Διάλεξη 13: Γλώσσα Επεξεργασίας Δεδομένων/ Data Manipulation Language (SQL DML) I

Στην ενότητα αυτή θα μελετηθούν τα εξής επιμέρους θέματα:
Εισαγωγή στις έννοιες:

- Εισαγωγή στην SQL-DML
- SELECT, FROM, WHERE, ORDER BY
- DISTINCT
- Πράξεις σε σύνολα

Διδάσκων: Παναγιώτης Ανδρέου

Εισαγωγή

- Η Γλώσσα επεξεργασίας Δεδομένων (SQL-DML) επιτρέπει την επεξεργασία των δεδομένων της ΒΔ
  - Η SQL-DDL επέτρεπε την επεξεργασία της δομής της ΒΔ
- 4 Βασικές Πράξεις: Ανάκτηση, Εισαγωγή, Διαγραφή, Ενημέρωση
- Το πρότυπο SQL:99-DML υλοποιείται σε μεγάλο βαθμό από τους κατασκευαστές βάσεων δεδομένων (π.χ., στην TSQL, PL/SQL)
  - Αντίθετα με την SQL-DDL η οποία αντιμετωπίζει αρκετά προβλήματα συμβατότητας μεταξύ κατασκευαστών.
- Συμβατότητα με το πρότυπο σημαίνει (συνήθως):
  - Αύξηση Μεταφερσιμότητας (Portability)
  - Μείωση Επίδοσης (Performance)

Απλή Ανάκτηση Δεδομένων (SELECT-FROM-WHERE)

- Η βασική έκφραση SQL για διατύπωση επερωτήσεων ονομάζεται **SELECT-FROM-WHERE** μπλοκ *(ή mapping*)

  SELECT <attribute list>

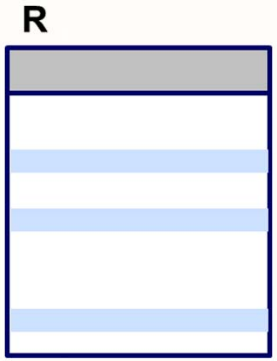
  FROM <table list>

  [**WHERE** <condition>]

  - **<attribute list>** Είναι μια **λίστα γνωρισμάτων** των οποίων η τιμή πρέπει να ανακτηθεί από μια επερώτηση.
    - Αν χρησιμοποιηθεί το * τότε επιστρέφονται όλα τα γνωρίσματα
  - **<table list>** είναι μια **λίστα από ονόματα πινάκων** από τα οποία θα γίνει η ανάκτηση των αποτελεσμάτων.
  - **<condition> είναι μια συνθήκη** (επιστρέφει True/False) η οποία πρέπει να ικανοποιείται από μία πλειάδα που θα συμπεριληφθεί στο τελικό αποτέλεσμα

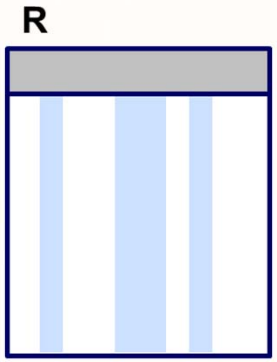
- Τα αποτελέσματα επιστρέφονται κάποτε σε αύξουσα σειρά του attribute list (όχι σε SQL Server).

Αντιστοίχηση Σχεσιακής Άλγεβρας με SQL-DML



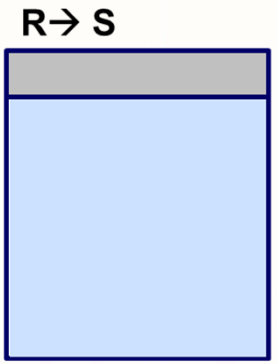
Επιλογή/Selection (σ)

SELECT * FROM R WHERE θ



Προβολή/Projection (π)

SELECT A_1, A_2, … FROM R

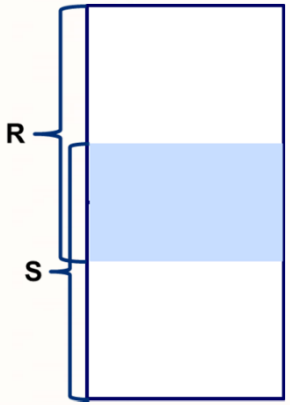


Μετονομασία/Rename (ρ)

