

Databases Laboratory Work Nr 6

Title: Tables and indices creation

Prerequisites: SQL Server 2019 and SSMS

Objectives: Get practical experience in creation of tables and indices.

Tasks:

1. Sa se scrie o instructiune T-SQL, care ar popula co Joana Adresa _ Postala _ Profesor din tabelul *profesori* cu valoarea 'mun. Chisinau', unde adresa este necunoscuta.
2. Sa se modifice schema tabelului *grupe*, ca sa corespunda urmatoarelor cerinte:
 - a) Campul *Cod_ Grupa* sa accepte numai valorile unice ~i sa nu accepte valori necunoscute.
 - b) Sa se tina cont ca cheie primarii, deja, este definitii asupra coloanei *Id_ Grupa*.
3. La tabelul *grupe*, sa se adauge 2 coloane noi *Sef_grupa* ~i *Prof_Indrumator*, ambele de tip *INT*. Sii se populeze campurile nou-create cu cele mai potrivite candidaturi ill baza criteriilor de maijos:
 - a) *Seful grupei* trebuie sa aiba cea mai buna reu~itii (medie) din grupa la toate formele de evaluare ~i la toate disciplinele. Un student nu poate fi ~ef de grupa la mai multe grupe.
 - b) *Profesorul fndrumator* trebuie sa predea un numiir maximal posibil de discipline la grupa data. Daca nu existii o singurii candidaturii, care corespunde primei cerinte, atunci este ales din grupul de candidati acel cu identificatorul (*Id_Profesor*) minimal. Un profesor nu poate fi illdrumator la mai multe grupe.
 - c) Sii se scrie instructiunile *ALTER*, *SELECT*, *UPDATE* necesare pentru crearea coloanelor in tabelul *grupe*, pentru selectarea candidatilor ~i inserarea datelor . .
4. Sa se scrie o instructiune T-SQL, care ar mari toate notele de evaluare ~efilor de grupe cu un punct. Nota maximala (10) nu poate fi miirita.
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 ~i, ill baza ei, sa fie construit un index *CLUSTERED*.
 - b) Campul *Localitate* trebuie sa posede proprietatea *DEFA ULT= 'mun. Chisinau'*.
 - c) Sa se insereze toate datele din tabelul *profesori* ill tabelul *profesori_new*. Sa se scrie, cu acest scop, un numiir potrivit de instructiuni T-SQL. Datele trebuie sa fie transferate in felul urmator:

Coloana-sursă	Coloana-destinație
<i>Id Profesor</i>	<i>Id Profesor</i>
<i>Nume Profesor</i>	<i>Nume Profesor</i>
<i>Prenume Profesor</i>	<i>Prenume Profesor</i>
<i>Adresa Postala Profesor</i>	<i>Localitate</i>
<i>Adresa Postala Profesor</i>	<i>Adresa 1</i>
<i>Adresa Postala Profesor</i>	<i>Adresa 2</i>

in coloana *Localitate* sii fie inserata doar informatia despre denumirea localitiitii din coloana-sursa *Adresa_Postala_Profesor*. in coloana *Adresa_1*, doar denumirea striizii. in coloana *Adresa_2*, sii se piistreze numiirul casei ~i (posibil) a apartamentului.

6. Sa se insereze datele in tabelul *orarul* pentru *Grupa= 'CIBJ 71'* (*Id_ Grupa= 1*) pentru ziua de luni. Toate lectiile vor avea loc ill blocul de studii 'B'. Mai jos, sunt prezentate detaliile de

inserare:

```
(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);
```

7. Sii 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* ~i introduse in tabelul-destinatie, ~tiind 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')

8. Sa se scrie interogariile de creare a indecilor 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 ~i dupa crearea indecilor.

Indecii nou-creati sa fie plasati fizic in grupul de fiere *userdatafgroup1* (Crearea \$i intrefinerea bazei de date - sectiunea 2.2.2)

