**Ministerul Educației și Cercetării al Republicii Moldova**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

**Departamentul Informatică şi Ingineria Sistemelor**

**RAPORT**

Lucrare de laborator nr.2

la cursul „Baze de date”

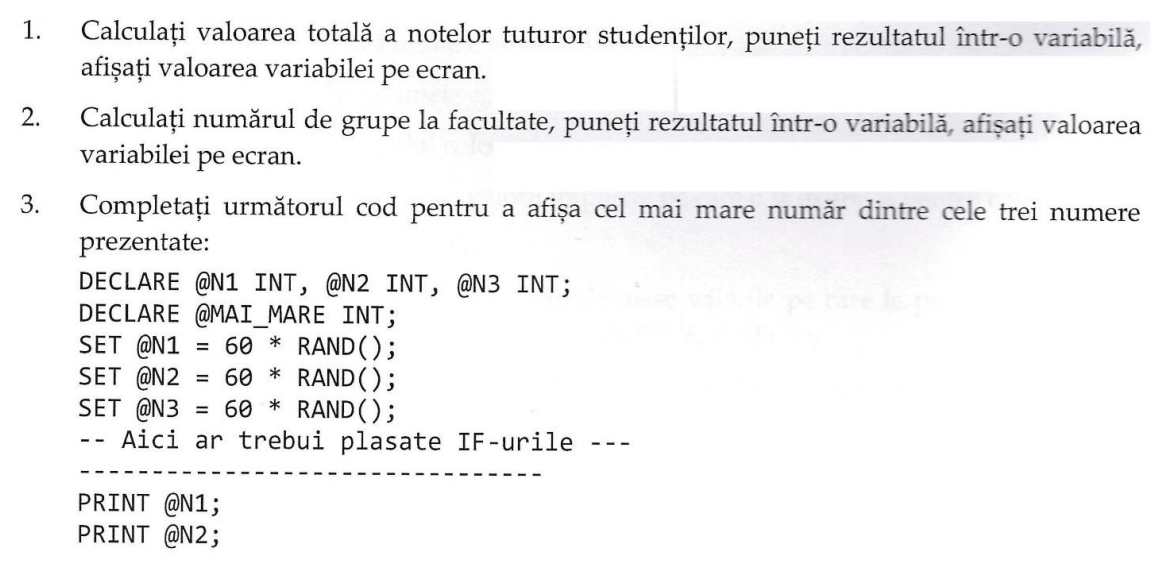
**Tema 1:** „ Instructiuni Procedurale”

