# Laborator 3 & 4

## Crearea, ştergerea şi modificarea tabelelor. Utilizarea comenzilor CREATE TABLE, DESCRIBE table, ALTER TABLE, DROP TABLE, RENAME TABLE, SHOW TABLES

După cum s-a discutat în lucrarea de laborator anterioara, o baza de date este o colectie de informații dintr-un anumit domeniu. Informatia într-o baza de date este structurată sub forma de tabele.

În limbajul SQL lucrul cu tabele este materializat printr-o serie de instructiuni pentru creare, modificare, ştergere şi interogare a tabelelor şi înregistrărilor din tabele.

În lucrarea de laborator de astazi vom parcurge câteva instructiuni de creare, ştergere, modificare şi redenumire de tabele.

### 1. Instrucțiunea CREATE TABLE.

Formatul general al acestei instrucțiuni este:

CREATE TABLE [IF NOT EXISTS] tbl_nume [(definiții de creare,...)]
[PRIMARY KEY] [reference_definition]

Instrucțiunea CREATE TABLE, crează un tabel cu numele specificat în *tbl_nume* iar crearea se face în baza de date curentă (selectată cu comanda USE). De reținut este faptul că numele tabelului poate sa cuprinda orice caracter cu exceptia caracterelor \, /, ., :, *, “, ?, <, >.

Un mesaj de eroare va fi afişat în cazul în care nici o baza de date nu a fost selectată anterior utilizarii. Ca şi în cazul creării bazelor de date, clauza IF NOT EXISTS va preveni apariția mesajului de eroare daca un alt tabel cu numele *tbl_nume* a fost creat anterior.

Definițiile de creare trebuie să se încadreze în următorul format:

col_nume type [NOT NULL | NULL] [DEFAULT default_value] [AUTO_INCREMENT]

Unde

- col_nume este numele coloanei ce urmează a fi definită in tabel
- type este tipul de date ce defineste col_nume
- opțiunile NULL sau NOT NULL specifică dacă o valoare trebuie sau nu specificată în coloana respectivă. Dacă nici una din valori nu este

specificată, atunci se va considera valoarea NULL a fi implicita, adică nu este necesară specificarea unei valori.

- DEFAULT value reprezinta o valoare ce va fi inregistrată în coloana respectivă în situația în care nici o valoare nu este specificată. Valoarea precizată in default_value trebuie sa fie o constantă şi nu poate fi o funcție sau o expresie.
- pentru tipurile de date numerice, opțiunea AUTO_INCREMENT specifica faptul ca la inserarea unei noi inregistrări in tabel, valoarea coloanei va fi incrementată. Coloana marcată ca AUTO_INCREMENT trebuie sa fie cheie primara in tabel şi trebuie specificata in definiția cheii primare
- PRIMARY KEY arată că, coloana specificată este cheie unică în tabel. De retinut este faptul că, coloana marcată ca PRIMARY KEY trebuie specificată ca NOT NULL.

Ca un exemplu pentru aceasta instrucțiune, vom reconsidera exemplul din lucrarea de laborator anterioara pentru crearea unui tabel cu datele studentilor din facultate.

```
CREATE DATABASE IF NOT EXISTS facultate;
USE facultate;
CREATE TABLE IF NOT EXISTS student (
        nrleg int NOT NULL AUTO_INCREMENT primary key,
        nume varchar(20) not null,
        prenume varchar(20) not null,
        init char(1),
        sex char(1) default 'm' not null,
        datanastere date);
```

Tabelul creat in acest exemplu cuprinde urmatoarele campuri: nrleg, nume, prenume, init, sex şi datanastere.

Pentru:

- nrleg, tipul de date este intreg ce nu poate lua valori nule si se incrementează automat iar acest câmp este cheie primară în tabel;
- nume, tipul de date este şir de caractere cu lungimea de 20 ce nu poate lua valori nule;
- prenume, tipul de date este şir de caractere cu lungimea de 20 ce nu poate lua valori nule;
- init, tipul de date este caracter cu lungimea de 1 care are ca valoare implicită valoarea ‘m’
- datanastere, tipul de date este dată calendaristică de forma YYYY-MM-DD

## 2. Instrucțiunea DESCRIBE.

Formatul general al acestei instrucțiuni este:

{DESCRIBE | DESC} tbl_nume

Instrucțiunea DESCRIBE, afisează informații despre coloanele tabelului specificat in *tbl_nume.*

Instrucțiunea poate fi scrisa prescurtat DESC şi are acelaşi efect.

