**Лабораторна робота № 6**

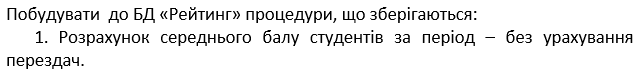
**Процедури, що зберігаються. Тригери**

Мета роботи: дослідження властивостей та можливостей процедур, що зберігаються та тригерів.

Хід роботи:

A close up of text

Description automatically generated



Лістинг:

CREATE PROCEDURE CalculateAverageGradeWithoutRetakes

AS

BEGIN

SELECT Kod\_student, AVG(Reiting) AS AvgGrade

FROM Reiting

WHERE Prisutn = 1

GROUP BY Kod\_student;

END

EXEC CalculateAverageGradeWithoutRetakes;

A screenshot of a computer

Description automatically generated

Рис. 1 Результат виконання роботи



Лістинг:

CREATE PROCEDURE CalculateAverageGradeWithRetakes

AS

BEGIN

SELECT Kod\_student, AVG(Reiting) AS AvgGrade

FROM Reiting

GROUP BY Kod\_student;

END

EXEC CalculateAverageGradeWithRetakes

A white sheet with black numbers

Description automatically generated

Рис. 2 Результат виконання роботи



Лістинг:

CREATE PROCEDURE GetStudentsWithGrades4And5

AS

BEGIN

SELECT DISTINCT s.\*

FROM dbo\_student s

INNER JOIN Reiting r ON s.Kod\_stud = r.Kod\_student

WHERE r.Reiting >= 74

END

EXEC GetStudentsWithGrades4And5

A white background with black text

Description automatically generated

Рис. 3 Результат виконання роботи



Лістинг:

CREATE PROCEDURE CalculateTotalGradeValues

@studentCode INT

AS

BEGIN

DECLARE @totalPoints INT

DECLARE @nationalGrade NVARCHAR(10)

DECLARE @ectsGrade NVARCHAR(10)

SELECT @totalPoints = SUM(Reiting)

FROM Reiting

WHERE Kod\_student = @studentCode;

IF @totalPoints BETWEEN 60 AND 63

BEGIN

SET @nationalGrade = '3'

SET @ectsGrade = 'E'

END

ELSE IF @totalPoints BETWEEN 64 AND 73

BEGIN

SET @nationalGrade = '3'

SET @ectsGrade = 'D'

END

ELSE IF @totalPoints BETWEEN 74 AND 81

BEGIN

SET @nationalGrade = '4'

SET @ectsGrade = 'C'

END

ELSE IF @totalPoints BETWEEN 82 AND 89

BEGIN

SET @nationalGrade = '4'

SET @ectsGrade = 'B'

END

ELSE IF @totalPoints BETWEEN 90 AND 100

BEGIN

SET @nationalGrade = '5'

SET @ectsGrade = 'A'

END

ELSE IF @totalPoints < 60

BEGIN

SET @nationalGrade = '2'

SET @ectsGrade = 'F'

END

ELSE

BEGIN

SET @nationalGrade = NULL

SET @ectsGrade = NULL

END

SELECT @totalPoints AS 'TotalPoints', @nationalGrade AS 'NationalGrade', @ectsGrade AS 'ECTSGrade'

END

EXEC CalculateTotalGradeValues @studentCode = 9

A screenshot of a computer

Description automatically generated

Рис. 4 Результат виконання роботи



Лістинг:

CREATE TRIGGER AddNewGroupOnInsert

ON dbo\_student

AFTER INSERT

AS

BEGIN

INSERT INTO dbo\_groups (Kod\_group, K\_navch\_plan)

SELECT DISTINCT i.Kod\_group, 17

FROM inserted i

LEFT JOIN dbo\_groups g ON i.Kod\_group = g.Kod\_group

WHERE g.Kod\_group IS NULL;

END

INSERT INTO dbo\_student (Sname, Name, FName, N\_ingroup, Kod\_group)

VALUES ('Додо', 'Валерій', 'Петрович', 1, N'c-58');

A screenshot of a computer

Description automatically generated

Рис. 5 Результат виконання роботи



Лістинг:

CREATE TRIGGER DeleteEmptyGroupOnUpdate

ON dbo\_student

AFTER DELETE

AS

BEGIN

DELETE FROM dbo\_groups

WHERE Kod\_group IN (

SELECT g.Kod\_group

FROM dbo\_groups g

LEFT JOIN dbo\_student s ON g.Kod\_group = s.Kod\_group

GROUP BY g.Kod\_group

HAVING COUNT(s.Kod\_stud) = 0

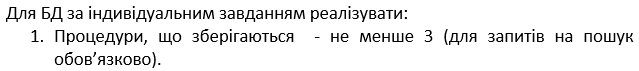
);

