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:
2. запит для створення тимчасової таблиці через змінну типу TABLE;
3. запит з використанням умовної конструкції IF;
4. запит з використанням циклу WHILE;
5. створення процедури без параметрів;
6. створення процедури з вхідним параметром;
7. створення процедури з вхідним параметром та RETURN;
8. створення процедури оновлення даних в деякій таблиці БД;
9. створення процедури, в котрій робиться вибірка даних.
10. Функції:
11. створити функцію, котра повертає деяке скалярне значення;
12. створити функцію, котра повертає таблицю з динамічним набором стовпців;
13. створити функцію, котра повертає таблицю заданої структури.
14. Робота з курсорами:
15. створити курсор;
16. відкрити курсор;
17. вибірка даних, робота з курсорами.
18. Робота з тригерами:
19. створити тригер, котрий буде спрацьовувати при видаленні даних;
20. створити тригер, котрий буде спрацьовувати при модифікації даних;
21. створити тригер, котрий буде спрацьовувати при додаванні даних.
22. **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). Здача 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.

1. **Виконання**
2. Збережені процедури

-- a. 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



-- b. 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

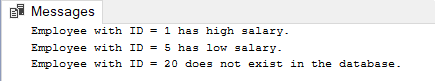
GO

EXECUTE HasHighSalary 5

GO

EXECUTE HasHighSalary 20

GO



-- c. 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 \*

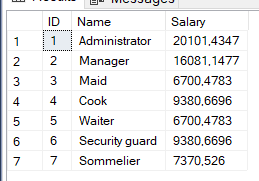
FROM Positions

END

GO

EXECUTE RaiseWages 1.05, 20000

GO



-- 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

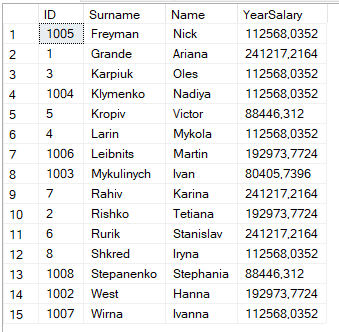
ORDER BY Surname, Name

END

GO

EXECUTE EmployeesYearSalary

GO



-- 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

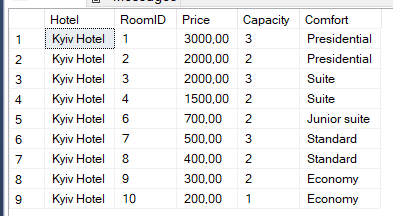
AND Rooms.ComfortID = Comfort.ID AND Rooms.IsAvailable = 1

END

GO

EXECUTE HotelsAvailableRooms 1

GO



-- 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

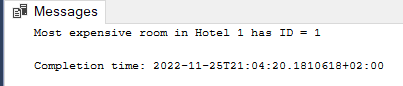
GO

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 \*

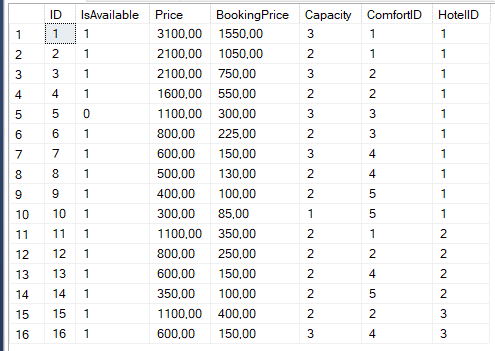
FROM MyDB.dbo.Rooms

END

GO

EXECUTE ChangeRoomPrices 100

GO



-- 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

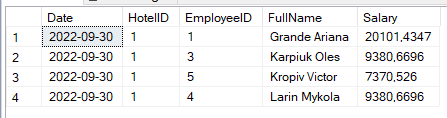
ORDER BY FullName

END

GO

EXECUTE MonthReportForHotel 1, '2022-09'

GO



1. Функції

-- a. 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



-- b. 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



-- c. 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;

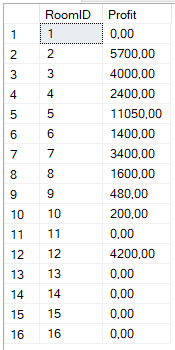
END

GO

SELECT \*

FROM ProfitFromTheRooms()

