑ.Κωδικός AND  
ΤΜΗΜΑ.Κωδικός="Έρευνας"
```



# SQL – Select (12)

- Η SQL επιτρέπει τη μετονομασία σχέσεων και χαρακτηριστικών με τη χρήση της πρότασης **as**
- Παράδειγμα: Βρείτε όλους τους υπαλλήλους που δουλεύουν στο τμήμα Έρευνας

```
select Y.Επώνυμο, Y.Όνομα  
from ΥΠΑΛΛΗΛΟΣ as Y, ΤΜΗΜΑ as T  
where Y.Κωδ_Τμήμ=T.Κωδικός AND  
T.Κωδικός="Έρευνας"
```



# SQL – Select (13)

- Παράδειγμα: Βρείτε τα ονόματα των διευθυντών κάθε τμήματος της εταιρίας

```
select Y.Επώνυμο, Y.Όνομα  
from ΥΠΑΛΛΗΛΟΣ as Y, ΤΜΗΜΑ as T  
where Y.Κωδικός=T.Κωδ_Διευθ
```

- Παράδειγμα: Βρείτε τα ονόματα των επιβλέπων της εταιρίας (αναδρομή)

```
select Y.Επώνυμο, Y.Όνομα  
from ΥΠΑΛΛΗΛΟΣ as Y, ΥΠΑΛΛΗΛΟΣ as E  
where Y.Κωδικός=E.Κωδ_Επιβλ
```



# SQL – Select (14)

- Παράδειγμα: Βρείτε τα ονόματα των υπαλλήλων της εταιρίας που δουλεύουν στο έργο «Μηχανογράφηση»

```
select Υ.Επώνυμο, Υ.Όνομα  
from ΥΠΑΛΛΗΛΟΣ as Υ, ΔΟΥΛΕΥΕΙ as Δ, ΕΡΓΟ as Ε  
where Υ.Κωδικός=Δ.Κωδ_Υπαλ AND  
Δ.Κωδ_Εργου=Ε.Κωδικός AND  
Ε.Όνομα="Μηχανογράφηση"
```

- Παράδειγμα: Βρείτε τα ονόματα των υπαλλήλων που μένουν στην Καβάλα και στη Δράμα

```
select Επώνυμο, Όνομα  
from ΥΠΑΛΛΗΛΟΣ  
where Πόλη="Καβάλα" OR Πόλη="Δράμα"
```



# SQL – Select (15)

- Παράδειγμα: Βρείτε τις τοποθεσίες στις οποίες η εταιρία έχει τμήματα

```
select distinct Τοποθεσία  
from ΤΜΗΜΑ_ΤΟΠ
```

- Παράδειγμα: Βρείτε τα ονόματα των Καβαλιωτών υπαλλήλων που δουλεύουν σε έργα που εκτελούνται στη Δράμα

```
select Υ.Επώνυμο, Υ.Όνομα  
from ΥΠΑΛΛΗΛΟΣ as Υ, ΔΟΥΛΕΥΕΙ as Δ, ΕΡΓΟ as Ε  
where Υ.Κωδικός=Δ.Κωδ_Υπαλ AND  
Δ.Κωδ_Εργου=Ε.Κωδικός AND  
Ε.Τοποθεσία="Δράμα" AND  
Υ.Πόλη="Καβάλα"
```





# SQL – Select (16)

- Παράδειγμα: Εμφανίστε τα ονόματα και τους μισθούς των υπαλλήλων που έχουν εργαστεί σε δύο διαφορετικά έργα περισσότερο από 25 ώρες

```
select  Υ.Επώνυμο, Υ.Όνομα, Υ.Μισθός
from    ΥΠΑΛΛΗΛΟΣ as Υ, ΔΟΥΛΕΥΕΙ as Δ1, ΔΟΥΛΕΥΕΙ as Δ2
where   Υ.Κωδικός=Δ1.Κωδ_Υπαλ AND
        Υ.Κωδικός=Δ2.Κωδ_Υπαλ AND
        Δ1.Ώρες>25 AND
        Δ2.Ώρες>25 AND
        Δ1.Κωδ_Έργου<>Δ2.Κωδ_Έργου
```



# SQL – Γενική Μορφή

- Γενική μορφή της select