SELECT A_1 as id, A_2, … FROM R as S

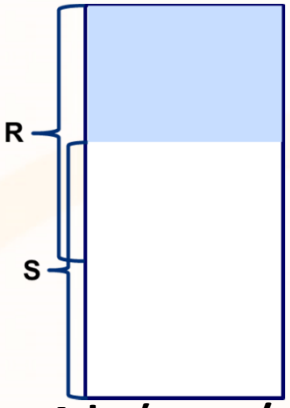
Ένωση/Union (∪)

R UNION S



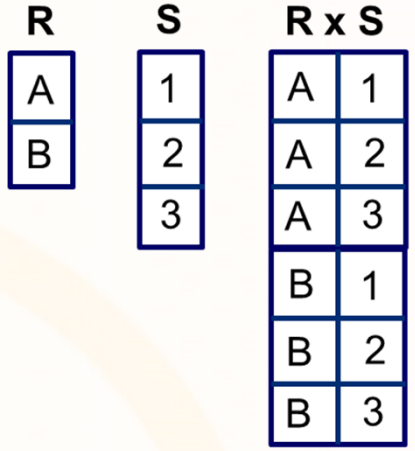
Τομή/Intersection (∩)

R INTERSECT S



Αφαίρεση/
Set Difference (−)

R – S



Καρτεσιανό Γινόμενο/
Cartesian Product (x)

R, S

Αντιστοίχηση Σχεσιακής Άλγεβρας με SQL-DML

R

A	B
a	1
b	2
c	4

S

B	C
1	x
2	y
3	z

$R \bowtie_{\Theta=B\ (R.B=S.B)} S$

A	B	C
a	1	x
b	2	y

Συνένωση Θ/ Θ-Join ($\bowtie_\Theta$)

θ: R.B=S.B

SELECT R.A, R.B, S.C
FROM R JOIN S ON θ

R x S

A	B	B	C
a	1	1	x
a	1	2	y
a	1	3	z
b	2	1	x
b	2	2	y
b	2	3	z
c	4	1	x
c	4	2	y
c	4	3	z

SELECT R.A, R.B, S.C
FROM R , S (CROSS JOIN)

$R * S$

A	B	C
a	1	x
b	2	y

Φυσική Συνένωση/ Natural Join (*)

SELECT A,B,C
FROM R * S

$R \ltimes_B S$

A	B
a	1
b	2

Ημι-Συνένωση / Semi Join ($\ltimes_\Theta$)

SELECT R.A, R.B
FROM R JOIN S ON θ

$R \triangleright_B S$

A	B
c	4

Αντι-Συνένωση / Anti Join ($\triangleright_\Theta$)

?

$R \ \overline{\bowtie}_B \ S$

A	B	C
a	1	x
b	2	y
c	4	

Αριστερή Εξ. Συνένωση/ Left (Outer) Join ($\overline{\bowtie}_\Theta$)

SELECT R.A, R.B , S.C
FROM R
LEFT JOIN S ON θ

$R \ \overline{\bowtie}_B \ S$

A	B	C
a	1	x
b	2	y
c	4	
	3	z

Ολική Εξ. Συνένωση/ Full (Outer) Join ($\overline{\bowtie}_\Theta$)

FROM R FULL
OUTER JOIN S ON θ

Διαφορές της SQL και Σχεσιακής Άλγεβρας

Α) Η SQL στηρίζεται σε **Σύνολα-μη-Διακριτών-Τιμών** ή αλλιώς **Πολυσύνολα (Multi-set),** ενώ το Σχεσιακό Μοντέλο / Άλγεβρα σε **Απλά Σύνολα (διακριτών-τιμών)**.

- Η SQL επιτρέπει τα **διπλότυπα (duplicates)** σε **σχέσεις** και **αποτελέσματα** επερωτήσεων.
- Σημειώστε ότι οι **σχέσεις SQL** μπορούν να **περιοριστούν** έτσι ώστε να **συμπεριφέρονται ως μαθηματικά σύνολα**
  - κάνοντας χρήση περιορισμών **PRIMARY KEY**, **UNIQUE**, ή με χρήση του **DISTINCT**

Β) Σχέσεις SQL και **αποτελέσματα** έχουν **διάταξη (order)** ενώ στο Σχεσιακό Μοντέλο / Άλγεβρα δεν έχουν, δηλ.,