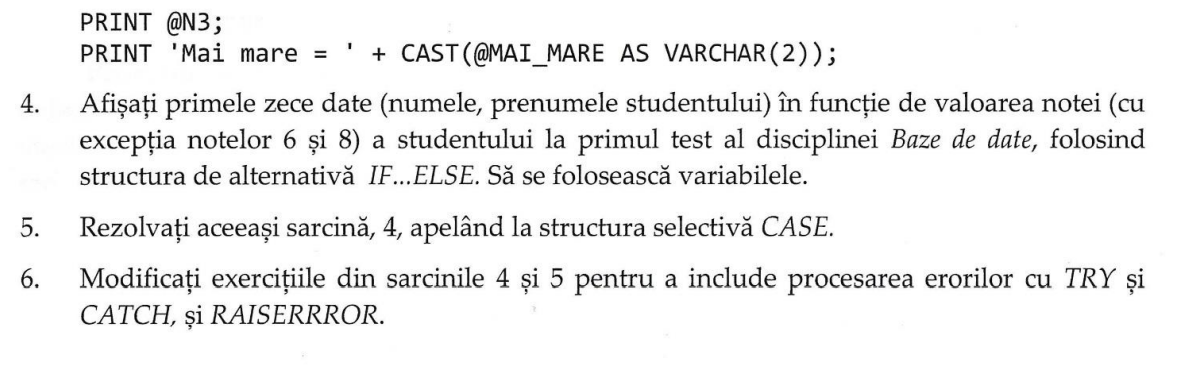
A efectuat : **TI-214 Reguș Ruslan**

A verificat:  **Gabriela Cebotar**

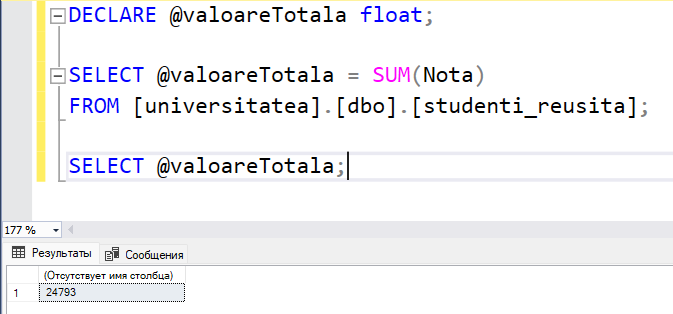
**Chișinău 2024**

**Exercitii practice:**

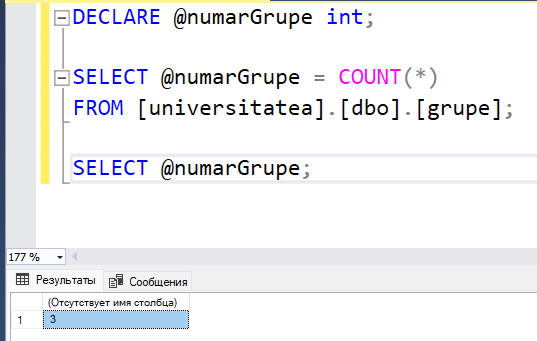




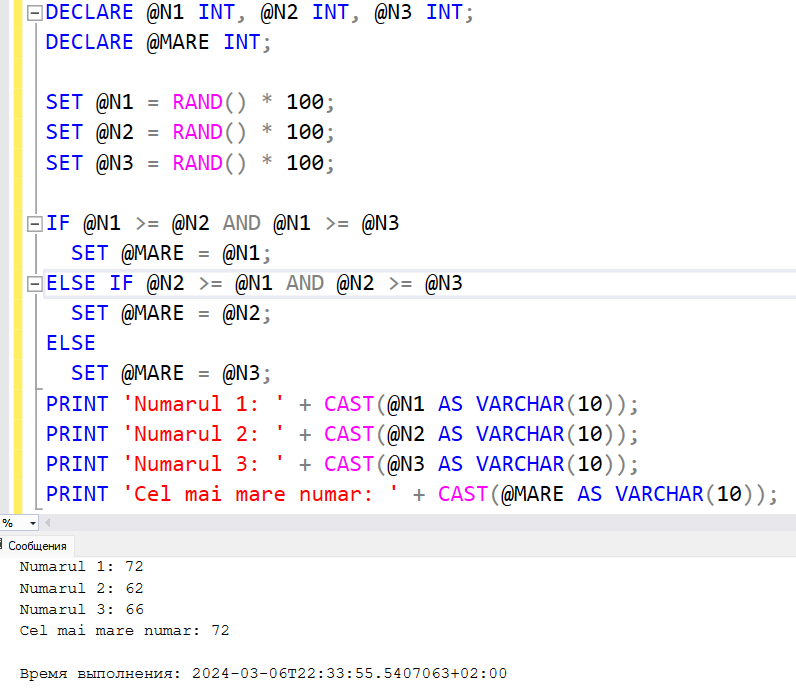
1



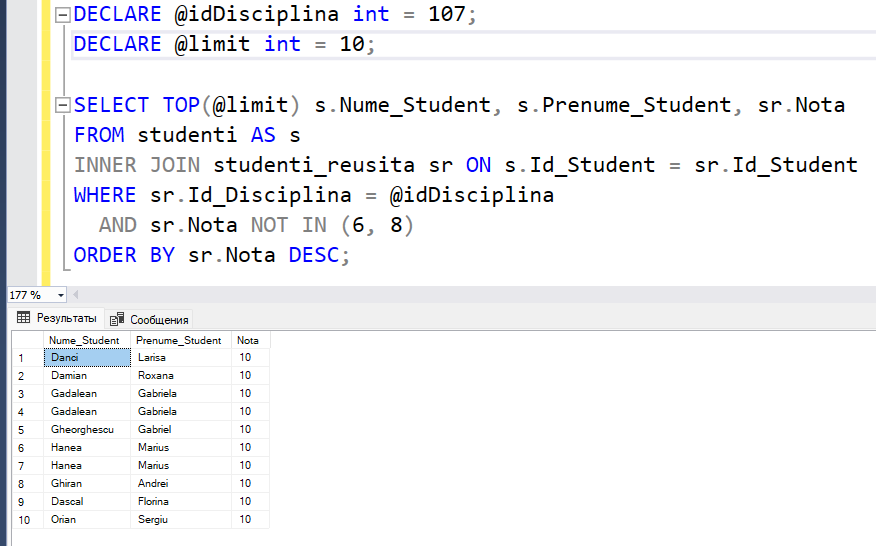
2



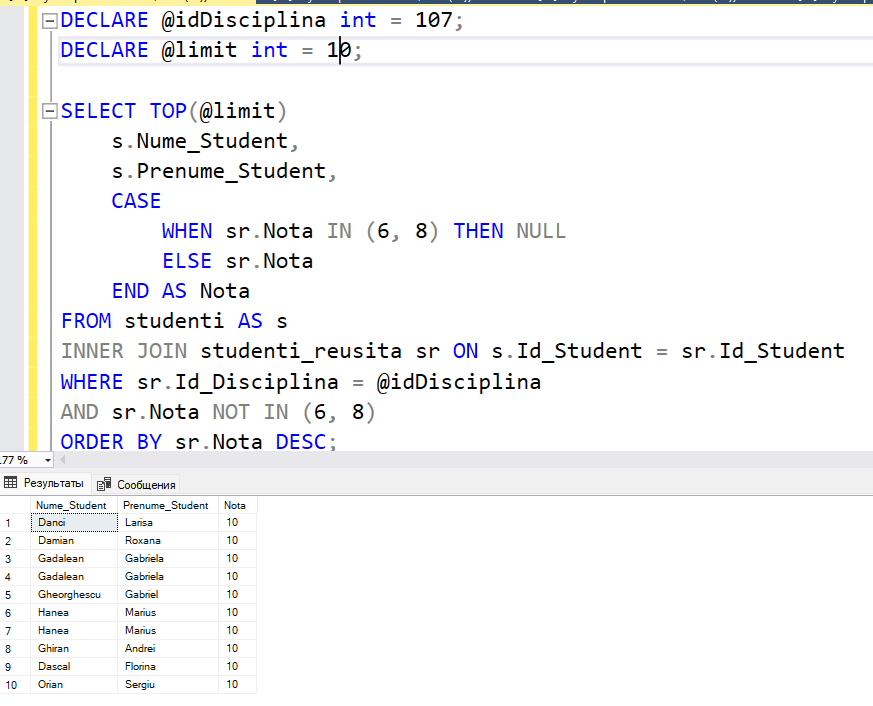
3



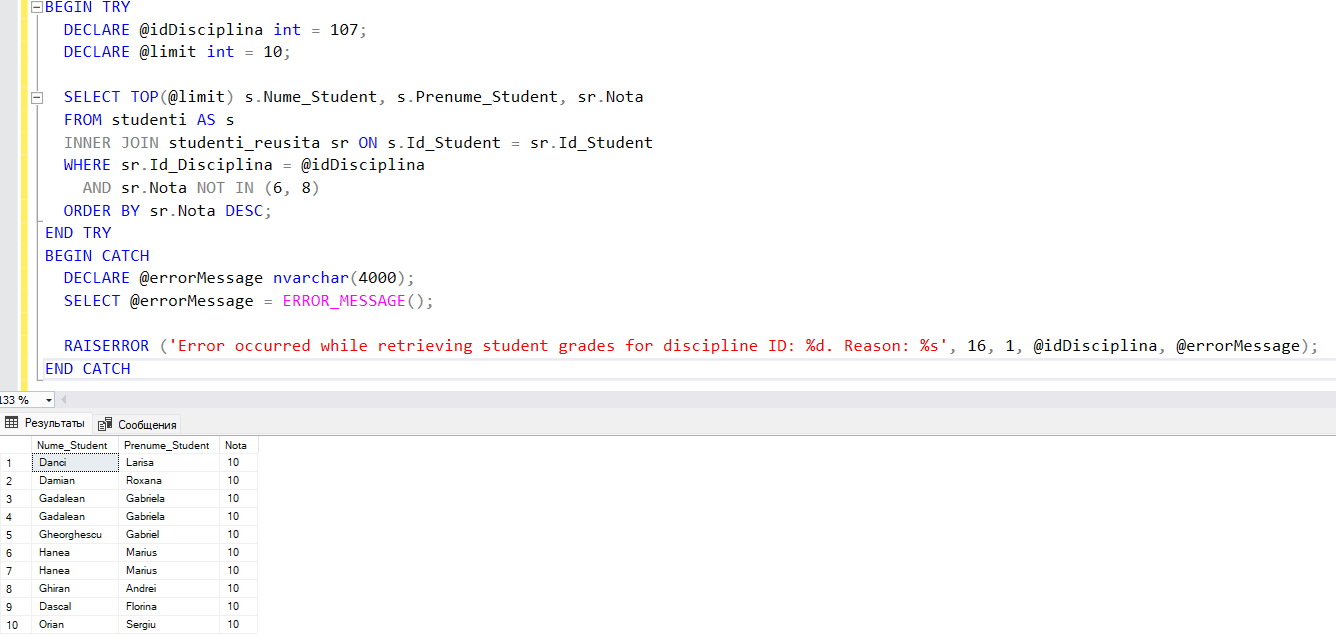
4



