

Οργάνωση Βάσεων Βιοϊατρικών Δεδομένων Εξόρυξη Γνώσης Βιοϊατρικών Δεδομένων

Σεμινάριο 5: SQL

**Ευάγγελος Καρκαλέτσης,
Αναστασία Κριθαρά, Γεώργιος Πετάσης**

Εργαστήριο Τεχνολογίας Γνώσεων & Λογισμικού,

Ινστιτούτο Πληροφορικής & Τηλεπικοινωνιών, Ε.Κ.Ε.Φ.Ε. “Δημόκριτος”

Τηλ.: 210-6503197, Fax: 210-6532175, {vangelis, akrihara, petasis}@iit.demokritos.gr

Ακαδημαϊκό Έτος: 2016 – 2017

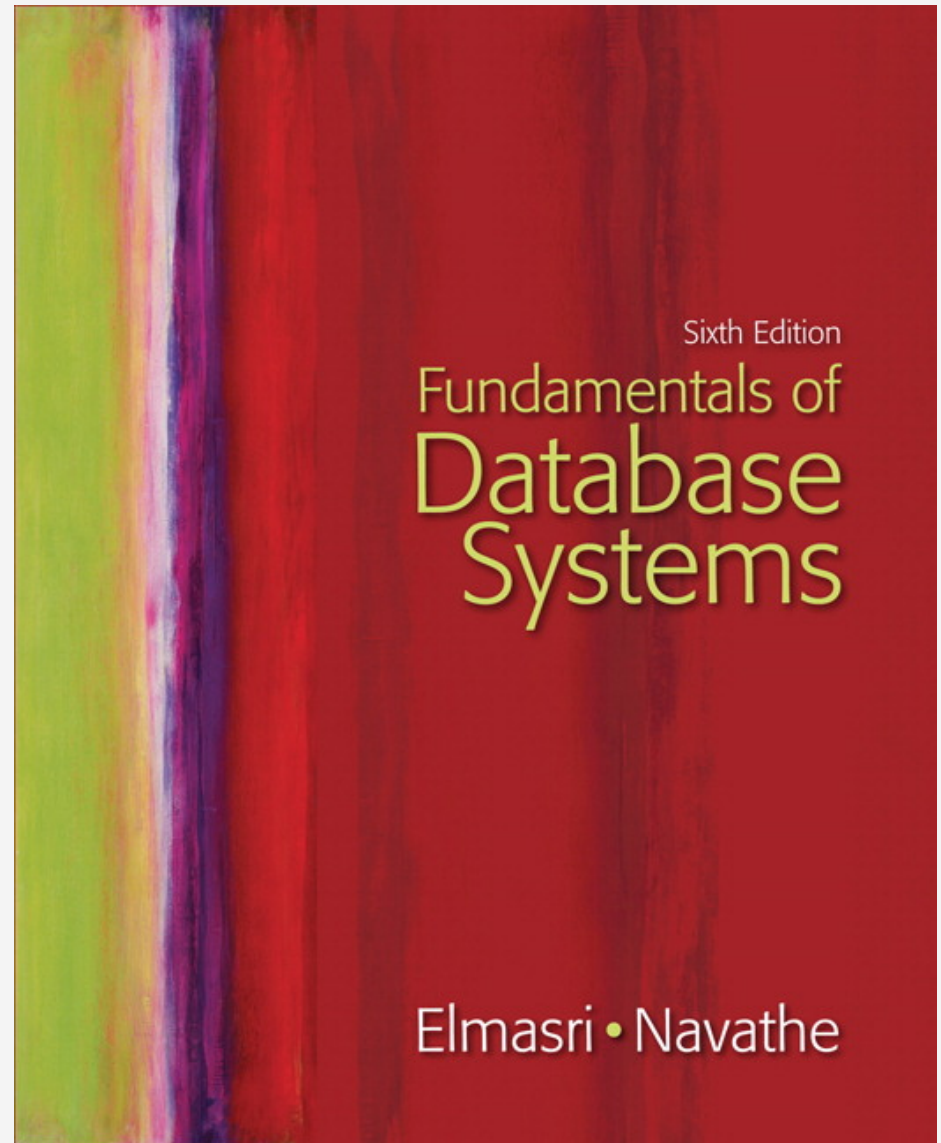
Οι διαφάνειες αυτού του σεμιναρίου βασίζονται
σε διαφάνειες των:

Δ. Γουνόπουλος, Ι. Ιωαννίδης

<http://eclass.uoa.gr/modules/document/document.php?course=D463&openDir=/4c2b32a5i4iq>

ΚΕΦΑΛΑΙΟ 4

Βασική SQL



Copyright © 2011 Ramez Elmasri and Shamkant B. Navathe, Ελληνική Έκδοση Δίαυλος

SQL (Structured Query Language)

- Υψηλού επιπέδου δηλωτική διεπαφή γλώσσας
- Βασίζεται περισσότερο στον σχεσιακό λογισμό πλειάδων
- Φιλική διεπαφή στο χρήστη
- Πλήρης γλώσσα βάσεων δεδομένων:
 - Εντολές για ορισμό δεδομένων, ερωτήσεις και ενημερώσεις

Ορισμός Δεδομένων, Περιορισμοί, και Αλλαγές Σχήματος

- Οι CREATE, DROP, και ALTER χρησιμοποιούνται για αλλαγές της περιγραφής των πινάκων (σχέσεων) μιας βάσης δεδομένων

CREATE TABLE

- Ορίζει μια νέα βασική σχέση δίνοντας της ένα όνομα και ορίζοντας κάθε γνώρισμα και τον τύπο δεδομένων του γνωρίσματος (INTEGER, FLOAT, DECIMAL(i,j), CHAR(n), VARCHAR(n))
- Σε ένα γνώρισμα μπορεί να ορισθεί ένας περιορισμός NOT NULL

```
CREATE TABLE ΤΜΗΜΑ (  
    Τ_ΟΝΟΜΑ          VARCHAR(10)  NOT NULL,  
    ΚΩΔ_ΤΜΗΜ        INTEGER      NOT NULL,  
    ΔΙΕΥΘΥΝΤΗΣ      CHAR(9),  
    ΗΜΕΡ_ΕΝΑΡΞΗΣ    CHAR(9)  
);
```

CREATE TABLE

- Στην SQL2, μπορούμε να χρησιμοποιήσουμε την εντολή CREATE TABLE για τον προσδιορισμό των γνωρισμάτων του πρωτεύοντος κλειδιού, των δευτερευόντων κλειδιών, και των αναφορικών περιορισμών ακεραιότητας (ξένα κλειδιά).
- Τα γνωρίσματα κλειδιού προσδιορίζονται με τις προτάσεις PRIMARY KEY and UNIQUE.

```
CREATE TABLE ΤΜΗΜΑ (  
    Τ ΟΝΟΜΑ                VARCHAR(10)          NOT NULL,  
    ΚΩΔ ΤΜΗΜ                INTEGER                NOT NULL,  
    ΔΙΕΥΘΥΝΤΗΣ             CHAR(9),  
    ΗΜΕΡ ΕΝΑΡΞΗΣ           CHAR(9),  
    PRIMARY KEY (ΚΩΔ ΤΜΗΜ),  
    UNIQUE (Τ ΟΝΟΜΑ),  
    FOREIGN KEY(ΔΙΕΥΘΥΝΤΗΣ) REFERENCES ΕΡΓΑΖΟΜΕΝΟΣ  
);
```

DROP TABLE

- Χρησιμοποιείται για τη διαγραφή μιας σχέσης (βασικός πίνακας) και του ορισμού του
- Η σχέση δεν μπορεί να χρησιμοποιηθεί περαιτέρω σε ερωτήσεις, ενημερώσεις, ή οποιαδήποτε άλλη εντολή αφού δεν υπάρχει ο ορισμός της
- Παράδειγμα:

```
DROP TABLE ΕΞΑΡΤΩΜΕΝΟΣ;
```


ALTER TABLE

- Χρησιμοποιείται για την προσθήκη ενός γνώρισματος σε μια από τις βασικές σχέσεις
- Το νέο γνώρισμα θα έχει τιμές NULL σε όλες τις πλειάδες της σχέσης μετά την εκτέλεση της εντολής· επομένως, δεν επιτρέπεται ο περιορισμός NOT NULL για το γνώρισμα αυτό
- Παράδειγμα:

```
ALTER TABLE ΕΡΓΑΖΟΜΕΝΟΣ ADD ΕΡΓΑΣΙΑ VARCHAR(12);
```
- Οι χρήστες της βάσης δεδομένων πρέπει πάλι να εισάγουν μια τιμή για το νέο γνώρισμα ΕΡΓΑΣΙΑ για κάθε πλειάδα ΕΡΓΑΖΟΜΕΝΟΣ.
 - Αυτό μπορεί να επιτευχθεί με την εντολή UPDATE.

