

# ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

SQL

Φροντιστήριο 6<sup>ο</sup>





# Η ΓΛΩΣΣΑ SQL

- Δομικές μονάδες μίας τυπικής εκδοχής της **SQL**
  - **Γλώσσα ορισμού δεδομένων (Data Definition Language - DDL)**
    - Εντολές που μας επιτρέπουν να υλοποιήσουμε σχέσεις και γενικά όλη τη δομή μιας ΒΔ - ορισμός, δημιουργία, τροποποίηση, διαγραφή σχήματος
  - **Γλώσσα χειρισμού δεδομένων (Data Manipulation Language - DML)**
    - Διαχείριση των δεδομένων της εφαρμογής, όπως την εισαγωγή, διαγραφή, ανάκτηση και τροποποίηση δεδομένων → **ΓΛΩΣΣΑ ΕΡΩΤΗΜΑΤΩΝ**
  - **Ορισμός όψεων της βάσης (View definition):** δημιουργία εικονικών πινάκων (**views-virtual tables**) που περιέχουν δεδομένα από έναν ή περισσότερους πίνακες της βάσης
  - **Ορισμός εξουσιοδοτήσεων (Authorization):** δημιουργία ομάδων χρηστών και απόδοση διαφορετικών δικαιωμάτων πρόσβασης σε καθένα
  - **Διαχείρισης ακεραιότητας (Integrity):** επιτρέπει το λεπτομερή έλεγχο των δεδομένων που καταχωρούνται ώστε να μην παραβιάζονται οι **κανόνες** ακεραιότητας, οι οποίοι όταν τηρούνται, απομακρύνουν τον κίνδυνο **ασυνεπών δεδομένων**

# ΓΛΩΣΣΑ ΟΡΙΣΜΟΥ ΔΕΔΟΜΕΝΩΝ

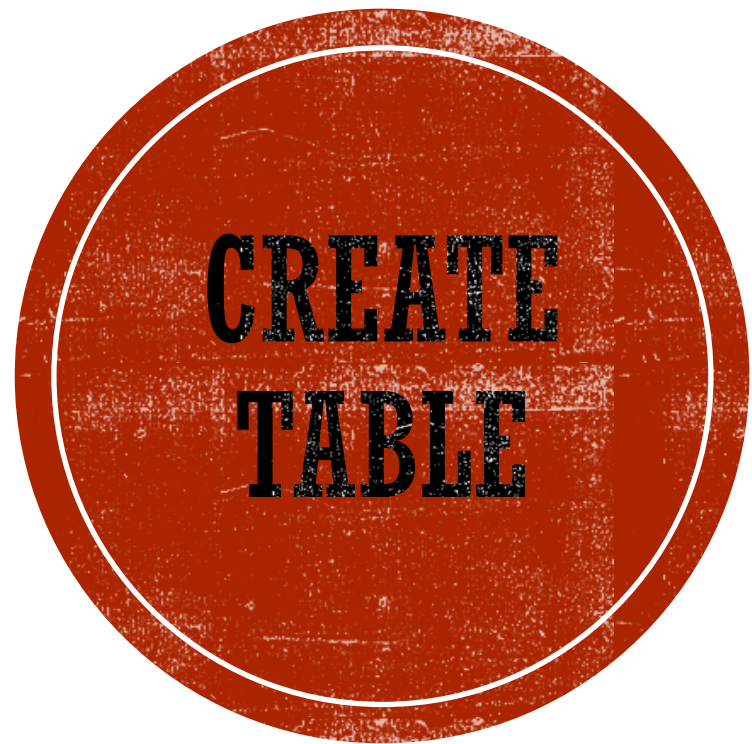
- Η διαχείριση μέσω της **DDL** περιλαμβάνει τον **ορισμό**, τη **μεταβολή της δομής** και τη **διαγραφή** των παρακάτω αντικειμένων:
  - **Πίνακες (Tables)**: το δομικό χαρακτηριστικό μιας σχεσιακής ΒΔ, καθώς περιέχουν τα δεδομένα που καταχωρούνται σε αυτή
  - **Όψεις (Views)**: εικονικοί πίνακες (**virtual tables**) οι οποίοι περιέχουν δεδομένα από έναν ή περισσότερους πίνακες της βάσης
  - **Δείκτες (Indices)**: ειδικές δομές δεδομένων που επιταχύνουν τη διαδικασία της αναζήτησης πληροφοριών από τη βάση



# ΕΝΤΟΛΕΣ ΟΡΙΣΜΟΥ ΔΕΔΟΜΕΝΩΝ

- **CREATE TABLE** (δημιουργία πίνακα)
- **CREATE INDEX** (δημιουργία ευρετηρίου)
- **CREATE VIEW** (δημιουργία όψης)
  
- **UPDATE TABLE** (ενημέρωση εγγραφών)
- **INSERT INTO** (εισαγωγή εγγραφών/πλειάδων)
  
- **DROP TABLE** (διαγραφή πίνακα)
- **DROP INDEX** (διαγραφή ευρετηρίου)
- **DROP VIEW** (διαγραφή όψης)
  
- **ALTER** (τροποποίηση της δομής των αντικειμένων της ΒΔ, δηλ. πινάκων, ευρετηρίων, όψεων)





CREATE TABLE πίνακας (όνομα\_στήλης τύπος\_στήλης, ...);

```
CREATE TABLE EMPLOYEE (  
    FNAME      VARCHAR (15)    NOT NULL,  
    MINIT      CHAR (1),  
    LNAME      VARCHAR (15)    NOT NULL,  
    SSN        CHAR (9)        NOT NULL,  
    BDATE      DATE,  
    ADDRESS    VARCHAR (30),  
    SEX        VARCHAR (30),  
    SALARY     INTEGER,  
    SUPERSSN   CHAR (9),  
    DNO        INTEGER ,  
    PRIMARY KEY (SSN) );
```



# ΠΑΡΑΔΕΙΓΜΑ DROP - ALTER

## DROP TABLE DEPENDENTS

*Διαγράφει τον πίνακα DEPENDENTS και όλα τα δεδομένα*

**ALTER TABLE** table\_name **ADD** column\_name datatype

**ALTER TABLE** EMPLOYEE **ADD** JOB VARCHAR(12)

*προσθέτει στον πίνακα EMPLOYEE το πεδίο JOB που είναι συμβολοσειρά με μέγιστο μήκος ίσο με 12 χαρακτήρες.*