Exemplu:

DESCRIBE student;

Această comandă va afişa următorul rezultat:

| Field | Type | NULL | Key | Default | Extra |
|---|---|---|---|---|---|
| nrleg | int(11) | | PRI | [NULL] | auto_increment |
| nume | varchar(20) | | | | |
| prenume | varchar(20) | | | | |
| init | char(1) | YES | | [NULL] | |
| sex | char(1) | YES | | m | |
| datanastere | date | YES | | [NULL] | |

## 3. Instrucțiunea ALTER TABLE.

Formatul general al instrucțiunii ALTER TABLE este:

ALTER TABLE tbl_nume alter_spec [, alter_spec ...]

Această instrucțiune da posibilitatea utilizatorului de a modifica structura unui table deja creat. Pentru exemplu, această instrucțiune poate adăuga sau şterge coloane, crea sau distruge indexi, să schimbe tipul unei coloane existente sau să redenumească coloane sau tabelul insuşi.

Formatul specificațiilor de tabel este:

alter_spec:

ADD [COLUMN] create_definition [FIRST | AFTER column_name ]
sau ADD [COLUMN] (create_definition, create_definition,...)
sau ADD INDEX [index_name] (index_col_name,...)
sau ADD PRIMARY KEY (index_col_name,...)
sau ALTER [COLUMN] col_name {SET DEFAULT literal | DROP DEFAULT}
sau CHANGE [COLUMN] old_col_name create_definition
sau MODIFY [COLUMN] create_definition [FIRST | AFTER column_name]
sau DROP [COLUMN] col_name

sau RENAME [TO] new_tbl_name

După executarea comenzii ALTER TABLE, se va crea o copie a tabelului original iar alterarea este executată pe copie. În momentul terminării execuției, tabelul original este şters şi copia va fi redenumită cu numele său. Aceasta tehnică este folosită pentru a preveni orice eroare de modificare a tabelului.

**Descrierea specificațiile instrucțiunii ALTER TABLE**

- opțiunea **ADD [COLUMN] create_definition [FIRST | AFTER column_name ]** – adaugă o coloana nouă în tablelul specificat în *tbl_nume*. Coloana adaugată poate fi introdusă la începutul tabelului cu opțiunea FIRST sau după coloana specificată în column_name din opțiunea AFTER

*Ex:*

```
ALTER TABLE student ADD new INT DEFAULT 0 AFTER nrleg;
```

Comanda de mai sus va adăuga în tabelul student o nouă coloana cu numele new de tip întreg cu valoare implicită 0, iar coloana new va fi introdusă dupa coloana nrleg.

- opțiunea **ADD INDEX [index_name] (index_col_name,...)** – are ca efect adăugarea unui nou index tabelului specificat în tbl_nume.

*Ex:*

```
ALTER TABLE student ADD INDEX new_index (new);
```

Comanda de mai sus va adăuga un nou index tabelului prin câmpul new iar indexul se va numi new_index.

- opțiunea **ADD PRIMARY KEY (index_col_name,...)** – va adauga o cheie primară tabelului specificat în tbl_nume.

*Ex:*

```
CREATE TABLE IF NOT EXISTS student1 (
        nrleg int NOT NULL,
        new int NOT NULL,
        nume varchar(20) not null,
        prenume varchar(20) not null,
        init char(1),
        sex char(1) default 'm' not null,
        datanastere date);
ALTER TABLE student1 ADD PRIMARY KEY (new);
```

Comenzile de mai sus vor crea in prima linie un nou tabel cu numele student1 şi apoi va adăuga coloana new ca fiind cheie primară a tabelului student1.

- opțiunea **ALTER [COLUMN] col_name {SET DEFAULT literal | DROP DEFAULT}** – adaugă, modifică sau sterge o valoare default pentru câmpul specificat in col_name din tabelul tbl_nume.

*Ex:*

ALTER TABLE student ALTER nrleg SET DEFAULT 5

Comanda de mai sus va seta in tabelul student pentru campul nrleg valoarea implicită 5

ALTER TABLE student ALTER nrleg SET DEFAULT 2

Comanda de mai sus va modifica in tabelul student pentru campul nrleg valoarea implicită 2

ALTER TABLE student ALTER nrleg DROP DEFAULT

Comanda de mai sus va sterge in tabelul student pentru campul nrleg valoarea implicită.

- opțiunea **CHANGE [COLUMN] old_col_name create_definition** – schimba numele sau tipul coloanei specificate in old_col_name.

