Universitatea Tehnica din Republica Moldova

Facultatea Calculatoare, Informatica si Microelectronica

Departament Inginerie Software si Automatica

Specialitatea Tehnologia Informației

Raport

**Lucare de laborator Nr.3**

Curs: Baze de Date

Tema:Instructiunea select transact -SQL

Grupa: TI-205

A elaborat: Triboi Andreia

A verificat: Asistent universitar Scrob Sergiu

**Chișinău U.T.M. 2022**

**Lucare de laborator Nr.3**

1. Scopul lucrării: Studierea elementelor de bază ale limbajului Transact-SQL, a structurii instrucțiunii SELECT și a tipurilor de funcții utilizate în interogări. Familiarizarea cu aspectele de utilizare a interfeței Editorului de Interogări pentru crearea și gestiunea interogărilor SQL.

2. Mersul lucrării:

Să se scrie interogările în limbajul SQL și să se afișeze răspunsurile la aceste interogări:

16) Furnizați numele și prenumele studenților, care au studiat discipline cu un volum de lecții mai mic de 60 de ore, precum și profesorii (identificatorii) respectivi, care le-au predat.

SELECT DISTINCT Nume\_Student, Prenume\_Student, Disciplina, Nr\_ore\_plan\_disciplina, Id\_Profesor

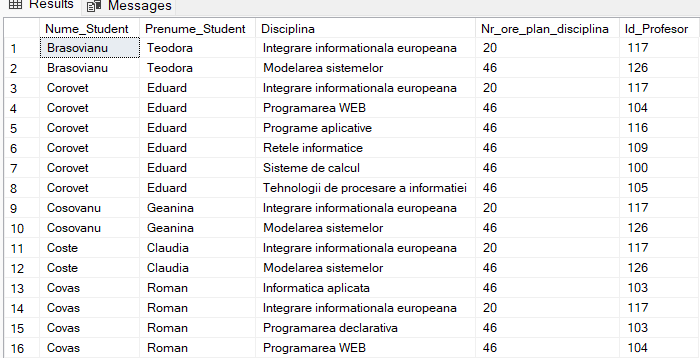
FROM studenti AS s INNER JOIN studenti\_reusita AS sr

ON s.Id\_Student = sr.Id\_Student

INNER JOIN discipline AS d

ON sr.Id\_Disciplina = d.Id\_Disciplina

WHERE Nr\_ore\_plan\_disciplina < 60



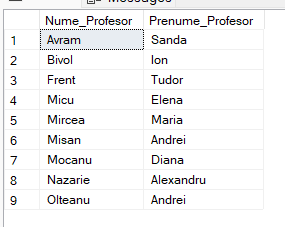
17) Afișați numele și prenumele profesorilor, care au predat cel puțin o disciplină studentului cu identificator 100.

SELECT DISTINCT Nume\_Profesor, Prenume\_Profesor

FROM profesori AS p INNER JOIN studenti\_reusita AS sr

ON p.Id\_Profesor = sr.Id\_Profesor

WHERE Id\_Student = 100



18) Afișați numele și prenumele profesorilor, care au predat doar discipline cu o încărcare orară mai mică de 60 de ore.

SELECT DISTINCT Nume\_Profesor, Prenume\_Profesor

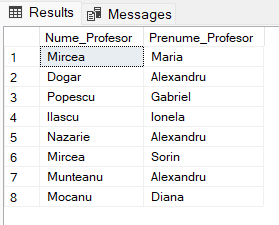
FROM profesori AS p INNER JOIN studenti\_reusita AS sr

ON p.Id\_Profesor = sr.Id\_Profesor

INNER JOIN discipline AS d

ON sr.Id\_Disciplina = d.Id\_Disciplina

WHERE Nr\_ore\_plan\_disciplina < 60



19) Găsiți numele și prenumele profesorilor, care au predat discipline, în care studentul “Cosovanu” a fost respins (nota < 5) la cel puțin o probă.

SELECT DISTINCT Nume\_Profesor, Prenume\_Profesor

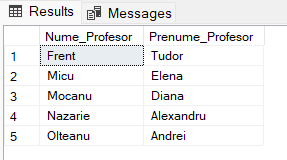
FROM profesori AS p INNER JOIN studenti\_reusita AS sr

ON p.Id\_Profesor = sr.Id\_Profesor

INNER JOIN studenti AS s

ON sr.Id\_Student = s.Id\_Student

WHERE Nume\_Student = 'Cosovanu' AND Nota < 5



20) Afișați numărul de studenți care au susținut testul (Testul 2) la disciplina Baze de date în 2018.