Πρόσθετα χαρακτηριστικά της SQL2 και της SQL-99

- Create schema
- Επιλογές Αναφορικής Ακεραιότητας

CREATE SCHEMA

- Ορίζει το σχήμα μιας νέας βάσης δεδομένων δίνοντας της ένα όνομα.

```
CREATE SCHEMA ΕΤΑΙΡΙΑ AUTHORIZATION JSMITH;
```

ΕΠΙΛΟΓΕΣ ΑΝΑΦΟΡΙΚΗΣ ΑΚΕΡΑΙΟΤΗΤΑΣ

- Στου αναφορικούς περιορισμούς ακεραιότητας (ξένα κλειδιά) μπορούμε να προσδιορίσουμε τις **CASCADE**, **SET NULL** ή **SET DEFAULT**

```
CREATE TABLE ΤΜΗΜΑ (  
    Τ_ΟΝΟΜΑ                VARCHAR(10) NOT NULL,  
    ΚΩΔ_ΤΜΗΜ              INTEGER      NOT NULL,  
    ΔΙΕΥΘΥΝΤΗΣ            CHAR(9) ,  
    ΗΜΕΡ_ΕΝΑΡΞΗΣ          CHAR(9) ,  
    PRIMARY KEY (ΚΩΔ_ΤΜΗΜ) ,  
    UNIQUE (Τ_ΟΝΟΜΑ) ,  
    FOREIGN KEY (ΔΙΕΥΘΥΝΤΗΣ) REFERENCES ΕΡΓΑΖΟΜΕΝΟΣ  
    ON DELETE SET DEFAULT ON UPDATE CASCADE);
```

ΕΠΙΛΟΓΕΣ ΑΝΑΦΟΡΙΚΗΣ ΑΚΕΡΑΙΟΤΗΤΑΣ (συνέχεια)

```
CREATE TABLE ΕΡΓ (  
    Ε_ΟΝΟΜΑ VARCHAR(30) NOT NULL,  
    Ε_ΑΡΤΑΥΤ CHAR(9),  
    ΗΜ_ΓΕΝ DATE,  
    ΑΡΙΘ_Τ INTEGER DEFAULT 1,  
    ΠΡΟΙΣΤΑΜΕΝΟΣ CHAR(9),  
    PRIMARY KEY (Ε_ΑΡΤΑΥΤ),  
    FOREIGN KEY (ΑΡΙΘ_Τ) REFERENCES ΤΜΗΜΑ  
    ON DELETE SET DEFAULT ON UPDATE CASCADE,  
    FOREIGN KEY (ΠΡΟΙΣΤΑΜΕΝΟΣ) REFERENCES ΕΡΓ  
    ON DELETE SET NULL ON UPDATE CASCADE);
```

Πρόσθετοι τύποι δεδομένων των SQL2 και SQL-99

- Έχει τους τύπους δεδομένων DATE, TIME, and TIMESTAMP
- **DATE:**
 - Σχηματίζεται από τα year-month-day σε μορφή yyyy-mm-dd
- **TIME:**
 - Σχηματίζεται από:minute:second σε μορφή hh:mm:ss
- **TIME(i):**
 - Σχηματίζεται από hour:minute:second με επιπλέον ψηφία που ορίζουν δέκατα δευτερολέπτου
 - η μορφή είναι hh:mm:ss:ii...i

Πρόσθετοι τύποι δεδομένων των SQL2 και SQL-99(συνεχ.)

- **TIMESTAMP:**
- Έχει και τις δύο, DATE και TIME συνιστώσες
- **INTERVAL:**
 - Προσδιορίζει μια σχετική τιμή παρά σαν απόλυτη τιμή.
 - Μπορεί να είναι διαστήματα DAY/TIME ή YEAR/MONTH
 - Μπορεί να είναι θετική ή αρνητική όταν προστεθεί ή αφαιρεθεί από μια απόλυτη τιμή, το αποτέλεσμα είναι μια απόλυτη τιμή

Ερωτήσεις ανάκτησης στην SQL

- Η SQL έχει μια βασική εντολή για ανάκτηση πληροφοριών από μια βάση δεδομένων· την εντολή **SELECT**
 - Δεν είναι ίδια με την πράξη SELECT (επιλογή) της σχεσιακής άλγεβρας
- Σημαντική διαφορά μεταξύ της SQL και του τυπικού σχεσιακού μοντέλου:
 - Η SQL επιτρέπει σε ένα πίνακα (σχέση) να έχει δύο ή περισσότερες ταυτόσημες πλειάδες (ίδια τιμή σε όλα τους τα γνωρίσματα)
 - Επομένως, μια SQL σχέση (πίνακας) είναι ένα **πολυ-σύνολο** (ονομάζεται και **bag**) από πλειάδες· δεν είναι ένα σύνολο από πλειάδες
- Οι σχέσεις της SQL μπορούν να περιορισθούν σε σύνολα με προσδιορισμό γνωρισμάτων PRIMARY KEY ή UNIQUE, ή χρησιμοποιώντας την επιλογή DISTINCT σε μια ερώτηση

Ερωτήσεις Ανάκτησης στην SQL (συν.)

- Ένα **bag** ή **πολυ-σύνολο** μοιάζει με σύνολο, αλλά ένα στοιχείο μπορεί να εμφανισθεί περισσότερες από μια φορές.
 - Παράδειγμα το: $\{A, B, C, A\}$ είναι bag. Το $\{A, B, C\}$ είναι επίσης bag που είναι και σύνολο.
 - Τα bags μοιάζουν με λίστες, αλλά στο bag δεν ενδιαφέρει η σειρά.
- Παράδειγμα:
 - $\{A, B, A\} = \{B, A, A\}$ είναι bags
 - Ωστόσο, τα $[A, B, A]$ και $[B, A, A]$ δεν ίσα σαν λίστες

Ερωτήσεις Ανάκτησης στην SQL (συν.)

- Η βασική μορφή της SQL εντολής SELECT ονομάζεται μια *απεικόνιση* ή ένα SELECT-FROM-WHERE μπλοκ

SELECT <λίστα γνωρισμάτων>
FROM <λίστα πινάκων>
WHERE <συνθήκη>

- < *λίστα γνωρισμάτων* > είναι μια λίστα ονομάτων γνωρισμάτων που η τιμή τους πρέπει να ανακτηθεί από την ερώτηση
- < *λίστα πινάκων* > είναι μια λίστα από ονόματα σχέσεων που απαιτούνται για επεξεργασία της ερώτησης
- < *συνθήκη* > είναι μια έκφραση συνθήκης (Boolean) που προσδιορίζει τις πλειάδες που θα ανακτηθούν από την ερώτηση.

Σχισιακό σχήμα βάσης δεδομένων

ΕΡΓΑΖΟΜΕΝΟΣ

ΟΝΟΜΑ	ΑΡΧ_ΠΑΤ	ΕΠΙΘΕΤΟ	<u>ΑΡ_ΤΑΥΤ</u>	ΗΜ_ΓΕΝ	ΔΙΕΥΘΥΝΣΗ	ΦΥΛΟ	ΜΙΣΘΟΣ	ΠΡΟΪΣΤΑΜΕΝΟΣ	ΑΡΙΘ_Τ
-------	---------	---------	----------------	--------	-----------	------	--------	--------------	--------

+ ΤΜΗΜΑ

T_ΟΝΟΜΑ	<u>ΚΩΔ_ΤΜΗΜ</u>	ΔΙΕΥΘΥΝΤΗΣ	ΗΜΕΡ_ΕΝΑΡΞΗΣ
---------	-----------------	------------	--------------

ΤΟΠΟΘ_ΤΜΗΜΑ

<u>ΚΩΔ_ΤΜΗΜ</u>	<u>Τ_ΤΟΠΟΘΕΣΙΑ</u>
-----------------	--------------------

ΕΡΓΟ

E_ΟΝΟΜΑ	<u>ΚΩΔ_ΕΡΓΟΥ</u>	ΤΟΠ_ΕΡΓΟΥ	Κ_ΤΜΗΜΑ
---------	------------------	-----------	---------

ΑΠΑΣΧΟΛΗΣΗ

<u>E_ΑΡΤΑΥΤ</u>	<u>Κ_ΕΡΓΟ</u>	ΩΡΕΣ
-----------------	---------------	------

ΕΞΑΡΤΩΜΕΝΟΣ

<u>E_ΑΡΤΑΥΤ</u>	<u>ΟΝΟΜΑ_ΕΞΑΡΤΩΜΕΝΟΥ</u>	ΦΥΛΟ	ΗΜ_ΓΕΝ	ΣΧΕΣΗ
-----------------	--------------------------	------	--------	-------

Βάση με δεδομένα

ΕΡΓΑΖΟΜΕΝΟΣ	ΟΝΟΜΑ	ΑΡΧ_ΠΑΤ	ΕΠΙΘΕΤΟ	ΑΡ_ΤΑΥΤ	ΗΜ_ΓΕΝ	ΔΙΕΥΘΥΝΣΗ	ΦΥΛΟ	ΜΙΣΘΟΣ	ΠΡΩΤΑ ΜΕΝΟΣ	ΑΡΙΘ_Τ
	John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
	Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888667777	5
	Allela	J	Zelaya	999887777	1968-07-19	3321 Castle, Spring, TX	F	25000	987654321	4
	Jennifer	S	Wallace	987654321	1941-06-20	975 Fire Oak, Jumble, TX	F	43000	888665555	4
	Ramesh	K	Narayan	666884444	1962-09-15	3321 Castle, Spring, TX	M	38000	333445555	5
	Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
	Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
	James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	null	1

ΤΜΗΜΑ	T_ΟΝΟΜΑ	ΚΩΔ_ΤΜΗΜ	ΔΙΕΥΘΥΝΤΗΣ	ΗΜΕΡ_ΕΝΑΡΞΗΣ
	Research	5	333445555	1988-05-22
	Administration	4	987654321	1995-01-01
	Headquarters	1	888665555	1981-06-19

ΤΟΠΟΘ_ΤΜΗΜΑ	ΚΩΔ_ΤΜΗΜΑ	T_ΤΟΠΟΘΕΣΙΑ
	1	Houston
	4	Stafford
	5	Bellaire
	5	Sugarland
	5	Houston

Βάση με δεδομένα (συν.)

