Universitatea Tehnică a Moldovei

Facultatea Calculatoare Informatică și Microelectronică  
Departamentul Ingineria Software și Automatică

**RAPORT**

**Lucrarea de laborator nr. 10**

**la****Baze de date**

A realizat: st. gr. SI-211 Chihai Adrian

A verificat: Rodica Bulai

**Chișinău – 2023**

**Tema: Creare tabelelor si actualizarea tabelelor**

**Sarcini practice:**

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.

UPDATE profesori

SET adresa\_postala\_profesor = 'mun. Chisinau'

WHERE adresa\_postala\_profesor IS NULL;

1. Sa se modifice schema tabelului grupe, ca sa corespunda urmatoarelor cerinte:
2. Campul Cod\_ Grupa sa accepte numai valorile unice si sa nu accepte valori necunoscute.
3. Sa se tina cont ca cheie primarii, deja, este definitii asupra coloanei Id\_ Grupa.

-- Update rows with null Cod\_Grupa

UPDATE grupe

SET cod\_grupa = 'XX000'

WHERE cod\_grupa IS NULL;

-- Alter column to not allow null values

ALTER TABLE grupe

ALTER COLUMN cod\_grupa SET NOT NULL;

-- Add a unique constraint on Cod\_Grupa

ALTER TABLE grupe

ADD CONSTRAINT unique\_cod\_grupa UNIQUE (cod\_grupa);

1. La tabelul grupe, sa se adauge 2 coloane noi Sef\_grupa și Prof\_Indrumator, ambele de tip INT. Sii se populeze campurile nou-create cu cele mai potrivite candidaturi ill baza criteriilor de maijos:

a) $eful grupei trebuie sa aiba cea mai buna reușitii (medie) din grupa la toate formele de evaluare și la toate disciplinele. Un student nu poate fi șef 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 și inserarea datelor .

ALTER TABLE grupe

ADD COLUMN Sef\_grupa INT,

ADD COLUMN Prof\_Indrumator INT;

DO $$

DECLARE

rec\_grupa RECORD;

BEGIN

FOR rec\_grupa IN (SELECT DISTINCT id\_grupa FROM studenti\_reusita)

LOOP

-- Update Sef\_grupa

UPDATE grupe

SET Sef\_grupa = (

SELECT id\_student

FROM (

SELECT id\_student, ROW\_NUMBER() OVER (ORDER BY AVG(CAST(nota AS FLOAT)) DESC) AS rnk

FROM studenti\_reusita

WHERE id\_grupa = rec\_grupa.id\_grupa

GROUP BY id\_student

) ranked\_students

WHERE rnk = 1

LIMIT 1

)

WHERE grupe.id\_grupa = rec\_grupa.id\_grupa;

-- Update Prof\_Indrumator

UPDATE grupe

SET Prof\_Indrumator = (

SELECT Id\_Profesor

FROM (

SELECT Id\_Profesor, COUNT(DISTINCT id\_disciplina) AS num\_disciplines

FROM studenti\_reusita

WHERE id\_grupa = rec\_grupa.id\_grupa

GROUP BY Id\_Profesor

ORDER BY num\_disciplines DESC, Id\_Profesor ASC

) ranked\_professors

LIMIT 1

)

WHERE grupe.id\_grupa = rec\_grupa.id\_grupa;

END LOOP;

END $$;

1. 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 EXISTS (

SELECT id\_student

FROM studenti\_reusita sr

JOIN grupe g ON g.id\_grupa = sr.id\_grupa

)

AND nota < 10;

5. Sa se creeze un tabel profesori\_new, care include urmatoarele coloane: Id\_Profesor, Nume \_ Profesor, Prenume \_ Profesor, Localitate, Adresa \_ 1, Adresa \_ 2.

1. -- Create the profesori\_new table
2. CREATE TABLE profesori\_new (
3. Id\_Profesor INT PRIMARY KEY,
4. Nume\_Profesor VARCHAR(255),
5. Prenume\_Profesor VARCHAR(255),
6. Localitate VARCHAR(255) DEFAULT 'mun. Chisinau',
7. Adresa\_1 VARCHAR(255),
8. Adresa\_2 VARCHAR(255)
9. );
10. -- Insert data into profesori\_new from the existing profesori table
11. INSERT INTO profesori\_new (Id\_Profesor, Nume\_Profesor, Prenume\_Profesor, Localitate, Adresa\_1, Adresa\_2)
12. SELECT
13. Id\_Profesor,
14. Nume\_Profesor,
15. Prenume\_Profesor,
16. SPLIT\_PART(Adresa\_Postala\_Profesor, ',', 1) AS Localitate,
17. SPLIT\_PART(SPLIT\_PART(Adresa\_Postala\_Profesor, ', ', 2), ', ', 1) AS Adresa\_1,
18. SPLIT\_PART(Adresa\_Postala\_Profesor, ', ', -1) AS Adresa\_2
19. FROM profesori;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: (ld\_Disciplina = 107, Id\_Profesor= 101, Ora ='08:00', Auditoriu = 202); (Id\_Disciplina = 108, Id\_Profesor= 101, Ora ='11:30', Auditoriu = 501); (ld\_Disciplina = 119, Id\_Profesor= 117, Ora ='13:00', Auditoriu = 501);

