



### ΠΕΡΙΕΧΟΜΕΝΑ:

1. Κλειδιά:
  1. Δημιουργία Κλειδιών
    1. Πρωτεύοντα Κλειδιά
    2. Ξένα Κλειδιά
  2. Τροποποίηση Κλειδιών
2. Άλλοι τύποι περιορισμών και υπολογιζόμενες στήλες
3. Σύνολα Χαρακτήρων
4. Μηχανισμοί Αποθήκευσης

Νίκος Π.

Χρυσός Χορηγός Μαθήματος

Άννα

Ασημένιος Χορηγός Μαθήματος

**Δημιουργία Πρωτεύοντος Κλειδιού (Primary Key):**

- Α' τρόπος: Ορίζοντας ως χαρακτηριστικό το "PRIMARY KEY" κατά τη δημιουργία του πίνακα:

```
column_name1 datatype1 PRIMARY KEY [attributes]
```

- Β' τρόπος: Με το εξής συντακτικό:

```
[CONSTRAINT [name]] PRIMARY KEY (col1, col2, ...)
```

- Μπαίνει ακριβώς μετά τις δηλώσεις των στηλών
- Προαιρετικά μπορούμε να καθορίσουμε ένα όνομα.
- Με το συντακτικό αυτό μπορούμε να ορίσουμε πρωτεύον κλειδί - συνδυασμό 2 ή περισσότερων στηλών

**Παράδειγμα 1: createpk.sql**

```
CREATE TABLE employees (  
    emp_no INT PRIMARY KEY NOT NULL UNIQUE AUTO_INCREMENT  
    ...  
)
```

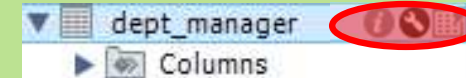
```
CREATE TABLE dept_emp (  
    ...  
    CONSTRAINT dept_emp_pk PRIMARY KEY (emp_no, dept_no)  
);
```

**Σημείωση:**

- Ο Α' τρόπος λέμε ότι είναι περιορισμός σε επίπεδο στήλης (column level), ενώ ο Β' τρόπος σε επίπεδο πίνακα (table level)

**Δημιουργία πρωτεύοντος κλειδιού με το MySQL Workbench**

- Κατσαβίδι στον πίνακα:



- Κλικ στις στήλες που θέλουμε να είναι πρωτεύοντα κλειδιά:

Column Name	Datatype	PK	NN
dept_no	CHAR(4)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
emp_no	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

- Κλικ στο Apply

**Άσκηση 1:**

Αφού τροποποιήσετε τον πίνακα departments (της βάσης emp), ώστε να ορίζεται ότι η στήλη dept\_no είναι το πρωτεύον κλειδί, επεκτείνετε το script του παραδείγματος 1, ώστε να ενσωματώνει τις πληροφορίες για το πρωτεύον κλειδί του πίνακα dept\_manager (συνδυασμός των dept\_no και emp\_no).

### Δημιουργία Ξένου Κλειδιού (FOREIGN Key):

- Α' τρόπος: Ορίζοντας ως χαρακτηριστικό το κατά τη δημιουργία του πίνακα:

```
col ... REFERENCES table(col) [ON DELETE CASCADE|SET NULL]
```

- Η στήλη col είναι ξένο κλειδί στη στήλη col του πίνακα table
- ON DELETE CASCADE: Αν διαγραφεί η εγγραφή του πρωτεύοντος κλειδιού, θα διαγραφούν και όλες οι εγγραφές που έχουν ως ξένο κλειδί αυτό το πρωτεύον κλειδί.
- ON DELETE SET NULL: Αν διαγραφεί η εγγραφή του πρωτεύοντος κλειδιού, η στήλη στο ξένο κλειδί θα τεθεί NULL.
- Β' τρόπος: Να έπεται του ορισμού των στηλών:

```
[CONSTRAINT [name]] FOREIGN KEY (col1, col2, ...)
REFERENCES table(col) [ON DELETE CASCADE|SET NULL]
```

