

CITY

Cid	Name	Population	Country
2	Lamia	50000	Greece
3	Nicosia	500000	Cyprus
1	Thessaloniki	1000000	Greece

ALTER TABLE CITY
ADD COLUMN IsCapital VARCHAR(10);

Αλλάζει τη μορφή του πίνακα
Προσθέτοντας μία νέα στήλη

Cid	Name	Population	Country	IsCapital
2	Lamia	50000	Greece	No
3	Nicosia	500000	Cyprus	Yes
1	Thessaloniki	1000000	Greece	No

UPDATE CITIES
SET IsCapital = "No"
WHERE cid IN (2, 1);

Αλλάζει τις τιμές στηλών μίας συγκεκριμένης εγγραφής

UPDATE CITIES
SET IsCapital = "Yes"
WHERE cid = 3;

ALTER TABLE CITIES
RENAME COLUMN cid TO CityID;

CityID	Name	Population	Country	IsCapital
2	Lamia	50000	Greece	No
3	Nicosia	500000	Cyprus	Yes
1	Thessaloniki	1000000	Greece	No

UPDATE CITIES
SET Name = "Lemesos", Population = 200000, IsCapital = "No"
WHERE Cid = 3;

Cid	Name	Population	Country	IsCapital
2	Lamia	50000	Greece	No
3	Lemesos	200000	Cyprus	No
1	Thessaloniki	1000000	Greece	No

ALTER TABLE CITIES
DROP COLUMN IsCapital;

CityID	Name	Population	Country
--------	------	------------	---------

2	Lamia	50000	Greece
3	Lemesos	200000	Cyprus
1	Thessaloniki	1000000	Greece

Στη βάση δεδομένων δημιουργήθηκε ένα νέο πίνακάκι με στοιχεία χωρών, και έχουμε πλέον CountryID για να περιγράφουμε τις χώρες. Αρα θέλουμε να σβήσουμε τα στοιχεία που υπάρχουν για τις χώρες και να αλλάξουμε τον τύπο από VARCHAR σε INT.

```
UPDATE CITIES
SET Country = NULL
WHERE Country IS NOT NULL;
```

CityID	Name	Population	Country
2	Lamia	50000	NULL
3	Lemesos	200000	NULL
1	Thessaloniki	1000000	NULL

```
ALTER TABLE
MODIFY COLUMN Country INT;
```

```
ALTER TABLE
RENAME COLUMN Country TO CountryID;
```

CityID	Name	Population	CountryID
2	Lamia	50000	NULL
3	Lemesos	200000	NULL
1	Thessaloniki	1000000	NULL

```
UPDATE CITIES
SET CountryID = 50
WHERE CityID IN (1, 2);
```

```
UPDATE CITIES
SET CountryID = 32
WHERE CityID IN (3);
```

CityID	Name	Population	CountryID
2	Lamia	50000	50
3	Lemesos	200000	32
1	Thessaloniki	1000000	50

CREATE TABLE: Δημιουργεί πίνακες
Να θυμόμαστε τα PRIMARY KEYS, τα FOREIGN KEYS και τους τύπους της κάθε στήλης

DROP TABLE: Διαγράφει πίνακες

ALTER TABLE: Αλλάζει τη μορφή του πίνακα
ADD COLUMN
DROP COLUMN
MODIFY COLUMN
RENAME COLUMN

SELECT: Δημιουργία ερωτήματος

```
SELECT *
FROM table
```

```
SELECT *
FROM table
WHERE conditions.....
```

```
SELECT column1, column2, column3 AS Konstantina
FROM table
WHERE conditions.....
```

```
SELECT column1, column2, column3 AS Konstantina
FROM table
WHERE conditions.....
(ORDER BY column ASC/DESC);
```

Αν θέλουμε να ομαδοποιήσουμε σύμφωνα με μία συγκεκριμένη ιδιότητα (πχ πανεπιστήμιο που πηγαίνουν οι φοιτητές, χώρα καταγωγής, τμήματα κλπ)

```
SELECT column1, column2, column3 AS Konstantina
FROM table
WHERE conditions.....
GROUP BY column...
```

