



Mini Project

Private Bus Company Management System

A report submitted to the

Department of Electrical and Information Engineering
Faculty of Engineering
University of Ruhuna Sri Lanka

On 9th of April 2024

In completing an assignment for the module

EE4350 Data Base systems

By:

Group 42

EG/2021/4661	MADURANGANEE J.A.S.M
EG/2021/4662	MADUSANKA W.G.I.S
EG/2021/4664	MADUWANTHA S.A.D.N.

Contents

Chapter 1 - Requirement Analysis	3
1.1. Functional requirements	4
1.2. Data Requirements	4
Chapter 2 - Conceptual Design	7
Chapter 3 - Implementation	8
3.1. Table implementation and creating Foreign key constraint.....	8
3.2. Inserting Data for tables	13
3.3. Update and Delete	14
Chapter 4 - Transactions	31
Chapter 5 - Tuning	39

List of figures

Figure 2-1:ER diagram of the farmhouse database management system 7

Chapter 1 - Requirement Analysis

1.1. FUNCTIONAL REQUIREMENTS

The Private Bus Company Management System schema has been meticulously designed to ensure seamless operations and comprehensive data management. It encompasses a range of functionalities to cater to the diverse needs of the system. The key features of the schema include:

1. Data Retrieval: The system facilitates efficient retrieval of data, enabling users to access the required information promptly.
2. Dynamic Content Alteration: Users have the flexibility to alter the database's content as needed, ensuring adaptability to changing requirements.
3. Data Integrity Measures: The schema is equipped to address potential data gaps or missing information, ensuring completeness and accuracy of the dataset.
4. Data Editing Capabilities: Users can edit existing data entries within the database, allowing for updates and corrections as necessary.
5. Commenting System: The system incorporates a commenting feature, enabling users to add comments to data entries, particularly when information is incomplete or ambiguous, thus enhancing data clarity and context.
6. Enhanced Performance: The schema is optimized for performance, ensuring efficient handling of data operations and delivering an optimal user experience.

The schema encompasses a comprehensive set of entities, ensuring that all essential aspects of the Private Bus Company Management System are represented. These entities include:

- Employee: Represents all personnel associated with the company.
- Manager: Handles managerial responsibilities within the organization.
- Driver: Represents individuals responsible for operating buses.
- Bus: Represents the fleet of buses owned or operated by the company.
- Route: Defines the various routes serviced by the buses.
- Fare: Specifies the fare structure for different routes.
- Supervisor: Oversees operational activities and staff.
- Maintenance: Tracks maintenance activities for the bus fleet.
- Fuel: Manages fuel-related data for the buses.
- Schedule: Represents the timetable for bus services.
- Fuel_cost: Records fuel costs for analysis and budgeting.
- Service_cost: Tracks maintenance and servicing costs.
- Depot: Represents the company's operational base or depot.

By incorporating these entities and functionalities, the schema provides a robust foundation for the Private Bus Company Management System, ensuring efficient operations and effective data management.

1.2. DATA REQUIREMENTS

The attributes of each entity are shown below. These attributes capture the essential information associated with each entity, facilitating effective data management and operations within the Private Bus Company Management System.

Entity	Attributes
Employee	<ul style="list-style-type: none">• Employee_ID• Name - First Name, Last Name• Birthday• Age• Contact_No• Position• Salary• Address – Number, Street, City
Manager	<ul style="list-style-type: none">• Employee_ID• Area• Experience
Driver	<ul style="list-style-type: none">• Employee_ID• Licence_No• Leader_ID• Experience
Bus	<ul style="list-style-type: none">• Bus_ID• Schedule_ID• Deport_ID• Brand• Category• Year• Capacity• Efficiency
Route	<ul style="list-style-type: none">• Route_No• Supervisor_ID• Distance• Start• End• Duration

Fare	<ul style="list-style-type: none"> • Fare_ID • Route_No • Ticket_type • Price_rate
Supervisor	<ul style="list-style-type: none"> • Employee_ID • Supervisor_ID
Maintenance	<ul style="list-style-type: none"> • Service_ID • Bus_No • Last_service • Last_oil_change – Brand, Date • Last_break_test • Last_tyre_change – Brand, Date
Fuel	<ul style="list-style-type: none"> • Bus_ID • Bill_No • Volume • Date
Schedule	<ul style="list-style-type: none"> • Schedule_ID • Route_No • Time – Start, End
Fuel_cost	<ul style="list-style-type: none"> • Amount • Bill_No • Payment_method
Service_cost	<ul style="list-style-type: none"> • Service_ID • Amount • Payment_method
Depot	<ul style="list-style-type: none"> • Deport_ID • Capacity • Contact

And also, we are included some important entities and relationships. They are,

- ✓ Weak entities

There are three weak entities in our ER diagram shown, and we have shown them using a double-line rectangle. Manager is one of them. It depends on the employee entity for its existence and does not possess a key attribute of its own. Therefore, the manager entity is considered weak. Other two weak entities we have considered are the Fuel_cost entity and Service_cost entity. They does not have a key, and they also totally depends on the Fuel entity and Maintenance entity respectively.

- ✓ Recursive relationship

Within the schema, a recursive relationship exists between Leads and Drivers. This relationship signifies that a Leader can leads a team of drivers, while one of those Drivers within the team may also function as a lead.

- ✓ Composite attribute

There are five composite attributes in this schema. Name attribute in employee entity can be subdivided into First_name and Last_name. Address attribute can be subdivided into Number, Street, City. Time attribute in schedule entity can be subdivided into Start_time and End_time. Last_tyre_change attribute and Last_oil_change attributes in Maintenance entity can be subdivided into Brand and Date.

- ✓ Multivalued attribute

Contact_No in Employee entity is a multivalued attribute. Because same employee can have different contact numbers. Therefore it represented by a double line oval in ER diagram.

- ✓ Derived attribute

The Age attribute in Employee entity can be derived from the Birthday attribute, which is stored within the system. By calculating the difference between the current date and the date of birth, the system can automatically determine the age of an individual. Therefore it is a derived attribute. Hence it represented using a dashed oval in an ER diagram.

- ✓ Multiple candidate key

There are two multiple candidate keys in this shema. The Employee_ID attribute serves as a candidate key within the Driver entity. It uniquely identifies each driver and is also utilized as a foreign key referencing the Employee entity. The License_No attribute is another candidate key within the Driver entity. Each driver possesses a unique license number, which serves as an alternative identifier for distinguishing individual drivers. And The Depot_ID attribute serves as a candidate key within the Depot entity. It uniquely identifies each depot and is used to distinguish between different operational bases or depots. The Contact attribute is another candidate key within the Depot entity. It represents the contact information associated with each depot and serves as an alternative identifier for distinguishing individual depots.

Chapter 2 - Conceptual Design

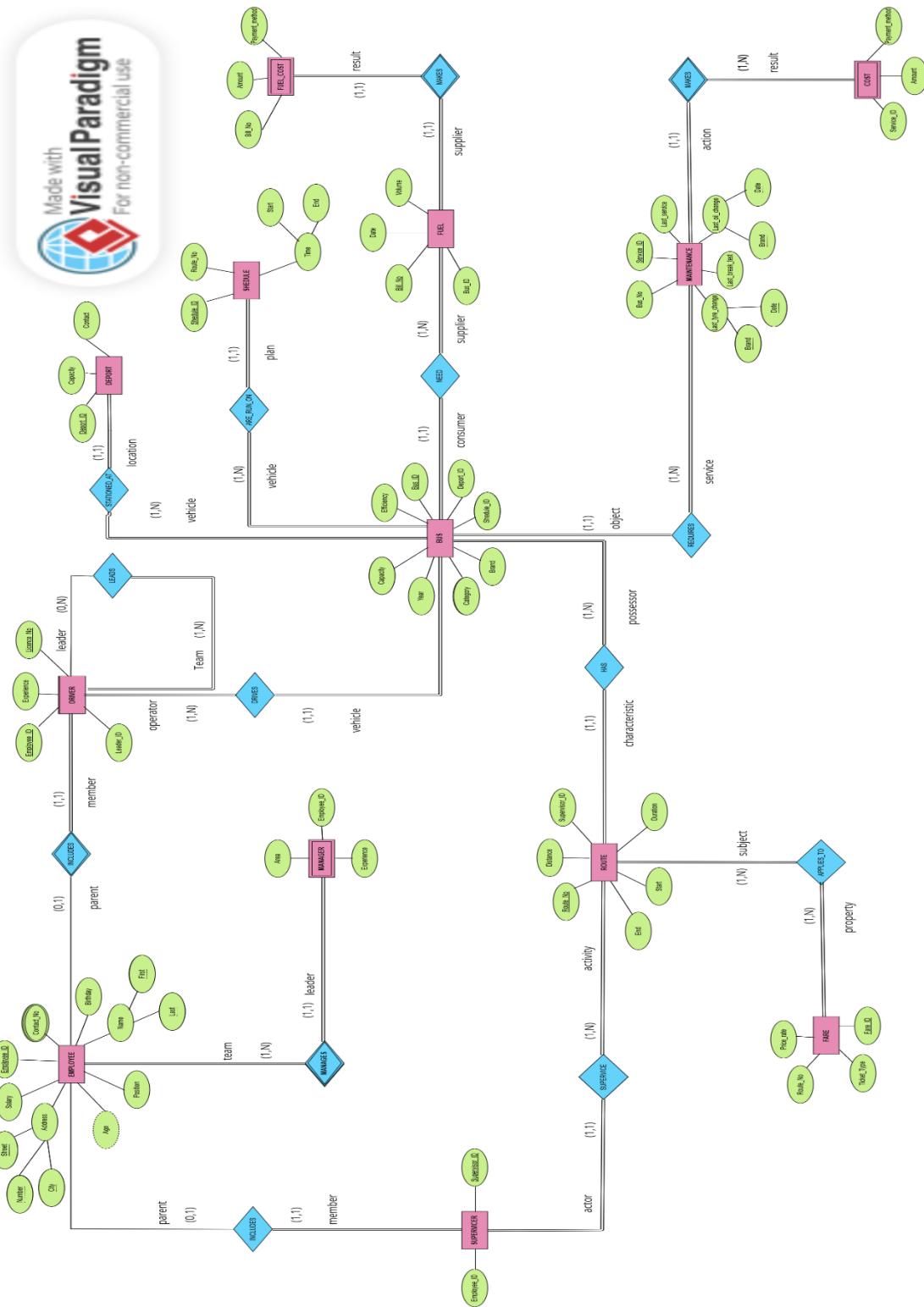


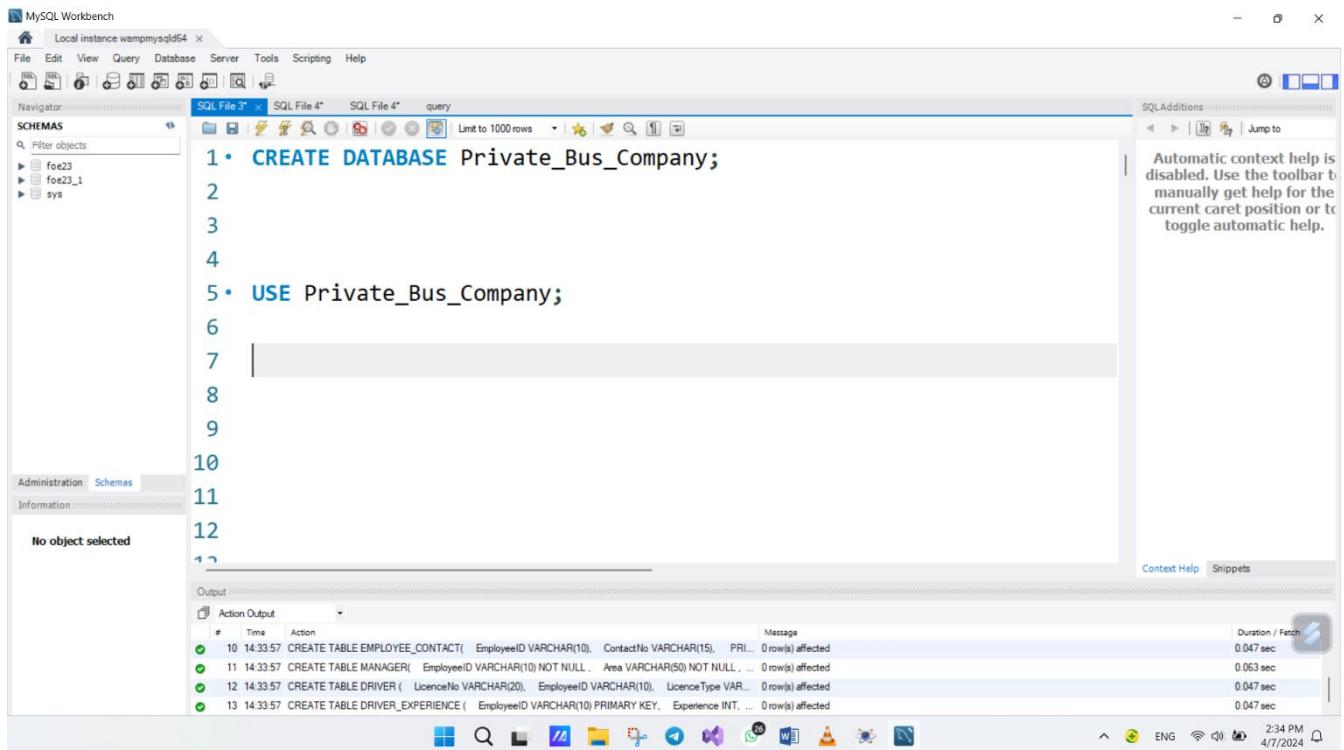
Figure 2-1:ER diagram of the farmhouse database management system

Chapter 3 - Implementation

In this chapter, all the screenshots of implemented database model are included.

