

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 № 5 з дисципліни
«Бази даних»

**„Основи програмування з використанням мови SQL. Збережені процедури.
Курсори. Створення, програмування та керування тригерами.”**

Виконав(ла)

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

Перевірів

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

Київ 2022

Лабораторна робота №5

Варіант 19

Тема: Основи програмування з використанням мови SQL. Збережені процедури. Курсори. Створення, програмування та керування тригерами.

Мета:

- Вивчити правила побудови ідентифікаторів, правила визначення змінних та типів. Визначити правила роботи з циклами та умовними конструкціями, роботу зі змінними типу Table.
- Вивчити синтаксис та семантику функцій та збережених процедур, способів їх ідентифікації, методів визначення та специфікації параметрів та повертаємих значень, виклик функцій та збережених процедур.
- Застосування команд для створення, зміни та видалення як скалярних, так і табличних функцій, збережених процедур.
- Вивчити призначення та типи курсорів, синтаксис та семантику команд мови SQL для створення курсорів, вибірки даних з курсорів, зміни даних із застосуванням курсорів.
- Вивчити призначення та типи тригерів, умов їх активації, синтаксису та семантики для їх створення, модифікації, перейменування, програмування та видалення.

Постановка задачі:

1. Збережені процедури:
 - a. запит для створення тимчасової таблиці через змінну типу TABLE;
 - b. запит з використанням умовної конструкції IF;
 - c. запит з використанням циклу WHILE;
 - d. створення процедури без параметрів;
 - e. створення процедури з вхідним параметром;
 - f. створення процедури з вхідним параметром та RETURN;
 - g. створення процедури оновлення даних в деякій таблиці БД;

h. створення процедури, в котрій робиться вибірка даних.

2. Функції:

- a. створити функцію, котра повертає деяке скалярне значення;
- b. створити функцію, котра повертає таблицю з динамічним набором стовпців;
- c. створити функцію, котра повертає таблицю заданої структури.

3. Робота з курсорами:

- a. створити курсор;
- b. відкрити курсор;
- c. вибірка даних, робота з курсорами.

4. Робота з тригерами:

- a. створити тригер, котрий буде спрацьовувати при видаленні даних;
- b. створити тригер, котрий буде спрацьовувати при модифікації даних;
- c. створити тригер, котрий буде спрацьовувати при додаванні даних

Виконання завдання.

Збережені процедури:

Запит для створення тимчасової таблиці через змінну типу TABLE

```
DECLARE @InstPidrozdil TABLE
(
    ID INT IDENTITY (1,1) PRIMARY KEY,
    Institution NVARCHAR (500) NOT NULL,
    Pidrozdil NVARCHAR (500) NOT NULL
)
```

Запит з використанням умовної конструкції IF

```
IF (SELECT AVG(Sum) FROM Platizhne_doruchennia)>100000
    SELECT AVG(Sum) AS 'Average sum' FROM Platizhne_doruchennia
ELSE
    SELECT MIN(Sum) AS 'mIN sum' FROM Platizhne_doruchennia
```

Запит з використанням циклу WHILE

```
DECLARE @InstPidrozdil TABLE
(
    ID INT IDENTITY (1,1) PRIMARY KEY,
    Institution NVARCHAR (500) NOT NULL,
```

```

        Pidrozdil NVARCHAR (500) NOT NULL
    )
    INSERT INTO @InstPidrozdil (Institution, Pidrozdil) SELECT Budget_institution.Name,
    Pidrozdil_kaznacheystva.Name FROM Budget_institution, Pidrozdil_kaznacheystva WHERE
    Pidrozdil_ID = Pidrozdil_kaznacheystva.ID
    SELECT * FROM @InstPidrozdil
    WHILE (SELECT COUNT(*) FROM @InstPidrozdil) > 5
    BEGIN
        DELETE TOP (3) FROM @InstPidrozdil
    END
    SELECT * FROM @InstPidrozdil

```

Створення процедури без параметрів

```

CREATE PROCEDURE CheckSumPlatizh AS
BEGIN
    SELECT SUM(Sum) AS 'SUM' FROM Platizhne_doruchennia
END

```

Створення процедури з вхідним параметром

```

CREATE PROCEDURE SelectTopRowsDoruchennia
@variableInt INT
AS
BEGIN
    SELECT TOP (@variableInt) * FROM Platizhne_doruchennia
END