Αν θέλουμε να έχουμε και συγκεκριμένες ιδιότητες που να αφορούν τις ομάδες (πχ υπολογισμός πλήθους μελών και αυτός να είναι μεγαλύτερος ένος αριθμού, υπολογισμός μέσου όρου, αθροίσματος κλπ) χρησιμοποιούμε το HAVING

```
SELECT column1, column2, column3 AS Konstantina
FROM table
WHERE conditions.....
GROUP BY column...
HAVING ...
```

Κάτω κάτω μπαίνει και το ORDER BY αν θες...

Συναρτήσεις υπολογισμού:

```
COUNT(column)
SUM(column)
AVG(column)
MIN(column)
MAX(column)
```

Υποερωτήματα: Επειδή κάθε ερώτημα επιστρέφει ένα σύνολο πίσω, η SQL τα θεωρεί όλα σύνολα. Με τον τελεστή IN και τον EXISTS μπορούμε να πάρουμε το εξής:

```
ΚΛΑΣΙΚΟ ΕΡΩΤΗΜΑ SQL
WHERE column IN ( Κλασικό ερώτημα SQL)
```

Πχ Αν ήθελες να εμφανίσεις τα ονόματα των τμημάτων που έχουν μαθήτρια η οποία λέγεται Κωνσταντίνα θα μπορούσες να κάνεις το εξής:

STUDENTS

Sid	Name	Did
1	Konstantina	1
2	Nikos	3
3	Konstantina	3
4	Petros	2
5	Panagiotis	1
6	Arsenios	4
7	Anastasis	1
8	Despina	5
9	Konstantina	5

DEPARTMENTS

Did	Name
1	ΠΑΜΑΚ
2	ΠΑΔΜ
3	ΠΑΔΘ
4	ΠΑΔΑ
5	ΕΜΠ

Παρατηρώ ότι οι δύο πίνακες έχουν κοινό το Did. Θα μπορούσα να λύσω την άσκηση με ένα απλό γινόμενο:

```
SELECT DISTINCT D.Name
FROM STUDENTS S, DEPARTMENTS D
WHERE S.Did = D.Did AND S.Name = "Konstantina";
```

Παρ' όλα αυτά μπορώ να επωφεληθώ από το κοινό DID με άλλον τρόπο.

Θα βρω όλα τα Did τα οποία έχουν Konstantina σαν Name

```
SELECT DISTINCT Did
FROM STUDENTS
WHERE Name = "Konstantina";
```

Αυτό επιστρέφει:

DID
1
3
5

Το οποίο αλλιώς μπορούμε να το θεωρήσουμε σαν ένα σύνολο (1, 3, 5).

Άρα τώρα έχω τα Did των departments που έχουν Konstantina. Μπορώ να τα πάρω πλέον και να Κάνω ένα IN στον άλλο πίνακα, επιλέγοντας το όνομα του τμήματος ως τη στήλη που θα προβάλλω.

```
SELECT Name
FROM Departments
WHERE Did IN (
    SELECT DISTINCT Did
    FROM STUDENTS
    WHERE Name = "Konstantina";
);
```

Name
ΠΑΜΑΚ
ΠΑΔΘ
ΕΜΠ

Επίσης για γινόμενα έχουμε τα JOINS, είναι ένας άλλος τρόπος να τα κάνεις (το αφήνω πάνω για να το διαβάσεις)

Υπάρχει επίσης το LIKE για να ψάξουμε αν κάτι ξεκινάει με μία λέξη ή ένα γράμμα, αν περιέχει μία λέξη ή ένα γράμμα ή αν τελειώνει με μία λέξη ή ένα γράμμα.