3.1. TABLE IMPLEMENTATION AND CREATING FOREIGN KEY CONSTRAINT

1. Create database



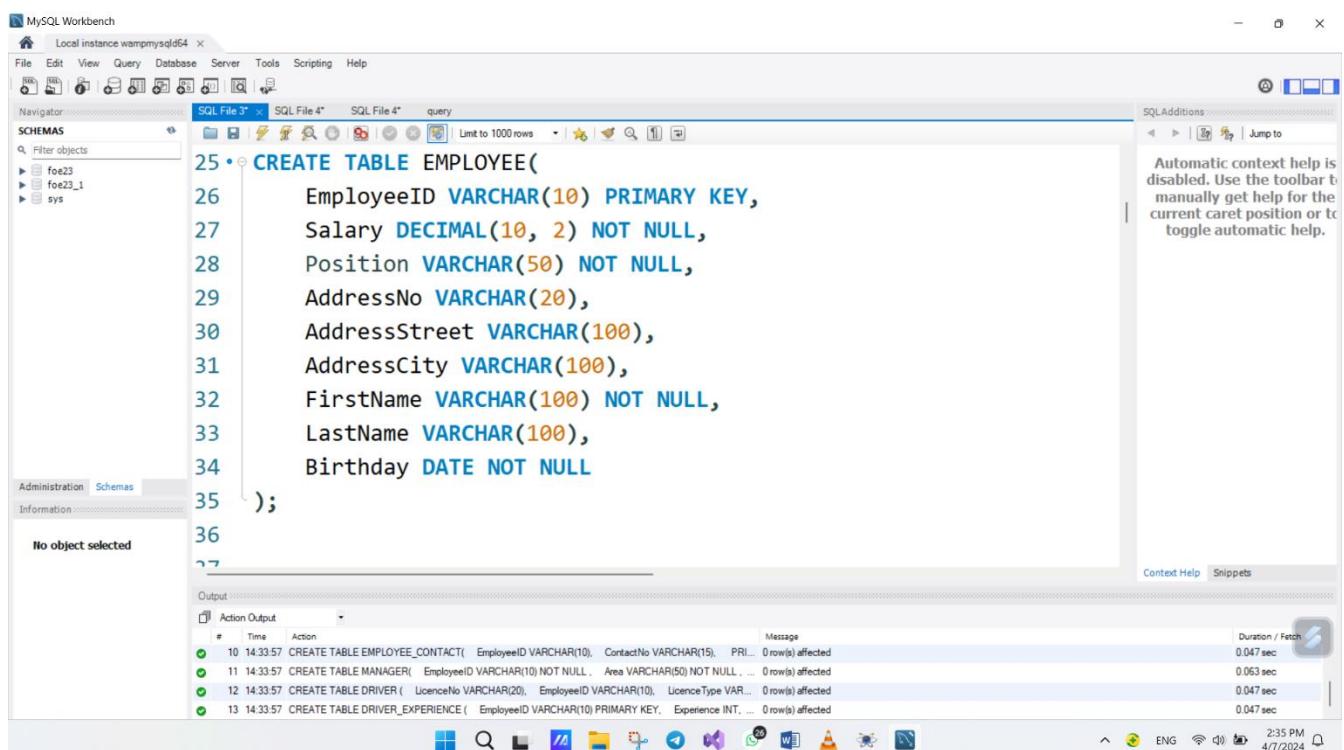
The screenshot shows the MySQL Workbench interface with the following details:

- Title Bar:** MySQL Workbench, Local instance wampmysqld64
- File Menu:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help
- Navigator:** Schemas (foe23, foe23_1, sys)
- SQL Editor:** SQL File 3* (query)
Content:

```
1 • CREATE DATABASE Private_Bus_Company;
2
3
4
5 • USE Private_Bus_Company;
```
- Output:** Action Output
List of actions and their results:

#	Time	Action	Message	Duration / Fetch
10	14:33:57	CREATE TABLE EMPLOYEE_CONTACT(EmployeeID VARCHAR(10), ContactNo VARCHAR(15), PRI...)	0 row(s) affected	0.047 sec
11	14:33:57	CREATE TABLE MANAGER(EmployeeID VARCHAR(10) NOT NULL, Area VARCHAR(50) NOT NULL, ...)	0 row(s) affected	0.063 sec
12	14:33:57	CREATE TABLE DRIVER (LicenceNo VARCHAR(20), EmployeeID VARCHAR(10), LicenceType VAR...	0 row(s) affected	0.047 sec
13	14:33:57	CREATE TABLE DRIVER_EXPERIENCE (EmployeeID VARCHAR(10) PRIMARY KEY, Experience INT, ...)	0 row(s) affected	0.047 sec

2. Create table Employee



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

- Title Bar:** MySQL Workbench, Local instance wampmysqld64
- File Menu:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help
- Navigator:** Schemas (foe23, foe23_1, sys)
- SQL Editor:** SQL File 3* (query)
Content:

```
25 • CREATE TABLE EMPLOYEE(
    EmployeeID VARCHAR(10) PRIMARY KEY,
    Salary DECIMAL(10, 2) NOT NULL,
    Position VARCHAR(50) NOT NULL,
    AddressNo VARCHAR(20),
    AddressStreet VARCHAR(100),
    AddressCity VARCHAR(100),
    FirstName VARCHAR(100) NOT NULL,
    LastName VARCHAR(100),
    Birthday DATE NOT NULL
);
```
- Output:** Action Output
List of actions and their results:

#	Time	Action	Message	Duration / Fetch
10	14:33:57	CREATE TABLE EMPLOYEE_CONTACT(EmployeeID VARCHAR(10), ContactNo VARCHAR(15), PRI...)	0 row(s) affected	0.047 sec
11	14:33:57	CREATE TABLE MANAGER(EmployeeID VARCHAR(10) NOT NULL, Area VARCHAR(50) NOT NULL, ...)	0 row(s) affected	0.063 sec
12	14:33:57	CREATE TABLE DRIVER (LicenceNo VARCHAR(20), EmployeeID VARCHAR(10), LicenceType VAR...	0 row(s) affected	0.047 sec
13	14:33:57	CREATE TABLE DRIVER_EXPERIENCE (EmployeeID VARCHAR(10) PRIMARY KEY, Experience INT, ...)	0 row(s) affected	0.047 sec

3. Create table Employee_contact

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

- Schemas:** fox23, fox23_1, sys
- SQL Editor:** SQL File 3* (query)
- Code:**

```
37
38 • CREATE TABLE EMPLOYEE_CONTACT(
39     EmployeeID VARCHAR(10),
40     ContactNo VARCHAR(15),
41
42     PRIMARY KEY(EmployeeID,ContactNo),
43
44     FOREIGN KEY(EmployeeID) REFERENCES EMPLOYEE(EmployeeID)
45         ON DELETE CASCADE
46         ON UPDATE CASCADE
47 );
48
```

- Output:** Shows the execution log with the following entries:

Action	Time	Message	Duration / Fetch
CREATE TABLE EMPLOYEE_CONTACT(EmployeeID VARCHAR(10), ContactNo VARCHAR(15), PRIMARY KEY(EmployeeID,ContactNo), FOREIGN KEY(EmployeeID) REFERENCES EMPLOYEE(EmployeeID) ON DELETE CASCADE ON UPDATE CASCADE);	10 14:33:57	0 row(s) affected	0.047 sec
CREATE TABLE MANAGER(EmployeeID VARCHAR(10) NOT NULL , Area VARCHAR(50) NOT NULL , Experience INT , FOREIGN KEY (EmployeeID) REFERENCES EMPLOYEE(EmployeeID) ON DELETE CASCADE ON UPDATE CASCADE);	11 14:33:57	0 row(s) affected	0.053 sec
CREATE TABLE DRIVER (LicenceNo VARCHAR(20), EmployeeID VARCHAR(10), LicenceType VAR... PRIMARY KEY(EmployeeID));	12 14:33:57	0 row(s) affected	0.047 sec
CREATE TABLE DRIVER_EXPERIENCE (EmployeeID VARCHAR(10) PRIMARY KEY, Experience INT, ...);	13 14:33:57	0 row(s) affected	0.047 sec

- System Bar:** Shows the date and time as 2:35 PM 4/7/2024.

4. Create table Manager

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

- Schemas:** fox23, fox23_1, sys
- SQL Editor:** SQL File 3* (query)
- Code:**

```
52
53 • CREATE TABLE MANAGER(
54     EmployeeID VARCHAR(10) NOT NULL ,
55     Area VARCHAR(50) NOT NULL ,
56     Experience INT ,
57     FOREIGN KEY (EmployeeID) REFERENCES EMPLOYEE(EmployeeID)
58         ON DELETE CASCADE
59         ON UPDATE CASCADE
60 );
61
62
63
```

- Output:** Shows the execution log with the following entries:

Action	Time	Message	Duration / Fetch
CREATE TABLE EMPLOYEE_CONTACT(EmployeeID VARCHAR(10), ContactNo VARCHAR(15), PRIMARY KEY(EmployeeID,ContactNo), FOREIGN KEY(EmployeeID) REFERENCES EMPLOYEE(EmployeeID) ON DELETE CASCADE ON UPDATE CASCADE);	10 14:33:57	0 row(s) affected	0.047 sec
CREATE TABLE MANAGER(EmployeeID VARCHAR(10) NOT NULL , Area VARCHAR(50) NOT NULL , Experience INT , FOREIGN KEY (EmployeeID) REFERENCES EMPLOYEE(EmployeeID) ON DELETE CASCADE ON UPDATE CASCADE);	11 14:33:57	0 row(s) affected	0.053 sec
CREATE TABLE DRIVER (LicenceNo VARCHAR(20), EmployeeID VARCHAR(10), LicenceType VAR... PRIMARY KEY(EmployeeID));	12 14:33:57	0 row(s) affected	0.047 sec
CREATE TABLE DRIVER_EXPERIENCE (EmployeeID VARCHAR(10) PRIMARY KEY, Experience INT, ...);	13 14:33:57	0 row(s) affected	0.047 sec

- System Bar:** Shows the date and time as 2:35 PM 4/7/2024.

5. Create table Driver

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

- Title Bar:** MySQL Workbench - Local instance wampmysqld64
- File Menu:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help
- Navigator:** Schemas (foe23, foe23_1, sys)
- SQL Editor:** SQL File 3* (query)
64
65 • CREATE TABLE DRIVER (
66 LicenceNo VARCHAR(20),
67 EmployeeID VARCHAR(10),
68 LicenceType VARCHAR(20),
69 LeaderID VARCHAR(20),
70 PRIMARY KEY (LicenceNo, EmployeeID),
71 UNIQUE (EmployeeID),
72 FOREIGN KEY (EmployeeID) REFERENCES EMPLOYEE(EmployeeID)
73 ON DELETE CASCADE
74 ON UPDATE CASCADE
75);
76
- Output Window:** Action Output (Shows log entries for table creation)

#	Time	Action	Message	Duration / Fetch
10	14:33:57	CREATE TABLE EMPLOYEE_CONTACT(EmployeeID VARCHAR(10), ContactNo VARCHAR(15), PRI...)	0 row(s) affected	0.047 sec
11	14:33:57	CREATE TABLE MANAGER(EmployeeID VARCHAR(10) NOT NULL , Area VARCHAR(50) NOT NULL , ...)	0 row(s) affected	0.063 sec
12	14:33:57	CREATE TABLE DRIVER (LicenceNo VARCHAR(20), EmployeeID VARCHAR(10), LicenceType VAR...)	0 row(s) affected	0.047 sec
13	14:33:57	CREATE TABLE DRIVER_EXPERIENCE (EmployeeID VARCHAR(10) PRIMARY KEY, Experience INT, ...)	0 row(s) affected	0.047 sec
- System Tray:** Shows battery level, signal strength, and system status.

6. Create table Driver_experience

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

- Title Bar:** MySQL Workbench - Local instance wampmysqld64
- File Menu:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help
- Navigator:** Schemas (foe23, foe23_1, sys)
- SQL Editor:** SQL File 3* (query)
76
77
78
79 • CREATE TABLE DRIVER_EXPERIENCE (
80 EmployeeID VARCHAR(10) PRIMARY KEY,
81 Experience INT,
82 FOREIGN KEY (EmployeeID) REFERENCES EMPLOYEE(EmployeeID)
83 ON DELETE CASCADE
84 ON UPDATE CASCADE
85);
86
87
88
- Output Window:** Action Output (Shows log entries for table creation)

#	Time	Action	Message	Duration / Fetch
10	14:33:57	CREATE TABLE EMPLOYEE_CONTACT(EmployeeID VARCHAR(10), ContactNo VARCHAR(15), PRI...)	0 row(s) affected	0.047 sec
11	14:33:57	CREATE TABLE MANAGER(EmployeeID VARCHAR(10) NOT NULL , Area VARCHAR(50) NOT NULL , ...)	0 row(s) affected	0.063 sec
12	14:33:57	CREATE TABLE DRIVER (LicenceNo VARCHAR(20), EmployeeID VARCHAR(10), LicenceType VAR...)	0 row(s) affected	0.047 sec
13	14:33:57	CREATE TABLE DRIVER_EXPERIENCE (EmployeeID VARCHAR(10) PRIMARY KEY, Experience INT, ...)	0 row(s) affected	0.047 sec
- System Tray:** Shows battery level, signal strength, and system status.

7. Create table Supervisor

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

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Local instance wampmysqld64, Schemas (foe23, foe23_1, sys).
- SQL Editor:** SQL File 3* (query), SQL File 4*. The code is:

```
88
89
90
91 • CREATE TABLE SUPERVISOR(
92     SupervisorID VARCHAR(20) PRIMARY KEY,
93     EmployeeID VARCHAR(10) NOT NULL UNIQUE ,
94     FOREIGN KEY (EmployeeID) REFERENCES EMPLOYEE(EmployeeID)
95         ON DELETE CASCADE
96         ON UPDATE CASCADE
97 );
98
99
100
```
- Output Window:** Action Output table showing the execution of previous queries (10-13) and the creation of the SUPERVISOR table (14). The log includes:

#	Time	Action	Message	Duration / Fetch
10	14:33:57	CREATE TABLE EMPLOYEE_CONTACT(EmployeeID VARCHAR(10), ContactNo VARCHAR(15), PRI...)	0 row(s) affected	0.047 sec
11	14:33:57	CREATE TABLE MANAGER(EmployeeID VARCHAR(10) NOT NULL , Area VARCHAR(50) NOT NULL , ...)	0 row(s) affected	0.063 sec
12	14:33:57	CREATE TABLE DRIVER (LicenceNo VARCHAR(20), EmployeeID VARCHAR(10), LicenceType VAR...)	0 row(s) affected	0.047 sec
13	14:33:57	CREATE TABLE DRIVER_EXPERIENCE (EmployeeID VARCHAR(10) PRIMARY KEY, Experience INT, ...)	0 row(s) affected	0.047 sec
- Right Panel:** SQLAdditions, Jump to, Context Help, Snippets.

8. Create table Route

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

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Local instance wampmysqld64, Schemas (foe23, foe23_1, sys).
- SQL Editor:** SQL File 3* (query), SQL File 4*. The code is:

```
02 • CREATE TABLE ROUTE(
03     RouteNo INT PRIMARY KEY,
04     SupervisorID VARCHAR(20) ,
05     Distance INT NOT NULL ,
06     Duration INT ,
07     Start VARCHAR(50) NOT NULL ,
08     End VARCHAR(50) NOT NULL ,
09     FOREIGN KEY (SupervisorID) REFERENCES SUPERVISOR(SupervisorID)
10         ON DELETE SET NULL
11         ON UPDATE CASCADE
12 );
13
```
- Output Window:** Action Output table showing the execution of previous queries (10-13) and the creation of the ROUTE table (14). The log includes:

#	Time	Action	Message	Duration / Fetch
10	14:33:57	CREATE TABLE EMPLOYEE_CONTACT(EmployeeID VARCHAR(10), ContactNo VARCHAR(15), PRI...)	0 row(s) affected	0.047 sec
11	14:33:57	CREATE TABLE MANAGER(EmployeeID VARCHAR(10) NOT NULL , Area VARCHAR(50) NOT NULL , ...)	0 row(s) affected	0.063 sec
12	14:33:57	CREATE TABLE DRIVER (LicenceNo VARCHAR(20), EmployeeID VARCHAR(10), LicenceType VAR...)	0 row(s) affected	0.047 sec
13	14:33:57	CREATE TABLE DRIVER_EXPERIENCE (EmployeeID VARCHAR(10) PRIMARY KEY, Experience INT, ...)	0 row(s) affected	0.047 sec
- Right Panel:** SQLAdditions, Jump to, Context Help, Snippets.

