**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**.”

**Виконав(ла)**

(шифр, прізвище, ім'я, по батькові)

*ІП-15 Мєшков Андрій Ігорович*

**Перевірив**

(прізвище, ім'я, по батькові)

*Ліщук Олександр*

Київ 2023

**Лабораторна робота № 5**

**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.

**Теоретичні основи**

Викладені в лекційному матеріалі

**Постановка задачі лабораторної роботи №5**

При виконанні лабораторної роботи необхідно виконати наступні дії:

1) Збережені процедури:

* 1. запит для створення тимчасової таблиці через змінну типу TABLE;
  2. запит з використанням умовної конструкції IF;
  3. запит з використанням циклу WHILE;
  4. створення процедури без параметрів;
  5. створення процедури з вхідним параметром;
  6. створення процедури з вхідним параметром та RETURN;
  7. створення процедури оновлення даних в деякій таблиці БД;
  8. 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;

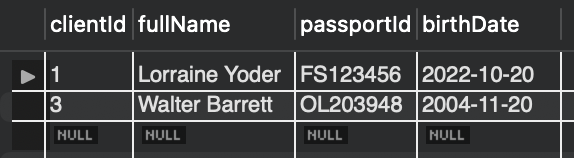
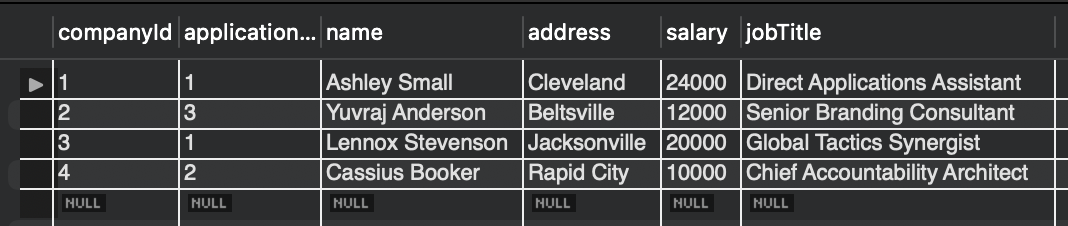
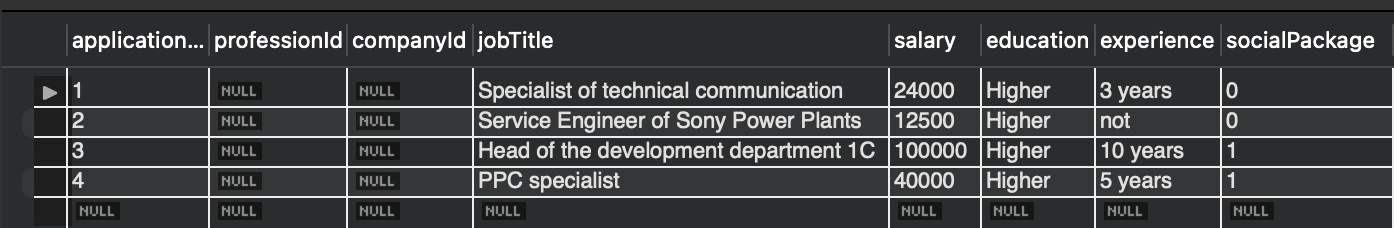
c. створити тригер, котрий буде спрацьовувати при додаванні даних.

Варіант 18

**Кадрова агенція**. Кадрова агенція займається підбором персоналу для підприємств і пошуком вакансій для людей. При здійсненні підбору персоналу підприємство звертається в кадрову агенцію і оформляє заявку на пошук працівника. У заявці вказується назва вакансії, професійні вимоги до претендента, пропонований рівень заробітної плати, наявність соціального пакету та інше. Працівники агенції розміщують заяву на сайті агенції, подають оголошення у відповідні друковані видання, вказуючи для контактів свої робочий телефон і e-mail. Претенденти на вакансію звертаються в кадрову агенцію, де проходять початкову співбесіду. Якщо працівник кадрової агенції вважає, що претендент задовольняє виставленим вимогам, то він реєструє його дані (ПІБ, контактний телефон або e-mail) в базі даних і повідомляє їх підприємству. При здійсненні підбору вакансій клієнт заповнює бланк, в якому вказує свої професійні можливості з переліку професійних вимог працедавців, паспортні дані, бажаний рівень заробітної плати. Працівник агенції виконує пошук вакансій по вказаних параметрах, при знаходженні вакансії він повідомляє клієнту контактні дані працедавця. Серед іншого адміністрація кадрової агенції може видавати статистику про середній рівень пропонованої заробітної плати по професіях, по конкретному підприємству і так далі.