```
select    A1, A2, ..., An, f(A)
from      r1, r2, ..., rm
where     P
group by  A1, A2, ..., An
having    Q
order by  Ax, ..., Ay
```

- $f(A)$  είναι μία συνάρτηση συνάθροισης (aggregate function)
- $Q$  είναι μία συνθήκη που αφορά το **group by**



# SQL – Ταξινόμηση (1)

- Διάταξη αποτελεσμάτων (order by)

```
select    A1, A2, ..., An, f(A)
from      r1, r2, ..., rm
where     P
order by  Ax, ..., Ay asc ή desc
```

- Παράδειγμα:

```
select Επώνυμο, Όνομα, Διεύθυνση
from ΥΠΑΛΛΗΛΟΣ
order by Επώνυμο
```



## SQL – Ταξινόμηση (2)

- Παράδειγμα: Εμφανίστε με αλφαβητική σειρά τους υπάλληλους που μένουν στην Καβάλα

```
select  Επώνυμο, Όνομα, Διεύθυνση
from    ΥΠΑΛΛΗΛΟΣ
where   Πόλη="Καβάλα"
order by Επώνυμο, Όνομα
```

- Παράδειγμα: Εμφανίστε τα ονόματα των υπαλλήλων που δουλεύουν στο τμήμα Έρευνας με φθίνουσα σειρά ως προς τον μισθό τους. Αυτοί που έχουν ίδιο μισθό, να εμφανίζονται αλφαβητικά

```
select  Υ.Επώνυμο, Υ.Όνομα, Υ.Μισθός
from    ΥΠΑΛΛΗΛΟΣ as Υ, ΤΜΗΜΑ as Τ
where   Υ.Κωδ_Τμήμ=Τ.Κωδικός AND
        Τ.Όνομα="Έρευνας"
order by Υ.Μισθός desc, Υ.Επώνυμο asc, Υ.Όνομα asc
```



# SQL – Πράξεις Συνόλων (1)

- Οι πράξεις συνόλων **union**, **intersect**, **except** μπορούν να εφαρμοστούν σε σχέσεις
- Καθεμία από τις πράξεις συνόλων αφαιρεί αυτόματα τις διπλοεγγραφές. Για να διατηρηθούν οι πολλαπλές εμφανίσεις των εγγραφών χρησιμοποιούνται οι εκδοχές **union all**, **intersect all**, **except all**



# SQL – Πράξεις Συνόλων (2)

- Παράδειγμα: Βρείτε τις τοποθεσίες που η εταιρία έχει είτε κάποιο τμήμα ή κάποιο έργο σε εξέλιξη

```
(select  Τοποθεσία  
from    ΤΜΗΜΑ_ΤΟΠ)  
  
union  
  
(select  Τοποθεσία  
from    ΕΡΓΟ)
```

- Παράδειγμα: Βρείτε τις τοποθεσίες που η εταιρία έχει και κάποιο τμήμα και ταυτόχρονα κάποιο έργο σε εξέλιξη

```
(select  Τοποθεσία  
from    ΤΜΗΜΑ_ΤΟΠ)  
  
intersect  
  
(select  Τοποθεσία  
from    ΕΡΓΟ)
```



# SQL – Πράξεις Συνόλων (3)

- Παράδειγμα: Βρείτε τις τοποθεσίες που η εταιρία έχει κάποιο τμήμα αλλά κανένα έργο σε εξέλιξη

```
(select  Τοποθεσία  
from    ΤΜΗΜΑ_ΤΟΠ)  
except  
(select  Τοποθεσία  
from    ΕΡΓΟ)
```



# SQL – Πράξεις Συνόλων (4)

- Παράδειγμα: Βρείτε τα τμήματα της εταιρίας στα οποία εργάζεται κάποιος υπάλληλος που ονομάζεται “Παπαδόπουλος” (εργάζεται είτε ως υπάλληλος, είτε ως διευθυντής)