ΑΠΑΣΧΟΛΗΣΗ	Ε_ΑΡΤΑΥΤ	Κ_ΕΡΓΟ	ΩΡΕΣ
	123456789	1	32.5
	123456789	2	7.5
	666884444	3	40.0
	453453453	1	20.0
	453453453	2	20.0
	333445555	2	10.0
	333445555	3	10.0
	333445555	10	10.0
	333445555	20	10.0
	999887777	30	30.0
	999887777	10	10.0
	999887777	10	35.5
	987987987	30	5.0
	987654321	30	20.0
	987654321	20	15.0
	888665555	20	null

ΕΡΓΟ	Ε_ΟΝΟΜΑ	ΚΩΔ_ΕΡΓΟΥ	ΤΟΠ_ΕΡΓΟΥ	Κ_ΤΜΗΜΑ
ProductX		1	Bellaire	5
ProductY		2	Sugarland	5
ProductZ		3	Houston	5
Computerization		10	Stafford	4
Reorganization		20	Houston	1
Newbenefits		30	Stafford	4

ΕΞΑΡΤΩΜΕΝΟΣ	Ε_ΑΡΤΑΥΤ	ΟΝΟΜΑ_ΕΞΑΡΤΩΜΕΝΟΥ	ΦΥΛΟ	ΗΜ_ΓΕΝ	ΣΧΕΣΗ
	333445555	Alice	F	1986-04-05	DAUGHTER
	333445555	Theodore	M	1983-10-25	SON
	333445555	Joy	F	1958-05-03	SPOUSE
	987654321	Abner	M	1942-02-28	SPOUSE
	123456789	Michael	M	1988-01-04	SON
	123456789	Alice	F	1988-12-30	DAUGHTER
	123456789	Elizabeth	F	1967-05-05	SPOUSE

Απλές SQL Ερωτήσεις

- Οι βασικές SQL ερωτήσεις αντιστοιχούν στη χρήση των παρακάτω πράξεων της σχεσιακής άλγεβρας:
 - **SELECT**
 - **PROJECT**
 - **JOIN**
- Όλα τα παραδείγματα που ακολουθούν χρησιμοποιούν τη βάση δεδομένων ΕΤΑΙΡΕΙΑ

Απλές SQL Ερωτήσεις(συν.)

- Παράδειγμα απλής ερώτησης σε μια σχέση
- Ερώτηση 0: Να βρεθεί η ημερομηνία γέννησης και η διεύθυνση του εργαζόμενου που το όνομα του είναι 'John B. Smith'
 - Q0: **SELECT** ΗΜ_ΓΕΝ, ΔΙΕΥΘΥΝΣΗ
FROM ΕΡΓΑΖΟΜΕΝΟΣ
WHERE ΟΝΟΜΑ='John' AND ΑΡΧ_ΠΑΤ='B' AND ΕΠΙΘΕΤΟ='Smith'
 - Μοιάζει με ένα ζεύγος SELECT-PROJECT πράξεων της σχεσιακής άλγεβρας:
 - Η πρόταση SELECT- προσδιορίζει τα γνωρίσματα της προβολής και η πρόταση WHERE- προσδιορίζει την συνθήκη επιλογής
 - Ωστόσο, το αποτέλεσμα της ερώτησης μπορεί να περιέχει διπλότυπες πλειάδες

Απλές SQL Ερωτήσεις (συν.)

- Ερώτηση 1: Να βρεθούν το όνομα και η διεύθυνση όλων των εργαζομένων στο τμήμα 'Research' .
 - **SELECT** ΟΝΟΜΑ, ΕΠΙΘΕΤΟ, ΔΙΕΥΘΥΝΣΗ
FROM ΕΡΓΑΖΟΜΕΝΟΣ, ΤΜΗΜΑ
WHERE T_ΟΝΟΜΑ='Research' **AND**
ΚΩΔ_ΤΜΗΜ=ΑΡΙΘ_Τ
 - Παρόμοια με μια ακολουθία SELECT-PROJECT-JOIN πράξεων της σχεσιακής άλγεβρας
 - (T_ΟΝΟΜΑ='Research') είναι μια συνθήκη επιλογής (αντιστοιχεί σε μια πράξη επιλογής στη σχεσιακή άλγεβρα)
 - (ΚΩΔ_ΤΜΗΜ=ΑΡΙΘ_Τ) είναι μια συνθήκη επιλογής (αντιστοιχεί σε μια πράξη JOIN της σχεσιακής άλγεβρας)

Απλές SQL Ερωτήσεις (συν.)

- Ερώτηση 2: Για κάθε έργο στο 'Stafford' να βρεθεί ο κωδικός του έργου, ο κωδικός του τμήματος που το ελέγχει και το επίθετο, η διεύθυνση και η ημερομηνία γέννησης του διευθυντή του τμήματος.
 - **SELECT** ΚΩΔ_ΕΡΓΟΥ, Κ_ΤΜΗΜΑ, ΕΠΙΘΕΤΟ, ΗΜ_ΓΕΝ, ΔΙΕΥΘΥΝΣΗ
FROM ΕΡΓΟ, ΤΜΗΜΑ, ΕΡΓΑΖΟΜΕΝΟΣ
WHERE Κ_ΤΜΗΜΑ=ΚΩΔ_ΤΜΗΜ **AND** ΔΙΕΥΘΥΝΤΗΣ=ΑΡ_ΤΑΥΤ **AND**
ΤΟΠ_ΕΡΓΟΥ='Stafford'
 - Στην ερώτηση, δύο συνθήκες συνένωσης
 - Η συνθήκη συνένωσης Κ_ΤΜΗΜΑ=ΚΩΔ_ΤΜΗΜ συσχετίζει ένα έργο με το τμήμα που το ελέγχει
 - Η συνθήκη συνένωσης ΔΙΕΥΘΥΝΤΗΣ=ΑΡ_ΤΑΥΤ συσχετίζει το τμήμα που ελέγχει με τον εργαζόμενο που το διευθύνει

Ψευδώνυμα, * και DISTINCT, Κενή πρόταση WHERE

- Στην SQL, μπορούμε να χρησιμοποιήσουμε το ίδιο όνομα για ένα (ή περισσότερα) γνωρίσματα αν τα γνωρίσματα είναι σε διαφορετικές σχέσεις
- Μια ερώτηση που αναφέρεται σε ένα ή περισσότερα γνωρίσματα με το ίδιο όνομα πρέπει να χαρακτηρίζει το όνομα του γνωρίσματος με το όνομα της σχέσης προτάσσοντας το όνομα της σχέσης στο όνομα του γνωρίσματος
- Παράδειγμα:
 - ΕΡΓΑΖΟΜΕΝΟΣ.ΕΠΙΘΕΤΟ, ΤΜΗΜΑ.Τ_ΟΝΟΜΑ

Ψευδώνυμα

- Μερικές ερωτήσεις χρειάζεται να αναφερθούν στην ίδια σχέση δύο φορές
 - Στην περίπτωση αυτή, δίνονται *ψευδώνυμα* στη σχέση αυτή
- Ερώτηση 8: Για κάθε εργαζόμενο να βρεθούν το όνομα και το επίθετό του, καθώς και το όνομα και το επίθετο του άμεσου προϊσταμένου του.

```
E8: SELECT E.ΟΝΟΜΑ, E.ΕΠΙΘΕΤΟ, S.ΟΝΟΜΑ, S.ΕΠΙΘΕΤΟ
FROM ΕΡΓΑΖΟΜΕΝΟΣ E S
WHERE E.ΠΡΟΙΣΤΑΜΕΝΟΣ=S.AP_TAYT
```

- Στην Ε8, τα εναλλακτικά ονόματα E και S της σχέσης ονομάζονται ψευδώνυμα ή μεταβλητές πλειάδες για τη σχέση ΕΡΓΑΖΟΜΕΝΟΣ
- Οι E και S μπορούν να θεωρηθούν σαν δύο διαφορετικά αντίγραφα της ΕΡΓΑΖΟΜΕΝΟΣ· η E αντιπροσωπεύει εργαζόμενους σε ρόλο υφισταμένων και η S αντιπροσωπεύει εργαζόμενους σε ρόλο προϊσταμένων.

