LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

**Module Code & Module Title**

**CC4057NI Introduction to Information Systems**

**Assessment Weightage & Type**

**30% Individual Coursework**

**Year and Semester**

**2021 Spring**

**Student Name: Aashna Shrestha**

**Group: C13**

**London Met ID:**

**College ID: NP01CP4S210103**

**Assignment Due Date: 30th April, 2021**

**Assignment Submission Date: 30th April, 2021**

*I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.*

## Table of Contents

## List of Figures

## List of Tables

# 1 Introduction

“A database is any logically coherent collection of data organized for storage and retrieval by computers, as a single, possibly large, repository of data that can be used simultaneously by multi-users” (Eze, et al., 2014). A database contains all sorts of data required in any organization in a structured manner. The data might be as simple as a person’s name or complicated such as an image (Silberschatz, et al., 2019). Users can search for data in any pattern with the help of syntax (Britannica, 2020).

Database is managed by a software called Database Management System (DBMS). DBMS interacts with the user to create a database, and to retrieve, edit or delete data from the database. DBMS may vary according to the systems requirement. Relational DBMS shows the database relationship in tabular form where the data are organized in rows and columns. Network DBMS are presented in a graphical form, and the database has many to many relationships. Object oriented DBMS has more advanced features such as data types for graphics, audio and video. (Eze, et al., 2014)

# 2 Database Model

## 2.1 Business Model

Suppose that there is a Ridesharing Service named Swish, which allows passengers to book a vehicle such as cars (four-wheelers), bikes or scooters(two-wheelers) to travel a short distance. Passengers can send a request through their phones mentioning their pickup point and destination. A rider accepts the request and picks up the passenger from the mentioned location. The rider then drops the passenger to the desired location.

## 2.2 Business Rules

- A person can register to become a rider only if they own a vehicle and a driving license.
- The amount to be paid will be auto generated by the ride sharing app according to the distance. Discounts will not be given on the cost.
- A passenger may cancel the request only with a genuine reason.

## 2.3 Goals and objectives

Ride Sharing services work as public vehicles since they carry passengers and take them to any location that they desire. The service saves time and drops people to the exact destination that they desire. In fact, they even pick the passengers from any place. They are convenient than the public vehicles in terms of traffic congestion and cost.

## 2.4 Entity Relationship Diagram

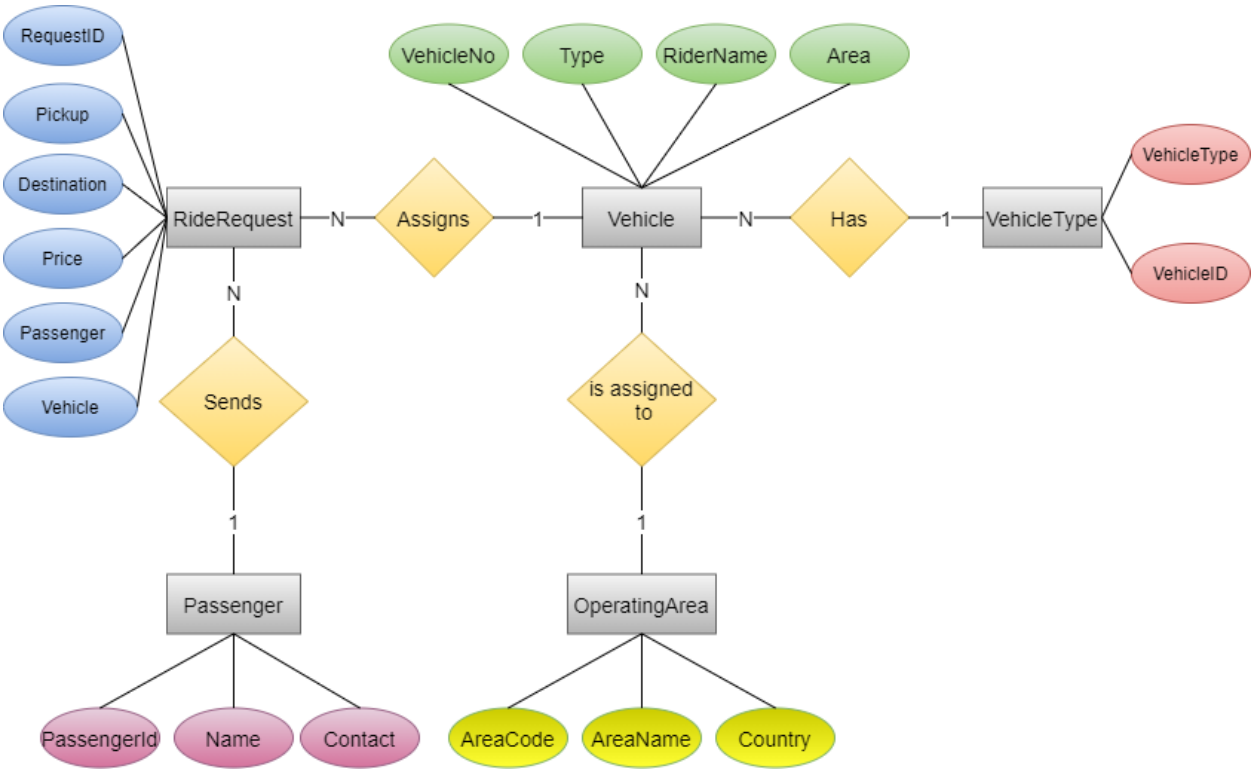


*Figure 1 Entity Relation Diagram of Swish Company*

## 2.5 Relational Diagram

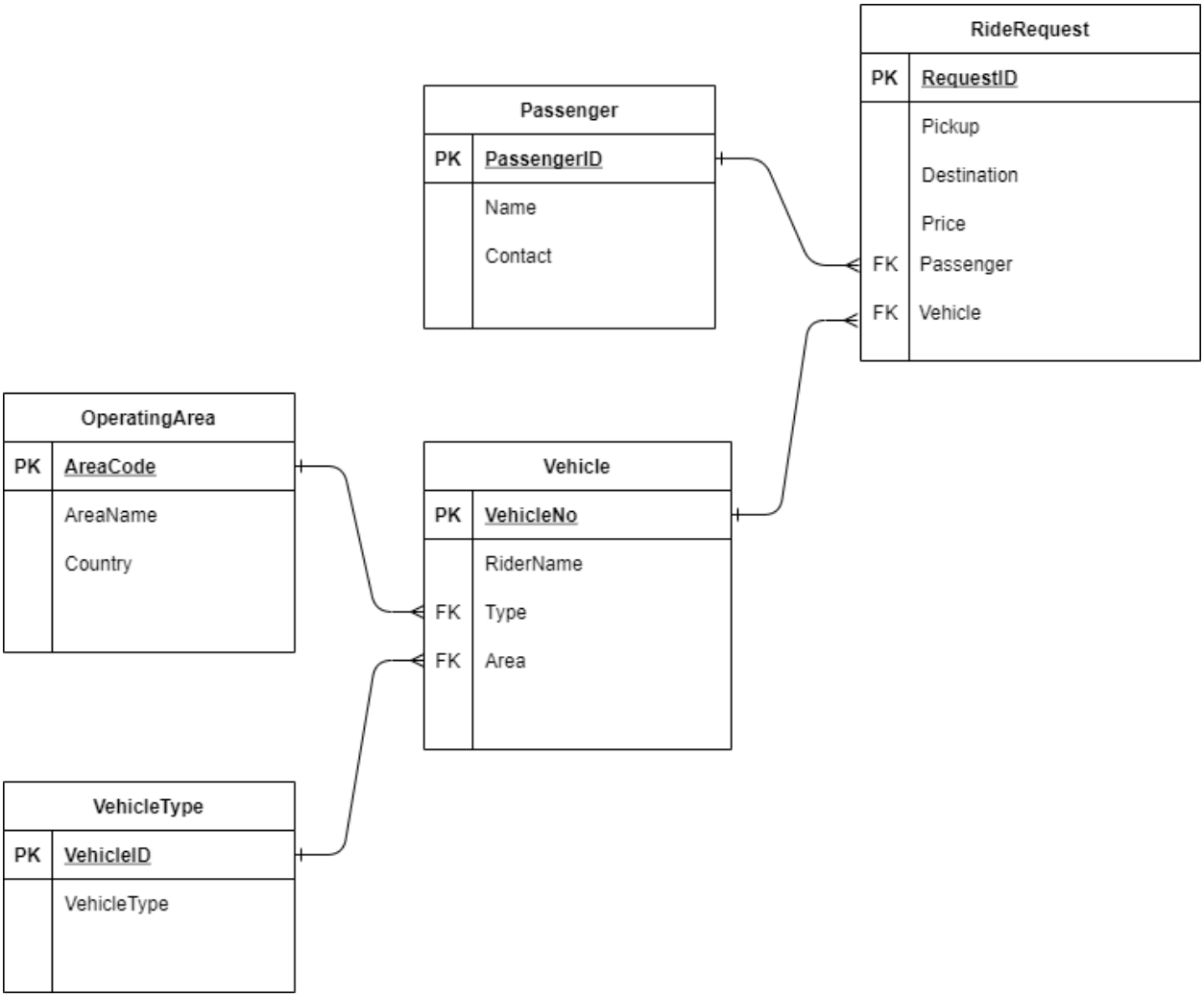


*Figure 2 Relational Diagram of Swish Company*

## 2.6 Table 1 Passenger

**Create Statement** - CREATE TABLE Passenger(PassengerID INT PRIMARY KEY AUTO_INCREMENT, Name VARCHAR(255), Contact VARCHAR(255));