### Παράδειγμα 2: createfk.sql

```
CREATE TABLE dept_emp (
emp_no ... REFERENCES employees(emp_no) ON DELETE CASCADE,
dept_no ... REFERENCES departments(dept_no) ON DELETE CASCADE,
...
);
CREATE TABLE dept_manager (
...
CONSTRAINT dept_mgr_fk_emp FOREIGN KEY (emp_no)
REFERENCES employees(emp_no)
);
```

### Δημιουργία ξένου κλειδιού με το MySQL Workbench

- Κατσαβίδι στον πίνακα:
- Κλικ στο tab "Foreign Keys":
- Κλικ στο Foreign Key Name και επιλέγουμε ένα όνομα:
- Επιλογή του πίνακα του πρωτεύοντος κλειδιού
- Επιλογή των στηλών ξένου - πρωτεύοντος κλειδιού
- Επιλογή CASCADE/SET NULL

### Άσκηση 2:

Ολοκληρώστε την κατασκευή των κλειδιών του πίνακα salaries: Πρωτεύον κλειδί (emp\_no, from\_date), ξένο κλειδί (emp\_no-> employees) και ενσωματώστε τα στο script.

**Τροποποίηση Πίνακα:**

```
ALTER TABLE [database_name.]table_name  
statement1,  
statement2,  
...;
```

- όπου ως statement παραθέτουμε μία από τις παρακάτω:

```
ADD PRIMARY KEY (col1, col2,...)
```

- προσθέτει πρωτεύον κλειδί στον πίνακα

```
DROP PRIMARY KEY
```

- διαγράφει το πρωτεύον κλειδί

```
ADD [CONSTRAINT [name]] FOREIGN KEY (col1, col2, ...)  
REFERENCES table(col)  
[ON DELETE CASCADE|SET NULL]
```

- προσθέτει ένα ξένο κλειδί στον πίνακα

```
DROP FOREIGN KEY name
```

- Διαγράφει το ξένο κλειδί

**Παρατήρηση:**

- Η διαγραφή ξένου κλειδιού δείχνει ότι είναι προτιμότερο να “βαφτίζουμε” με ένα όνομα τα ξένα κλειδιά για να μπορούμε να αναφερόμαστε σε αυτά.
- Επίσης είναι χρήσιμο τα ξένα κλειδιά να έχουν μία ονομασία για να γίνονται ευκολότερα αντιληπτά μηνύματα λάθους της MySQL

**Τροποποίηση κλειδιών με το MySQL Workbench**

- Μπορούμε να τροποποιήσουμε έναν πίνακα και με το MySQL Workbench.
- Επιλέγουμε τον πίνακα κάνουμε τις τροποποιήσεις μας και έπειτα κάνουμε Apply τις αλλαγές μας.

**Παράδειγμα 3: alterkeys.sql**

```
ALTER TABLE salaries
```

```
DROP PRIMARY KEY;
```

```
ALTER TABLE salaries
```

```
ADD PRIMARY KEY (emp_no, from_date);
```

```
ALTER TABLE salaries
```

```
DROP FOREIGN KEY salaries_fk_emp;
```

```
ALTER TABLE salaries
```

```
ADD CONSTRAINT salaries_fk_emp FOREIGN KEY (emp_no)  
REFERENCES employees(emp_no) ON DELETE CASCADE;
```

**Άσκηση 3:**

Τροποποιήστε το script της άσκησης 2, ώστε όλα τα κλειδιά να έχουν όνομα.

**Περιορισμοί Εγκυρότητας (CHECK constraints):**

- Είναι περιορισμοί που εξασφαλίζουν ότι μη έγκυρες τιμές δεν θα εισαχθούν σε στήλες του πίνακα.
- Α' τρόπος: Ορίζοντας το ως χαρακτηριστικό κατά τη δημιουργία του πίνακα:

```
column datatype ... CHECK(condition)
```

- Η συνθήκη είναι λογική έκφραση που αφορά την τιμή της στήλης.
- Β' τρόπος: Να έπεται του ορισμού των στηλών:

```
[CONSTRAINT [name]] CHECK (condition);
```

- Η συνθήκη είναι λογική έκφραση που αφορά πολλές στήλες.