*Ex:*

ALTER TABLE student CHANGE new new1 varchar(3);

Comanda de mai sus va schimba in tabelul student numele coloanei new in new1 iar tipul coloanei din integer va deveni sir de caractere de 3.

ALTER TABLE student CHANGE new1 new1 INT;

Comanda de mai sus va schimba de aceasta data numai tipul de date al campului new1 din sir de caractere in intreg.

- opțiunea **MODIFY [COLUMN] create_definition [FIRST | AFTER column_name]** – modifică definirea de tip a câmpului specificat în create_definition. Cu opțiunile FIRST sau AFTER se poate muta coloana specificată în prima poziție respectiv după coloana specificată în column_name.

*Ex:*

ALTER TABLE student MODIFY nume varchar(10) FIRST

Comanda de mai sus modifică tipul câmpului nume pe prima poziție din tabel şi schimbă tipul acesteia în şir de caractere cu lungimea de 10.

- opțiunea **DROP [COLUMN] col_name** – cauzează ştergerea coloanei specificată în col_name din tabelul tbl_nume.

*Ex:*

ALTER TABLE student DROP nume;

Comanda de mai sus şterge coloana nume din tabelul student.

- opțiunea **RENAME [TO] new_tbl_name** – cauzează redenumirea tabelului tbl_nume cu numele new_tbl_name

*Ex:*

ALTER TABLE student RENAME student_new

Comanda de mai sus modifică numele tabelului student în student_new

**4. Instrucțiunea DROP TABLE.**

Formatul general al acestei instrucțiuni este:

DROP TABLE [IF EXISTS] tbl_name [, tbl_name,...]

Această instrucțiune sterge unul sau mai multe tabele specificate în tbl_name.

Clauza IF EXISTS asociata acestei instrucțiuni permite ocolirea mesajului de eroare în situația în care tbl_name nu există în baza de date curentă.

*Ex:*

DROP TABLE IF EXISTS student1, student2, student3;

Comanda de mai sus sterge (daca există în baza de date curentă) tabelele student1, student2 şi student3.

**5. Instrucțiunea RENAME TABLE.**

Formatul general al acestei instrucțiuni este:

RENAME TABLE tbl_name TO new_tbl_name[, tbl_name2 TO new_tbl_name2,...]

Această instrucțiune redenumeşte tabelul tbl_name în new_tbl_name. Spre deosebire de comanda alter cu opțiunea rename care face o copie a tabelului înaintea executării instrucțiunii, această instrucțiune modifică direct numele tabelului cu noul nume.

Ex:

RENAME TABLE student_new TO student;

Comanda de mai sus redenumeşte tabelul student_new cu numele student.

**6. Instrucțiunea SHOW TABLES**

Formatul general al acestei instrucțiuni este simplu SHOW TABLES fără nici un argument. Efectul acestei instrucțiuni este acela de a afişa toate tabelele create din baza de date curentă.

## Teme de laborator

1. Să se creeze o bază de date pentru Facultatea de Automatică ce va cuprinde următoarele informații:
   a. informații despre fiecare student în parte,
   b. repartizarea pe grupe şi subgrupe

**Observație:** Pentru declararea tipurilor de date din tabele se vor folosi următoarele:

int - pentru valori intregi
char(1) – pentru valori de tip caracter
varchar(marime) – pentru siruri de caractere cu lungimea “marime”
date – pentru valori de tip dată calendaristică

După crearea tabelelor ce vor construi baza de date, se va alege la întâmplare un tabel în care:

a. se va adăuga un câmp cu numele **camp_nou** de tipul intreg nenul care sa fie penultima coloană din tabel cu valoare implicită 4;
b. se va adăuga un index cu numele index1 la tabelul ales pentru coloana camp_nou;
c. se va şterge valoare default a campului nou creat;
d. se va schimba coloana creată pentru a fi pe ultima poziție din tabel;
e. se va redenumi coloana camp_nou cu camp_redenumit iar după redenumire se va schimba tipul in şir de caractere de 10;
f. se va şterge penultima coloană din tabel;
g. se va crea un nou tabel cu structura identică cu tabelul modificat la punctele anterioare şi acesta va avea numele primului + _nou;
h. tabelului nou creat i se va adăuga o nouă coloana cu numele descriere_tabel_nou;
i. cu ajutorul unei singure linii de comandă se vor interschimba numele celor doua tabele;
j. cu ajutorul unei singure linii de comandă se vor sterge ambele tabele.