**ALTER TABLE** table\_name **DROP COLUMN** column\_name

**ALTER TABLE** table\_name **MODIFY** column\_name datatype

**ALTER TABLE** table\_name **RENAME COLUMN** old\_name to new\_name



# ΟΨΕΙΣ (VIEWS)

- **Όψη (view):** ένας απλός πίνακας που προκύπτει από το συνδυασμό των πεδίων ενός ή περισσότερων πινάκων, οι οποίοι μπορεί να είναι είτε βασικοί πίνακες ή άλλες όψεις (εικονικοί πίνακες)
- Οι βασικοί πίνακες από τους οποίους δημιουργείται μια όψη, αναφέρονται και ως **πίνακες ορισμού της όψης (defining tables)**.
- Για να δημιουργήσουμε μια όψη χρησιμοποιούμε την εντολή **CREATE VIEW**
- Για να διαγράψουμε μια όψη χρησιμοποιούμε την εντολή **DROP VIEW**
- Παρέχουν ένα μηχανισμό απόκρυψης συγκεκριμένων δεδομένων από συγκεκριμένους χρήστες, δημιουργώντας έναν μόνιμο/εικονικό πίνακα
- Παρέχουν ασφάλεια - το σχήμα της βάσης αποκρύπτεται
- Μπορούν να βελτιώσουν την απόδοση της βάσης
  - Συχνά χρησιμοποιούμενα ερωτήματα που περιλαμβάνουν συνενώσεις υπολογίζονται μία φορά – χρησιμοποιούνται πολλές
- Εντολή δημιουργίας:

**create view v as <query expression>**



Όνομα της  
Όψης



Κάθε νόμιμη  
έκφραση



```
CREATE VIEW WORKS_ON1  
  AS SELECT FNAME, LNAME, PNAME, HOURS  
  FROM EMPLOYEE, PROJECT, WORKS_ON  
  WHERE SSN = ESSN  
  AND PNO = PNUMBER
```

```
CREATE VIEW DEPT_INFO (DNAME, EMP_NO, AVG_SAL)  
  AS SELECT DNAME, COUNT (*), AVERAGE (SALARY)  
  FROM DEPARTMENT, EMPLOYEE  
  WHERE DNUMBER=DNO  
  GROUP BY DNAME
```

VIEW -  
ΠΑΡΑΔΕΙΓΜΑ





# ΔΙΑΧΕΙΡΙΣΗ ΔΕΙΚΤΩΝ I

- **Δείκτης (index):** μια δομή δεδομένων που αποθηκεύεται σε ειδικά αρχεία της βάσης (index files) και έχει στόχο να επιταχύνει τη διαδικασία αναζήτησης πληροφορίας από τους πίνακες της ΒΔ
- Οι δείκτες συνήθως ορίζονται για συγκεκριμένα πεδία αυτών των πινάκων (indexing fields) και για κάθε τιμή των εν λόγω πεδίων, αποθηκεύουν ένα σύνολο από pointers που δείχνουν τα δεδομένα της κάθε εγγραφής.
- Τα αρχεία δεικτών είναι αρκετά μικρότερα σε μέγεθος σε σχέση με τα αρχεία δεδομένων και η διαδικασία αναζήτησης της πληροφορίας γίνεται μέσω γνωστών τεχνικών αναζήτησης, όπως η **δυναμική αναζήτηση**
  - [http://en.wikipedia.org/wiki/Binary\\_search\\_algorithm](http://en.wikipedia.org/wiki/Binary_search_algorithm)
- Η δημιουργία ενός **δείκτη** γίνεται χρησιμοποιώντας την εντολή **CREATE INDEX**
- Η διαγραφή ενός δείκτη γίνεται με την εντολή **DROP INDEX**



```
CREATE INDEX LNAME_INDEX  
ON EMPLOYEE (LNAME)
```

```
CREATE INDEX NAMES_INDEX  
ON EMPLOYEE (LNAME ASC, FNAME DESC, MINIT)
```

```
CREATE UNIQUE INDEX SSN_INDEX  
ON EMPLOYEE (SSN)
```

```
DROP INDEX LNAME_INDEX
```

Δεν επιτρέπονται  
διπλότυπες τιμές

ΠΑΡΑΔΕΙΓΜΑΤΑ  
INDEX



# ΓΛΩΣΣΑ ΧΕΙΡΙΣΜΟΥ ΔΕΔΟΜΕΝΩΝ

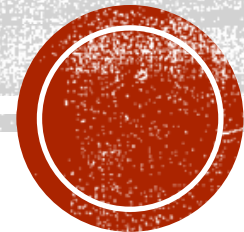
- Η γλώσσα χειρισμού δεδομένων (**Data Manipulation Language, DML**), επιτρέπει τη διαχείριση των δεδομένων των πινάκων της ΒΔ, και πιο συγκεκριμένα, την **εισαγωγή, διαγραφή, και τροποποίηση των εγγραφών** των πινάκων.
- Μέσω της DML έχουμε τη δυνατότητα να ανακτήσουμε από τους πίνακες, δεδομένα, τα οποία πληρούν κάποια συγκεκριμένα κριτήρια.
- Η DML **δεν ελέγχει** αν οι εγγραφές που επηρεάζονται ικανοποιούν τους κανόνες ακεραιότητας του λογικού σχεδιασμού που αποτυπώνονται στο σχήμα της ΒΔ



# SELECT — FROM - WHERE

- **SELECT**: επιλογή δεδομένων από τη ΒΔ
- **SELECT DISTINCT**: Απαλείφει διπλότυπες τιμές
- **FROM**:
  - Ορίζει από πιο σχήμα θα επιστραφούν οι εγγραφές
  - Μπορεί να αναφέρεται σε ένα σχήμα ή σε συνδυασμό >2 σχημάτων
- **WHERE**: Επιστρέφει μόνο τις πλειάδες για τις οποίες η συνθήκη P είναι αληθής (φιλτράρισμα εγγραφών)
  - Λογικοί σύνδεσμοι: **and**, **or**, **not**
  - Τελεστές σύγκρισης: **<**, **=**, **>**, **>=**, **<=**, **<>**, **between**, **not between**
  - Έλεγχος κενών τιμών: **is NULL**, **is not NULL**

```
SELECT <attribute list>  
FROM <table list>  
WHERE <condition>
```