Implementation

```
USE universitatea;
```

```
/* Task 1 */
```

```
BEGIN
```

```
UPDATE profesori
```

```
SET Adresa_Postala_Profesor = 'mun. Chisinau'
```

```
WHERE Adresa_Postala_Profesor IS NULL
```

```
SELECT * FROM profesori
```

```
END
```

```
/* Task 2 */
```

```
BEGIN
```

```
ALTER TABLE grupe
```

```
ALTER COLUMN Cod_Grupa VARCHAR(255) NOT NULL
```

```
ALTER TABLE grupe
```

```
ADD CONSTRAINT UQ_Cod_Grupa UNIQUE (Cod_Grupa)
```

```
ALTER TABLE grupe
```

```
ADD CONSTRAINT PK_grupe PRIMARY KEY (Id_Grupa)
```

```
SELECT * FROM grupe
```

```
END
```

```
/* Task 3 */
```

```
BEGIN
```

```
ALTER TABLE grupe
```

```
ADD Sef_grupa INT
```

```
ALTER TABLE grupe
```

```
ADD CONSTRAINT UQ_Sef_Grupa UNIQUE (Sef_Grupa)
```

```
ALTER TABLE grupe
```

```
ADD Prof_Indrumator INT
```

```

UPDATE grupe
SET Sef_grupa = (SELECT TOP(1) Id_Student, Id_Grupa FROM studenti_reusita
                WHERE studenti_reusita.Id_Grupa = grupe.Id_Grupa
                GROUP BY Id_Student
                ORDER BY SUM(Nota) DESC)

UPDATE grupe
SET Prof_Indrumator = (SELECT TOP(1) Id_Profesor FROM (SELECT TOP(1000) Id_Profesor FROM
studenti_reusita

                WHERE studenti_reusita.Id_Grupa = grupe.Id_Grupa

                GROUP BY Id_Profesor, Id_Grupa

                ORDER BY COUNT(DISTINCT Id_Disciplina) DESC, Id_Profesor DESC) AS temp)

SELECT * FROM grupe

END

/* Task 4 */
BEGIN

UPDATE studenti_reusita
SET Nota = Nota + 1
WHERE Nota < 10 AND Id_Student IN (SELECT Sef_grupa FROM grupe)

END

/* Task 5 */
GO

CREATE FUNCTION dbo.GetEnd (@STRING AS VARCHAR(255), @SEGMENT AS INT)
RETURNS INT
BEGIN
    DECLARE @END INT

    IF @SEGMENT = 1
    BEGIN
        IF (CHARINDEX(',', @STRING) != 0)
            SET @END = CHARINDEX(',', @STRING)
        ELSE
            SET @END = LEN(@STRING) + 1
        END
    ELSE IF @SEGMENT = 2
    BEGIN
        IF (CHARINDEX(',', @STRING) != 0)
            SET @END = CHARINDEX(',', @STRING, CHARINDEX(',', @STRING) + 2) - CHARINDEX(',', @STRING) - 2
        ELSE
            SET @END = 0
        END
    ELSE IF @SEGMENT = 3
    BEGIN
        IF (CHARINDEX(',', @STRING) != 0)
            SET @END = LEN(@STRING) - CHARINDEX(',', @STRING, CHARINDEX(',', @STRING) + 2) - 1
        ELSE
            SET @END = 0
        END
    END

    RETURN @END;
END

```

GO

```
CREATE FUNCTION dbo.GetStart (@STRING AS VARCHAR(255), @SEGMENT AS INT)
RETURNS INT
BEGIN
    DECLARE @START INT

    IF @SEGMENT = 1
        SET @START = 0
    ELSE IF @SEGMENT = 2
        BEGIN
            IF (CHARINDEX(',', @STRING) != 0)
                SET @START = CHARINDEX(',', @STRING) + 2
            ELSE
                SET @START = 0
        END
    ELSE IF @SEGMENT = 3
        BEGIN
            IF (CHARINDEX(',', @STRING) != 0)
                SET @START = CHARINDEX(',', @STRING, CHARINDEX(',', @STRING) + 1) + 2
            ELSE
                SET @START = 0
        END
    END

    RETURN @START;
END
```

GO

```
-- delete function
DROP FUNCTION dbo.GetEnd;
DROP FUNCTION dbo.GetStart;
GO
```

BEGIN

```
CREATE TABLE profesori_new (
    Id_Profesor INT,
    Nume_Profesor VARCHAR,
    Prenume_Profesor VARCHAR,
    Localitate VARCHAR DEFAULT('Mun. Chisinau'),
    Adresa_1 VARCHAR,
    Adresa_2 VARCHAR,
)
```

```
CREATE CLUSTERED INDEX profesori_new_PK_index
ON profesori_new(Id_Profesor)
```

SELECT

```
    Id_Profesor,
    Nume_Profesor,
    Prenume_Profesor,
    SUBSTRING(Adresa_Postala_Profesor, dbo.GetStart(Adresa_Postala_Profesor, 1),
    dbo.GetEnd(Adresa_Postala_Profesor, 1)) as Localitate,
    SUBSTRING(Adresa_Postala_Profesor, dbo.GetStart(Adresa_Postala_Profesor, 2),
    dbo.GetEnd(Adresa_Postala_Profesor, 2)) as Adresa_1,
    SUBSTRING(Adresa_Postala_Profesor, dbo.GetStart(Adresa_Postala_Profesor, 3),
    dbo.GetEnd(Adresa_Postala_Profesor, 3)) as Adresa_2
    INTO profesori_new
    FROM profesori
```

```
select * from profesori_new
```