WHERE CustomerName LIKE 'a%'	Finds any values that start with "a"
WHERE CustomerName LIKE '%a'	Finds any values that end with "a"
WHERE CustomerName LIKE '%or%'	Finds any values that have "or" in any position
WHERE CustomerName LIKE '_r%'	Finds any values that have "r" in the second position
WHERE CustomerName LIKE 'a_%'	Finds any values that start with "a" and are at least 2 characters in length
WHERE CustomerName LIKE 'a__%'	Finds any values that start with "a" and are at least 3 characters in length
WHERE ContactName LIKE 'a%o'	Finds any values that start with "a" and ends with "o"

From <https://www.w3schools.com/sql/sql_like.asp>

% οποιοδήποτε πλήθος χαρακτήρων (0-oo)

_ ένας χαρακτήρας (αλλά οποιοσδήποτε χαρακτήρας)

Αν θέλουμε ένωση με DISTINCT τρόπο κάνουμε UNION

ΕΡΩΤΗΜΑ 1

UNION

ΕΡΩΤΗΜΑ 2;

Αν θέλουμε ένωση που κάθε στοιχείο να φαίνεται όσες φορές είναι κάνουμε UNION ALL

ΕΡΩΤΗΜΑ 1

UNION ALL

ΕΡΩΤΗΜΑ 2;

Αν θέλουμε τομή κάνουμε INTERSECT

ΕΡΩΤΗΜΑ 1

INTERSECT

ΕΡΩΤΗΜΑ 2;

Αν θέλουμε να κάνουμε διαφορά (ΕΡΩΤΗΜΑ 1 - ΕΡΩΤΗΜΑ 2) κάνουμε EXCEPT

ΕΡΩΤΗΜΑ 1

EXCEPT

ΕΡΩΤΗΜΑ 2;

Πολλές φορές στη βάση δεδομένων που έχουμε χρειάζεται να δημιουργούμε επιπλέον πίνακες που να παίζουν κάποιο συγκεκριμένο ρόλο. Αυτοί οι πίνακες ονομάζονται προβολές VIEWS.

CREATE VIEW view_name AS

ΕΡΩΤΗΜΑ SQL

Αυτό παίρνει το αποτέλεσμα (πινακάκι) που σου επιστρέφει το ερώτημα της SQL και το αποθηκεύει στη βάση δεδομένων. Μετά αν θες μπορείς να το χρησιμοποιήσεις σε άλλα ερωτήματα λες και είναι βασικός πίνακας SQL.

Αν έχεις FOREIGN KEYS, υπάρχει ένα θέμα. Τι θα γίνει αν η εγγραφή του άλλου πίνακα διαγραφεί;;

CITIES

ID	Name	CountryID
1	Lemosos	32
2	Famagusta	32
3	Azido Naxos	32

COUNTRIES

1	Lemesos	32
2	Famagusta	32
3	Ayia Napa	32
4	Nicosia	32
5	Pafos	32
6	Larnaka	32
7	Kerinia	32
8	Lamia	50
9	Athens	50
10	Hamburg	34
11	Cologne	34
12	Berlin	34
13	Nurenmberg	34
14	Munich	34
15	Belgrade	2

COUNTRIES

ID	Name
1	Brazil
2	Serbia
...	..
31	Israel
32	Cyprus
33	Malta
34	Germany
...	...
50	Greece
...	..

Αν θα ήθελα να διαγράψω τη Σερβία, τότε το 2 που είναι το ID της Σερβίας μπορεί να επαναχρησιμοποιηθεί για κάποια άλλη (νέα) χώρα που θέλει να μπει στη βάση δεδομένων.

Έστω πχ ότι η χώρα αυτή είναι η Ολλανδία.

Τότε σύμφωνα με τον πίνακα CITIES, το Βελιγράδι ανήκει στην Ολλανδία.

Αυτό είναι κάτι το οποίο θέλουμε να αποφύγουμε. Δηλαδή, η τιμή 2 που υπάρχει στο βελιγράδι πρέπει να αλλάξει, ή να κάνουμε κάτι ακόμη πιο δραστικό.

Υπάρχουν 2 επιλογές:

- Είτε θα κάνουμε NULL την τιμή αυτή (ON DELETE SET NULL).
- Είτε θα διαγράψουμε όλη την εγγραφή (ON DELETE CASCADE).