Ψευδώνυμα (συνεχ.)

- Για ευκολία μπορούν να χρησιμοποιηθούν ψευδώνυμα σε οποιαδήποτε SQL ερώτηση
- Για τον προσδιορισμό ψευδωνύμων μπορεί να χρησιμοποιηθεί επίσης η λέξη κλειδί AS
- Ε8:

```
SELECT E.ΟΝΟΜΑ, E.ΕΠΙΘΕΤΟ, S.ΟΝΟΜΑ, S.ΕΠΙΘΕΤΟ
FROM ΕΡΓΑΖΟΜΕΝΟΣ AS E, ΕΡΓΑΖΟΜΕΝΟΣ AS S
WHERE E.ΠΡΟΙΣΤΑΜΕΝΟΣ=S.ΑΡ_ΤΑΥΤ
```

ΜΗ ΥΠΑΡΞΗ ΠΡΟΤΑΣΗΣ WHERE

- Η μη ύπαρξη της πρότασης *WHERE* σημαίνει ότι δεν υπάρχει συνθήκη· επομένως, επιλέγονται όλες οι πλειάδες από τις σχέσεις της πρότασης *FROM*
 - Είναι ισοδύναμο με την συνθήκη *WHERE TRUE*
- Ερώτηση 9: Να επιλεγούν όλοι οι *AP_TAYT* από την *ΕΡΓΑΖΟΜΕΝΟΣ*.
 - **E9: `SELECT AP_TAYT FROM ΕΡΓΑΖΟΜΕΝΟΣ`**
- Αν στην πρόταση *FROM* ορίζονται περισσότερες από μια σχέσεις και δεν υπάρχει συνθήκη συνένωσης, τότε επιλέγεται το *ΚΑΡΤΕΣΙΑΝΟ ΓΙΝΟΜΕΝΟ* των πλειάδων

ΜΗ ΥΠΑΡΞΗ ΠΡΟΤΑΣΗΣ WHERE (συν.)

- Παράδειγμα:
- Ε10:

```
SELECT AP_ΤΑΥΤ, Τ_ΟΝΟΜΑ  
FROM ΕΡΓΑΖΟΜΕΝΟΣ, ΤΜΗΜΑ
```

 - Είναι πολύ σημαντικό να μην παραβλεφθεί ο προσδιορισμός οποιονδήποτε συνθηκών επιλογής και συνένωσης στην WHERE-πρόταση· διαφορετικά, μπορεί να έχουμε λάθος αποτέλεσμα ή μεγάλες σχέσεις.

ΧΡΗΣΗ ΤΟΥ *

- Για ανάκτηση τις τιμές όλων των γνωρισμάτων των επιλεγμένων πλειάδων χρησιμοποιείται ένας *, που σημαίνει για όλα τα γνωρίσματα
- Παραδείγματα:
 - Ε1Γ:

```
SELECT *  
FROM ΕΡΓΑΖΟΜΕΝΟΣ  
WHERE ΑΡΙΘ_Τ=5
```
 - Ε1Δ:

```
SELECT *  
FROM ΕΡΓΑΖΟΜΕΝΟΣ, ΤΜΗΜΑ  
WHERE Τ_ΟΝΟΜΑ='Research'  
AND ΑΡΙΘ_Τ=ΚΩΔ_ΤΜΗΜ
```

ΧΡΗΣΗ ΤΗΣ DISTINCT

- Η SQL δεν αντιμετωπίζει μια σχέση σαν ένα σύνολο· μπορούν να εμφανισθούν διπλότυπες πλειάδες
- Για απαλοιφή διπλότυπων πλειάδων στο αποτέλεσμα μιας ερώτησης, χρησιμοποιείται η λέξη κλειδί **DISTINCT**
- Για παράδειγμα, το αποτέλεσμα της E11 μπορεί να περιέχει διπλότυπες τιμές ΜΙΣΘΟΣ ενώ η E11A δεν έχει διπλότυπες τιμές
 - E11: **SELECT** ΜΙΣΘΟΣ
FROM ΕΡΓΑΖΟΜΕΝΟΣ
 - E11A: **SELECT DISTINCT** ΜΙΣΘΟΣ
FROM ΕΡΓΑΖΟΜΕΝΟΣ

ΠΡΑΞΕΙΣ ΣΥΝΟΛΩΝ

- Η SQL έχει άμεσα ενσωματωμένες κάποιες πράξεις συνόλων
- Υπάρχει η πράξη της ένωσης (UNION), και σε μερικές εκδόσεις της SQL υπάρχουν οι πράξεις της διαφοράς συνόλων (MINUS) και της τομής (INTERSECT)
- Οι σχέσεις που προκύπτουν από αυτές τις πράξεις συνόλων είναι σύνολα πλειάδων· από το αποτέλεσμα απαλείφονται οι διπλότυπες πλειάδες
- Οι πράξεις των συνόλων εφαρμόζονται μόνο σε σχέσεις συμβατές προς την ένωση· θα πρέπει οι δύο σχέσεις να έχουν τα ίδια γνωρίσματα και θα πρέπει τα γνωρίσματα να εμφανίζονται με την ίδια σειρά.

ΠΡΑΞΕΙΣ ΣΥΝΟΛΩΝ (συν.)

- Ερώτηση 4: Βρείτε τη λίστα όλων των κωδικών έργων στα οποία εμφανίζονται εργαζόμενοι με επίθετο 'Smith', είτε ως απασχολούμενοι σε ένα έργο είτε ως διευθυντές του τμήματος που ελέγχει το έργο αυτό.

```
– E4:      ( SELECT DISTINCT ΚΩΔ_ΕΡΓΟΥ
            FROM ΕΡΓΟ, ΤΜΗΜΑ, ΕΡΓΑΖΟΜΕΝΟΣ
            WHERE Κ_ΤΜΗΜΑ=ΚΩΔ_ΤΜΗΜ AND
            ΔΙΕΥΘΥΝΤΗΣ=ΑΡ_ΤΑΥΤ
            AND ΕΠΙΘΕΤΟ='Smith' )
UNION
( SELECT DISTINCT ΚΩΔ_ΕΡΓΟΥ
  FROM ΕΡΓΟ, ΑΠΑΣΧΟΛΗΣΗ, ΕΡΓΑΖΟΜΕΝΟΣ
  WHERE ΚΩΔ_ΕΡΓΟΥ=Κ_ΕΡΓΟ AND
  Ε_ΑΡΤΑΥΤ=ΑΡ_ΤΑΥΤ
  AND ΕΠΙΘΕΤΟ='Smith' )
```

ΕΜΦΩΛΕΥΣΗ ΕΡΩΤΗΣΕΩΝ

- Σε μια πρόταση WHERE μιας ερώτησης μπορεί να υπάρχει μια άλλη πλήρης SELECT ερώτηση, που ονομάζεται μια *εμφωλευμένη ερώτηση*, ενώ η ερώτηση στην οποία περικλείεται λέγεται *εξωτερική ερώτηση*
 - Πολλές από τις μέχρι τώρα ερωτήσεις μπορούν εναλλακτικά να γραφούν με χρήση εμφώλευσης
- Ερώτηση 1: Να βρεθούν το όνομα και η διεύθυνση όλων των εργαζομένων στο τμήμα 'Research'
 - E1:

```
SELECT ONOMA, ΕΠΙΘΕΤΟ, ΔΙΕΥΘΥΝΣΗ
FROM ΕΡΓΑΖΟΜΕΝΟΣ
WHERE ΑΡΙΘ_Τ IN (SELECT ΚΩΔ_ΤΜΗΜ
FROM ΤΜΗΜΑ
WHERE Τ_ONOMA='Research' )
```

ΕΜΦΩΛΕΥΣΗ ΕΡΩΤΗΣΕΩΝ (συν.)

- Η εμφωλευμένη ερώτηση επιλέγει τον κωδικό του τμήματος 'Research'.
- Η εξωτερική ερώτηση επιλέγει μια πλειάδα ΕΡΓΑΖΟΜΕΝΟΣ αν η τιμή της ΑΡΙΘ_Τ είναι στο αποτέλεσμα της εμφωλευμένης
- Ο τελεστής σύγκρισης IN συγκρίνει μια τιμή n με ένα σύνολο (ή πολυσύνολο) τιμών V , και αποτιμάται σε TRUE αν το n είναι ένα από τα στοιχεία του V
- Γενικά, μπορούμε να έχουμε πολλά επίπεδα εμφωλευμένων ερωτήσεων
- Μια αναφορά σε ένα γνώρισμα σχέσης που δεν προσδιορίζεται αφορά την *πιο εσωτερικά εμφωλευμένη ερώτηση*
- Στο παράδειγμα αυτό, η εμφωλευμένη ερώτηση δεν συσχετίζεται με την εξωτερική ερώτηση.

