**Лабораторна робота №4-5**

**Тема: ВИКОРИСТАННЯ SELECT. ЗАПИТИ НА ВИБІБРУ ІЗ ГРУПОВИМИ ОПЕРАЦІЯМИ. ПІДЗАПИТИ У МОВІ SQL**

***Мета роботи:*** дослідження властивостей DML-інструкції SELECT для групування дани та відображення проміжних підсумків.

Хід роботи

**Завдання на лабораторну роботу:**

1. Створити самостійно наступні запити БД «Торгівельна фірма»:
   * вивести загальну кількість товарів на підприємстві.
   * вивести загальну кількість співробітників підприємства.
   * вивести загальну кількість постачальників підприємства.
   * вивести кількість за кожним товаром, що придбані у поточному місяці.
   * вивести суму, на яку були придбані товари у поточному місяці.
   * вивести суму продажу товарів за кожним постачальником.
   * вивести загальну кількість замовлень за кожним постачальником, що продає молоко.
   * вивести середню суму, на яку замовлявся товар.
   * вивести вартість замовлень усіх клієнтів, що мешкають у Житомирі.
   * вивести середню ціну на товари по кожному постачальнику.
2. Створити для БД «Рейтинг» наступні запити (база даних додається до лабораторної роботи):
   * Сумарний рейтинг студента з кожної дисципліни.
   * Розрахувати кількість студентів у кожній групі.
   * Розрахувати кількість дисциплін за групою.
   * Розрахувати кількість проведених занять у кожній групі.
   * Розрахувати середній бал за групою.
   * Розрахувати середній бал з дисципліни.
   * Розрахувати поточний рейтинг студента з кожної дисципліни.
   * Відобразити найменший рейтинг студентів з дисципліни.
   * Відобразити найбільший студентський рейтинг з дисципліни.
   * Розрахувати кількість проведених занять за видами для кожної дисципліни.
   * Розрахувати кількість груп за кожною спеціальністю.
   * Запит на знищення даних з таблиці «Reiting» за визначеним кодом студента (в поле параметра вводиться прізвище студента).
   * Запит на знищення даних з таблиці «Para» за визначеним кодом дисци-

пліни (у поле параметра вводиться назва дисципліни).

* + Запит на оновлення даних у таблиці «Reiting» – передбачити збільшення балів за модульні контролі на 15%.
  + Запит на оновлення даних в таблиці «Reiting»– передбачити зменшення балів за іспит на 15%.
  + Запит на вставку даних до таблиці «Reiting» – передбачити вставку даних студентів визначеної групи (код пари та початковий бал задається динамічно).
  + Запит на вставку даних до таблиці «Para» – передбачити вставку всіх дисциплін, назва яких починається з літери «М» (дата заняття та години задаються динамічно, однакові для всіх).
  + Запит на оновлення даних – передбачити зміну порядкового номера змістовного модуля за певною дисципліною на нове значення.
  + Передбачити знищення студентів з таблиці «Students» за визначеним номером групи.
  + Запит на вставку даних до таблиці «Reiting» – передбачити вставку даних студентів визначеної групи (код пари та присутність задаються динамічно)
  + Передбачити оновлення даних у таблиці «Reiting» − поле «Prisutnist», для студента із визначеним прізвищем автоматично встановлюється true.

1. Створити запити до БД у відповідності до індивідуального завдання:
   * Запити, що наводяться маркованим переліком у варіанті
   * Запити на вставку та оновлення даних – для одного запису в кожній таблиці
   * Запит на вставку та оновлення даних - для множини записів (можна створювати додаткові таблиці)

**Виконання:**

/\* вивести загальну кількість товарів на підприємстві\*/

SELECT SUM(tovar.NaSklade) as tovar\_sum FROM tovar GROUP BY id\_postav;

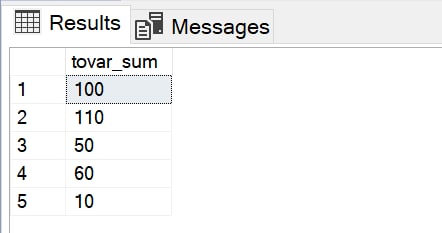


Рис. 4.1 Результат виконання запиту.

