**MINISTERUL EDUCAŢIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

**Departamentul Ingineria Software și Automatică**

**Programul de studii: Tehnologia informației**

RAPORT

# LUCRARE DE LABORATOR NR. 6

# la Baze de Date

**Tema: Interogări unirelație simple în SQL**

A efectuat:

st. gr. TI-211 Popa Cătălin

A verificat: Olga Grosu

UTM, Chișinău 2023

Materialele corespunde cu paragrafele 4.1, 4.2 din indrumarul SQL Server 2019. Interogările se vor face la baza de date Universitatea, creată anterior în SSMS, la Laboratorul 3.

În lab.6 se vor face interogări care realizează operațiile proiecție și selecție, utilizând:

* Clauzele SELECT, FROM, WHERE, ORDER BY;
* Operatorii IN, BETWEEN, LIKE, IS[NOT] NULL, TOP [WITH TIES];
* Operatorii de comparative (=, >, <, etc)
* Operatorii logici AND, OR, NOT,
* Aleasuri de atribuit.

**Clauzele:**

SELECT: Se utilizează pentru a specifica numele coloanelor ce vor afișa în rezultatul interogării.

FROM: Se utilizează pentru a specifica tabelele din care vor fi extrase coloanele specificate în clauza SELECT.

<listă coloane> cuprinde câmpurile care trebuie să apară în tabelul sau tabelele cu rezultatul interogării.

<listă tabele> reprezintă lista tabelelor de unde datele trebuie să fie selectate.

\* cere includerea tuturor înregistrărilor care îndeplinesc cindițiile impuse.

--acest select va returna toate datele cu toate coloanele si randurile din tabela discipline

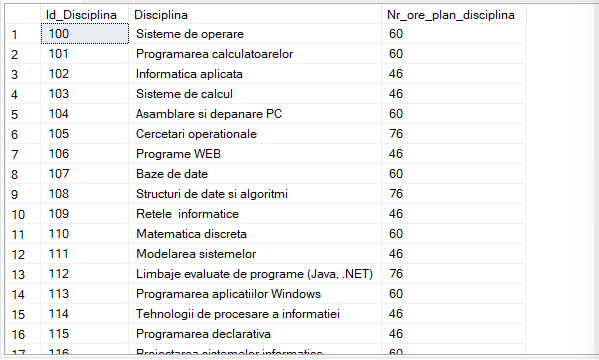
select \* from universitatea.univer.discipline;

Figura 1 – Select general.

WHERE: Clauză opțională utilizată în filtrarea înregistrărilor returnate de clauza FROM. WHERE este precedat de o expresie numită condiție de căutare.

ORDER BY: sortează datele setate la ieșire în ordinea specificată.

--acest select va returna datele din coloanele Id\_Student si Adresa\_Postala\_Student din tabela studenti

--prin where se specifica ca doar Id\_Student >= ca 5 vor satisface conditia

--la fel, prin order by are loc aranjarea descrescatoare in dependenta de id\_student

select a.Id\_Student, a.Adresa\_Postala\_Student from universitatea.univer.studenti a

where a.Id\_Student >= 5

order by a.Id\_Student desc;

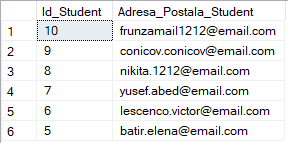


Figura 2 – Interogare folosind order by.

**Operatorii:**

IN: operatorul IN este folosit pentru a compara o anumită valoare cu o listă de valori specifice și se verifică dacă rezultatul evaluării expresiei se potrivește cu una dintre valorile din lista furnizată.

BETWEEN: permite specificarea unui domeniu mărginit de două valori între care trebuie să se afle valoarea testată.

LIKE: se folosește pentru a găsi valoarea care conține caracterul specificat.

--aceasta interogare va prelua datele din tabela stundeti\_reusita

--am folosit in pentru a specifica o lista de id\_discipline care satisface conditia

--la fel Nota a fost specificata sa fie intre 7 si 10, folosind operatorul BETWEEN

--in final am folosit like pentru a specifica care tip de evaluare va satisface conditia

select b.Id\_Student, b.Id\_Disciplina,b.Tip\_Evaluare, b.Nota

from universitatea.univer.studenti\_reusita b

where b.Id\_Disciplina in(110, 116, 120, 112,117)

and b.Nota between 7 and 10

and b.Tip\_Evaluare like 'oral' ;

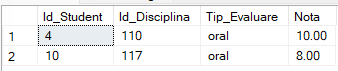


Figura 3 – Interogare folosind operatorii in, between si like.

IS NULL: se folosește pentru a compara valoarea curentă cu o valoare NULL.

--aceasta interogare va afisa randurile de date din tabela reusita studenti in care nota este nula

select b.\*

from universitatea.univer.studenti\_reusita b

where b.Nota is null

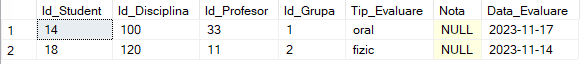


Figura 4 – Operatorul is null.

IS NOT NULL: este folosit pentru a verifica lipsa unei valori într-un câmp.