# Παράδειγμα 1<sup>ο</sup> - ΠΡΕΣΒΕΙΕΣ

TABLE: EMBASSY

ID_Embassy	Embassy_Name	District	Street	Num	Country_ID	City_ID
1	CHINESE EMBASSY IN ATHENS	PSYCHIKO	KRINON STR	2A	4	1
2	BRITISH EMBASSY IN ATHENS	ATHENS	PLOUTARHOU STR	1	5	1
3	CYPRIAN EMBASSY IN ATHENS	ATHENS	HERODOTOU STR	16	6	1
4	RUSSIAN EMBASSY IN ATHENS	PSYCHIKO	NIK LYTRA STR	28	7	1
5	FRENCH CONSULATE IN THESSALONIKI	THESSALONIKI	MAKENZY KING STR		3	3
6	FINLAND CONSULATE IN THESSALONIKI	ORAIOKASTRO	ORAIOKASTROU	7KM	2	3
7	GREEK EMBASSY IN PARIS	PARIS	RUE AUGUSTE VACQUERIE	17	1	4
8	GREEK CONSULATE IN MASELLIE	MASELLIE	RUE RIGNAN	17	1	5
9	GREEK EMBASSY IN NICOSIA	NICOSIA	VYRONOS AV	8-10	1	6
10	GREEK EMBASSY IN MOSCOW	MOSCOW	LEONTIEFSKI PER.	4	1	7
11	GREEK COSULATE IN ST.PETERSBURG	ST.PETERSBURG	MIKHAILOVSKAYA ULITSA-HOTEL EUROPE	1/7	1	8
12	GREEK COSULATE IN NOVOROSSISK	NOVOROSSISK	ICAeva-HOTEL NOVOROSSISK	2	1	9
13	GREEK ENBASSY IN LONDON	W11	HOLLAND PARK	1A	1	2
14	BRITISH EMBASSY IN MOSCOW	MOSCOW	SMOLENSKAYA NABEREZHNYA	10	5	7
15	BRITISH EMBASSY IN HELSINKI	HELSINKI	ITAINEN PUISTOTIE	17	5	9
16	RUSSIAN EMBASSY IN LONDON	W8 4QS	KENSINGTON PALACE GARDENS,	5	7	2
17	CHINESE EMBASSY IN LONDON	W1B 1JL	ORTLAND PLACE	49-51	4	2
18	FINLAND EMBASSY IN LONDON	SW1X 8HW	CHESHAM PLACE	38	2	2
19	FRENCH EMBASSY IN LONDON	SW1X 7JT	KNIGHTSBRIDGE		3	2

TABLE: AMBASSADOR

ID_Embassy	ID	NAME	SURNAME
8	1	NIKOLAOS	FOKAS
10	2	MARIA	KOUFOPOULOU
11	3	IOANNHS	KARAVATOS
12	4	IOANNIS	KOMNINOS
NULL	5	GEORGIOS	IOANNIDIS
13	6	EMMANOUIL	ATHANASIADIS
NULL	7	ELENI	PALAIOLOGOU

TABLE: CITY

ID_City	City_Name	IDCountry
1	ATHENS	1
2	LONDON	5
3	THESSALONIKI	1
4	PARIS	3
5	MASELLIE	3
6	LEYKOSIA	6
7	MOSCOW	7
8	ST.PETERSBURG	7
9	NOVOROSSISK	7
10	HELSINKI	2

TABLE: COUNTRY

ID_COUNTRY	COUNTRY_NAME
1	GREECE
2	FINLAND
3	FRENCH
4	CHINA
5	GREAT BRITAIN
6	CYPRUS
7	RUSSIA
8	HUNGARY





**Ερώτηση 1<sup>η</sup>** : Επιλέξετε τα στοιχεία των χωρών με ID μικρότερο από 3

```
SELECT * FROM COUNTRY WHERE ID_COUNTRY <3
```

Επιστρέφονται όλα τα γνωρίσματα

TABLE: COUNTRY

ID_COUNTRY	COUNTRY_NAME
1	GREECE
2	FINLAND

**Ερώτηση 2<sup>η</sup>** : Ποιο το όνομα της πρεσβείας και ποια η αντίστοιχη οδός για κάθε πρεσβεία που βρίσκεται στην περιοχή Ψυχικό

```
SELECT EMBASSY_NAME, STREET FROM EMBASSY WHERE DISCTRICT = "PSYCHIKO"
```

TABLE: EMBASSY

EMBASSY_NAME	STREET
CHINESE EMBASSY IN ATHENS	KRINON STR
RUSSIAN EMBASSY IN ATHENS	NIK.LYTRA STR



**Ερώτηση 3<sup>η</sup> : Επιλέξτε τα στοιχεία των χωρών με ID μικρότερο ίσο από 5 και μεγαλύτερο ίσο από 3**

```
SELECT * FROM COUNTRY WHERE ID_COUNTRY  
>=3 and ID_COUNTRY <=5
```

ή

```
SELECT * FROM COUNTRY WHERE ID_COUNTRY  
BETWEEN 3 AND 5
```

TABLE: COUNTRY

<u>ID_COUNTRY</u>	COUNTRY_NAME
3	FRENCH
4	CHINA

**Ερώτηση 4<sup>η</sup> : Επιλέξτε τα στοιχεία των χωρών με ID μεγαλύτερο από 5 ή μικρότερο από 3**

```
SELECT * FROM COUNTRY WHERE ID_COUNTRY  
>5 or ID_COUNTRY <3
```

ή

```
SELECT * FROM COUNTRY WHERE ID_COUNTRY  
NOT BETWEEN 3 AND 5
```

TABLE: COUNTRY

<u>ID_COUNTRY</u>	COUNTRY_NAME
1	GREECE
2	FINLAND
6	CYPRUS
7	RUSSIA
8	HUNGARY



Ερώτηση 5<sup>η</sup> :Επιλέξτε όλα τα στοιχεία των  
εν ενεργεία πρεσβευτών

```
SELECT * FROM AMBASSADOR WHERE  
ID_EMBASSY IS NOT NULL
```

ID_Embassy	ID	NAME	SURNAME
8	1	NIKOLAOS	FOKAS
10	2	MARIA	KOUFOPOULOU
11	3	IOANNIS	KARAVATOS
12	4	IOANNIS	KOMNINOS
13	6	EMMANOUIL	ATHANASIADIS