/\* вивести загальну кількість співробітників підприємства \*/ ---

select Count(sotrudnik.id\_sotrud) as Sotrudnikiv

from sotrudnik;

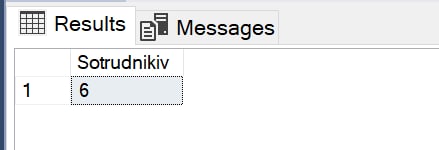


Рис. 4.2 Результат виконання запиту.

/\* вивести загальну кількість постачальників підприємства. \*/

select Count(postachalnik.id\_postach) as Postachalnikiv

from postachalnik;

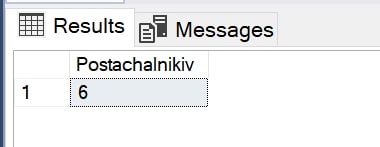


Рис. 4.3 Результат виконання запиту.

/\* вивести кількість за кожним товаром, що придбані у поточному місяці \*/

select sum(zakaz\_tovar.Kilkist) as Kilkist, tovar.id\_tovar

from tovar inner join zakaz\_tovar on tovar.id\_tovar = zakaz\_tovar.id\_tovar

inner join zakaz on zakaz\_tovar.id\_tovar = zakaz.id\_zakaz

where zakaz.date\_naznach>dateadd(MONTH, -12, getdate())

group by tovar.id\_tovar

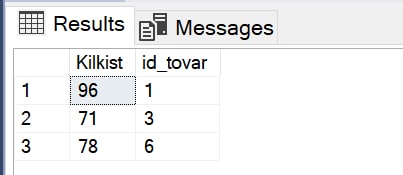


Рис. 4.4 Результат виконання запиту.

/\* вивести суму, на яку були придбані товари у поточному місяці.\*/

select Sum(tovar.Price) as SUMA

from tovar inner join zakaz\_tovar on tovar.id\_tovar = zakaz\_tovar.id\_tovar

inner join zakaz on zakaz\_tovar.id\_zakaz = zakaz.id\_zakaz

where zakaz.date\_naznach>dateadd(MONTH, -12, getdate())

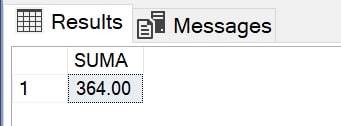


Рис. 4.5 Результат виконання запиту.

/\* вивести суму продажу товарів за кожним постачальником.\*/

SELECT postachalnik.Nazva, tovar.Nazva, SUM(tovar.Price)

FROM tovar INNER JOIN postachalnik ON tovar.id\_postav = postachalnik.id\_postach

GROUP BY postachalnik.Nazva, tovar.Nazva;

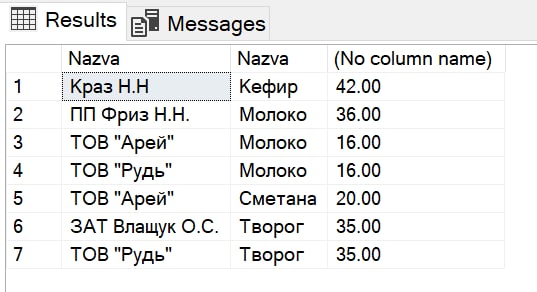


Рис. 4.6 Результат виконання запиту.

/\* вивести загальну кількість замовлень за кожним постачальником, що продає молоко.\*/

SELECT p.Nazva, count(zakaz.id\_zakaz)

FROM zakaz

INNER JOIN zakaz\_tovar ON zakaz\_tovar.id\_zakaz=zakaz.id\_zakaz

INNER JOIN tovar ON zakaz\_tovar.id\_tovar=tovar.id\_tovar

INNER JOIN postachalnik p ON p.id\_postach=tovar.id\_postav

WHERE tovar.Nazva='Молоко'

GROUP BY p.Nazva;

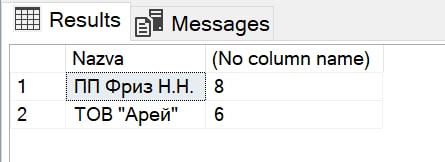


Рис. 4.7 Результат виконання запиту.

/\* вивести середню суму, на яку замовлявся товар. \*/

SELECT tovar.Nazva, AVG(tovar.Price) AS СередняСума

FROM tovar

Group By tovar.Nazva;

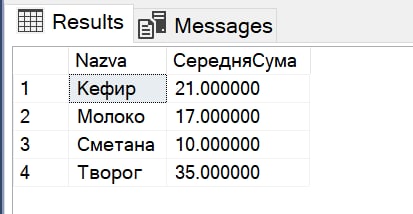


Рис. 4.8 Результат виконання запиту.