SELECT COUNT(Nume\_Student) AS Numar\_studenti

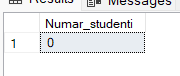
FROM studenti AS s INNER JOIN studenti\_reusita AS sr

ON s.Id\_Student = sr.Id\_Student

INNER JOIN discipline AS d

ON sr.Id\_Disciplina = d.Id\_Disciplina

WHERE Tip\_Evaluare = 'Testul 2' AND Disciplina = 'Baze de date' AND YEAR(Data\_Evaluare) = 2018 AND Nota > 4



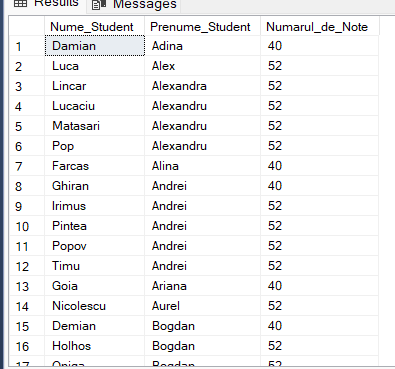
21) Câte note are fiecare student? Furnizați numele și prenumele lor.

SELECT DISTINCT Nume\_Student, Prenume\_Student, COUNT(Nota) AS Numarul\_de\_Note

FROM studenti AS s INNER JOIN studenti\_reusita sr

ON s.Id\_Student = sr.Id\_Student

GROUP BY Nume\_Student, Prenume\_Student



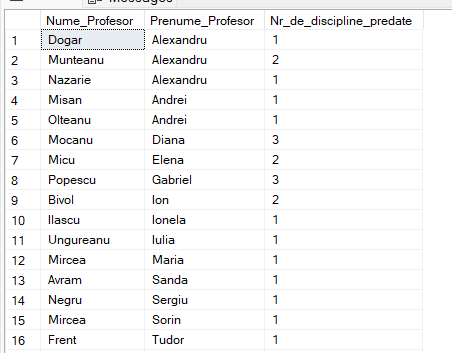
22) Să se obțină numărul de discipline predate de fiecare profesor (Nume\_Profesor, Prenume\_Profesor).

SELECT Nume\_Profesor, Prenume\_Profesor, COUNT(DISTINCT(Id\_Disciplina)) AS Nr\_de\_discipline\_predate

FROM profesori AS p INNER JOIN studenti\_reusita AS sr

ON p.Id\_Profesor = sr.Id\_Profesor

GROUP BY Nume\_Profesor, Prenume\_Profesor



23) Să se obțină lista disciplinelor (Disciplina) susținute de studenți cu nota medie de promovare la examen mai mare de 7, în ordine descrescătoare după denumirea disciplinei.

SELECT DISTINCT d.Disciplina

FROM discipline AS d INNER JOIN studenti\_reusita AS sr

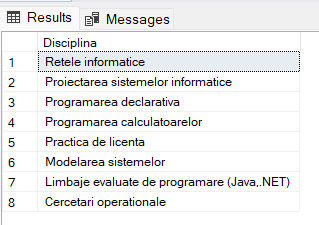
ON d.Id\_Disciplina = sr.Id\_Disciplina

WHERE Tip\_Evaluare = 'Examen'

GROUP BY Disciplina

HAVING AVG(1.0 \* Nota) > 7

ORDER BY Disciplina DESC



24) Să se afișeze lista disciplinelor (Disciplina) predate de cel puțin doi profesori.

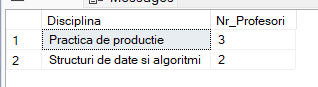
SELECT DISTINCT d.Disciplina, COUNT( DISTINCT (Id\_Profesor)) AS Nr\_Profesori

FROM discipline AS d INNER JOIN studenti\_reusita AS sr

ON d.Id\_Disciplina = sr.Id\_Disciplina

GROUP BY Disciplina

HAVING COUNT( DISTINCT (Id\_Profesor)) >= 2



25) În ce grupe de studii (Cod\_Grupa) figurează mai mult de 24 de studenți ?

SELECT Cod\_Grupa FROM (

SELECT Cod\_Grupa, COUNT(DISTINCT Id\_Student) AS c

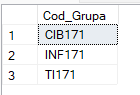
FROM studenti\_reusita

JOIN grupe g ON g.Id\_Grupa = studenti\_reusita.Id\_Grupa

GROUP BY Cod\_Grupa, Id\_Student) tab

GROUP BY Cod\_Grupa

HAVING COUNT(c) > 24;



26) Găsiți numele, prenumele și adresa studenților și ale profesorilor care locuiesc pe strada 31 August.