# Σειρά εμφάνισης Εγγραφών

- Order by attr-list

- `SELECT * FROM AMBASSADOR ORDER BY NAME desc`

- `SELECT * FROM AMBASSADOR ORDER BY NAME asc, ID desc;`

- Asc : κατά αύξουσα σειρά

- Desc: κατά φθίνουσα σειρά

- Ταξινόμηση: Χρονοβόρα πράξη

**Ερώτηση 6<sup>η</sup> :Επιλέξτε όλα τα στοιχεία των  
πρεσβευτών ταξινομημένα με βάση το  
όνομα κατά φθίνουσα σειρά**

•`SELECT * FROM AMBASSADOR ORDER BY NAME desc`

ID_Embassy	<u>ID</u>	NAME	SURNAME
8	1	NIKOLAOS	FOKAS
10	2	MARIA	KOUFOPOULOU
11	3	IOANNIS	KARAVATOS
12	4	IOANNIS	KOMNINOS
NULL	5	GEORGIOS	IOANNIDIS
13	6	EMMANOUIL	ATHANASIADIS
NULL	7	ELENH	PALAIOLOGOU





Ερώτηση 7<sup>η</sup> :Επιλέξτε όλα τα στοιχεία των  
πρεσβευτών ταξινομημένα με βάση το όνομα κατά  
φθίνουσα σειρά και με βάση το ID κατά φθίνουσα  
σειρά επίσης



```
•SELECT * FROM AMBASSADOR ORDER BY NAME desc, ID  
desc;
```

ID_Embassy	ID	NAME	SURNAME
8	1	NIKOLAOS	FOKAS
10	2	MARIA	KOUFOPOULOU
12	4	IOANNIS	KOMNINOS
11	3	IOANNIS	KARAVATOS
NULL	5	GEORGIOS	IOANNIDIS
13	6	EMMANOUIL	ATHANASIADIS
NULL	7	ELENI	PALAIOLOGOU

# ΠΡΑΞΕΙΣ ΜΕ ΣΥΜΒΟΛΟΣΕΙΡΕΣ

- `SELECT * FROM AMBASSADOR WHERE NAME = "GIANNIS"`
- `SELECT * FROM AMBASSADOR WHERE NAME like "GIANNIS"`
- `SELECT * FROM AMBASSADOR WHERE NAME not like "E%"`
- `SELECT * FROM AMBASSADOR WHERE NAME like "%S"`
- `SELECT * FROM AMBASSADOR WHERE NAME like "_O%S"`
- `SELECT * FROM AMBASSADOR WHERE NAME like "%I_N_S"`

Αναζήτηση ενός  
καθορισμένου  
μοτίβου

Βρες οποιαδήποτε τιμή έχει  
"O" στη δεύτερη θέση και  
τελειώνει σε "S"

%: ταιριάζει οποιαδήποτε συμβολοσειρά (n χαρακτήρες)

\_: ταιριάζει οποιοδήποτε χαρακτήρα (1 χαρακτήρας)

Η SQL υποστηρίζει ποικιλία πράξεων σε αλφαριθμητικές παραστάσεις όπως

- Η συνένωση (using "||")
- Μετατροπή σε converting από κεφαλαία σε μικρά και το αντίστροφο
- Εξαγωγή υπό-αλφαριθμητικών
- Υπολογισμός μήκους αλφαριθμητικού κτλ



# Cross Join

(υλοποίηση καρτεσιανού γινομένου)



Ερώτηση 9<sup>η</sup> : Επιλέξτε τα στοιχεία των πόλεων και των χωρών

**SELECT \* FROM COUNTRY, CITY**

(Το ερώτημα αυτό δεν έχει νόημα στον πραγματικό κόσμο!!! – Γιατί???)  
Εμφανίζονται μόνο τα 10 πρώτα αποτελέσματα...

TABLE: COUNTRY x CITY

ID_COUNTRY	COUNTRY_NAME	ID_CITY	CITY_NAME	IDCOUNTRY
1	GREECE	1	ATHENS	1
2	FINLAND	1	ATHENS	1
3	FRENCH	1	ATHENS	1
4	CHINA	1	ATHENS	1
5	GREAT BRITAIN	1	ATHENS	1
6	CYPRUS	1	ATHENS	1
7	RUSSIA	1	ATHENS	1
8	HUNGARY	1	ATHENS	1
1	GREECE	2	LONDON	5
2	FINLAND	2	LONDON	5

TABLE: COUNTRY

ID_COUNTRY	COUNTRY_NAME
1	GREECE
2	FINLAND
3	FRENCH
4	CHINA
5	GREAT BRITAIN
6	CYPRUS
7	RUSSIA
8	HUNGARY

TABLE: CITY

ID_City	City_Name	IDCountry
1	ATHENS	1
2	LONDON	5
3	THESSALONIKI	1
4	PARIS	3
5	MASSELLIE	3
6	LEYKOSIA	6
7	MOSCOW	7
8	ST.PETERSBURG	7
9	NOVOROSSISK	7
10	HELSINKI	2

# Equi-Join ΣΥΝΕΝΩΣΗ ΙΣΟΤΗΤΑΣ

- Χρησιμοποιείται
  - για να συνένωση πλειάδων για τις οποίες είναι αληθής η συνθήκη ισότητας.
- Χρησιμοποιούμε την συνθήκη **WHERE**
- Η συνθήκη ισότητας
  - περιλαμβάνει ένα **ζεύγος** από το ξένο κλειδί σχέσης A, το οποίο δείχνει στο πρωτεύον κλειδί μιας σχέσης B **τότε** επιστρέφονται εγγραφές που έχουν νόημα στο πραγματικό κόσμο
- Μπορούμε να εμπλουτίσουμε την συνθήκη με επιπλέον συνθήκη με τους λογικούς τελεστές (and, or, not κτλ)
- Επιστρέφονται όλα τα γνωρίσματα των επιμέρους σχέσεων που συμμετέχουν στην συνένωση
  - Εάν δεν περιορίσουμε την λίστα των γνωρισμάτων που θα επιστραφούν





**Ερώτηση 10<sup>η</sup> :Επιλέξτε όλα τα στοιχεία των πόλεων και των χωρών στις οποίες βρίσκονται**



```
SELECT * FROM COUNTRY, CITY WHERE  
CITY.IDCOUNTRY=COUNTRY.ID_COUNTRY
```

Το γνώρισμα ID\_COUNTRY της σχέσης  
COUNTRY

Εξισώνουμε τα  
δύο πεδία

TABLE: COUNTRY x CITY		CITY		
COUNTRY		CITY		
ID_COUNTRY	COUNTRY_NAME	ID_CITY	CITY_NAME	IDCOUNTRY
1	GREECE	1	ATHENS	1
1	GREECE	3	THESSALONIKI	1
2	FINLAND	10	HELSINKI	2
3	FRENCH	4	PARIS	3
3	FRENCH	5	MASSELLIE	3
5	GREAT BRITAIN	2	LONDON	5
6	CYPRUS	6	LEYKOSIA	6
7	RUSSIA	7	MOSCOW	7
7	RUSSIA	8	ST. PETERSBURG	7
7	RUSSIA	9	NOVOROSSISK	7



- Το προηγούμενο ερώτημα ισοδύναμο με το:

```
SELECT * FROM COUNTRY, CITY WHERE  
IDCOUNTRY=ID_COUNTRY
```

Τα γνωρίσματα δεν έχουν ακριβώς το ίδιο όνομα.