```
(select  T.Όνομα
from    ΥΠΑΛΛΗΛΟΣ as Y, ΤΜΗΜΑ as T
where   Y.Κωδ_Τμήμ=T.Κωδικός AND
        Y.Επώνυμο=“Παπαδόπουλος”)

union

(select  T.Όνομα
from    ΥΠΑΛΛΗΛΟΣ as Y, ΤΜΗΜΑ as T
where   Y.Κωδικός=T.Κωδ_Διευθ AND
        Y.Επώνυμο=“Παπαδόπουλος”)
```





# SQL – Φωλιασμένα Ερωτήματα (1)

- Η SQL παρέχει ένα μηχανισμό για εμφωλευμένα υποερωτήματα
- Ένα υποερώτημα είναι μία **select** έκφραση, η οποία είναι εμφωλευμένη σε μία άλλη ερώτηση
- Χρησιμοποιείται ο τελεστής **in**
  - Παράδειγμα: Βρείτε τους υπάλληλους που εργάζονται στο τμήμα Έρευνας

```
select  Επώνυμο, Όνομα
from    ΥΠΑΛΛΗΛΟΣ
where   Κωδ_Τμήμ in
        (select Κωδικός
         from  ΤΜΗΜΑ
         where Όνομα="Έρευνας")
```

# SQL – Φωλιασμένα Ερωτήματα (2)

- Παράδειγμα: Βρείτε τους υπάλληλους που έχουν προστατευόμενο μέλος του ίδιου φύλου με τον υπάλληλο

```
select  Υ.Επώνυμο, Υ.Όνομα
from    ΥΠΑΛΛΗΛΟΣ as Υ
where   Υ.Κωδικός in
        (select Π.Κωδ_Υπαλ
         from  ΠΡΟΣΤΑΤΕΥΟΜΕΝΟΣ as Π
         where Υ.Κωδικός=Π.Κωδ_Υπαλ AND
                Υ.Φύλο=Π.Φύλο)
```

```
select  Υ.Επώνυμο, Υ.Όνομα
from    ΥΠΑΛΛΗΛΟΣ as Υ, ΠΡΟΣΤΑΤΕΥΟΜΕΝΟΣ as Π
where   Υ.Κωδικός=Π.Κωδ_Υπαλ AND
        Υ.Φύλο=Π.Φύλο)
```



# SQL – Φωλιασμένα Ερωτήματα (3)

- Παράδειγμα: Βρείτε τους υπάλληλους που έχουν προστατευόμενο μέλος του ίδιου φύλου με τον υπάλληλο

```
select distinct Όνομα
from      ΤΜΗΜΑ
where     Κωδικός in
          (select  T.Κωδικός
           from     ΥΠΑΛΛΗΛΟΣ as Y, ΤΜΗΜΑ as T
           where    Y.Κωδ_Τμήμ=T.Κωδικός AND
                   Y.Επώνυμο="Παπαδόπουλος")
or Κωδικός in
          (select  T.Κωδικός
           from     ΥΠΑΛΛΗΛΟΣ as Y, ΤΜΗΜΑ as T
           where    Y.Κωδικός=T.Κωδ_Διευθ AND
                   Y.Επώνυμο="Παπαδόπουλος")
```



# SQL – Πράξεις με Συμβολοσειρές (1)

- Η SQL παρέχει έναν τελεστή ταιριάσματος συμβολοσειρών (**like**) για συγκρίσεις με χρήση δύο ειδικών χαρακτήρων “%” και “\_”
  - “%” Ο χαρακτήρας % ταιριάζει οποιαδήποτε συμβολοσειρά
  - “\_” Ο χαρακτήρας \_ ταιριάζει οποιονδήποτε χαρακτήρα
- Παράδειγμα: Βρείτε τους υπάλληλους που η διεύθυνση τους περιλαμβάνει το “Άγιος”

```
select  Επώνυμο, Όνομα  
from    ΥΠΑΛΛΗΛΟΣ  
where   Διεύθυνση like “%Άγιος%”
```



# SQL – Πράξεις με Συμβολοσειρές (2)

- Παράδειγμα: Βρείτε τους υπάλληλους που το επώνυμο τους τελειώνει σε “ίδης”

```
select  Επώνυμο, Όνομα  
from    ΥΠΑΛΛΗΛΟΣ  
where   Επώνυμο like “%ίδης”
```

- Παράδειγμα: Βρείτε τους υπάλληλους που το επώνυμο τους ξεκινάει από “Π”, το τρίτο γράμμα είναι “α” και τελειώνει σε “άκης”

```
select  Επώνυμο, Όνομα  
from    ΥΠΑΛΛΗΛΟΣ  
where   Επώνυμο like “Π_α%άκης”
```



# SQL – Συναρτήσεις (1)

- Ομαδοποίηση αποτελεσμάτων (group by)