- **Επίπεδο Σχέσης:** Σειρά αποθήκευσης στοιχείων στον δίσκο
- **Επίπεδο Επερώτησης:** Αύξουσα Σειρά / Φθίνουσα Σειρά

Διαφορές της SQL και Σχεσιακής Άλγεβρας

- Ένα **πολυσύνολο (multi-set or bag)** είναι ένα **μη-διατεταγμένο** σύνολο στοιχείων, όπου ένα στοιχείο μπορεί να εμφανίζεται **περισσότερο από μια φορά.**
  - Παράδειγμα: {A, B, C, A} είναι ένα **πολυσύνολο**, ενώ το {A, B, C} είναι **πολυσύνολο** και **απλό σύνολο**.
  - Η SQL παράγει πολυσύνολα στα οποία **υπάρχει διάταξη** (κάποια σειρά) στα **αποτελέσματα (ουσιαστικά λίστες)**.
- Παράδειγμα:
  - {A, B} = {B, A, A} ως Σύνολα
  - {A, B, A} = {B, A, A} ως Πολυσύνολα
  - [A, B, A] ≠ [B, A, A] ως Λίστες (που παράγονται στην SQL)

Σύνταξη SQL-DML: SELECT

[]: optional, Bold: τι θα δούμε στο μάθημα

[WITH <common_table_expression>]

SELECT *select_list* [INTO *new_table*]

[**FROM** *table_source*]

[**WHERE** *search_condition*]

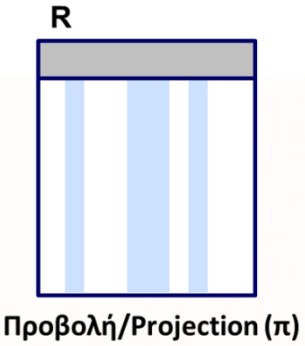
[GROUP BY *group_by_expression*]

[HAVING *search_condition*]

[**ORDER BY** *order_expression* [ASC | DESC]]

http://msdn.microsoft.com/en-us/library/ms189499.aspx

Προβολή/SELECT [FROM] (π)



- **Είσοδος:**
  - Λίστα από γνωρίσματα ή * (*: όλα τα γνωρίσματα)
  - Κενό σύνολο, ένας ή περισσότεροι πίνακες προβολής
- **Έξοδος:** Όψη ή οποία περιλαμβάνει όλες τις πλειάδες από τους πηγαίους πίνακες
- **Παράδειγμα:**
  - Αναφορά με **PID, PNAME** από τον πίνακα **PERSON**
  - **Σχεσιακή Άλγεβρα:** $\pi_{\text{PERSON\_ID,NAME}}$ (PERSON)
  - **SQL: SELECT PID, PNAME FROM** PERSON

PERSON		
PID	**PNAME**	**GENDER**
1	Andreas	M
2	Kostas	M
3	Maria	F
4	Eleni	F

PERSON	
PID	**PNAME**
1	Andreas
2	Kostas
3	Maria
4	Eleni

Γενικευμένη Προβολή/Generalized Projection

- **Γενικευμένη Προβολή (Generalized Projection):** Επεκτείνει την πράξη της προβολής επιτρέποντας να συμπεριληφθούν στη λίστα της προβολής:
  - **Σταθερές τιμές (με πίνακα)**
    - Π.χ., SELECT **'1'**, PID FROM PERSON
  - **Σταθερές τιμές (χωρίς πίνακα)**
    - Π.χ., SELECT **'1'**
  - **Συναρτήσεις γνωρισμάτων**
    - Π.χ., SELECT **PID*5** FROM PERSON
    - Π.χ., SELECT **'Mr/Mrs' + PNAME** FROM PERSON
- **Ισχύει και στην σχεσιακή άλγεβρα**

PERSON

PID	PNAME	GENDER
1	Andreas	M
2	Kostas	M
3	Maria	F
4	Eleni	F

???	PID
1	1
1	2
1	3
1	4

???
1

???
5
10
15
20

???
Mr/Mrs Andreas
Mr/Mrs Kostas
Mr/Mrs Maria
Mr/Mrs Eleni

Μη υπαρκτά Ονόματα

- Όπως είδαμε στη προηγούμενη διαφάνεια, η χρήση σταθερών τιμών ή συναρτήσεων δημιουργεί γνωρίσματα χωρίς επικεφαλίδα
- Μπορούμε να μετονομάζουμε (**aliasing**) κάνοντας χρήση του **AS**.
  - **Σταθερές τιμές (με πίνακα)**
    - Π.χ., SELECT '1' **AS CUSTOM**, PID FROM PERSON
  - **Σταθερές τιμές (χωρίς πίνακα)**
    - Π.χ., SELECT '1' **AS TEST**
  - **Συναρτήσεις γνωρισμάτων**
    - Π.χ., SELECT PID*5 **AS TEST_ID** FROM PERSON
    - Π.χ., SELECT 'Mr/Mrs' + PNAME **AS TEST_NAME** FROM PERSON

