

Ministerul Educației al Republicii Moldova  
Universitatea Tehnică a Moldovei  
Facultatea Calculatoare, Informatică și Microelectronică  
Ingineria Software

# **REPORT**

## **Laboratory work Nr.6**

### **Creating tables and indexes**

Realized by: Popa Eugeniu  
FAF-202

Chișinău, 2021

## The theoretical part

1. Basic syntax of the CREATE TABLE statement.

```
CREATE TABLE <table_name> (  
  <column1> <data_type1> [DEFAULT <expr1>] [<column_construction1>]  
  [, <column2> <data_type2> [DEFAULT <expr2>] [<column_construction1>], ...]  
  [<table_constructions>] );
```

2. Basic syntax of the ALTER TABLE statement.

```
ALTER TABLE table ADD(column data_type [DEFAULT expr.] [, column  
data_type ...]);  
ALTER TABLE table ALTER(column data_type [DEFAULT expr.] [, column  
data_type ...]);  
ALTER TABLE table DROP(column [, column...]);
```

3. Basic syntax of the INSERT statement.

```
INSERT INTO TABLE_NAME(column1, column2, column3, ... columnN)]  
VALUES(value1, value2, value3, ... valueN);
```

4. Basic syntax of the DELETE statement.

```
DELETE FROM table_name  
WHERE [condition];
```

5. Basic syntax of the UPDATE statement.

```
UPDATE table_name  
SET column1 = value1, column2 = value2 ...., columnN = valueN  
WHERE [condition];
```

6. The difference between the DELETE and TRUNCATE TABLE statements.

The DELETE statement removes rows one at a time and records an entry in the transaction log for each deleted row. TRUNCATE TABLE removes the data by deallocating the data pages used to store the table data and records only the page deallocations in the transaction log.

7. Types of indexes supported by MS SQL Server 2019.

Clustered	Filtered
Non-clustered	Full-Text
Covering Indexes	Primary XML and secondary XML Indexes
Unique	Column-store Indexes
Indexed views	Spatial

