**Лабораторна робота №6**

**Тема: Процедури, що зберігаються. Тригери**

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

Хід роботи

**Завдання на лабораторну роботу:**

Побудувати до БД «Рейтинг» процедури, що зберігаються:

1. Розрахунок середнього балу студентів за період – без урахування перездач.
2. Розрахунок середнього балу студентів за період – з урахування перездач.
3. Визначення студентів, що навчаються на 4 та 5.
4. Процедура виводить суму балів та її значення в національній системі та ECTS.
5. Тригер на вставку даних в таблицю студент – якщо код групи новий в таблицю додається група.
6. Тригер на модифікацію даних з таблиці студенти якщо більше немає студентів в групі група знищується.

Для БД за індивідуальним завданням реалізувати:

1. Процедури, що зберігаються - не менше 3 (для запитів на пошук обов’язково).
2. Тригери для таблиць – не менше 3.

**Виконання:**

1. /\*Розрахунок середнього балу студентів за період – без урахування перездач\*/

CREATE PROCEDURE SeredBalZaPeriod1 AS

BEGIN

select AVG(r.Reiting), s.SName, s.Name, s.FName from Reiting r

INNER JOIN Rozklad\_pids rp ON (r.K\_zapis=rp.K\_zapis)

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

WHERE Date BETWEEN '2017-01-01' AND '2017-10-01'

GROUP BY Sname, Name, Fname

HAVING COUNT(\*)<2;

END;

EXEC SeredBalZaPeriod1

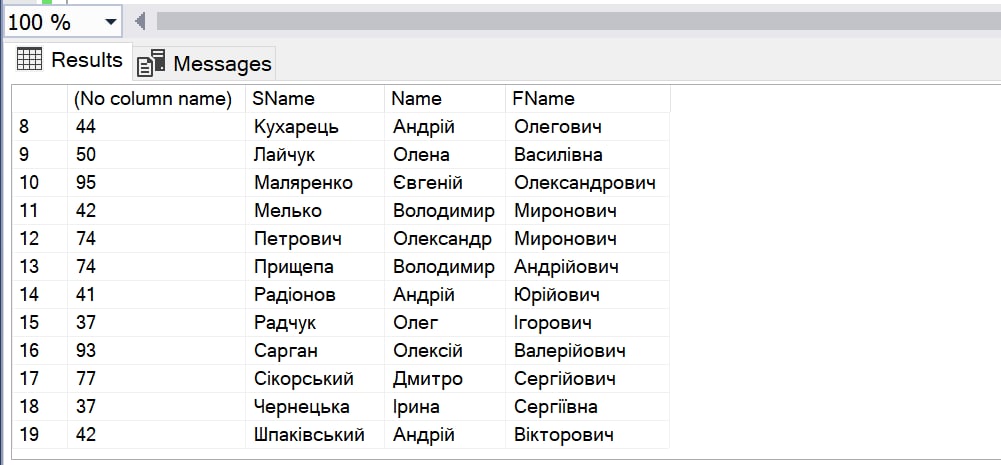


Рис. 6.1 Результат виконання процедури.

/\*Розрахунок середнього балу студентів за період – з урахування перездач.\*/

CREATE PROCEDURE SeredBalZaPeriod2 AS

BEGIN

select AVG(r.Reiting), s.SName, s.Name, s.FName from Reiting r

INNER JOIN Rozklad\_pids rp ON (r.K\_zapis=rp.K\_zapis)

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

WHERE Date BETWEEN '2017-01-01' AND '2017-10-01'

GROUP BY Kod\_student, Sname, Name, Fname;

END;