PERSON

PID	PNAME	GENDER
1	Andreas	M
2	Kostas	M
3	Maria	F
4	Eleni	F

Χρήση DISTINCT

- Η SQL χειρίζεται τις σχέσεις ως Πολυσύνολα με διάταξη, συνεπώς είναι δυνατό να υπάρχουν **διπλότυπα (duplicate tuples).**
- Για να **εξαλείψουμε τα διπλότυπα** σε μια επερώτηση, κάνουμε χρήση της λέξης **DISTINCT** στον όρο SELECT.
- **Παράδειγμα**

Q1:

```
SELECT  GENDER
FROM    PERSON
```

GENDER
M
M
F
F

Q2:

```
SELECT  DISTINCT GENDER
FROM    PERSON
```

GENDER
M
F

Χρήση DISTINCT/SELECT ALL

- Εκτός από το **SELECT DISTINCT** υπάρχει και το **SELECT ALL**, το οποίο ΔΕΝ αφαιρεί τα διπλότυπα.
- Συγκεκριμένα,

  SELECT **[DISTINCT | ALL]** <attribute-list>

  FROM <table-list>
- Το **SELECT ALL** αντιπροσωπεύει ουσιαστικά την **εξορισμού λειτουργία** του **SELECT**.
- Όπως αναφέραμε και σε προηγούμενες διαλέξεις, στην **SQL** πολλά πράγματα **δηλώνονται ρητά** γιατί έτσι:
  - **Ξεκαθαρίζει η πρόθεση του σχεδιαστή**
  - **Αποφεύγονται προβλήματα συμβατότητας** που μπορεί να προκύψουν από μεταφορά του κώδικα σε άλλη ΒΔ.

Προβολή /SELECT [FROM] (π)

- **Παράδειγμα με δύο ή περισσότερους πίνακες:**
  - Αναφορά με Ταυτότητα, Όνομα Υπαλλήλου και όνομα τμήματος του υπαλλήλου

E: EMPLOYEE

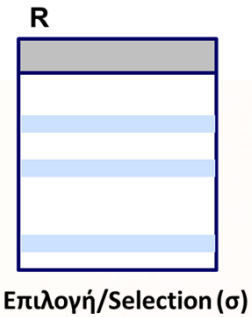
SSN	ENAME	DOB	GENDER	DNO
1	Andreas	1/1/80	M	1
2	Kostas	1/1/85	M	1
3	Maria	4/3/83	F	1
4	Eleni	6/7/85	F	2
5	Victoria	27/2/84	F	3

D: DEPARTMENT

DNO	DNAME
1	Finance
2	Audit
3	Management

- **Σχεσιακή Άλγεβρα:** $\pi_{SSN,ENAME,DNAME}$ (PERSON*DEPARMENT)
- **SQL:** **SELECT** SSN,ENAME,DNAME
  FROM PERSON, DEPARTMENT **(πως συνδέονται οι πίνακες??)**
  WHERE PERSON.DNO=DEPARTMENT.DNO

Επιλογή/ WHERE (σ)



- **Είσοδος:**
  - Λίστα από γνωρίσματα ή * (*: όλα τα γνωρίσματα)
  - Ένας ή περισσότεροι πίνακες
  - **Συνθήκη Θ** (υπολογίζεται σε TRUE/FALSE)
- **Έξοδος:** Όψη ή οποία περιλαμβάνει όλες τις πλειάδες από τους πίνακες που ικανοποιούν την **συνθήκη Θ**
- **Παράδειγμα:**
  - Αναφορά με όλα τα γνωρίσματα από τον πίνακα **PERSON όπου το PID είναι μεγαλύτερο του 2**
  - **Σχεσιακή Άλγεβρα:**
    $\sigma_{PID>2}$ (PERSON)
  - **SQL:**
    SELECT * FROM PERSON **WHERE PID>2**

