

**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)**

*ІІІ-15 Мешков Андрій Ігорович*  
(шифр, прізвище, ім'я, по батькові)

**Перевірів**

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

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:

- 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.

## 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:

	application...	professionId	companyId	jobTitle	salary	education	experience	socialPackage
▶	1	NULL	NULL	Specialist of technical communication	24000	Higher	3 years	0
	2	NULL	NULL	Service Engineer of Sony Power Plants	12500	Higher	not	0
	3	NULL	NULL	Head of the development department 1C	100000	Higher	10 years	1
	4	NULL	NULL	PPC specialist	40000	Higher	5 years	1
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

	companyId	application...	name	address	salary	jobTitle
▶	1	1	Ashley Small	Cleveland	24000	Direct Applications Assistant
	2	3	Yuvraj Anderson	Beltsville	12000	Senior Branding Consultant
	3	1	Lennox Stevenson	Jacksonville	20000	Global Tactics Synergist
	4	2	Cassius Booker	Rapid City	10000	Chief Accountability Architect
	NULL	NULL	NULL	NULL	NULL	NULL

	clientId	fullName	passportId	birthDate
▶	1	Lorraine Yoder	FS123456	2022-10-20
	3	Walter Barrett	OL203948	2004-11-20
	NULL	NULL	NULL	NULL

1) Збережені процедури:

а. 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 ;

```

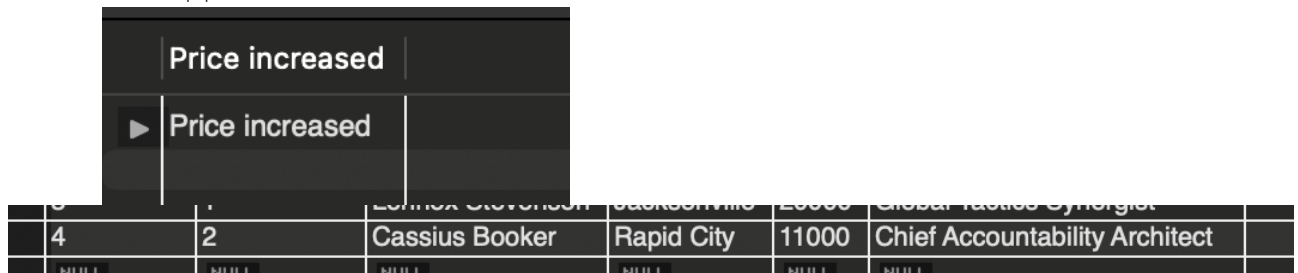
Stored Procedures		id name	
▼	create_temp_table	▶	1 John
			2 Jane

b. запит з використанням умовної конструкції IF;

```

DELIMITER $$
USE PersonalAgency;
DROP PROCEDURE IF EXISTS update_product;
CREATE PROCEDURE update_product(IN id INT, IN payment
DECIMAL(10,2))
BEGIN
    DECLARE old_payment DECIMAL(10,2);
    SELECT salary INTO old_payment FROM Company WHERE companyId =
id;
    IF old_payment < payment THEN
        UPDATE Company SET salary = payment WHERE companyId = id;
        SELECT 'Price increased';
    ELSE
        SELECT 'Price not increased';
    END IF;
END$$

```



The screenshot shows a database application window. At the top, a message box displays 'Price increased' with a play button. Below it, a table displays employee information. The table has columns for ID, Salary, Name, City, and Job Title. The data row shows an employee with ID 4, Salary 2, Name Cassius Booker, City Rapid City, and Job Title Chief Accountability Architect.

ID	Salary	Name	City	Job Title
4	2	Cassius Booker	Rapid City	Chief Accountability Architect

c. 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\$\$

	total	
▶	55	

Call stored procedure PersonalAgency.sum\_numbers

Enter values for parameters of your procedure and click <Execute> to create an SQL editor and run the call:

max\_number  [IN] INT

Cancel Execute

d. створення процедури без параметрів;

```
DELIMITER $$
USE PersonalAgency;
CREATE PROCEDURE new_count()
BEGIN
    SELECT COUNT(*) as companyId FROM Company;
END$$
```

	companyId	
▶	4	

e. створення процедури з вхідним параметром;

