

Міністерство освіти і науки України
Національний технічний університет України „Київський політехнічний
інститут імені Ігоря Сікорського"
Факультет інформатики та обчислювальної техніки

Кафедра ІІІ

Звіт

з лабораторної роботи № 3 з дисципліни
«Бази даних»

„Побудова простих запитів”

Виконав(ла)

ІІ-14 Шляхтун Денис Михайлович
(шифр, прізвище, ім'я, по батькові)

Перевірів

Ліщук Олександр Васильович
(прізвище, ім'я, по батькові)

Київ 2022

Лабораторна робота №3

Варіант 19

Тема: Побудова простих запитів.

Мета:

- Вивчити оператор, котрий використовується в реляційних СУБД, для вибірки даних з таблиць.
- Вивчити команди SQL для створення запитів з використанням підзапитів та з'єднань.

Постановка задачі:

1. Створити запити для вибірки даних з використанням (разом 12 запитів):
 - a. Найпростіших умов
 - b. Операторів порівняння
 - c. Умов з використанням логічних операторів AND, OR та NOT.
 - d. Умов з використанням комбінацій логічних операторів
 - e. З використанням виразів над стовпцями, як в якості новостворених стовпців, так і умовах
 - f. Використання операторів:
 - i. Приналежності множині
 - ii. Приналежності діапазону
 - iii. Відповідності шаблону
 - iv. Перевірка на невизначене значення
2. Створити запити з використанням підзапитів та з'єднань (разом 11 запитів) (в запитах повинні використовуватись 2 та більше таблиць):
 - a. Використання підзапитів в рядку вибірки полів (у секції select) та вибірки з таблиць (у секції from)
 - b. Використання підзапитів в умовах з конструкціями EXISTS, IN
 - c. Декартовий добуток

- d. З'єднання декількох таблиць за рівністю
- e. З'єднання декількох таблиць за рівністю та умовою відбору
- f. Внутрішнього з'єднання
- g. Лівого зовнішнього з'єднання
- h. Правого зовнішнього з'єднання
- i. Об'єднання запитів

Виконання завдання.

SQL-скрипти.

1. Створити запити для вибірки даних з використанням (разом 12 запитів):

- a. Найпростіших умов

```
SELECT * FROM KEKW WHERE Code = 2000
```

- b. Операторів порівняння

```
SELECT * FROM Platizhne_doruchennia WHERE Sum > 1000
SELECT Name FROM Budget_institution WHERE Pidrozdil_ID = 6;
SELECT * FROM Commercial_bank WHERE Code < 310000
```

- c. Умов з використанням логічних операторів AND, OR та NOT.

```
SELECT * FROM KEKW WHERE NOT Code > 3000
```

- d. Умов з використанням комбінацій логічних операторів

```
SELECT * FROM Budget_institution WHERE Pidrozdil_ID = 10 OR (Pidrozdil_ID > 5 AND NOT Pidrozdil_ID > 7)
```

- e. З використанням виразів над стовпцями, як в якості новостворених стовпців, так і умовах

```
SELECT KEKW_Code, SUM(Sum) AS 'KEKW_Sum' FROM Platizhne_doruchennia GROUP BY KEKW_Code HAVING SUM(Sum) > 1000
```

- f. Використання операторів:

- i. Приналежності множині

```
SELECT * FROM Commercial_bank WHERE Commercial_bank.Name IN (N'Ощадбанк', N'Укргазбанк')
```

- ii. Приналежності діапазону

```
SELECT * FROM Budget_institution WHERE Pidrozdil_ID BETWEEN 6 AND 8
```

iii. Відповідності шаблону

```
SELECT * FROM Pidrozdil_kaznacheystva WHERE Territory LIKE N'% , м. _Киї[абв]%' ;  
SELECT * FROM Commercial_bank WHERE Commercial_bank.Name LIKE N'%укр[ес]%'
```

iv. Перевірка на невизначене значення

```
SELECT * FROM Commercial_bank WHERE Code IS NOT NULL
```