SELECT Nume\_Student, Prenume\_Student, Adresa\_Postala\_Student

FROM studenti

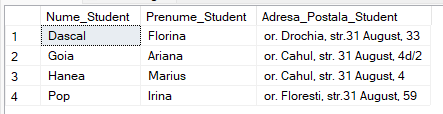
WHERE Adresa\_Postala\_Student like '%31 August%'

UNION

SELECT Nume\_Profesor, Prenume\_Profesor, Adresa\_Postala\_Profesor

FROM profesori

WHERE Adresa\_Postala\_Profesor like '%31 August%'



27) Afișați studenții (identificatorii) care au suținut (evaluare examen) la toate disciplinile predate de prof.Ion.

SELECT DISTINCT sr.Id\_Student, Nume\_Student, Prenume\_Student

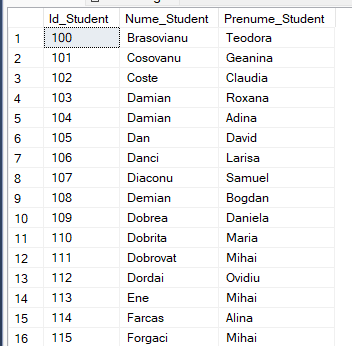
FROM studenti\_reusita AS sr INNER JOIN profesori AS p

ON sr.Id\_Profesor = p.Id\_Profesor

INNER JOIN studenti AS s

ON sr.Id\_Student = s.Id\_Student

WHERE Tip\_Evaluare ='Examen' AND Nota > 4 AND Prenume\_Profesor='Ion'



29) Determinați numele și prenumele studenților, care au susținut examene cu o notă mai mică decât oricare dintre notele studenților cu identificatorul 100.

SELECT DISTINCT s.Id\_Student, Nume\_Student, Prenume\_Student

FROM studenti AS s INNER JOIN studenti\_reusita AS sr

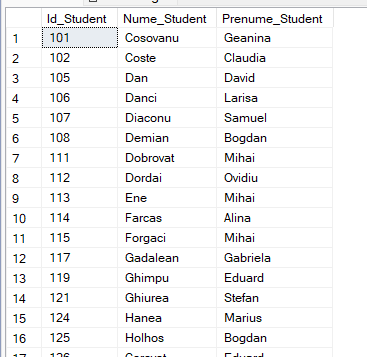
ON s.Id\_Student = sr.Id\_Student

WHERE Tip\_Evaluare ='Examen' AND Nota < ALL

(SELECT Nota

FROM studenti\_reusita

WHERE Id\_Student = 100 AND Tip\_Evaluare='Examen')



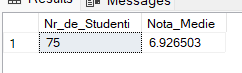
30) Câți studenți au studiat, deja, disciplina Baze de date în 2018 și 2019 și care este media lor la reușita curentă ?

SELECT COUNT (DISTINCT(Id\_Student)) AS Nr\_de\_Studenti, AVG(Nota\*1.0) AS Nota\_Medie

FROM studenti\_reusita AS sr INNER JOIN discipline AS d

ON sr.Id\_Disciplina = d.Id\_Disciplina

WHERE YEAR(Data\_Evaluare) = 2018 OR YEAR(Data\_Evaluare) =2019 AND Disciplina ='Baze de date'



31) Furnizați numele și prenumele studenților, care au avut mai mult de două dezaprobări într-un an la o disciplină.

SELECT sr.Id\_Student, s.Nume\_Student, s.Prenume\_Student, Id\_Disciplina, COUNT(Id\_Disciplina) AS Numar\_De\_Dezaprobari

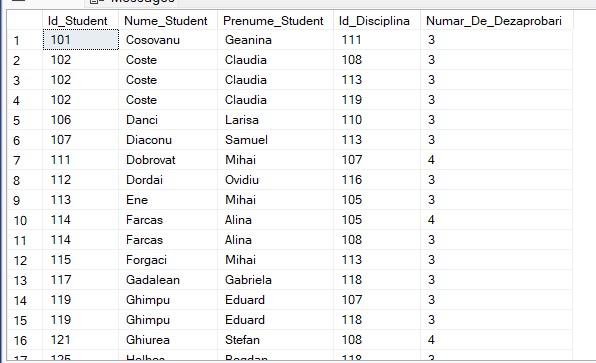
FROM studenti\_reusita AS sr INNER JOIN studenti AS s

ON sr.Id\_Student = s.Id\_Student

WHERE Nota < 5

GROUP BY Id\_Disciplina, sr.Id\_Student, s.Nume\_Student, s.Prenume\_Student

HAVING COUNT(Id\_Disciplina) > 2



32) Furnizați numele, prenumele și media notelor pe grupe pentru studenți.

