



SOUTHEAST UNIVERSITY

Department of Computer Science and Engineering

Database Design Project

Course Name: Database Design Lab

Course Code : CSE384.6

Section : 06

SUBMITTED TO:

S.I.M. ADNAN[IN]

Lecturer,Dept of CSE

Southeast University

SUBMITTED BY:

Name : Md.Shaurov Hayat Emon -2023100000315

Name : Shakila Khan Nawreen -2023100000286

Name : Shoaeb Islam -2023100000120

Date of Submission : 15th January 2026

● INTRODUCTION

◆ Overview of the System

The **Shuttle Management System** is a database-based system designed to manage shuttle transportation services in an organization or university. The system helps in maintaining information about shuttles, drivers, routes, schedules, and passengers in an organized and efficient manner.

This system reduces manual work, improves data accuracy, and ensures smooth transportation management.

◆ Objectives of the Project

The main objectives of this project are:

1. To store and manage shuttle, driver, and route information efficiently
2. To keep track of passenger details and shuttle schedules
3. To reduce manual record keeping
4. To provide easy access to transportation data
5. To improve overall transportation management efficiency

◆ Scope of the Project

The scope of this project includes:

- 1) Managing shuttle details such as shuttle number, capacity, and type
- 2) Managing driver information
- 3) Managing routes and timings
- 4) Tracking passenger usage
- 5) Generating useful reports using SQL queries

● REQUIREMENTS DESCRIPTION

◆ System Description

The **Shuttle Management System** is designed to manage the transportation services of a university. It helps in organizing shuttle vehicles, drivers, routes, schedules, and passenger details in a structured database. The system reduces manual record keeping and makes shuttle operations more efficient and reliable.

The system stores all necessary information digitally so that administrators can easily manage shuttle operations and retrieve information when needed.

◆ Entities Involved

The main entities involved in the system are:

1. **Shuttle**
Stores information about each shuttle such as shuttle ID, registration number, capacity, and type.
2. **Driver**
Stores driver details like driver ID, name, contact number, and license number.
3. **Route**
Contains route information including route ID, starting point, destination, and distance.
4. **Schedule**
Stores shuttle timing details such as departure time, arrival time, and assigned route.
5. **Passenger**
Stores passenger details like passenger ID, name, department, and contact number.

◆ System Users

The system can be accessed by the following users:

1)Administrator

- >Manages shuttles, drivers, routes, and schedules.
- >Updates and maintains the database.

2)Passengers (Students/Staff)

- >Use shuttle services and are registered in the system.

◆ Functional Requirements

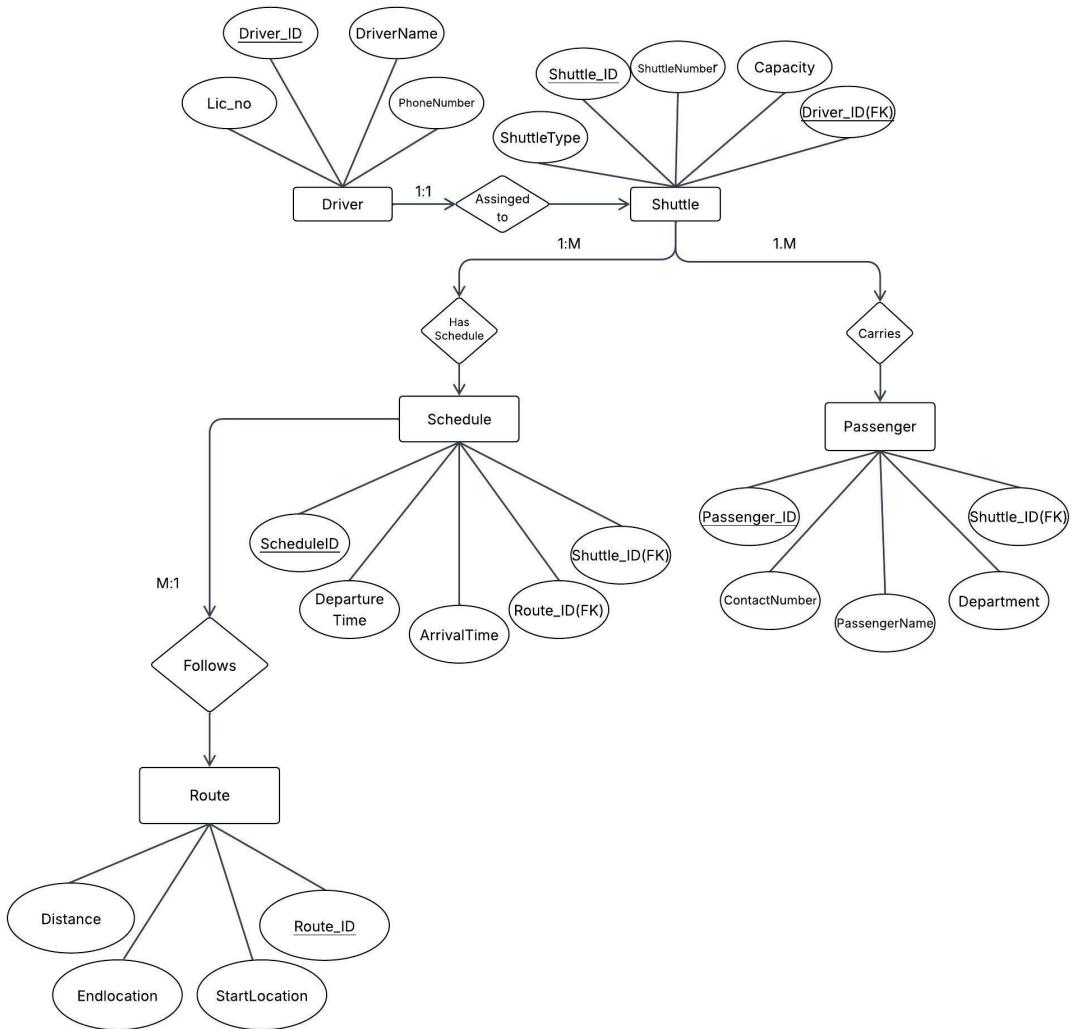
The system should be able to:

1. Add, update, and delete shuttle information
2. Maintain driver and passenger records

3. Assign drivers to shuttles
4. Assign routes and schedules to shuttles
5. Store and retrieve passenger travel data
6. Generate reports related to shuttles, routes, and passengers

● E-R Diagram:

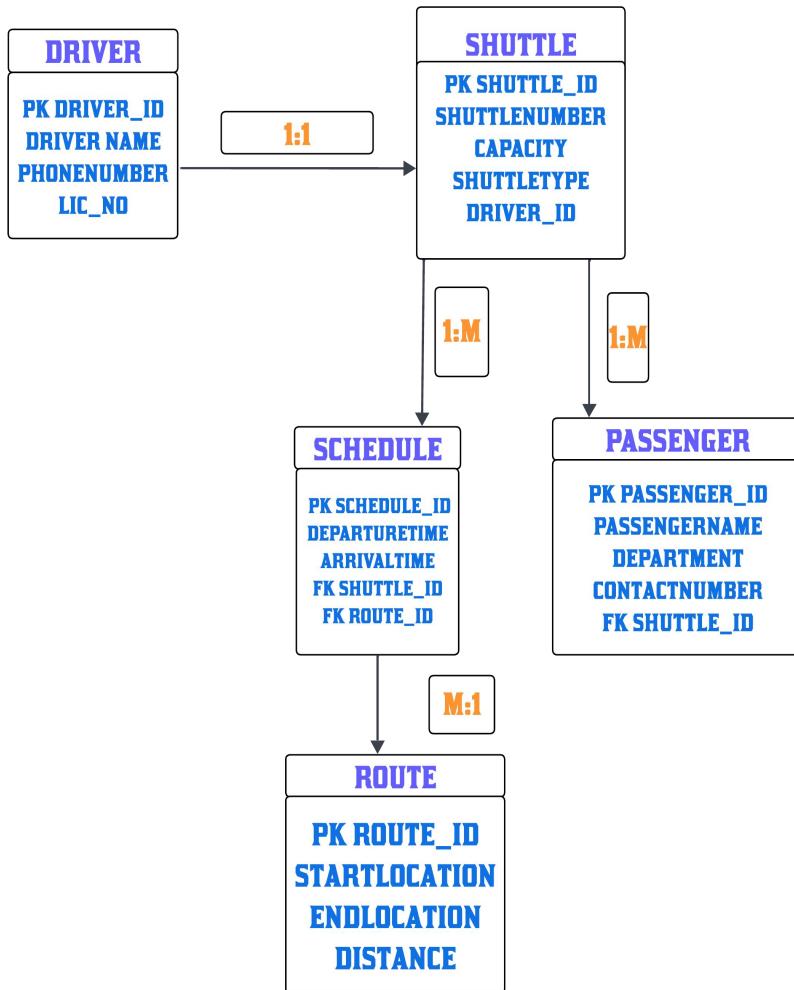
The ER diagram shows the overall structure of the Shuttle Management System by representing entities, attributes, and relationships. It provides a clear conceptual view of the database design. Tool used: **Lucidchart**