PERSON

PID	PNAME	GENDER
1	Andreas	M
2	Kostas	M
3	Maria	F
4	Eleni	F

PERSON

PID	PNAME	GENDER
3	Maria	F
4	Eleni	F

Επιλογή/ WHERE (σ)

- Το WHERE υποστηρίζει πολλά είδη συνθηκών:
  - Αριθμητικές Συγκρίσεις: <, >, =, κτλ
  - Συγκρίσεις Εύρους: BETWEEN a AND b
  - Συγκρίσεις Σε Σύνολα: CITY IN ('City-A', 'City-B', 'City-C')
  - Συγκρίσεις με NULL: VALUE IS NULL
  - Πολλά άλλα
- Αν υπάρχουν πολλές **Συνθήκες/Κριτήρια Επιλογής** τότε διαχωρίζονται με λογικούς τελεστές: **AND**, **OR, NOT**
  - Π.χ,. NOT GENDER='F' AND PERSON_ID>3
- Ισχύει η αντιμεταθετική ιδιότητα **(commutative)**
  - Π.χ,. **WHERE GENDER='F' AND PERSON_ID>3** ισοδύναμο με **WHERE PERSON_ID>3 AND GENDER='F'**

Μετονομασία/Rename (ρ)

- Η μετονομασία επιτυγχάνεται με το aliasing (AS)
- Δεν χρειάζεται να γραφεί το AS για την μετονομασία πινάκων
- Ιδιαίτερα χρήσιμο όταν χρειαζόμαστε να χρησιμοποιήσουμε τον ίδιο πίνακα ≥2 φορές (π.χ., supervisor)
- Παραδείγματα:
  - Μετονομασία Πινάκων
    - Π.χ., SELECT * FROM **PERSON (AS) P**
  - Μετονομασία Γνωρισμάτων
    - Π.χ., SELECT **PID AS PERSON_ID** FROM PERSON
  - Ταυτόχρονη Μετονομασία Πινάκων και Γνωρισμάτων
    - Π.χ., SELECT **P.PID AS PERSON_ID** FROM **PERSON (AS) P**

Ταξινόμηση (ORDER BY)

- Η ταξινόμηση επιτυγχάνεται με το **(ORDER BY)**
- Επιτρέπεται ταξινόμηση σε **αύξουσα (ASCENDING-ASC)** και **φθίνουσα (DESCENDING-DESC)** σειρά
- Η εξ' ορισμού ταξινόμηση είναι η ASC
- Αν δεν δηλωθεί το ORDER BY τότε η ταξινόμηση γίνεται βάση του πρωτεύων κλειδιού
- **Παραδείγματα (SELECT * FROM PERSON …)**
  - **Ένα γνώρισμα, μία σειρά (… ORDER BY ENAME)**
  - **Πολλά γνωρίσματα, πολλές σειρές (… ORDER BY DNO DESC, ENAME ASC)**

E: EMPLOYEE

SSN	ENAME	GENDER	DNO
1	Andreas	M	AC
2	Kostas	M	AC
3	Maria	F	CS
4	Eleni	F	CS
5	Victoria	F	MAS

E: EMPLOYEE

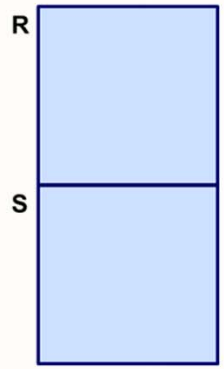
SSN	ENAME	GENDER	DNO
1	Andreas	M	AC
4	Eleni	F	CS
2	Kostas	M	AC
3	Maria	F	CS
5	Victoria	F	MAS

E: EMPLOYEE

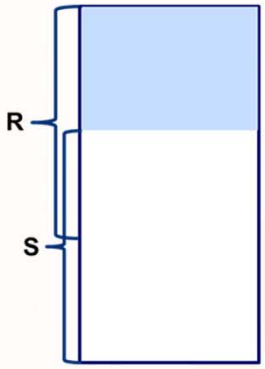
SSN	ENAME	GENDER	DNO
5	Victoria	F	MAS
4	Eleni	F	CS
3	Maria	F	CS
1	Andreas	M	AC
2	Kostas	M	AC

Πράξεις Συνόλων σε SQL (SQL Set Operations)