/\* вивести вартість замовлень усіх клієнтів, що мешкають у Житомирі.\*/

SELECT klient.Nazva, SUM(zakaz\_tovar.Kilkist \* tovar.Price) AS total\_price

from (klient INNER JOIN zakaz ON klient.id\_klient = zakaz.id\_klient)

INNER JOIN zakaz\_tovar ON zakaz.id\_zakaz = zakaz\_tovar.id\_zakaz

INNER JOIN tovar ON tovar.id\_tovar = zakaz\_tovar.id\_tovar

WHERE klient.City = 'Житомир'

GROUP BY klient.Nazva;

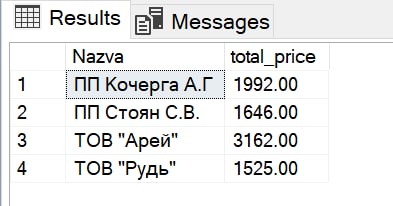


Рис. 4.9 Результат виконання запиту.

/\* вивести середню ціну на товари по кожному постачальнику.\*/

SELECT postachalnik.Nazva, AVG(tovar.Price) AS СередняСума

FROM tovar INNER JOIN postachalnik ON postachalnik.id\_postach = tovar.id\_postav

Group By postachalnik.Nazva;

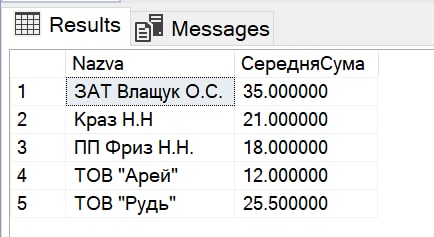


Рис. 4.10 Результат виконання запиту.

/\*Сумарний рейтинг студента з кожної дисципліни.\*/

SELECT dbo\_student.Name\_ini, sum(Reiting) AS SummReiting, predmet.Nazva

FROM dbo\_groups

INNER JOIN Predmet\_plan ON dbo\_groups.K\_navch\_plan = Predmet\_plan.K\_navch\_plan

INNER JOIN predmet ON Predmet\_plan.K\_predmet = predmet.K\_predmet

INNER JOIN dbo\_student ON dbo\_groups.Kod\_group = dbo\_student.Kod\_group

INNER JOIN Reiting ON Reiting.Kod\_student = dbo\_student.Kod\_stud

GROUP BY Kod\_student, dbo\_student.Name\_ini, predmet.Nazva

ORDER BY Kod\_student

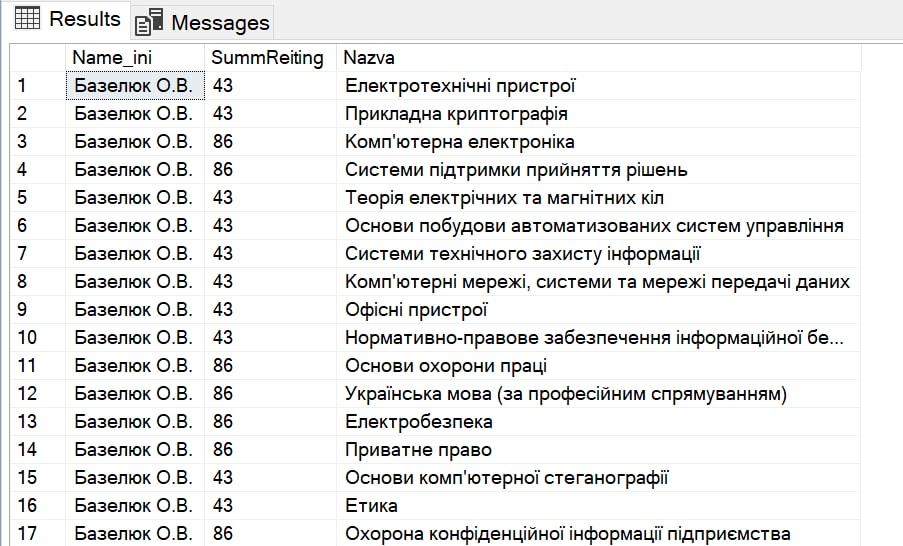


Рис. 4.11 Результат виконання запиту.