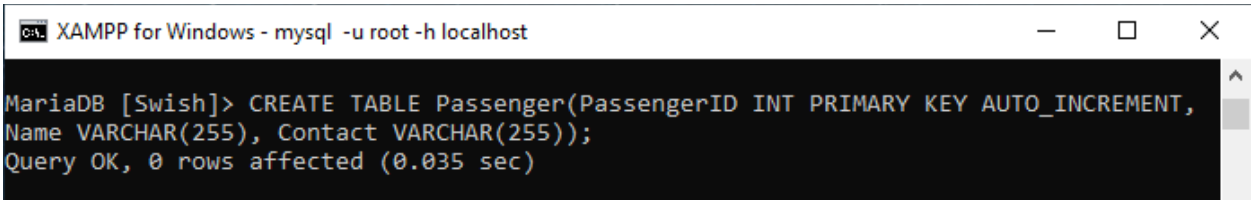


*Figure 3 Creating Passenger table*

**Describe Statement -** DESCRIBE Passenger;

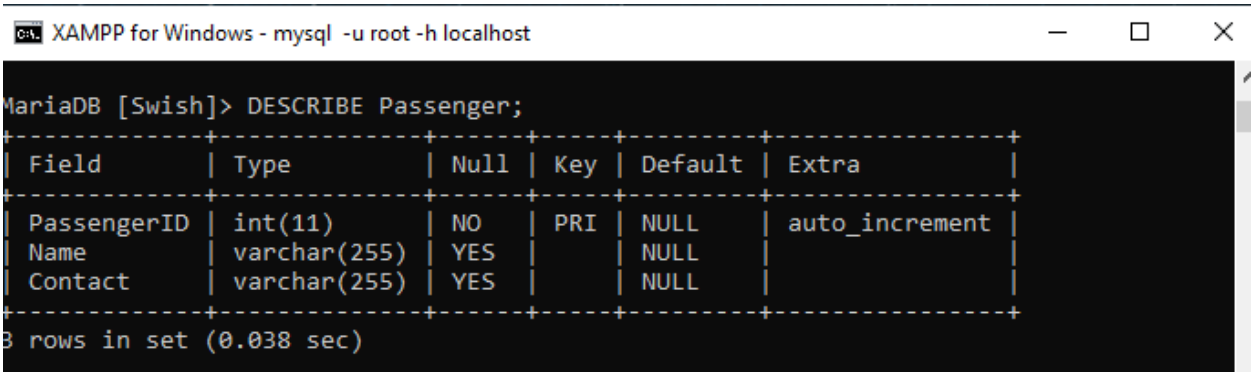


*Figure 4 Format of Passenger Table*

**Insert Statement -** INSERT INTO Passenger(name, contact) VALUES

("Liam Hemsworth", "122-458-249"),

("Shawn Harrington", "289-622-155"),

("Veronica Nielsen", "102-759-514"),

("Aaron Watson", "203-559-678"),

("Arial Lodge", "847-234-470"),

("Taylor Merch", "453-704-219"),

("Bob Brown", "389-555-104"),

("Ross Gilbert", "670-573-284"),

("Amy Merrell", "499-290-374"),

("Vanessa Crews", "385-883-058");