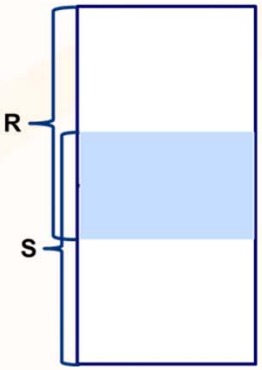
- **Είσοδος: Δύο Πίνακες**
- **Έξοδος: Ένας καινούριος Πίνακας**
- Η SQL παρόμοια με την σχεσιακή άλγεβρα υποστηρίζει τις ακόλουθες πράξεις σε σύνολα
  - Ένωση/Union: **UNION (∪)**
  - Αφαίρεση/ Set Difference: **EXCEPT, MINUS (-)**
  - Τομή/Intersection: **INTERSECTION** (∩)
  - Καρτεσιανό Γινόμενο/ Cartesian Product: **, CROSS JOIN (x)**



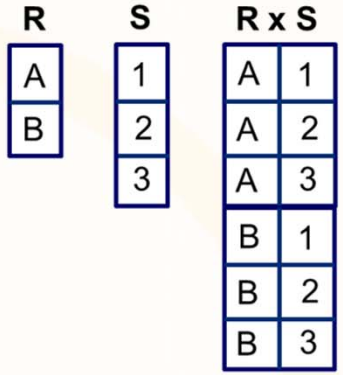
Ένωση/Union (∪)



Αφαίρεση/
Set Difference (–)



Τομή/Intersection (∩)



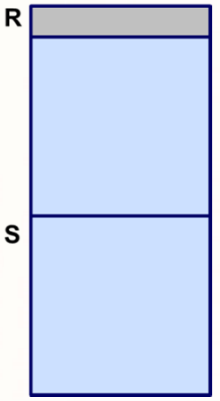
Καρτεσιανό Γινόμενο/
Cartesian Product (x)

Πράξεις Συνόλων σε SQL

- Η SQL:99 υλοποιεί αρκετές **πράξεις συνόλων** τις οποίες ορίσαμε στα πλαίσια της Σχεσιακής Άλγεβρας.
- Συγκεκριμένα, υποστηρίζονται οι ακόλουθες πράξεις:
  - **Ένωση: UNION [ALL]**
  - **Τομή: INTERSECT**
  - **Διαφορά Συνόλων: MINUS/EXCEPT/-**
- Προϋποθέτουν ότι τα σύνολα είναι **i) συμβατά-προς-τον-τύπο και ii) η διάταξη των γνωρισμάτων είναι η ίδια** (δεν χρειάζεται να έχουν το ίδιο όνομα)
- **Διαφορά Πράξεων Συνόλων από άλλες πράξεις SQL:**
  - Τα αποτελέσματα είναι **ΣΥΝΟΛΑ** όχι **ΠΟΛΥΣΥΝΟΛΑ** (συνεπώς δεν υπάρχουν διπλότυπα στις πράξεις αυτές)
  - Θα δούμε πως παράγονται Πολυσύνολα σε λίγο.

Παράδειγμα: **Ένωση/Union (∪)**

Παράδειγμα:




```
SELECT SID, SNAME FROM STUDENT
UNION
SELECT TID, TNAME FROM TEACHER
```

STUDENT

SID	SNAME
1	Andreas
2	Kostas
3	Maria
4	Eleni

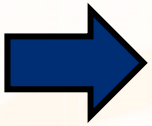
r(STUDENT)=4

∪

TEACHER

TID	TNAME
1	Andreas
2	Anna

r(TEACHER)=2



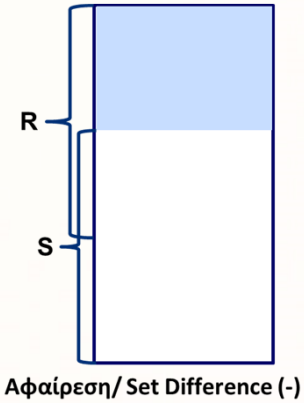
STUDENT_TEACHER

SID	SNAME
1	Andreas
2	Kostas
3	Maria
4	Eleni
2	Anna

r(STUDENT_TEACHER)=???