/\*Розрахувати кількість студентів у кожній групі.\*/

SELECT dbo\_groups.Kod\_group, COUNT(dbo\_student.Name) AS CountStuden

FROM dbo\_groups INNER JOIN dbo\_student ON dbo\_student.Kod\_group = dbo\_groups.Kod\_group

GROUP BY dbo\_groups.Kod\_group;

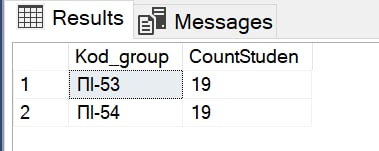


Рис. 4.12 Результат виконання запиту.

/\*Розрахувати кількість дисциплін за групою.\*/

SELECT q.Kod\_group, COUNT(q.Nazva) AS CountSubjects FROM (SELECT DISTINCT dbo\_groups.Kod\_group, predmet.Nazva

FROM (dbo\_groups INNER JOIN Rozklad\_pids ON Rozklad\_pids.Kod\_group = dbo\_groups.Kod\_group)

INNER JOIN Predmet\_plan ON Predmet\_plan.K\_predm\_pl = Rozklad\_pids.K\_predm\_pl

INNER JOIN predmet ON predmet.K\_predmet = Predmet\_plan.K\_predmet) As q

GROUP BY q.Kod\_group;

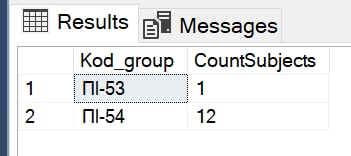


Рис. 4.13 Результат виконання запиту.

/\*Розрахувати кількість проведених занять у кожній групі.\*/

SELECT Rozklad\_pids.Kod\_group, Count(Rozklad\_pids.K\_zapis) AS CountLessons

from Rozklad\_pids

GROUP BY Rozklad\_pids.Kod\_group;

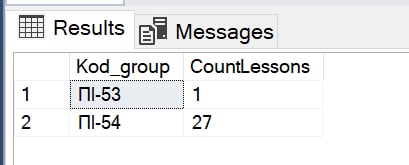


Рис. 4.14 Результат виконання запиту.

/\*Розрахувати середній бал за групою.\*/

SELECT dbo\_groups.Kod\_group, AVG(Reiting.Reiting) AS СередБал

FROM (dbo\_groups INNER JOIN dbo\_student ON dbo\_student.Kod\_group = dbo\_groups.Kod\_group)

INNER JOIN Reiting ON Reiting.Kod\_student = dbo\_student.Kod\_stud

GROUP BY dbo\_groups.Kod\_group;

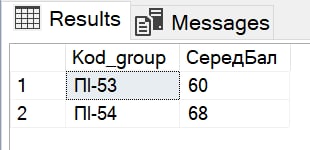


Рис. 4.15 Результат виконання запиту.

/\*Розрахувати середній бал з дисципліни.\*/

SELECT

predmet.Nazva,

AVG(Reiting.Reiting) AS СередБал

FROM

predmet

INNER JOIN Predmet\_plan ON Predmet\_plan.K\_predmet = predmet.K\_predmet

INNER JOIN Rozklad\_pids ON Rozklad\_pids.K\_predm\_pl = Predmet\_plan.K\_predm\_pl

INNER JOIN Reiting ON Reiting.K\_zapis = Rozklad\_pids.K\_zapis

GROUP BY predmet.Nazva;

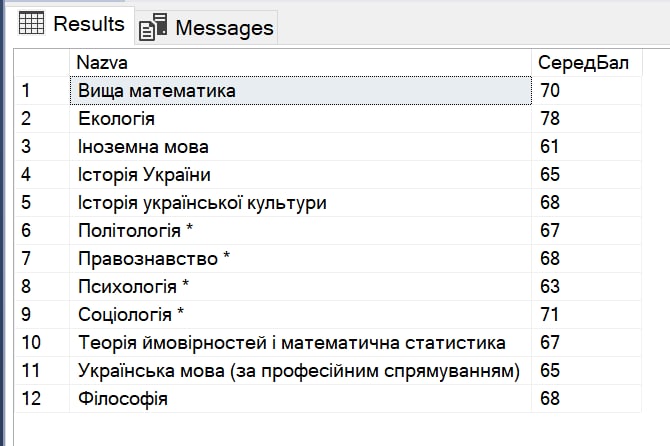


Рис. 4.16 Результат виконання запиту.