EXEC SeredBalZaPeriod2

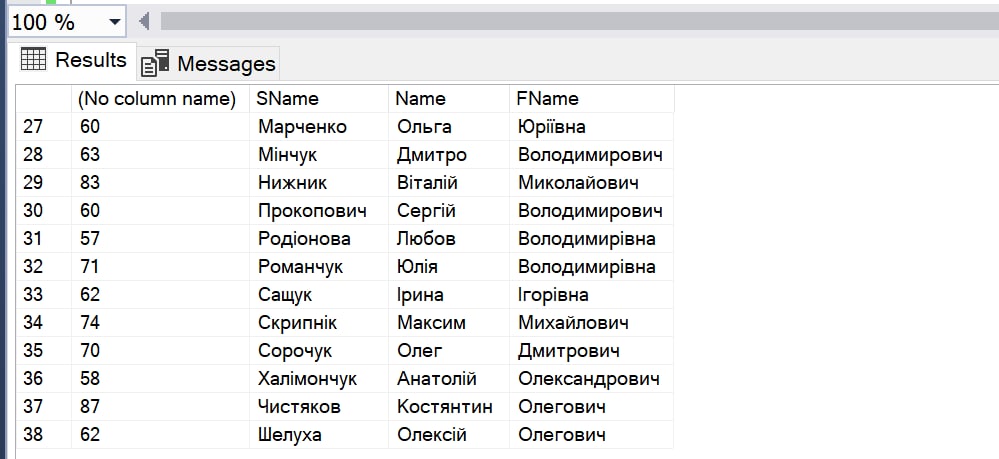


Рис. 6.2 Результат виконання процедури.

/\*Визначення студентів, що навчаються на 4 та 5.\*/

CREATE PROCEDURE OcinkiBilshe4i5 AS

BEGIN

select s.Sname,s.Name,s.Fname,r.Reiting from dbo\_student s

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

WHERE Reiting>=74;

END;

EXEC OcinkiBilshe4i5

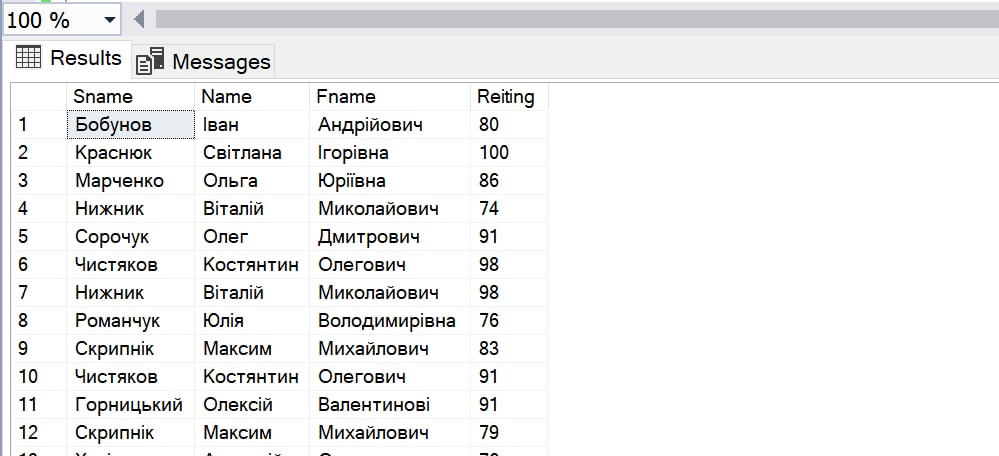


Рис. 6.3 Результат виконання процедури.

/\*Процедура виводить суму балів та її значення в національній системі та ECTS.\*/

CREATE PROC OcinkaNationalAandECTS AS

SELECT R.Name\_ini,

CASE

WHEN R.AverageReitingWithRetakes > 90 THEN 'Відмінно'

WHEN R.AverageReitingWithRetakes > 80 THEN 'Добре'

WHEN R.AverageReitingWithRetakes > 65 THEN 'Добре'

WHEN R.AverageReitingWithRetakes > 55 THEN 'Задовільно'

WHEN R.AverageReitingWithRetakes > 50 THEN 'Задовільно'

WHEN R.AverageReitingWithRetakes > 35 THEN 'Незадовільно'