**SQL-скрипти**

Будуть використані дані з таблиць Company, Application, Client:



1) Збережені процедури:

* 1. запит для створення тимчасової таблиці через змінну типу TABLE;

DELIMITER //

CREATE PROCEDURE create\_temp\_table()

BEGIN

DROP TEMPORARY TABLE IF EXISTS temp\_table;

CREATE TEMPORARY TABLE temp\_table(id INT, name VARCHAR(255) NOT NULL);

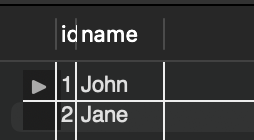
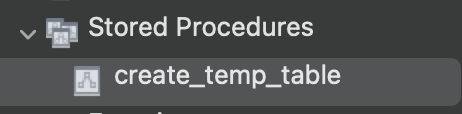
INSERT INTO temp\_table (id, name) VALUES (1, 'John'), (2, 'Jane');

SELECT \* FROM temp\_table;

DROP TEMPORARY TABLE temp\_table;

END//

DELIMITER ;



* 1. запит з використанням умовної конструкції IF;

DELIMITER $$

USE PersonalAgency;

DROP PROCEDURE IF EXISTS update\_product;

CREATE PROCEDURE update\_product(IN id INT, IN payment DECIMAL(10,2))

BEGIN

DECLARE old\_payment DECIMAL(10,2);

SELECT salary INTO old\_payment FROM Company WHERE companyId = id;

IF old\_payment < payment THEN

UPDATE Company SET salary = payment WHERE companyId = id;

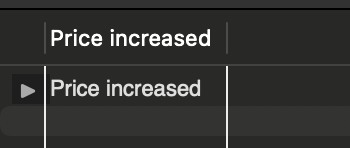
SELECT 'Price increased';

ELSE

SELECT 'Price not increased';

END IF;

END$$



* 1. запит з використанням циклу WHILE;

DELIMITER $$

USE PersonalAgency;

DROP PROCEDURE IF EXISTS sum\_numbers;

CREATE PROCEDURE sum\_numbers(IN max\_number INT)

BEGIN

-- DROP TEMPORARY TABLE IF EXISTS temp\_table;

-- CREATE TEMPORARY TABLE temp\_table(id INT, name VARCHAR(255) NOT NULL);

-- INSERT INTO temp\_table (id, name) VALUES

DECLARE current\_number INT DEFAULT 1;

DECLARE total INT DEFAULT 0;

WHILE current\_number <= max\_number DO

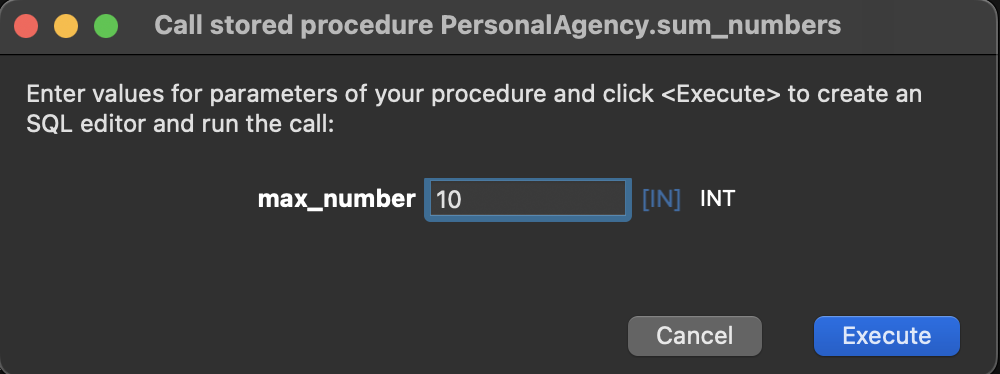
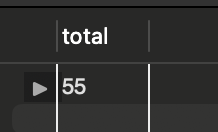
SET total = total + current\_number;