8. Basic syntax of the CREATE INDEX statement.

```
CREATE INDEX "index_name" ON "table_name" (column_name);
```

## The practical part

```
--1. Sa se scrie o instructiune T-SQL, care ar popula coloana Adresa _ Postala _ Profesor
--din tabelul profesori cu valoarea 'mun. Chisinau', unde adresa este necunoscuta.

UPDATE profesori
SET Adresa_Postala_Profesor = 'mun. Chisinau'
WHERE Adresa_Postala_Profesor IS NULL;
SELECT * FROM profesori

--2. Sa se modifice schema tabelului grupe, ca sa corespunda urmatoarelor cerinte:
--a). Campul Cod_Grupa sa accepte numai valorile unice si sa nu accepte valori necunoscute.
--b). Sa se tina cont ca cheie primara, deja, este definita asupra coloanei Id_Grupa.

ALTER TABLE grupe
ADD UNIQUE (Cod_Grupa);
ALTER TABLE grupe
ALTER COLUMN Cod_Grupa CHAR(6) NOT NULL
SELECT * FROM grupe

--3. La tabelul grupe, sa se adauge 2 coloane noi Sef_grupa si Prof_Indrumator, ambele de tip INT. Sa se populeze campurile nou-create
--cu cele mai potrivite candidaturi in baza criteriilor de mai jos:
--a). Seful grupei trebuie sa aiba cea mai buna reusita (medie) din grupa la toate formele de evaluare si la toate disciplinele.
--Un student nu poate fi sef de grupa la mai multe grupe.
--b). Profesorul indrumator trebuie sa predea un numar maximal posibil de discipline la grupa data. Daca nu exista o singura candidat,
--care corespunde primei cerinte, atunci este ales din grupul de candidati acel cu identificatorul (Id_Profesor) minimal.
--Un profesor nu poate fi indrumator la mai multe grupe.
--c). Sa se scrie instructiunile ALTER, SELECT, UPDATE necesare pentru crearea coloanelor in tabelul grupe, pentru
--selectarea candidatilor si inserarea datelor.

--ALTER TABLE grupe
--ADD Sef_Grupa INT, Prof_Indrumator INT

DECLARE @nr_de_grupe INT = (SELECT COUNT(Id_Grupa) FROM grupe)
DECLARE @index INT = 1

WHILE (@index <= @nr_de_grupe)
BEGIN
    UPDATE grupe
    SET Sef_Grupa = (SELECT TOP(1) Id_Student FROM
    (SELECT Id_Student, AVG(Nota) AS nrNota
    from studenti_reusita where Id_Grupa = @index group by Id_Student) as q1
    order by q1.nrNota desc),

    Prof_Indrumator = (SELECT TOP(1) Id_Profesor FROM
    (SELECT Id_Profesor, COUNT(DISTINCT Id_Disciplina) AS nr_Disciplina
    FROM studenti_reusita WHERE Id_Grupa = @index GROUP BY Id_Profesor) AS q2
    ORDER BY q2.nr_Disciplina DESC) WHERE Id_Grupa = @index;

    SET @index += 1;
END

SELECT * FROM grupe;
--ALTER TABLE grupe DROP COLUMN sef_grupa;
--ALTER TABLE grupe DROP COLUMN prof_indrumator;

--4. Sa se scrie o instructiune T-SQL, care ar mari toate notele de evaluare sefilor de grupe cu un punct. Nota maximala (10) nu poate fi marita.

UPDATE studenti_reusita
SET Nota = Nota + 1
WHERE Nota <> 10 AND Nota IN (SELECT Nota
FROM studenti_reusita
WHERE Id_Student IN
(SELECT Sef_Grupa
FROM grupe));
SELECT * FROM studenti_reusita;
```

```

--5. Sa se creeze un tabel profesori_new, care include urmatoarele coloane:
-- Id_Profesor, Nume_Profesor, Prenume_Profesor, Localitate, Adresa_1, Adresa_2.
--a). Coloana Id_Profesor trebuie sa fie definita drept cheie primara si, in baza ei, sa fie construit un index CLUSTERED.
--b). Campul Localitate trebuie sa posede proprietatea DEFAULT = 'mun. Chisinau'.
--c). Sa se insereze toate datele din tabelul profesori si tabelul profesori_new.
--Sa se scrie, cu acest scop, un numar potrivit de instructiuni T-SQL. Datele trebuie sa fie transferate in felul urmator:
--Coloana-sursa      Coloana-destinatie
--Id Profesor        Id Profesor
--Nume Profesor       Nume Profesor
--Prenume Profesor    Prenume Profesor
--Adresa Postala Profesor Localitate
--Adresa Postala Profesor Adresa 1
--Adresa Pastala Profesor Adresa 2
--In coloana Localitate sa fie inserata doar informatia despre denumirea localitatii din coloana-sursa Adresa_Postala_Profesor.
--In coloana Adresa_1, doar denumirea strazii. in coloana Adresa_2, sa se pastreze numarul casei si (posibil) a apartamentului.

CREATE TABLE profesori_new(
  Id_Profesor INT NOT NULL,
  Nume_Profesor CHAR(255),
  Prenume_Profesor CHAR(255),
  Localitate CHAR(255) DEFAULT('mun. Chisinau'),
  Adresa_1 CHAR(255),
  Adresa_2 CHAR(255),
  CONSTRAINT [PK_profesori_new] PRIMARY KEY CLUSTERED
  (Id_Profesor)) ON [PRIMARY];

INSERT INTO profesori_new (Id_Profesor, Nume_Profesor, Prenume_Profesor, Localitate, Adresa_1, Adresa_2)
  (SELECT Id_Profesor, Nume_Profesor, Prenume_Profesor, Adresa_Postala_Profesor, Adresa_Postala_Profesor, Adresa_Postala_Profesor
  from profesori)

SELECT * FROM profesori_new;

UPDATE profesori_new
  SET Localitate = (CASE WHEN CHARINDEX(' ', Localitate) > 0
    THEN CASE WHEN CHARINDEX('str.', Localitate) > 0
      THEN SUBSTRING(Localitate, 1, CHARINDEX('str.', Localitate) - 1)
    WHEN CHARINDEX('bd.', Localitate) > 0
      THEN SUBSTRING(Localitate, 1, CHARINDEX('bd.', Localitate) - 1)
    END
    WHEN CHARINDEX('or.', Localitate) > 0
      THEN CASE WHEN CHARINDEX('str.', Localitate) > 0
        THEN SUBSTRING(Localitate, 1, CHARINDEX('str.', Localitate) - 3)
      WHEN CHARINDEX('bd.', Localitate) > 0
        THEN SUBSTRING(Localitate, 1, CHARINDEX('bd.', Localitate) - 3)
      END
    WHEN CHARINDEX('mun.', Localitate) > 0
      THEN SUBSTRING(Localitate, 1, CHARINDEX('nau', Localitate) + 2)
    END),
  Adresa_1 = (CASE WHEN CHARINDEX('str.', Adresa_1) > 0
    THEN SUBSTRING(Adresa_1, CHARINDEX('str.', Adresa_1), PATINDEX('%[0-9]%', Adresa_1) - CHARINDEX('str.', Adresa_1))
    WHEN CHARINDEX('bd.', Adresa_1) > 0
    THEN SUBSTRING(Adresa_1, CHARINDEX('bd.', Adresa_1), PATINDEX('%[0-9]%', Adresa_1) - CHARINDEX('bd.', Adresa_1))
    END),
  Adresa_2 = (CASE WHEN PATINDEX('%[0-9]%', Adresa_2) > 0
    THEN SUBSTRING(Adresa_2, PATINDEX('%[0-9]%', Adresa_2) + 1, len(Adresa_2) - PATINDEX('%[0-9]%', Adresa_2) + 1)
    END)

