

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 15

Виконала студентка ІП-12 Кушнір Ганна Вікторівна

Перевірила Марченко Олена Іванівна

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.

Теоретичні основи

Викладені в лекційному матеріалі

1. Постановка задачі лабораторної роботи № 5

При виконанні лабораторної роботи необхідно виконати наступні дії:

1) Збережені процедури:

- запит для створення тимчасової таблиці через змінну типу TABLE;
- запит з використанням умовної конструкції IF;
- запит з використанням циклу WHILE;
- створення процедури без параметрів;
- створення процедури з вхідним параметром;
- створення процедури з вхідним параметром та RETURN;
- створення процедури оновлення даних в деякій таблиці БД;
- створення процедури, в котрій робиться вибірка даних.

2) Функції:

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

3) Робота з курсорами:

- 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.

2. Завдання варіанту

Програмне забезпечення готелю.

Основна задача програмного забезпечення, котре проєктується - є відстеження фінансової сторони роботи готелю. Діяльність організована наступним чином: готель надає номери клієнтам на певний термін. Кожен номер характеризується місткістю, комфортністю (люкс, напівлюкс, звичайний, тощо) і ціною. Клієнтами є різні особи, по яким збирається певна інформація (прізвище, ім'я, по-батькові, номер та серія паспорту або іншого документу, що посвідчує особу, додаткова інформація). Задача номера клієнтові проводиться за наявності вільних місць в номерах, за параметрами, котрі вказав клієнт. При поселенні фіксується дата поселення, при виїзді – дата звільнення. номеру. Необхідно не лише зберігати інформацію за фактом здачі номера клієнтові, але і здійснювати бронювання номерів. Крім того, для постійних клієнтів, а також для певних категорій клієнтів передбачена система знижок. Знижки можуть підсумовуватися.

3. Виконання

3.1. Збережені процедури

-- а. запит для створення тимчасової таблиці через змінну типу TABLE --

```
CREATE OR ALTER PROCEDURE CreateTempTable
```

```
AS
```

```
BEGIN
```

```
CREATE TABLE ##ClientsWithCategories (
```

```
ID INT,
```

```
FirstName NVARCHAR(20),
```

```
LastName NVARCHAR(20),
```

```
Document NVARCHAR(12),
```

```
Category NVARCHAR(30)
```

```
)
```

```
INSERT INTO ##ClientsWithCategories (
```

```
ID, FirstName, LastName, Document, Category
```

```
)
```

```
SELECT
```

```
    Clients.ID, Clients.Name, Clients.Surname, Clients.Document,
```

```
Categories.Name
```

```
FROM MyDB.dbo.Clients, MyDB.dbo.Categories, MyDB.dbo.ClientCategory
```

```
WHERE Clients.ID = ClientCategory.ClientID
```

```
    AND ClientCategory.CategoryID = Categories.ID
```

```
SELECT *
```

```
FROM ##ClientsWithCategories
```

```
END
```

```
GO
```

```
EXECUTE CreateTempTable
```

```
GO
```

	ID	FirstName	LastName	Document	Category
1	2	Bob	Tate	002046183263	Disabled
2	3	Sheril	Blossom	003791936729	Orphan
3	6	Taras	Shevchenko	57913729	Orphan
4	8	Alisa	Stetsenko	001729372	Student
5	9	Tetiana	Pogoreltseva	002738171	Single mother
6	9	Tetiana	Pogoreltseva	002738171	Large family
7	16	Artem	Shostak	001729362	Student

