Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи №5 з дисципліни

«Бази даних»

# «Основи програмування з використанням мови SQL. Збережені процедури. Курсори. Створення, програмування та керування тригерами.»

Варіант 3

Виконав студент ІП-13 Бондаренко Максим Вікторович

(шифр, прізвище, ім'я, по батькові)

Перевірив Марченко Олена Іванівна

(прізвище, ім'я, по батькові)

Київ 2022

**Лабораторна робота №5**

**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**

При виконанні лабораторної роботи необхідно виконати наступні дії:

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**

*use* db\_labs;  
  
*drop procedure if exists* create\_temp\_table;  
*create procedure* create\_temp\_table()  
*begin  
 drop table if exists* temp\_tbl;  
 *create temporary table* temp\_tbl *as  
 select* \*  
 *from client  
 limit* 10;  
*end*;  
*call* create\_temp\_table();  
  
*drop procedure if exists* if\_condition;  
*create procedure* if\_condition()  
*begin  
 select* e.first\_name,  
 e.last\_name,  
 c.salary,  
 *if*(c.salary >= 3500, 'Big salary', 'Small salary') *as* salary\_level  
 *from* employee *as* e, contract *as* c  
 *where* e.id = c.employee\_id;  
*end*;  
*call* if\_condition();  
  
*drop procedure if exists* while\_loop;  
*create procedure* while\_loop()  
*begin  
 declare* var *int default* 0;  
 *while* var < 5 *do  
 select* var;  
 *set* var = var + 1;  
 *end while*;  
*end*;  
*call* while\_loop();  
  
*drop procedure if exists* no\_params;  
*create procedure* no\_params()  
*begin  
 select* \*  
 *from client  
 where* LENGTH(payment\_card\_number) < 16;  
*end*;  
*call* no\_params();  
  
*drop procedure if exists* with\_params;  
*create procedure* with\_params(*in* sex *char*)  
*begin  
 select* \*  
 *from* employee *as* e  
 *where* e.sex = sex;  
*end*;  
*call* with\_params('F');  
  
*drop procedure if exists* in\_out\_procedure;  
*create procedure* in\_out\_procedure(*in value int*, *out* res *int*)  
*begin  
 select* sum(salary)  
 *from* contract  
 *where* salary < *value  
 into* res;  
  
 *select* res;  
*end*;  
*call* in\_out\_procedure(4000, @res);  
  
*drop procedure if exists* data\_update;  
*create procedure* data\_update()  
*begin  
 update* contract  
 *set* salary = salary + 500  
 *where* start\_date < *date*('2022-02-01');  
*end*;  
*call* data\_update();  
  
*drop procedure if exists* data\_select;  
*create procedure* data\_select()  
*begin  
 select month*, sum(amount\_of\_payment)  
 *from* monthly\_payments  
 *group by month*;  
*end*;  
*call* data\_select();  
  
*drop function if exists* scalar\_function;  
*create function* scalar\_function(client\_id *int*) *returns decimal  
 deterministic  
begin  
 declare* total *decimal*;  
 *select* sum(amount\_of\_payment)  
 *from* monthly\_payments *as* mp  
 *where* mp.client\_id = client\_id  
 *into* total;  
 *return* total;  
*end*;  
*select* scalar\_function(1);  
  
*drop procedure if exists* dynamic\_columns;  
*create procedure* dynamic\_columns(*in* flag *bool*)  
*begin  
 if* flag = *true then  
 select* concat(first\_name, ' ', last\_name), address, phone\_number  
 *from client*;  
 *else  
 select* \*  
 *from client*;  
 *end if*;  
*end*;  
*call* dynamic\_columns(*true*);  
  
*drop procedure if exists* curs;  
*create procedure* curs()  
*begin  
 declare name varchar*(50);  
 *declare* i *int default* 0;  
 *declare* cursor\_client *CURSOR FOR  
 select* concat(first\_name, ' ', last\_name)  
 *from client*;  
  
 *open* cursor\_client;  
  
 *while* i < (*select count*(\*) *from client*) *do  
 fetch* cursor\_client *into name*;  
 *select name*;  
 *set* i = i + 1;  
 *end while*;  
*end*;  
*call* curs();  
  
*drop trigger if exists* trigger\_on\_delete;  
*create trigger* trigger\_on\_delete *before delete on client for each row begin  
 if OLD*.id *in* (*select* client\_id *from* treaty) *then  
 signal sqlstate* '45000'  
 *set message\_text* = 'You cant delete the user with active retail outlet';  
 *end if*;  
*end*;  
*delete from client where* id = 2;  
  
*drop trigger if exists* trigger\_on\_update;  
*create trigger* trigger\_on\_update *before update on* contract *for each row  
begin  
 if* new.salary > 6000 *then  
 set* NEW.salary = 6000;  
 *end if*;  
*end*;  
*update* contract  
*set* salary = 7000  
*where* id = 1;  
  
*drop trigger if exists* trigger\_on\_insert;  
*create trigger* trigger\_on\_insert *before insert on* contract *for each row  
begin  
 if* new.salary < 1000 *then  
 set* NEW.salary = 1000;  
 *end if*;  
*end*;  
*insert into* contract (salary)  
*values* (500);

**Висновок:**

Протягом даної лабораторної роботи я вивчив правила побудови ідентифікаторів, правила визначення змінних та типів, визначив правила роботи з циклами та умовними конструкціями, вивчив синтаксис та семантику функцій та збережених процедур, способів їх ідентифікації, методів визначення та специфікації параметрів та повертаємих значень, виклик функцій та збережених процедур. Застосовував команди для створення, зміни та видалення як скалярних, так і табличних функцій, збережених процедур, вивчив призначення та типи курсорів, синтаксис та семантику команд мови SQL для створення курсорів, вибірки даних з курсорів, зміни даних із застосуванням курсорів, вивчив призначення та типи тригерів, умов їх активації, синтаксису та семантики для їх створення, модифікації, перейменування, програмування та видалення.