● Relational Schema:

Based on the ER diagram, the relational schema is developed by transforming entities into relations and relationships into foreign key constraints. The relational schema clearly represents table structures,

primary keys, and foreign key relationships of the Shuttle Management System.



● Normalization and Final Relational Schema:

Normalization is the process of organizing data in a database to reduce redundancy and improve data integrity. In this project, the database is normalized up to Third Normal Form (3NF).

◆ First Normal Form (1NF):

Definition:

A relation is in First Normal Form (1NF) if:

1. All attributes contain atomic values.
2. There are no repeating groups.

3. Each table has a primary key.

Application in Shuttle Management System:

All tables in this project contain atomic values and have defined primary keys. No multi-valued or repeating attributes exist.

The Conclusion is all relations satisfy **1NF**.

◆ Second Normal Form (2NF):

Definition:

A relation is in Second Normal Form (2NF) if:

1. It is already in 1NF.
2. There is no partial dependency on a composite primary key.

Application in Shuttle Management System:

All relations use single-attribute primary keys. Therefore, no partial dependency exists.

The Conclusion is all relations satisfy **2NF**.

◆ Third Normal Form (3NF):

Definition:

A relation is in Third Normal Form (3NF) if:

1. It is already in 2NF.
2. There is no transitive dependency among non-key attributes.

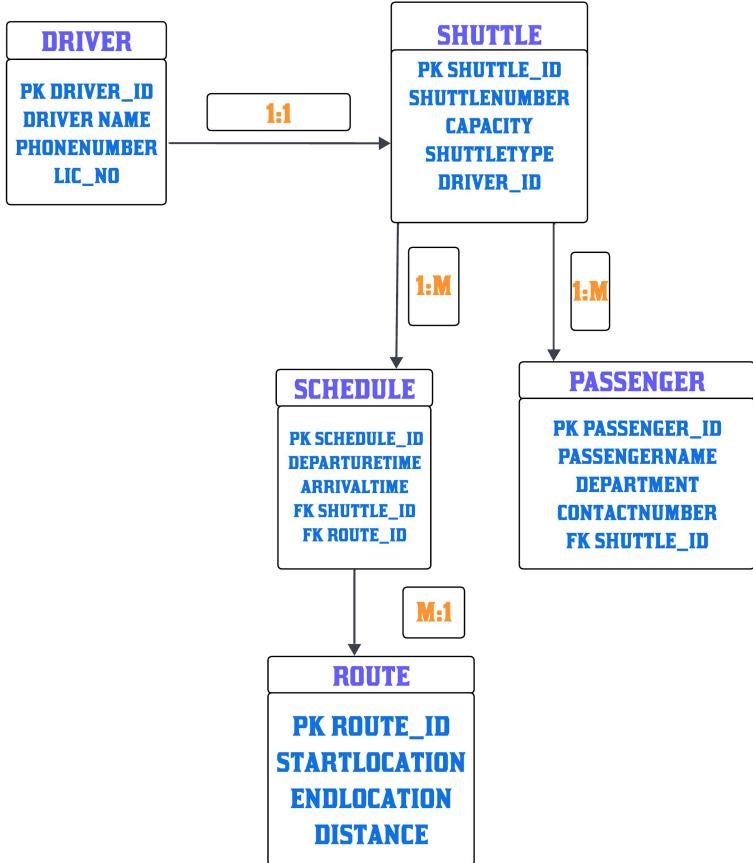
Application in Shuttle Management System:

All non-key attributes in each table depend only on their respective primary keys. No transitive dependency is found.

The Conclusion is all relations satisfy **3NF**.

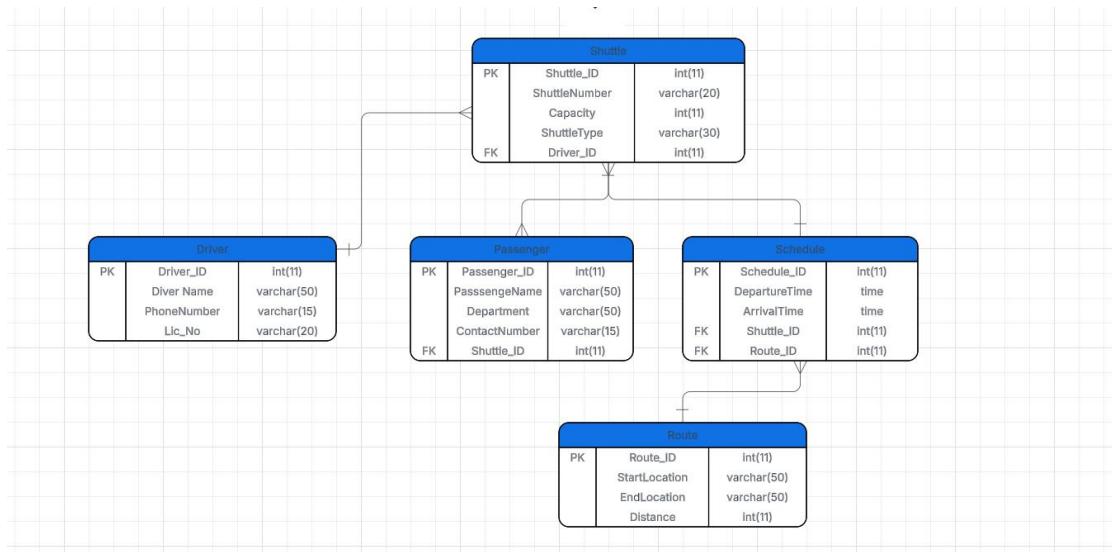
◆ Final Relational Schema:

So the Relational Schema remain same after Normalization.



● Database Schema

The database schema defines the physical structure of the database using SQL. It includes table definitions, data types, primary keys, and foreign key constraints. The schema is implemented using SQL CREATE DATABASE and CREATE TABLE statements for the Shuttle Management System.



- **Database Implementation (SQL Scripts):**

- CREATE DATABASE:

Input:

The screenshot shows a MySQL query editor window. The title bar says "Run SQL query/queries on server “127.0.0.1”". The main area contains the following SQL code:

```
1 CREATE DATABASE shuttledb;
2 USE shuttledb;
```

Output:

The screenshot shows the MySQL Workbench interface. The title bar indicates "Server: 127.0.0.1" and "Database: shuttledb". The toolbar includes "Structure", "SQL", "Search", "Query", "Export", "Import", and "Operations". A message bar at the top says "No tables found in database." Below the toolbar, there is a "Create new table" button. The "Table name" field is empty, and the "Number of columns" field contains the value "4". A "Create" button is visible.