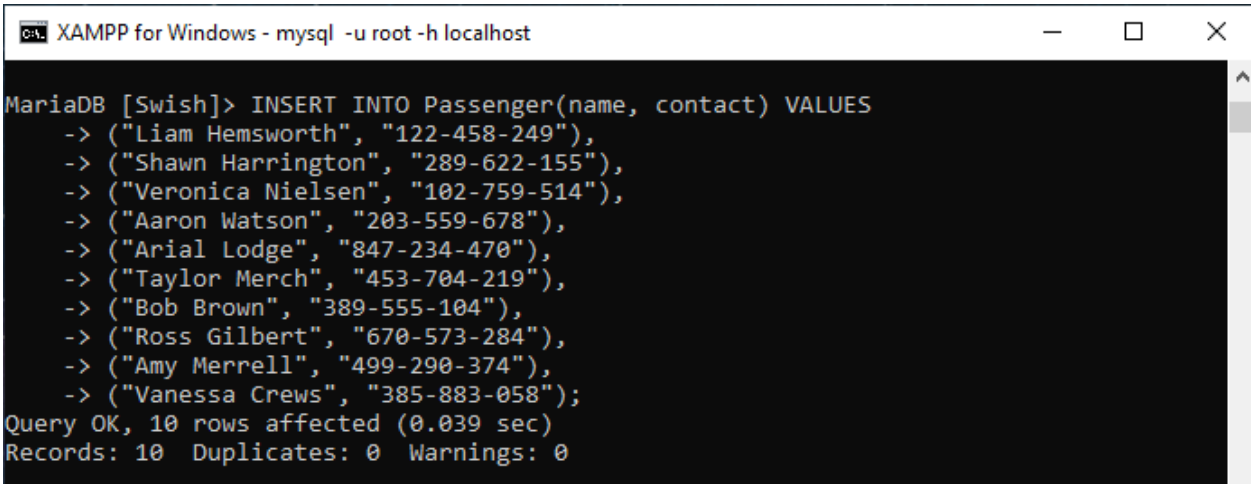


*Figure 5 Inserting values into Passenger table*

**Select Statement -** SELECT * FROM Passenger;

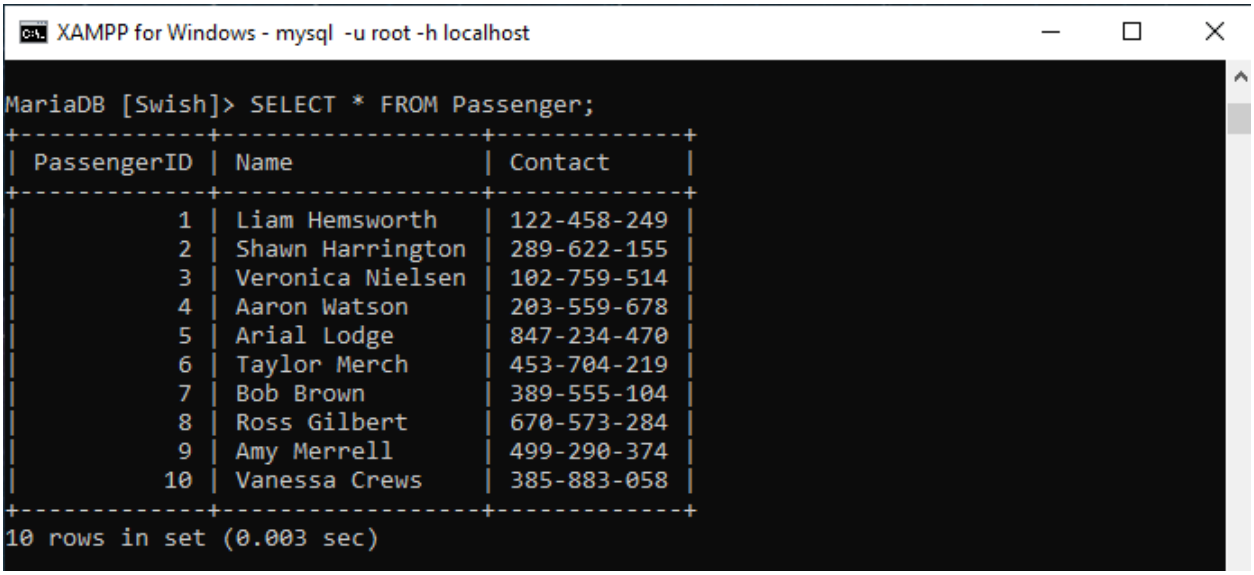


*Figure 6 Displaying the data of Passengers*

## 2.7 Table 2 VehicleType

**Create Statement -** CREATE TABLE VehicleType(VehicleID INT PRIMARY KEY, VehicleType VARCHAR(255));

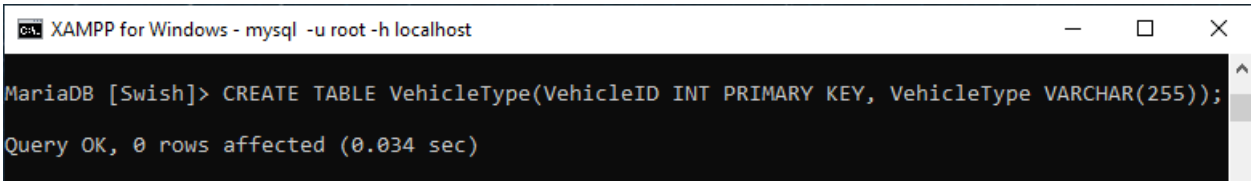


*Figure 7 Creating VehicleType Table*

**Describe Statement -** DESCRIBE VehicleType;

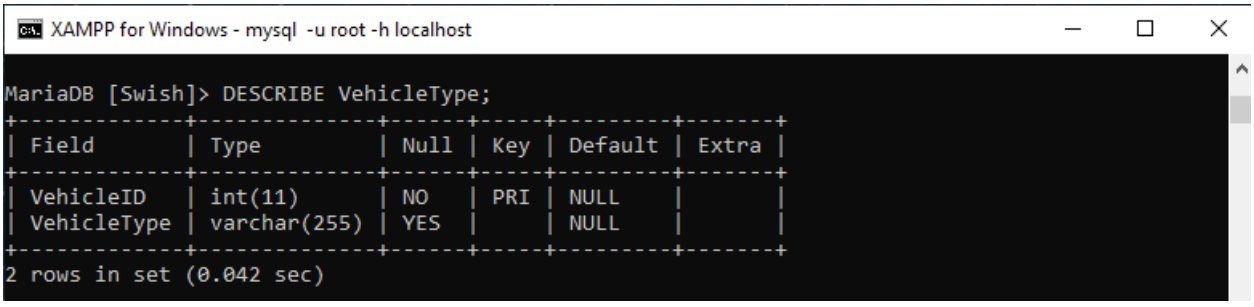


*Figure 8 Format of VehicleType table*

**Insert Statement -** INSERT INTO VehicleType VALUES

(1, "Two-Wheeler"),

(2, "Four-Wheeler");

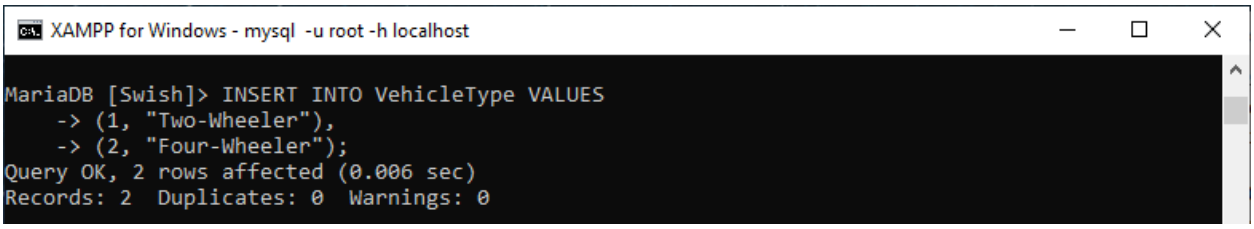


*Figure 9 Inserting values into VehicleType table*

**Select Statement -** SELECT * FROM VehicleType;

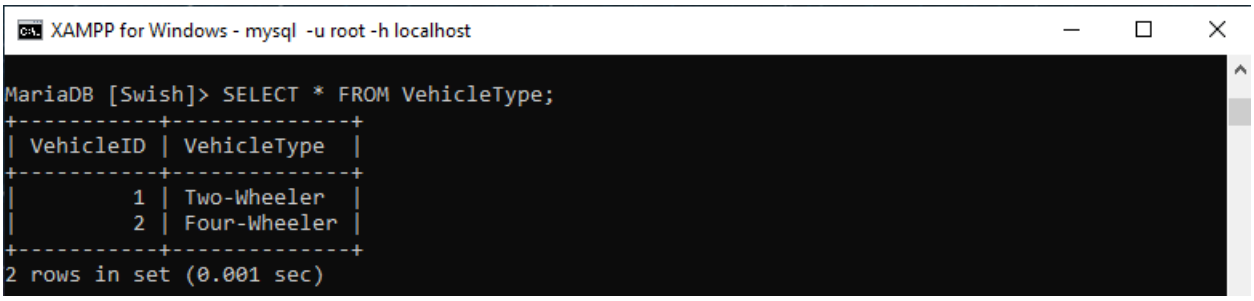


*Figure 10 Displaying data of VehicleType table*

## 2.8 Table 3 OperatingArea

**Create Statement -** CREATE TABLE OperatingArea(AreaCode INT PRIMARY KEY AUTO_INCREMENT, AreaName VARCHAR(255) UNIQUE, Country VARCHAR(255) DEFAULT "Nepal");

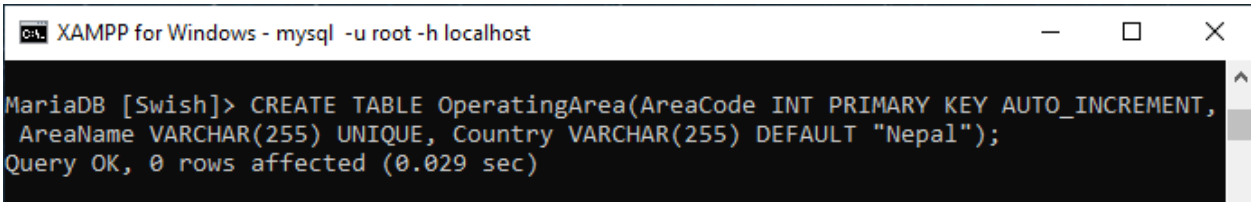


*Figure 11 Creating OperatingArea table*

**Describe Statement -** DESCRIBE OperatingArea;

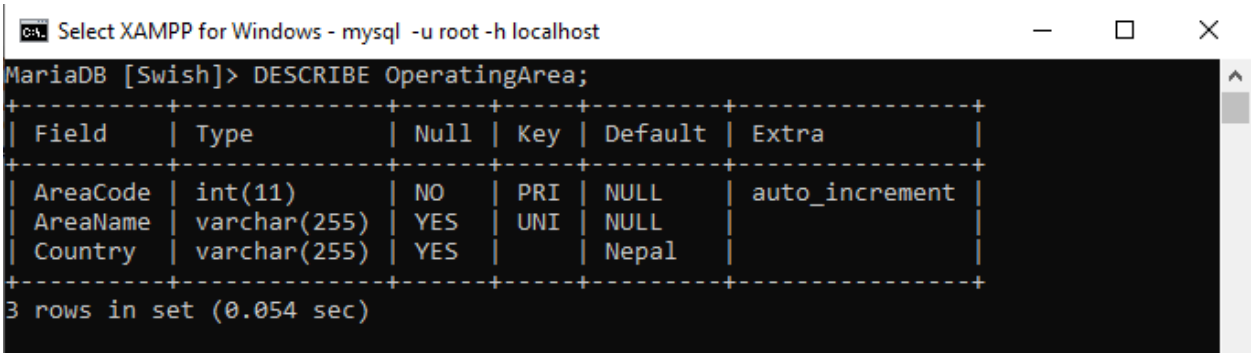


*Figure 12 Format of OperatingArea table*

**Insert Statement -** INSERT INTO OperatingArea(AreaName) VALUES

("Kathmandu"),

("Chitwan"),

("Pokhara"),

("Hetauda"),

("Biratnagar"),

("Birgunj"),

("Banepa"),

("Dharan"),

("Dhulikhel"),

("Janakpur");