9. Create table Bus_schedule

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

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Local instance wampmysqld64, showing schemas fo23, fo23_1, and sys.
- SQL Editor:** SQL File 3* (selected), SQL File 4*, query. The code is:

```
17 • CREATE TABLE BUS_SCHEDULE(
18     ScheduleID VARCHAR(10) PRIMARY KEY,
19     RouteNo INT NOT NULL ,
20     StartTime TIME NOT NULL ,
21     EndTime TIME NOT NULL ,
22     FOREIGN KEY (RouteNo) REFERENCES ROUTE(RouteNo)
23         ON DELETE CASCADE
24         ON UPDATE CASCADE
25 );
26
27
28
29
```

- Output Window:** Action Output table showing the execution of previous queries:

#	Time	Action	Message	Duration / Fetch
10	14:33:57	CREATE TABLE EMPLOYEE_CONTACT(EmployeeID VARCHAR(10), ContactNo VARCHAR(15), PRI...)	0 row(s) affected	0.047 sec
11	14:33:57	CREATE TABLE MANAGER(EmployeeID VARCHAR(10) NOT NULL , Area VARCHAR(50) NOT NULL , ...)	0 row(s) affected	0.063 sec
12	14:33:57	CREATE TABLE DRIVER (LicenceNo VARCHAR(20), EmployeeID VARCHAR(10), LicenceType VAR...)	0 row(s) affected	0.047 sec
13	14:33:57	CREATE TABLE DRIVER_EXPERIENCE (EmployeeID VARCHAR(10) PRIMARY KEY, Experience INT, ...)	0 row(s) affected	0.047 sec

10. Create table Depot

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

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Local instance wampmysqld64, showing schemas fo23, fo23_1, and sys.
- SQL Editor:** SQL File 3* (selected), SQL File 4*, query. The code is:

```
28
29
30 • CREATE TABLE DEPOT(
31     DepotID VARCHAR(10) PRIMARY KEY,
32     Capacity INT,
33     ContactNo VARCHAR(15) NOT NULL UNIQUE
34 );
```

- Output Window:** Action Output table showing the execution of previous queries:

#	Time	Action	Message	Duration / Fetch
10	14:33:57	CREATE TABLE EMPLOYEE_CONTACT(EmployeeID VARCHAR(10), ContactNo VARCHAR(15), PRI...)	0 row(s) affected	0.047 sec
11	14:33:57	CREATE TABLE MANAGER(EmployeeID VARCHAR(10) NOT NULL , Area VARCHAR(50) NOT NULL , ...)	0 row(s) affected	0.063 sec
12	14:33:57	CREATE TABLE DRIVER (LicenceNo VARCHAR(20), EmployeeID VARCHAR(10), LicenceType VAR...)	0 row(s) affected	0.047 sec
13	14:33:57	CREATE TABLE DRIVER_EXPERIENCE (EmployeeID VARCHAR(10) PRIMARY KEY, Experience INT, ...)	0 row(s) affected	0.047 sec

11. Create table Bus

The screenshot shows the MySQL Workbench interface with the 'Schemas' tab selected. In the central query editor, the SQL code for creating the BUS table is displayed. The table has columns for BusID (primary key), ScheduleID, DepotID, Brand, Efficiency, Capacity, Year, and Category. It includes foreign key constraints linking to the BUS_SCHEDULE and DEPOT tables.

```
.43 • CREATE TABLE BUS(
.44     BusID VARCHAR(10) PRIMARY KEY,
.45     ScheduleID VARCHAR(10),
.46     DepotID VARCHAR(10),
.47     Brand VARCHAR(20),
.48     Efficiency INT,
.49     Capacity INT,
.50     Year YEAR,
.51     Category VARCHAR(20) NOT NULL,
.52     FOREIGN KEY (ScheduleID) REFERENCES BUS_SCHEDULE(ScheduleID)
.53         ON DELETE SET NULL
.54         ON UPDATE CASCADE,
.55     FOREIGN KEY (DepotID) REFERENCES DEPOT(DepotID)
.56         ON DELETE SET NULL
.57         ON UPDATE CASCADE
.58 );
.59
```

12. Create table Fare

The screenshot shows the MySQL Workbench interface with the 'Schemas' tab selected. In the central query editor, the SQL code for creating the FARE table is displayed. The table has columns for FareID (primary key), RouteNo, PriceRate, and TicketType. It includes a foreign key constraint linking to the ROUTE table.

```
.59
.60
.61
.62 • CREATE TABLE FARE(
.63     FareID VARCHAR(20) PRIMARY KEY,
.64     RouteNo INT NOT NULL,
.65     PriceRate DECIMAL(10, 2) NOT NULL,
.66     TicketType VARCHAR(50) NOT NULL,
.67     FOREIGN KEY (RouteNo) REFERENCES ROUTE(RouteNo)
.68         ON DELETE CASCADE
.69         ON UPDATE CASCADE
.70 );
.71
.72
.73
.74
.75
```

13. Create table Maintenance

The screenshot shows the MySQL Workbench interface with the 'Maintenance' schema selected. The SQL editor contains the following code:

```
.77
.78 • CREATE TABLE MAINTENANCE(
.79     ServiceID VARCHAR(20) PRIMARY KEY,
.80     BusID VARCHAR(10),
.81     LastTireChange DATE,
.82     LastTireBrand VARCHAR(50),
.83     LastBrakeChange DATE,
.84     LastOilChange DATE,
.85     LastOilBrand VARCHAR(50),
.86     LastService DATE,
.87     FOREIGN KEY (BusID) REFERENCES BUS(BusID)
.88         ON DELETE SET NULL
.89         ON UPDATE CASCADE
.90 );
.91
.92
.93
```

The status bar at the bottom right shows the date and time as 4/7/2024 2:36 PM.

14. Create table Service_cost

The screenshot shows the MySQL Workbench interface with the 'Maintenance' schema selected. The SQL editor contains the following code:

```
.95
.96 • CREATE TABLE SERVICE_COST(
.97     ServiceID VARCHAR(20),
.98     Amount DECIMAL(10, 2) NOT NULL,
.99     Payment_method VARCHAR(20) NOT NULL,
.100    FOREIGN KEY (ServiceID) REFERENCES MAINTENANCE(ServiceID)
.101        ON DELETE CASCADE
.102        ON UPDATE CASCADE
.103 );
.104
.105
.106
.107
.108
.109
```

The status bar at the bottom right shows the date and time as 4/7/2024 2:37 PM.

#	Time	Action	Message	Duration / Fetch
10	14:33:57	CREATE TABLE EMPLOYEE_CONTACT(EmployeeID VARCHAR(10), ContactNo VARCHAR(15), PRI...)	0 row(s) affected	0.047 sec
11	14:33:57	CREATE TABLE MANAGER(EmployeeID VARCHAR(10) NOT NULL, Area VARCHAR(50) NOT NULL, ...)	0 row(s) affected	0.063 sec
12	14:33:57	CREATE TABLE DRIVER (LicenceNo VARCHAR(20), EmployeeID VARCHAR(10), LicenceType VAR...)	0 row(s) affected	0.047 sec

15. Create table Fuel

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

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Local instance wampmysqld54, showing schemas foec23, foec23_1, and sys.
- SQL Editor:** SQL File 4* tab, containing the SQL code for creating the FUEL table:

```
!10 • CREATE TABLE FUEL(
!11     BillNo VARCHAR(20) PRIMARY KEY,
!12     BusID VARCHAR(10),
!13     FillDate DATE NOT NULL,
!14     Volume FLOAT NOT NULL,
!15     FOREIGN KEY (BusID) REFERENCES BUS(BusID)
!16         ON DELETE SET NULL
!17         ON UPDATE CASCADE
!18 );
```
- Output Window:** Shows the results of the previous queries:

Action	Time	Message	Duration / Fetch
CREATE TABLE FUEL	14:33:57	0 row(s) affected	0.047 sec
CREATE TABLE EMPLOYEE_CONTACT	14:33:57	0 row(s) affected	0.063 sec
CREATE TABLE DRIVER	14:33:57	0 row(s) affected	0.047 sec
- System Tray:** Shows battery level, signal strength, and system status.

16. Create table Fuel_cost

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

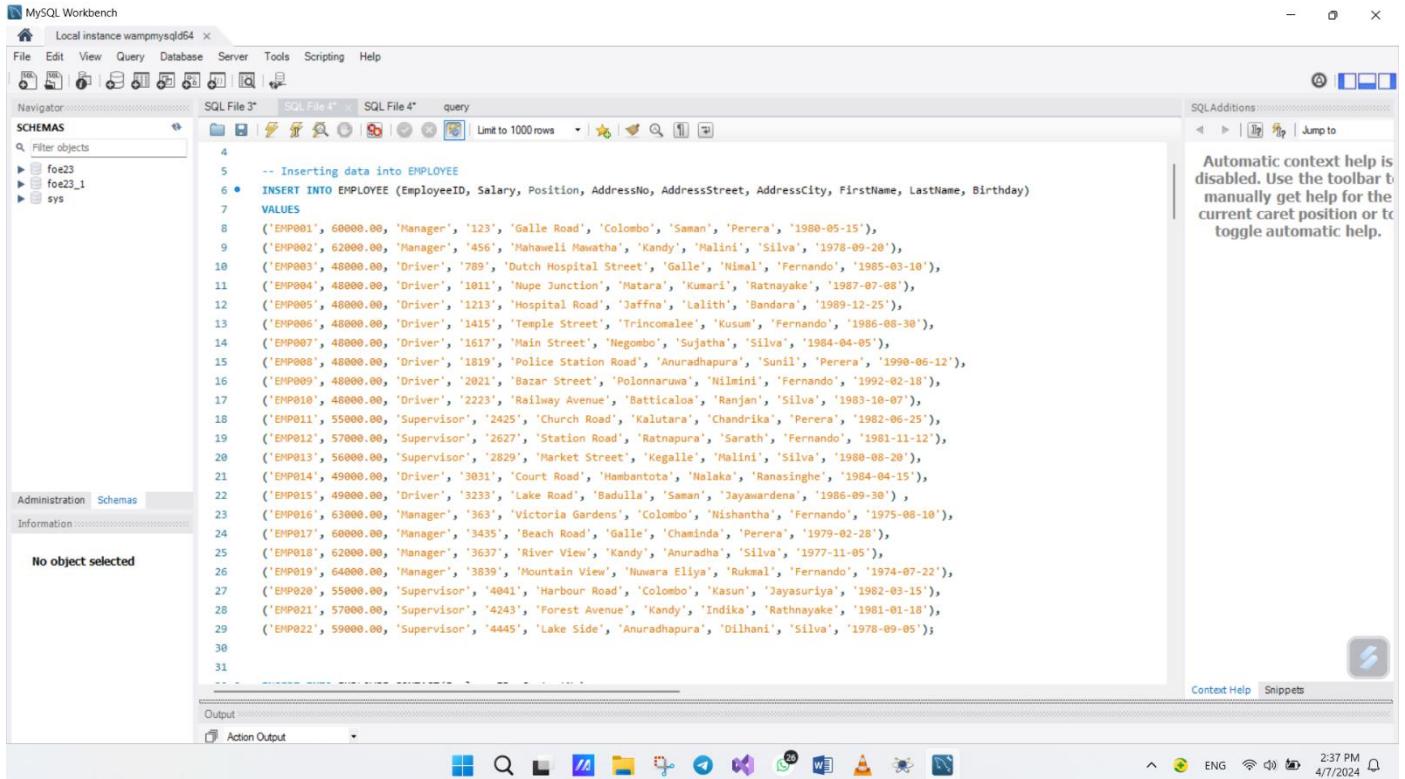
- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Local instance wampmysqld54, showing schemas foec23, foec23_1, and sys.
- SQL Editor:** SQL File 4* tab, containing the SQL code for creating the FUEL_COST table:

```
!23
!24
!25 • CREATE TABLE FUEL_COST(
!26     BillNo VARCHAR(20) NOT NULL,
!27     Amount DECIMAL(10, 2) NOT NULL,
!28     Payment_method VARCHAR(20),
!29     FOREIGN KEY (BillNo) REFERENCES FUEL(BillNo)
!30         ON DELETE CASCADE
!31         ON UPDATE CASCADE
!32 );
```
- Output Window:** Shows the results of the previous queries:

Action	Time	Message	Duration / Fetch
CREATE TABLE FUEL_COST	14:33:57	0 row(s) affected	0.047 sec
CREATE TABLE EMPLOYEE_CONTACT	14:33:57	0 row(s) affected	0.063 sec
CREATE TABLE DRIVER	14:33:57	0 row(s) affected	0.047 sec
- System Tray:** Shows battery level, signal strength, and system status.