- CREATE TABLE Driver:

Input:

Run SQL query/queries on database **shuttledb**: 

```
1 CREATE TABLE Driver (
2     Driver_ID INT PRIMARY KEY,
3     DriverName VARCHAR(50),
4     PhoneNumber VARCHAR(15),
5     Lic_no VARCHAR(20)
6 );
```

Output:

Show query box

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0009 seconds.)

```
CREATE TABLE Driver ( Driver_ID INT PRIMARY KEY, DriverName VARCHAR(50), PhoneNumber VARCHAR(15), Lic_no VARCHAR(20) );
```

[Edit inline] [Edit] [Create PHP code]

● CREATE TABLE Route:

Input:

Run SQL query/queries on database **shuttledb**: 

```
1 CREATE TABLE Route (
2     Route_ID INT PRIMARY KEY,
3     StartLocation VARCHAR(50),
4     EndLocation VARCHAR(50),
5     Distance INT
6 );
```

Output:

Show query box

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0005 seconds.)

```
CREATE TABLE Route ( Route_ID INT PRIMARY KEY, StartLocation VARCHAR(50), EndLocation VARCHAR(50), Distance INT );
```

[Edit inline] [Edit] [Create PHP code]

● CREATE TABLE Shuttle:

Input:

Run SQL query/queries on database **shuttledb**: 

```
1 CREATE TABLE Shuttle (
2     Shuttle_ID INT PRIMARY KEY,
3     ShuttleNumber VARCHAR(20),
4     Capacity INT,
5     ShuttleType VARCHAR(30),
6     Driver_ID INT,
7     FOREIGN KEY (Driver_ID) REFERENCES Driver(Driver_ID)
8 );
```

Output:

Show query box

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0008 seconds.)

```
CREATE TABLE Shuttle ( Shuttle_ID INT PRIMARY KEY, ShuttleNumber VARCHAR(20), Capacity INT, ShuttleType VARCHAR(30), Driver_ID INT, FOREIGN KEY (Driver_ID) REFERENCES Driver(Driver_ID) );
```

[Edit inline] [Edit] [Create PHP code]

● CREATE TABLE Schedule:

Input:

Run SQL query/queries on database **shuttledb**: 

```
1 CREATE TABLE Schedule (
2     Schedule_ID INT PRIMARY KEY,
3     DepartureTime TIME,
4     ArrivalTime TIME,
5     Shuttle_ID INT,
6     Route_ID INT,
7     FOREIGN KEY (Shuttle_ID) REFERENCES Shuttle(Shuttle_ID),
8     FOREIGN KEY (Route_ID) REFERENCES Route(Route_ID)
9 );
```

Output:

Show query box

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0006 seconds.)

```
CREATE TABLE Schedule ( Schedule_ID INT PRIMARY KEY, DepartureTime TIME, ArrivalTime TIME, Shuttle_ID INT, Route_ID INT, FOREIGN KEY (Shuttle_ID) REFERENCES Shuttle(Shuttle_ID), FOREIGN KEY (Route_ID) REFERENCES Route(Route_ID) );
```

[Edit inline] [Edit] [Create PHP code]

● CREATE TABLE Passenger:

Input:

Run SQL query/queries on database **shuttledb**: 

```
1 CREATE TABLE Passenger (
2     Passenger_ID INT PRIMARY KEY,
3     PassengerName VARCHAR(50),
4     Department VARCHAR(50),
5     ContactNumber VARCHAR(15),
6     Shuttle_ID INT,
7     FOREIGN KEY (Shuttle_ID) REFERENCES Shuttle(Shuttle_ID)
8 );
```

Output:

Show query box

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0004 seconds.)

```
CREATE TABLE Passenger ( Passenger_ID INT PRIMARY KEY, PassengerName VARCHAR(50), Department VARCHAR(50), ContactNumber VARCHAR(15), Shuttle_ID INT, FOREIGN KEY (Shuttle_ID) REFERENCES Shuttle(Shuttle_ID) );
```

[Edit inline] [Edit] [Create PHP code]

- INSERT:

INSERT INTO Driver VALUES:

Input:

Run SQL query/queries on table **shuttledb.driver**: 

```
1 INSERT INTO Driver VALUES
2 (1,'Rahim','01711111111','LIC101'),
3 (2,'Karim','01722222222','LIC102'),
4 (3,'Salam','01733333333','LIC103'),
5 (4,'Jamal','01744444444','LIC104'),
6 (5,'Kamal','01755555555','LIC105'),
7 (6,'Babul','01766666666','LIC106'),
8 (7,'Rasel','01777777777','LIC107'),
9 (8,'Imran','01788888888','LIC108'),
10 (9,'Noman','01799999999','LIC109'),
11 (10,'Arif','01800000000','LIC110');
```

Output:

Extra options

	Driver_ID	DriverName	PhoneNumber	Lic_no
<input type="checkbox"/>   	1 Rahim	01711111111	LIC101	
<input type="checkbox"/>   	2 Karim	01722222222	LIC102	
<input type="checkbox"/>   	3 Salam	01733333333	LIC103	
<input type="checkbox"/>   	4 Jamal	01744444444	LIC104	
<input type="checkbox"/>   	5 Kamal	01755555555	LIC105	
<input type="checkbox"/>   	6 Babul	01766666666	LIC106	
<input type="checkbox"/>   	7 Rasel	01777777777	LIC107	
<input type="checkbox"/>   	8 Imran	01788888888	LIC108	
<input type="checkbox"/>   	9 Noman	01799999999	LIC109	
<input type="checkbox"/>   	10 Arif	01800000000	LIC110	

 Check all With selected:    

- INSERT INTO Route VALUES:

Input:

Run SQL query/queries on table **shuttledb.route**: 

```
1 INSERT INTO Route VALUES
2 (1,'Campus','Dhanmondi',12),
3 (2,'Campus','Uttara',18),
4 (3,'Campus','Mirpur',15),
5 (4,'Campus','Gulshan',14),
6 (5,'Campus','Banani',10),
7 (6,'Campus','Motijheel',20),
8 (7,'Campus','Mohakhali',11),
9 (8,'Campus','Farmgate',9),
10 (9,'Campus','Tejgaon',13),
11 (10,'Campus','Jatrabari',22);
```

Output:

The screenshot shows the MySQL Workbench interface with the database 'shuttledb' and table 'route'. The table has columns: Route_ID, StartLocation, EndLocation, and Distance. The data consists of 10 rows, each representing a route from a campus location to another, with distances ranging from 9 to 22.

	Route_ID	StartLocation	EndLocation	Distance
<input type="checkbox"/>	1	Campus	Dhanmondi	12
<input type="checkbox"/>	2	Campus	Uttara	18
<input type="checkbox"/>	3	Campus	Mirpur	15
<input type="checkbox"/>	4	Campus	Gulshan	14
<input type="checkbox"/>	5	Campus	Banani	10
<input type="checkbox"/>	6	Campus	Motijheel	20
<input type="checkbox"/>	7	Campus	Mohakhali	11
<input type="checkbox"/>	8	Campus	Farmgate	9
<input type="checkbox"/>	9	Campus	Tejgaon	13
<input type="checkbox"/>	10	Campus	Jatrabari	22

● INSERT INTO Shuttle VALUES:

Input:

Run SQL query/queries on table **shuttledb.shuttle**:

```
1 INSERT INTO Shuttle VALUES
2 (1,'S-01',40,'AC',1),
3 (2,'S-02',35,'Non-AC',2),
4 (3,'S-03',45,'AC',3),
5 (4,'S-04',30,'Non-AC',4),
6 (5,'S-05',40,'AC',5),
7 (6,'S-06',35,'Non-AC',6),
8 (7,'S-07',45,'AC',7),
9 (8,'S-08',30,'Non-AC',8),
10 (9,'S-09',40,'AC',9),
11 (10,'S-10',35,'Non-AC',10);
```

Output:

The screenshot shows the MySQL Workbench interface with the database 'shuttledb' and table 'shuttle'. The table has columns: Shuttle_ID, ShuttleNumber, Capacity, ShuttleType, and Driver_ID. The data consists of 10 rows, each representing a shuttle with its details, matching the inserted values.