Την επιλογή αυτή την παίρνουμε όταν φτιάχνουμε τον πίνακα CITIES:

```
CREATE TABLE CITIES (
  ID INT NOT NULL,
  NAME VARCHAR(255) NOT NULL,
  CountryID INT,
  PRIMARY KEY (ID),
  FOREIGN KEY (CountryID) REFERENCES COUNTRY.ID ON DELETE SET NULL
);
```

```
CREATE TABLE CITIES (
  ID INT NOT NULL,
  NAME VARCHAR(255) NOT NULL,
  CountryID INT,
  PRIMARY KEY (ID),
  FOREIGN KEY (CountryID) REFERENCES COUNTRY.ID ON DELETE CASCADE
);
```

Γενικά διαφορετικά προγράμματα SQL αποδέχονται και διαφορετικά πράγματα. Δεν πειράζει αυτό, εμείς γράφουμε κανονικά τις εντολές μας στο χαρτί ακόμη και αν δε δουλεύουν σε ένα, γιατί θα δουλεύουν σε άλλο.

Επίσης έχουμε το LIMIT και το SELECT TOP

Το LIMIT επιλέγει ένα μέγιστο πλήθος από εγγραφές

SELECT *
FROM CITIES
LIMIT 3;

SELECT TOP 3 *
FROM CITIES;

ID	Name	CountryID
1	Lemesos	32
2	Famagusta	32
3	Ayia Napa	32

CD

Cid	Ctitle	Artist
1	TPAB	Kendrick Lamar
2	Midnights	Taylor Swift
3	Section 80.	Kendrick Lamar
4	Hello	Jim Butler

SONG

Sid	Stitle	Artist
1	Midnights	Peter Son
2	Hello	Adele
3	United In Grief	Kendrick Lamar
4	N95	Kendrick Lamar

Cid	Ctitle	C.Artist	Sid	Stitle	S.Artist
1	TPAB	Kendrick Lamar	1	Midnights	Peter Son
1	TPAB	Kendrick Lamar	2	Hello	Adele
1	TPAB	Kendrick Lamar	3	United In Grief	Kendrick Lamar
1	TPAB	Kendrick Lamar	4	N95	Kendrick Lamar
2	Midnights	Taylor Swift	1	Midnights	Peter Son
2	Midnights	Taylor Swift	2	Hello	Adele
2	Midnights	Taylor Swift	3	United In Grief	Kendrick Lamar
2	Midnights	Taylor Swift	4	N95	Kendrick Lamar
3	Section 80	Kendrick Lamar	1	Midnights	Peter Son
3	Section 80	Kendrick Lamar	2	Hello	Adele
3	Section 80	Kendrick Lamar	3	United In Grief	Kendrick Lamar
3	Section 80	Kendrick Lamar	4	N95	Kendrick Lamar
4	Hello	Jim Butler	1	Midnights	Peter Son
4	Hello	Jim Butler	2	Hello	Adele
4	Hello	Jim Butler	3	United In Grief	Kendrick Lamar
4	Hello	Jim Butler	4	N95	Kendrick Lamar

Έστω ότι θέλουμε να βρούμε το ID της χώρας με το γήπεδο που έχει τη μεγαλύτερη χωρητικότητα.

ID	Name	Capacity	CountryID
1	OAKA	65000	50
2	Allianz Arena	72000	34
3	O2 Stadium	89000	5
4	Kalamaria	13000	50
5	Panthessaliko	19000	50

Με το MAX() μπορούμε να μάθουμε τη μεγαλύτερη χωρητικότητα

```
SELECT MAX(Capacity) AS MaximumCapacity
FROM Stadiums;
```

MaximumCapacity
89000

```
SELECT CountryID
FROM Stadiums
WHERE Capacity IN (
    SELECT MAX(Capacity) AS MaximumCapacity
    FROM Stadiums);
```

Υπάρχει και ένας ακόμη τρόπος, που είναι με το EXCEPT.
Σύνολο όλων των CountryID - Σύνολο των CountryID τα οποία έχουν Capacity μικρότερο από κάποιον άλλον.