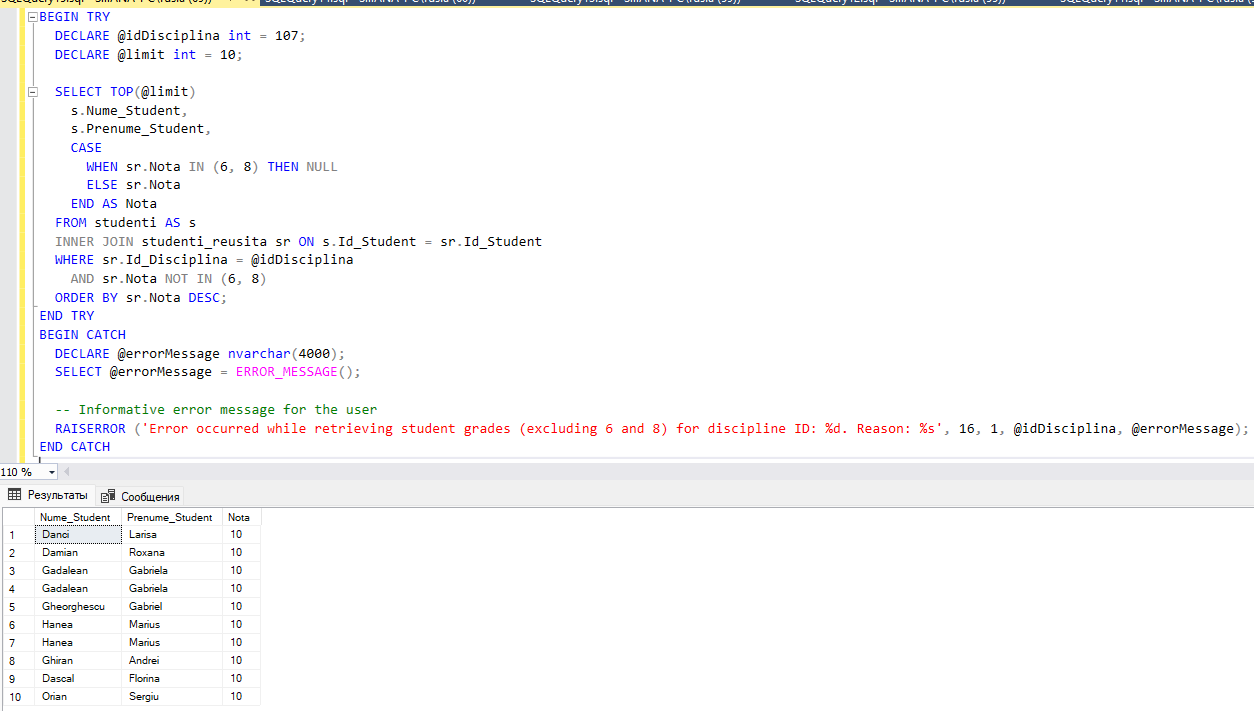
5



6 pentru 4



6 pentru 5



Codul :

1

DECLARE @valoareTotala float;

SELECT @valoareTotala = SUM(Nota)

FROM [universitatea].[dbo].[studenti\_reusita];

SELECT @valoareTotala;

2

DECLARE @numarGrupe int;

SELECT @numarGrupe = COUNT(\*)

FROM [universitatea].[dbo].[grupe];

SELECT @numarGrupe;

3

DECLARE @N1 INT, @N2 INT, @N3 INT;

DECLARE @MARE INT;

SET @N1 = RAND() \* 100;

SET @N2 = RAND() \* 100;

SET @N3 = RAND() \* 100;

IF @N1 >= @N2 AND @N1 >= @N3

SET @MARE = @N1;

ELSE IF @N2 >= @N1 AND @N2 >= @N3

SET @MARE = @N2;

ELSE

SET @MARE = @N3;

PRINT 'Numarul 1: ' + CAST(@N1 AS VARCHAR(10));

PRINT 'Numarul 2: ' + CAST(@N2 AS VARCHAR(10));

PRINT 'Numarul 3: ' + CAST(@N3 AS VARCHAR(10));

PRINT 'Cel mai mare numar: ' + CAST(@MARE AS VARCHAR(10));

4

DECLARE @idDisciplina int = 107;

DECLARE @limit int = 10;