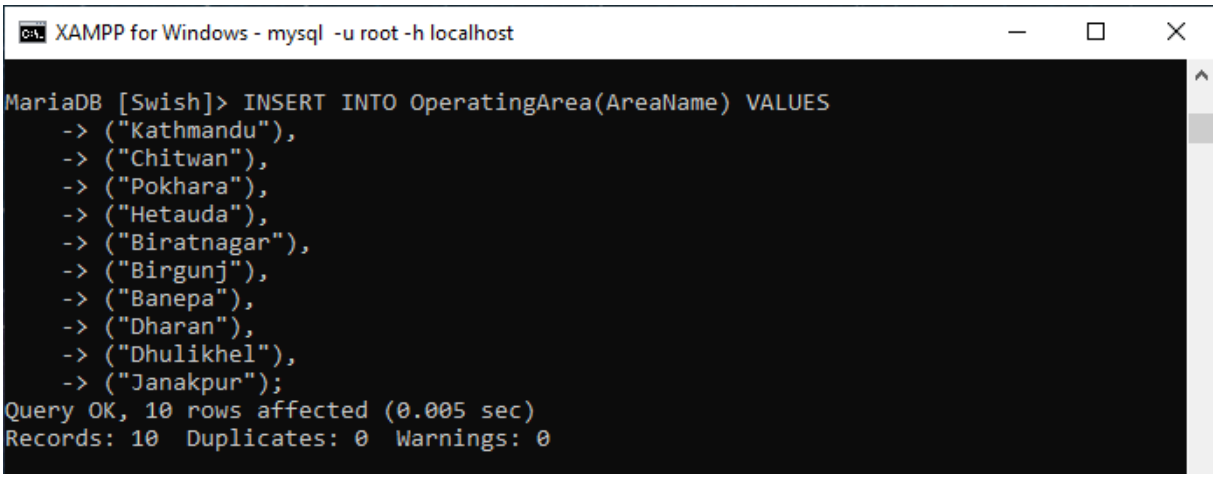


*Figure 13 Inserting values into OpeartingArea table*

**Select Statement -** SELECT * FROM OperatingArea;

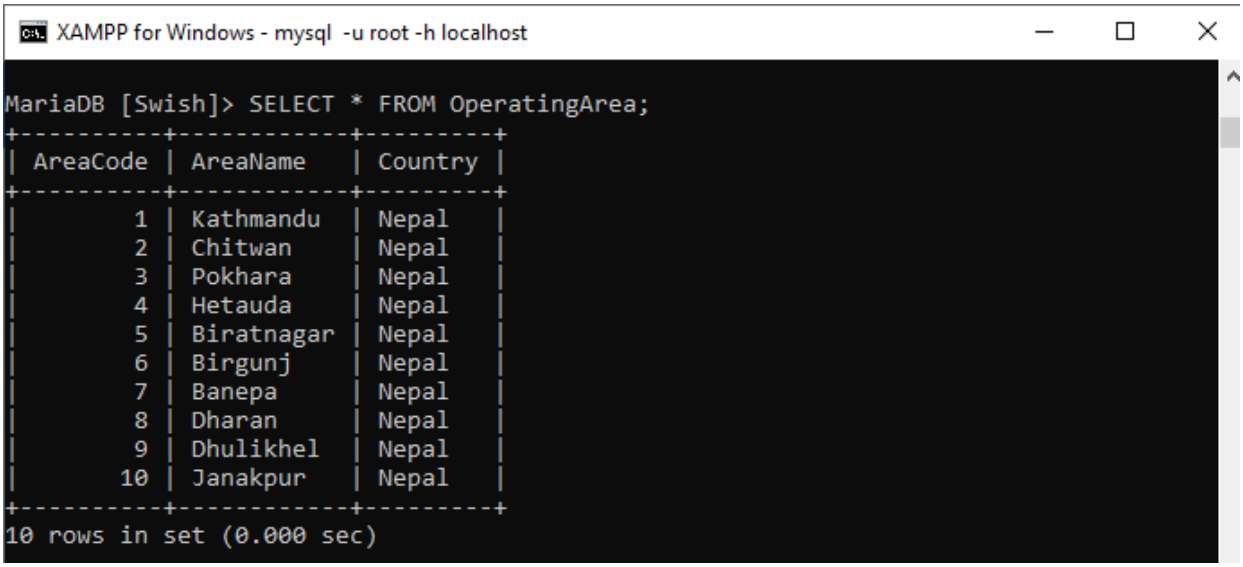


*Figure 14 Displaying data from OperatingArea table*

## 2.9 Table 4 Vehicle

**Create Statement -** CREATE TABLE Vehicle(VehicleNo INT PRIMARY KEY, RiderName VARCHAR(255), Type INT, Area INT, FOREIGN KEY(Type) REFERENCES VehicleType(VehicleID), FOREIGN KEY(Area) REFERENCES OperatingArea(AreaCode));

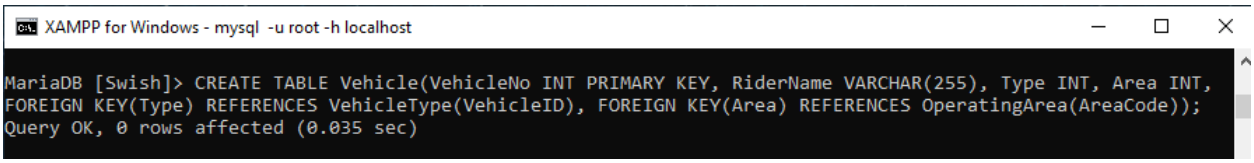


*Figure 15 Creating Vehicle table*

**Describe Statement -** DESCRIBE Vehicle;

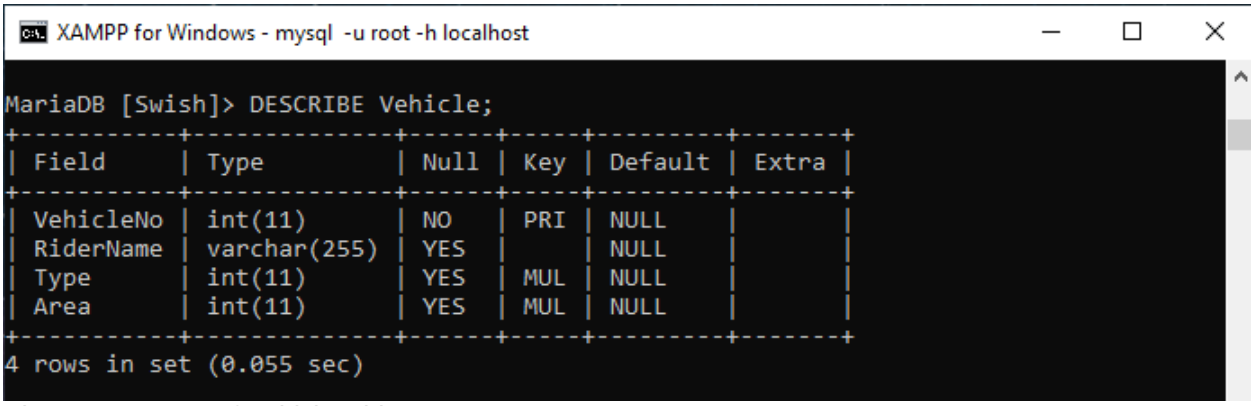


*Figure 16 Format of Vehicle table*

**Insert Statement -** INSERT INTO Vehicle VALUES

(1008, "Erin Smith", 2, 4),

(1458, "Bella Thomas", 1, 6),

(2495, "Michael Stark", 1, 8),

(4932, "Raen West", 1, 7),

(5659, "Derek Burris", 2, 3),

(5660, "Mindy Holmes", 1, 3),

(1048, "Rylie Green", 2, 2),

(7584, "John Reeves", 2, 5),

(7592, "Eva White", 2, 9),

(3849, "Allen Bieber", 1, 9),

(9208, "Levi Gomez", 1, 10),

(7593, "Zera Horan", 2, 1);