ΣΥΣΧΕΤΙΣΜΕΝΕΣ ΕΜΦΩΛΕΥΜΕΝΕΣ ΕΡΩΤΗΣΕΙΣ

- Αν μια συνθήκη στην WHERE-πρόταση μιας εμφωλευμένης ερώτησης αναφέρεται σε ένα γνώρισμα μιας σχέσης που ορίζεται στην εξωτερική ερώτηση, λέμε ότι οι δύο σχέσεις είναι *συσχετισμένες*
 - Το αποτέλεσμα μιας συσχετισμένης εμφωλευμένης ερώτησης είναι διαφορετικό για κάθε πλειάδα (ή συνδυασμό πλειάδων) της(των) σχέσης(ων) της εξωτερικής ερώτησης
- Ερώτηση 12: Να ανακτηθεί το όνομα και το επίθετο κάθε εργαζόμενου με ένα εξαρτώμενο μέλος με το ίδιο όνομα με τον εργαζόμενο.

```
– E12:      SELECT E.ONOMA, E.ΕΠΙΘΕΤΟ
            FROM ΕΡΓΑΖΟΜΕΝΟΣ AS E
            WHERE E.AP_TAYT IN
              (SELECT E.AP_TAYT
               FROM ΕΞΑΡΤΩΜΕΝΟΣ
               WHERE E.AP_TAYT =E.AP_TAYT
                AND E.ONOMA=ONOMA_ΕΞΑΡΤΩΜΕΝΟΥ)
```

ΣΥΣΧΕΤΙΣΜΕΝΕΣ ΕΜΦΩΛΕΥΜΕΝΕΣ ΕΡΩΤΗΣΕΙΣ (συν)

- Στην E12, η εμφωλευμένη ερώτηση έχει διαφορετικό αποτέλεσμα στην εξωτερική ερώτηση
- Μια εμφωλευμένη ερώτηση με `SELECT... FROM... WHERE...` μπλοκ και χρήση των τελεστών σύγκρισης `=` ή `IN` μπορεί πάντα να γραφεί σαν μια απλή ερώτηση ενός μπλοκ.
- Για παράδειγμα, η E12 μπορεί να γραφεί σαν E12A
 - E12A:

```
SELECT E.ΟΝΟΜΑ, E.ΕΠΙΘΕΤΟ
FROM ΕΡΓΑΖΟΜΕΝΟΣ E, ΕΞΑΡΤΩΜΕΝΟΣ D
WHERE E.ΑΡ_ΑΤΑΥΤ=D.ΑΡ_ΤΑΥΤ AND
E.ΟΝΟΜΑ=D.ΟΝΟΜΑ_ΕΞΑΡΤΩΜΕΝΟΥ
```

ΣΥΣΧΕΤΙΣΜΕΝΕΣ ΕΜΦΩΛΕΥΜΕΝΕΣ ΕΡΩΤΗΣΕΙΣ (συν)

- Η αρχική SQL όπως ορίστηκε για το SYSTEM R έχει και τον τελεστή σύγκρισης CONTAINS, που χρησιμοποιείται σε συνδυασμό με εμφωλευμένες συσχετισμένες ερωτήσεις
 - Ο τελεστής αυτός εγκαταλείφτηκε από τη γλώσσα, πιθανόν λόγω δυσκολίας για αποτελεσματική υλοποίηση
 - Οι περισσότερες υλοποιήσεις της SQL δεν υποστηρίζουν αυτόν τον τελεστή
 - Ο τελεστής CONTAINS συγκρίνει δύο σύνολα τιμών, και αποτιμάται σε TRUE αν το ένα σύνολο περιέχει όλες τις τιμές του άλλου συνόλου
 - Που θυμίζει την πράξη της διαίρεσης στην άλγεβρα

ΣΥΣΧΕΤΙΣΜΕΝΕΣ ΕΜΦΩΛΕΥΜΕΝΕΣ ΕΡΩΤΗΣΕΙΣ (συν)

- Ερώτηση 3: Να βρεθεί το ονοματεπώνυμο κάθε εργαζόμενου που εργάζεται σε όλα τα έργα που ελέγχονται από το τμήμα 5

```
– Ε3:  SELECT ΟΝΟΜΑ, ΕΠΙΘΕΤΟ
FROM ΕΡΓΑΖΟΜΕΝΟΣ
WHERE ( (SELECT Κ_ΕΡΓΟ
          FROM ΑΠΑΣΧΟΛΗΣΗ
          WHERE ΑΡ_ΤΑΥΤ=E_ΑΡΤΑΥΤ)
        CONTAINS
        (SELECT ΚΩΔ_ΕΡΓΟΥ
          FROM ΕΡΓΟ
          WHERE Κ_ΤΜΗΜΑ=5) )
```


ΣΥΣΧΕΤΙΣΜΕΝΕΣ ΕΜΦΩΛΕΥΜΕΝΕΣ ΕΡΩΤΗΣΕΙΣ (συν)

- Στην Ε3, η δεύτερη εμφωλευμένη ερώτηση, που *δεν συσχετίζεται* με την εξωτερική ερώτηση, ανακτά τους κωδικούς όλων των έργων που ελέγχονται από το τμήμα 5
- Η πρώτη εμφωλευμένη ερώτηση, που συσχετίζεται, τους κωδικούς των έργων που απασχολείται ο εργαζόμενος, που είναι *διαφορετικοί για κάθε πλειάδα εργαζόμενου* λόγω της συσχέτισης.

Η ΣΥΝΑΡΤΗΣΗ EXISTS

- Η EXISTS χρησιμοποιείται για να ελεγχθεί αν το αποτέλεσμα μιας συσχετισμένης εμφωλευμένης ερώτησης είναι κενό (δεν περιέχει πλειάδες) ή όχι
 - Μπορούμε να γράψουμε της ερώτηση E12 με εναλλακτική μορφή που χρησιμοποιεί την EXISTS όπως η E12B

Η ΣΥΝΑΡΤΗΣΗ EXISTS (συν.)

- Ερώτηση 12: Να ανακτηθεί το όνομα και το επίθετο κάθε εργαζόμενου με ένα εξαρτώμενο μέλος με το ίδιο όνομα με τον εργαζόμενο.

```
– E12B: SELECT ΟΝΟΜΑ, ΕΠΙΘΕΤΟ
FROM ΕΡΓΑΖΟΜΕΝΟΣ
WHERE EXISTS (SELECT *
FROM ΕΞΑΡΤΩΜΕΝΟΣ
WHERE ΑΡ_ΤΑΥΤ=E_ΑΡΤΑΥΤ
AND
ΟΝΟΜΑ=ΟΝΟΜΑ_ΕΞΑΡΤΩΜΕΝΟΥ )
```

Η ΣΥΝΑΡΤΗΣΗ EXISTS (συν.)

- Ερώτηση 6: Να ανακτηθούν το όνομα και το επίθετο όσων δεν έχουν εξαρτώμενους.

– E6: **SELECT** ONOMA, ΕΠΙΘΕΤΟ
FROM ΕΡΓΑΖΟΜΕΝΟΣ
WHERE NOT EXISTS (**SELECT** *
FROM ΕΞΑΡΤΩΜΕΝΟΣ
WHERE AP_TAYT=E_APTAYT)

- Στην E6, η συσχετισμένη εμφωλευμένη ερώτηση ανακτά όλες τις πλειάδες ΕΞΑΡΤΩΜΕΝΟΣ που σχετίζονται με μια πλειάδα ΕΡΓΑΖΟΜΕΝΟΣ. Αν δεν υπάρχουν, επιλέγεται η πλειάδα ΕΡΓΑΖΟΜΕΝΟΣ
 - Η EXISTS είναι απαραίτητη για τις εκφραστικές δυνατότητες της SQL

ΡΗΤΑ ΣΥΝΟΛΑ

- Είναι επίσης δυνατόν να χρησιμοποιηθεί **ρητά (απαριθμήσιμο) σύνολο τιμών** στην πρόταση WHERE αντί μιας εμφωλευμένης ερώτησης
- Ερώτηση 13: Να ανακτηθούν οι αριθμοί ταυτότητας όλων των εργαζομένων που απασχολούνται στα έργα με κωδικούς 1, 2, ή 3.

```
– E13:  SELECT DISTINCT E_ΑΡΤΑΥΤ  
       FROM ΑΠΑΣΧΟΛΗΣΗ  
       WHERE Κ_ΕΡΓΟ IN (1, 2, 3)
```

TΙΜΕΣ NULL ΣΕ SQL ΕΡΩΤΗΣΕΙΣ

- Η SQL υποστηρίζει ερωτήσεις που ελέγχουν αν μια τιμή είναι **NULL** (δεν έχει καταγραφεί, ή δεν ορίζεται ή δεν ισχύει)
- Η SQL χρησιμοποιεί την **IS** ή την **IS NOT** για σύγκριση τιμών NULL επειδή θεωρεί κάθε NULL τιμή διακριτή από τις άλλες NULL τιμές, έτσι *δεν είναι κατάλληλη η σύγκριση ισότητας*.
- Ερώτηση 14: Να ανακτηθούν το όνομα και το επίθετο των εργαζομένων που δεν έχουν προϊσταμένους.
 - E14:

```
SELECT ONOMA, ΕΠΙΘΕΤΟ
FROM ΕΡΓΑΖΟΜΕΝΟΣ
WHERE ΠΡΟΙΣΤΑΜΕΝΟΣ IS NULL
```
 - Σημείωση: Αν ορίζεται μια συνθήκη συνένωσης, οι πλειάδες με τιμές NULL στα γνωρίσματα συνένωσης δεν συμπεριλαμβάνονται στο αποτέλεσμα.

