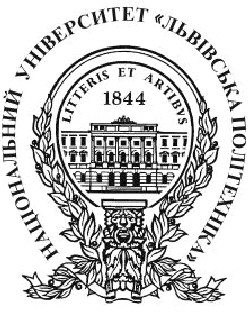
Міністерство освіти і науки України НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»



Кафедра ЕОМ

Звіт

З лабораторної роботи №3, варіант 3 З дисципліни «Організація баз даних»

На тему: «Об’єднання таблиць за допомогою операторів JOIN (LEFT, RIGHT, INNER), використання індексів»

Виконав: ст. гр. КІ-205

Говдиш Д.А

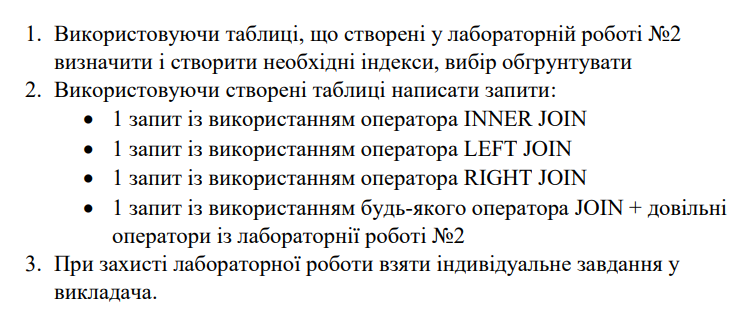
Прийняв: викладач

Трач І.Б.

Львів – 2023

**Мета:** отримати навики використання операторів JOIN та індексів

**Завдання:**

****

**База даних турагенства.**

**Код**

use tour\_agency;

CREATE TABLE Customers (

CustomerID INT PRIMARY KEY,

Name VARCHAR(50),

Email VARCHAR(100),

Phone VARCHAR(20)

);

CREATE TABLE Bookings (

BookingID INT PRIMARY KEY,

Customer\_booking\_ID INT,

TourID INT,

BookingDate DATE,

CustomerID int,

Foreign key (CustomerID) references Customers(CustomerID)

);

CREATE TABLE TourDates (

TourDateID INT PRIMARY KEY,

Tour\_dates\_ID INT,

Date DATE,

AvailableSlots INT

-- Foreign key (Tour\_dates\_ID) references TourGuides (Tour\_guides\_ID)

);

CREATE TABLE TourGuides (

GuideID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Email VARCHAR(100),

Tour\_guides\_ID INT

-- Foreign key (Tour\_guides\_ID) references TourDates (Tour\_dates\_ID)

);

CREATE TABLE BookingTourDates (

BookingID INT,

TourDateID INT,

PRIMARY KEY (BookingID, TourDateID),

FOREIGN KEY (BookingID) REFERENCES Bookings(BookingID),

FOREIGN KEY (TourDateID) REFERENCES TourDates(TourDateID)

);

INSERT INTO Customers (CustomerID, Name, Email, Phone)

VALUES

(1, 'John Smith', 'john.smith@example.com', '0961234567'),

(2, 'Jane Doe', 'jane.doe@example.com', '0962345678'),

(3, 'Bob Johnson', 'bob.johnson@example.com', '0963456789'),

(4, 'Alice Lee', 'alice.lee@example.com', '0964567890'),

(5, 'Tom Wilson', 'tom.wilson@example.com', '0965678901'),

(6, 'Sarah Brown', 'sarah.brown@example.com', '0966789012'),

(7, 'David Davis', 'david.davis@example.com', '0967890123'),

(8, 'Emily Smith', 'emily.smith@example.com', '0968901234'),

(9, 'Oliver Brown', 'oliver.brown@example.com', '0969012345'),

(10, 'Grace Taylor', 'grace.taylor@example.com', '0960123456');

INSERT INTO Bookings (BookingID, BookingDate, TourID)

VALUES

(1, '2022-05-15', 123),

(2, '2022-06-20', 456),

(3, '2022-07-12', 789),

(4, '2022-08-05', 123),

(5, '2022-09-01', 456),

(6, '2022-10-10', 789);

INSERT INTO TourGuides (GuideID, FirstName, LastName, Email)

VALUES

(1, 'John', 'Smith', 'john.smith@example.com'),

(2, 'Jane', 'Doe', 'jane.doe@example.com'),

(3, 'Bob', 'Johnson', 'bob.johnson@example.com'),

(4, 'Alice', 'Lee', 'alice.lee@example.com'),

(5, 'Tom', 'Wilson', 'tom.wilson@example.com'),

(6, 'Sarah', 'Brown', 'sarah.brown@example.com'),

(7, 'David', 'Davis', 'david.davis@example.com');

SELECT \*

FROM Bookings

INNER JOIN Customers

ON Bookings.BookingID = Customers.CustomerID;

SELECT \*

FROM Bookings

LEFT JOIN Customers

ON Bookings.BookingID = Customers.CustomerID;

SELECT \*

FROM Bookings

RIGHT JOIN Customers

ON Bookings.BookingID = Customers.CustomerID;

SELECT \*

FROM Bookings

INNER JOIN TourGuides

ON Bookings.BookingID = TourGuides.GuideID;