```
select    A1, A2, ..., An, f(A)
from      r1, r2, ..., rm
where     P
group by  A1, A2, ..., An
having    Q
```

- Ομαδοποιεί τα αποτελέσματα εφαρμόζοντας τη συνάρτηση  $f(A)$  σε κάθε ομάδα χωριστά. Η πρόταση **having** καθορίζει συνθήκη αναζήτησης για τις ομάδες



# SQL – Συναρτήσεις (2)

## ➤ Συναρτήσεις συνάθροισης (aggregate functions)

- SUM(A)      Άθροισμα τιμών του A
- AVG(A)      Μέσος όρος τιμών του A
- MAX(A)      Η μέγιστη τιμή του A
- MIN(A)      Η ελάχιστη τιμή του A
- COUNT(A)    Το πλήθος των τιμών του A



# SQL – Συναρτήσεις (3)

- Παράδειγμα: Βρείτε τον μέγιστο και τον ελάχιστο μισθό καθώς και τον μέσο μισθό όλων των υπαλλήλων

```
select    max(Μισθός), min(Μισθός), avg(Μισθός)
from      ΥΠΑΛΛΗΛΟΣ
```

- Παράδειγμα: Βρείτε τον αριθμό των υπαλλήλων

```
select    count(*)
from      ΥΠΑΛΛΗΛΟΣ
```

- Παράδειγμα: Βρείτε τον μέσο μισθό των υπαλλήλων που εργάζονται στο τμήμα με κωδικό 4

```
select    avg(Μισθός)
from      ΥΠΑΛΛΗΛΟΣ
where     Κωδ_Τμήμ=4
```



# SQL – Συναρτήσεις (4)

- Παράδειγμα: Βρείτε το πλήθος των υπαλλήλων που δουλεύουν στο τμήμα Έρευνας

```
select    count(*)  
from      ΥΠΑΛΛΗΛΟΣ as Y, ΤΜΗΜΑ as T  
where     Y.Κωδ_Τμήμ=T.Κωδικός AND  
          T.Όνομα='Έρευνας'
```

- Παράδειγμα: Βρείτε τους υπαλλήλους που έχουν περισσότερα από δύο προστατευόμενα μέλη

```
select    Επώνυμο, Όνομα  
from      ΥΠΑΛΛΗΛΟΣ as Y  
where     (select count(*)  
          from ΠΡΟΣΤΑΤΕΥΟΜΕΝΟΣ as Π  
          where Y.Κωδικός=Π.Κωδ_Υπαλ)>=2
```



# SQL – Συναρτήσεις (5)

- Παράδειγμα: Βρείτε το ονοματεπώνυμο του υπαλλήλου που παίρνει τον μεγαλύτερο μισθό

```
select  Επώνυμο, Όνομα  
from    ΥΠΑΛΛΗΛΟΣ  
where   Μισθός=  
        (select max(Μισθός)  
         from ΥΠΑΛΛΗΛΟΣ)
```

- Παράδειγμα: Βρείτε τους υπαλλήλους που παίρνουν περισσότερα από τον μέσο μισθό των όλων υπαλλήλων

```
select  Επώνυμο, Όνομα  
from    ΥΠΑΛΛΗΛΟΣ  
where   Μισθός>=  
        (select avg(Μισθός)  
         from ΥΠΑΛΛΗΛΟΣ)
```



# SQL – Συναρτήσεις (6)

- Παράδειγμα: Βρείτε για κάθε τμήμα της εταιρίας, τον κωδικό του, το πλήθος των υπαλλήλων που εργάζονται σε αυτό και το μέσο όρο των μισθών τους

```
select    Κωδ_Τμήμ, count(*), avg(Μισθός)
from      ΥΠΑΛΛΗΛΟΣ
group by  Κωδ_Τμήμ
```

- Παράδειγμα: Βρείτε για κάθε τμήμα, την ελάχιστη ηλικία των υπαλλήλων που έχουν μισθό μεγαλύτερο από 1000€

```
select    Κωδ_Τμήμ, max(ΗμερΓεν)
from      ΥΠΑΛΛΗΛΟΣ
where     Μισθός>1000
group by  Κωδ_Τμήμ
```



# SQL – Συναρτήσεις (7)

- Παράδειγμα: Βρείτε για κάθε τμήμα της εταιρίας, το όνομα του και το πλήθος των υπαλλήλων που εργάζονται σε αυτό