END

/* Task 6 */

GO

DROP TABLE IF EXISTS orarul;

CREATE TABLE orarul (

Id_Disciplina INT,

Id_Profesor INT,

Id_Grupa INT,

Zi VARCHAR(10),

Ora TIME,

Auditoriu INT,

Bloc VARCHAR)

GO

INSERT INTO orarul VALUES

(107, 101, 1, 'Luni', '08:00', 202, 'B'),

(108, 101, 1, 'Luni', '11:30', 501, 'B'),

(119, 117, 1, 'Luni', '13:00', 501, 'B')

SELECT * FROM orarul

GO

/* Task 7 */

INSERT INTO orarul

SELECT Id_Disciplina,

(SELECT Id_Profesor FROM profesori WHERE Nume_Profesor = 'Bivol' AND
Prenume_Profesor = 'Ion') AS Id_Profesor,

(SELECT TOP(1) Id_Grupa FROM grupe WHERE Cod_Grupa = 'INF171') AS

Id_Grupa,

'Luni' as Zi, '08:00' as Ora, 202 as Auditoriu, 'B' as Bloc

FROM discipline WHERE Disciplina = 'Structuri de date si algoritmi';

INSERT INTO orarul

SELECT Id_Disciplina,

(SELECT Id_Profesor FROM profesori WHERE Nume_Profesor = 'Mircea' AND
Prenume_Profesor = 'Sorin') AS Id_Profesor,

(SELECT TOP(1) Id_Grupa FROM grupe WHERE Cod_Grupa = 'INF171') AS

Id_Grupa,

'Luni' as Zi, '11:30' as Ora, 202 as Auditoriu, 'B' as Bloc

FROM discipline WHERE Disciplina = 'Programe aplicative';

INSERT INTO orarul

SELECT Id_Disciplina,

(SELECT Id_Profesor FROM profesori WHERE Nume_Profesor = 'Micu' AND
Prenume_Profesor = 'Elena') AS Id_Profesor,

(SELECT TOP(1) Id_Grupa FROM grupe WHERE Cod_Grupa = 'INF171') AS

Id_Grupa,

'Luni' as Zi, '13:00' as Ora, 202 as Auditoriu, 'B' as Bloc

FROM discipline WHERE Disciplina = 'Baze de date';

SELECT * FROM orarul

GO

/* Task 8 */

CREATE NONCLUSTERED COLUMNSTORE INDEX non_clust_discipline

ON studenti(Id_Student, Nume_Student, Prenume_Student)

WITH (DATA_COMPRESSION=COLUMNSTORE) ON userdatafgroup01

CREATE NONCLUSTERED COLUMNSTORE INDEX non_clust_discipline

ON studenti_reusita(Id_Student, Nota)

