Universitatea Tehnică a Moldovei

Facultatea - Calculatoare Informatică și Microelectronică

**Raport**

**la Baze de Date**

Lucrarea de laborator Nr.3

Tema: Tipurile de constrangeri și relațiile

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

A verificat: Rodica Bulai

Chișinău 2023

**1.Scopul Lucrarii**

Constrangerile si construirea coloanelor cu cu diferite constrangeri intrun table

**2.Obiective**

Această lucrare de laborator are ca scop cercetarea și înțelegerea detaliată a constrângerilor în PostgreSQL. Obiectivele includ identificarea tipurilor de constrângeri de integritate disponibile în PostgreSQL, precum chei primare, chei străine și constrângeri de unicitate. De asemenea, se va explora modul în care aceste constrângeri pot afecta performanța operațiunilor în baza de date, precum și comportamentul lor în situații limită. De asemenea, se vor examina mecanismele de notificare și raportare a erorilor generate de constrângeri. Această lucrare va oferi o înțelegere detaliată a modului în care constrângerile contribuie la menținerea integrității datelor într-o bază de date PostgreSQL.

3. Desfășurarea lucrării de laborator

În PostgreSQL (psql), există mai multe tipuri de constrângeri pe care le puteți aplica la nivelul bazelor de date. Iată câteva dintre cele mai comune tipuri de constrângeri disponibile:

1. **Cheie Primară (Primary Key):** Această constrângere asigură că o coloană sau un grup de coloane într-o tabelă este unică și nu poate conține valori nule. Aceasta servește drept identificator unic pentru fiecare înregistrare în tabelă.
2. **Cheie Străină (Foreign Key):** O cheie străină este o constrângere care definește o relație între două tabele. Aceasta indică faptul că valorile dintr-o coloană (sau grup de coloane) dintr-o tabelă trebuie să corespundă cu valorile dintr-o altă tabelă.
3. **Unicitate (Unique):** Această constrângere asigură că valorile dintr-o coloană sau un grup de coloane sunt unice în întreaga tabelă. Deși poate permite valori nule, nu permite duplicarea altor valori.
4. **Verificare (Check):** Această constrângere permite definirea unei expresii logice care trebuie să fie adevărată pentru fiecare înregistrare din tabelă. Dacă expresia este falsă, operațiunile de inserare sau actualizare vor fi refuzate.
5. **Valoare Implicită (Default):** Aceasta permite specificarea unei valori implicite pentru o coloană. Dacă nu se furnizează o valoare la momentul inserării, valoarea implicită va fi utilizată.
6. **Cheie de Control (Check Constraint):** Aceasta permite definirea unei expresii care trebuie să fie adevărată pentru fiecare înregistrare în tabelă. Dacă expresia este falsă, operațiunile de inserare sau actualizare vor fi refuzate.

Aceste constrângeri sunt instrumente puternice în gestionarea integrității datelor și în stabilirea relațiilor între tabele într-o bază de date PostgreSQL.

Am creat doua tabele cu multe tipuri de constrangeri  
CREATE TABLE users (

user\_id SERIAL PRIMARY KEY,

username VARCHAR(50) UNIQUE NOT NULL,

email VARCHAR(100) UNIQUE ,

age INT CHECK (age >= 18) NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE orders (

order\_id SERIAL PRIMARY KEY,

user\_id INT NOT NULL,

product\_name VARCHAR(100) NOT NULL,

quantity INT NOT NULL,

order\_date DATE,

description VARCHAR(250) default 'description of item',

FOREIGN KEY (user\_id) REFERENCES users(user\_id)

);

În lucrarea numarul 2 am creat tablele pentru niversitate și multe din ele au relatia de 1 la multi

codul pentru creare

CREATE TABLE discipline (

id\_disciplina SERIAL PRIMARY KEY,

disciplina VARCHAR(100),

nr\_ore\_plan\_disciplina INT

);

CREATE TABLE grupe (

id\_grupa SERIAL PRIMARY KEY,

cod\_grupa VARCHAR(10),

specialitatea VARCHAR(100),

facultatea VARCHAR(100)

);

CREATE TABLE studenti (

id\_student SERIAL PRIMARY KEY,

nume\_student VARCHAR(50),

prenume\_student VARCHAR(50),

data\_nasterii\_student DATE,

adresa\_postala\_student VARCHAR(200)

);

CREATE TABLE profesori (

id\_profesor SERIAL PRIMARY KEY,

nume\_profesor VARCHAR(50),

prenume\_profesor VARCHAR(50),

adresa\_postala\_profesor VARCHAR(200)

);

CREATE TABLE studenti\_reusita (

id\_student INT,

id\_disciplina INT,

id\_profesor INT,

id\_grupa INT,

tip\_evaluare VARCHAR(50),

nota INT,

data\_evaluare DATE,

FOREIGN KEY (id\_student) REFERENCES studenti(id\_student),

FOREIGN KEY (id\_disciplina) REFERENCES discipline(id\_disciplina),

FOREIGN KEY (id\_profesor) REFERENCES profesori(id\_profesor),

FOREIGN KEY (id\_grupa) REFERENCES grupe(id\_grupa)

);

Acesta table este cu relatia de 1 la 1 cu tabelul studenti



Concluzie:

Această lucrare de laborator a oferit o perspectivă detaliată asupra conceptului de constrângeri în PostgreSQL. Prin explorarea diferitelor tipuri de constrângeri, am înțeles importanța acestora în menținerea integrității datelor și în definirea relațiilor dintre tabele. Am observat că utilizarea constrângerilor este esențială în gestionarea eficientă a informațiilor într-o bază de date.

De asemenea, am experimentat aplicarea și gestionarea constrângerilor într-un mediu practic, ceea ce ne-a permis să obținem o înțelegere mai profundă a modului în care acestea influențează comportamentul operațiunilor de manipulare a datelor.

Pe lângă aceasta, am constatat că constrângerile contribuie semnificativ la asigurarea coerenței și corectitudinii datelor stocate în baza de date, oferind un nivel suplimentar de securitate și fiabilitate.

În ansamblu, această lucrare de laborator a fost o experiență educativă și aplicată, care ne-a permis să dobândim competențe esențiale în utilizarea și gestionarea constrângerilor în PostgreSQL. Aceste cunoștințe sunt cruciale în dezvoltarea și administrarea eficientă a bazelor de date în cadrul proiectelor viitoare.