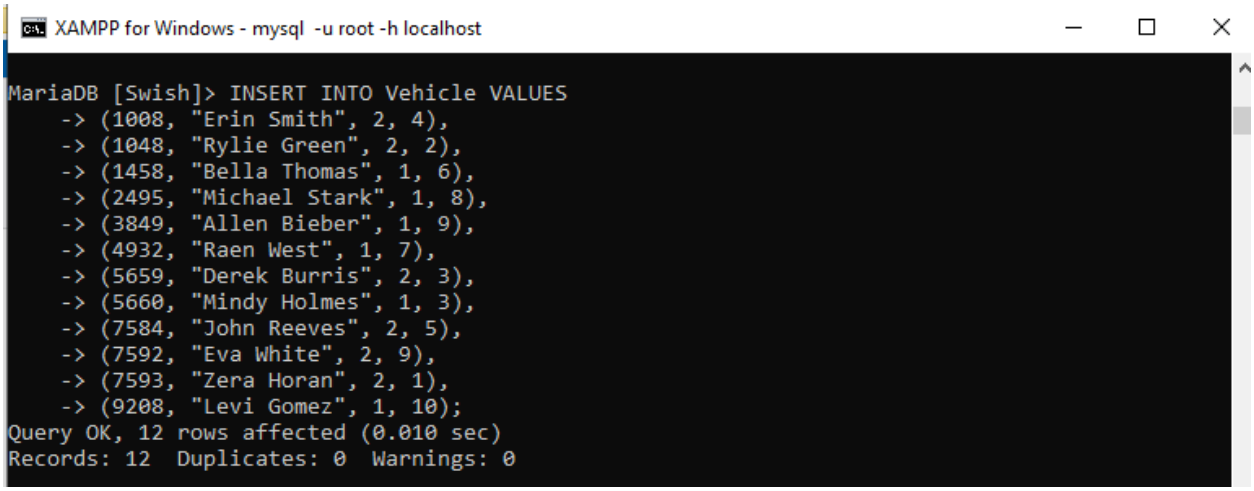


*Figure 17 Inserting values into Vehicle table*

**Select Statement -** SELECT * FROM Vehicle;

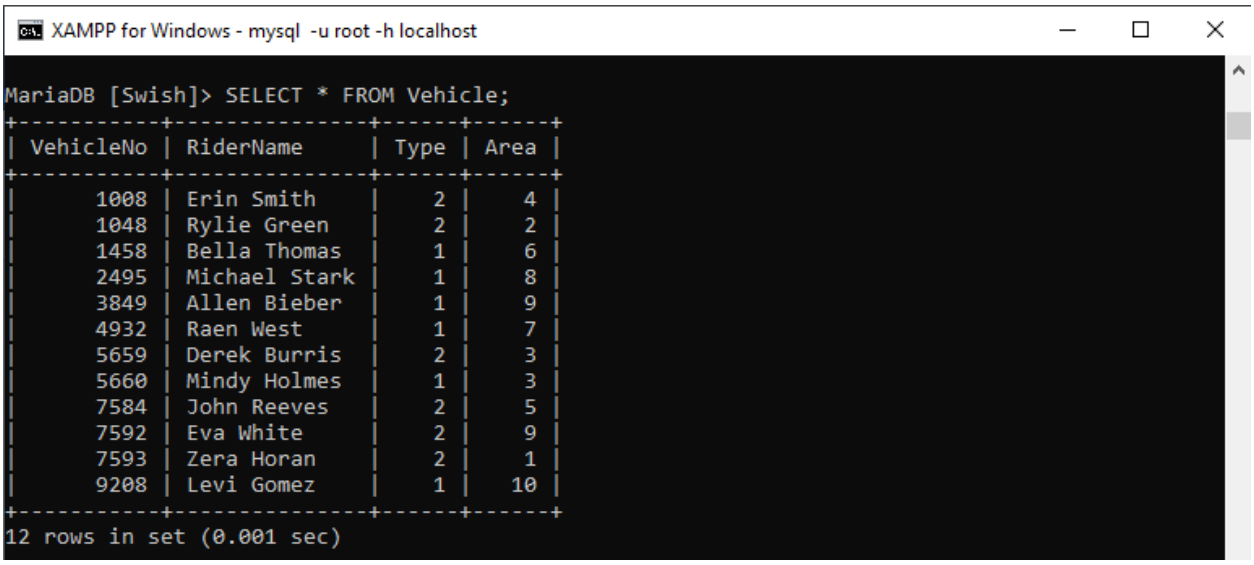


*Figure 18 Displaying data from Vehicle table*

## 2.10 Table 5 RideRequest

**Create Statement -** CREATE TABLE RideRequest(RequestID INT PRIMARY KEY AUTO_INCREMENT, Pickup VARCHAR(255) NOT NULL, Destination VARCHAR(255) NOT NULL, Passenger INT, Vehicle INT, Price FLOAT(2), FOREIGN KEY(Passenger) REFERENCES Passenger(PassengerID), FOREIGN KEY(Vehicle) REFERENCES Vehicle(VehicleNo));

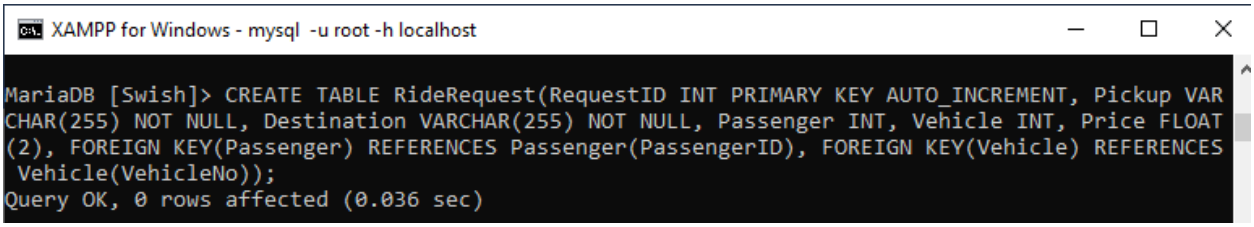


*Figure 19 Creating RideRequest table*

**Describe Statement -** DESCRIBE RideRequest;

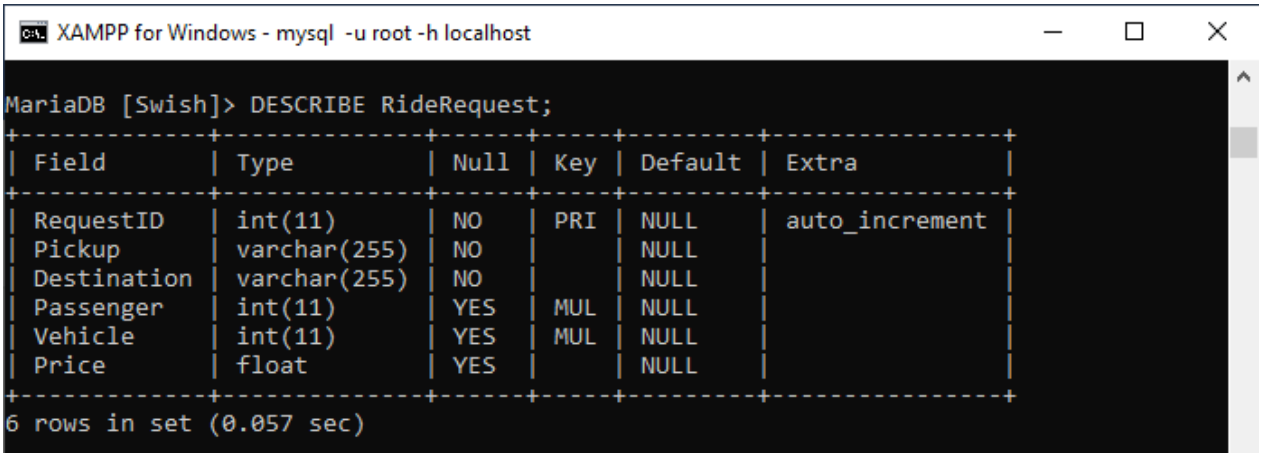


*Figure 20 Format of RideRequest*

**Insert Statement -** INSERT INTO RideRequest (Pickup, Destination, Passenger, Vehicle, Price) VALUES

("Lagankhel", "Basantapur Durbar Sqaure", 2, 7593, 121.50),

("Shahid Smarak Park", "Pathibhara Temple", 8, 1008, 236.00),

("Sauraha", "Chitwan National Park", 4, 1048, 360.00),

("Zero Mile", "Janaki Temple", 7, 9208, 51.00),

("Tin Dhara Chowk", "Chandeshwori Temple", 3, 4932, 48.00),

("Swayambhunath Stupa", "Lazimpat", 1, 7593, 264.00),

("Mahendra Cave", "Devi's Falls", 1, 5659, 536.00),

("Mirabel Resort Hotel", "Kathmandu University", 5, 3849, 79.50),

("Biratnagar Airport", "Biratnagar Eye Hospital", 7, 7584, 444.00),

("Bindhyabasini Temple", "Fewa Lake", 1, 5660, 205.50),

("Ghantaghar", "Raxaul", 4, 1458, 121.50),

("Bijayapur", "Rai Tole", 10, 4932, 61.5);