END

A white grid with black text

Description automatically generated

Рис. 6 Результат виконання роботи



Лістинг:

CREATE PROCEDURE FindEmployeeByLastName (@lastName VARCHAR(50))

AS

BEGIN

SELECT \* FROM Employees WHERE Last\_Name = @lastName;

END;

EXEC FindEmployeeByLastName @lastName = N'Іванов'; 

Рис. 7 Результат виконання роботи

Лістинг:

CREATE PROCEDURE FindEmployeesByDepartment (@departmentName VARCHAR(100))

AS

BEGIN

SELECT \* FROM Employees e JOIN Departments d ON e.Department\_ID = d.Department\_ID WHERE d.Department\_Name = @departmentName;

END;

EXEC FindEmployeesByDepartment @departmentName = N'Відділ розробки';

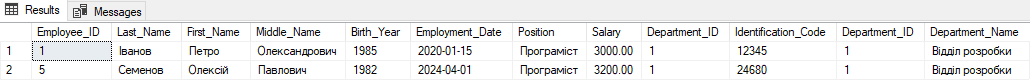


Рис. 8 Результат виконання роботи

Лістинг:

CREATE PROCEDURE CalculateAverageSalaryByDepartment (@departmentName VARCHAR(100))

AS

BEGIN

SELECT AVG(Salary) AS AverageSalary FROM Employees e JOIN Departments d ON e.Department\_ID = d.Department\_ID WHERE d.Department\_Name = @departmentName;

END;

EXEC CalculateAverageSalaryByDepartment @departmentName = N'Відділ маркетингу';

A screenshot of a computer screen

Description automatically generated

Рис. 9 Результат виконання роботи

Лістинг:

CREATE TRIGGER AfterInsertEmployee

ON Employees

AFTER INSERT

AS

BEGIN

PRINT 'Новий працівник був доданий!';

END;

INSERT INTO Employees (Employee\_ID, Last\_Name, First\_Name, Middle\_Name, Birth\_Year, Employment\_Date, Position, Salary, Department\_ID, Identification\_Code)

VALUES (8, 'Сидоренко', 'Олександр', 'Іванович', 1990, '2024-05-06', 'Аналітик', 3000.00, 1, '12345');

A screenshot of a computer

Description automatically generated

Рис. 10 Результат виконання роботи

Лістинг:

CREATE TRIGGER UpdatePositionOnDepartmentChange

ON Employees

AFTER UPDATE

AS

BEGIN

IF EXISTS (

SELECT \* FROM inserted i

JOIN deleted d ON i.Employee\_ID = d.Employee\_ID

WHERE i.Department\_ID = 4 AND d.Department\_ID <> 4

)

BEGIN

UPDATE Employees

SET Position = 'HR менеджер'

WHERE Employee\_ID IN (SELECT Employee\_ID FROM inserted);

END

END;

UPDATE Employees

SET Department\_ID = 4

WHERE Employee\_ID = 1;



Рис. 11 Результат виконання роботи

Лістинг:

CREATE TABLE EmployeeChanges (

Change\_ID INT PRIMARY KEY IDENTITY,

Employee\_ID INT,

Change\_Date DATETIME,

Changed\_Field VARCHAR(100),

Old\_Value VARCHAR(100),

New\_Value VARCHAR(100)

);

CREATE TRIGGER LogEmployeeChanges

ON Employees

AFTER UPDATE

AS

BEGIN

INSERT INTO EmployeeChanges (Employee\_ID, Change\_Date, Changed\_Field, Old\_Value, New\_Value)

SELECT i.Employee\_ID, GETDATE(), c.COLUMN\_NAME, d.Position, i.Position

FROM inserted i

INNER JOIN deleted d ON i.Employee\_ID = d.Employee\_ID

CROSS JOIN INFORMATION\_SCHEMA.COLUMNS c

WHERE c.TABLE\_NAME = 'Employees' AND c.COLUMN\_NAME NOT IN ('Employee\_ID', 'Employment\_Date')

AND i.Position <> d.Position;

END;

UPDATE Employees

SET Position = 'Системний адміністратор'

WHERE Employee\_ID = 1;

select \* from EmployeeChanges

A screenshot of a computer

Description automatically generated

Рис. 12 Результат виконання роботи

Висновок: дослідив властивості та можливості процедур, що зберігаються та тригерів.