/\*Розрахувати поточний рейтинг студента з кожної дисципліни.\*/

SELECT dbo\_student.Name\_ini, AVG (Reiting.reiting) AS Reiting, predmet.Nazva

FROM dbo\_student

INNER JOIN Reiting ON Reiting.Kod\_student = dbo\_student.Kod\_stud

INNER JOIN Rozklad\_pids ON Rozklad\_pids.K\_zapis = Reiting.K\_zapis

INNER JOIN Predmet\_plan ON Predmet\_plan.K\_predm\_pl = Rozklad\_pids.K\_predm\_pl

INNER JOIN predmet ON predmet.K\_predmet = Predmet\_plan.K\_predmet

where dbo\_student.Name\_ini = 'Шелуха О.О.'

GROUP BY dbo\_student.Name\_ini, predmet.Nazva;

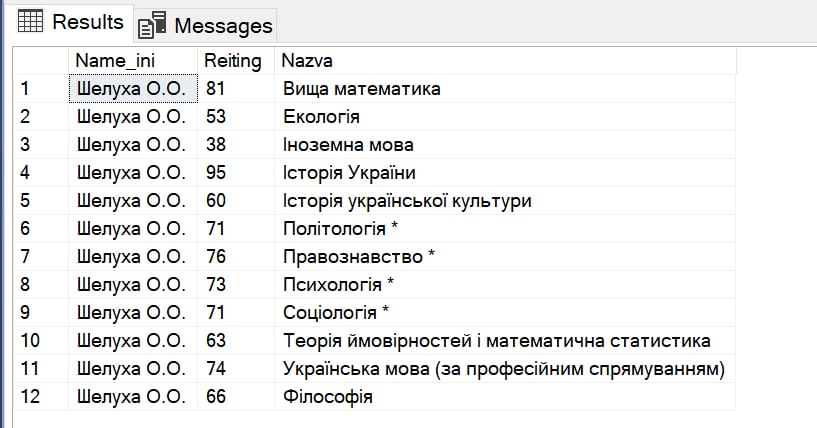


Рис. 4.17 Результат виконання запиту.

/\*Відобразити найменший рейтинг студентів з дисципліни.\*/

SELECT predmet.Nazva, Min(Reiting.Reiting) AS MinReiting

FROM predmet

INNER JOIN Predmet\_plan ON Predmet\_plan.K\_predmet = predmet.K\_predmet

INNER JOIN Rozklad\_pids ON Rozklad\_pids.K\_predm\_pl = Predmet\_plan.K\_predm\_pl

INNER JOIN Reiting ON Reiting.K\_zapis = Rozklad\_pids.K\_zapis

GROUP BY predmet.Nazva

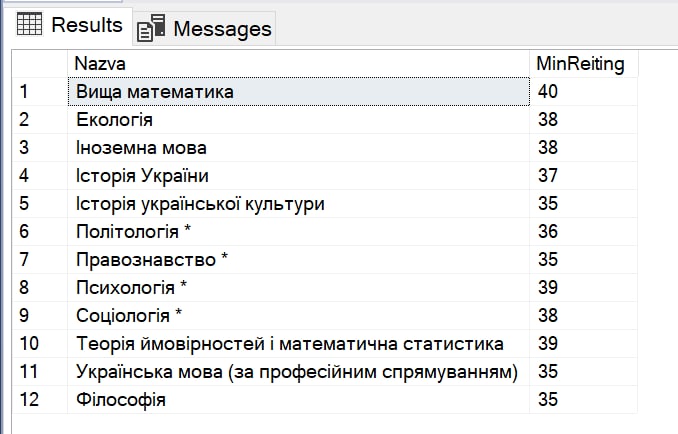


Рис. 4.18 Результат виконання запиту.

/\*Відобразити найбільший студентський рейтинг з дисципліни.\*/

SELECT predmet.Nazva, Max(Reiting.Reiting) AS MaxReiting

FROM predmet

INNER JOIN Predmet\_plan ON Predmet\_plan.K\_predmet = predmet.K\_predmet

INNER JOIN Rozklad\_pids ON Rozklad\_pids.K\_predm\_pl = Predmet\_plan.K\_predm\_pl