	Shuttle_ID	ShuttleNumber	Capacity	ShuttleType	Driver_ID
<input type="checkbox"/>	1	S-01	40	AC	1
<input type="checkbox"/>	2	S-02	35	Non-AC	2
<input type="checkbox"/>	3	S-03	45	AC	3
<input type="checkbox"/>	4	S-04	30	Non-AC	4
<input type="checkbox"/>	5	S-05	40	AC	5
<input type="checkbox"/>	6	S-06	35	Non-AC	6
<input type="checkbox"/>	7	S-07	45	AC	7
<input type="checkbox"/>	8	S-08	30	Non-AC	8
<input type="checkbox"/>	9	S-09	40	AC	9
<input type="checkbox"/>	10	S-10	35	Non-AC	10

- INSERT INTO Schedule VALUES:

Input:

Run SQL query/queries on table **shuttledb.schedule**: 

```
1 INSERT INTO Schedule VALUES
2 (1, '08:00', '09:00', 1, 1),
3 (2, '09:00', '10:00', 2, 2),
4 (3, '10:00', '11:00', 3, 3),
5 (4, '11:00', '12:00', 4, 4),
6 (5, '12:00', '13:00', 5, 5),
7 (6, '13:00', '14:00', 6, 6),
8 (7, '14:00', '15:00', 7, 7),
9 (8, '15:00', '16:00', 8, 8),
10 (9, '16:00', '17:00', 9, 9),
11 (10, '17:00', '18:00', 10, 10);
```

Output:



The screenshot shows the MySQL Workbench interface with the following details:

- Server: 127.0.0.1
- Database: shuttledb
- Table: schedule
- Toolbar buttons: Browse, Structure, SQL, Search, Insert, Export, Import, Privilege.
- Status bar message: Showing rows 0 - 9 (10 total, Query took 0.0004 seconds.)
- SQL tab content:

```
SELECT * FROM `schedule`
```
- Table data view:

	<input type="checkbox"/>	Schedule_ID	DepartureTime	ArrivalTime	Shuttle_ID	Route_ID
	<input type="checkbox"/>	1	08:00:00	09:00:00	1	1
	<input type="checkbox"/>	2	09:00:00	10:00:00	2	2
	<input type="checkbox"/>	3	10:00:00	11:00:00	3	3
	<input type="checkbox"/>	4	11:00:00	12:00:00	4	4
	<input type="checkbox"/>	5	12:00:00	13:00:00	5	5
	<input type="checkbox"/>	6	13:00:00	14:00:00	6	6
	<input type="checkbox"/>	7	14:00:00	15:00:00	7	7
	<input type="checkbox"/>	8	15:00:00	16:00:00	8	8
	<input type="checkbox"/>	9	16:00:00	17:00:00	9	9
	<input type="checkbox"/>	10	17:00:00	18:00:00	10	10

- INSERT INTO Passenger VALUES:

Input:

Run SQL query/queries on table **shuttledb.passenger**: [?](#)

```
1 INSERT INTO Passenger VALUES
2 (1,'Arafat','CSE','01911111111',1),
3 (2,'Nila','EEE','01922222222',2),
4 (3,'Sumon','BBA','01933333333',3),
5 (4,'Rina','CSE','01944444444',4),
6 (5,'Jamil','EEE','01955555555',5),
7 (6,'Karim','BBA','01966666666',6),
8 (7,'Sadia','CSE','01977777777',7),
9 (8,'Tanim','EEE','01988888888',8),
10 (9,'Rafi','BBA','01999999999',9),
11 (10,'Mitu','CSE','01811111111',10);
```

Output:

The screenshot shows the MySQL Workbench interface with the following details:

- Server: 127.0.0.1
- Database: shuttledb
- Table: passenger
- Toolbar buttons: Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Oper.
- Message bar: ✓ Showing rows 0 - 9 (10 total, Query took 0.0008 seconds.)
- SQL pane: SELECT * FROM `passenger`
- Table view:

	Passenger_ID	PassengerName	Department	ContactNumber	Shuttle_ID
<input type="checkbox"/>	1	Arafat	CSE	01911111111	1
<input type="checkbox"/>	2	Nila	EEE	01922222222	2
<input type="checkbox"/>	3	Sumon	BBA	01933333333	3
<input type="checkbox"/>	4	Rina	CSE	01944444444	4
<input type="checkbox"/>	5	Jamil	EEE	01955555555	5
<input type="checkbox"/>	6	Karim	BBA	01966666666	6
<input type="checkbox"/>	7	Sadia	CSE	01977777777	7
<input type="checkbox"/>	8	Tanim	EEE	01988888888	8
<input type="checkbox"/>	9	Rafi	BBA	01999999999	9
<input type="checkbox"/>	10	Mitu	CSE	01811111111	10

■ SQL Queries:

1. Show all drivers:

Input:

Run SQL query/queries on table **shuttledb.driver**: [?](#)

```
1 SELECT * FROM Driver;
```

Output:

Server: 127.0.0.1 » Database: shuttledb » Table: Driver

Browse Structure SQL Search Insert Export Import

Showing rows 0 - 9 (10 total, Query took 0.0005 seconds.)

SELECT * FROM `Driver` WHERE 1;

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this table Sort by

Extra options

		Driver_ID	DriverName	PhoneNumber	Lic_no
<input type="checkbox"/>	Edit	Copy	Delete	1 Rahim	01711111111 LIC101
<input type="checkbox"/>	Edit	Copy	Delete	2 Karim	01722222222 LIC102
<input type="checkbox"/>	Edit	Copy	Delete	3 Salam	01733333333 LIC103
<input type="checkbox"/>	Edit	Copy	Delete	4 Jamal	01744444444 LIC104
<input type="checkbox"/>	Edit	Copy	Delete	5 Kamal	01755555555 LIC105
<input type="checkbox"/>	Edit	Copy	Delete	6 Babul	01766666666 LIC106
<input type="checkbox"/>	Edit	Copy	Delete	7 Rasel	01777777777 LIC107
<input type="checkbox"/>	Edit	Copy	Delete	8 Imran	01788888888 LIC108
<input type="checkbox"/>	Edit	Copy	Delete	9 Noman	01799999999 LIC109
<input type="checkbox"/>	Edit	Copy	Delete	10 Arif	01800000000 LIC110

Check all With selected: Edit Copy Delete Export

2. Show all shuttles

Input:

Run SQL query/queries on table shuttledb.Shuttle:

```
1 SELECT * FROM Shuttle;
```

Output:

Server: 127.0.0.1 » Database: shuttledb » Table: Shuttle

Browse Structure SQL Search Insert Export Import Privileges Options

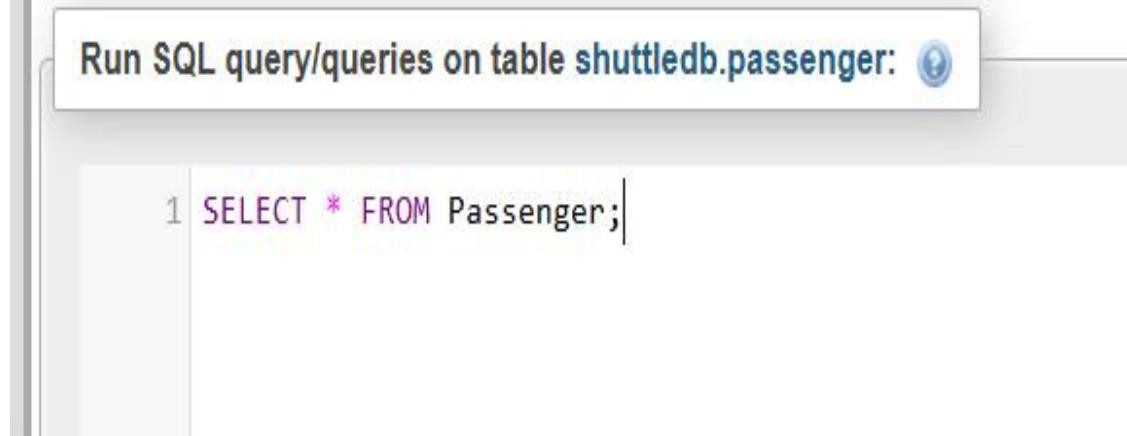
Extra options

		Shuttle_ID	ShuttleNumber	Capacity	ShuttleType	Driver_ID
<input type="checkbox"/>	Edit	Copy	Delete	1 S-01	40 AC	1
<input type="checkbox"/>	Edit	Copy	Delete	2 S-02	35 Non-AC	2
<input type="checkbox"/>	Edit	Copy	Delete	3 S-03	45 AC	3
<input type="checkbox"/>	Edit	Copy	Delete	4 S-04	30 Non-AC	4
<input type="checkbox"/>	Edit	Copy	Delete	5 S-05	40 AC	5
<input type="checkbox"/>	Edit	Copy	Delete	6 S-06	35 Non-AC	6
<input type="checkbox"/>	Edit	Copy	Delete	7 S-07	45 AC	7
<input type="checkbox"/>	Edit	Copy	Delete	8 S-08	30 Non-AC	8
<input type="checkbox"/>	Edit	Copy	Delete	9 S-09	40 AC	9
<input type="checkbox"/>	Edit	Copy	Delete	10 S-10	35 Non-AC	10