SET current\_number = current\_number + 1;

END WHILE;

SELECT total;

END$$



* 1. створення процедури без параметрів;

DELIMITER $$

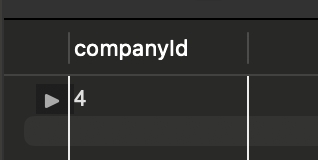
USE PersonalAgency;

CREATE PROCEDURE new\_count()

BEGIN

SELECT COUNT(\*) as companyId FROM Company;

END$$



* 1. створення процедури з вхідним параметром;

DELIMITER $$

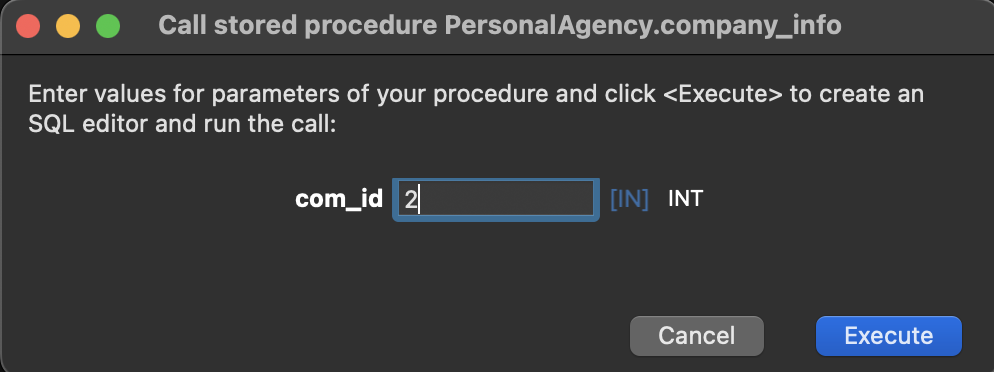
USE PersonalAgency;

CREATE PROCEDURE company\_info(IN com\_id INT)

BEGIN

SELECT \* FROM company WHERE companyId = com\_id;

END$$



* 1. створення процедури з вхідним параметром та RETURN;

-

* 1. створення процедури оновлення даних в деякій таблиці БД;

DELIMITER //

DROP PROCEDURE IF EXISTS update\_employee\_info //

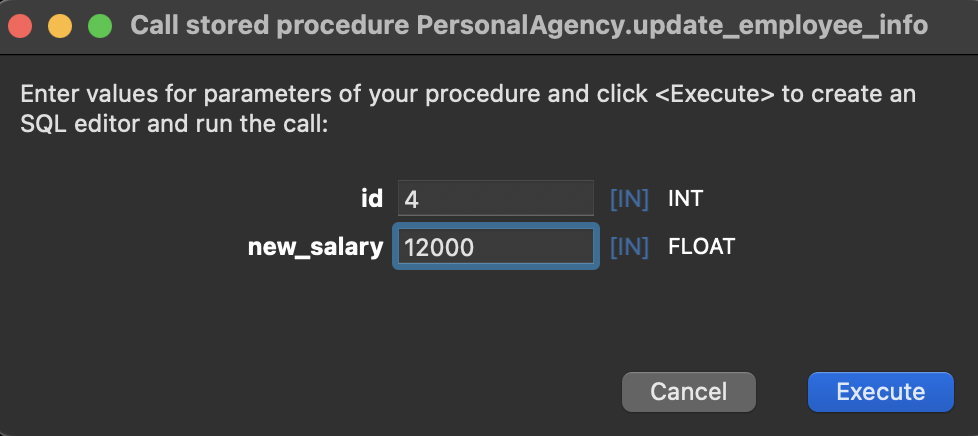
CREATE PROCEDURE update\_employee\_info(IN id INT, IN new\_salary FLOAT)

BEGIN

UPDATE Company SET salary = new\_salary WHERE companyId = id;

END; //

DELIMITER ;



* 1. створення процедури, в котрій робиться вибірка даних.