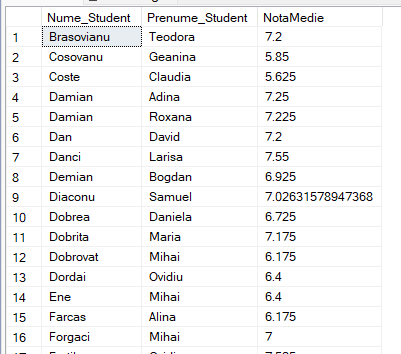
SELECT Nume\_Student, Prenume\_Student, AVG(CAST(Nota AS FLOAT)) as NotaMedie

FROM studenti AS s INNER JOIN studenti\_reusita as sr

ON s.Id\_Student = sr.Id\_Student

GROUP BY Nume\_Student, Prenume\_Student, Id\_Grupa

ORDER BY Id\_Grupa



33) Găsiți numele și prenumele studenților, care nu au luat notă de promovare la reușită curentă la nicio disciplină.

SELECT Nume\_Student, Prenume\_Student

FROM studenti

EXCEPT

SELECT DISTINCT Nume\_Student, Prenume\_Student

FROM studenti\_reusita INNER JOIN studenti s

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

WHERE Nota >= 5 AND Tip\_Evaluare = 'Reusita curenta'



34) Găsiți numele și prenumele studenților, care nu au luat note de promovare (inclusiv note intermediare) la nicio disciplină.

SELECT Nume\_Student, Prenume\_Student

FROM studenti

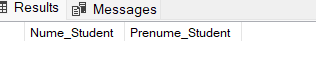
EXCEPT

SELECT DISTINCT Nume\_Student, Prenume\_Student

FROM studenti\_reusita INNER JOIN studenti s

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

WHERE Nota >= 5



35) Găsiți denumirile disciplinelor și media notelor pe disciplină. Afișați numai disciplinile cu medii mai mare de 7.0.

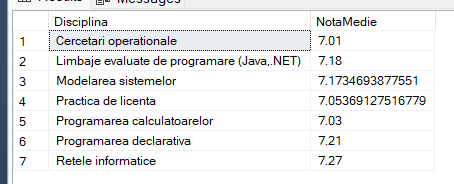
SELECT Disciplina, AVG(CAST (Nota AS FLOAT)) as NotaMedie

FROM discipline AS d INNER JOIN studenti\_reusita AS sr

ON d.Id\_Disciplina = sr.Id\_Disciplina

GROUP BY Disciplina

HAVING AVG(CAST (Nota AS FLOAT)) > 7.0



36) Găsiți numele, prenumele și adresele studenților și ale profesorilor care locuiesc într-o localitate. Să se afișeze denumirea localității și numărul de locuitori incluși în cele două categorii. Datele se afișează în ordinea crescătoare a numărului membrilor din categoria menționată anterior.

SELECT localitate, adresa, nume, prenume, COUNT(numar) OVER ( partition by localitate) AS locuitori FROM(

SELECT SUBSTRING (adresa, 1, charindex(',', adresa)) as localitate, nume, prenume, adresa, COUNT(nume) AS numar

FROM (SELECT DISTINCT Nume\_Student AS nume, Prenume\_Student AS prenume, Adresa\_Postala\_Student AS adresa

FROM studenti

UNION

SELECT DISTINCT Nume\_Profesor AS nume, Prenume\_Profesor AS prenume, Adresa\_Postala\_Profesor AS adresa

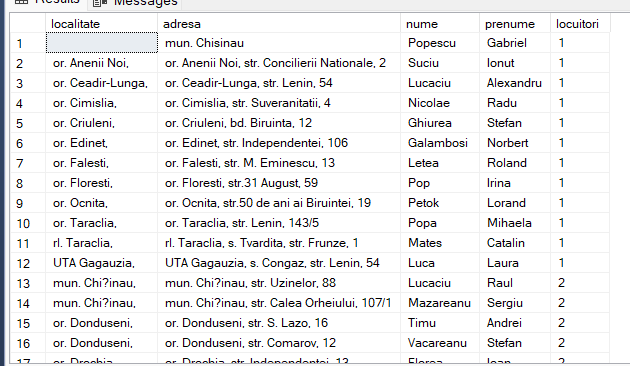
FROM profesori) t

GROUP BY adresa, prenume, nume, substring(adresa, 1, charindex(',', adresa))

) g

GROUP BY localitate, adresa, nume, prenume, numar

ORDER BY locuitori, localitate



37) Găsiți disciplina susținută cu nota medie (la examen) cea mai înaltă.