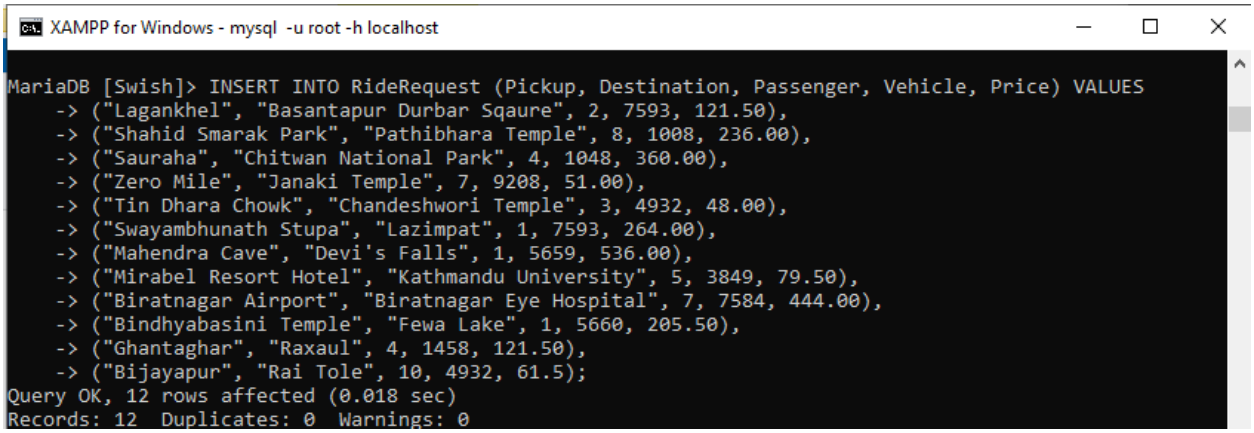


*Figure 21 Inserting values into RideRequest table*

**Select Statement -** SELECT * FROM RideRequest

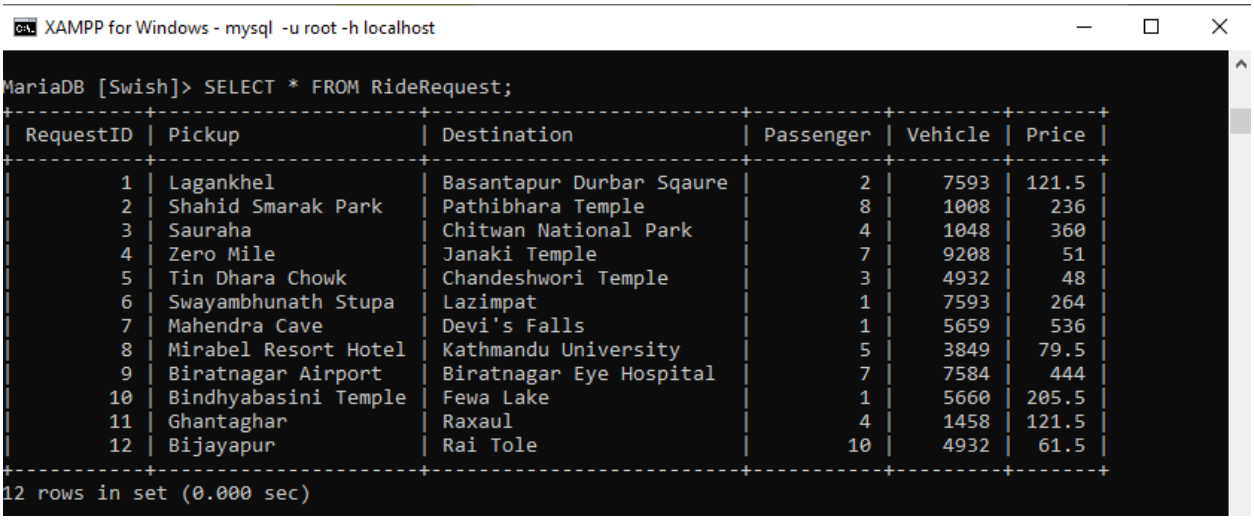


*Figure 22 Displaying data from RideRequest table*

# 3 Data Dictionary

| Entity name | Entity description | Column name | Column description | Data Type | Length | Primary key | Foreign key | Nullable | Unique | Notes |
|---|---|---|---|---|---|---|---|---|---|---|
| Passenger | Passengers are the people who use the ride sharing service to travel to a particular destination | PassengerID | An ID given to each passenger to identify them | INT | | True | False | False | True | Auto Incremented |
| | | Name | Name of the passenger | VARCHAR | 255 | False | False | True | False | |
| | | Contact | The passenger's phone number | VARHCHAR | 255 | False | False | True | False | |

*Table 1 Data Dictionary of Passenger table*

| Entity name | Entity description | Column name | Column description | Data Type | Length | Primary key | Foreign key | Nullable | Unique | Notes |
|---|---|---|---|---|---|---|---|---|---|---|
| VehicleType | It is the type of vehicle owned by the ride sharing company | VehicleID | An ID given to each type of vehicle | INT | | True | False | True | True | |
| | | VehicleType | The type of vehicle | VARCHAR | 255 | False | False | True | False | |

*Table 2 Data Dictionary of VehicleType table*

| Entity name | Entity description | Column name | Column description | Data Type | Length | Primary key | Foreign key | Nullable | Unique | Notes |
|---|---|---|---|---|---|---|---|---|---|---|
| OperatingArea | It is the area that a vehicle has been assigned. A vehicle can run only inside its assigned area. | AreaCode | A code given to a certain area | INT | | True | False | False | True | Auto Incremented |
| | | AreaName | The area's name | VARCHAR | 255 | False | False | True | False | |
| | | Country | The country in which the area is located | VARCHAR | 255 | False | False | True | False | Default value is set as "Nepal" |

*Table 3 Data Dictionary of OperatingArea table*

| Entity name | Entity description | Column name | Column description | Data Type | Length | Primary key | Foreign key | Nullable | Unique | Notes |
|---|---|---|---|---|---|---|---|---|---|---|
| Vehicle | The transport medium registered to the ride sharing company which lets the passenger travel to their destination | VehicleNo | The number written on the number plate of the vehicle | INT | | True | False | False | True | |
| | | RiderName | The name of the rider who owns the vehicle | VARCHAR | 255 | False | False | True | False | |
| | | Type | ID of the type of vehicle | INT | | False | True | True | False | References to VehicleID Column of VehicleType table |
| | | Area | The code of the area where the vehicle can operate | INT | | False | True | True | False | References to AreaCode column of OperatingArea table |

*Table 4 Data Dictionary of Vehicle table*

| Entity name | Entity description | Column name | Column description | Data Type | Length | Primary key | Foreign key | Nullable | Unique | Notes |
|---|---|---|---|---|---|---|---|---|---|---|
| RideRequest | A request sent by the passenger to pick them up and drop them to a certain destination | RequestID | An ID given to each request sent by the passenger | INT | | True | False | False | True | |
| | | Pickup | The location from where the passenger is picked up | VARCHAR | 255 | False | False | True | True | |
| | | Destination | The location where the passenger is to be dropped | VARCHAR | 255 | False | False | True | True | |
| | | Price | The cost that a passenger needs to pay | FLOAT | 2 | False | False | True | False | |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Passenger | The ID of the passenger who sent the request | INT | | False | True | True | False | References PassengerID column from Passenger table |
| | | Vehicle | The ID of the vehicle that has been assigned | INT | | False | True | True | False | References VehicleNo column from Vehicle table |

*Table 5 Data Dictionary of RideRequest table*

# 4 Queries

Query 1

Query - SELECT * FROM OperatingArea ORDER BY AreaName;

Keyword - ORDER BY

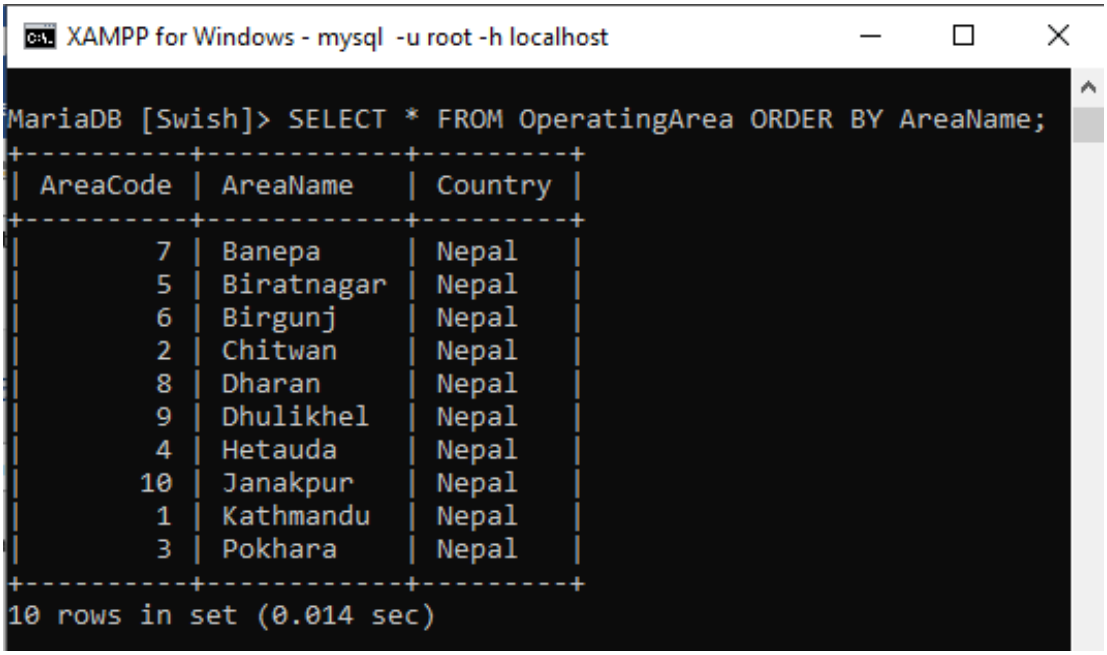


*Figure 23 Arranging the name of Area alphabetically*

Query 2

Query - SELECT SUM(Price) AS Total FROM RideRequest WHERE Vehicle = 7593;

Keyword - WHERE, SUM(Price)

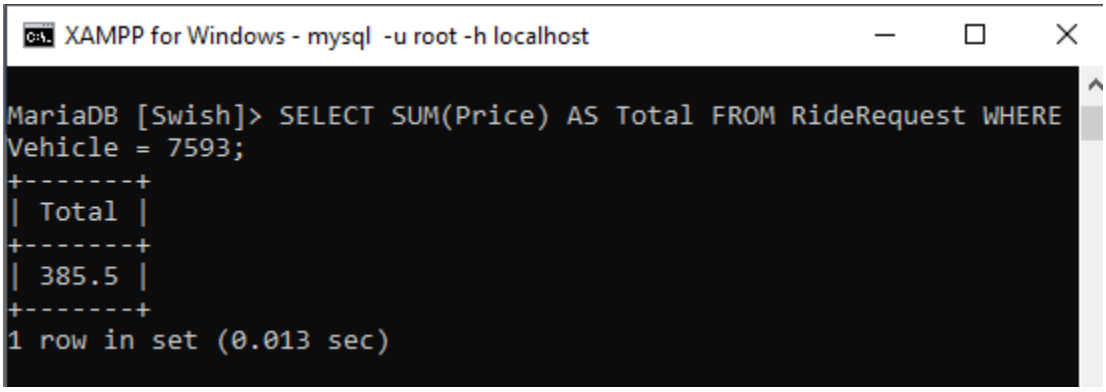


*Figure 24 Total price earned by the vehicle 7593*

Query 3

SELECT Pickup, Destination, Vehicle FROM RideRequest WHERE Pickup LIKE "S%";