DELIMITER //

DROP PROCEDURE IF EXISTS a\_name //

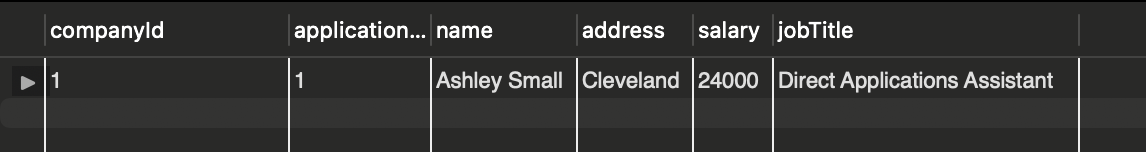
CREATE PROCEDURE a\_name()

BEGIN

SELECT \* FROM Company WHERE name like 'A%';

END //

DELIMITER ;



2) Функції:

a. створити функцію, котра повертає деяке скалярне значення;

DELIMITER //

SET GLOBAL log\_bin\_trust\_function\_creators = 1;

CREATE FUNCTION total\_n() RETURNS INT

BEGIN

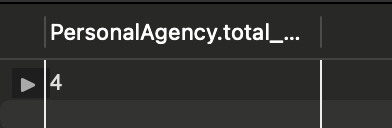
DECLARE n\_count INT;

SELECT COUNT(\*) INTO n\_count FROM Company;

RETURN n\_count;

END //

DELIMITER ;



3) Робота з курсорами:

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

DELIMITER //

CREATE PROCEDURE cur()

BEGIN

DECLARE done INT DEFAULT 0;

DECLARE c\_name VARCHAR(255);

DECLARE c\_address VARCHAR(255);

DECLARE c\_salary FLOAT;

DECLARE cur CURSOR FOR SELECT name, address, salary FROM Company;

DECLARE CONTINUE HANDLER FOR SQLSTATE '02000' SET done = 1;

OPEN cur;

DROP TEMPORARY TABLE IF EXISTS name\_salary;

CREATE TEMPORARY TABLE name\_salary(nameAddress VARCHAR(255), salary FLOAT);

WHILE done=0 DO

FETCH NEXT FROM cur INTO c\_name, c\_address, c\_salary;

INSERT INTO name\_salary VALUES(CONCAT(c\_name," - ", c\_address), c\_salary);

END WHILE;

SELECT \* FROM name\_salary;

DROP TEMPORARY TABLE name\_salary;

CLOSE cur;

END //



4) Робота з тригерами:

a. Створити тригер, котрий буде спрацьовувати при видаленні даних;

DELIMITER //

DROP TRIGGER IF EXISTS delete\_employee\_audit;

CREATE TRIGGER delete\_employee\_audit

AFTER DELETE ON Company

FOR EACH ROW

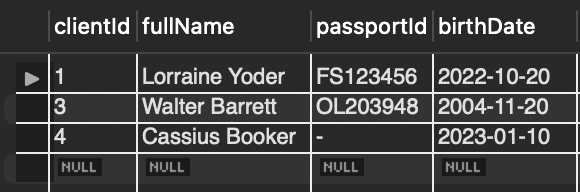
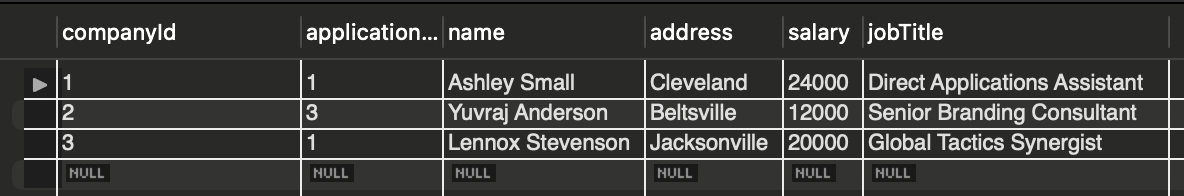
BEGIN

INSERT INTO Client (clientId, fullName, passportId, birthDate)