-- Create the table orarul

CREATE TABLE orarul (

Id\_Disciplina INT NOT NULL,

Id\_Profesor INT,

Id\_Grupa SMALLINT,

Zi CHAR(2),

Ora TIME,

Auditoriu INT,

Bloc CHAR(1) NOT NULL DEFAULT 'B',

PRIMARY KEY (Id\_Grupa, Zi, Ora, Auditoriu)

);

-- Insert values into orarul

INSERT INTO orarul VALUES (107, 101, 1, 'Lu', '08:00', 202, 'B');

INSERT INTO orarul VALUES (108, 101, 1, 'Lu', '11:30', 501, 'B');

INSERT INTO orarul VALUES (109, 117, 1, 'Lu', '13:00', 501, 'B');

1. Sii se scrie expresiile T-SQL necesare pentru a popula tabelul orarul pentru grupa INFl 71 , ziua de luni. Datele necesare pentru inserare trebuie sa fie colectate cu ajutorul instructiunii/instructiunilor SELECT și introduse in tabelul-destinatie, știind 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 for 'Structuri de date si algoritmi' at 08:00

INSERT INTO orarul (Id\_Disciplina, Id\_Profesor, Id\_Grupa, Zi, Ora, Auditoriu, Bloc)

SELECT DISTINCT

d.Id\_Disciplina,

p.Id\_Profesor,

g.Id\_Grupa,

'Lu' as Zi,

'08:00'::time as Ora,

200 as Auditoriu,

'B' as Bloc

FROM

discipline as d, profesori as p, grupe as g, studenti\_reusita as r

WHERE

r.Id\_Disciplina = d.Id\_Disciplina

AND r.Id\_Grupa = g.Id\_Grupa

AND r.Id\_Profesor = p.Id\_Profesor

AND Cod\_Grupa = '101'

AND Disciplina = 'Structuri de date si algoritmi';

-- Insert into orarul for 'Programare aplicativa' at 11:30

INSERT INTO orarul (Id\_Disciplina, Id\_Profesor, Id\_Grupa, Zi, Ora, Auditoriu, Bloc)

SELECT DISTINCT

d.Id\_Disciplina,

p.Id\_Profesor,

g.Id\_Grupa,

'Lu' as Zi,

CAST('11:30' AS time) as Ora,

205 as Auditoriu,

'B' as Bloc

FROM

discipline as d, profesori as p, grupe as g, studenti\_reusita as r

WHERE

r.Id\_Disciplina = d.Id\_Disciplina

AND r.Id\_Grupa = g.Id\_Grupa

AND r.Id\_Profesor = p.Id\_Profesor

AND Cod\_Grupa = '101'

AND Disciplina = 'Programare aplicativa';

-- Insert into orarul for 'Baze de date' at 13:00

INSERT INTO orarul (Id\_Disciplina, Id\_Profesor, Id\_Grupa, Zi, Ora, Auditoriu, Bloc)

SELECT DISTINCT

d.Id\_Disciplina,

p.Id\_Profesor,

g.Id\_Grupa,

'Lu' as Zi,

TIME '13:00' as Ora, -- Use TIME constructor to convert string to time

205 as Auditoriu,

'B' as Bloc

FROM

discipline as d, profesori as p, grupe as g, studenti\_reusita as r

WHERE

r.Id\_Disciplina = d.Id\_Disciplina

AND r.Id\_Grupa = g.Id\_Grupa

AND r.Id\_Profesor = p.Id\_Profesor

AND Cod\_Grupa = '101'

AND Disciplina = 'Baze de date';

**Concluzie:**

În această lucrare de laborator am aplicat cunoștințele despre scrierea și executarea interogărilor în PostgreSQL, precum și despre scrierea instrucțiunilor procedurale în limbajul specific PL/pgSQL. Avantajul folosirii acestui limbaj constă în capacitatea de a declara variabile și de a stoca date, precum și în posibilitatea de a controla fluxul de execuție al unui proces. Acest lucru permite realizarea unor operațiuni complexe și personalizate în cadrul bazelor de date PostgreSQL.