WHEN R.AverageReitingWithRetakes > 0 THEN 'Незадовільно'

END NationalReiting,

CASE

WHEN R.AverageReitingWithRetakes > 90 THEN 'A'

WHEN R.AverageReitingWithRetakes > 80 THEN 'B'

WHEN R.AverageReitingWithRetakes > 65 THEN 'C'

WHEN R.AverageReitingWithRetakes > 55 THEN 'D'

WHEN R.AverageReitingWithRetakes > 50 THEN 'E'

WHEN R.AverageReitingWithRetakes > 35 THEN 'FX'

WHEN R.AverageReitingWithRetakes > 0 THEN 'F'

END ECTS

FROM (SELECT S.Name\_ini, AVG(R.Reiting) as AverageReitingWithRetakes FROM

dbo\_student S

JOIN Reiting R ON S.Kod\_stud = R.Kod\_student

JOIN Rozklad\_pids RP ON R.K\_zapis = RP.K\_zapis

JOIN Predmet\_plan PP ON RP.K\_predm\_pl = PP.K\_predm\_pl

JOIN predmet P ON PP.K\_predmet = P.K\_predmet

GROUP BY S.Name\_ini) AS R

EXEC OcinkaNationalAandECTS

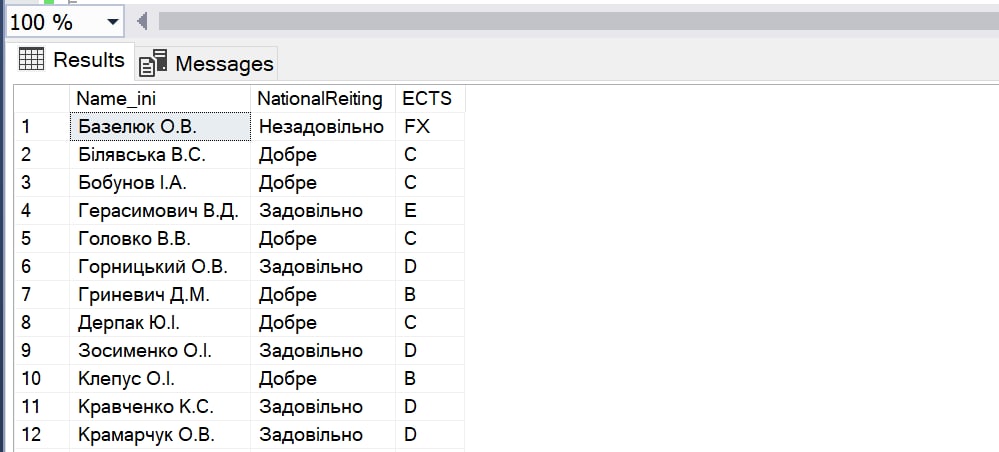


Рис. 6.4 Результат виконання процедури.

/\*Тригер на вставку даних в таблицю студент – якщо код групи новий в таблицю додається група.\*/

CREATE TRIGGER Insert\_Student ON dbo\_student INSTEAD OF INSERT AS

IF NOT EXISTS(SELECT \* FROM inserted I

JOIN dbo\_groups G on I.Kod\_group = G.Kod\_group)

BEGIN

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

SELECT Kod\_group, 13

FROM inserted

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

SELECT Sname, [Name], Fname, N\_ingroup, Kod\_group

FROM inserted

END

ELSE

BEGIN

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

SELECT Sname, Name, Fname, N\_ingroup, Kod\_group

FROM inserted

END

6.

