DMA Project

Table creation

```
CREATE DATABASE companya_db;

USE companya_db;
```

Table Company

```
CREATE TABLE Company (
    Company_name VARCHAR(255) NOT NULL,
    Company_Id varchar(255) NOT NULL PRIMARY KEY,
    Location VARCHAR(255) NOT NULL,
    Area VARCHAR(255),
    Industry VARCHAR(255),
    Country VARCHAR(255) NOT NULL
);
```

Table User

```
CREATE TABLE User (
    Employee_Id INT NOT NULL PRIMARY KEY,
    Company_ID INT NOT NULL,
    Email_Id VARCHAR(255),
    First_Name VARCHAR(255),
    Last_Name VARCHAR(255),
    Department_name VARCHAR(255),
    FOREIGN KEY (Company_ID) REFERENCES Company(Company_ID) ON DELETE RESTRICT ON UPDATE CASCADE
);
```

Table Account

```
CREATE TABLE Account (
    Company_ID INT NOT NULL,
    Employee_Id INT NOT NULL,
    First_Name VARCHAR(255),
    Last_Name VARCHAR(255),
    Email_Id VARCHAR(255),
    Password VARCHAR(255),
    PRIMARY KEY (Company_ID, Employee_Id),
    FOREIGN KEY (Employee_Id) REFERENCES User(Employee_Id) ON DELETE RESTRICT ON UPDATE CASCADE,
    FOREIGN KEY (Company_ID) REFERENCES Company(Company_ID) ON DELETE RESTRICT ON UPDATE CASCADE
);
```

Table Booking

```
CREATE TABLE Booking (
    Booking_ID INT NOT NULL PRIMARY KEY,
    Employee_ID INT NOT NULL,
    Date DATE,
    Start_Time TIME,
    End_Time TIME,
    Time_Zone VARCHAR(255),
    FOREIGN KEY (Employee_ID) REFERENCES User(Employee_Id) ON DELETE RESTRICT ON UPDATE CASCADE
);
```

Table Car_Service

```
CREATE TABLE Cab_Service (
    Vendor_Name VARCHAR(255),
    Vendor_ID INT NOT NULL PRIMARY KEY,
    Employee_Id INT NOT NULL,
    Company_ID INT NOT NULL,
    Fuel_Type VARCHAR(255),
    Date DATE,
    Start_location VARCHAR(255),
    End_Location VARCHAR(255),
    Pickup_Time TIME,
    Passenger_Count INT,
    Distance DECIMAL(10,2),
    FOREIGN KEY (Company_ID) REFERENCES Company(Company_ID) ON DELETE RESTRICT ON UPDATE CASCADE,
    FOREIGN KEY (Employee_Id) REFERENCES User(Employee_Id) ON DELETE RESTRICT ON UPDATE CASCADE
);
```

Table Building

```
CREATE TABLE Building (
    Building_Name VARCHAR(255) NOT NULL PRIMARY KEY,
    Location VARCHAR(255) NOT NULL,
    Company_name VARCHAR(255),
    id int,
    FOREIGN KEY (id) REFERENCES Company(company_id)
);
```

Table Room_type

```
CREATE TABLE Room_Type (
    Room_ID INT NOT NULL PRIMARY KEY,
    Room_Name VARCHAR(255),
    Booking_Id INT,
    Capacity INT,
    Media VARCHAR(255),
    Building_Name VARCHAR(255) NOT NULL,
    Floor INT,
    Utilities VARCHAR(255),
    FOREIGN KEY (Building_Name) REFERENCES Building(Building_Name)
);
```

Querying

1. **List all the companies located in the USA and their respective users' booking information for the current month.**

```
SELECT c.Company_name, b.Booking_ID, b.Employee_ID, b.Date, b.Start_Time, b.End_Time
FROM Company c
INNER JOIN User u ON c.Company_Id = u.Company_ID
INNER JOIN Booking b ON u.Employee_Id = b.Employee_ID
WHERE c.Country = 'USA' AND MONTH(b.Date) = MONTH(CURRENT_DATE())
```