```

```

--6. Sa se insereze datele in tabelul orarul pentru Grupa = 'CIB171' (Id_Grupa= 1) pentru ziua de luni. Toate lectiile vor avea loc in blocul de studii 'B'.
--Mai jos, sunt prezentate detaliile de insertare: (Id_Disciplina = 107, Id_Profesor = 101, Ora = '08:00', Auditoriu = 202);
--(Id_Disciplina = 108, Id_Profesor = 101, Ora = '11:30', Auditoriu = 501); (Id_Disciplina = 119, Id_Profesor = 117, Ora = '13:00', Auditoriu = 501);

CREATE TABLE orarul (Id_Disciplina INT, Id_Profesor INT, Id_Grupa INT DEFAULT(1), Zi CHAR(255), Ora TIME, Auditoriu INT,
Bloc CHAR(1) DEFAULT('B'), PRIMARY KEY(Id_Grupa, Zi, Ora))

INSERT INTO orarul (Id_Disciplina, Id_Profesor, Zi, Ora, Auditoriu)
VALUES(107, 101, 'Luni', '08:00', 202)
INSERT INTO orarul (Id_Disciplina, Id_Profesor, Zi, Ora, Auditoriu)
VALUES(108, 101, 'Luni', '11:30', 501)
INSERT INTO orarul (Id_Disciplina, Id_Profesor, Zi, Ora, Auditoriu)
VALUES(109, 117, 'Luni', '13:00', 501)

SELECT * FROM orarul;

--DROP TABLE orarul;

--7. Sa se scrie expresiile T-SQL necesare pentru a popula tabelul orarul pentru grupa INF171, ziua de luni.
--Datele necesare pentru inserare trebuie sa fie colectate cu ajutorul instructiunii/instructiunilor SELECT si
--introduse in tabelul-destinatie, stiind ca:
--lectie #1 (Ora = '08:00', Disciplina = 'Structuri de date si algoritmi', Profesor = 'Bivol Ion')
--lectie #2 (Ora = '11 :30', Disciplina = 'Programe aplicative', Profesor = 'Mircea Sorin')
--lectie #3 (Ora = '13:00', Disciplina = 'Baze de date', Profesor = 'Micu Elena')

INSERT INTO orarul (Id_Disciplina, Id_Profesor, Id_Grupa, Zi, Ora)
VALUES((SELECT Id_Disciplina FROM disciplina WHERE Disciplina = 'Structuri de date si algoritmi'),
(SELECT Id_Profesor FROM profesori WHERE Nume_Profesor = 'Bivol' AND Prenume_Profesor = 'Ion'),
(SELECT Id_Grupa FROM grupe WHERE Cod_Grupa = 'INF171'), 'Luni', '08:00')

INSERT INTO orarul (Id_Disciplina, Id_Profesor, Id_Grupa, Zi, Ora)
VALUES((SELECT Id_Disciplina FROM disciplina WHERE Disciplina = 'Programe aplicative'),
(SELECT Id_Profesor FROM profesori WHERE Nume_Profesor = 'Mircea' AND Prenume_Profesor = 'Sorin'),
(SELECT Id_Grupa FROM grupe WHERE Cod_Grupa = 'INF171'), 'Luni', '11:30')

INSERT INTO orarul (Id_Disciplina, Id_Profesor, Id_Grupa, Zi, Ora)
VALUES((SELECT Id_Disciplina FROM disciplina WHERE Disciplina = 'Baze de date'),
(SELECT Id_Profesor FROM profesori WHERE Nume_Profesor = 'Micu' AND Prenume_Profesor = 'Elena'),
(SELECT Id_Grupa FROM grupe WHERE Cod_Grupa = 'INF171'), 'Luni', '13:00')

--8. Sa se scrie interogari de creare a indecsilor asupra tabelelor din baza de date universitatea pentru a asigura o performanta sporita la executarea interogarilor
--SELECT din Lucrarea practica 4. Rezultatele optimizarii sa fie analizate in baza planurilor de executie, pana la si dupa crearea indecsilor.
--Indecsii nou creati sa fie plasati fizic in grupul de fisiere userdatafgroup1 (Crearea si intretinerea bazei de date - sectiunea 2.2.2)

ALTER DATABASE universitatea ADD FILE( NAME = Indexes, FILENAME = 'd:\indexes_universitateaDB.ndf', SIZE = 1MB)
TO FILEGROUP userdatafgroup1 GO

DROP INDEX pk_discipline ON disciplina

CREATE NONCLUSTERED INDEX pk_id_disciplina ON disciplina (id_disciplina)

DROP INDEX pk_grupe ON grupe

CREATE NONCLUSTERED INDEX pk_id_grupa ON grupe (id_grupa)

DROP INDEX pk_profesori ON profesori

CREATE NONCLUSTERED INDEX pk_id_profesor ON profesori (id_profesor)

DROP INDEX pk_studenti ON studenti

CREATE NONCLUSTERED INDEX pk_id_student ON studenti (id_student)

```