```
DELIMITER $$
USE PersonalAgency;
CREATE PROCEDURE company_info(IN com_id INT)
BEGIN
    SELECT * FROM company WHERE companyId = com_id;
END$$
```

**Call stored procedure PersonalAgency.company\_info**

Enter values for parameters of your procedure and click <Execute> to create an SQL editor and run the call:

com\_id  [IN] INT

Cancel Execute

	companyId	application...	name	address	salary	jobTitle
▶	2	3	Yuvraj Anderson	Beltsville	12000	Senior Branding Consultant

f. створення процедури з вхідним параметром та RETURN;

-

g. 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 ;

**Call stored procedure PersonalAgency.update\_employee\_info**

Enter values for parameters of your procedure and click <Execute> to create an SQL editor and run the call:

id  [IN] INT

new\_salary  [IN] FLOAT

Cancel Execute

	companyId	application...	name	address	salary	jobTitle
▶	1	1	Ashley Small	Cleveland	24000	Direct Applications Assistant
	2	3	Yuvraj Anderson	Beltsville	12000	Senior Branding Consultant
	3	1	Lennox Stevenson	Jacksonville	20000	Global Tactics Synergist
	4	2	Cassius Booker	Rapid City	12000	Chief Accountability Architect
	NULL	NULL	NULL	NULL	NULL	NULL

h. 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 ;
```

	companyId	application...	name	address	salary	jobTitle	
▶	1	1	Ashley Small	Cleveland	24000	Direct Applications Assistant	

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 ;
```

	PersonalAgency.total_...	
▶	4	

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 //

```

	nameAddress	salary	
▶	Ashley Small - Cleveland	24000	
	Yuvraj Anderson - Beltsville	12000	
	Lennox Stevenson - Jacksonville	20000	
	Cassius Booker - Rapid City	12000	
	Cassius Booker - Rapid City	12000	

#### 4) Робота з тригерами:

```

а. 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 ;

```

	companyId	application...	name	address	salary	jobTitle
▶	1	1	Ashley Small	Cleveland	24000	Direct Applications Assistant
	2	3	Yuvraj Anderson	Beltsville	12000	Senior Branding Consultant
	3	1	Lennox Stevenson	Jacksonville	20000	Global Tactics Synergist
	NULL	NULL	NULL	NULL	NULL	NULL

	clientId	fullName	passportId	birthDate
▶	1	Lorraine Yoder	FS123456	2022-10-20
	3	Walter Barrett	OL203948	2004-11-20
	4	Cassius Booker	-	2023-01-10
	NULL	NULL	NULL	NULL

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 ;

	companyId	application...	name	address	salary	jobTitle
▶	1	1	Ashley Small	Cleveland	24000	Direct Applications Assistant
	2	3	Yuvraj Anderson	Beltsville	30000	Senior Branding Consultant
	3	1	Lennox Stevenson	Jacksonville	20000	Global Tactics Synergist
	NULL	NULL	NULL	NULL	NULL	NULL

	clientId	fullName	passportId	birthDate
▶	1	Lorraine Yoder	FS123456	2022-10-20
	2	Yuvraj Anderson	-	2023-01-10
	3	Walter Barrett	OL203948	2004-11-20
	4	Cassius Booker	-	2023-01-10
	NULL	NULL	NULL	NULL

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 ;

	companyId	application...	name	address	salary	jobTitle
▶	1	1	Ashley Small	Cleveland	24000	Direct Applications Assistant
	2	3	Yuvraj Anderson	Beltsville	30000	Senior Branding Consultant
	3	1	Lennox Stevenson	Jacksonville	20000	Global Tactics Synergist
	4	2	Andrii Mieshkov	Kiev	16000	-
	NULL	NULL	NULL	NULL	NULL	NULL

	clientId	fullName	passportId	birthDate
▶	1	Lorraine Yoder	FS123456	2022-10-20
	2	Yuvraj Anderson	-	2023-01-10
	3	Walter Barrett	OL203948	2004-11-20
	4	Cassius Booker	-	2023-01-10
	5	Andrii Mieshkov	-	2023-01-10
	NULL	NULL	NULL	NULL

**Висновок:** Під час виконання лабораторної роботи 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.