-- б. запит з використанням умовної конструкції IF --

```
CREATE OR ALTER PROCEDURE HasHighSalary @EmployeeID INT
```

```
AS
```

```
BEGIN
```

```
IF (SELECT COUNT(*) FROM Employees WHERE Employees.ID = @EmployeeID)
```

```
= 0
```

```
    PRINT 'Employee with ID = ' + CONVERT(varchar(10), @EmployeeID)
```

```
+ ' does not exist in the database.'
```

```
ELSE
```

```
    IF (SELECT Positions.Salary
```

```
        FROM MyDB.dbo.Employees, MyDB.dbo.Positions
```

```

        WHERE Employees.ID = @EmployeeID AND
              Employees.PositionID = Positions.ID) >= 10000
        PRINT 'Employee with ID = ' + CONVERT(varchar(10),
@EmployeeID) + ' has high salary.'
    ELSE
        PRINT 'Employee with ID = ' + CONVERT(varchar(10),
@EmployeeID) + ' has low salary.'
END
GO

```

```

EXECUTE HasHighSalary 1
GO
EXECUTE HasHighSalary 5
GO
EXECUTE HasHighSalary 20
GO

```

Messages	
Employee with ID = 1	has high salary.
Employee with ID = 5	has low salary.
Employee with ID = 20	does not exist in the database.

```

-- с. запит з використанням циклу WHILE --
CREATE OR ALTER PROCEDURE RaiseWages @Coefficient REAL, @MaxSalary REAL
AS
BEGIN
    WHILE (SELECT MAX(Positions.Salary) FROM Positions) <= @MaxSalary
    BEGIN
        UPDATE MyDB.dbo.Positions
            SET Salary = Salary * @Coefficient
    END

    UPDATE Employees
    SET Salary = Positions.Salary
    FROM Positions
    WHERE Employees.PositionID = Positions.ID;

    SELECT *
    FROM Positions
END
GO

```

```

EXECUTE RaiseWages 1.05, 20000
GO

```

	ID	Name	Salary
1	1	Administrator	20101,4347
2	2	Manager	16081,1477
3	3	Maid	6700,4783
4	4	Cook	9380,6696
5	5	Waiter	6700,4783
6	6	Security guard	9380,6696
7	7	Sommelier	7370,526

```
-- d. створення процедури без параметрів --
CREATE OR ALTER PROCEDURE EmployeesYearSalary
AS
BEGIN
    SELECT Employees.ID, Employees.Surname, Employees.Name,
Employees.Salary * 12 AS YearSalary
    FROM MyDB.dbo.Employees
    ORDER BY Surname, Name
END
GO
```

```
EXECUTE EmployeesYearSalary
GO
```

	ID	Surname	Name	YearSalary
1	1005	Freyman	Nick	112568,0352
2	1	Grande	Ariana	241217,2164
3	3	Karpiuk	Oles	112568,0352
4	1004	Klymenko	Nadiya	112568,0352
5	5	Kropiv	Victor	88446,312
6	4	Larin	Mykola	112568,0352
7	1006	Leibnits	Martin	192973,7724
8	1003	Mykulynych	Ivan	80405,7396
9	7	Rahiv	Karina	241217,2164
10	2	Rishko	Tetiana	192973,7724
11	6	Rurik	Stanislav	241217,2164
12	8	Shkred	Iryna	112568,0352
13	1008	Stepanenko	Stephania	88446,312
14	1002	West	Hanna	192973,7724
15	1007	Wirna	Ivanna	112568,0352

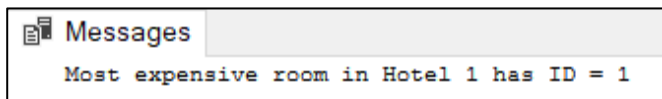
```
-- e. створення процедури з вхідним параметром --
CREATE OR ALTER PROCEDURE HotelsAvailableRooms @HotelID INT
AS
BEGIN
    SELECT Hotels.Name AS Hotel, Rooms.ID AS RoomID, Rooms.Price,
Rooms.Capacity, Comfort.Name AS Comfort
    FROM Hotels, Rooms, Comfort
    WHERE Rooms.HotelID = @HotelID AND Hotels.ID = @HotelID
    AND Rooms.ComfortID = Comfort.ID AND Rooms.IsAvailable = 1
END
GO
```

```
EXECUTE HotelsAvailableRooms 1
GO
```