SELECT DISTINCT Disciplina, AVG(CAST (Nota AS FLOAT))

FROM discipline AS d INNER JOIN studenti\_reusita AS sr

ON d.Id\_Disciplina = sr.Id\_Disciplina

WHERE Tip\_Evaluare = 'Examen'

GROUP BY Disciplina

HAVING AVG(CAST (Nota AS FLOAT)) =(

SELECT MAX(NotaMedie) FROM(

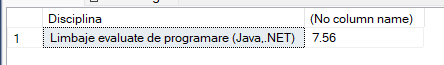
SELECT DISTINCT Disciplina, Tip\_Evaluare, AVG(CAST (Nota AS FLOAT)) as NotaMedie

FROM discipline AS d INNER JOIN studenti\_reusita AS sr

ON d.Id\_Disciplina = sr.Id\_Disciplina

WHERE Tip\_Evaluare = 'Examen'

GROUP BY Disciplina, Tip\_Evaluare) u)



38) Furnizați denumirile disciplinelor cu o medie mai mică decât media notelor la disciplină Baze de date.

SELECT DISTINCT Disciplina, AVG(CAST (Nota AS FLOAT)) AS NotaMedie

FROM discipline AS d INNER JOIN studenti\_reusita AS sr

ON d.Id\_Disciplina = sr.Id\_Disciplina

GROUP BY Disciplina

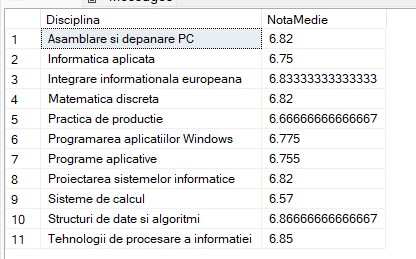
HAVING AVG(CAST (Nota AS FLOAT)) < (

SELECT AVG(CAST (Nota AS FLOAT)) AS NotaMedie

FROM discipline AS d INNER JOIN studenti\_reusita AS sr

ON d.Id\_Disciplina = sr.Id\_Disciplina

WHERE Disciplina = 'Baze de date')



39) Găsiți denumirile disciplinelor la care nu au susținut examenul, în medie, peste 5% de studenți.

SELECT Disciplina, count(Id\_Student) as restanta, total from(

SELECT Id\_Student, Id\_Disciplina

FROM studenti\_reusita

WHERE Tip\_Evaluare = 'Examen'

AND Nota < 5

GROUP BY Id\_Student, Id\_Disciplina

) AS t, (

SELECT Id\_Disciplina, COUNT(DISTINCT Id\_Student) as total

FROM studenti\_reusita

GROUP BY Id\_Disciplina

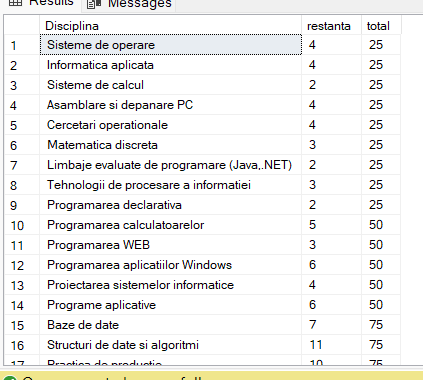
) AS g, discipline d

WHERE t.Id\_Disciplina = d.Id\_Disciplina

AND g.Id\_Disciplina = d.Id\_Disciplina

GROUP BY t.Id\_Disciplina, Disciplina, total

HAVING (cast(count(Id\_Student) as float) / cast(total as float) \* 100 ) > 5



**Concluzie:**

In concluzie ,la realizarea acestei lucrari de laborator am studiat elementele de bază ale limbajului Transact-SQL, a structurii instrucțiunii SELECT și a tipurilor de funcții utilizate în interogări. M-am familiarizat cu aspectele de utilizare a interfeței Editorului de Interogări pentru crearea și gestiunea interogărilor SQL.