SELECT \*

FROM Bookings

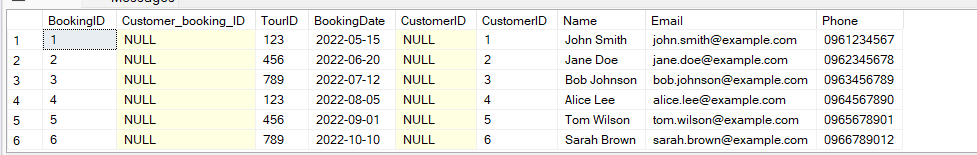
INNER JOIN TourGuides

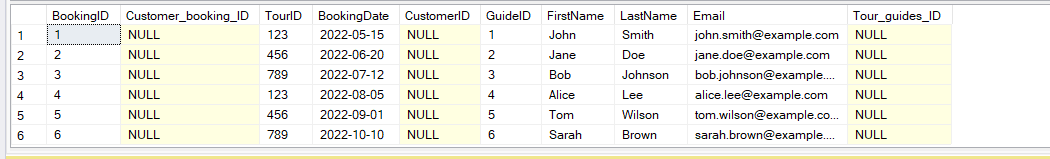
ON Bookings.BookingID = TourGuides.GuideID

WHERE Bookings.TourID > 200;

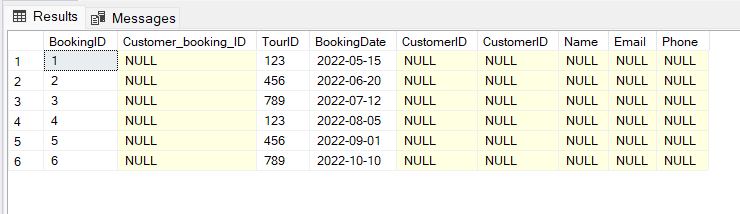
CREATE INDEX idx\_Customers\_CustomerID ON Customers(CustomerID);

**inner join**

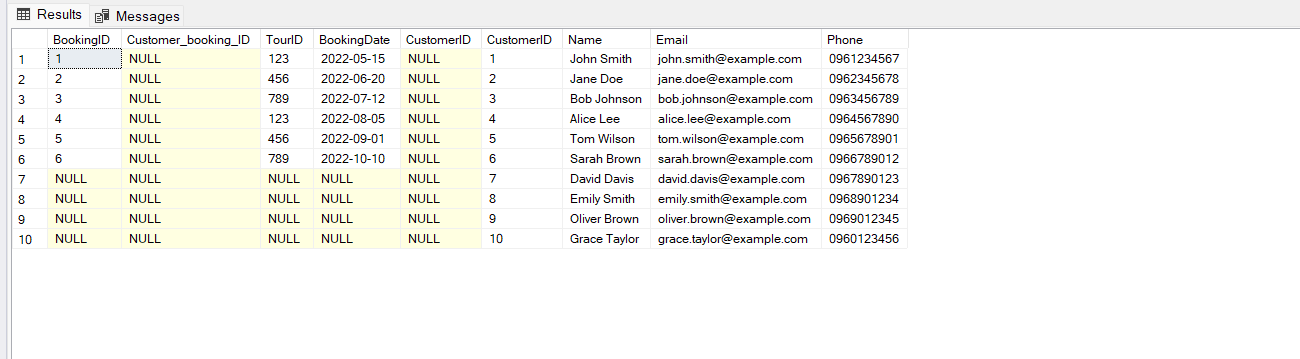
****

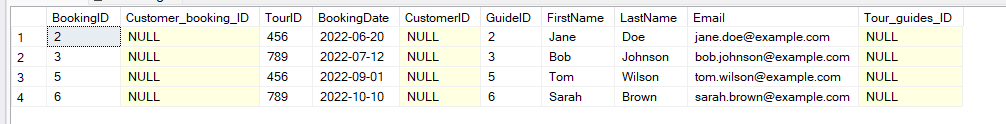
****

**left join**

****

**right join**

****

**join + TourID > 200**

**Контрольні питання:**

1. Індекс у базі даних - це механізм, що дозволяє швидко знаходити записи у таблиці бази даних за певними полями. Він включає в себе структуру даних, яка містить відображення значень поля та розташування записів у таблиці.
2. Індекси використовуються в базі даних для покращення продуктивності запитів на вибірку даних. Вони дозволяють зменшити час, необхідний для пошуку та вибірки даних з таблиці, а також для виконання операцій сортування та злиття.
3. Некластерний індекс - це індекс, що використовується для пошуку записів у таблиці за певним полем, але не змінює порядку зберігання даних у таблиці. Такі індекси дозволяють швидко знаходити записи за певним полем, але можуть призводити до збільшення розміру бази даних.
4. Оператор Join використовується для об'єднання даних з двох або більше таблиць у базі даних. Він дозволяє виконувати запити на отримання даних з різних таблиць і отримувати звіти з включенням даних з кількох джерел. Join можна використовувати для вибірки даних з двох таблиць зв'язаних між собою за допомогою зовнішнього ключа або для вибірки даних з декількох таблиць, які не мають прямого зв'язку між собою.

**Висновок:** У цій лабораторній роботі я перевірив свої навички з використання індексів та операторів Join.