Check all With selected: Edit Copy Delete Export

3. Show all passengers

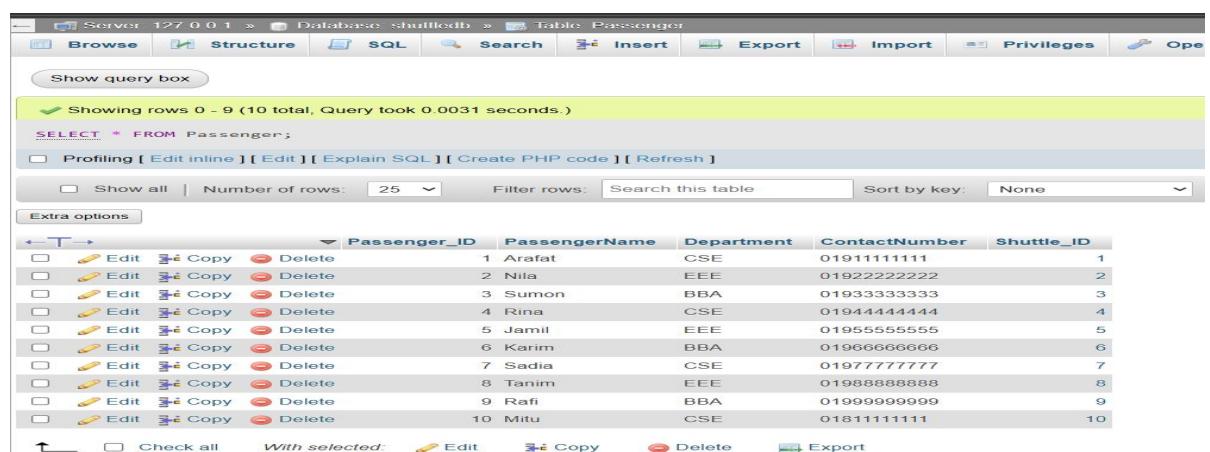
Input:



Run SQL query/queries on table **shuttledb.passenger**: 

```
1 SELECT * FROM Passenger;
```

Output:

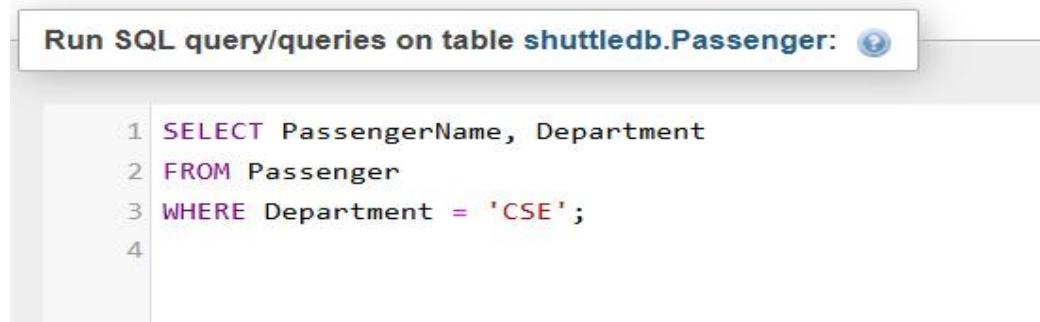


Showing rows 0 - 9 (10 total, Query took 0.0031 seconds.)

	Passenger_ID	PassengerName	Department	ContactNumber	Shuttle_ID
<input type="checkbox"/>	1	Arafat	CSE	01911111111	1
<input type="checkbox"/>	2	Nila	EEE	01922222222	2
<input type="checkbox"/>	3	Sumon	BBA	01933333333	3
<input type="checkbox"/>	4	Rina	CSE	01944444444	4
<input type="checkbox"/>	5	Jamil	EEE	01955555555	5
<input type="checkbox"/>	6	Karim	BBA	01966666666	6
<input type="checkbox"/>	7	Sadia	CSE	01977777777	7
<input type="checkbox"/>	8	Tanim	EEE	01988888888	8
<input type="checkbox"/>	9	Rafi	BBA	01999999999	9
<input type="checkbox"/>	10	Mitu	CSE	01811111111	10

4.Find passengers from CSE department

Input:



Run SQL query/queries on table **shuttledb.Passenger**: 

```
1 SELECT PassengerName, Department
2 FROM Passenger
3 WHERE Department = 'CSE';
4
```

Output:

The screenshot shows the MySQL Workbench interface with the following details:

- Server:** 127.0.0.1
- Database:** shuttledb
- Table:** Passenger
- Query Result:**

```
SELECT PassengerName, Department FROM Passenger WHERE Department = 'CSE';
+-----+-----+
| PassengerName | Department |
+-----+-----+
| Arafat      | CSE       |
| Rina        | CSE       |
| Sadia       | CSE       |
| Mitu        | CSE       |
+-----+-----+
```
- Extra Options:** Includes buttons for Edit, Copy, Delete, and Export.

5.Find shuttles with capacity greater than 40

Input:

The screenshot shows the MySQL Workbench interface with the following details:

- Run SQL query/queries on table shuttledb.passenger:**
- Query:**

```
1 SELECT ShuttleNumber, Capacity
2 FROM Shuttle
3 WHERE Capacity > 40;
```

Output:

The screenshot shows the MySQL Workbench interface with the following details:

- Server:** 127.0.0.1
- Database:** shuttledb
- Table:** Shuttle
- Query Result:**

```
SELECT ShuttleNumber, Capacity FROM Shuttle WHERE Capacity > 40;
+-----+-----+
| ShuttleNumber | Capacity |
+-----+-----+
| S-03          | 45       |
| S-07          | 45       |
+-----+-----+
```
- Extra Options:** Includes buttons for Edit, Copy, Delete, and Export.

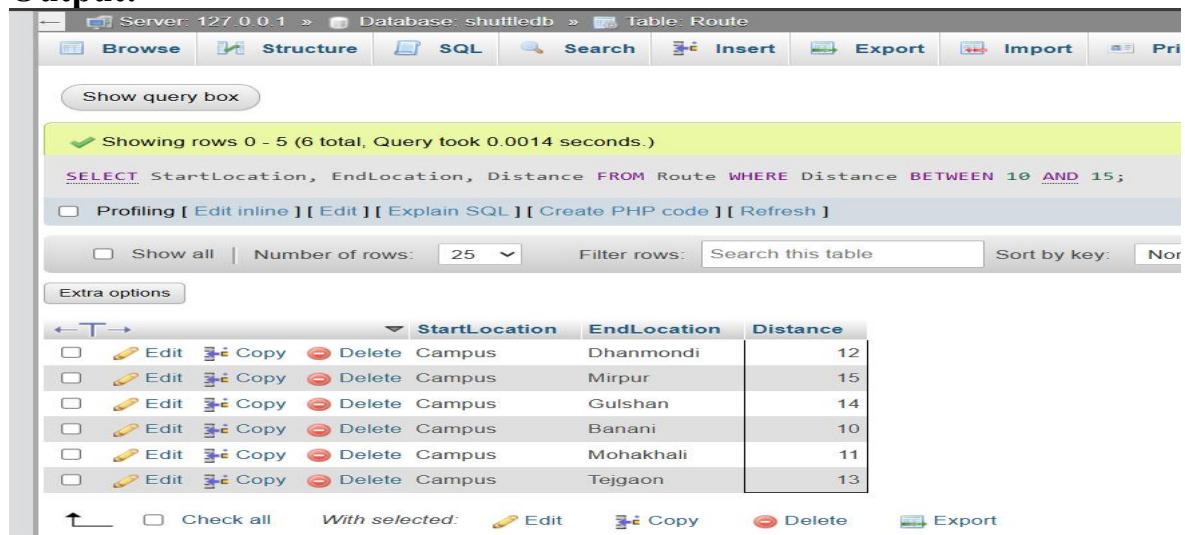
6.Find routes with distance between 10 and 15