3.2. INSERTING DATA FOR TABLES

1. Employee Table

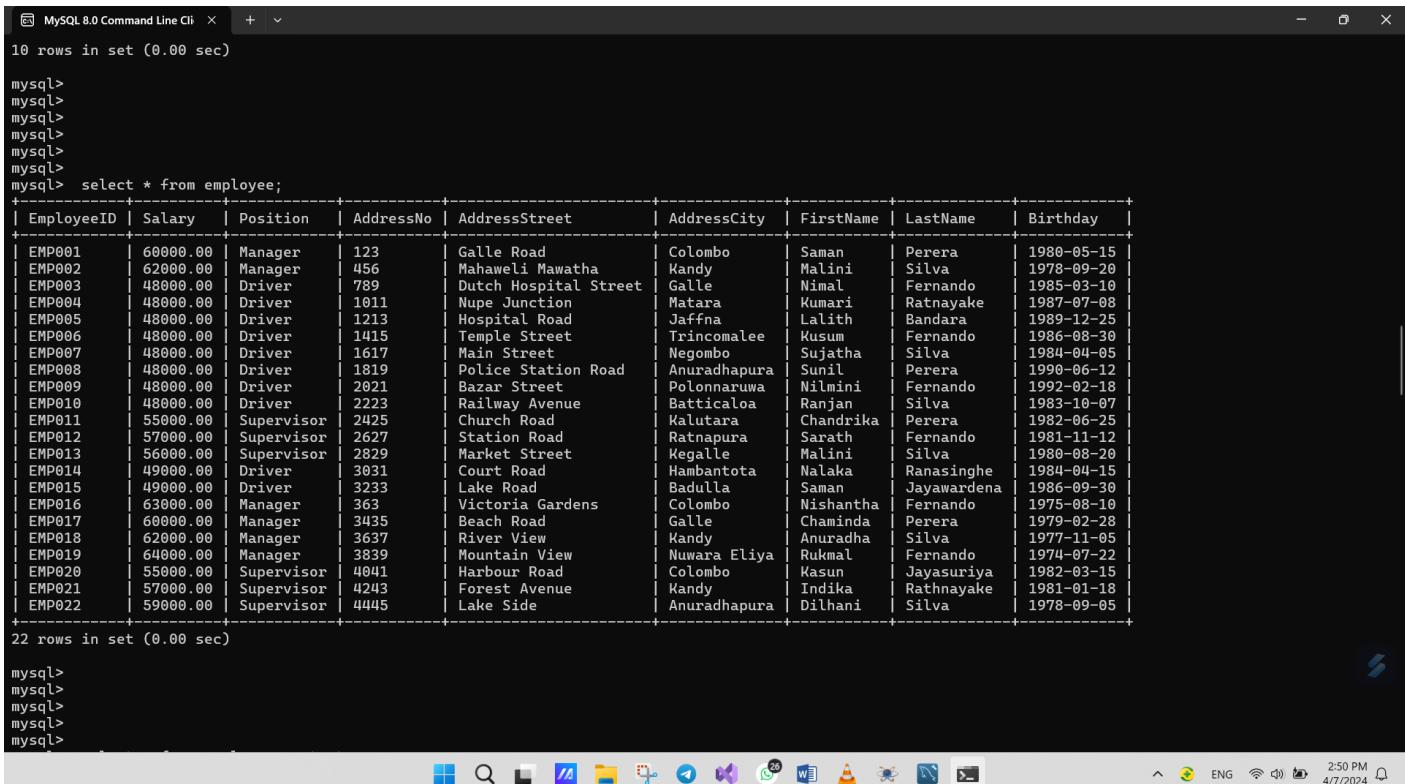


The screenshot shows the MySQL Workbench interface with a SQL editor window. The code is a script for inserting data into the 'EMPLOYEE' table. The script includes a comment '-- Inserting data into EMPLOYEE', followed by an 'INSERT INTO' statement with specific values for each column: EmployeeID, Salary, Position, AddressNo, AddressStreet, AddressCity, FirstName, LastName, and Birthday. The data consists of 22 rows, each representing an employee with their details like position, address, and birthdate.

```

-- Inserting data into EMPLOYEE
INSERT INTO EMPLOYEE (EmployeeID, Salary, Position, AddressNo, AddressStreet, AddressCity, FirstName, LastName, Birthday)
VALUES
('EMP001', 60000.00, 'Manager', '123', 'Galle Road', 'Colombo', 'Saman', 'Perera', '1980-05-15'),
('EMP002', 62000.00, 'Manager', '456', 'Mahaweli Mawatha', 'Kandy', 'Malini', 'Silva', '1978-09-20'),
('EMP003', 48000.00, 'Driver', '789', 'Dutch Hospital Street', 'Galle', 'Nimal', 'Fernando', '1985-03-10'),
('EMP004', 48000.00, 'Driver', '1011', 'Nupe Junction', 'Matara', 'Kumari', 'Ratnayake', '1987-07-08'),
('EMP005', 48000.00, 'Driver', '1213', 'Hospital Road', 'Jaffna', 'Lalith', 'Bandara', '1989-12-25'),
('EMP006', 48000.00, 'Driver', '1415', 'Temple Street', 'Trincomalee', 'Kusum', 'Fernando', '1986-08-30'),
('EMP007', 48000.00, 'Driver', '1617', 'Main Street', 'Negombo', 'Sujatha', 'Silva', '1984-04-05'),
('EMP008', 48000.00, 'Driver', '1819', 'Police Station Road', 'Anuradhapura', 'Sunil', 'Perera', '1990-06-12'),
('EMP009', 48000.00, 'Driver', '2021', 'Bazar Street', 'Polonnaruwa', 'Nilmini', 'Fernando', '1992-02-18'),
('EMP010', 48000.00, 'Driver', '2223', 'Railway Avenue', 'Batticaloa', 'Ranjan', 'Silva', '1983-10-07'),
('EMP011', 55000.00, 'Supervisor', '2425', 'Church Road', 'Kalutara', 'Chandrika', 'Perera', '1982-06-25'),
('EMP012', 57000.00, 'Supervisor', '2627', 'Station Road', 'Ratnapura', 'Sarah', 'Fernando', '1981-11-12'),
('EMP013', 56000.00, 'Supervisor', '2829', 'Market Street', 'Kegalle', 'Malini', 'Silva', '1980-08-20'),
('EMP014', 49000.00, 'Driver', '3031', 'Court Road', 'Hambantota', 'Nalaka', 'Ranasinghe', '1984-04-15'),
('EMP015', 49000.00, 'Driver', '3233', 'Lake Road', 'Badulla', 'Saman', 'Jayawardena', '1986-09-30'),
('EMP016', 63000.00, 'Manager', '363', 'Victoria Gardens', 'Colombo', 'Nishantha', 'Fernando', '1975-08-10'),
('EMP017', 60000.00, 'Manager', '3435', 'Beach Road', 'Galle', 'Chaminda', 'Perera', '1979-02-28'),
('EMP018', 62000.00, 'Manager', '3637', 'River View', 'Kandy', 'Anuradha', 'Silva', '1977-11-05'),
('EMP019', 64000.00, 'Manager', '3839', 'Mountain View', 'Nuwara Eliya', 'Rukmal', 'Fernando', '1974-07-22'),
('EMP020', 55000.00, 'Supervisor', '4041', 'Harbour Road', 'Colombo', 'Kasun', 'Jayasuriya', '1982-03-15'),
('EMP021', 57000.00, 'Supervisor', '4243', 'Forest Avenue', 'Kandy', 'Indika', 'Rathnayake', '1981-01-18'),
('EMP022', 59000.00, 'Supervisor', '4445', 'Lake Side', 'Anuradhapura', 'Dilhani', 'Silva', '1978-09-05');

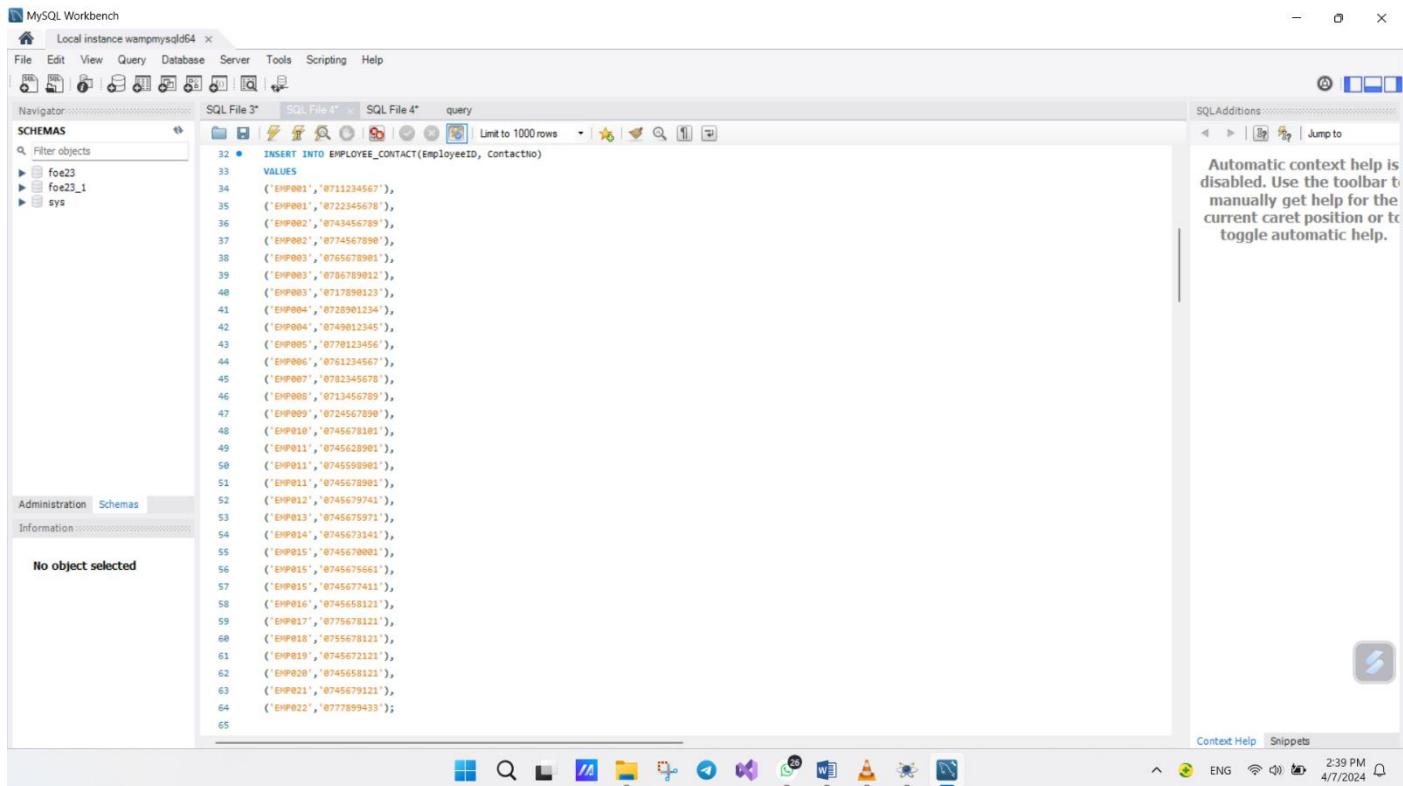
```



The screenshot shows the MySQL Command Line Client interface. A command 'select * from employee;' is run, resulting in 22 rows of data being displayed. The data is identical to the one shown in the MySQL Workbench screenshot above, representing the same 22 employees with their respective details.

EmployeeID	Salary	Position	AddressNo	AddressStreet	AddressCity	FirstName	LastName	Birthday
EMP001	60000.00	Manager	123	Galle Road	Colombo	Saman	Perera	1980-05-15
EMP002	62000.00	Manager	456	Mahaweli Mawatha	Kandy	Malini	Silva	1978-09-20
EMP003	48000.00	Driver	789	Dutch Hospital Street	Galle	Nimal	Fernando	1985-03-10
EMP004	48000.00	Driver	1011	Nupe Junction	Matara	Kumari	Ratnayake	1987-07-08
EMP005	48000.00	Driver	1213	Hospital Road	Jaffna	Lalith	Bandara	1989-12-25
EMP006	48000.00	Driver	1415	Temple Street	Trincomalee	Kusum	Fernando	1986-08-30
EMP007	48000.00	Driver	1617	Main Street	Negombo	Sujatha	Silva	1984-04-05
EMP008	48000.00	Driver	1819	Police Station Road	Anuradhapura	Sunil	Perera	1990-06-12
EMP009	48000.00	Driver	2021	Bazar Street	Polonnaruwa	Nilmini	Fernando	1992-02-18
EMP010	48000.00	Driver	2223	Railway Avenue	Batticaloa	Ranjan	Silva	1983-10-07
EMP011	55000.00	Supervisor	2425	Church Road	Kalutara	Chandrika	Perera	1982-06-25
EMP012	57000.00	Supervisor	2627	Station Road	Ratnapura	Sarah	Fernando	1981-11-12
EMP013	56000.00	Supervisor	2829	Market Street	Kegalle	Malini	Silva	1980-08-20
EMP014	49000.00	Driver	3031	Court Road	Hambantota	Nalaka	Ranasinghe	1984-04-15
EMP015	49000.00	Driver	3233	Lake Road	Badulla	Saman	Jayawardena	1986-09-30
EMP016	63000.00	Manager	363	Victoria Gardens	Colombo	Nishantha	Fernando	1975-08-10
EMP017	60000.00	Manager	3435	Beach Road	Galle	Chaminda	Perera	1979-02-28
EMP018	62000.00	Manager	3637	River View	Kandy	Anuradha	Silva	1977-11-05
EMP019	64000.00	Manager	3839	Mountain View	Nuwara Eliya	Rukmal	Fernando	1974-07-22
EMP020	55000.00	Supervisor	4041	Harbour Road	Colombo	Kasun	Jayasuriya	1982-03-15
EMP021	57000.00	Supervisor	4243	Forest Avenue	Kandy	Indika	Rathnayake	1981-01-18
EMP022	59000.00	Supervisor	4445	Lake Side	Anuradhapura	Dilhani	Silva	1978-09-05

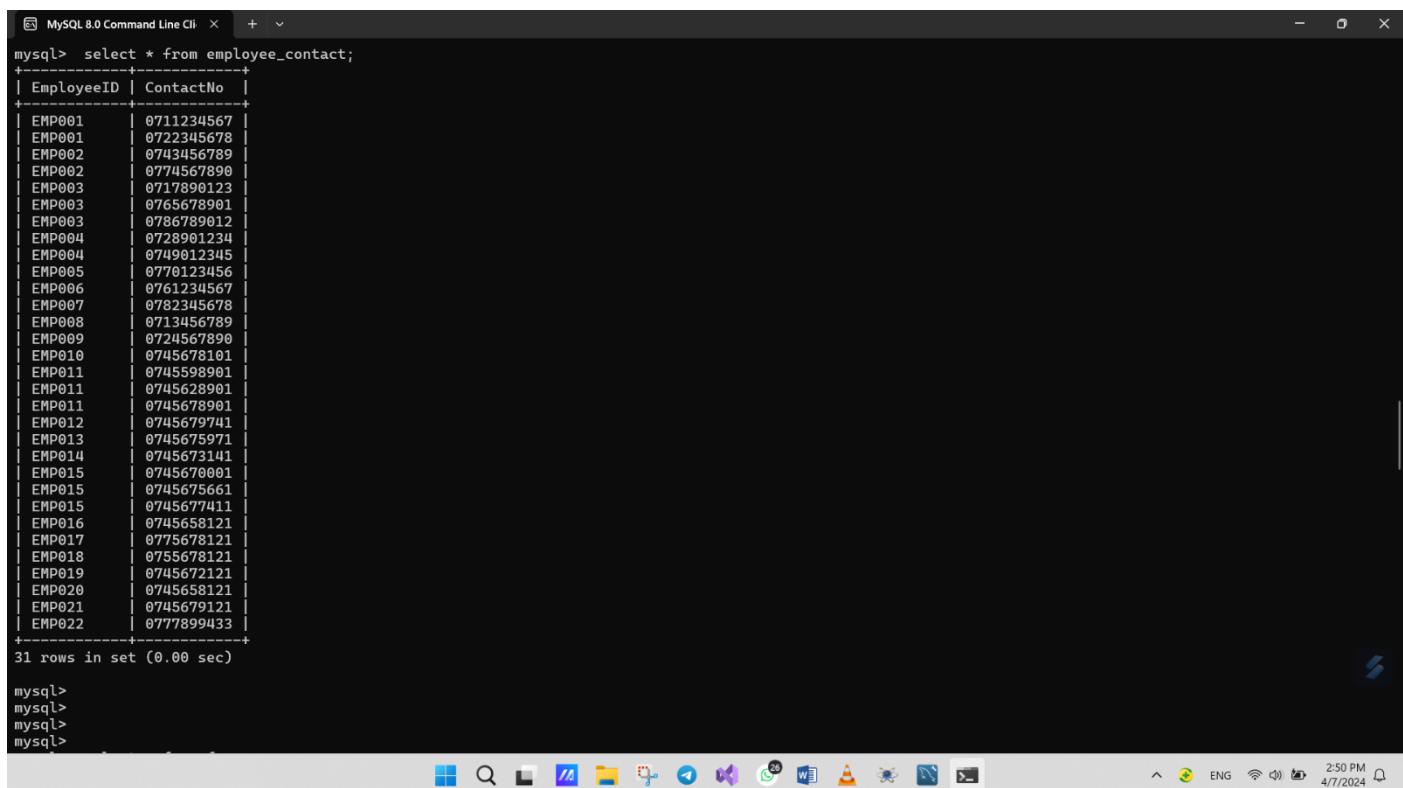
2. Employee_contact Table



The screenshot shows the MySQL Workbench interface. In the center, there is a query editor window titled "query". The code in the editor is an INSERT INTO statement for the "EMPLOYEE_CONTACT" table:

```
32 • INSERT INTO EMPLOYEE_CONTACT(EmployeeID, ContactNo)
33 VALUES
34 ('EMP001','0711234567'),
35 ('EMP001','0722345678'),
36 ('EMP002','0743456789')
37 ('EMP002','0774567890'),
38 ('EMP003','0765678901'),
39 ('EMP003','0786789012'),
40 ('EMP003','0717890123'),
41 ('EMP004','0728901234'),
42 ('EMP004','0749012345'),
43 ('EMP005','0770123456')
44 ('EMP006','0761234567'),
45 ('EMP007','0782345678'),
46 ('EMP008','0713456789')
47 ('EMP009','0724567890'),
48 ('EMP009','0745678901'),
49 ('EMP010','0745678101'),
50 ('EMP011','0745628901'),
51 ('EMP011','0745678901'),
52 ('EMP012','0745679741')
53 ('EMP013','0745675971')
54 ('EMP014','0745673141')
55 ('EMP015','0745670001')
56 ('EMP015','0745675661')
57 ('EMP015','0745677411')
58 ('EMP016','0745658121')
59 ('EMP017','0775678121')
60 ('EMP018','0755678121')
61 ('EMP019','0745672121')
62 ('EMP020','0745658121')
63 ('EMP021','0745679121')
64 ('EMP022','0777899433');
```

The code consists of 64 rows of data, each containing an EmployeeID and a ContactNo. A note on the right side of the interface states: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."