	Hotel	RoomID	Price	Capacity	Comfort
1	Kyiv Hotel	1	3000,00	3	Presidential
2	Kyiv Hotel	2	2000,00	2	Presidential
3	Kyiv Hotel	3	2000,00	3	Suite
4	Kyiv Hotel	4	1500,00	2	Suite
5	Kyiv Hotel	6	700,00	2	Junior suite
6	Kyiv Hotel	7	500,00	3	Standard
7	Kyiv Hotel	8	400,00	2	Standard
8	Kyiv Hotel	9	300,00	2	Economy
9	Kyiv Hotel	10	200,00	1	Economy

```
-- f. створення процедури з вхідним параметром та RETURN --
CREATE OR ALTER PROCEDURE MostExpensiveRoom @HotelID INT, @RoomID INT OUTPUT
AS
BEGIN
    SELECT @RoomID = Rooms.ID
    FROM Rooms
    WHERE Rooms.HotelID = @HotelID AND
           Rooms.Price = (SELECT MAX(Price) FROM Rooms WHERE HotelID =
@HotelID);
    RETURN;
END
GO
```

```
DECLARE @MostExpensiveRoomID INT;
EXECUTE MostExpensiveRoom @HotelID = 1, @RoomID = @MostExpensiveRoomID
OUTPUT;
PRINT 'Most expensive room in Hotel 1 has ID = ' + CONVERT(varchar(10),
@MostExpensiveRoomID);
GO
```



```
-- g. створення процедури оновлення даних в деякій таблиці БД --
CREATE OR ALTER PROCEDURE ChangeRoomPrices @RaiseOn INT
AS
BEGIN
    UPDATE MyDB.dbo.Rooms
        SET Price = Price + @RaiseOn
    UPDATE MyDB.dbo.Rooms
        SET BookingPrice = BookingPrice + @RaiseOn / 2
    SELECT *
    FROM MyDB.dbo.Rooms
END
GO
```

```
EXECUTE ChangeRoomPrices 100
GO
```

	ID	IsAvailable	Price	BookingPrice	Capacity	ComfortID	HotelID
1	1	1	3100,00	1550,00	3	1	1
2	2	1	2100,00	1050,00	2	1	1
3	3	1	2100,00	750,00	3	2	1
4	4	1	1600,00	550,00	2	2	1
5	5	0	1100,00	300,00	3	3	1
6	6	1	800,00	225,00	2	3	1
7	7	1	600,00	150,00	3	4	1
8	8	1	500,00	130,00	2	4	1
9	9	1	400,00	100,00	2	5	1
10	10	1	300,00	85,00	1	5	1
11	11	1	1100,00	350,00	2	1	2
12	12	1	800,00	250,00	2	2	2
13	13	1	600,00	150,00	2	4	2
14	14	1	350,00	100,00	2	5	2
15	15	1	1100,00	400,00	2	2	3
16	16	1	600,00	150,00	3	4	3


```
-- h. створення процедури, в котрій робиться вибірка даних --
CREATE OR ALTER PROCEDURE MonthReportForHotel @HotelID INT, @Month
VARCHAR(7)
AS
BEGIN
    SELECT Payroll.Date, Employees.HotelID, Employees.ID AS EmployeeID,
    Employees.Surname + ' ' + Employees.Name AS FullName, Employees.Salary
    FROM Employees, SalaryPayment, Payroll
    WHERE Date LIKE (@Month + '%') AND
        Payroll.ID = SalaryPayment.PayrollID AND
        SalaryPayment.EmployeeID = Employees.ID AND
        Employees.HotelID = @HotelID
    ORDER BY FullName
END
GO

EXECUTE MonthReportForHotel 1, '2022-09'
GO
```

	Date	HotelID	EmployeeID	FullName	Salary
1	2022-09-30	1	1	Grande Ariana	20101,4347
2	2022-09-30	1	3	Karpiuk Oles	9380,6696
3	2022-09-30	1	5	Kropiv Victor	7370,526
4	2022-09-30	1	4	Larin Mykola	9380,6696