## Ερώτηση 11<sup>η</sup> :Επιλέξτε το όνομα της πόλης και το όνομα της χώρας στην οποία βρίσκονται



```
SELECT COUNTRY_NAME AS COUNTRY, CITY_NAME AS CITY  
FROM COUNTRY, CITY WHERE CITY.IDCOUNTRY=COUNTRY.ID_COUNTRY
```

Μετονομάζουμε το COUNTRY\_NAME σε  
COUNTRY (ΧΡΗΣΗ ΨΕΥΔΩΝΥΜΩΝ)

TABLE: COUNTRY x CITY

COUNTRY	CITY
GREECE	ATHENS
GREECE	THESSALONIKI
FINLAND	HELSINKI
FRENCH	PARIS
FRENCH	MASSELLIE
GREAT BRITAIN	LONDON
CYPRUS	LEYKOSIA
RUSSIA	MOSCOW
RUSSIA	ST. PETERSBURG
RUSSIA	NOVOROSSISK

Μπορούμε να συνενώσουμε και  
παραπάνω από δύο σχέσεις ...

Πως?



## Ερώτηση 12<sup>η</sup> :Να προβληθεί το όνομα της κάθε πρεσβείας που βρίσκεται στην πόλη “ATHENS” και το όνομα της χώρας που αντιπροσωπεύει η πρεσβεία



```
SELECT COUNTRY_NAME AS COUNTRY,  
       EMBASSY_NAME AS EMBASSY,  
       CITY_NAME AS inCITY  
FROM   COUNTRY AS C1,  
       EMBASSY AS E,  
       CITY AS C2  
WHERE  C1.ID_COUNTRY=E.COUNTRY_ID  
AND    E.CITY_ID=C2.ID_CITY  
AND    CITY_NAME='ATHENS' ;
```

Μετονομάζουμε το ΣΧΗΜΑ COUNTRY σε C1

- Για μεγαλύτερη ευκολία
- Για πρακτικούς λόγους

TABLE: COUNTRY x EMBASSY x CITY

COUNTRY	EMBASSY	inCITY
CHINA	CHINESE EMBASSY IN ATHENS	ATHENS
GREAT BRITAIN	BRITISH EMBASSY IN ATHENS	ATHENS
CYPRUS	CYPRIAN EMBASSY IN ATHENS	ATHENS
RUSSIA	RUSSIAN EMBASSY IN ATHENS	ATHENS

- αναφερόμαστε παραπάνω από μία φορά στον ίδιο πίνακα
- Αναδρομικές σχέσεις



## Theta join



- Παράγει όλους τους συνδυασμούς των εγγραφών από τις σχέσεις R1 και R2 που ικανοποιούν την συνθήκη συνένωσης
- Η συνθήκη επιλογής δεν είναι αποκλειστικά ισότητα
- Μπορεί να χρησιμοποιηθεί οποιοσδήποτε από τους παρακάτω τελεστές:  
 $<, =, >, \geq, \leq, <>$





**Ερώτηση 13<sup>η</sup>** : Να προβληθεί το όνομα κάθε πρεσβευτή με ID>5 σε συνδυασμό με το όνομα κάθε πρεσβείας που βρίσκονται σε οδό με νούμερο (NUM) μικρότερο από τον κωδικό του αντίστοιχου πρεσβευτή



```
SELECT NAME, EMBASSY_NAME AS  
EMBASSY  
FROM EMBASSY,AMBASSADOR  
WHERE  
ID>num  
AND ID>5;
```

TABLE: EMBASSY x AMBASSADOR

NAME	EMBASSY
EMMANOHL	CHINESE EMBASSY IN ATHENS
ELENH	CHINESE EMBASSY IN ATHENS
EMMANOHL	BRITISH EMBASSY IN ATHENS
ELENH	BRITISH EMBASSY IN ATHENS
EMMANOHL	FRENCH CONSULATE IN THESSALONIKI
ELENH	FRENCH CONSULATE IN THESSALONIKI
EMMANOHL	GREEK EMBASSY IN MOSCOW
ELENH	GREEK EMBASSY IN MOSCOW
EMMANOHL	GREEK COSULATE IN ST.PETERSBURG
ELENH	GREEK COSULATE IN ST.PETERSBURG
EMMANOHL	GREEK COSULATE IN NOVOROSSISK
ELENH	GREEK COSULATE IN NOVOROSSISK
EMMANOHL	GREEK ENBASSY IN LONDON
ELENH	GREEK ENBASSY IN LONDON
EMMANOHL	RUSSIAN EMBASSY IN LONDON
ELENH	RUSSIAN EMBASSY IN LONDON
EMMANOHL	FRENCH EMBASSY IN LONDON
ELENH	FRENCH EMBASSY IN LONDON



## Inner join

- Ισοδύναμο με το EQUI JOIN
- Περισσότερο αποδοτική προσέγγιση



## Ερώτηση 14<sup>η</sup> :Επιλέξτε το όνομα της πόλης και το όνομα της χώρας στην οποία βρίσκεται

```
SELECT COUNTRY_NAME AS COUNTRY, CITY_NAME AS CITY  
FROM  
  COUNTRY INNER JOIN CITY  
    ON CITY.IDCOUNTRY=COUNTRY.ID_COUNTRY
```

TABLE: COUNTRY x CITY

COUNTRY	CITY
GREECE	ATHENS
GREECE	THESSALONIKI
FINLAND	HELSINKI
FRENCH	PARIS
FRENCH	MASSELLIE
GREAT BRITAIN	LONDON
CYPRUS	LEYKOSIA
RUSSIA	MOSCOW
RUSSIA	ST. PETERSBURG
RUSSIA	NOVOROSSISK