**Παράδειγμα 4: check.sql**

```
CREATE TABLE employees (  
...  
birth_date DATE NOT NULL CHECK(birth_date>='1950-01-01'),  
...  
CONSTRAINT date_constraint CHECK (birth_date<hire_date),  
CONSTRAINT last_name CHECK (LENGTH(last_name)>4)  
);
```

**Υπολογιζόμενες Στήλες (generated columns):**

- Είναι στήλες που υπολογίζονται αυτόματα από τα δεδομένα προηγούμενων στηλών του ίδιου πίνακα
- Ορισμός: Ορίζοντας τη στήλη κατά τη δημιουργία του πίνακα:

```
column datatype ... AS (definition)
```

- Ο ορισμός πρέπει να είναι ένας απλός υπολογισμός που αφορά προηγούμενες στήλες.

**Παράδειγμα 5: generated.sql**

```
CREATE TABLE employees (  
...  
first_name VARCHAR(30) NOT NULL,  
last_name VARCHAR(50) NOT NULL,  
...  
full_name VARCHAR(80) AS (CONCAT(first_name,' ',last_name)),  
...  
);
```

**Παρατήρηση:**

- Αν και ελκυστικές, οι υπολογιζόμενες στήλες περιπλέκουν την ατομικότητα των δεδομένων και δεν θα τις χρησιμοποιούμε στην πράξη.
- Και μία υπενθύμιση - **άσκηση**: Ξανατρέξτε το script της άσκησης 3, και έπειτα εξάγετε το EER (βλέπε μάθημα 0: Μέρος 6/7)

**Σύνολα Χαρακτήρων και collations:**

- (Για μία σύντομη ανάλυση των κωδικοποιήσεων χαρακτήρων κ.λπ., βλέπε Python Μάθημα 9, μέρος 11/11: Κωδικοποιήσεις)
- **Character Set: Σύνολο χαρακτήρων**
  - **latin1**: Ιδανικό για για αγγλικούς χαρακτήρες (1B/char)
  - **utf8**: Περιέχει “τα πάντα” (και ελληνικά) (<=3B/char)
- **Collation**: Τρόπος ταξινόμησης των χαρακτήρων του character set:
  - \* **bin**: Με βάση την αριθμητική τους κωδικοποίηση
  - \* **unicode ci**: Ταξινόμηση που δεν είναι case sensitive

**Ορισμός Character Set και Collation:**

- Σε επίπεδο βάσης δεδομένων:

```
CREATE DATABASE [IF NOT EXISTS] database_name
    CHARSET charset_name COLLATE collation_name;
```

- Σε επίπεδο πίνακα:

```
CREATE TABLE [database_name.]table_name
(
    ...
) CHARSET charset_name COLLATE collation_name;
```

- Σε επίπεδο στήλης πίνακα (σε εντολή CREATE TABLE ως χαρακτηριστικό στήλης):

```
...
col1 ... CHARSET charset_name COLLATE collation_name;
...
```

**Συνήθως:**

- Θέτουμε στη βάση δεδομένων μας ένα default CHARSET και COLLATION
- Έπειτα, αν χρειάζεται, θέτουμε σε κάποιον συγκεκριμένο πίνακα ή κάποια συγκεκριμένη στήλη άλλες κωδικοποιήσεις/ταξινομήσεις.

**Σημαντική Σημείωση:**

- Το unicode είναι διαρκώς σε εξέλιξη.
- Η πιο πρόσφατη έκδοση και μάλλον η καλύτερη που μπορούμε να χρησιμοποιήσουμε είναι η “utf8mb4” με collate “utf8mb4\_0900\_ai\_ci” που δουλεύει σωστά και με ελληνικά.