Σχέσεις Συνένωσης στην SQL2

- Μπορούμε να προσδιορίσουμε «μια σχέση συνένωσης» στην πρόταση FROM
 - Μοιάζει με οποιαδήποτε άλλη σχέση αλλά είναι το αποτέλεσμα μιας συνένωσης
 - Επιτρέπει στον χρήστη να ορίζει διαφορετικούς τύπους συνένωσης (κανονικές «θήτα» JOIN, NATURAL JOIN, LEFT OUTER JOIN, RIGHT OUTER JOIN, CROSS JOIN, κλπ)

Σχέσεις Συνένωσης στην SQL2 (συν.)

- Παραδείγματα:

- Ε8:

```
SELECT E.ΟΝΟΜΑ, E.ΕΠΙΘΕΤΟ,  
       S.ΟΝΟΜΑ, S.ΕΠΙΘΕΤΟ  
FROM ΕΡΓΑΖΟΜΕΝΟΣ E S  
WHERE E.ΠΡΟΙΣΤΑΜΕΝΟΣ=S.ΑΡ_ΤΑΥΤ
```

- Μπορεί να γραφτεί:

- Ε8:

```
SELECT E.ΟΝΟΜΑ, E.ΕΠΙΘΕΤΟ,  
       S.ΟΝΟΜΑ, S.ΕΠΙΘΕΤΟ  
FROM (ΕΡΓΑΖΟΜΕΝΟΣ E LEFT OUTER JOIN  
      ΕΡΓΑΖΟΜΕΝΟΣ ON  
      E.ΠΡΟΙΣΤΑΜΕΝΟΣ=S.ΑΡ_ΤΑΥΤ)
```


Σχέσεις Συνένωσης στην SQL2 (συν.)

- Παραδείγματα:

- E1:

```
SELECT ONOMA, ΕΠΙΘΕΤΟ, ΔΙΕΥΘΥΝΣΗ
FROM ΕΡΓΑΖΟΜΕΝΟΣ, ΤΜΗΜΑ
WHERE T_ONOMA='Research' AND
      ΚΩΔ_ΤΜΗΜ=ΑΡΙΘ_Τ
```

- Μπορεί να γραφεί ως:

- E1:

```
SELECT ONOMA, ΕΠΙΘΕΤΟ, ΔΙΕΥΘΥΝΣΗ
FROM (ΕΡΓΑΖΟΜΕΝΟΣ JOIN ΤΜΗΜΑ
      ON ΚΩΔ_ΤΜΗΜ=ΑΡΙΘ_Τ)
WHERE T_ONOMA='Research'
```

- ή σαν:

- E1:

```
SELECT ONOMA, ΕΠΙΘΕΤΟ, ΔΙΕΥΘΥΝΣΗ
FROM (ΕΡΓΑΖΟΜΕΝΟΣ NATURAL JOIN ΤΜΗΜΑ
      AS ΤΜΗΜ(T_ONOMA, ΑΡΙΘ_Τ, ΔΙΕΥΘ,
             ΗΜΕΡ_ΕΝΑΡ))
WHERE T_ONOMA='Research'
```

Σχέσεις Συνένωσης στην SQL2 (συν.)

- Άλλο παράδειγμα: Η Ε2 θα μπορούσε να γραφεί σαν· εδώ έχουμε πολλαπλή συνένωση πινάκων

```
– Ε2:  SELECT ΚΩΔ_ΕΡΓΟΥ, Κ_ΤΜΗΜΑ,  
        ΕΠΙΘΕΤΟ, ΗΜ_ΓΕΝ, ΔΙΕΥΘΥΝΣΗ  
FROM (ΕΡΓΟ JOIN ΤΜΗΜΑ ON  
        Κ_ΤΜΗΜΑ=ΚΩΔ_ΤΜΗΜ) JOIN  
        ΕΡΓΑΖΟΜΕΝΟΣ ON  
        ΔΙΕΥΘΥΝΤΗΣ=ΑΡ_ΑΤΑΥΤ) )  
WHERE ΤΟΠ_ΕΡΓΟΥ='Stafford'
```

ΣΥΝΑΘΡΟΙΣΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ

- Είναι οι **COUNT**, **SUM**, **MAX**, **MIN**, και **AVG**
- Ερώτηση 15: Να βρεθεί ο μέγιστος μισθός, ο ελάχιστος μισθός, και ο μέσος μισθός των εργαζομένων.
 - E15:

```
SELECT MAX(ΜΙΣΘΟΣ) ,  
       MIN(ΜΙΣΘΟΣ) ,  
       AVG(ΜΙΣΘΟΣ)  
FROM ΕΡΓΑΖΟΜΕΝΟΣ
```
- Μερικές υλοποιήσεις της SQL μπορεί να μην επιτρέπουν περισσότερες από μια συναρτήσεις στην SELECT-πρόταση

ΣΥΝΑΘΡΟΙΣΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ (συν.)

- Ερώτηση 16: Να βρεθεί ο μέγιστος μισθός, ο ελάχιστος μισθός, και ο μέσος μισθός των εργαζομένων που εργάζονται στο τμήμα 'Research'.

– E16: **SELECT MAX**(ΜΙΣΘΟΣ) ,
MIN(ΜΙΣΘΟΣ) , **AVG**(ΜΙΣΘΟΣ)
FROM EMPLOYEE , DEPARTMENT
WHERE ΑΡΙΘ_Τ=ΚΩΔ_ΤΜΗΜ
AND Τ_ΟΝΟΜΑ='Research'

ΣΥΝΑΘΡΟΙΣΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ (συν.)

- Ερωτήσεις 17 και 18: Να βρεθεί ο συνολικός αριθμός των εργαζομένων στην εταιρεία (E21) και ο αριθμός των εργαζομένων στο τμήμα 'Research' (E22).

– E17: **SELECT COUNT (*) FROM ΕΡΓΑΖΟΜΕΝΟΣ**

– E18: **SELECT COUNT (*)
FROM ΕΡΓΑΖΟΜΕΝΟΣ, ΤΜΗΜΑ
WHERE ΑΡΙΘ_Τ=ΚΩΔ_ΤΜΗΜ
AND Τ_ΟΝΟΜΑ='Research'**

ΟΜΑΔΟΠΟΙΗΣΗ

- Σε πολλές περιπτώσεις, θέλουμε να εφαρμόσουμε τις συναθροιστικές συναρτήσεις σε *υποομάδες πλειάδων* σε μια σχέση
- Κάθε υποομάδα πλειάδων αποτελείται από το σύνολο των πλειάδων που έχουν την *ίδια τιμή* για τα *γνωρίσματα ομαδοποίησης*
- Η συνάρτηση εφαρμόζεται ανεξάρτητα σε κάθε υποομάδα
- Η SQL έχει την πρόταση **GROUP BY** για προσδιορισμό των γνωρισμάτων της ομάδας που *πρέπει να εμφανισθούν και στην πρόταση SELECT*.

ΟΜΑΔΟΠΟΙΗΣΗ (συν.)

- Ερώτηση 20: Για κάθε τμήμα να βρεθεί ο κωδικός του τμήματος, το πλήθος των εργαζομένων στο τμήμα και ο μέσος μισθός τους
 - Ε20:

```
SELECT ΑΡΙΘ_Τ, COUNT (*), AVG (ΜΙΣΘΟΣ)  
FROM ΕΡΓΑΖΟΜΕΝΟΣ GROUP BY ΑΡΙΘ_Τ
```
 - Στην Ε20, οι πλειάδες της ΕΡΓΑΖΟΜΕΝΟΣ χωρίζονται σε ομάδες
 - Κάθε ομάδα έχει την ίδια τιμή για το γνώρισμα ομαδοποίησης ΑΡΙΘ_Τ
 - Οι συναρτήσεις COUNT και AVG εφαρμόζονται σε κάθε ομάδα πλειάδων ξεχωριστά
 - Η πρόταση SELECT περιλαμβάνει μόνο το γνώρισμα ομαδοποίησης και τις συναρτήσεις που θα εφαρμοσθούν σε κάθε ομάδα πλειάδων.
 - Σε συνδυασμό με την ομαδοποίηση μπορεί να χρησιμοποιηθεί μια συνθήκη συνένωσης

ΟΜΑΔΟΠΟΙΗΣΗ (συν.)