--aceasta interogare va afisa randurile de date din tabela reusita studenti in care nota nu este nula

select b.\*

from universitatea.univer.studenti\_reusita b

where b.Nota is not null;

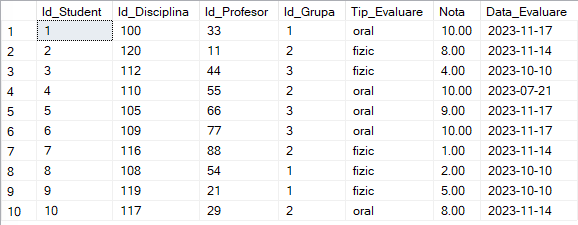


Figura 5 – Operatorul is not null.

TOP: permite extragerea unui număr sau un procent de linii returnate de o interogare.

--va afisa primele 5 randuri cu date din tabela discipline

select TOP(5) a.\* from universitatea.univer.discipline a

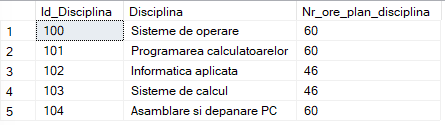


Figura 6 – TOP.

TOP WITH TIES: poate fi definită numai în instrucțiunea SELECT, și numai dacă este menționată clauza ORDER BY.

--va returna primele 2 randuri si plus restul randurilor care au numarul de ore la fel ca la primele 2

select TOP(2) WITH TIES a.\* from universitatea.univer.discipline a

order by a.Nr\_ore\_plan\_disciplina asc;

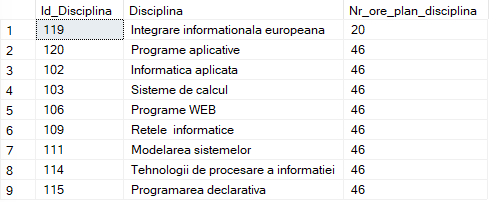


Figura 7 – TOP WITH TIES.

**Operatori de comparatie:**

= verifică daca valorile sunt egale.

--aceasta interogare va afisa datele din tabela profesori unde id la profesor este egala cu 55

select a.\* from universitatea.univer.profesori a

where a.Id\_Profesor = 55;



Figura 8 – Interogare folosind operatorul = .

< verifică dacă valoarea este mai mică.

> verifică dacă valoarea este mai mare.

--aceasta interogare va afisa datele despre stundetii care au nota mai mare ca 7 si mai mica ca 10

select b.\* from universitatea.univer.studenti\_reusita b

where b.Nota > 7 and b.Nota < 10;

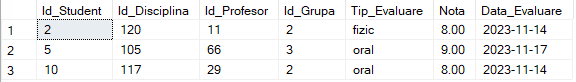


Figura 9 – Operatorii < si >.

**Operatori logici:**

AND:

--aceast select va afisa datele care au numarul de ore a disciplinei = cu 60 si discipline are in nume caracterele Sisteme

select a.\* from universitatea.univer.discipline a

where a.Nr\_ore\_plan\_disciplina = 60 and a.Disciplina like '%Sisteme%' ;

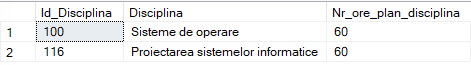


Figura 10 – Operatorul logic AND.

OR:

--acest select va returna datele care au numarul de ore = cu 60 sau 76

select a.\* from universitatea.univer.discipline a

where a.Nr\_ore\_plan\_disciplina = 60 or a.Nr\_ore\_plan\_disciplina = 76 ;

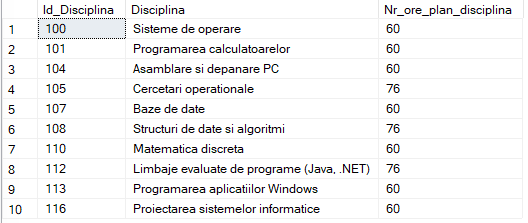


Figura 11 – Operatorul login OR.

NOT:

--acest select va elimina din afisare numele stundetilor precum Catalin, Dan, Nikita

select a.\* from universitatea.univer.studenti a

where a.Prenume\_Student not in ('Catalin', 'Dan', 'Nikita');

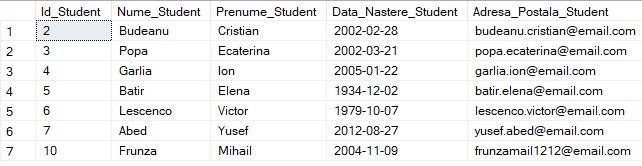


Figura 12 – Operatorul logic NOT.

**Concluzie**

Laboratorul 6 s-a concentrat pe interogări SQL pentru operațiile de proiecție și selecție în bazele de date. Utilizând clauzele SELECT, FROM, WHERE, și ORDER BY, am învățat să extragem și să organizăm datele eficient. Operatorii IN, BETWEEN, LIKE, IS[NOT] NULL, și TOP [WITH TIES] au adăugat versatilitate la selecția datelor, în timp ce operatorii de comparație (=, >, <) și logici (AND, OR, NOT) au permis condiționarea complexă a rezultatelor. Aleasurile de atribuit au oferit flexibilitate în redenumirea și structurarea coloanelor. În concluzie, acest laborator a consolidat competențele esențiale pentru manipularea datelor în bazele de date relaționale folosind SQL, dezvoltându-se o înțelegere detaliată a proceselor fundamentale.