**Παράδειγμα 6: charset.sql**

Τροποποίηση του script ώστε να είναι default η utfmb4:

```
CREATE DATABASE emp
    CHARSET 'utf8mb4' COLLATE 'utf8mb4_unicode_ci';
...
```

**Άσκηση 4:**

Επαληθεύστε (με εισαγωγές δεδομένων) ότι η ταξινόμηση λειτουργεί σωστά με μικρά/κεφαλαία, ακόμη κι αν το 1ο γράμμα έχει τόνο).

**Παρατήρηση:**

- Τα CHARSET/COLLATE μπορούν να ενσωματωθούν σε εντολές ALTER TABLE και τροποποιούνται και μέσω του MySQL Workbench



**Μηχανισμοί Αποθήκευσης (Storage Engines):**

- Ο default μηχανισμός είναι ο **InnoDB**. Υποστηρίζει:
  - **Referential Integrity**: Υποστηρίζει τη δημιουργία πινάκων που σχετίζονται με κλειδιά (πρωτεύοντα - ξένα)
  - **MVCC (Multiversion Concurrency Control)**: Ταυτόχρονα, ένας χρήστης μπορεί να κάνει τροποποιήσεις στα δεδομένα και ένας άλλος να τα διαβάζει, χωρίς να προκαλούνται ασυνέπειες (π.χ. να διαβάζονται ημιτελή δεδομένα). Υπάρχουν δηλαδή πολλαπλές εκδοχές των δεδομένων, με συνεπείς εκδοχές ανάλογα με τη χρήση που γίνεται.
  - **Transaction Isolation**: Κάθε σύνδεση στη βάση μπορεί να διαχωρίσει τις ενέργειές της, ώστε να αποτελούν μία αυτόνομη οντότητα που δεν επικαλύπτεται με άλλες συνδέσεις.
- **ACID**:
  - **Atomicity**: Αν αποτύχει ένα σύνολο ενεργειών, τότε είναι δυνατόν να αναιρεθούν οι αλλαγές που προκλήθηκαν στη βάση
  - **Consistency**: Όταν πραγματοποιηθούν οι αλλαγές, η βάση περνά σε μία νέα κατάσταση χωρίς ασυνέπειες.
  - **Isolation**: Κάθε δέσμη ενεργειών δεν έρχεται σε σύγκρουση με άλλες δέσμες ενεργειών
  - **Durability**: Εγγύηση ότι αν ολοκληρωθεί μία δέσμη ενεργειών, θα αποθηκευτεί σωστά στο δίσκο.

**Άλλοι Μηχανισμοί:**

- **MyISAM**: Παρόμοιος με τον InnoDB, διαφέρει στο ότι το κλείδωμα γίνεται σε επίπεδο πίνακα και όχι γραμμής. Χρησιμοποιείται όταν η βάση είναι σχεδόν αποκλειστικά για διάβασμα και υποστηρίζει γεωγραφικά δεδομένα (spatial data) και αναζήτηση σε μεγάλα κείμενα (full text search). Δεν υποστηρίζει ξένα κλειδιά.
  - **MEMORY**: Αποθηκεύει τα δεδομένα στη μνήμη (για γρήγορη απόκριση)
  - **CSV**: Αποθηκεύει αρχεία που οι τιμές τους είναι τιμές διαχωρισμένες με κόμμα.
  - ... και υπάρχουν άλλοι 12 τύποι μηχανισμοί αποθήκευσης
- Αλλαγή του μηχανισμού αποθήκευσης (σε επίπεδο πίνακα):

```
CREATE TABLE [database_name.]table_name  
(  
    ...  
) ENGINE = MyISAM;
```

**Άσκηση 5:**

Εξάγετε τον ορισμό του πίνακα emp (δεξί κλικ στον πίνακα -> copy to clipboard -> create statement) και μελετήστε τον.