- Ερώτηση 21: Για κάθε έργο να ανακτηθεί ο κωδικός του έργου, το όνομά του, καθώς και το πλήθος των εργαζομένων που απασχολούνται στο έργο αυτό.
 - E21:

```
SELECT ΚΩΔ_ΕΡΓΟΥ, Ε_ΟΝΟΜΑ, COUNT (*)  
FROM ΕΡΓΟ, ΑΠΑΣΧΟΛΗΣΗ  
WHERE ΚΩΔ_ΕΡΓΟΥ=Κ_ΕΡΓΟ  
GROUP BY ΚΩΔ_ΕΡΓΟΥ, Ε_ΟΝΟΜΑ
```
 - Στην περίπτωση αυτή, η ομαδοποίηση και οι συναρτήσεις εφαρμόζονται μετά την συνένωση των δύο σχέσεων

Η ΠΡΟΤΑΣΗ HAVING

- Μερικές φορές θέλουμε να ανακτήσουμε τις τιμές αυτών των συναρτήσεων μόνο για εκείνες τις ομάδες που ικανοποιούν κάποιες συνθήκες
- Χρησιμοποιείται η πρόταση **HAVING** για τον προσδιορισμό μιας συνθήκης επιλογής στις ομάδες (αντί στις πλειάδες ατομικά)

Η ΠΡΟΤΑΣΗ HAVING (συν.)

- Ερώτηση 22: Για κάθε έργο στο οποίο απασχολούνται περισσότεροι από δύο εργαζόμενοι, να ανακτηθεί ο κωδικός του έργου, το όνομά του και το πλήθος των εργαζομένων που απασχολούνται σε αυτό.

```
– E22:  SELECT ΚΩΔ_ΕΡΓΟΥ, Ε_ΟΝΟΜΑ,  
        COUNT ( * )  
FROM ΕΡΓΟ, ΑΠΑΣΧΟΛΗΣΗ  
WHERE ΚΩΔ_ΕΡΓΟΥ=Κ_ΕΡΓΟ  
GROUP BY ΚΩΔ_ΕΡΓΟΥ, Ε_ΟΝΟΜΑ  
HAVING COUNT ( * ) > 2
```

ΣΥΓΚΡΙΣΕΙΣ ΣΥΜΒΟΛΟΣΕΙΡΩΝ

- Ο τελεστής σύγκρισης **LIKE** χρησιμοποιείται για μερική σύγκριση συνόλων
- Χρησιμοποιούνται δύο δεσμευμένοι χαρακτήρες: το '%' (ή το '*' σε μερικές υλοποιήσεις) αντικαθιστά ένα αυθαίρετο πλήθος από χαρακτήρες και το '_' αντικαθιστά ένα χαρακτήρα

ΣΥΓΚΡΙΣΕΙΣ ΣΥΜΒΟΛΟΣΕΙΡΩΝ (συν.)

- Ερώτηση 25: Να ανακτηθούν όλοι οι εργαζόμενοι με διεύθυνση στο Houston, Texas. Εδώ, η τιμή του γνωρίσματος ΔΙΕΥΘΥΝΣΗ πρέπει να περιέχει την συμβολοσειρά 'Houston, TX'.

```
– E25:  SELECT ΟΝΟΜΑ, ΕΠΙΘΕΤΟ  
        FROM ΕΡΓΑΖΟΜΕΝΟΣ  
        WHERE ΔΙΕΥΘΥΝΣΗ  
        LIKE ' %Houston, TX% '
```

ΣΥΓΚΡΙΣΕΙΣ ΣΥΜΒΟΛΟΣΕΙΡΩΝ (συν.)

- Ερώτηση 26: Να βρεθούν όλοι οι εργαζόμενοι που γεννήθηκαν την δεκαετία του '50 .
 - Εδώ, το '5' πρέπει να είναι ο τρίτος χαρακτήρας της συμβολοσειράς (σύμφωνα με την μορφή της ημερομηνίας), ώστε η τιμή HM_ΓΕΝ είναι '__5_____', με την υπογράμμιση σαν θέση αυθαίρετα ενός χαρακτήρα.
 - E26:

```
SELECT ΟΝΟΜΑ, ΕΠΙΘΕΤΟ
FROM ΕΡΓΑΖΟΜΕΝΟΣ
WHERE HM_ΓΕΝ LIKE '__5_____'
```
 - Ο τελεστής LIKE μας επιτρέπει να ξεπερνάμε το γεγονός ότι κάθε τιμή θεωρείτε ατομική και μη διαχωρίσιμη
 - Επομένως, στην SQL, οι τιμές των γνωρισμάτων συμβολοσειρών δεν είναι ατομικές

ΑΡΙΘΜΗΤΙΚΟΙ ΤΕΛΕΣΤΕΣ

- Στις αριθμητικές τιμές του αποτελέσματος μιας SQL ερώτησης μπορούν να εφαρμοσθούν οι τυπικοί αριθμητικοί τελεστές '+', '-', '*', και '/' (πρόσθεση, αφαίρεση, πολλαπλασιασμός, και διαίρεση, αντίστοιχα)
- Ερώτηση 27: Να βρεθούν οι μισθοί που προκύπτουν αν δοθεί αύξηση 10% σε κάθε εργαζόμενο που απασχολείται στο έργο 'ProductX' .
 - E27: **SELECT** ΟΝΟΜΑ, ΕΠΙΘΕΤΟ, 1.1*ΜΙΣΘΟΣ
FROM ΕΡΓΑΖΟΜΕΝΟΣ, ΑΠΑΣΧΟΛΗΣΗ, ΕΡΓΟ
WHERE ΑΡ ΤΑΥΤ=Ε ΑΡΤΑΥΤ **AND**
Κ ΕΡΓΟ=ΚΩΔ ΕΡΓΟΥ **AND**
Ε_ΟΝΟΜΑ='ProductX'

ORDER BY

- Η πρόταση **ORDER BY** χρησιμοποιείται για ταξινόμηση των πλειάδων του αποτελέσματος μιας ερώτησης με βάση τις τιμές κάποιων γνωρισμάτων
- Ερώτηση 28: Να κατασκευαστεί μια λίστα εργαζομένων και έργων στα οποία απασχολούνται κατά διάταξη τμήματος, και σε κάθε τμήμα αλφαβητικά ως προς το επίθετο και το όνομα.

```
– E28:  SELECT T_ΟΝΟΜΑ, ΕΠΙΘΕΤΟ, ΟΝΟΜΑ,  
        Ε_ΟΝΟΜΑ  
FROM ΤΜΗΜΑ, ΕΡΓΑΖΟΜΕΝΟΣ, ΑΠΑΣΧΟΛΗΣΗ,  
        ΕΡΓΟ  
WHERE ΚΩΔ_ΤΜΗΜ=ΑΡΙΘ_Τ AND  
        ΑΡ_ΤΑΥΤ=Ε_ΑΡΤΑΥΤ AND  
        Κ_ΕΡΓΟ=ΚΩΔ_ΕΡΓΟΥ  
ORDER BY Τ_ΟΝΟΜΑ, ΕΠΙΘΕΤΟ, ΟΝΟΜΑ
```

ORDER BY (συν.)

- Η προκαθορισμένη διάταξη είναι αύξουσα σειρά τιμών
- Αν θέλουμε φθίνουσα σειρά μπορούμε να χρησιμοποιήσουμε τη λέξη κλειδί **DESC** · για να ορισθεί ρητά αύξουσα σειρά μπορούμε να χρησιμοποιήσουμε τη λέξη κλειδί **ASC** αν και είναι η προκαθορισμένη

Περίληψη των SQL Ερωτήσεων

- Μια ερώτηση στην SQL μπορεί να αποτελείται από μέχρι 6 προτάσεις, αλλά μόνο οι δύο πρώτες, SELECT και FROM, είναι υποχρεωτικές. Οι προτάσεις προσδιορίζονται με την παρακάτω σειρά:

SELECT	<λίστα γνωρισμάτων>
FROM	<λίστα πινάκων>
[WHERE	<συνθήκη>]
[GROUP BY	<γνώρισμα(τα) ομαδοποίησης>]
[HAVING	<συνθήκη ομάδας>]
[ORDER BY	<λίστα γνωρισμάτων>]

Περίληψη των SQL Ερωτήσεων (συν.)

- Η πρόταση SELECT παραθέτει τα γνωρίσματα ή τις συναρτήσεις που θα ανακτηθούν
- Η πρόταση FROM καθορίζει όλες τις σχέσεις (ή τα ψευδώνυμα) που απαιτούνται στην ερώτηση αλλά όχι όσες απαιτούνται στις εμφωλευμένες ερωτήσεις
- Η πρόταση WHERE καθορίζει τις συνθήκες για την επιλογή και την συνένωση πλειάδων από σχέσεις που καθορίζονται στην πρόταση FROM
- GROUP BY καθορίζει γνωρίσματα ομαδοποίησης
- HAVING καθορίζει την συνθήκη επιλογής των ομάδων
- ORDER BY καθορίζει την διάταξη παρουσίασης του αποτελέσματος
 - Μια ερώτηση υπολογίζεται με πρώτα εφαρμογή της πρότασης WHERE, στη συνέχεια η GROUP BY και η HAVING, και τέλος η πρόταση SELECT

Ενημερώσεις στην SQL

- Υπάρχουν τρεις SQL εντολές για τροποποίηση της βάσης δεδομένων: **INSERT**, **DELETE**, και **UPDATE**

INSERT

- Στην απλούστερη μορφή της, χρησιμοποιείται για εισαγωγή πλειάδων σε μια σχέση
- Οι τιμές των γνωρισμάτων πρέπει να δοθούν με την ίδια σειρά που ορίσθηκαν στην εντολή **CREATE TABLE**

INSERT (συν.)