GO



1. Курсори

-- a. створити курсор --

DECLARE my\_cursor CURSOR SCROLL

FOR SELECT \* FROM Orders

FOR UPDATE

-- b. відкрити курсор --

OPEN my\_cursor

-- c. вибірка даних, робота з курсорами

FETCH FIRST FROM my\_cursor

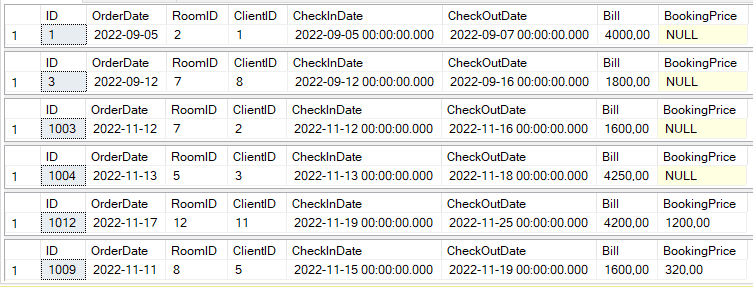
FETCH RELATIVE 2 FROM my\_cursor

FETCH NEXT FROM my\_cursor

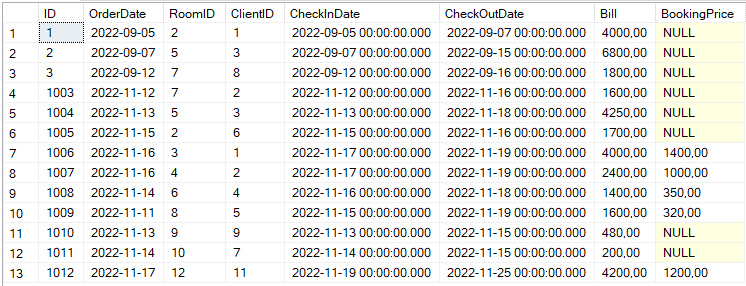
FETCH ABSOLUTE 5 FROM my\_cursor

FETCH LAST FROM my\_cursor

FETCH RELATIVE -3 FROM my\_cursor



SELECT \* FROM Orders



UPDATE Orders

SET ClientID = 14

WHERE CURRENT OF my\_cursor

FETCH LAST FROM my\_cursor

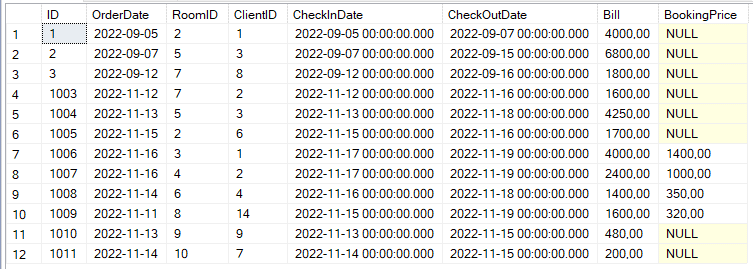


DELETE

FROM Orders

WHERE CURRENT OF my\_cursor

SELECT \* FROM Orders



CLOSE my\_cursor

DEALLOCATE my\_cursor

1. Тригери

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

CREATE OR ALTER TRIGGER on\_delete

ON Clients

INSTEAD OF DELETE

AS

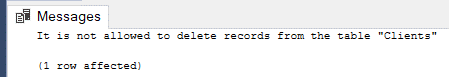
PRINT 'It is not allowed to delete records from the table "Clients"'

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

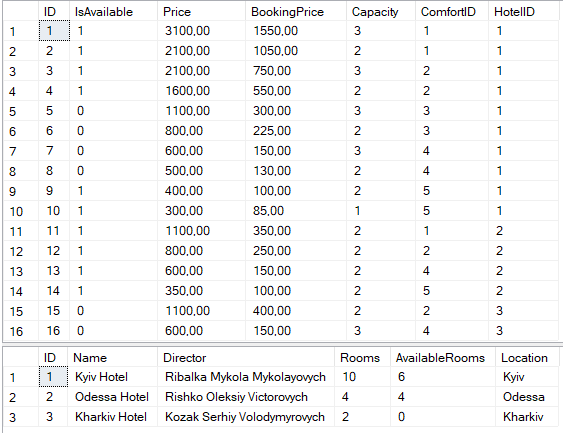
GO

SELECT \* FROM Rooms

GO

SELECT \* FROM Hotels

GO



-- c. 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

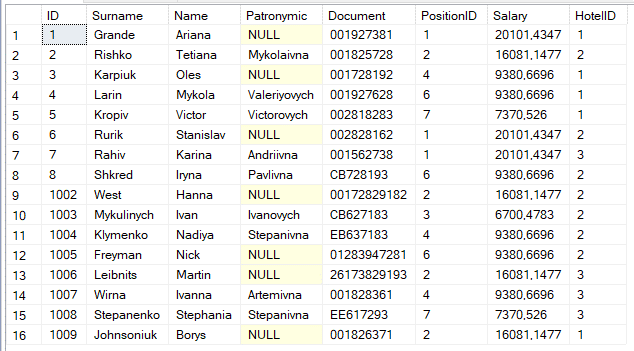
('Johnsoniuk', 'Borys', '001826371', 2, 1)

;

GO

SELECT \* FROM Employees

GO



1. **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.