2. **Retrieve the total number of bookings made by users who are part of a specific company, sorted by date in descending order.**

```
SELECT u.Company_ID, b.Date, COUNT(*) AS total_bookings
FROM User u
INNER JOIN Booking b ON u.Employee_Id = b.Employee_ID
WHERE u.Company_ID = 12345
GROUP BY u.Company_ID, b.Date
ORDER BY b.Date DESC
```

3. **List all the cab services provided for a specific company, including the vendor name, employee name, pickup time, and the total distance covered for each ride.**

```
SELECT cs.Vendor_Name, u.First_Name, u.Last_Name, cs.Pickup_Time, cs.Distance
FROM Cab_Service cs
INNER JOIN User u ON cs.Employee_Id = u.Employee_Id
WHERE cs.Company_ID = 12345
```

4. **Retrieve the total number of bookings made by users in each department of a specific company, sorted by the number of** bookings **in descending order.**

```
SELECT u.Department_name, COUNT(*) AS total_bookings
FROM User u
INNER JOIN Booking b ON u.Employee_Id = b.Employee_ID
WHERE u.Company_ID = 12345
GROUP BY u.Department_name
ORDER BY total_bookings DESC
```

5. **List all the rooms that are currently available for booking in a specific building, including the room name, capacity, and utilities.**

```
SELECT rt.Room_Name, rt.Capacity, rt.Utilities
FROM Room_Type rt
LEFT JOIN Booking b ON rt.Booking_Id = b.Booking_ID
WHERE rt.Building_Name = 'ABC Building' AND b.Booking_ID IS NULL
```

6. **Retrieve the top 5 users who have made the highest number of bookings across all companies, including their names and the total number of bookings made.**

```
SELECT u.First_Name, u.Last_Name, COUNT(*) AS total_bookings
FROM User u
INNER JOIN Booking b ON u.Employee_Id = b.Employee_ID
GROUP BY u.First_Name, u.Last_Name
ORDER BY total_bookings DESC
LIMIT 5
```

7. **List all the bookings made by a specific user, including the date, start time, end time, and the room booked (if any).**

```
SELECT b.Date, b.Start_Time, b.End_Time, rt.Room_Name
FROM Booking b
LEFT JOIN Room_Type rt ON b.Booking_ID = rt.Booking_Id
WHERE b.Employee_ID = 12345
```

8. **Retrieve the number of cab rides made by each vendor for a specific company, sorted by the number of rides in descending order.**

```
SELECT cs.Vendor_Name, COUNT(*) AS total_rides
FROM Cab_Service cs
WHERE cs.Company_ID = 12345
GROUP BY cs.Vendor_Name
ORDER BY total_rides DESC
```

9. **List all the buildings owned by a specific company, including their names and locations, along with the total number of rooms available for booking in each building.**

```
SELECT b.Building_Name, b.Location, COUNT(*) AS total_rooms
FROM Building b
LEFT JOIN Room_Type rt ON b.Building_Name = rt.Building_Name
WHERE b.Company_name = 'Google’
```

10. **Find the average distance traveled by each passenger for every company in each industry that provides cab services:**

```
SELECT c.Industry, c.Company_name, AVG(cs.Distance / cs.Passenger_Count) AS
Avg_Distance_Per_Passenger
FROM Company c
```

```
INNER JOIN Cab_Service cs ON c.Company_Id = cs.Company_Id
GROUP BY c.Industry, c.Company_name
ORDER BY c.Industry, c.Company_name;
```

11. Find the total number of bookings made by each department for each company:

```
SELECT c.Company_name, u.Department_name, COUNT(*) AS Total_Bookings
FROM Company c
INNER JOIN User u ON c.Company_Id = u.Company_Id
INNER JOIN Booking b ON u.Employee_Id = b.Employee_Id
GROUP BY c.Company_name, u.Department_name
ORDER BY c.Company_name, u.Department_name;
```

Office Space Optimization

Milestone: Implementation in MySQL

Group 36

Krishna Teja Samudrala Saksham Saxena

+1 (380)2401366

+1 (341)2018552

Samudrala.k@northeastern.edu

Saxena.sa@northeastern.edu

Percentage of effort contributed by the student 1= 50%

Percentage of effort contributed by the student 2= 50%

Signature of Student 1: Krishna Teja Samudrala

Signature of Student 1: Saksham Saxena

Submission Date: 03/19/2023