3.2. Функції

```
-- а. створити функцію, котра повертає деяке скалярне значення --
CREATE OR ALTER FUNCTION GetInfoAboutEmployee(@EmployeeID INT)
RETURNS VARCHAR(125)
AS
BEGIN
    DECLARE @Result VARCHAR(125);
    IF(SELECT COUNT(*) FROM Employees WHERE Employees.ID = @EmployeeID) = 0
        SET @Result = 'Employee with ID = ' + CONVERT(varchar(10),
@EmployeeID) + ' does not exist in the database.'
    ELSE
        SET @Result = (SELECT Employees.Name + ' ' + Employees.Surname +
' works as ' + LOWER(Positions.Name) + ' at ' + Hotels.Name + '.'
FROM MyDB.dbo.Employees
JOIN MyDB.dbo.Positions ON Employees.PositionID = Positions.ID
JOIN MyDB.dbo.Hotels ON Employees.HotelID = Hotels.ID
WHERE Employees.ID = @EmployeeID);
    RETURN (@Result);
END
GO

SELECT MyDB.dbo.GetInfoAboutEmployee(7) AS 'Employee Info'
GO
```

	Employee Info
1	Karina Rahiv works as administrator at Kharkiv Hotel.

```

-- б. створити функцію, котра повертає таблицю з динамічним набором
стовпців --
-- На жаль, MS SQL SERVER не дозволяє повертати з функцій таблиці з
динамічними наборами стовпців, це можливо лише з використанням процедур.
-- Наведена функція, яка повертає таблицю з динамічним набором рядків.
CREATE OR ALTER FUNCTION GetEmployeesAndClients (@EmployeesCount INT,
@ClientsCount INT)
RETURNS TABLE
AS
RETURN
(
    SELECT TOP (@EmployeesCount) Employees.Surname, Employees.Name,
Employees.Document
    FROM Employees
    UNION
    SELECT TOP (@ClientsCount) Clients.Surname, Clients.Name,
Clients.Document
    FROM Clients
);
GO

SELECT *
FROM GetEmployeesAndClients(5, 5)
GO

```

	Surname	Name	Document
1	Blossom	Sheril	003791936729
2	Diesel	Vin	003527193001
3	Grande	Ariana	001927381
4	Karpiuk	Oles	001728192
5	Kropiv	Victor	002818283
6	Larin	Mykola	001927628
7	Rishko	Tetiana	001825728
8	Tate	Bob	002046183263
9	Tate	Tabita	001956782426
10	Ukrainka	Lesya	CX271835

```

-- с. створити функцію, котра повертає таблицю заданої структури --
CREATE OR ALTER FUNCTION ProfitFromTheRooms ()
RETURNS @ResultTable TABLE (
    RoomID INT NOT NULL,
    Profit MONEY NOT NULL
)
AS
BEGIN
    INSERT @ResultTable
    SELECT Rooms.ID, ISNULL((SELECT SUM(Orders.Bill) FROM Orders WHERE
Orders.RoomID = Rooms.ID), 0)
    FROM Rooms
    RETURN;
END
GO

```

```

SELECT *
FROM ProfitFromTheRooms()
GO

```

	RoomID	Profit
1	1	0,00
2	2	5700,00
3	3	4000,00
4	4	2400,00
5	5	11050,00
6	6	1400,00
7	7	3400,00
8	8	1600,00
9	9	480,00
10	10	200,00
11	11	0,00
12	12	4200,00
13	13	0,00
14	14	0,00
15	15	0,00
16	16	0,00

3.3. Курсори

```

-- а. створити курсор --
DECLARE my_cursor CURSOR SCROLL
    FOR SELECT * FROM Orders
    FOR UPDATE

-- б. відкрити курсор --
OPEN my_cursor

-- с. вибірка даних, робота з курсорами
FETCH FIRST FROM my_cursor
FETCH RELATIVE 2 FROM my_cursor
FETCH NEXT FROM my_cursor
FETCH ABSOLUTE 5 FROM my_cursor
FETCH LAST FROM my_cursor
FETCH RELATIVE -3 FROM my_cursor