```
select    T.Όνομα, count(*)  
from      ΥΠΑΛΛΗΛΟΣ as Y, ΤΜΗΜΑ as T  
where     Y.Κωδ_Τμήμ=T.Κωδικός  
group by  T.Όνομα
```

- Παράδειγμα: Βρείτε πόσες ώρες δουλεύει συνολικά κάθε υπάλληλος

```
select    Y.Επώνυμο, sum(Ωρες)  
from      ΥΠΑΛΛΗΛΟΣ as Y, ΔΟΥΛΕΥΕΙ as Δ  
where     Y.Κωδικός=Δ.Κωδ_Υπαλ  
group by  Y.Επώνυμο
```



# SQL – Συναρτήσεις (8)

- Παράδειγμα: Βρείτε τον μέσο μισθό των υπαλλήλων που γεννήθηκαν μετά το 1950 για κάθε τμήμα με περισσότερους από 10 υπαλλήλους

```
select    Κωδ_Τμήμ, avg(Μισθός)
from      ΥΠΑΛΛΗΛΟΣ
where     ΗμερΓεν>"01/01/1950"
group by  Κωδ_Τμήμ
having    count(*)>10
```

- Παράδειγμα: Βρείτε το πλήθος των υπαλλήλων ανά έργο, αλλά μόνο για τα έργα που εργάζονται περισσότεροι από 2 υπάλληλοι

```
select    Ε.Όνομα, count(*)
from      ΕΡΓΟ as Ε, ΔΟΥΛΕΥΕΙ as Δ
where     Ε.Κωδικός=Δ.Κωδ_Έργου
group by  Ε.Όνομα
having    count(*)>2
```



# SQL – Συναρτήσεις (9)

- Παράδειγμα: Βρείτε το πλήθος των υπαλλήλων που εργάζονται σε κάθε τμήμα και έχουν μισθό μεγαλύτερο από 1000€, αλλά μόνο για τα τμήματα στα οποία εργάζονται παραπάνω από 5 υπάλληλοι

```
select      T.Όνομα, count(*)
from        ΥΠΑΛΛΗΛΟΣ as Y, ΤΜΗΜΑ as T
where       Y.Κωδ_Τμήμ=T.Κωδικός AND
            Y.Μισθός>1000 AND
            T.Κωδικός in
            (select  Κωδ_Τμήμ
             from      ΥΠΑΛΛΗΛΟΣ
             group by  Κωδ_Τμήμ
             having    count(*)>5)
group by    T.Όνομα
```





Ανάπτυξη παντού. Ανάπτυξη για όλους.

ΥΠΟΥΡΓΕΙΟ ΕΘΝΙΚΗΣ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΕΑΕΚ  
ΕΥΡΩΠΑΪΚΗ ΕΝΩΣΗ  
ΣΥΓΧΡΗΜΑΤΟΔΟΤΗΣΗ  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ  
ΕΥΡΩΠΑΪΚΟ ΤΑΜΕΙΟ ΠΕΡΙΦΕΡΕΙΑΚΗΣ ΑΝΑΠΤΥΞΗΣ



Η ΠΑΙΔΕΙΑ ΣΤΗΝ ΚΟΡΥΦΗ  
Επιχειρησιακό Πρόγραμμα  
Εκπαίδευσης και Αρχικής  
Επαγγελματικής Κατάρτισης

Η ψηφιοποίηση του εκπαιδευτικού υλικού έγινε στο πλαίσιο υλοποίησης της πράξης με τίτλο «ΕΝΙΣΧΥΣΗ ΣΠΟΥΔΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ στο ΤΕΙ ΚΑΒΑΛΑΣ», του Μέτρου 2.2 «Αναμόρφωση Προγραμμάτων Σπουδών - Διεύρυνση Τριτοβάθμιας Εκπαίδευσης» του ΕΠΕΑΕΚ II, που συγχρηματοδοτείται από το Ευρωπαϊκό Κοινωνικό Ταμείο (Ε.Κ.Τ.) κατά 80% και Εθνικούς πόρους κατά 20%



Διαχείριση Πληροφοριών  
Τεχνολογικό Εκπαιδευτικό Ίδρυμα Καβάλας



Ανάπτυξη παντού. Ανάπτυξη για όλους.