**Ministerul Educaţiei şi Cercetării**

**al Republicii Moldova**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

RAPORT

la lucrarea de laborator nr. 9

La disciplina: Baze de Date

Tema: “Subinterogari“

A efectuat:     Buza Cătălin TI-214

A verificat:     Cebotar Gabriela

Chișinău 2023

**Subinterogări în Structurile SQL:**

O subinterogare reprezintă o interogare (SELECT, UPDATE, DELETE) imbricată în interiorul unei alte interogări. Aceasta poate fi folosită în diferite contexte, și anume:

**1. Subinterogări corelate și necorelate:**

**Corelate**:

O subinterogare corelată este dependentă de rezultatul interogării externe și poate fi evaluată pentru fiecare rând returnat de interogarea principală.

SELECT Specialitate, Cod\_Grupa FROM grupe

WHERE Id\_Grupa = ( SELECT Id\_Grupa

FROM grupe WHERE Cod\_Grupa = 'CIB171'

);

**Necorelate**:

O subinterogare necorelată este independentă de interogarea exterioară și este evaluată o singură dată, înainte de a fi aplicată interogării principale.

SELECT Nume\_Student, Prenume\_Student

FROM studenti WHERE Id\_Student IN (

SELECT Id\_Student FROM studenti

WHERE Adresa\_Postala\_Student LIKE '%mun. Chisinau, str. Tighina, 23/3%');

**2. Subinterogări cu operatori de comparație:**

Subinterogările pot fi utilizate cu operatori de comparație (>, <, =, etc.) pentru a compara rezultatele acestora cu o valoare specificată.

SELECT Nume\_Profesor, Prenume\_Profesor FROM profesori

WHERE Id\_Profesor < ( SELECT MAX(Id\_Profesor)

FROM profesori);

**3. Subinterogări cu operatorii IN, ALL, ANY, [NOT] EXISTS:**

IN: Verifică dacă o valoare se află într-o listă de valori returnată de subinterogare.

ALL și ANY: Se folosesc pentru a compara o valoare cu toate sau oricare dintre valorile returnate de subinterogare.

[NOT] EXISTS: Verifică dacă subinterogarea returnează sau nu vreo înregistrare.

/\* Operatori IN, ALL, ANY, [NOT] EXISTS \*/

SELECT Disciplina FROM discipline

WHERE Nr\_ore\_plan\_disciplina > ALL ( SELECT Nr\_ore\_plan\_disciplina

FROM discipline WHERE Disciplina LIKE '%CIB171%'

);

**4. Subinterogări în clauza FROM:**

Subinterogările pot fi plasate în clauza FROM pentru a obține seturi de date pe care apoi să se facă alte operațiuni.

SELECT DISTINCT Nume\_Student, Prenume\_Student FROM (

SELECT studenti.Nume\_Student, studenti.Prenume\_Student

FROM studenti

JOIN studenti\_reusita ON studenti\_reusita.Id\_Student = studenti.Id\_Student

WHERE studenti\_reusita.Id\_Profesor=100

) AS StudentiGrupe;

**Concluzie**

Subinterogările reprezintă o tehnică puternică în limbajul SQL, oferind flexibilitate și complexitate în formularea interogărilor pentru baze de date. În cadrul acestei lucrări de laborator, am explorat subiecte precum subinterogările corelate și necorelate, utilizarea operatorilor de comparație, precum și operatorii IN, ALL, ANY și [NOT] EXISTS.

Subinterogările corelate sunt utile atunci când condițiile interogării depind de rezultatele exterioare, în timp ce subinterogările necorelate pot fi evaluate independent de contextul exterioar. Operatorii de comparație și cei logici oferă posibilitatea de a restrânge rezultatele în funcție de condiții specifice, iar operatorii IN, ALL, ANY și [NOT] EXISTS permit compararea și filtrarea datelor în moduri variate.

De asemenea, am examinat cum subinterogările pot fi integrate în clauza FROM pentru a obține seturi de date complexe, deschizând astfel posibilitatea pentru analize și rapoarte mai avansate.