The screenshot shows the MySQL 8.0 Command Line Client interface. In the terminal window, the user has run a SELECT query:

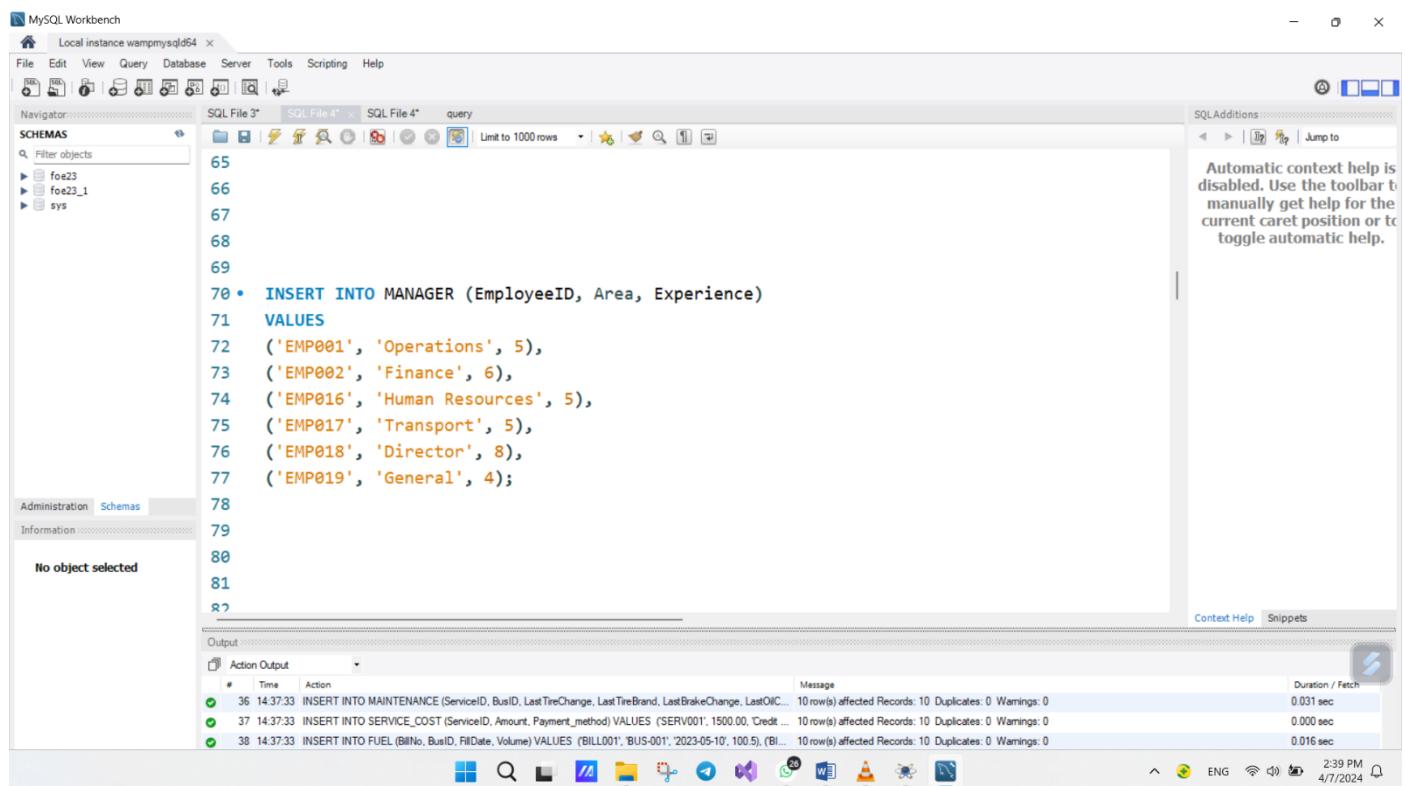
```
mysql> select * from employee_contact;
```

The results are displayed as a table:

EmployeeID	ContactNo
EMP001	0711234567
EMP001	0722345678
EMP002	0743456789
EMP002	0774567890
EMP003	0717890123
EMP003	0765678901
EMP003	0786789012
EMP004	0728901234
EMP004	0749012345
EMP005	0770123456
EMP006	0761234567
EMP007	0782345678
EMP008	0713456789
EMP009	0724567890
EMP010	0745678101
EMP011	0745598901
EMP011	0745628901
EMP011	0745678901
EMP012	0745679741
EMP013	0745675971
EMP014	0745673141
EMP015	0745670001
EMP015	0745675661
EMP015	0745677411
EMP016	0745658121
EMP017	0775678121
EMP018	0755678121
EMP019	0745672121
EMP020	0745658121
EMP021	0745679121
EMP022	0777899433

At the bottom of the terminal, it says "31 rows in set (0.00 sec)".

3. Manager Table



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

- Schemas:** Local instance wampmysqld54 contains schemas: foe23, foe23_1, and sys.
- SQL Editor:** The code being run is:

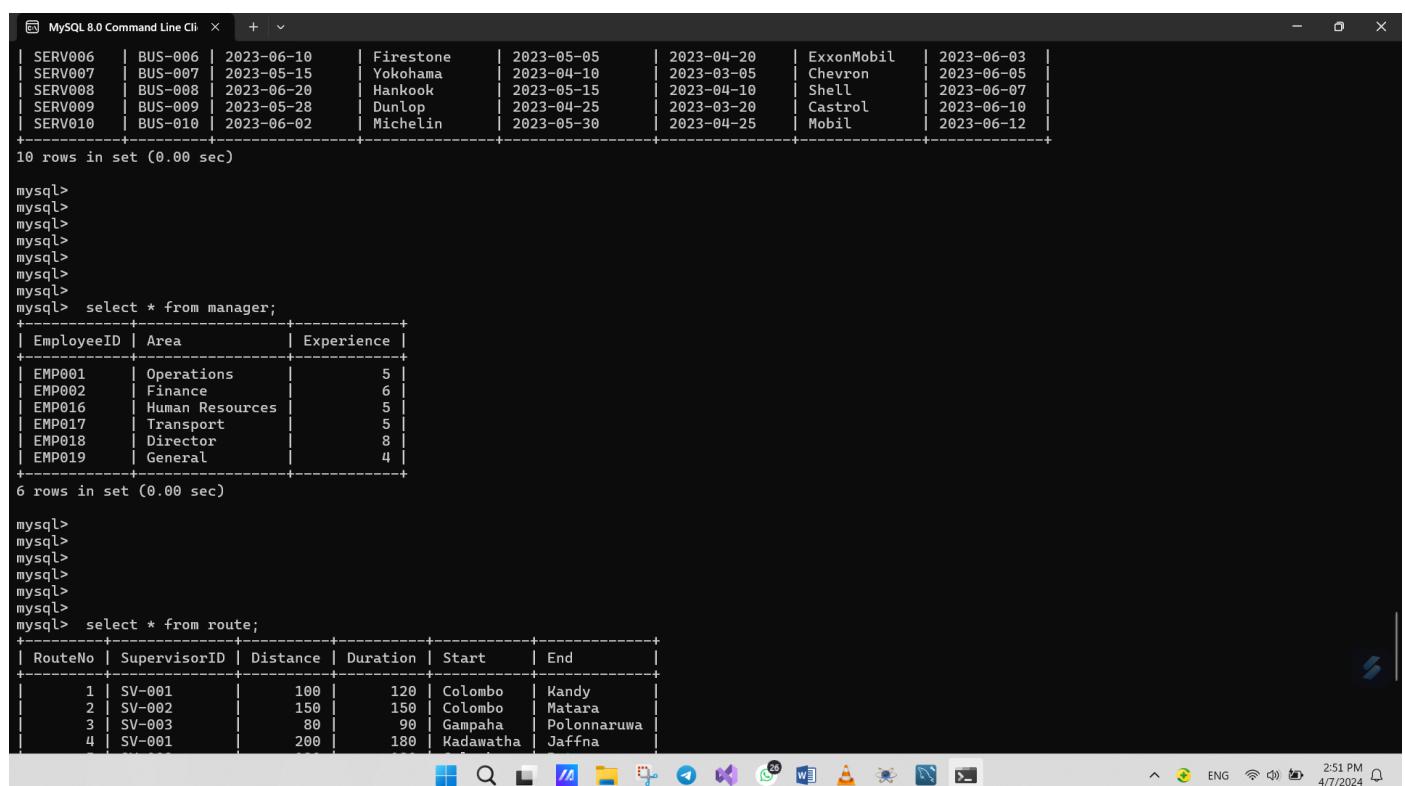
```

65
66
67
68
69
70 • INSERT INTO MANAGER (EmployeeID, Area, Experience)
71     VALUES
72     ('EMP001', 'Operations', 5),
73     ('EMP002', 'Finance', 6),
74     ('EMP016', 'Human Resources', 5),
75     ('EMP017', 'Transport', 5),
76     ('EMP018', 'Director', 8),
77     ('EMP019', 'General', 4);
78
79
80
81
82

```

- Output:** The output shows three successful insert operations:

Action	Time	Action	Message	Duration / Fetch
36	14:37:33	INSERT INTO MAINTENANCE (ServiceID, BusID, LastTireChange, LastTireBrand, LastBrakeChange, LastOilChange)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.031 sec
37	14:37:33	INSERT INTO SERVICE_COST (ServiceID, Amount, Payment_method) VALUES ('SERV001', 1500.00, 'Credit')	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
38	14:37:33	INSERT INTO FUEL (BillNo, BusID, RlDate, Volume) VALUES ('BILL001', 'BUS-001', 2023-05-10, 100.5), ('BILL002', 'BUS-002', 2023-05-10, 100.5), ('BILL003', 'BUS-003', 2023-05-10, 100.5), ('BILL004', 'BUS-004', 2023-05-10, 100.5), ('BILL005', 'BUS-005', 2023-05-10, 100.5), ('BILL006', 'BUS-006', 2023-05-10, 100.5), ('BILL007', 'BUS-007', 2023-05-10, 100.5), ('BILL008', 'BUS-008', 2023-05-10, 100.5), ('BILL009', 'BUS-009', 2023-05-10, 100.5), ('BILL010', 'BUS-010', 2023-05-10, 100.5)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.016 sec



The screenshot shows the MySQL 8.0 Command Line Cli interface with the following details:

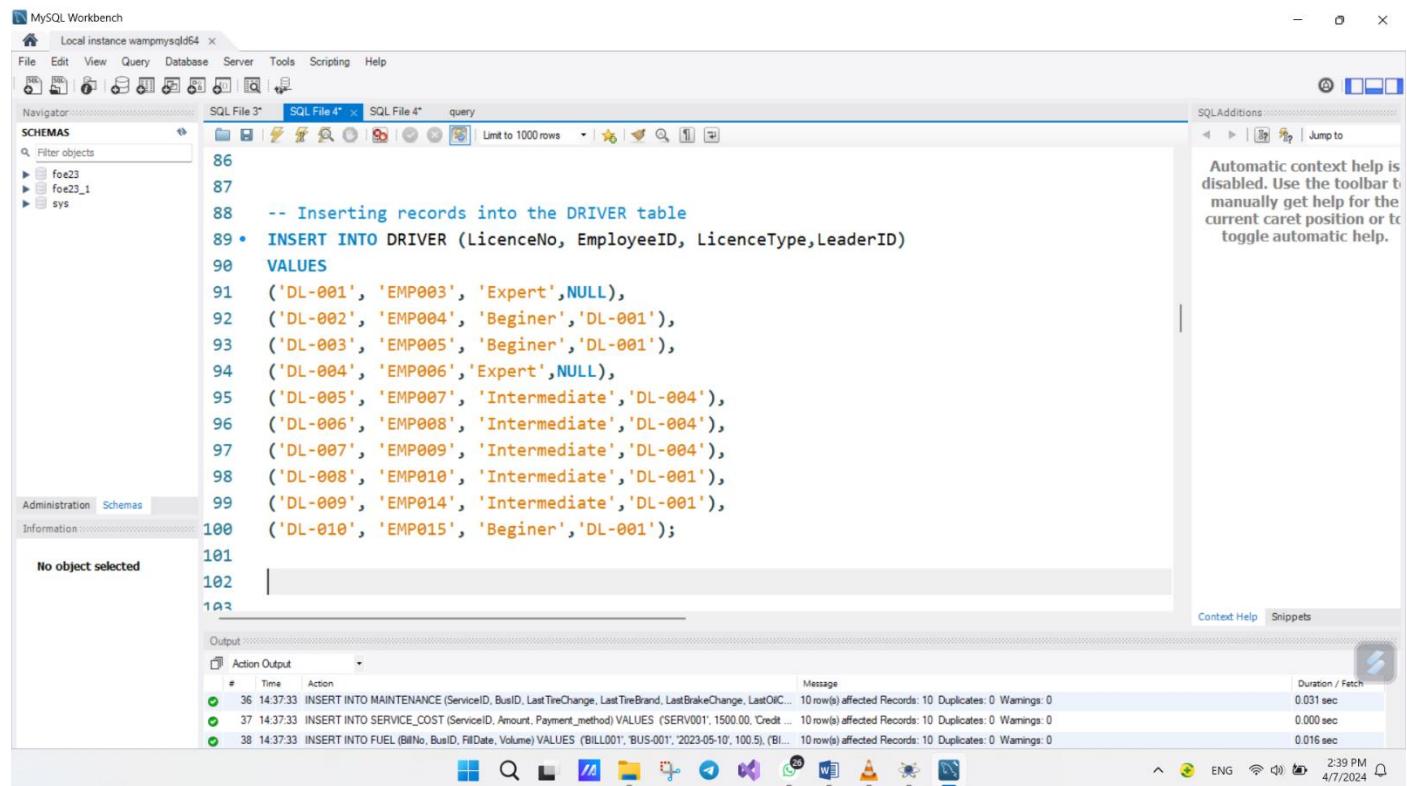
- Output:** The results of the query `select * from manager;` are displayed as a table:

EmployeeID	Area	Experience
EMP001	Operations	5
EMP002	Finance	6
EMP016	Human Resources	5
EMP017	Transport	5
EMP018	Director	8
EMP019	General	4

- Output:** The results of the query `select * from route;` are displayed as a table:

RouteNo	SupervisorID	Distance	Duration	Start	End
1	SV-001	100	120	Colombo	Kandy
2	SV-002	150	150	Colombo	Matara
3	SV-003	80	90	Gampaha	Polonnaruwa
4	SV-001	200	180	Kadawatha	Jaffna