Keyword LIKE, WHERE

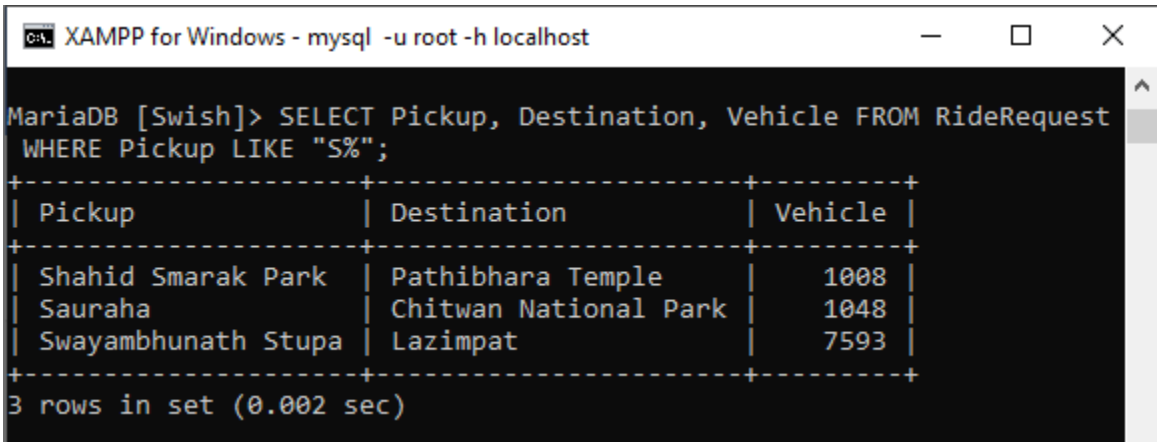


*Figure 25 Pickup Locations starting from the letter 'S'*

Query 4 LIMIT

Query - SELECT Vehicle, Price FROM RideRequest ORDER BY Price LIMIT 1;

Keyword LIMIT

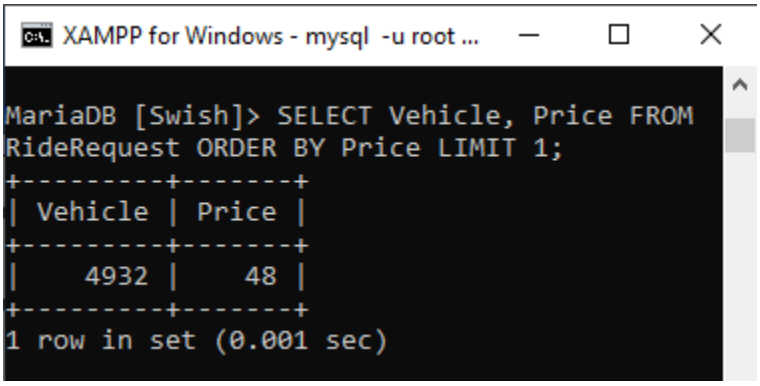


*Figure 26 Vehicle which earned the least*

Query 5

Query - SELECT Passenger.PassengerID, Passenger.Name AS PassengerName, Vehicle.VehicleNo, VehicleType.VehicleType, RideRequest.Price
FROM Passenger JOIN RideRequest ON Passenger.PassengerID = RideRequest.Passenger
JOIN Vehicle ON RideRequest.Vehicle = Vehicle.VehicleNo
JOIN VehicleType ON Vehicle.Type = VehicleType.VehicleID;

Keyword JOIN

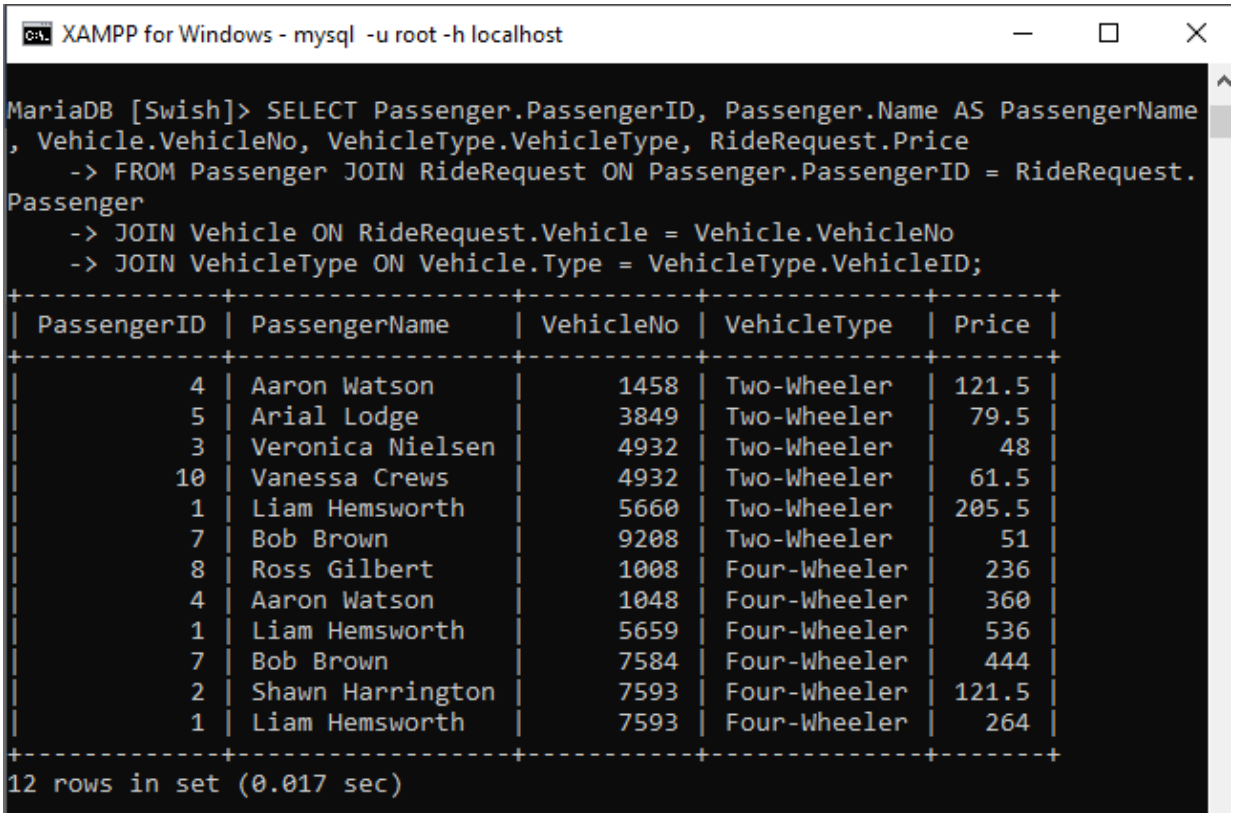


*Figure 27 Displaying some details from the tables Passenger, RideRequest and VehicleType*

Query 6 LEFT JOIN

Query SELECT * FROM Passenger LEFT JOIN RideRequest ON Passenger.PassengerID = RideRequest.Passenger;

Keyword LEFT JOIN

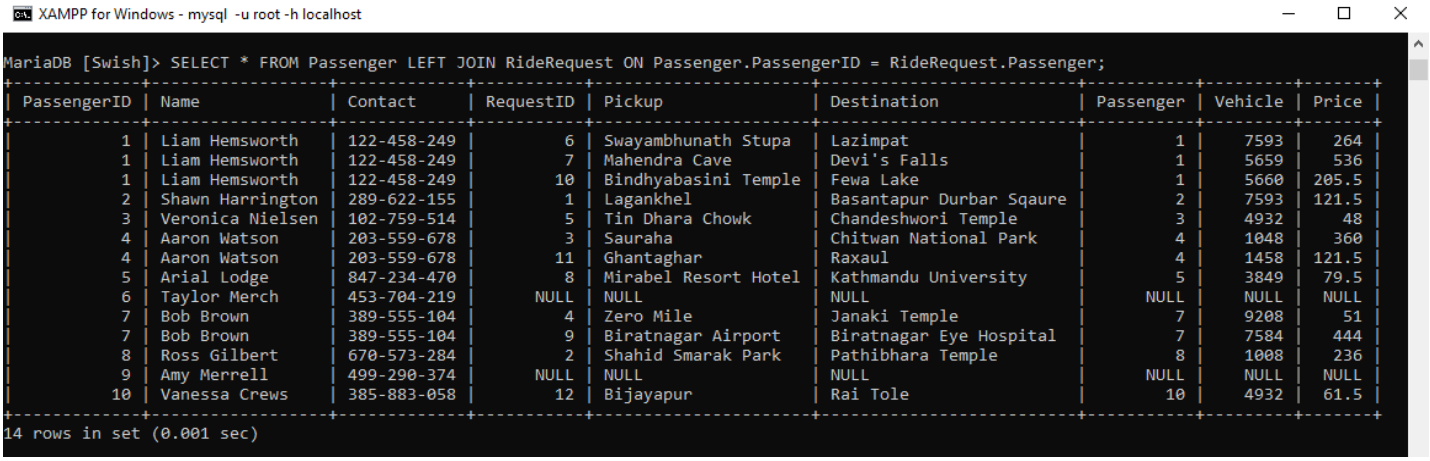


*Figure 28 Using Left Join to display all the details of passengers and their ride requests*

