

ΒΑΣΙΚΕΣ ΕΝΤΟΛΕΣ ΣΤΗ MySQL

ΔΙΑΧΕΙΡΙΣΗ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

SHOW DATABASES;

- Εμφανίζει μια λίστα με όλες τις βάσεις στο σύστημα

CREATE DATABASE diek;

- Δημιουργία μιας νέας βάσης

DROP DATABASE diek;

- Διαγραφή υπάρχουσας βάσης

USE diek;

- Ορισμός επιλεγμένης βάσης

SHOW TABLES;

- Εμφανίζει τη λίστα των πινάκων της επιλεγμένης βάσης

ΒΑΣΙΚΟΙ ΤΥΠΟΙ ΔΕΔΟΜΕΝΩΝ

- INT ακέραιος
- FLOAT (m, n) κινητής υποδιαστολής m ψηφίων n δεκαδικών
- DATE ημερομηνία σε μορφή ΕΕΕΕ-ΜΜ-ΗΗ
- TIME ώρα σε μορφή ΩΩ:ΛΛ:ΔΔ
- CHAR (m) αλφαριθμητικό σταθερού μήκους m χαρακτήρων
- VARCHAR (m) αλφαριθμητικό μεταβλητού μήκους m χαρακτήρων

ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΠΕΔΙΩΝ

- NOT NULL Δεν επιτρέπεται να μείνει κενό. Υποχρεωτική απόδοση τιμής
- UNIQUE Μοναδικό. Δεν επιτρέπεται δύο εγγραφές να έχουν την ίδια τιμή
- DEFAULT Η τιμή που θα αποδοθεί αν δεν την καθορίσουμε ρητά
- AUTO_INCREMENT Αυτόματη αύξηση της τιμής κατά 1
- PRIMARY KEY Πρωτεύων κλειδί
- FOREIGN KEY Ξένο κλειδί

ΠΙΝΑΚΕΣ

```
CREATE TABLE mathites (  
  id INT NOT NULL AUTO_INCREMENT,  
  lastname VARCHAR(45) NOT NULL,  
  firstname VARCHAR(45) NOT NULL,  
  birthday DATE,  
  PRIMARY KEY (id)  
);
```

- Δημιουργία νέου πίνακα

DROP TABLE mathites;

- Διαγραφή πίνακα

DESC mathites;

- Εμφανίζει τη δομή (τα πεδία) του πίνακα

TRUNCATE TABLE mathites;

- Αδειάζει τον πίνακα (διαγραφή όλων των δεδομένων)

ALTER TABLE mathites ADD COLUMN apousies INT;

- Προσθήκη στήλης σε πίνακα

ALTER TABLE mathites DROP COLUMN firstname;

- Διαγραφή στήλης σε πίνακα

```
ALTER TABLE mathites RENAME TO spoudastes;
```

- Μετονομασία πίνακα

```
ALTER TABLE mathites CHANGE id kodikos INT;
```

- Μετονομασία στήλης πίνακα

```
ALTER TABLE mathites MODIFY firstname VARCHAR(25);
```

- Αλλαγή τύπου πεδίου

ΕΙΣΑΓΩΓΗ ΔΕΔΟΜΕΝΩΝ ΣΕ ΠΙΝΑΚΑ

```
INSERT INTO mathites VALUES (1, "Papadopoulos", "Kostas", "1991-04-12");
```

- Εισαγωγή δεδομένων σε πίνακα με απαρίθμηση τιμών

```
INSERT INTO mathites (firstname, lastname) VALUES ("Giannis", "Papadopoulos");
```

- Εισαγωγή δεδομένων σε πίνακα με ζεύξη στηλών - τιμών

```
INSERT INTO mathites VALUES (1, "Papadopoulos", "Kostas", "1991-04-12"), (2, "Oikonomou", "Panagiwths", "1992-01-31"), (3, "Karadimitriou", "Kostas", "1984-04-20");
```

- Εισαγωγή πολλών εγγραφών με μια εντολή.