INNER JOIN Reiting ON Reiting.K\_zapis = Rozklad\_pids.K\_zapis

GROUP BY predmet.Nazva

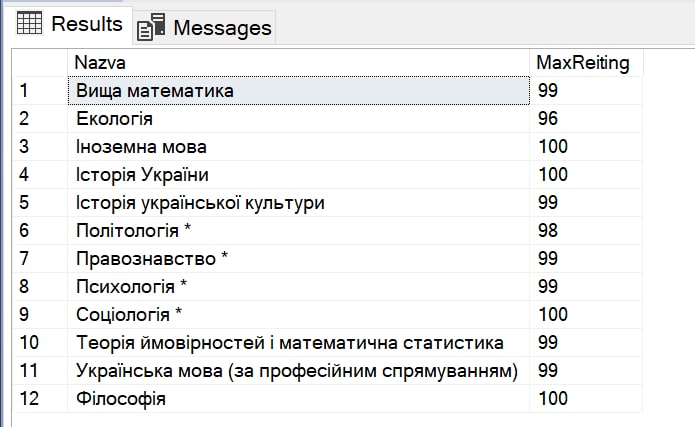


Рис. 4.19 Результат виконання запиту.

/\*Розрахувати кількість проведених занять за видами для кожної дисципліни.\*/

SELECT predmet.Nazva, COUNT (Rozklad\_pids.K\_zapis) as heldLessons

FROM predmet

INNER JOIN Predmet\_plan ON Predmet\_plan.K\_predmet = predmet.K\_predmet

INNER JOIN Rozklad\_pids ON Rozklad\_pids.K\_predm\_pl = Predmet\_plan.K\_predm\_pl

GROUP BY predmet.Nazva;

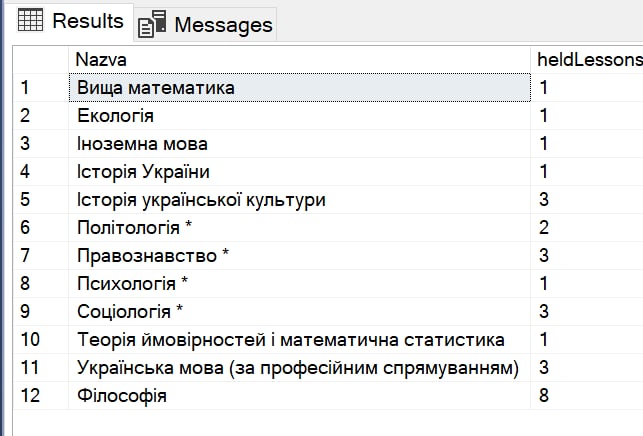


Рис. 4.20 Результат виконання запиту.

/\*Розрахувати кількість груп за кожною спеціальністю.\*/

SELECT Spetsialnost.Nazva, COUNT(dbo\_groups.Kod\_group) AS kilkistGroup

FROM Spetsialnost

INNER JOIN Navch\_plan ON Navch\_plan.K\_spets = Spetsialnost.K\_spets

INNER JOIN dbo\_groups ON dbo\_groups.K\_navch\_plan = Navch\_plan.K\_navch\_plan

GROUP BY Spetsialnost.Nazva

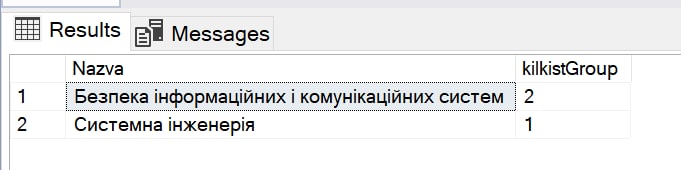


Рис. 4.21 Результат виконання запиту.

/\*Запит на знищення даних з таблиці «Reiting» за визначеним кодом студента (в поле параметра вводиться прізвище студента).\*/

DELETE r from Reiting r INNER JOIN dbo\_student s ON r.Kod\_student='55' WHERE s.Sname='Маляренко';

/\*Запит на оновлення даних у таблиці «Reiting» – передбачити збільшення балів за модульні контролі на 15%.\*/

update Reiting set Reiting.Reiting = (Reiting.Reiting \* 1.15)

where Reiting.K\_zapis in (select Rozklad\_pids.K\_zapis from Rozklad\_pids where month(Rozklad\_pids.Date) = 12);

/\*Запит на оновлення даних в таблиці «Reiting»– передбачити зменшення балів за іспит на 15%.\*/

update Reiting set Reiting.Reiting = (Reiting.Reiting \* 0.75)

where Reiting.K\_zapis in (select Rozklad\_pids.K\_zapis from Rozklad\_pids

inner join Predmet\_plan on Predmet\_plan.K\_predm\_pl = Rozklad\_pids.K\_predm\_pl

inner join form\_kontr on form\_kontr.k\_fk = Predmet\_plan.k\_fk);