4. Driver Table



The screenshot shows the MySQL Workbench interface. In the center, there is a SQL editor window titled "query". The code in the editor is as follows:

```

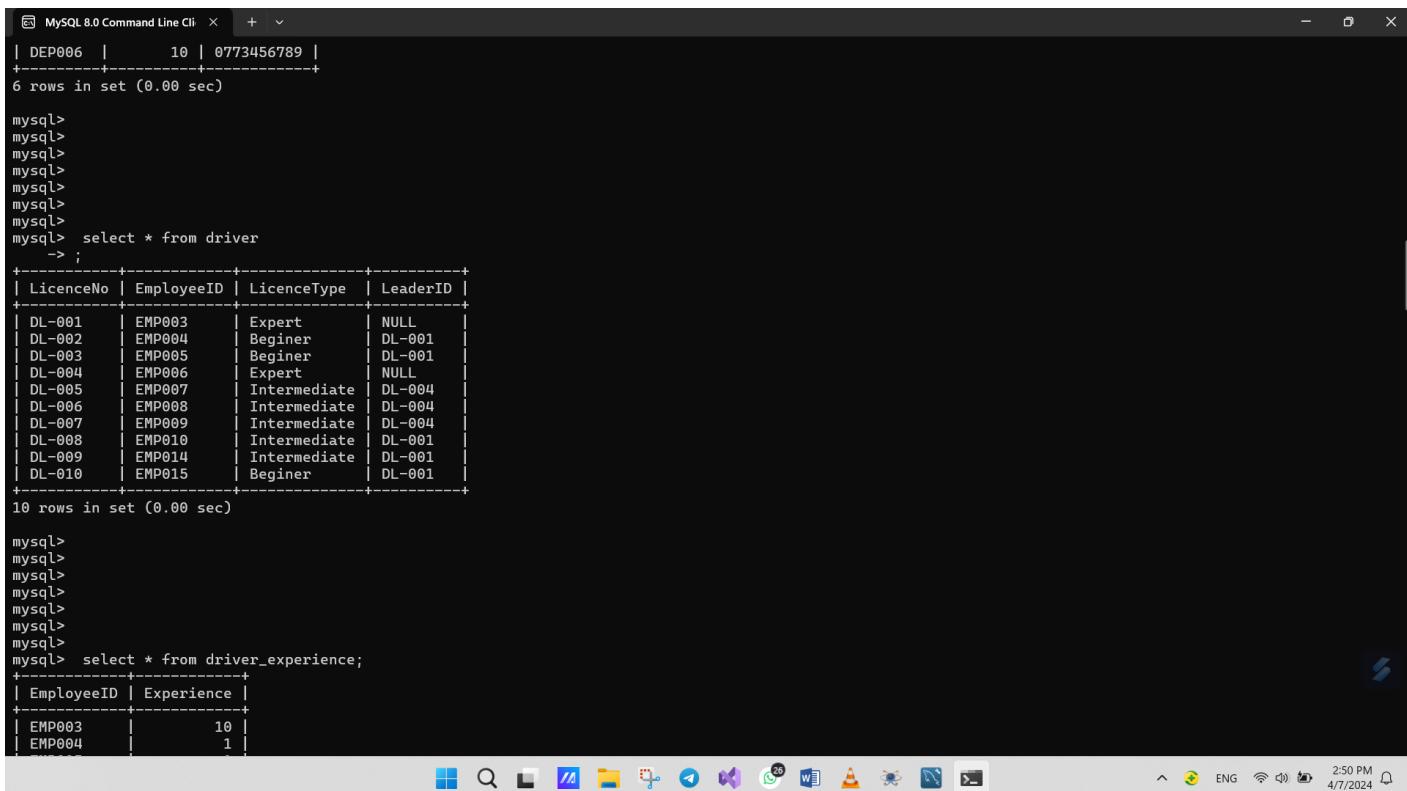
86
87
88 -- Inserting records into the DRIVER table
89 • INSERT INTO DRIVER (LicenceNo, EmployeeID, LicenceType,LeaderID)
90 VALUES
91 ('DL-001', 'EMP003', 'Expert',NULL),
92 ('DL-002', 'EMP004', 'Beginer','DL-001'),
93 ('DL-003', 'EMP005', 'Beginer','DL-001'),
94 ('DL-004', 'EMP006','Expert',NULL),
95 ('DL-005', 'EMP007', 'Intermediate','DL-004'),
96 ('DL-006', 'EMP008', 'Intermediate','DL-004'),
97 ('DL-007', 'EMP009', 'Intermediate','DL-004'),
98 ('DL-008', 'EMP010', 'Intermediate','DL-001'),
99 ('DL-009', 'EMP014', 'Intermediate','DL-001'),
100 ('DL-010', 'EMP015', 'Beginer','DL-001');

101
102
103

```

To the right of the editor, a status bar displays the message: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help." Below the editor, the "Output" pane shows the results of the insertions:

#	Action	Time	Message	Duration / Fetch
36	INSERT INTO MAINTENANCE	14:37:33	(ServiceID, BusID, LastTireChange, LastTireBrand, LastBrakeChange, LastOilC... 10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0)	0.031 sec
37	INSERT INTO SERVICE_COST	14:37:33	(ServiceID, Amount, Payment_method) VALUES (SERV001, 1500.00, 'Credit... 10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0)	0.000 sec
38	INSERT INTO FUEL	14:37:33	(BillNo, BusID, FillDate, Volume) VALUES (BILL001, 'BUS-001', 2023-05-10, 100.5), (BIL... 10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0)	0.016 sec



The screenshot shows the MySQL 8.0 Command Line Cli window. The command entered was:

```
mysql> select * from driver;
```

The output shows the data inserted into the DRIVER table:

LicenceNo	EmployeeID	LicenceType	LeaderID
DL-001	EMP003	Expert	NULL
DL-002	EMP004	Beginer	DL-001
DL-003	EMP005	Beginer	DL-001
DL-004	EMP006	Expert	NULL
DL-005	EMP007	Intermediate	DL-004
DL-006	EMP008	Intermediate	DL-004
DL-007	EMP009	Intermediate	DL-004
DL-008	EMP010	Intermediate	DL-001
DL-009	EMP014	Intermediate	DL-001
DL-010	EMP015	Beginer	DL-001

10 rows in set (0.00 sec)

MySQL prompt: mysql>

MySQL prompt: mysql> select * from driver_experience;

EmployeeID	Experience
EMP003	10
EMP004	1

5. Driver_experience Table

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

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Local instance wampmysqld64, Schemas (foe23, foe23_1, sys).
- SQL Editor:** SQL File 3*, SQL File 4*, SQL File 4* (query). The code is as follows:


```

101
102
103
104 -- Inserting records into the DRIVER_EXPERENCE table
105 • INSERT INTO DRIVER_EXPERENCE(EmployeeID,Experience)
106   VALUES
107   ('EMP003',10),
108   ('EMP004',1),
109   ('EMP005',1),
110   ('EMP006',6),
111   ('EMP007',2),
112   ('EMP008',3),
113   ('EMP009',1),
114   ('EMP010',4),
115   ('EMP014',5),
116   ('EMP015',1);
117
118
      
```
- Output Window:** Shows the results of three recent queries:

Action	Time	Message	Duration / Fetch
36	14:37:33	INSERT INTO MAINTENANCE (ServiceID, BusID, LastTireChange, LastTireBrand, LastBrakeChange, LastOilChange) VALUES ('SERV001', 1500.00, 'Credit', 'Pirelli', 'Brembo', 'Mobil 1')	0.031 sec
37	14:37:33	INSERT INTO SERVICE_COST (ServiceID, Amount, Payment_method) VALUES ('SERV001', 1500.00, 'Credit')	0.000 sec
38	14:37:33	INSERT INTO FUEL (BillNo, BusID, RDate, Volume) VALUES ('BILL001', 'BUS-001', '2023-05-10', 100.5)	0.016 sec

The screenshot shows the MySQL Command Line Client interface with the following details:

- Output:**

EmployeeID	Experience
EMP003	10
EMP004	1
EMP005	1
EMP006	6
EMP007	2
EMP008	3
EMP009	1
EMP010	4
EMP014	5
EMP015	1
- Output:**

EmployeeID	Salary	Position	AddressNo	AddressStreet	AddressCity	FirstName	LastName	Birthday
EMP001	60000.00	Manager	123	Galle Road	Colombo	Saman	Perera	1980-05-15
EMP002	62000.00	Manager	456	Mahaweli Mawatha	Kandy	Malini	Silva	1978-09-20
EMP003	48000.00	Driver	789	Dutch Hospital Street	Galle	Nimal	Fernando	1985-03-10

6. Supervisor Table

The screenshot shows the MySQL Workbench interface. In the center, there are four tabs labeled 'SQL File 3*', 'SQL File 4*', 'SQL File 4*', and 'query'. The 'query' tab contains the following SQL code:

```
122
123  -- Inserting records into the SUPERVISOR table
124 • INSERT INTO SUPERVISOR (SupervisorID, EmployeeID)
125   VALUES
126   ('SV-001', 'EMP011'),
127   ('SV-002', 'EMP012'),
128   ('SV-003', 'EMP013'),
129   ('SV-004', 'EMP020'),
130   ('SV-005', 'EMP021'),
131   ('SV-006', 'EMP022');
132
133
134
135
136
137
138
```

In the bottom right corner of the main window, there is a note: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."

At the bottom of the screen, the Windows taskbar is visible with various icons.

The screenshot shows the MySQL 8.0 Command Line Client window. It displays the results of two queries:

```
| SERV006 | 1600.00 | Cash
| SERV007 | 1750.00 | Credit Card
| SERV008 | 1950.00 | Bank Transfer
| SERV009 | 1850.00 | Cash
| SERV010 | 2100.00 | Credit Card
+-----+
10 rows in set (0.01 sec)

mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> select * from supervisor;
+-----+-----+
| SupervisorID | EmployeeID |
+-----+-----+
| SV-001       | EMP011    |
| SV-002       | EMP012    |
| SV-003       | EMP013    |
| SV-004       | EMP020    |
| SV-005       | EMP021    |
| SV-006       | EMP022    |
+-----+
6 rows in set (0.00 sec)
```

At the bottom of the screen, the Windows taskbar is visible with various icons.

7. Route Table

The screenshot shows the MySQL Workbench interface. In the center, there is a SQL editor window with the following code:

```

136
137
138
139
140    -- Inserting records into the ROUTE table
141 • INSERT INTO ROUTE (RouteNo, SupervisorID, Distance, Duration, Start, End)
142     VALUES
143     (1, 'SV-001', 100, 120, 'Colombo', 'Kandy'),
144     (2, 'SV-002', 150, 150, 'Colombo', 'Matara'),
145     (3, 'SV-003', 80, 90, 'Gampaha', 'Polonnaruwa'),
146     (4, 'SV-001', 200, 180, 'Kadawatha', 'Jaffna'),
147     (5, 'SV-002', 120, 130, 'Colombo', 'Ratnapura'),
148     (6, 'SV-003', 140, 120, 'Colombo', 'Galle');
149
150
151
152
153

```

The right side of the interface has a "SQLAdditions" panel with the message: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."

At the bottom, the "Output" tab shows the results of the insertions:

#	Time	Action	Message	Duration / Fetch
36	14:37:33	INSERT INTO MAINTENANCE (ServiceID, BusID, LastTireChange, LastTireBrand, LastBrakeChange, LastOilChange, LastMaintenanceDate)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.031 sec
37	14:37:33	INSERT INTO SERVICE_COST (ServiceID, Amount, Payment_method) VALUES ('SERV001', 1500.00, 'Credit Card')	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
38	14:37:33	INSERT INTO FUEL (BillNo, BusID, FillDate, Volume) VALUES ('BILL001', 'BUS-001', 2023-05-10, 100.5), ('BILL002', 'BUS-002', 2023-05-10, 100.5), ('BILL003', 'BUS-003', 2023-05-10, 100.5)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.016 sec

The screenshot shows the MySQL Command Line Client window. It displays the following output:

```

| EMP016 | Human Resources | 5 |
| EMP017 | Transport | 5 |
| EMP018 | Director | 8 |
| EMP019 | General | 4 |
+-----+
6 rows in set (0.00 sec)

mysql>
mysql>
mysql>
mysql>
mysql>
mysql> select * from route;
+-----+-----+-----+-----+-----+-----+
| RouteNo | SupervisorID | Distance | Duration | Start | End |
+-----+-----+-----+-----+-----+-----+
| 1 | SV-001 | 100 | 120 | Colombo | Kandy |
| 2 | SV-002 | 150 | 150 | Colombo | Matara |
| 3 | SV-003 | 80 | 90 | Gampaha | Polonnaruwa |
| 4 | SV-001 | 200 | 180 | Kadawatha | Jaffna |
| 5 | SV-002 | 120 | 130 | Colombo | Ratnapura |
| 6 | SV-003 | 140 | 120 | Colombo | Galle |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.01 sec)

mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> select * from service_cost;
+-----+-----+-----+
| ServiceID | Amount | Payment_method |
+-----+-----+-----+
| SERV001 | 1500.00 | Credit Card |
| SERV002 | 1800.00 | Cash |
| SERV003 | 2000.00 | Bank Transfer |
+-----+-----+-----+

```

8. Bus_schedule Table

The screenshot shows the MySQL Workbench interface. In the center, there is a SQL editor window titled "query". The code in the editor is as follows:

```
153
154
155
156    -- Inserting records into the BUS_SCHEDULE table
157 • INSERT INTO BUS_SCHEDULE (ScheduleID, RouteNo, StartTime, EndTime)
158     VALUES
159     ('SCH-001', 1, '08:00:00', '10:00:00'),
160     ('SCH-002', 2, '09:00:00', '12:00:00'),
161     ('SCH-003', 3, '10:00:00', '12:30:00'),
162     ('SCH-004', 4, '11:00:00', '14:30:00'),
163     ('SCH-005', 5, '12:00:00', '14:00:00'),
164     ('SCH-006', 1, '13:00:00', '15:00:00'),
165     ('SCH-007', 2, '14:00:00', '17:00:00'),
166     ('SCH-008', 3, '15:00:00', '17:30:00'),
167     ('SCH-009', 4, '16:00:00', '19:30:00'),
168     ('SCH-010', 5, '17:00:00', '19:00:00');
169
170
```

In the bottom right corner of the SQL editor, there is a message: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."

Below the SQL editor, there is an "Output" pane showing the results of the insertion queries:

Action	Time	Action	Message	Duration / Fetch
36	14:37:33	INSERT INTO MAINTENANCE (ServiceID, BusID, LastTireChange, LastBrakeChange, LastOilChange)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.031 sec
37	14:37:33	INSERT INTO SERVICE_COST (ServiceID, Amount, Payment_method) VALUES ('SERV001', 1500.00, 'Credit')	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
38	14:37:33	INSERT INTO FUEL (BillNo, BusID, RfDate, Volume) VALUES ('BILL001', 'BUS-001', '2023-05-10', 100.5)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.016 sec

The screenshot shows the MySQL Command Line Client window. The session starts with several empty lines of MySQL prompt (mysql>). Then, the user runs a query to select all columns from the bus_schedule table:

```
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> select * from bus_schedule;
```

An error message follows, indicating that the table 'bus_schedule' does not exist:

```
ERROR 1146 (42S02): Table 'private_bus_company.bus_schedule' doesn't exist
```

After this, the user runs the same query again, and the system returns the expected result set:

ScheduleID	RouteNo	StartTime	EndTime
SCH-001	1	08:00:00	10:00:00
SCH-002	2	09:00:00	12:00:00
SCH-003	3	10:00:00	12:30:00
SCH-004	4	11:00:00	14:30:00
SCH-005	5	12:00:00	14:00:00
SCH-006	1	13:00:00	15:00:00
SCH-007	2	14:00:00	17:00:00
SCH-008	3	15:00:00	17:30:00
SCH-009	4	16:00:00	19:30:00
SCH-010	5	17:00:00	19:00:00

Finally, the user runs a query to select all columns from the depot table:

```
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> select * from depot;
```

The system returns the following result set:

DepotID	Capacity	ContactNo

9. Depot Table

The screenshot shows the MySQL Workbench interface. In the center, a query editor window displays the following SQL code:

```
171
172
173 -- Insert records into the DEPOT table
174 • INSERT INTO DEPOT (DepotID, Capacity, ContactNo)
175 VALUES
176 ('DEP001', 10, '0711234567'),
177 ('DEP002', 5, '0712345678'),
178 ('DEP003', 5, '0762345678'),
179 ('DEP004', 3, '0722345678'),
180 ('DEP005', 4, '0742345678'),
181 ('DEP006', 10, '0773456789');
182
183
184
185
186
187
188
```

The right side of the interface includes a status bar with the message "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help." Below the status bar, there are tabs for "Context Help" and "Snippets". At the bottom, a toolbar contains various icons for file operations.