VALUES (OLD.companyId, OLD.name, "-", CURDATE());

END//

DELIMITER ;



b. створити тригер, котрий буде спрацьовувати при модифікації даних;

DELIMITER //

CREATE TRIGGER update\_audit

AFTER UPDATE ON Company

FOR EACH ROW

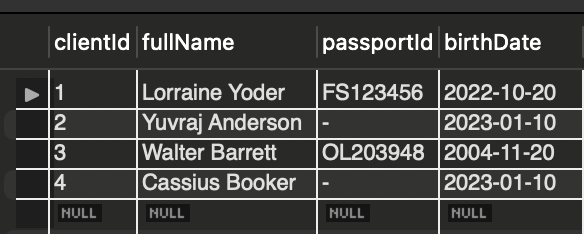
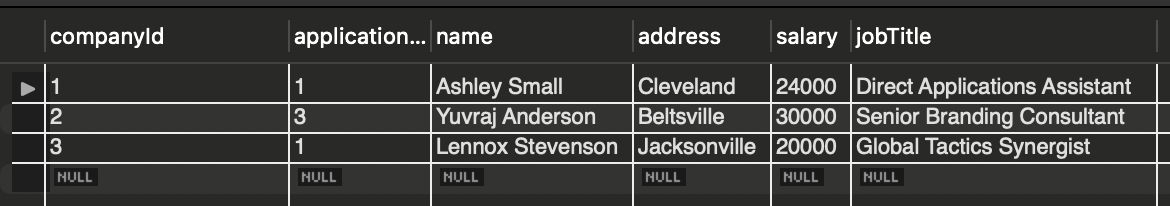
BEGIN

INSERT INTO Client (clientId, fullName, passportId, birthDate)

VALUES (NEW.companyId, NEW.name, "-", CURDATE());

END//

DELIMITER ;



c. створити тригер, котрий буде спрацьовувати при додаванні даних.

DELIMITER //

CREATE TRIGGER insert\_audit

AFTER INSERT ON Company

FOR EACH ROW

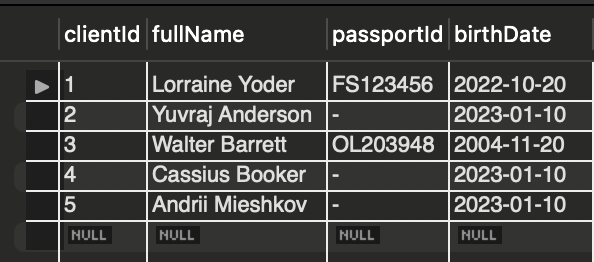
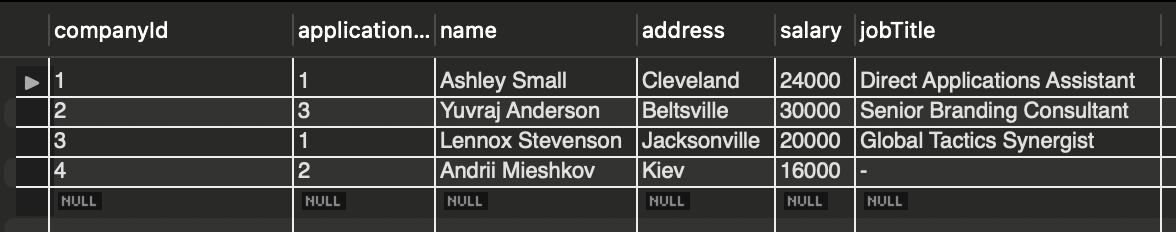
BEGIN

INSERT INTO Client (clientId, fullName, passportId, birthDate)

VALUES (NEW.companyId+1, NEW.name, "-", CURDATE());

END//

DELIMITER ;



**Висновок**: Під час виконання лабораторної работи я вивчив правила побудови ідентифікаторів, правила визначення змінних та типів. Визначив правила роботи з циклами та умовними конструкціями, роботу зі змінними типу Table. Вивчив синтаксис та семантику функцій та збережених процедур, способів їх ідентифікації, методів визначення та специфікації параметрів та повертаємих значень, виклик функцій та збережених процедур. Застосував команд 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.