# OUTER JOINS



- Επιστρέφονται και εγγραφές που στο γνώρισμα συνένωσης παίρνουν τιμή Null.
- LEFT OUTER JOIN:
  - Επιστρέφονται όλες οι εγγραφές του πίνακα «αριστερά» της συνένωσης (έστω A) και όλες εγγραφές του πίνακα «δεξιά» της συνένωσης (έστω Δ) για τις οποίες ισχύει  $A.join\_attribute = Δ.join\_attribute$
- RIGHT OUTER JOIN:
  - Αντίστοιχα
- FULL OUTER JOIN:
  - Επιστρέφονται όλες οι πλειάδες αριστερά και δεξιά της συνένωσης. Για όλες ισχύει  $A.join\_attribute = Δ.join\_attribute$  εμφανίζονται στην ίδια πλειάδα



**Ερώτηση 16<sup>η</sup> : Προβάλετε το όνομα και τον κωδικό όλων των πρεσβευτών καθώς και το όνομα της πρεσβείας στην οποία υπηρετεί έκαστος.**

```
SELECT ID,NAME,EMBASSY_NAME
FROM
AMBASSADOR LEFT OUTER JOIN EMBASSY
ON
AMBASSADOR.ID_EMBASSY = EMBASSY.ID_EMBASSY
```

TABLE: AMBASSADOR ⋈ EMBASSY

ID	NAME	EMBASSY_NAME
1	NIKOLAOS	GREEK CONSULATE IN MASELLIE
2	MARIA	GREEK EMBASSY IN MOSCOW
3	IOANNIS	GREEK COSULATE IN ST.PETERSBURG
4	IOANNIS	GREEK COSULATE IN NOVOROSSISK
5	GEORGIOS	NULL
6	EMMANOUEL	GREEK EMBASSY IN LONDON
7	ELENI	NULL





**Ερώτηση 17<sup>η</sup> :** Προβάλετε το όνομα της κάθε πρεσβείας και τον κωδικό και το όνομα όλων των πρεσβευτών που υπηρετούν σε αυτή, εφόσον η πληροφορία είναι διαθέσιμη

```
SELECT  
  ID,NAME,EMBASSY_NAME  
FROM  
  AMBASSADOR RIGHT OUTER  
  JOIN EMBASSY  
  ON  
  AMBASSADOR.ID EMBASSY  
  = EMBASSY.ID_ EMBASSY
```

TABLE: AMBASSADOR X EMBASSY

ID	NAME	EMBASSY_NAME
NULL	NULL	CHINESE EMBASSY IN ATHENS
NULL	NULL	BRITISH EMBASSY IN ATHENS
NULL	NULL	CYPRIAN EMBASSY IN ATHENS
NULL	NULL	RUSSIAN EMBASSY IN ATHENS
NULL	NULL	FRENCH CONSULATE IN THESSALONIKI
NULL	NULL	FINLAND CONSULATE IN THESSALONIKI
NULL	NULL	GREEK EMBASSY IN PARIS
1	NIKOLAOS	GREEK CONSULATE IN MASELLIE
NULL	NULL	GREEK EMBASSY IN NICOSIA
2	MARIA	GREEK EMBASSY IN MOSCOW
3	IOANNIS	GREEK COSULATE IN ST.PETERSBURG
4	IOANNIS	GREEK COSULATE IN NOVOROSSISK
6	EMMANOUEL	GREEK ENBASSY IN LONDON
NULL	NULL	BRITISH EMBASSY IN MOSCOW
NULL	NULL	BRITISH EMBASSY IN HELSINKI
NULL	NULL	RUSSIAN EMBASSY IN LONDON
NULL	NULL	CHINESE EMBASSY IN LONDON
NULL	NULL	FINLAND EMBASSY IN LONDON
NULL	NULL	FRENCH EMBASSY IN LONDON



# Πράξεις Συνόλων

- Οι πράξεις συνόλων ένωση (union), τομή (intersect), και διαφορά (except)
    - Εφαρμόζονται σε σχέσεις που έχουν το ίδιο πλήθος γνωρισμάτων και τα αντίστοιχα γνωρίσματα τους, έχουν το ίδιο πεδίο ορισμού.
    - Αντιστοιχούν στις πράξεις της σχεσιακής άλγεβρας  $\cup$ ,  $\cap$ ,  $-$ .
  - Κάθε μία από τις παραπάνω πράξεις εξαλείφει αυτόματα τις διπλότυπες εγγραφές.
  - Για να επιστραφούν όλες οι εγγραφές θα πρέπει να χρησιμοποιηθούν οι αντίστοιχες multiset εκδόσεις των προηγούμενων πράξεων *union all*, *intersect all* και *except all*.
- Υποθέστε ότι μια πλειάδα εμφανίζεται  $m$  φορές σε μία σχέση  $r$  και  $n$  φορές σε μία σχέση  $s$ , τότε εμφανίζεται:
- $m + n$  φορές στην σχέση  $r$  **union all**  $s$
  - $\min(m, n)$  φορές στην σχέση  $r$  **intersect all**  $s$
  - $\max(0, m - n)$  φορές στην σχέση  $r$  **except all**  $s$



## Ερώτηση 18<sup>η</sup> : Προβάλετε το όνομα και τον κωδικό όλων των πρεσβειών της Κίνας και της Ρωσίας

```
(SELECT ID_EMBASSY, EMBASSY_NAME
FROM EMBASSY AS E INNER JOIN COUNTRY AS C
ON E.COUNTRY_ID=C.ID COUNTRY
WHERE COUNTRY_NAME='RUSSIA' )
UNION
(SELECT ID_EMBASSY, EMBASSY_NAME
FROM EMBASSY AS E INNER JOIN COUNTRY AS C
ON E.COUNTRY_ID=C.ID COUNTRY
WHERE COUNTRY_NAME='CHINA' )
```

Χρησιμοποιείται για να συνδυάσει τα αποτελέσματα από 2 ή περισσότερα **SELECT**:

- Κάθε select πρέπει να έχει τον ίδιο αριθμό στηλών
- Οι στήλες πρέπει να έχουν παρόμοιους τύπους δεδομένων
- Οι στήλες σε κάθε select πρέπει να είναι με την ίδια σειρά

TABLE: EMBASSY

<u>ID_EMBASSY</u>	<u>EMBASSY_NAME</u>
4	RUSSIAN EMBASSY IN ATHENS
16	RUSSIAN EMBASSY IN LONDON
1	CHINESE EMBASSY IN ATHENS
17	CHINESE EMBASSY IN LONDON