The screenshot shows the MySQL 8.0 Command Line Cli interface. The command line shows the following session:

```
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> select * from depot;
```

The result of the query is displayed as a table:

DepotID	Capacity	ContactNo
DEP001	10	0711234567
DEP002	5	0712345678
DEP003	5	0762345678
DEP004	3	0722345678
DEP005	4	0742345678
DEP006	10	0773456789

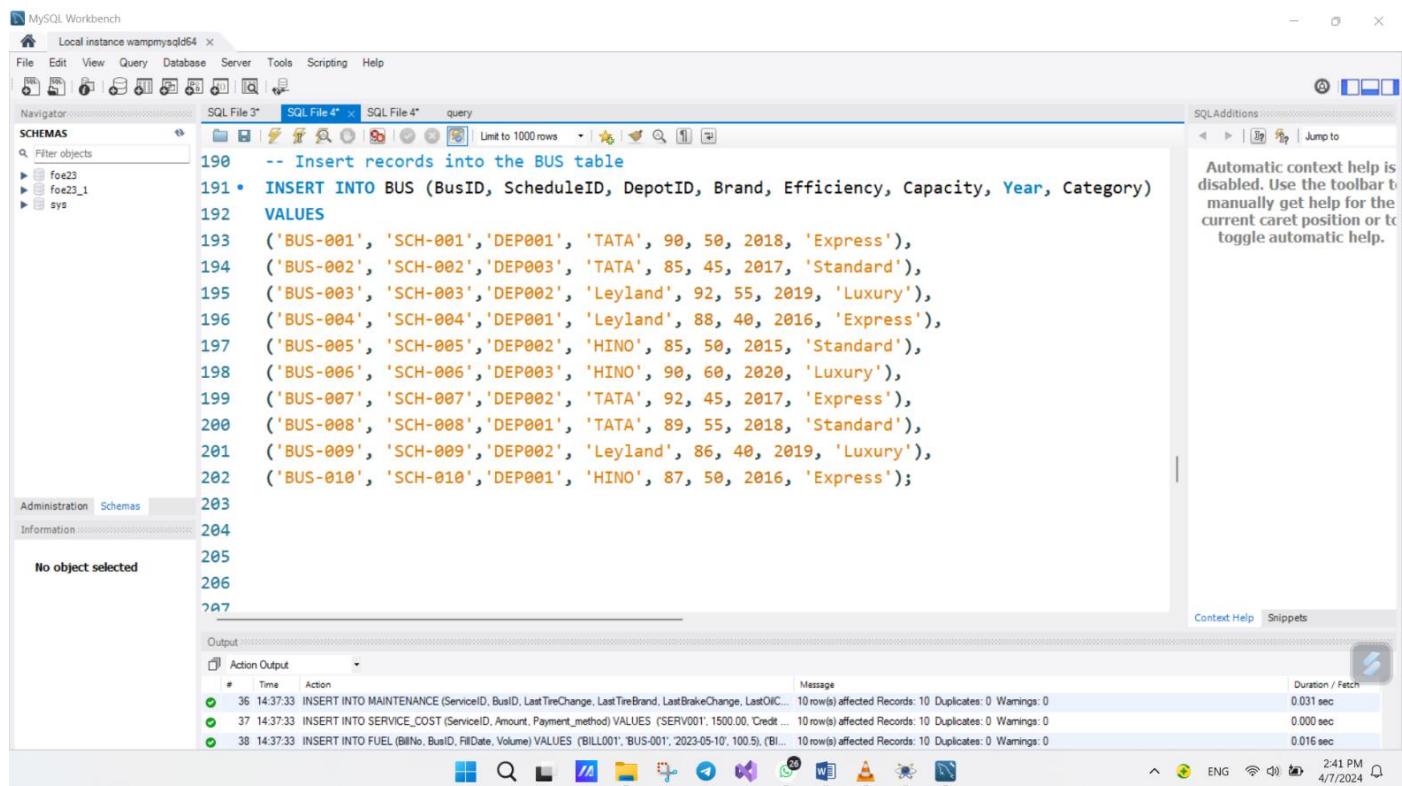
Below this, another command is shown:

```
mysql> select * from driver
-> ;
```

The result of this query is also displayed as a table:

LicenceNo	EmployeeID	LicenceType	LeaderID
DL-001	EMP003	Expert	NULL
DL-002	EMP004	Beginer	DL-001
DL-003	EMP005	Beginer	DL-001
DL-004	EMP006	Expert	NULL
DL-005	EMP007	Intermediate	DL-004
DL-006	EMP008	Intermediate	DL-004
DL-007	EMP009	Intermediate	DL-004
DL-008	EMP010	Intermediate	DL-001

10. Bus Table



The screenshot shows the MySQL Workbench interface. The SQL Editor tab is active, displaying the following SQL code:

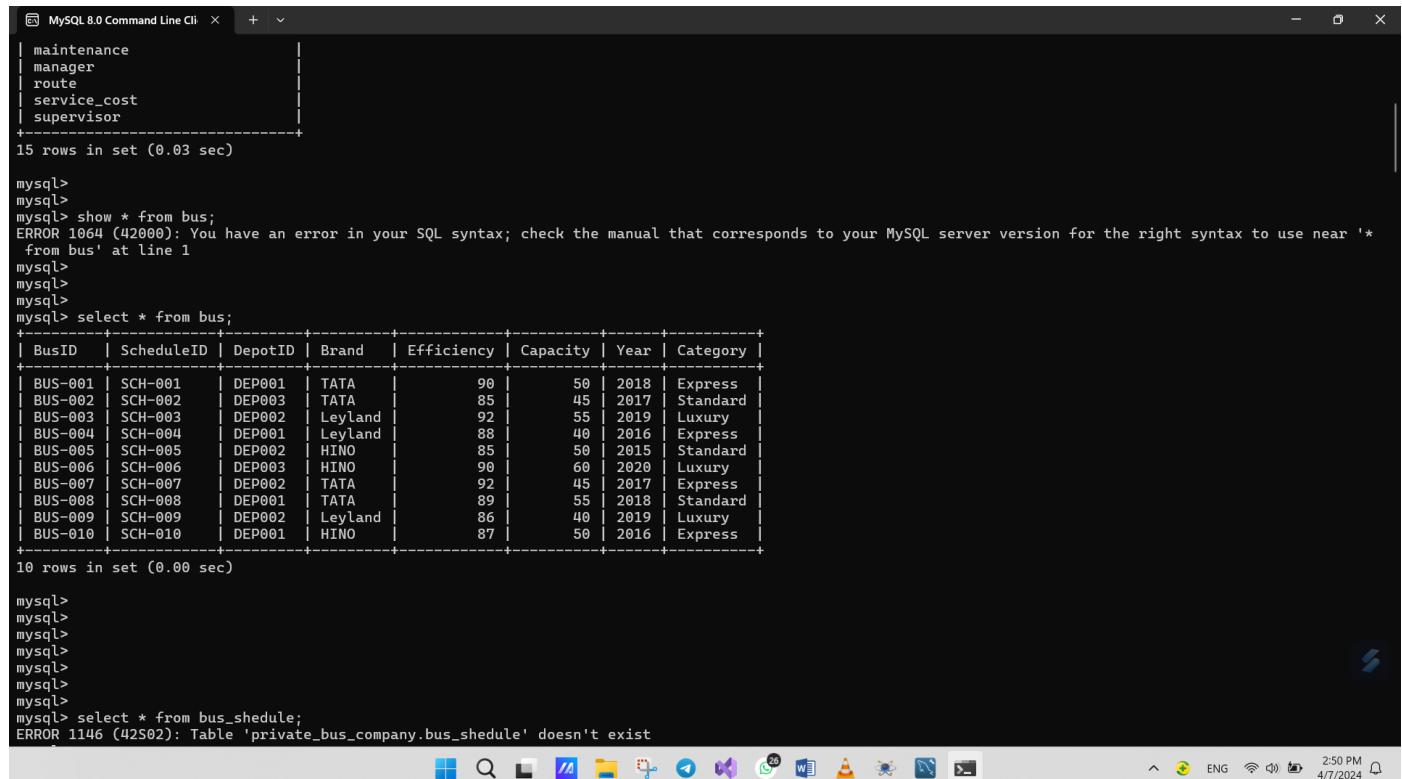
```

190 -- Insert records into the BUS table
191 • INSERT INTO BUS (BusID, ScheduleID, DepotID, Brand, Efficiency, Capacity, Year, Category)
192     VALUES
193     ('BUS-001', 'SCH-001', 'DEP001', 'TATA', 90, 50, 2018, 'Express'),
194     ('BUS-002', 'SCH-002', 'DEP003', 'TATA', 85, 45, 2017, 'Standard'),
195     ('BUS-003', 'SCH-003', 'DEP002', 'Leyland', 92, 55, 2019, 'Luxury'),
196     ('BUS-004', 'SCH-004', 'DEP001', 'Leyland', 88, 40, 2016, 'Express'),
197     ('BUS-005', 'SCH-005', 'DEP002', 'HINO', 85, 50, 2015, 'Standard'),
198     ('BUS-006', 'SCH-006', 'DEP003', 'HINO', 90, 60, 2020, 'Luxury'),
199     ('BUS-007', 'SCH-007', 'DEP002', 'TATA', 92, 45, 2017, 'Express'),
200     ('BUS-008', 'SCH-008', 'DEP001', 'TATA', 89, 55, 2018, 'Standard'),
201     ('BUS-009', 'SCH-009', 'DEP002', 'Leyland', 86, 40, 2019, 'Luxury'),
202     ('BUS-010', 'SCH-010', 'DEP001', 'HINO', 87, 50, 2016, 'Express');
203
204
205
206
207

```

The Output pane shows the results of the insertions:

Action	Time	Action	Message	Duration / Fetch
36	14:37:33	INSERT INTO MAINTENANCE (ServiceID, BusID, LastTireChange, LastTireBrand, LastBrakeChange, LastOilChange)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.031 sec
37	14:37:33	INSERT INTO SERVICE_COST (ServiceID, Amount, Payment_method) VALUES (SERV001, 1500.00, Credit ...)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
38	14:37:33	INSERT INTO FUEL (BillNo, BusID, FillDate, Volume) VALUES (BILL001, 'BUS-001', 2023-05-10, 100.5), (BIL...	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.016 sec



The screenshot shows the MySQL 8.0 Command Line Cli window. The command history includes:

```

| maintenance
| manager
| route
| service_cost
| supervisor
+-----+
15 rows in set (0.03 sec)

mysql>
mysql>
mysql> show * from bus;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'from bus' at line 1
mysql>
mysql>
mysql>
mysql> select * from bus;
+-----+
| BusID | ScheduleID | DepotID | Brand | Efficiency | Capacity | Year | Category |
+-----+
| BUS-001 | SCH-001 | DEP001 | TATA | 90 | 50 | 2018 | Express |
| BUS-002 | SCH-002 | DEP003 | TATA | 85 | 45 | 2017 | Standard |
| BUS-003 | SCH-003 | DEP002 | LeyLand | 92 | 55 | 2019 | Luxury |
| BUS-004 | SCH-004 | DEP001 | LeyLand | 88 | 40 | 2016 | Express |
| BUS-005 | SCH-005 | DEP002 | HINO | 85 | 50 | 2015 | Standard |
| BUS-006 | SCH-006 | DEP003 | HINO | 90 | 60 | 2020 | Luxury |
| BUS-007 | SCH-007 | DEP002 | TATA | 92 | 45 | 2017 | Express |
| BUS-008 | SCH-008 | DEP001 | TATA | 89 | 55 | 2018 | Standard |
| BUS-009 | SCH-009 | DEP002 | LeyLand | 86 | 40 | 2019 | Luxury |
| BUS-010 | SCH-010 | DEP001 | HINO | 87 | 50 | 2016 | Express |
+-----+
10 rows in set (0.00 sec)

mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> select * from bus_schedule;
ERROR 1146 (42S02): Table 'private_bus_company.bus_schedule' doesn't exist

```

11. Fare Table

MySQL Workbench

Local instance wampmysqld54

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

foe23
foe23_1
sys

SQL File 3* SQL File 4* query

208 -- Insert records into the FARE table with varying price rates
209 • INSERT INTO FARE (FareID, RouteNo, PriceRate, TicketType)
210 VALUES
211 ('F1T1', 1, 250.00, 'full'),
212 ('F1T2', 1, 150.00, 'half'),
213 ('F2T1', 2, 300.00, 'full'),
214 ('F2T2', 2, 180.00, 'half'),
215 ('F3T1', 3, 280.00, 'full'),
216 ('F3T2', 3, 160.00, 'half'),
217 ('F4T1', 4, 220.00, 'full'),
218 ('F4T2', 4, 130.00, 'half'),
219 ('F5T1', 5, 270.00, 'full'),
220 ('F5T2', 5, 160.00, 'half');

Administration Schemas

Information

No object selected

Output

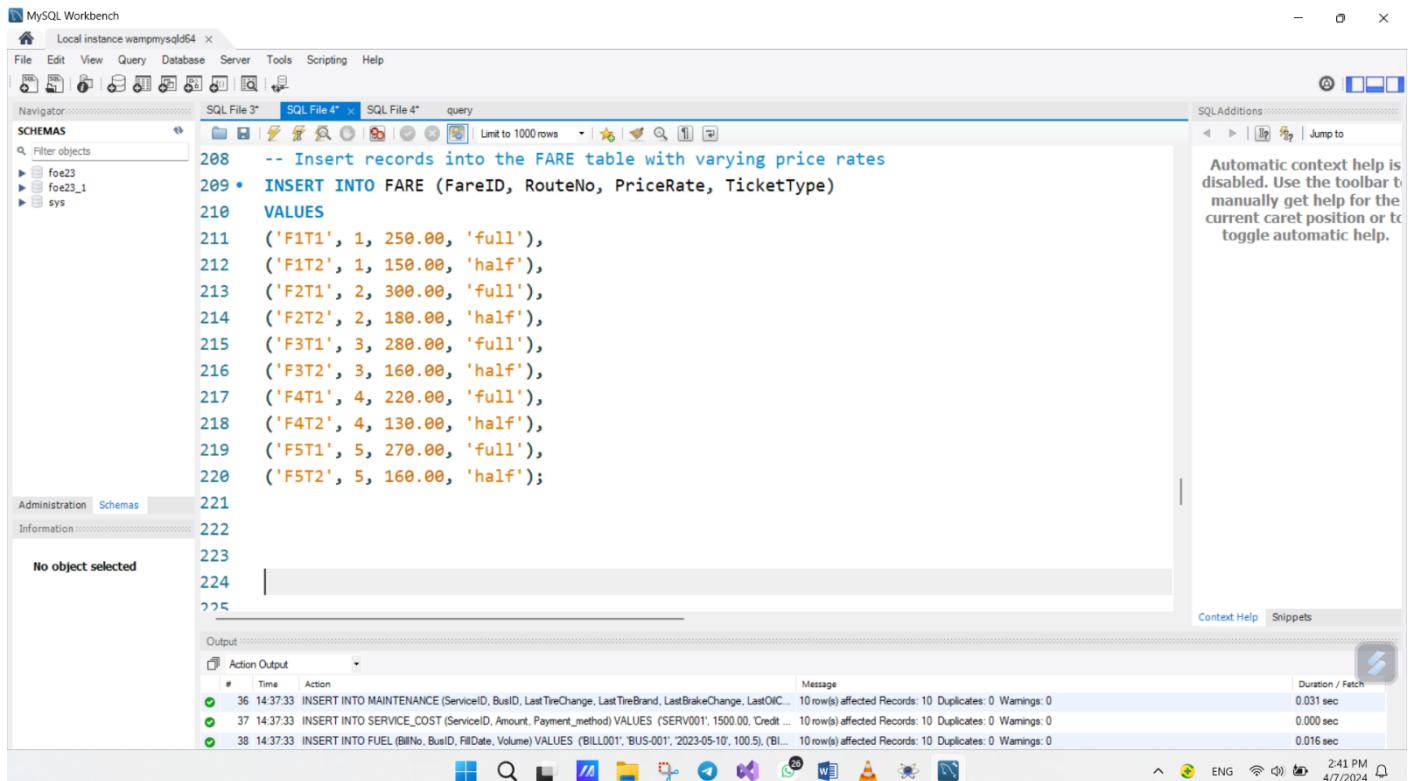
Action Output

Time Action Message Duration / Fetch

36 14:37:33 INSERT INTO MAINTENANCE (ServiceID, BusID, LastTireChange, LastTireBrand, LastBrakeChange, LastOIC... 10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0 0.031 sec
37 14:37:33 INSERT INTO SERVICE_COST (ServiceID, Amount, Payment_Method) VALUES (SERV001, 1500.00, 'Credit ... 10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0 0.000 sec
38 14:37:33 INSERT INTO FUEL (BillNo, BusID, FillDate, Volume) VALUES ('BILL001', 'BUS-001', '2023-05-10', 100.5), ('BIL... 10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0 0.016 sec