/\*Запит на вставку даних до таблиці «Reiting» – передбачити вставку даних студентів визначеної групи (код пари та початковий бал задається динамічно).\*/

insert into Reiting (Kod\_student, Reiting)

values ('kod\_stud', 'bal');

/\*Запит на оновлення даних – передбачити зміну порядкового номера змістовного модуля за певною дисципліною на нове значення.\*/

update Rozklad\_pids set Rozklad\_pids.N\_vedomost = Null

where Rozklad\_pids.K\_predm\_pl in (

select Predmet\_plan.K\_predm\_pl from Predmet\_plan

inner join predmet on predmet.K\_predmet = Predmet\_plan.K\_predmet

where predmet.Nazva = 'Системи технічного захисту інформації');

/\*Передбачити знищення студентів з таблиці «Students» за визначеним номером групи.\*/

delete from dbo\_student

where dbo\_student.Kod\_group = kod\_group;

/\*Запит на вставку даних до таблиці «Reiting» – передбачити вставку даних студентів визначеної групи (код пари та присутність задаються динамічно) \*/

insert into Reiting (Kod\_student, Prisutn, K\_zapis)

select dbo\_student.Kod\_stud, prisutn, k\_zapis from dbo\_student

inner join dbo\_groups on dbo\_groups.Kod\_group = dbo\_student.Kod\_group;

/\*Передбачити оновлення даних у таблиці «Reiting» поле «Prisutnist», для студента із визначеним прізвищем автоматично встановлюється true.\*/

update Reiting set Reiting.Prisutn = 1

where Reiting.Kod\_student in (select dbo\_student.Kod\_stud from dbo\_student where dbo\_student.SName='Кравченко');

/\*Запити на вставку та оновлення даних – для одного запису в кожній таблиці\*/

UPDATE kafedra

SET name\_zav\_kafedru = 'Владимир'

WHERE name\_zav\_kafedru = 'Олег'

UPDATE Pruchuna\_vidrahuvanna

SET nazva\_vidrah = 'Аналітична геометрія'

WHERE nazva\_vidrah = 'Прикладная математика'

UPDATE Specialnist

SET shufr\_specialnosti = '110'

WHERE id\_kafedru = '1'

UPDATE grupa

SET shufr\_group = 'ІПЗ-19'

WHERE id\_specialnosti= '4'

UPDATE Student

SET id\_pruchuna\_vidrah = '2'

WHERE nom\_zalikovki= '0349243'

insert into StudentsLab2.dbo.kafedra (name\_zav\_kafedru,surname\_zav\_kafedru,middlename\_zav\_kafedru,kabinet,telefon)

values ('Кафедра физики и высшей математикие', 'Иван', 'Иконников', 'Сергеевич', '117', '(095)8631578');

Insert into StudentsLab2.dbo.Pruchuna\_vidrahuvanna (nazva\_vidrah)

Values ('По состоянию здоровья');

Insert into StudentsLab2.dbo.Specialnist (nazva\_specialnosti,shufr\_specialnosti,kvalifikacia, id\_kafedru)

Values ('Аналитическая геометрия', '116', 'Специалист в сфере аналитики', '2');

Insert into StudentsLab2.dbo.grupa (shufr\_group,forma\_navchanna, id\_specialnosti)

Values ('ИПЗ-19-3', 'Заочная','4' );

Insert into

StudentsLab2.dbo.Student (name\_st,surname\_st,middlename\_st,datanarod\_st,adresa\_st,tel\_st,rik\_vstupu\_st,byudzhet\_contrack,nom\_zalikovki,data\_vidrahuvanna, id\_pruchuna\_vidrah,id\_group)

Values ('Антон', 'Назаренко', 'Иванович', '2002-04-02', 'Житомирская область, город Киев, ул. Генерала Жадова, 81', '(067)6539039', '2016-09-01', 'Бюджет', '033477', '2021-06-30','2','3');

/\*Запит на вставку та оновлення даних - для множини записів\*/

INSERT INTO

StudentsLab2.dbo.grupa (shufr\_group,forma\_navchanna, id\_specialnosti)

Values ('ИПЗ-19-3', 'Заочная','4' );

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

**Висновок**: в ході виконання лабороторної роботи, було досліджено властивості DML-інструкції SELECT для групування дани та відображення проміжних підсумків.