## Ερώτηση 19<sup>η</sup> : Προβάλετε τα ονόματα των χωρών οι οποίες διατηρούν πρεσβείες στην Ελλάδα και στην Μεγάλη Βρετανία

```
(SELECT FOREIGN_C.COUNTRY_NAME FROM  
COUNTRY AS FOREIGN_C  
INNER JOIN EMBASSY AS E ON E.COUNTRY_ID=FOREIGN_C.ID_COUNTRY  
INNER JOIN CITY AS C2 ON E.CITY_ID=C2.ID_CITY  
INNER JOIN COUNTRY AS HOSTING_C ON HOSTING_C.ID_COUNTRY=C2.IDCOUNTRY  
WHERE HOSTING_C.COUNTRY_NAME= 'GREECE')
```

**INTERSECT**

```
(SELECT FOREIGN_C.COUNTRY_NAME FROM  
COUNTRY AS FOREIGN_C  
INNER JOIN EMBASSY AS E ON E.COUNTRY_ID=FOREIGN_C.ID_COUNTRY  
INNER JOIN CITY AS C2 ON E.CITY_ID=C2.ID_CITY  
INNER JOIN COUNTRY AS HOSTING_C ON HOSTING_C.ID_COUNTRY=C2.IDCOUNTRY  
WHERE HOSTING_C.COUNTRY_NAME= 'GREAT BRITAIN')
```

Δημιουργούμε  
δύο στιγμιότυπα  
της ίδιας σχέσης

TABLE: FOREIGN_C ⋈ EMBASSY ⋈ CITY ⋈ HOSTING_C	
COUNTRY_NAME	
FINLAND	
FRENCH	
CHINA	
RUSSIA	

**Ερώτηση 20<sup>η</sup> : Προβάλετε τα ονόματα των χωρών  
οι οποίες διατηρούν πρεσβείες στην Ελλάδα αλλά  
όχι στην Μεγάλη Βρετανία**

```
(SELECT FOREIGN_C.COUNTRY_NAME FROM  
COUNTRY AS FOREIGN_C  
INNER JOIN EMBASSY AS E ON E.COUNTRY_ID=FOREIGN_C.ID_COUNTRY  
INNER JOIN CITY AS C2 ON E.CITY_ID=C2.ID_CITY  
INNER JOIN COUNTRY AS HOSTING_C ON HOSTING_C.ID_COUNTRY=C2.ID_COUNTRY  
WHERE HOSTING_C.COUNTRY_NAME= 'GREECE')
```

***EXCEPT***

```
(SELECT FOREIGN_C.COUNTRY_NAME FROM  
COUNTRY AS FOREIGN_C  
INNER JOIN EMBASSY AS E ON E.COUNTRY_ID=FOREIGN_C.ID_COUNTRY  
INNER JOIN CITY AS C2 ON E.CITY_ID=C2.ID_CITY  
INNER JOIN COUNTRY AS HOSTING_C ON HOSTING_C.ID_COUNTRY=C2.ID_COUNTRY  
WHERE HOSTING_C.COUNTRY_NAME= 'GREAT BRITAIN')
```

TABLE: FOREIGN\_C ⋈ EMBASSY ⋈ CITY ⋈ HOSTING\_C

COUNTRY_NAME
GREAT BRITAIN
CYPRUS



# Συναθροιστικές Συναρτήσεις

- Οι συγκεκριμένες συναρτήσεις εφαρμόζονται σε γνωρίσματα σχέσεων και επιστρέφουν μία τιμή
- Παραδείγματα τέτοιων συναρτήσεων
  - avg:** μέσος όρος
  - min:** minimum value (ελάχιστη τιμή)
  - max:** maximum value (μέγιστη τιμή)
  - sum:** άθροισμα τιμών
  - count:** καταμέτρηση εγγραφών
- Όλες οι συναθροιστικές συναρτήσεις αγνοούν τις πλειάδες με null τιμές στα γνωρίσματα στα οποία εφαρμόζονται
  - εκτός της count(\*)
- Δεν μπορεί να χρησιμοποιηθούν σύνθετες συναθροιστικές συναρτήσεις στην SQL. Πχ **max (avg (...))**





Ερώτηση 21" : Να βρεθεί το πλήθος των  
πρεσβειών που είναι καταγεγραμμένες  
στην βάση

```
SELECT count(*) AS Plithos FROM  
EMBASSY
```

Εφαρμόζουμε μετονομασία

TABLE: EMBASSY

PLITHOS
---------

Ερώτηση 22<sup>η</sup> : Να βρεθεί το πλήθος των διακριτών περιοχών που φιλοξενούνται πρεσβείες

```
SELECT count(distinct district) AS  
Plithos FROM  
EMBASSY
```

Επιστρέφει μόνο τις διακριτές τιμές

TABLE: EMBASSY

PLITHOS
16

```
SELECT count(district) AS Plithos FROM  
EMBASSY
```

TABLE: EMBASSY

PLITHOS
19



# Εντολή ομαδοποίησης

## Group By

- Ομαδοποιεί τις εγγραφές με βάση ένα ή περισσότερα γνωρίσματα
- Για να εφαρμόσουμε τις συναθροιστικές συναρτήσεις όχι μόνο σε πλειάδες αλλά και σε υπο-σύνολα πλειάδων μιας σχέσης χρησιμοποιούμε το **group by**
  - Δηλαδή, ομαδοποιούμε τις πλειάδες που έχουν την ίδια τιμή για κάποια γνωρίσματα και εφαρμόζουμε τη συναθροιστική συνάρτηση σε κάθε ομάδα ανεξάρτητα

**Σημ:** Τα γνωρίσματα που εμφανίζονται σε μία δήλωση select και δεν εφαρμόζεται σε αυτά μία συναθροιστική συνάρτηση θα πρέπει να περιλαμβάνονται στην λίστα των γνωρισμάτων ομαδοποίησης



**Ερώτηση 23<sup>η</sup> : Να προβληθεί το όνομα  
κάθε χώρας και το αντίστοιχο πλήθος των  
πρεσβειών που διατηρεί η χώρα στο  
εξωτερικό**



```
SELECT FOREIGN_C.COUNTRY_NAME, COUNT(*) AS PLITHOS
FROM
    COUNTRY AS FOREIGN_C
INNER JOIN      EMBASSY AS E ON E.COUNTRY_ID=FOREIGN_C.ID_COUNTRY
GROUP BY FOREIGN_C.ID_COUNTRY;
```

Ομαδοποιούμε τις εγγραφές με βάση  
των κωδικό της χώρας που διατηρεί  
πρεσβεία στο εξωτερικό