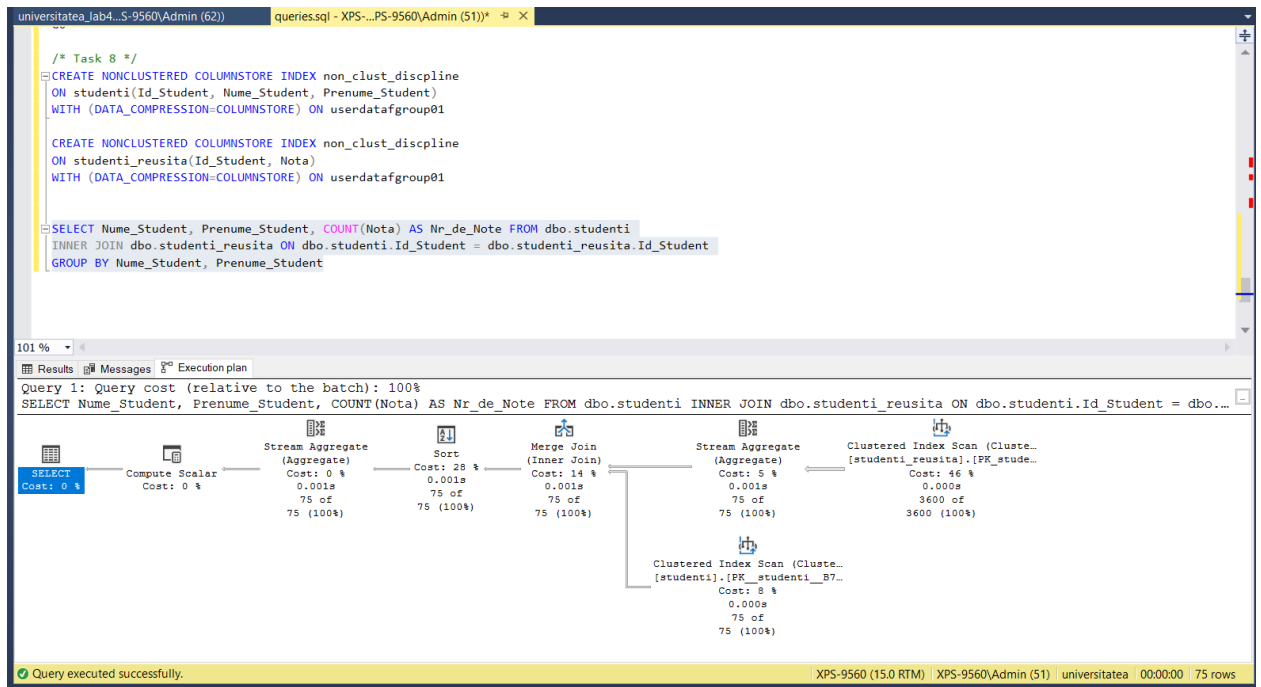
WITH (DATA_COMPRESSION=COLUMNSTORE) ON userdatafgroup01

```

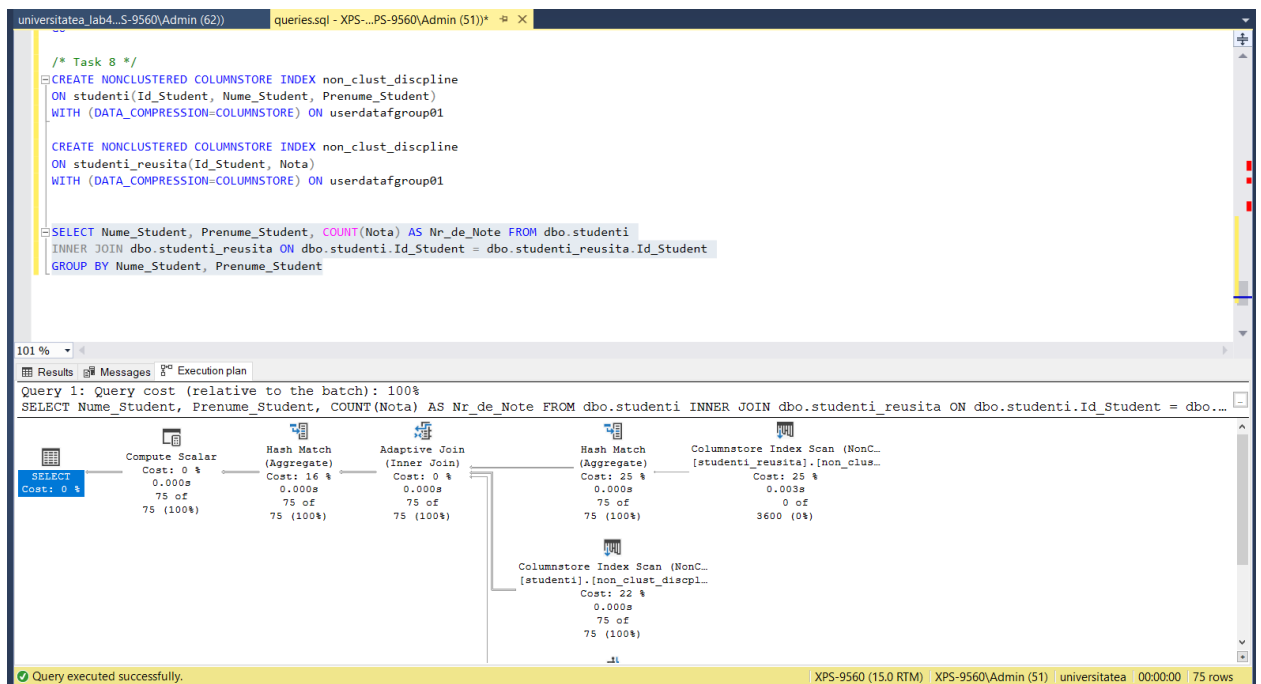
SELECT Nume_Student, Prenume_Student, COUNT(Nota) AS Nr_de_Note FROM dbo.studenti
INNER JOIN dbo.studenti_reusita ON dbo.studenti.Id_Student =
dbo.studenti_reusita.Id_Student
GROUP BY Nume_Student, Prenume_Student

```

Before:



After:



Conclusion:

In this work I had a possibility to work with CREATE and INSERT commands as well as practice different variations of them, such as INSERT INTO SELECT and SELECT INTO. Also, I've got a better understanding of indices and worked more with nested queries.