Παράδειγμα: **Αφαίρεση/Set Difference (-)**

Παράδειγμα:



Αφαίρεση/ Set Difference (-)

SELECT SID, SNAME FROM STUDENT

MINUS/EXCEPT/-

SELECT TID, TNAME FROM TEACHER

STUDENT

SID	SNAME
1	Andreas
2	Kostas
3	Maria
4	Eleni

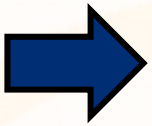
r(STUDENT)=4

—

TEACHER

TID	TNAME
1	Andreas
2	Anna

r(TEACHER)=2



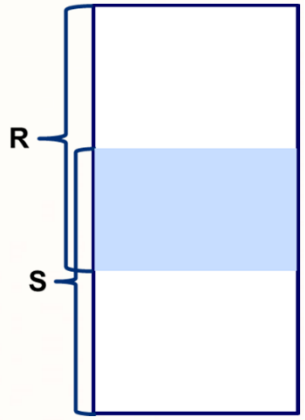
STUDENT_TEACHER

SID	SNAME
2	Kostas
3	Maria
4	Eleni

r(STUDENT_TEACHER)=3

Παράδειγμα: **Τομή/Intersection (∩)**

Παράδειγμα:



SELECT SID, SNAME FROM STUDENT

INTERSECTION

SELECT TID, TNAME FROM TEACHER

STUDENT

SID	SNAME
1	Andreas
2	Kostas
3	Maria
4	Eleni

r(STUDENT)=4

∩

TEACHER

TID	TNAME
1	Andreas
2	Anna

r(TEACHER)=2

STUDENT_TEACHER

SID	SNAME
1	Andreas

r(STUDENT_TEACHER)=???

Πράξεις Συνόλων σε SQL

- Για να επιστραφούν αποτελέσματα Πολυσυνόλων, αντί αποτελέσματα συνόλων, (δηλ., δεν διαγράφονται τα διπλότυπα από ένα αποτέλεσμα) μπορεί να γίνει χρήση των ακόλουθων εντολών:
  - **UNION ALL,** υποστηρίζεται και σε TSQL

    Π.χ., {(1,Pet), (2,Cat)} **UNION ALL** {(2,Cat),(1, Pet)}

    Επιστρέφει: { (1,Pet), (2,Cat), (2,Cat), (1, Pet) }
  - **INTERSECT ALL (δεν υλοποιείται σε TSQL)**

    Π.χ., {(1,Pet), (2,Cat)} **INTERSECT ALL** {(1, Pet)}

    Επιστρέφει: { (1,Pet), (1, Pet) }
  - **EXCEPT ALL (δεν υλοποιείται σε TSQL)**

    Π.χ., {(1,Pet), (2,Cat) , (2,Cat)} **EXCEPT ALL** {(1, Pet)}

    Επιστρέφει:{(2,Cat), (2,Cat) }

Άλλες Πράξεις Συνόλων σε SQL

- Σημειώστε ότι εάν ένας τελεστής **δεν υλοποιείται στην SQL** τότε μπορεί να **υλοποιηθεί με χρήση βασικών τελεστών.**
- **Παραδείγματα**
  - Συμμετρική Διαφορά, $R \oplus S = (R - S) \cup (S - R)$
  - Διαίρεση, $R \div S = \pi_A (R) - \pi_A ((\pi_A (R) \times S) - R)$

- Στην συνέχεια θα δούμε ότι υπάρχουν **πολλές άλλες πράξεις σύγκρισης με σύνολα** που χρησιμοποιούνται για διατύπωση επερωτήσεων σε SQL
  - **IN, ANY, ALL, CONTAINS, EXISTS, κτλ.**

Παραδείγματα/Ασκήσεις

E: EMPLOYEE

SSN	NAME	DOB	GENDER	DNO
1	Andreas	1/1/80	M	1
2	Kostas	1/1/85	M	1
3	Maria	4/3/83	F	1
4	Eleni	6/7/85	F	2
5	Victoria	27/2/84	F	3

D: DEPARTMENT

DNO	DNAME
1	Finance
2	Audit
3	Management

- **Τύπωσε τα ονόματα των Τμημάτων (σε φθίνουσα σειρα)**
- **Τύπωσε όνομα και dno των υπάλληλων που είναι γυναίκες**
- **Τύπωσε όλα τα ονόματα της βάσης δεδομένων**
- **Τύπωσε τα ονόμ. τμήμ. των υπαλλήλων που έχουν γεννηθεί το 85**

- **Τύπωσε όλα τα ονόματα των υπαλλήλων που έχουν SSN που υπάρχει σαν αριθμός τμήματος**