Για να καταφέρεις να φτιάξεις το δεύτερο σύνολο πρέπει να συγκρίνεις χωρητικότητες γηπέδων μεταξύ δύο διαφορετικών εγγραφών. Αλλά ένα απλό SELECT με WHERE στον πίνακα Stadiums δεν μπορεί να το κάνει αυτό. Πρέπει με κάποιον τρόπο να δημιουργηθούν εγγραφές οι οποίες έχουν δύο στήλες χωρητικότητας, μία για ένα γήπεδο και μία για ένα άλλο γήπεδο, για όλα τα ζευγάρια γηπέδων που υπάρχουν.

S1.ID	S1.Name	S1.Capacity	S1.CountryID	S2.ID	S2.Name	S2.Capacity	S2.CountryID
1	OAKA	65000	50	1	OAKA	65000	50
1	OAKA	65000	50	2	Allianz Arena	72000	34
1	OAKA	65000	50	3	O2 Stadium	89000	5
1	OAKA	65000	50	4	Kalamaria	13000	50
1	OAKA	65000	50	5	Panthessaliko	19000	50
2	Allianz Arena	72000	34	1	OAKA	65000	50
2	Allianz Arena	72000	34	2	Allianz Arena	72000	34
2	Allianz Arena	72000	34	3	O2 Stadium	89000	5
2	Allianz Arena	72000	34	4	Kalamaria	13000	50
2	Allianz Arena	72000	34	5	Panthessaliko	19000	50
3	O2 Stadium	89000	5	1	OAKA	65000	50
3	O2 Stadium	89000	5	2	Allianz Arena	72000	34
3	O2 Stadium	89000	5	3	O2 Stadium	89000	5
3	O2 Stadium	89000	5	4	Kalamaria	13000	50
3	O2 Stadium	89000	5	5	Panthessaliko	19000	50
4	Kalamaria	13000	50	1	OAKA	65000	50
4	Kalamaria	13000	50	2	Allianz Arena	72000	34
4	Kalamaria	13000	50	3	O2 Stadium	89000	5
4	Kalamaria	13000	50	4	Kalamaria	13000	50
4	Kalamaria	13000	50	5	Panthessaliko	19000	50
5	Panthessaliko	19000	50	1	OAKA	65000	50
5	Panthessaliko	19000	50	2	Allianz Arena	72000	34
5	Panthessaliko	19000	50	3	O2 Stadium	89000	5
5	Panthessaliko	19000	50	4	Kalamaria	13000	50
5	Panthessaliko	19000	50	5	Panthessaliko	19000	50

Βρίσκω τα ID των σταδίων που έχουν μικρότερη χωρητικότητα από κάποιο άλλο στάδιο,
Το οποίο σημαίνει ότι δεν είναι τα στάδια με τη μέγιστη χωρητικότητα.


```
SELECT S1.ID  
FROM STADIUM S1, STADIUM S2  
WHERE S1.Capacity < S2.Capacity;
```

(1, 2, 4, 5)

Από όλα τα ID των σταδίων που υπάρχουν, αφαιρώ τα ID των σταδίων που έχουν μικρότερη χωρητικότητα από κάποιο άλλο στάδιο. Έτσι θα μείνω με ένα μοναδικό ID, το ID του σταδίου με τη μέγιστη χωρητικότητα.

```
SELECT ID  
FROM STADIUM
```

```
EXCEPT
```

```
SELECT S1.ID  
FROM STADIUM S1, STADIUM S2  
WHERE S1.Capacity < S2.Capacity;
```

Για να βρούμε τη χώρα

```
SELECT CountryID  
FROM Stadium  
WHERE ID IN (  
    SELECT ID  
    FROM STADIUM
```

```
EXCEPT
```

```
SELECT S1.ID  
FROM STADIUM S1, STADIUM S2  
WHERE S1.Capacity < S2.Capacity  
);
```