```

Створення процедури з вхідним параметром та RETURN

```

CREATE PROCEDURE ReturnSumTopRowsDoruchennia
@variableInt INT
AS
BEGIN
    RETURN (SELECT SUM(Sum) FROM Platizhne_doruchennia WHERE ID < @variableInt+1)
END

```

Створення процедури оновлення даних в деякій таблиці БД

```

CREATE PROCEDURE RiseKoshtorysChangeByPercent
@percent INT
AS
BEGIN
    UPDATE Koshtorys_Change SET Changed_lim = Changed_lim / 100 * (100+@percent)
END

```

Створення процедури, в котрій робиться вибірка даних

```

CREATE PROCEDURE SelectDoruchennia
AS
BEGIN
    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
END

```

Функції:

Створити функцію, котра повертає деяке скалярне значення

```
CREATE FUNCTION ReturnDateDoruchennia (@doruchenniaID INT)
RETURNS Date
AS
BEGIN
    DECLARE @date DATE
    SET @date = (SELECT Date FROM Platizhne_doruchennia WHERE ID = @doruchenniaID)
    RETURN @date
END
```

Створити функцію, котра повертає таблицю з динамічним набором
стовпців

```
CREATE FUNCTION DoruchenniaRows (@number INT)
RETURNS TABLE
AS
RETURN
(
    SELECT TOP (@number)
        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
);
```

Створити функцію, котра повертає таблицю заданої структури

```
CREATE FUNCTION InstitutionTerritory ()
RETURNS @table TABLE
(
    ID INT IDENTITY (1,1) PRIMARY KEY,
    Name NVARCHAR(200),
    Territory NVARCHAR(100)
)
AS
BEGIN
    INSERT INTO @table (Name, Territory) (SELECT Budget_institution.Name,
    Pidrozdil_kaznacheystva.Territory FROM Budget_institution JOIN Pidrozdil_kaznacheystva ON
    Pidrozdil_ID = Pidrozdil_kaznacheystva.ID)
    RETURN
END
```

Робота з курсорами:

Створити курсор

```
DECLARE curs CURSOR READ_ONLY FOR SELECT Institution_ID, Sum, Bank_Code FROM
Platizhne_doruchennia ORDER BY ID DESC;
```

Відкрити курсор

```
OPEN curs
```

Вибірка даних, робота з курсорами

```
DECLARE @counter INT
SET @counter = 10
DECLARE @inst_id INT, @sum MONEY, @bank INT
FETCH NEXT FROM curs INTO @inst_id, @sum, @bank
DECLARE @table TABLE (Institution NVARCHAR(200), Sum MONEY, Bank NVARCHAR(30))
WHILE @counter > 0 AND @@FETCH_STATUS = 0
BEGIN
    IF EXISTS(SELECT COUNT(*) FROM @table HAVING COUNT(*)=0 OR AVG(Sum)<@sum)
        INSERT INTO @table VALUES ((SELECT Name FROM Budget_institution WHERE ID =
@inst_id), @sum, (SELECT Name FROM Commercial_bank WHERE Code = @bank))
    FETCH NEXT FROM curs INTO @inst_id, @sum, @bank
    SET @counter = @counter - 1
END

CLOSE curs
DEALLOCATE curs
SELECT * FROM @table
```

Робота з тригерами:

Створити тригер, котрий буде спрацьовувати при видаленні даних

```
CREATE TRIGGER DeletePlatizh ON Platizhne_doruchennia FOR DELETE
AS SELECT * FROM Platizhne_doruchennia
```

Створити тригер, котрий буде спрацьовувати при модифікації даних

```
CREATE TRIGGER UpdateTrigger ON Koshtorys_Change FOR UPDATE
AS
BEGIN
    SELECT Year, Institution_ID, Changed_lim FROM Koshtorys JOIN Koshtorys_Change ON
Koshtorys.ID = Koshtorys_ID
END
```

Створити тригер, котрий буде спрацьовувати при додаванні даних

```
CREATE TRIGGER InsertTrigger ON Koshtorys_Change FOR INSERT
AS SELECT * FROM Koshtorys_Change
```

Висновок.

У ході виконання лабораторної роботи було здобуто навички основ програмування мови SQL. Було вивчено правила визначення змінних та типів, синтаксис і семантику функцій та збережених процедур, призначення та типи курсорів та тригерів.