- Παράδειγμα:

- EN1:

```
INSERT INTO ΕΡΓΑΖΟΜΕΝΟΣ
VALUES ('Richard', 'K', 'Marini',
       '653298653', '30-DEC-52',
       '98 Oak Forest, Katy, TX', 'M',
       37000, '987654321', 4 )
```

- Μια εναλλακτική μορφή της INSERT ορίζει ρητά τα ονόματα των γνωρισμάτων που αντιστοιχούν στις τιμές στη νέα πλειάδα

- Μπορούν να παραληφθούν γνωρίσματα με τιμή NULL

- Παράδειγμα: Εισαγωγή μιας νέας πλειάδα στη σχέση ΕΡΓΑΖΟΜΕΝΟΣ που ξέρουμε μόνο τα γνωρίσματα ΟΝΟΜΑ, ΕΠΙΘΕΤΟ, και ΑΡ_ΤΑΥΤ.

- EN1A:

```
INSERT INTO ΕΡΓΑΖΟΜΕΝΟΣ
      (ΟΝΟΜΑ, ΕΠΙΘΕΤΟ, ΑΡ ΤΑΥΤ)
VALUES ('Richard', 'Marini', 653298653')
```

INSERT (συν.)

- Σημείωση: Κατά την ενημέρωση, επιβάλλονται αυτόματα, από το ΣΔΒΔ, μόνο οι περιορισμοί που προσδιορίζονται στις εντολές της ΓΟΔ
 - Μια άλλη παραλλαγή της INSERT επιτρέπει εισαγωγή πολλών πλειάδων που προκύπτουν από μια ερώτηση σε μια σχέση

INSERT (συν.)

- Παράδειγμα: Έστω ότι θέλουμε να δημιουργήσουμε ένα προσωρινό πίνακα που έχει το όνομα, το πλήθος των εργαζομένων, και τη συνολική μισθοδοσία για κάθε τμήμα.
 - Δημιουργείται ένας πίνακας ΠΛΗΡ_ΤΜΗΜΑ από την ΕΝ3Α, και φορτώνεται με τις πληροφορίες που ανακτώνται από τη βάση δεδομένων από την ερώτηση ΕΝ3Β.
 - ΕΝ3Α:

```
CREATE TABLE ΠΛΗΡ_ΤΜΗΜΑ
    (ΤΜ_ΟΝΟΜΑ VARCHAR(10),
    ΠΛΗΘΟΣ_ΕΡΓ INTEGER,
    ΟΛΙΚΟΣ_ΜΙΣΘ INTEGER);
```
 - ΕΝ3Β:

```
INSERT INTO ΠΛΗΡ_ΤΜΗΜΑ (ΤΜ_ΟΝΟΜΑ,
    ΠΛΗΘΟΣ_ΕΡΓ, ΟΛΙΚΟΣ_ΜΙΣΘ)
SELECT T_ΟΝΟΜΑ, COUNT (*), SUM (ΜΙΣΘΟΣ)
FROM ΤΜΗΜΑ, ΕΡΓΑΖΟΜΕΝΟΣ
WHERE ΚΩΔ_ΤΜΗΜ=ΑΡΙΘ_Τ
GROUP BY Τ_ΟΝΟΜΑ;
```

INSERT (συν.)

- Σημείωση: Ο πίνακας ΠΛΗΡ_ΤΜΗΜΑ μπορεί να μην είναι ενημερωμένος αν αλλάξουμε τις πλειάδες είτε στη σχέση ΤΜΗΜΑ ή στην ΕΡΓΑΖΟΜΕΝΟΣ μετά εκτέλεση της ΕΝ3Β. Πρέπει να δημιουργήσουμε μια νέα όψη* για να είναι ένας τέτοιος πίνακας ενημερωμένος

*Θα το δούμε σε επόμενα μαθήματα

DELETE

- Διαγράφει πλειάδες από μια σχέση
 - Έχει μια πρόταση WHERE για επιλογή των πλειάδων που θα διαγραφούν
 - Πρέπει να επιβληθεί η αναφορική ακεραιότητα
 - Οι πλειάδες διαγράφονται μόνο από ένα πίνακα τη φορά (εκτός αν έχει ορισθεί αναφορικός περιορισμός ακεραιότητας CASCADE)
 - Έλλειψη της πρότασης WHERE ορίζει ότι όλες οι πλειάδες της σχέσης θα διαγραφούν · ο πίνακας που προκύπτει είναι κενός
 - Το πλήθος των πλειάδων που διαγράφονται εξαρτάται από το πλήθος των πλειάδων της σχέσης που ικανοποιούν την πρόταση WHERE

DELETE (συν.)

- Παραδείγματα:

- EN4A: **DELETE FROM** ΕΡΓΑΖΟΜΕΝΟΣ
WHERE ΕΠΙΘΕΤΟ='Brown'

- EN4B: **DELETE FROM** ΕΡΓΑΖΟΜΕΝΟΣ
WHERE ΑΡ_ΤΑΥΤ='123456789'

- EN4C: **DELETE FROM** ΕΡΓΑΖΟΜΕΝΟΣ
WHERE ΑΡΙΘ_Τ IN (**SELECT** ΚΩΔ_ΤΜΗΜ
FROM ΤΜΗΜΑ
WHERE
Τ_ΟΝΟΜΑ='Research')

- EN4D: **DELETE FROM** ΕΡΓΑΖΟΜΕΝΟΣ

UPDATE

- Χρησιμοποιείται για τροποποίηση τιμών γνωρισμάτων μιας ή περισσότερων πλειάδων
- Μια πρόταση WHERE επιλέγει τις πλειάδες που θα τροποποιηθούν
- Μια επιπλέον πρόταση SET καθορίζει τα γνωρίσματα που θα τροποποιηθούν και τις νέες τιμές τους
- Κάθε εντολή τροποποιεί πλειάδες στην ίδια σχέση
- Πρέπει να επιβληθούν οι αναφορικοί περιορισμοί ακεραιότητας

UPDATE (συν.)

- Παράδειγμα: Να αλλάξει η τοποθεσία και το τμήμα που ελέγχει το έργο 10 σε 'Bellaire' και 5 αντίστοιχα.

```
– EN5:  UPDATE ΕΡΓΟ
        SET  ΤΟΠ_ΕΡΓΟΥ = 'Bellaire',
            Κ_ΤΜΗΜΑ = 5
        WHERE ΚΩΔ_ΕΡΓΟΥ=10
```

UPDATE (συν.)

- Παράδειγμα: Να πάρουν αύξηση στο μισθό 10% όλοι οι εργαζόμενοι στο τμήμα 'Research'.
 - EN6:

```
UPDATE ΕΡΓΑΖΟΜΕΝΟΣ
SET ΜΙΣΘΟΣ = ΜΙΣΘΟΣ *1.1
WHERE ΑΡΙΘ_Τ IN (SELECT ΚΩΔ_ΤΜΗΜ
                  FROM ΤΜΗΜΑ
                  WHERE Τ_ΟΝΟΜΑ='Research' )
```
- Σε αυτήν, η τροποποιημένη τιμή ΜΙΣΘΟΣ εξαρτάται από την αρχική τιμή ΜΙΣΘΟΣ σε κάθε πλειάδα
 - Η αναφορά στο γνώρισμα ΜΙΣΘΟΣ δεξιά του = αφορά την παλαιά τιμή ΜΙΣΘΟΣ πριν την τροποποίηση
 - Η αναφορά στο γνώρισμα ΜΙΣΘΟΣ αριστερά του = αφορά την νέα τιμή ΜΙΣΘΟΣ μετά την τροποποίηση

Σύνοψη των SQL Ερωτήσεων

- Μια ερώτηση στην SQL μπορεί να αποτελείται από μέχρι 6 προτάσεις, αλλά μόνο οι δύο πρώτες, **SELECT** και **FROM**, είναι υποχρεωτικές. Οι προτάσεις προσδιορίζονται με την παρακάτω σειρά:

SELECT	<λίστα γνωρισμάτων>
FROM	<λίστα πινάκων>
[WHERE	<συνθήκη>]
[GROUP BY	<γνώρισμα(τα) ομαδοποίησης>]
[HAVING	<συνθήκη ομάδας>]
[ORDER BY	<λίστα γνωρισμάτων>]

- Υπάρχουν τρεις SQL εντολές για τροποποίηση βάσεων δεδομένων: **INSERT**, **DELETE**, και **UPDATE**