SELECT TOP(@limit) s.Nume\_Student, s.Prenume\_Student, sr.Nota

FROM studenti AS s

INNER JOIN studenti\_reusita sr ON s.Id\_Student = sr.Id\_Student

WHERE sr.Id\_Disciplina = @idDisciplina

AND sr.Nota NOT IN (6, 8)

ORDER BY sr.Nota DESC;

5

DECLARE @idDisciplina int = 107;

DECLARE @limit int = 10;

SELECT TOP(@limit)

s.Nume\_Student,

s.Prenume\_Student,

CASE

WHEN sr.Nota IN (6, 8) THEN NULL

ELSE sr.Nota

END AS Nota

FROM studenti AS s

INNER JOIN studenti\_reusita sr ON s.Id\_Student = sr.Id\_Student

WHERE sr.Id\_Disciplina = @idDisciplina

AND sr.Nota NOT IN (6, 8)

ORDER BY sr.Nota DESC;

6 pentru 4

BEGIN TRY

DECLARE @idDisciplina int = 107;

DECLARE @limit int = 10;

SELECT TOP(@limit) s.Nume\_Student, s.Prenume\_Student, sr.Nota

FROM studenti AS s

INNER JOIN studenti\_reusita sr ON s.Id\_Student = sr.Id\_Student

WHERE sr.Id\_Disciplina = @idDisciplina

AND sr.Nota NOT IN (6, 8)

ORDER BY sr.Nota DESC;

END TRY

BEGIN CATCH

DECLARE @errorMessage nvarchar(4000);

SELECT @errorMessage = ERROR\_MESSAGE();

RAISERROR ('Error occurred while retrieving student grades for discipline ID: %d. Reason: %s', 16, 1, @idDisciplina, @errorMessage);

END CATCH

6 pentru 5

BEGIN TRY

DECLARE @idDisciplina int = 107;

DECLARE @limit int = 10;

SELECT TOP(@limit)

s.Nume\_Student,

s.Prenume\_Student,

CASE

WHEN sr.Nota IN (6, 8) THEN NULL

ELSE sr.Nota

END AS Nota

FROM studenti AS s

INNER JOIN studenti\_reusita sr ON s.Id\_Student = sr.Id\_Student

WHERE sr.Id\_Disciplina = @idDisciplina

AND sr.Nota NOT IN (6, 8)

ORDER BY sr.Nota DESC;

END TRY

BEGIN CATCH

DECLARE @errorMessage nvarchar(4000);

SELECT @errorMessage = ERROR\_MESSAGE();

-- Informative error message for the user

RAISERROR ('Error occurred while retrieving student grades (excluding 6 and 8) for discipline ID: %d. Reason: %s', 16, 1, @idDisciplina, @errorMessage);

END CATCH

**Concluzie**

În urma realizării acestor sarcini, ne-am concentrat pe scrierea unor interogări SQL pentru a extrage și prelucra date din baza de date universitate. Am acoperit o varietate de sarcini, inclusiv:

Calculul valorii totale a notelor: Am implementat o interogare care calculează suma notelor din tabela studenti\_reusita și stochează rezultatul într-o variabilă.

Numărarea grupelor: Am scris o interogare care determină numărul total de grupuri din tabela grupe și stochează rezultatul într-o variabilă.

Găsirea celui mai mare număr: Am creat cod care generează trei numere aleatorii, le compară și identifică cel mai mare număr folosind instrucțiuni IF imbricate.

Recuperarea datelor studenților cu note specifice: Am dezvoltat o interogare care utilizează o îmbinare INNER JOIN între tabelele studenti și studenti\_reusita pentru a recupera numele și prenumele primilor 10 studenți cu note (excluzând 6 și 8) pentru disciplina "Baze de date" (ID 107). Ordonarea se face descrescător în funcție de note.

Recuperarea datelor studenților cu note specifice (cu gestionarea valorilor 6 și 8): Similar sarcinii 4, am implementat o interogare care recuperează datele studenților, dar utilizează o expresie CASE pentru a înlocui notele 6 și 8 cu NULL

Implementarea gestionării erorilor (pentru sarcinile 4 și 5): Am adăugat blocuri TRY...CATCH în interogările din sarcinile 4 și 5 pentru a gestiona erorile potențiale. Blocul CATCH capturează erorile, preia mesajul de eroare și generează un mesaj personalizat pentru utilizator, oferind detalii despre eroare și disciplina afectată.

Acest raport demonstrează competența în scrierea interogărilor SQL pentru diferite scopuri, inclusiv calcule simple, preluarea datelor din tabele multiple și gestionarea erorilor.