Input:

Run SQL query/queries on table **shuttledb.route**: 

```
1 SELECT StartLocation, EndLocation, Distance
2 FROM Route
3 WHERE Distance BETWEEN 10 AND 15;
```

Output:



The screenshot shows the phpMyAdmin interface for the 'Route' table. The results pane displays the following data:

	StartLocation	EndLocation	Distance
1	Campus	Dhanmondi	12
2	Campus	Mirpur	15
3	Campus	Gulshan	14
4	Campus	Banani	10
5	Campus	Mohakhali	11
6	Campus	Tejgaon	13

7. Find passengers whose name starts with 'A'

Input:

Run SQL query/queries on table **shuttledb.passenger**: 

```
1 SELECT PassengerName
2 FROM Passenger
3 WHERE PassengerName LIKE 'A%';
```

Output:

The screenshot shows the phpMyAdmin interface for a database named 'shuttledb'. The current table is 'Passenger'. The search query is 'SELECT PassengerName FROM Passenger WHERE PassengerName LIKE 'A%';'. The results show one row: 'Arafat'. There are buttons for 'Edit', 'Copy', and 'Delete' for this row.

8. Sort shuttles by capacity (descending)

Input:

The screenshot shows the phpMyAdmin interface with the following SQL query entered into the query box:

```
1 SELECT ShuttleNumber, Capacity
2 FROM Shuttle
3 ORDER BY Capacity DESC;
```

Output:

The screenshot shows the phpMyAdmin interface displaying the results of the previous SQL query. The shuttles are listed in descending order of capacity:

ShuttleNumber	Capacity
S-03	45
S-07	45
S-01	40
S-09	40
S-05	40
S-06	35
S-02	35
S-10	35
S-04	30
S-08	30

9. Count total number of passengers:

Input:

Run SQL query/queries on table shuttledb.passenger:

```
1 SELECT COUNT(Passenger_ID) AS Total_Passengers
2 FROM Passenger;
```

Output:

The screenshot shows the MySQL Workbench interface with the following details:

- Server: 127.0.0.1
- Database: shuttledb
- Table: Shuttle
- Toolbar buttons: Browse, Structure, SQL, Search, Insert, Export, Import.
- Query results:
 - Showing rows 0 - 9 (10 total, Query took 0.0009 seconds.) [Capacity: 45... - 30...]
 - SELECT ShuttleNumber, Capacity FROM Shuttle ORDER BY Capacity DESC;
 - Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
- Table view:
 - Extra options: Show all, Number of rows: 25, Filter rows: Search this table, Sort by key:.
 - Table headers: ShuttleNumber, Capacity.
 - Data rows:

ShuttleNumber	Capacity
S-03	45
S-07	45
S-01	40
S-09	40
S-05	40
S-06	35
S-02	35
S-10	35
S-04	30
S-08	30

10. Find average shuttle capacity

Input:

Run SQL query/queries on table shuttledb.shuttle:

```
1 SELECT AVG(Capacity) AS Avg_Capacity
2 FROM Shuttle;
```

Output:

Server: 127.0.0.1 » Database: shuttledb » Table: Shuttle

Browse Structure SQL Search Insert Export Import Privileges

Show query box

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available

Showing rows 0 - 0 (1 total, Query took 0.0019 seconds.)

```
SELECT AVG(Capacity) AS Avg_Capacity FROM Shuttle;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this table

Extra options

Avg_Capacity
37.5000

11. Find maximum and minimum route distance

Input:

Run SQL query/queries on table shuttledb.route:

```
1 SELECT MAX(Distance) AS Max_Distance,
2 MIN(Distance) AS Min_Distance
3 FROM Route;
```

Output:

Server: 127.0.0.1 » Database: shuttledb » Table: Route

Browse Structure SQL Search Insert Export Import Privileges

Show query box

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 0 (1 total, Query took 0.0009 seconds.)

```
SELECT MAX(Distance) AS Max_Distance, MIN(Distance) AS Min_Distance FROM Route;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this table

Extra options

Max_Distance	Min_Distance
22	9

12.Count passengers per department:

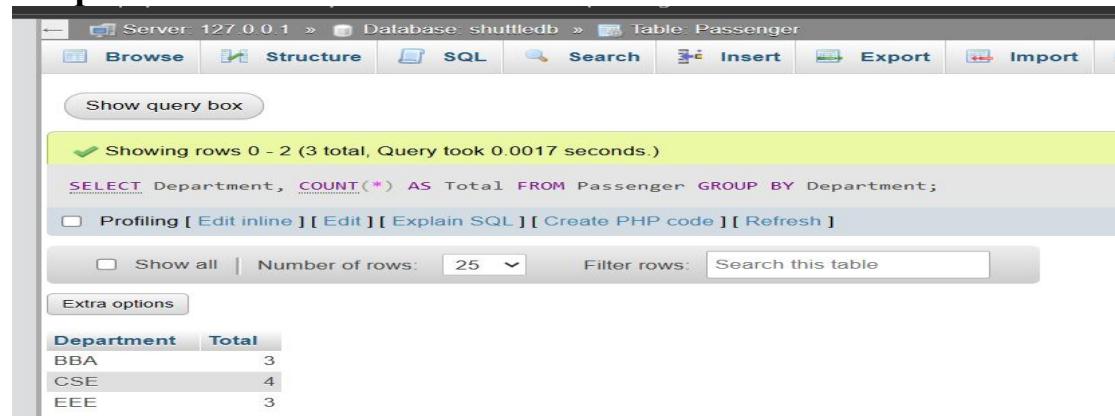
Input:



Run SQL query/queries on table **shuttledb.passenger**: 

```
1 SELECT Department, COUNT(*) AS Total
2 FROM Passenger|
3 GROUP BY Department
```

Output:



Server: 127.0.0.1 » Database: shuttledb » Table: Passenger

Browse Structure SQL Search Insert Export Import

Show query box

Showing rows 0 - 2 (3 total, Query took 0.0017 seconds.)

```
SELECT Department, COUNT(*) AS Total FROM Passenger GROUP BY Department;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

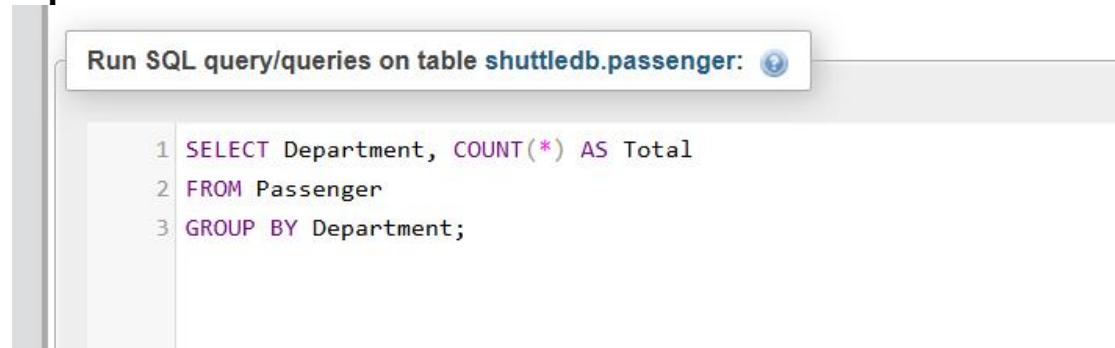
Show all Number of rows: 25 Filter rows: Search this table

Extra options

Department	Total
BBA	3
CSE	4
EEE	3

13.Count passengers per department

Input:



Run SQL query/queries on table **shuttledb.passenger**: 

```
1 SELECT Department, COUNT(*) AS Total
2 FROM Passenger
3 GROUP BY Department;
```

Output:



Server: 127.0.0.1 » Database: shuttledb » Table: Passenger

Browse Structure SQL Search Insert Export Import

Show query box

Showing rows 0 - 2 (3 total, Query took 0.0004 seconds.)