TABLE: FOREIGN\_C ⋈ EMBASSY

COUNTRY_NAME	PLITHOS
GREECE	7
FINLAND	2
FRENCH	2
CHINA	2
GREAT BRITAIN	3
CYPRUS	1
RUSSIA	2





# Having

- Μας επιτρέπει να επιστρέψουμε πλειάδες οι οποίες ικανοποιούν κάποιο κριτήριο εάν εφαρμοστεί σε αυτές μία συναθροιστική συνάρτηση.
  - Having avg(Price)<10
  - Having count(ID) between 2 and 4
  - Having max(grade) < 15
- Είναι δυνατό να έχουμε στην ίδια εντολή **select** ταυτόχρονα και δήλωση **where** και δήλωση **having**
  - Η εφαρμογή των κριτηρίων που επιφέρει η δήλωση Having εφαρμόζεται στις εγγραφές μετά την ομαδοποίηση ενώ η εφαρμογή των κριτηρίων που επιφέρει η δήλωση where εφαρμόζεται πριν την ομαδοποίηση





**Ερώτηση 24<sup>η</sup> : Να προβληθεί το όνομα  
κάθε χώρας που διατηρεί περισσότερες  
από 2 πρεσβείες στο εξωτερικό και το  
αντίστοιχο πλήθος των πρεσβειών**

```
SELECT FOREIGN_C.COUNTRY_NAME,COUNT(*) AS PLITHOS
FROM
    COUNTRY AS FOREIGN_C
INNER JOIN      EMBASSY AS E ON E.COUNTRY_ID=FOREIGN_C.ID_COUNTRY
GROUP BY FOREIGN_C.ID_COUNTRY
HAVING COUNT(FOREIGN_C.ID_COUNTRY)>2;
```

Εφαρμόζουμε την συνθήκη Having  
προκειμένου να επιλέξουμε μόνο τις  
εγγραφές με τις επιθυμητές ιδιότητες

TABLE: FOREIGN\_C ⋈ EMBASSY

COUNTRY_NAME	PLITHOS
GREECE	7
GREAT BRITAIN	3

# Εμφωλευμένες Επερωτήσεις

- Ένα υπο-ερώτημα είναι μια παράσταση `select-from where` που είναι ένθετο μέσα σε ένα άλλο ερώτημα.
- Μπορούμε να εμφωλεύσουμε παραπάνω από ένα ερωτήματα
- Τελεστές :
  - `in` : ελέγχει την ύπαρξη μιας εγγραφής σε μία σχέση
  - `not in` : ελέγχει την μη ύπαρξη μιας εγγραφής σε μία σχέση
  - συγκρίνουν μια τιμή με ένα σύνολο τιμών

- Επιτρέπει να καθορίσουμε πολλαπλές τιμές σε μία πρόταση `WHERE`
- Είναι συντομογραφία για πολλαπλές συνθήκες `OR`



## Ερώτηση 27<sup>η</sup> :Επιλέξτε το όνομα των πρεσβειών που βρίσκονται στην χώρα “GREECE”

```
SELECT EMBASSY_NAME  
FROM EMBASSY  
WHERE CITY ID IN  
(SELECT ID_CITY  
  FROM CITY  
   WHERE IDCOUNTRY IN  
    (SELECT ID_COUNTRY FROM COUNTRY  
     WHERE COUNTRY_NAME = 'GREECE'))  
)
```

TABLE: CITY

ID_CITY
1
3

TABLE: EMBASSY

EMBASSY
CHINESE EMBASSY IN ATHENS
BRITISH EMBASSY IN ATHENS
CYPRIAN EMBASSY IN ATHENS
RUSSIAN EMBASSY IN ATHENS
FRENCH CONSULATE IN THESSALONIKI
FINLAND CONSULATE IN THESSALONIKI

TABLE: COUNTRY

ID_COUNTRY
1



Ερώτηση 28<sup>η</sup> :Επιλέξτε το όνομα των  
πρεσβειών που δεν βρίσκονται στην χώρα  
“GREECE”



```
SELECT EMBASSY_NAME
FROM EMBASSY
WHERE CITY_ID IN
(SELECT ID_CITY
FROM CITY
WHERE
IDCOUNTRY NOT IN
(SELECT ID_COUNTRY
FROM COUNTRY
WHERE
COUNTRY_NAME =
'GREECE')
)
```

TABLE: EMBASSY

EMBASSY
GREEK EMBASSY IN PARIS
GREEK CONSULATE IN MASELLIE
GREEK EMBASSY IN NICOSIA
GREEK EMBASSY IN MOSCOW
GREEK COSULATE IN ST.PETERSBURG
GREEK COSULATE IN NOVOROSSISK
GREEK ENBASSY IN LONDON
BRITISH EMBASSY IN MOSCOW
BRITISH EMBASSY IN HELSINKI
RUSSIAN EMBASSY IN LONDON
CHINESE EMBASSY IN LONDON
FINLAND EMBASSY IN LONDON
FRENCH EMBASSY IN LONDON





## Σύγκριση Συνόλων

- Τελεστές:
  - **>all** : μεγαλύτερο απ' όλα,
  - **<>all** διαφορετικό από όλα
    - **<>all** ισοδύναμο με το **'not in'**
  - **>some** : μεγαλύτερο από τουλάχιστον ένα,
  - **>= some** : μεγαλύτερο ίσο από τουλάχιστον ένα
  - **= some**
    - Τουλάχιστον ένα
    - ισοδύναμο με το **'in'**
  - **any** : ισοδύναμη με την **some** στην SQL



Ερώτηση 29<sup>η</sup> : Να προβληθεί το όνομα της πόλης που φιλοξενεί τις περισσότερες πρεσβείες απ' όλες

```
SELECT CITY_NAME
FROM EMBASSY INNER JOIN CITY
ON CITY_ID=ID_CITY
GROUP BY ID_CITY
HAVING COUNT(ID_EMBASSY)
        >= all
(SELECT COUNT(ID_EMBASSY)
FROM EMBASSY
GROUP BY ID_CITY) ;
```

TABLE: CITY

CITY_NAME
-----------

LONDON
--------



## Έλεγχος για κενές σχέσεις

- Η δομή
  - **Exists**: επιστρέφει true αν το όρισμα του υπο-ερωτήματος δεν είναι κενό.
  - **Not exists**: επιστρέφει true αν το όρισμα του υποερωτήματος είναι κενό.
- Δομή ερωτήματος:

```
SELECT a1,...,an FROM TABLE_A  
WHERE EXISTS  
    (SELECT a1,...,an FROM TABLE_B [WHERE ...])
```



# SQL Quick Reference

*From W3Schools*

<https://www.w3schools.com/sql/default.asp>