```

	ID	OrderDate	RoomID	ClientID	CheckInDate	CheckOutDate	Bill	BookingPrice
1	1	2022-09-05	2	1	2022-09-05 00:00:00.000	2022-09-07 00:00:00.000	4000,00	NULL
	ID	OrderDate	RoomID	ClientID	CheckInDate	CheckOutDate	Bill	BookingPrice
1	3	2022-09-12	7	8	2022-09-12 00:00:00.000	2022-09-16 00:00:00.000	1800,00	NULL
	ID	OrderDate	RoomID	ClientID	CheckInDate	CheckOutDate	Bill	BookingPrice
1	1003	2022-11-12	7	2	2022-11-12 00:00:00.000	2022-11-16 00:00:00.000	1600,00	NULL
	ID	OrderDate	RoomID	ClientID	CheckInDate	CheckOutDate	Bill	BookingPrice
1	1004	2022-11-13	5	3	2022-11-13 00:00:00.000	2022-11-18 00:00:00.000	4250,00	NULL
	ID	OrderDate	RoomID	ClientID	CheckInDate	CheckOutDate	Bill	BookingPrice
1	1012	2022-11-17	12	11	2022-11-19 00:00:00.000	2022-11-25 00:00:00.000	4200,00	1200,00
	ID	OrderDate	RoomID	ClientID	CheckInDate	CheckOutDate	Bill	BookingPrice
1	1009	2022-11-11	8	5	2022-11-15 00:00:00.000	2022-11-19 00:00:00.000	1600,00	320,00

SELECT * FROM Orders

	ID	OrderDate	RoomID	ClientID	CheckInDate	CheckOutDate	Bill	BookingPrice
1	1	2022-09-05	2	1	2022-09-05 00:00:00.000	2022-09-07 00:00:00.000	4000,00	NULL
2	2	2022-09-07	5	3	2022-09-07 00:00:00.000	2022-09-15 00:00:00.000	6800,00	NULL
3	3	2022-09-12	7	8	2022-09-12 00:00:00.000	2022-09-16 00:00:00.000	1800,00	NULL
4	1003	2022-11-12	7	2	2022-11-12 00:00:00.000	2022-11-16 00:00:00.000	1600,00	NULL
5	1004	2022-11-13	5	3	2022-11-13 00:00:00.000	2022-11-18 00:00:00.000	4250,00	NULL
6	1005	2022-11-15	2	6	2022-11-15 00:00:00.000	2022-11-16 00:00:00.000	1700,00	NULL
7	1006	2022-11-16	3	1	2022-11-17 00:00:00.000	2022-11-19 00:00:00.000	4000,00	1400,00
8	1007	2022-11-16	4	2	2022-11-17 00:00:00.000	2022-11-19 00:00:00.000	2400,00	1000,00
9	1008	2022-11-14	6	4	2022-11-16 00:00:00.000	2022-11-18 00:00:00.000	1400,00	350,00
10	1009	2022-11-11	8	5	2022-11-15 00:00:00.000	2022-11-19 00:00:00.000	1600,00	320,00
11	1010	2022-11-13	9	9	2022-11-13 00:00:00.000	2022-11-15 00:00:00.000	480,00	NULL
12	1011	2022-11-14	10	7	2022-11-14 00:00:00.000	2022-11-15 00:00:00.000	200,00	NULL
13	1012	2022-11-17	12	11	2022-11-19 00:00:00.000	2022-11-25 00:00:00.000	4200,00	1200,00

UPDATE Orders

SET ClientID = 14

WHERE CURRENT OF my_cursor

FETCH LAST FROM my_cursor

	ID	OrderDate	RoomID	ClientID	CheckInDate	CheckOutDate	Bill	BookingPrice
1	1012	2022-11-17	12	11	2022-11-19 00:00:00.000	2022-11-25 00:00:00.000	4200,00	1200,00