Query 7 RIGHT JOIN

Query - SELECT Vehicle.VehicleNo, Vehicle.Ridername, RideRequest.RequestID, RideRequest.Price FROM Vehicle RIGHT JOIN RideRequest ON Vehicle.VehicleNo = RideRequest.Vehicle;

Keyword RIGHT JOIN

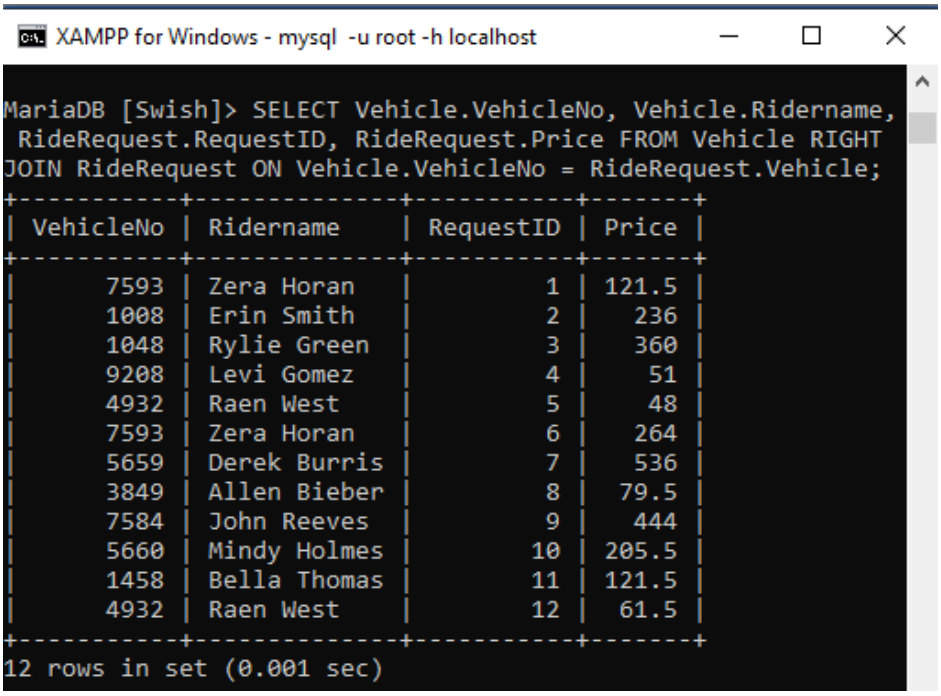


*Figure 29 Using Right Join to display some details of Vehicle and RideRequest*

Query 8

Query - SELECT Type, COUNT(*) AS Total_Vehicles FROM Vehicle GROUP BY Type;

Keyword - COUNT(*), GROUP BY

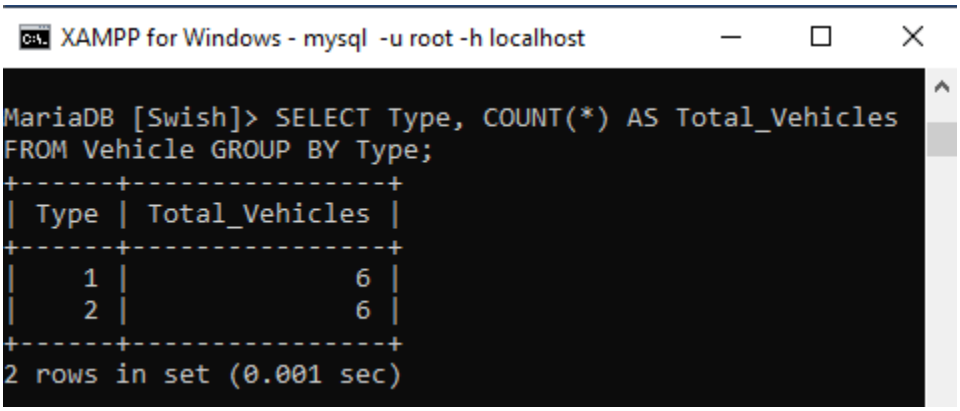


*Figure 30 Total number of different types of vehicles*

Query 9

Query - SELECT Vehicle, Price FROM RideRequest WHERE Price BETWEEN 200 and 400;

Keyword - BETWEEN

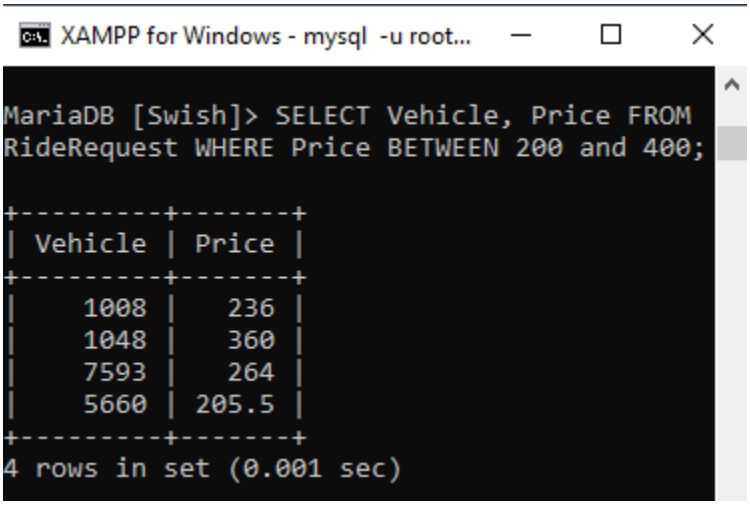


*Figure 31 The vehicles which earned 200 - 400*

Query 10

SELECT DISTINCT(VehicleType) FROM VehicleType;

Keyword DISTINCT

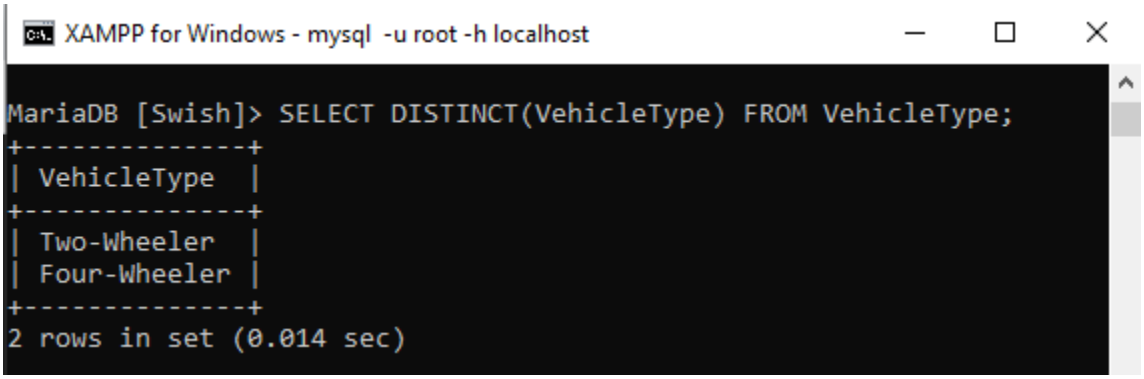


*Figure 32 Displaying types of vehicles*

# 5 Conclusion

A database model reflects the entire system of a company. In this project, various components helped to create the database model. The entity relationship diagram helped to plan what entities will be used by the ride sharing service, Swish, and the way they will be connected with other entities. The relational diagram further clarified the attributes that will be used to link the entities i.e., the primary key and the foreign key. The two diagrams helped to set a draft about the way the company will operate.

After preparing a plan about the company's system, it was the turn to create a database through MYSQL. Once the database was created, the tables were created along with the attributes. Then, data were inserted into the tables. A data dictionary, which defined all the attributes inside a table, was created for each entity. It provides reference to the user whenever they need to work with the database.

Since the database has been created and records have been added to it, a user can search for any pattern of data. They will only be required to use syntaxes for the purpose. In this project, select queries were used to search for data such as the names of passengers in alphabetical order, the number of two-wheeler and four-wheeler vehicles, the details of ride booked by the passengers and so on.

# 6 Bibliography

Britannica, E., 2020. *Database.* [Online]
Available at: https://www.britannica.com/technology/database
[Accessed 2 April 2021].

Eze, U. F., Etus, C. & Uzukwu, J. E., 2014. *Database System Concepts, Implementations and Organizations-A Detailed Survey.* [Online]
Available at: https://www.researchgate.net/profile/Chukwuemeka-Etus/publication/326468693_Database_System_Concepts_Implementations_and_Organizations-A_Detailed_Survey/links/5b4fca66a6fdcc8dae2b4139/Database-System-Concepts-Implementations-and-Organizations-A-Detailed-S
[Accessed 2 April 2021].

Silberschatz, A., Korth, H. & Sudarshan, S., 2019. *Database System Concepts.* [Online]
Available at: http://libgen.li/item/index.php?md5=3B9A3801ED66581C28C040B370511818
[Accessed 3 April 2021].