ΕΠΙΛΟΓΗ ΔΕΔΟΜΕΝΩΝ ΑΠΟ ΠΙΝΑΚΑ

```
SELECT * FROM mathites;
```

- Προβολή όλων των περιεχομένων του πίνακα

```
SELECT lastname, firstname FROM mathites;
```

- Προβολή ορισμένων πεδίων του πίνακα

```
SELECT lastname FROM mathites WHERE firstname="Kostas";
```

- Προβολή με κριτήρια ορισμένων εγγραφών

ΤΕΛΕΣΤΕΣ ΣΥΓΚΡΙΣΗΣ

- < , > , >= , <= , = , <>

π.χ. Ποιες εγγραφές έχουν τιμή στο πεδίο id μεγαλύτερη ή ίση με 2;

```
SELECT * FROM mathites WHERE id>=2;
```

π.χ. Ποιες εγγραφές δεν έχουν ως όνομα Kostas;

```
SELECT * FROM mathites WHERE firstname<>"Kostas";
```

```
SELECT DISTINCT firstname FROM mathites;
```

- Απαλοιφή διπλοεγγραφών

```
SELECT lastname AS eponymo FROM mathites;
```

- Μετονομάζει ονόματα πεδίων

ΔΙΑΓΡΑΦΗ ΔΕΔΟΜΕΝΩΝ ΣΕ ΠΙΝΑΚΑ

```
DELETE FROM mathites WHERE lastname="Papadopoulos";
```

- Διαγραφή δεδομένων

```
DELETE FROM mathites;
```

- Διαγραφή όλων των δεδομένων ενός πίνακα

ΕΝΗΜΕΡΩΣΗ ΔΕΔΟΜΕΝΩΝ ΣΕ ΠΙΝΑΚΑ

```
UPDATE mathites SET firstname="Giorgos" WHERE lastname="Karadimitriou";
```

- Ενημέρωση τιμών σε υπάρχουσα εγγραφή σε πίνακα

ΛΟΓΙΚΟΙ ΤΕΛΕΣΤΕΣ [AND, OR, NOT]

Μπορούμε να έχουμε λογικούς τελεστές σε συνδυασμό με συγκριτικούς τελεστές.

```
SELECT * FROM product WHERE cost>=20 AND cost<=50;
SELECT * FROM product WHERE cost=20 OR cost=50;
SELECT * FROM product WHERE NOT cost=20;
```

ΤΕΛΕΣΤΗΣ ΕΥΡΟΥΣ ΤΙΜΩΝ [BETWEEN.. AND]

Ελέγχει αν μια τιμή βρίσκεται μεταξύ των δύο φραγμάτων.

```
SELECT * FROM product WHERE cost BETWEEN 20 AND 50;
SELECT * FROM product WHERE cost NOT BETWEEN 20 AND 50;
```

ΤΕΛΕΣΤΗΣ ΣΥΝΟΛΟΥ [IN, NOT IN]

Ελέγχει αν μια τιμή ανήκει ή δεν ανήκει σε ένα σύνολο τιμών.

```
SELECT * FROM persons WHERE FirstName IN ('Ελένη', 'Πέτρος');
SELECT * FROM persons WHERE FirstName NOT IN ('Ελένη', 'Πέτρος');
```

ΤΕΛΕΣΤΗΣ ΤΑΙΡΙΑΣΜΑΤΟΣ ΣΥΜΒΟΛΟΣΕΙΡΩΝ [LIKE]

- (%) Ο χαρακτήρας % ταιριάζει με οποιαδήποτε συμβολοσειρά.
- () Ο χαρακτήρας _ ταιριάζει με οποιαδήποτε χαρακτήρα.

```
SELECT * FROM persons WHERE city LIKE 'A%';
SELECT * FROM persons WHERE firstName LIKE '_ia';
```

ΣΥΝΑΘΡΟΙΣΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ

- AVG(..) Επιστρέφει το μέσο όρο μίας στήλης
- MIN(..) Επιστρέφει την ελάχιστη τιμή μίας στήλης
- MAX(..) Επιστρέφει την μέγιστη τιμή μίας στήλης
- SUM(..) Επιστρέφει το άθροισμα μίας στήλης
- COUNT(..) Επιστρέφει το πλήθος μίας στήλης

π.χ. `SELECT AVG(price) FROM orders;`

ΟΜΑΔΟΠΟΙΗΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ [GROUP BY]

Μπορούμε να χρησιμοποιήσουμε την GROUP BY σε συνδυασμό με μια συνάρτηση για να ομαδοποιήσουμε δεδομένα.

```
SELECT customerName, SUM(orderPrice)
FROM orders
GROUP BY customerName;
```

ΠΡΟΤΑΣΗ [HAVING]

Η πρόταση HAVING επιλέγει τις εγγραφές για τις οποίες ισχύει η συνθήκη που ακολουθεί. Η πρόταση HAVING τοποθετείται μετά την εντολή GROUP BY.

```
SELECT customerName, SUM(orderPrice)
FROM orders
GROUP BY customerName
HAVING SUM(orderPrice)<2000;
```

ΔΙΑΤΑΞΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ [ORDER BY..LIMIT]

Η ORDER BY ταξινομεί σε αύξουσα (ASC) ή φθίνουσα (DESC) σειρά τα δεδομένα. Η προκαθορισμένη τιμή είναι ASC.

```
SELECT * FROM orders ORDER BY customername DESC;
```

Περιορισμός του μεγέθους του αποτελέσματος.

```
SELECT * FROM product ORDER BY cost LIMIT 2;
```

ΣΥΝΤΑΞΗ ΕΡΩΤΗΜΑΤΟΣ SELECT

```
select [distinct] target_list
from tuple_variable_list
[where qualification]
[group by grouping_attributes]
[having group_condition]
[order by target_list_subset]
```

ΑΝΑΠΑΡΑΣΤΑΣΗ ΣΥΣΧΕΤΙΣΕΩΝ

Συσχέτιση 1:1

Όταν η συσχέτιση είναι 1:1, τότε κάθε εγγραφή του ενός πίνακα συσχετίζεται με μία εγγραφή του άλλου πίνακα. Επομένως, όταν η συσχέτιση είναι 1:1 τότε το κλειδί του ενός πίνακα θα γίνει ξένο κλειδί στον άλλο πίνακα. Μπορούμε να επιλέξουμε όποιον από τους δύο πίνακες θέλουμε για να βάλουμε το ξένο κλειδί.

Συσχέτιση 1:N

Η δεύτερη περίπτωση συσχέτισης πινάκων είναι όταν μια εγγραφή του πρώτου πίνακα αντιστοιχεί σε πολλές εγγραφές του άλλου πίνακα. Αν λοιπόν έχουμε δύο πίνακες A και B και κάθε εγγραφή του A αντιστοιχεί σε πολλές εγγραφές του B, τότε βάζουμε το κλειδί του A ως ξένο κλειδί στον B.

Συσχέτιση M:N

Η τρίτη και τελευταία περίπτωση συσχέτισης είναι όταν πολλές εγγραφές του ενός πίνακα συσχετίζονται με πολλές εγγραφές του άλλου πίνακα. Η περίπτωση αυτή είναι ιδιόμορφη γιατί απαιτεί την προσθήκη ενός επιπλέον πίνακα που θα συσχετίζει τους δύο βασικούς. Αν λοιπόν θεωρήσουμε ότι έχουμε τους πίνακες A και B που πολλές εγγραφές τους ενός συσχετίζονται με εγγραφές του άλλου, τότε δημιουργούμε ένα τρίτο πίνακα Γ και βάζουμε τα κλειδιά των πινάκων A και B ως πεδία στον πίνακα Γ (ξένα κλειδιά).

```
CREATE TABLE teacher(
    am CHAR(7) NOT NULL,
    surname VARCHAR(20),
    speciality VARCHAR(20),
    PRIMARY KEY(am)
);
```

```
CREATE TABLE subject(
    id INT NOT NULL,
    title VARCHAR(20),
    s_am CHAR(7),
    PRIMARY KEY(id),
    FOREIGN KEY (s_am) REFERENCES teacher(am)
);
```

ΣΥΝΕΝΩΣΕΙΣ (JOINS)

Οι συνενώσεις (joins) χρησιμοποιούνται για ερωτήματα σε δύο οι περισσότερους πίνακες για τους οποίους υπάρχει κάποια σχέση ανάμεσα στα γνωρίσματα (συνήθως ανάμεσα στα κλειδιά).

- **ON** χρησιμοποιείται για συνένωση στηλών με διαφορετικό όνομα

```
SELECT * FROM table_a JOIN table_b ON table_a.a1 = table_b.b1;
```

- **USING** χρησιμοποιείται για συνένωση στηλών με ίδιο όνομα (έστω η στήλη w). Επίσης, με τον τελεστή USING επιλέγονται μόνο οι εγγραφές για τις οποίες οι αντίστοιχες τιμές στο πεδίο ή πεδία με ίδιο όνομα (w) είναι ίσες.

```
SELECT * FROM table_a JOIN table_b USING (w);
```

Στην περίπτωση που τα πεδία που συγκρίνουμε έχουν ίδιο όνομα τα ξεχωρίζουμε βάζοντας το όνομα του πίνακα και μια τελεία (<όνομα_πίνακα>.<όνομα_πεδίου>) όπως στο παρακάτω παράδειγμα:

```
SELECT t.name, t.surname FROM teacher AS t, subject AS s
WHERE s.type='LAB' AND s.tar_tautotitas=t.ar_tautotitas;
```