DELETE

FROM Orders

WHERE CURRENT OF my_cursor

SELECT * FROM Orders

	ID	OrderDate	RoomID	ClientID	CheckInDate	CheckOutDate	Bill	BookingPrice
1	1	2022-09-05	2	1	2022-09-05 00:00:00.000	2022-09-07 00:00:00.000	4000,00	NULL
2	2	2022-09-07	5	3	2022-09-07 00:00:00.000	2022-09-15 00:00:00.000	6800,00	NULL
3	3	2022-09-12	7	8	2022-09-12 00:00:00.000	2022-09-16 00:00:00.000	1800,00	NULL
4	1003	2022-11-12	7	2	2022-11-12 00:00:00.000	2022-11-16 00:00:00.000	1600,00	NULL
5	1004	2022-11-13	5	3	2022-11-13 00:00:00.000	2022-11-18 00:00:00.000	4250,00	NULL
6	1005	2022-11-15	2	6	2022-11-15 00:00:00.000	2022-11-16 00:00:00.000	1700,00	NULL
7	1006	2022-11-16	3	1	2022-11-17 00:00:00.000	2022-11-19 00:00:00.000	4000,00	1400,00
8	1007	2022-11-16	4	2	2022-11-17 00:00:00.000	2022-11-19 00:00:00.000	2400,00	1000,00
9	1008	2022-11-14	6	4	2022-11-16 00:00:00.000	2022-11-18 00:00:00.000	1400,00	350,00
10	1009	2022-11-11	8	14	2022-11-15 00:00:00.000	2022-11-19 00:00:00.000	1600,00	320,00
11	1010	2022-11-13	9	9	2022-11-13 00:00:00.000	2022-11-15 00:00:00.000	480,00	NULL
12	1011	2022-11-14	10	7	2022-11-14 00:00:00.000	2022-11-15 00:00:00.000	200,00	NULL

CLOSE my_cursor

DEALLOCATE my_cursor

3.4. Тригери

-- а. створити тригер, котрий буде спрацьовувати при видаленні даних --

CREATE OR ALTER TRIGGER on_delete

ON Clients

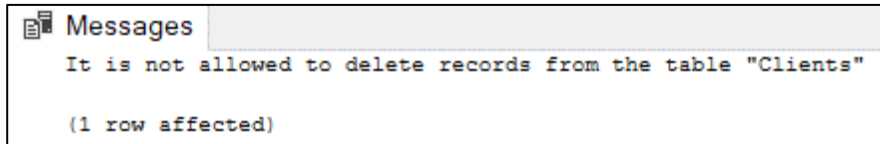
INSTEAD OF DELETE

AS

PRINT 'It is not allowed to delete records from the table "Clients"'

GO

```
DELETE FROM Clients
WHERE ID = 5
GO
```



```
-- b. створити тригер, котрий буде спрацьовувати при модифікації даних --
CREATE OR ALTER TRIGGER on_update
ON Rooms
AFTER UPDATE
AS
    UPDATE Hotels
    SET Rooms = (SELECT COUNT(*) FROM Rooms WHERE Rooms.HotelID =
Hotels.ID)
    UPDATE Hotels
    SET AvailableRooms = (SELECT COUNT(*) FROM Rooms WHERE Rooms.HotelID
= Hotels.ID AND Rooms.IsAvailable = 1)
GO
```

```
UPDATE Rooms
SET IsAvailable = 0
WHERE ID BETWEEN 5 AND 8 OR ID > 14
GO
```

```
SELECT * FROM Rooms
GO
SELECT * FROM Hotels
GO
```