/\*Тригер на модифікацію даних з таблиці студенти якщо більше немає студентів в групі група знищується.\*/

CREATE TRIGGER Update\_STUDENT ON dbo\_student AFTER UPDATE AS

IF EXISTS(SELECT \* FROM dbo\_groups G

LEFT JOIN dbo\_student S ON S.Kod\_group = G.Kod\_group

WHERE Kod\_stud IS NULL)

BEGIN

DELETE FROM dbo\_groups

WHERE Kod\_group IN

(SELECT G.Kod\_group FROM dbo\_groups G

LEFT JOIN dbo\_student S ON S.Kod\_group = G.Kod\_group

WHERE Kod\_stud IS NULL)

END

Для БД за індивідуальним завданням:

/\*Процедура запиту для пошуку за вказаним параметом.\*/

CREATE PROCEDURE Procedure1 AS

BEGIN

SELECT \* FROM Pruchuna\_vidrahuvanna WHERE nazva\_vidrah LIKE 'По%'

END;

EXEC Procedure1

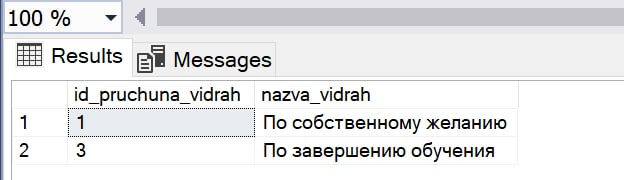


Рис. 6.5 Результат виконання процедури.

/\*Процедура запиту для оновлення запису.\*/

CREATE PROCEDURE Procedure2 AS

BEGIN

INSERT INTO

Lab6StudentZaVariantom.dbo.grupa (shufr\_group,forma\_navchanna, id\_specialnosti)

Values ('ИПЗ-19-3', 'Заочная','4' );

UPDATE grupa SET shufr\_group=1 WHERE id\_specialnosti='4'

END;

EXEC Procedure2

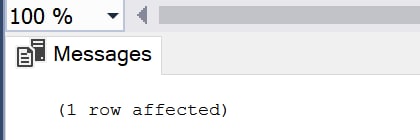


Рис. 6.6 Результат виконання процедури.

/\*Процедура запиту для видалення запису.\*/

CREATE PROCEDURE Procedure3 AS

BEGIN

DELETE FROM kafedra WHERE surname\_zav\_kafedru='Гнатюк';

END;

EXEC Procedure3

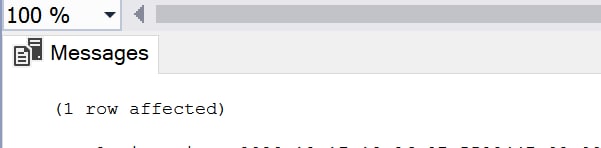


Рис. 6.7 Результат виконання процедури.

/\*Тригер для виведення інформації про помилку при додаванні нового, але вже існуючого запису.\*/

CREATE TRIGGER Insert\_Pruchuna\_vidrahuvanna ON Pruchuna\_vidrahuvanna FOR INSERT AS

IF EXISTS(SELECT \* FROM inserted I

JOIN Pruchuna\_vidrahuvanna T ON I.[nazva\_vidrah] = T.[nazva\_vidrah])

BEGIN

ROLLBACK TRAN PRINT 'Ви додаєте вже існуючу причину відрахування'

END

/\*Тригер для виведення інформації про неможливість видалення певного запису.\*/

CREATE TRIGGER Delete\_Student ON Student FOR DELETE AS

IF NOT EXISTS(SELECT \* FROM deleted D

WHERE adresa\_st like '%Киев')

BEGIN

ROLLBACK TRAN PRINT 'Студент проживає в Києві. Помилка видалення'

END

/\*Тригер для виведення інформації при неправильному заповненні поля "Дата народження".\*/

CREATE TRIGGER Vozrast ON dbo\_Student FOR INSERT AS

IF EXISTS((SELECT count(\*)

FROM Student

WHERE datanarod\_st(current\_date(), interval 16 year) < new.birthday) > 0)

BEGIN

ROLLBACK TRAN PRINT 'Вступнику не має 16 років';

END

**Висновок**: в ході виконання лабороторної роботи, було досліджено властивості та можливості процедур, що зберігаються та тригерів.