```
SELECT Department, COUNT(*) AS Total FROM Passenger GROUP BY Department;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

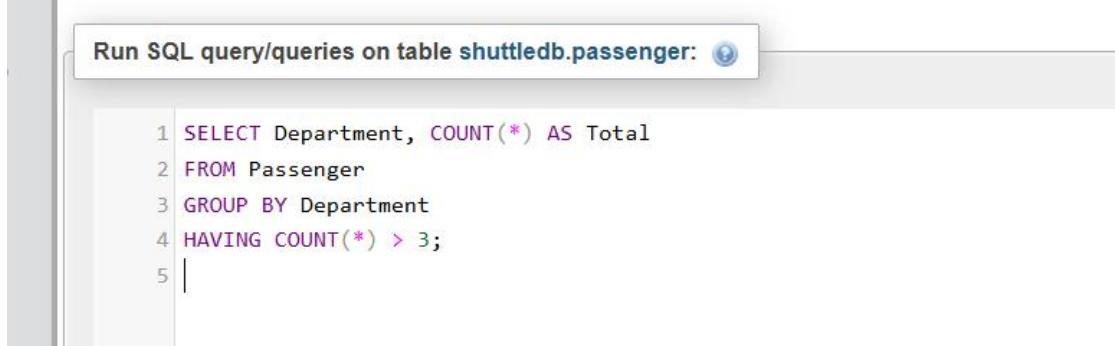
Show all Number of rows: 25 Filter rows: Search this table

Extra options

Department	Total
BBA	3
CSE	4
EEE	3

14. Show departments having more than 3 passengers

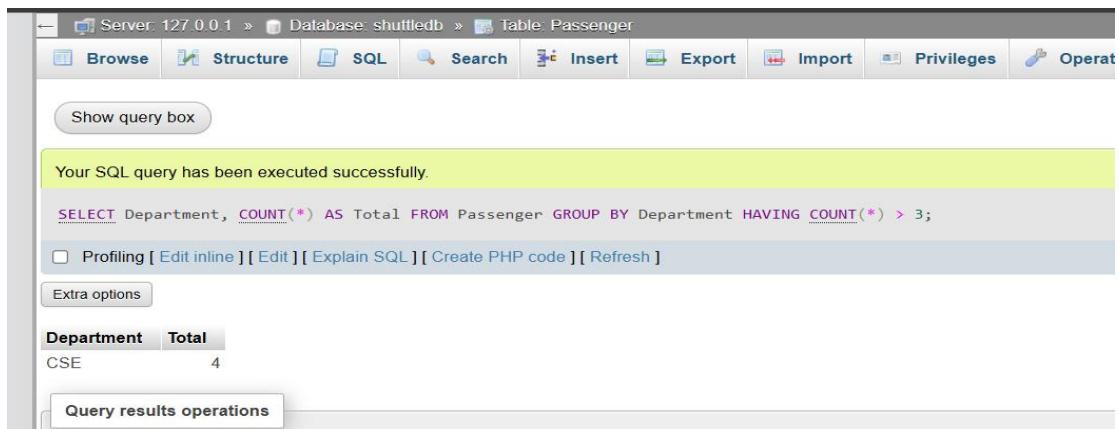
Input:



Run SQL query/queries on table **shuttledb.passenger**: [?](#)

```
1 SELECT Department, COUNT(*) AS Total
2 FROM Passenger
3 GROUP BY Department
4 HAVING COUNT(*) > 3;
5 |
```

Output:



Your SQL query has been executed successfully.

```
SELECT Department, COUNT(*) AS Total FROM Passenger GROUP BY Department HAVING COUNT(*) > 3;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Extra options

Department	Total
CSE	4

Query results operations

15. Find passengers who travel in AC shuttles

Input:



Run SQL query/queries on table **shuttledb.shuttle**: [?](#)

```
1 SELECT PassengerName
2 FROM Passenger
3 WHERE Shuttle_ID IN (
4     SELECT Shuttle_ID
5     FROM Shuttle
6     WHERE ShuttleType = 'AC'
7 );
8
```

Output:

The screenshot shows the phpMyAdmin interface for a database named 'shuttledb'. The current table is 'Passenger'. A query has been run:

```
SELECT PassengerName FROM Passenger WHERE Shuttle_ID IN ( SELECT Shuttle_ID FROM Shuttle WHERE ShuttleType = 'AC' );
```

The results show five rows of passenger names: Arafat, Sumon, Jamil, Sadia, and Rafi. Each row has options to edit, copy, or delete.

15. Find shuttles having capacity greater than average capacity

Input:

The screenshot shows a MySQL command-line interface window. The query entered is:

```
1 SELECT ShuttleNumber, Capacity
2 FROM Shuttle
3 WHERE Capacity > (
4     SELECT AVG(Capacity)
5     FROM Shuttle
6 );
7
```

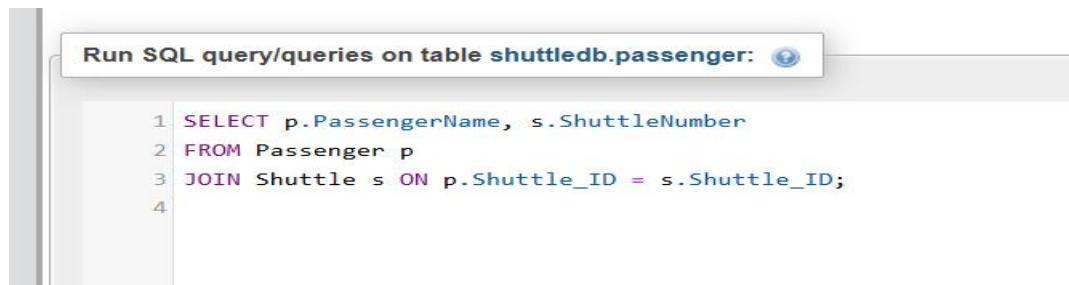
Output:

The screenshot shows the phpMyAdmin interface for the 'Shuttle' table. The query ran successfully, showing 5 rows of shuttle numbers and capacities. The shuttle with ID S-03 has the highest capacity at 45.

ShuttleNumber	Capacity
S-01	40
S-03	45
S-05	40
S-07	45
S-09	40

16. Show passenger names with their shuttle numbers

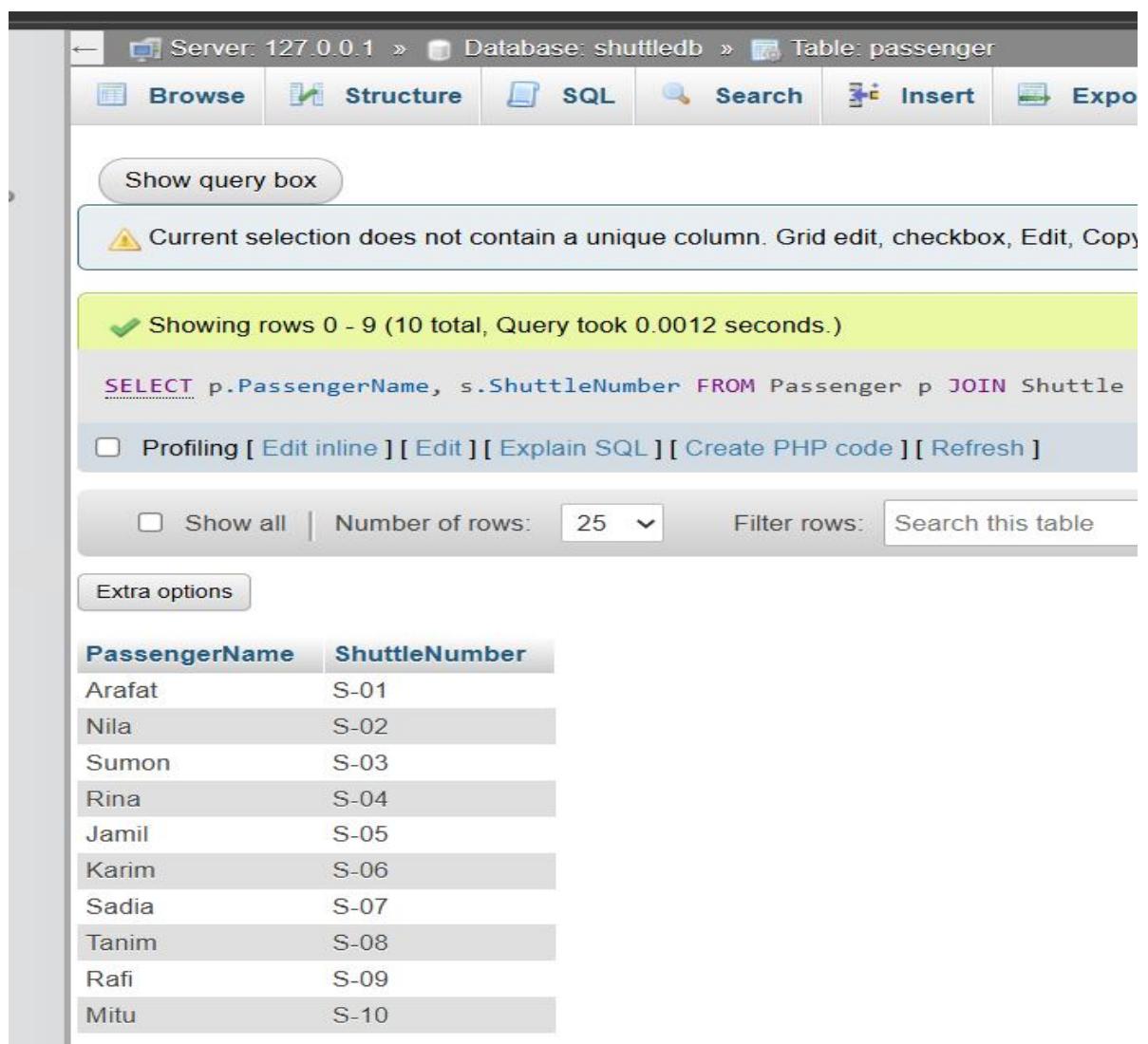
Input: Inner Join



The screenshot shows the MySQL Workbench interface with the SQL tab selected. A query is being run against the 'shuttledb.passenger' table:

```
1 | SELECT p.PassengerName, s.ShuttleNumber
2 | FROM Passenger p
3 | JOIN Shuttle s ON p.Shuttle_ID = s.Shuttle_ID;
4 |
```

Output:



The screenshot shows the MySQL Workbench interface with the Browse tab selected, displaying the results of the executed SQL query. The results are presented in a grid format:

PassengerName	ShuttleNumber
Arafat	S-01
Nila	S-02
Sumon	S-03
Rina	S-04
Jamil	S-05
Karim	S-06
Sadia	S-07
Tanim	S-08
Rafi	S-09
Mitu	S-10

17. Show drivers with their assigned shuttle numbers

Input: Inner Join

Run SQL query/queries on table **shuttledb.shuttle**:

```
1 SELECT d.DriverName, s.ShuttleNumber
2 FROM Driver d
3 JOIN Shuttle s ON d.Driver_ID = s.Driver_ID;
4
```

Output:

Server: 127.0.0.1 » Database: shuttledb » Table: shuttle

Browse Structure SQL Search Insert Export Import

Show query box

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete

Showing rows 0 - 9 (10 total, Query took 0.0005 seconds.)

```
SELECT d.DriverName, s.ShuttleNumber FROM Driver d JOIN Shuttle s ON d.Driver_ID = s.Driver_ID
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this table Sort

Extra options

DriverName	ShuttleNumber
Rahim	S-01
Karim	S-02
Salam	S-03
Jamal	S-04
Kamal	S-05
Babul	S-06
Rasel	S-07
Imran	S-08
Noman	S-09
Arif	S-10

18. Show shuttle number and route details

Input: Join

Run SQL query/queries on table **shuttledb.schedule**:

```
1 SELECT s.ShuttleNumber, r.StartLocation, r.EndLocation
2 FROM Schedule sc
3 JOIN Shuttle s ON sc.Shuttle_ID = s.Shuttle_ID
4 JOIN Route r ON sc.Route_ID = r.Route_ID;
5
```

Output:

min/index.php?route=/table/sql&db=shuttledb&table=schedule

← Server: 127.0.0.1 » Database: shuttledb » Table: schedule

Browse Structure SQL Search Insert

Show query box

⚠ Current selection does not contain a unique column. Grid edit, checkbox,

Showing rows 0 - 9 (10 total, Query took 0.0018 seconds.)

```
SELECT s.ShuttleNumber, r.StartLocation, r.EndLocation FROM S
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this table

Extra options

ShuttleNumber	StartLocation	EndLocation
S-01	Campus	Dhanmondi
S-02	Campus	Uttara
S-03	Campus	Mirpur
S-04	Campus	Gulshan
S-05	Campus	Banani
S-06	Campus	Motijheel
S-07	Campus	Mohakhali
S-08	Campus	Farmgate
S-09	Campus	Tejgaon
S-10	Campus	Jatrabari

19. Find total number of schedules per shuttle

Input:

BROWSE STRUCTURE SQL Search Insert EXPORT IMPORT

Run SQL query/queries on table shuttledb.schedule: ⚡

```
1 SELECT Shuttle_ID, COUNT(*) AS Total_Schedules
2 FROM Schedule
3 GROUP BY Shuttle_ID;
4 |
```

Output:

The screenshot shows the phpMyAdmin interface for the 'Schedule' table in the 'shuttledb' database. The table has two columns: 'Shuttle_ID' and 'Total_Schedules'. All 10 rows show a value of 1 for 'Total_Schedules'.

Shuttle_ID	Total_Schedules
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1

20. Find passengers who use shuttle ID 5

Input:

The screenshot shows the SQL query editor with the following query:

```
1 SELECT PassengerName
2 FROM Passenger
3 WHERE Shuttle_ID = 5;
```

Output:

The screenshot shows the 'Passenger' table in the 'shuttledb' database. It contains one row with the name 'Jamil'. The table has a single column 'PassengerName'.

PassengerName
Jamil

● Conclusion and Future Advancement:

In this project, we designed and implemented a Shuttle Management System database using the concepts learned in the Database Design Lab. The project included requirement analysis, ER diagram creation, and conversion into relational and database schema. SQL was used to create tables, insert data, and execute different types of queries such as selection, joins, and aggregation.

Also through this work, we gained practical knowledge of database design, SQL queries, and handling relationships between tables. We also faced some challenges in maintaining foreign key relationships, which helped improve our understanding of database concepts.

The future advancements are,

In the future, this system can be extended by adding features like user authentication, real-time shuttle tracking, and online seat booking. More entities such as Admin and Payment can be included, and a web or mobile interface can be developed to make the system more efficient and user-friendly.

● Contribution of each team member:

Member Name	ID	Contribution
1. Md.Shaurov Hayat Emon	2023100000315	Relational Schema,SQL Table Creation,Insert Data,Screenshots,Creating the full Database
2. Shakila Khan Nawreen	2023100000286	ER Diagram,Normalization,SQL Queries,Report Writing and Formatting
3. Shoaeb Islam	2023100000120	Database Schema,Finalizing Relational Schema,Conceptualizing full project