	ID	IsAvailable	Price	BookingPrice	Capacity	ComfortID	HotelID
1	1	1	3100,00	1550,00	3	1	1
2	2	1	2100,00	1050,00	2	1	1
3	3	1	2100,00	750,00	3	2	1
4	4	1	1600,00	550,00	2	2	1
5	5	0	1100,00	300,00	3	3	1
6	6	0	800,00	225,00	2	3	1
7	7	0	600,00	150,00	3	4	1
8	8	0	500,00	130,00	2	4	1
9	9	1	400,00	100,00	2	5	1
10	10	1	300,00	85,00	1	5	1
11	11	1	1100,00	350,00	2	1	2
12	12	1	800,00	250,00	2	2	2
13	13	1	600,00	150,00	2	4	2
14	14	1	350,00	100,00	2	5	2
15	15	0	1100,00	400,00	2	2	3
16	16	0	600,00	150,00	3	4	3

	ID	Name	Director	Rooms	AvailableRooms	Location
1	1	Kyiv Hotel	Ribalka Mykola Mykolayowych	10	6	Kyiv
2	2	Odessa Hotel	Rishko Oleksiy Victorowych	4	4	Odessa
3	3	Kharkiv Hotel	Kozak Serhiy Volodymyrowych	2	0	Kharkiv

```

-- с. створити тригер, котрий буде спрацьовувати при додаванні даних --
CREATE OR ALTER TRIGGER on_insert
ON Employees
AFTER INSERT
AS
    UPDATE Employees
    SET Salary = (SELECT Positions.Salary FROM Positions WHERE
Positions.ID = PositionID)
GO

INSERT INTO Employees (
    Surname, Name, Document, PositionID, HotelID
)
VALUES
    ('Johnsoniuk', 'Borys', '001826371', 2, 1)
;
GO

SELECT * FROM Employees
GO

```

	ID	Surname	Name	Patronymic	Document	PositionID	Salary	HotelID
1	1	Grande	Ariana	NULL	001927381	1	20101,4347	1
2	2	Rishko	Tetiana	Mykolaivna	001825728	2	16081,1477	2
3	3	Karpiuk	Oles	NULL	001728192	4	9380,6696	1
4	4	Larin	Mykola	Valeriyovych	001927628	6	9380,6696	1
5	5	Kropiv	Victor	Victorovych	002818283	7	7370,526	1
6	6	Rurik	Stanislav	NULL	002828162	1	20101,4347	2
7	7	Rahiv	Karina	Andriivna	001562738	1	20101,4347	3
8	8	Shkred	Iryna	Pavlivna	CB728193	6	9380,6696	2
9	1002	West	Hanna	NULL	00172829182	2	16081,1477	2
10	1003	Mykulynych	Ivan	Ivanovych	CB627183	3	6700,4783	2
11	1004	Klymenko	Nadiya	Stepanivna	EB637183	4	9380,6696	2
12	1005	Freyman	Nick	NULL	01283947281	6	9380,6696	2
13	1006	Leibnits	Martin	NULL	26173829193	2	16081,1477	3
14	1007	Wirna	Ivanna	Artemivna	001828361	4	9380,6696	3
15	1008	Stepanenko	Stephania	Stepanivna	EE617293	7	7370,526	3
16	1009	Johnsoniuk	Borys	NULL	001826371	2	16081,1477	1

4. Висновок

Під час виконання лабораторної роботи з теми «Основи програмування з використанням мови SQL. Збережені процедури. Курсори. Створення, програмування та керування тригерами» було отримано навички побудови ідентифікаторів, визначення змінних та типів, роботи з циклами та умовними конструкціями, а також навички роботи зі змінними типу Table. Було вивчено синтаксис та семантику функцій та збережених процедур, способів їх ідентифікації, методів визначення та специфікації параметрів та повертаємих значень, було освоєно способи виклику функцій та збережених процедур.

У ході роботи було застосовано команди для створення, зміни та видалення як скалярних, так і табличних функцій та збережених процедур. Було вивчено призначення та типи курсорів, синтаксис та семантику команд мови SQL для створення курсорів, вибірки даних з курсорів, зміни даних із застосуванням курсорів. Також було отримано знання та навички для роботи з тригерами, вивчено умови їх активації, синтаксис та семантику для їх створення, модифікації, перейменування, програмування та видалення.