2. Створити запити з використанням підзапитів та з'єднань (разом 11 запитів) (в запитах повинні використовуватись 2 та більше таблиць):

a. Використання підзапитів в рядку вибірки полів (у секції select) та вибірки з таблиць (у секції from)

```
SELECT Budget_institution.Name AS 'Budget_institution',  
       (SELECT Name FROM Pidrozdil_kaznacheystva WHERE  
        Budget_institution.Pidrozdil_ID = ID) AS 'Pidrozdil_kaznacheystva'  
FROM Budget_institution  
SELECT * FROM Budget_institution WHERE Pidrozdil_ID IN (SELECT ID FROM  
Pidrozdil_kaznacheystva)
```

b. Використання підзапитів в умовах з конструкціями EXISTS,
IN

```
SELECT * FROM Pidrozdil_kaznacheystva WHERE NOT EXISTS (SELECT * FROM  
Budget_institution WHERE Pidrozdil_ID = Pidrozdil_kaznacheystva.ID)  
SELECT * FROM Pidrozdil_kaznacheystva WHERE ID NOT IN (SELECT Pidrozdil_ID FROM  
Budget_institution)
```

c. Декартовий добуток

```
SELECT Budget_institution.Name AS 'Budget institution', Commercial_bank.Name AS  
'Bank' FROM Budget_institution CROSS JOIN Commercial_bank
```

d. З'єднання декількох таблиць за рівністю

```
SELECT Budget_institution.Name, Pidrozdil_kaznacheystva.Territory FROM  
Budget_institution, Pidrozdil_kaznacheystva WHERE Pidrozdil_ID =  
Pidrozdil_kaznacheystva.ID
```

e. З'єднання декількох таблиць за рівністю та умовою відбору

```
SELECT ID, Date, Sum, Name, EDRPOU FROM Platizhne_doruchennia, Commercial_bank  
WHERE Bank_Code = Code AND Sum > 5000
```

f. Внутрішнього з'єднання

```
SELECT Platizhne_doruchennia.ID, Date, Sum, KEKW_Code, Name FROM
Platizhne_doruchennia INNER JOIN Budget_institution ON Institution_ID =
Budget_institution.ID
```

г. Лівого зовнішнього з'єднання

```
SELECT Pidrozdil_kaznacheystva.Name, Budget_institution.Name FROM
Pidrozdil_kaznacheystva LEFT OUTER JOIN Budget_institution ON Pidrozdil_ID =
Pidrozdil_kaznacheystva.ID
```

h. Правого зовнішнього з'єднання

```
SELECT Pidrozdil_kaznacheystva.Name, Budget_institution.Name FROM
Pidrozdil_kaznacheystva RIGHT OUTER JOIN Budget_institution ON Pidrozdil_ID =
Pidrozdil_kaznacheystva.ID
```

і. Об'єднання запитів

```
SELECT
    Platizhne_doruchennia.ID,
    Date,
    (SELECT Name FROM Budget_institution WHERE Institution_ID =
    Budget_institution.ID) AS 'Budget institution',
    Recipient,
    Purpose,
    KEKW_Code,
    (SELECT Name FROM Commercial_bank WHERE Code =
    Platizhne_doruchennia.Bank_Code) AS 'Bank',
    Pidrozdil_kaznacheystva.Name AS 'Kaznacheystvo'
FROM Platizhne_doruchennia LEFT OUTER JOIN Pidrozdil_kaznacheystva ON
Pidrozdil_ID = Pidrozdil_kaznacheystva.ID
```

Висновок.

У ході виконання лабораторної роботи було здобуто навички використання оператора для вибірки даних з таблиць в реляційних СУБД. При виконанні роботи було розроблено скрипти з використанням оператора SELECT відповідно до постановки задачі, зокрема з використанням підзапитів та різних видів з'єднань.