Context Help Snippets

4/7/2024 2:41 PM



MySQL 8.0 Command Line Cli

EMP018		0755678121	
EMP019		0745672121	
EMP020		0745658121	
EMP021		0745679121	
EMP022		0777899433	

+-----+-----+-----+

31 rows in set (0.00 sec)

mysql>

mysql>

mysql>

mysql>

mysql> select * from fare;

FareID	RouteNo	PriceRate	TicketType
F1T1	1	250.00	full
F1T2	1	150.00	half
F2T1	2	300.00	full
F2T2	2	180.00	half
F3T1	3	280.00	full
F3T2	3	160.00	half
F4T1	4	220.00	full
F4T2	4	130.00	half
F5T1	5	270.00	full
F5T2	5	160.00	half

+-----+-----+-----+

10 rows in set (0.00 sec)

mysql>

mysql>

mysql>

mysql>

mysql>

mysql>

mysql>

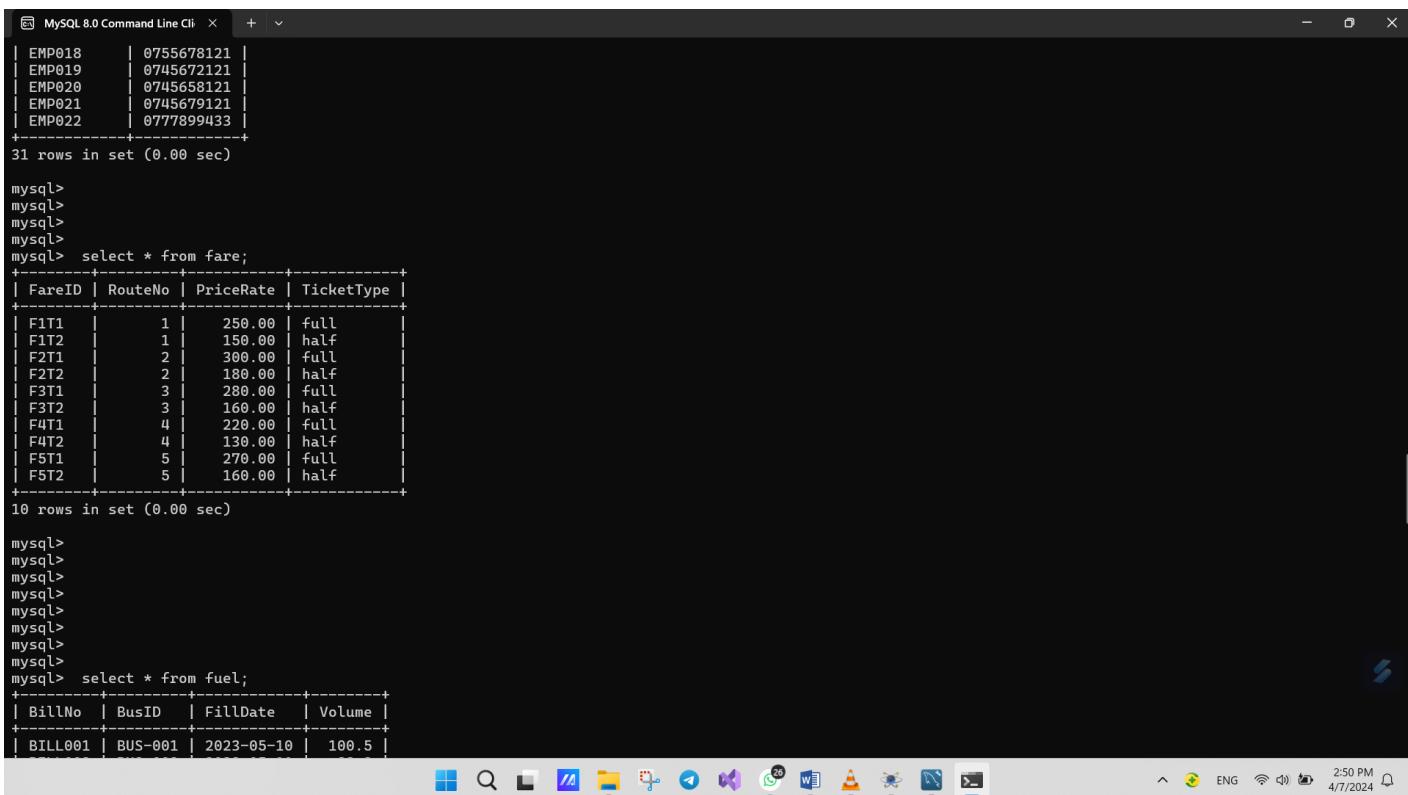
mysql>

mysql> select * from fuel;

BillNo	BusID	FillDate	Volume
BILL001	BUS-001	2023-05-10	100.5

+-----+-----+-----+

2:50 PM 4/7/2024



12. Maintenance Table

MySQL Workbench

Local instance wampmysqld64

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

SCHEMAS: fo23, fo23_1, sys

SQL File 3*, SQL File 4*, SQL File 4* query

```

226 -- Insert records into the MAINTENANCE table
227 • INSERT INTO MAINTENANCE (ServiceID, BusID, LastTireChange, LastTireBrand, LastBrakeChange, LastOilChange, LastOilBrand, LastService)
228 VALUES
229 ('SERV001', 'BUS-001', '2023-05-20', 'Michelin', '2023-04-15', '2023-03-10', 'Shell', '2023-05-25'),
230 ('SERV002', 'BUS-002', '2023-04-18', 'Bridgestone', '2023-03-12', '2023-02-20', 'Castrol', '2023-05-28'),
231 ('SERV003', 'BUS-003', '2023-06-05', 'Goodyear', '2023-05-02', '2023-04-10', 'Mobil', '2023-05-30'),
232 ('SERV004', 'BUS-004', '2023-05-25', 'Continental', '2023-04-20', '2023-03-15', 'Total', '2023-06-01'),
233 ('SERV005', 'BUS-005', '2023-04-30', 'Pirelli', '2023-03-25', '2023-02-28', 'BP', '2023-06-02'),
234 ('SERV006', 'BUS-006', '2023-06-10', 'Firestone', '2023-05-05', '2023-04-20', 'ExxonMobil', '2023-06-03'),
235 ('SERV007', 'BUS-007', '2023-05-15', 'Yokohama', '2023-04-10', '2023-03-05', 'Chevron', '2023-06-05'),
236 ('SERV008', 'BUS-008', '2023-06-20', 'Hankook', '2023-05-15', '2023-04-10', 'Shell', '2023-06-07'),
237 ('SERV009', 'BUS-009', '2023-05-28', 'Dunlop', '2023-04-25', '2023-03-20', 'Castrol', '2023-06-10'),
238 ('SERV010', 'BUS-010', '2023-06-02', 'Michelin', '2023-05-30', '2023-04-25', 'Mobil', '2023-06-12')
239
240
241
242
243
244
245
246
247
248
249

```

Administration Schemas

No object selected

Output:

#	Time	Action	Message	Duration / Fetch
36	14:37:33	INSERT INTO MAINTENANCE (ServiceID, BusID, LastTireChange, LastTireBrand, LastBrakeChange, LastOilChange, LastOilBrand, LastService)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.031 sec
37	14:37:33	INSERT INTO SERVICE_COST (ServiceID, Amount, Payment_method) VALUES ('SERV001', 1500.00, 'Credit Card')	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
38	14:37:33	INSERT INTO FUEL (BillNo, BusID, FillDate, Volume) VALUES ('BILL001', 'BUS-001', '2023-05-10', 100.5), ('BILL002', 'BUS-002', '2023-05-10', 100.5), ('BILL003', 'BUS-003', '2023-05-10', 100.5), ('BILL004', 'BUS-004', '2023-05-10', 100.5), ('BILL005', 'BUS-005', '2023-05-10', 100.5), ('BILL006', 'BUS-006', '2023-05-10', 100.5), ('BILL007', 'BUS-007', '2023-05-10', 100.5), ('BILL008', 'BUS-008', '2023-05-10', 100.5), ('BILL009', 'BUS-009', '2023-05-10', 100.5), ('BILL010', 'BUS-010', '2023-05-10', 100.5)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.016 sec

Context Help Snippets

MySQL 8.0 Command Line Cli

BILL009 24200.00 Credit Card
BILL010 26100.00 Cash

10 rows in set (0.00 sec)

mysql>

mysql>

mysql>

mysql>

mysql>

mysql>

mysql>

mysql> select * from maintenance;

ServiceID	BusID	LastTireChange	LastTireBrand	LastBrakeChange	LastOilChange	LastOilBrand	LastService
SERV001	BUS-001	2023-05-20	Michelin	2023-04-15	2023-03-10	Shell	2023-05-25
SERV002	BUS-002	2023-04-18	Bridgestone	2023-03-12	2023-02-20	Castrol	2023-05-28
SERV003	BUS-003	2023-06-05	Goodyear	2023-05-02	2023-04-10	Mobil	2023-05-30
SERV004	BUS-004	2023-05-25	Continental	2023-04-20	2023-03-15	Total	2023-06-01
SERV005	BUS-005	2023-04-30	Pirelli	2023-03-25	2023-02-28	BP	2023-06-02
SERV006	BUS-006	2023-06-10	Firestone	2023-05-05	2023-04-20	ExxonMobil	2023-06-03
SERV007	BUS-007	2023-05-15	Yokohama	2023-04-10	2023-03-05	Chevron	2023-06-05
SERV008	BUS-008	2023-06-20	Hankook	2023-05-15	2023-04-10	Shell	2023-06-07
SERV009	BUS-009	2023-05-28	Dunlop	2023-04-25	2023-03-20	Castrol	2023-06-10
SERV010	BUS-010	2023-06-02	Michelin	2023-05-30	2023-04-25	Mobil	2023-06-12

10 rows in set (0.00 sec)

mysql>

mysql>

mysql>

mysql>

mysql>

mysql>

mysql>

mysql>

mysql>

mysql> select * from manager;

EmployeeID	Area	Experience
EMP001	Operations	5
EMP002	Finance	6

13. Service_cost Table

The screenshot shows the MySQL Workbench interface. In the SQL Editor tab, the following SQL code is displayed:

```
250 -- Insert records into the SERVICE_COST table
251 • INSERT INTO SERVICE_COST (ServiceID, Amount, Payment_method)
252 VALUES
253 ('SERV001', 1500.00, 'Credit Card'),
254 ('SERV002', 1800.00, 'Cash'),
255 ('SERV003', 2000.00, 'Bank Transfer'),
256 ('SERV004', 1700.00, 'Cash'),
257 ('SERV005', 1900.00, 'Credit Card'),
258 ('SERV006', 1600.00, 'Cash'),
259 ('SERV007', 1750.00, 'Credit Card'),
260 ('SERV008', 1950.00, 'Bank Transfer'),
261 ('SERV009', 1850.00, 'Cash'),
262 ('SERV010', 2100.00, 'Credit Card');
263
264
265
266
267
268
269
```

In the Output tab, the results of the insertions are shown:

Action	Time	Message	Duration / Fetch
36	14:37:33	INSERT INTO MAINTENANCE (ServiceID, BusID, LastTireChange, LastBrakeChange, LastOilChange) VALUES ('SERV001', 1, '2023-05-10', '2023-05-10', '2023-05-10')	0.031 sec
37	14:37:33	INSERT INTO SERVICE_COST (ServiceID, Amount, Payment_method) VALUES ('SERV001', 1500.00, 'Credit Card')	0.000 sec
38	14:37:33	INSERT INTO FUEL (BillNo, BusID, FillDate, Volume) VALUES ('BILL001', 'BUS-001', '2023-05-10', 100.5)	0.016 sec

The screenshot shows the MySQL Command Line Client window. The user has run the following queries:

```
mysql> select * from service_cost;
+-----+-----+-----+
| ServiceID | Amount | Payment_method |
+-----+-----+-----+
| SERV001 | 1500.00 | Credit Card |
| SERV002 | 1800.00 | Cash |
| SERV003 | 2000.00 | Bank Transfer |
| SERV004 | 1700.00 | Cash |
| SERV005 | 1900.00 | Credit Card |
| SERV006 | 1600.00 | Cash |
| SERV007 | 1750.00 | Credit Card |
| SERV008 | 1950.00 | Bank Transfer |
| SERV009 | 1850.00 | Cash |
| SERV010 | 2100.00 | Credit Card |
+-----+-----+-----+
10 rows in set (0.01 sec)

mysql> select * from supervisor;
+-----+-----+
| SupervisorID | EmployeeID |
+-----+-----+
| SV-001 | EMP011 |
| SV-002 | EMP012 |
| SV-003 | EMP013 |
| SV-004 | EMP020 |
+-----+-----+
```

14. Fuel Table

The screenshot shows the MySQL Workbench interface. In the center, there is a code editor window titled "query" containing the following SQL script:

```
271 -- Insert records into the FUEL table
272 • INSERT INTO FUEL (BillNo, BusID, FillDate, Volume)
273 VALUES
274 ('BILL001', 'BUS-001', '2023-05-10', 100.5),
275 ('BILL002', 'BUS-002', '2023-05-11', 98.2),
276 ('BILL003', 'BUS-003', '2023-05-12', 105.8),
277 ('BILL004', 'BUS-004', '2023-05-13', 102.3),
278 ('BILL005', 'BUS-005', '2023-05-14', 97.6),
279 ('BILL006', 'BUS-006', '2023-05-15', 101.1),
280 ('BILL007', 'BUS-007', '2023-05-16', 99.9),
281 ('BILL008', 'BUS-008', '2023-05-17', 103.5),
282 ('BILL009', 'BUS-009', '2023-05-18', 96.8),
283 ('BILL010', 'BUS-010', '2023-05-19', 104.7);
284
285
286
287
288
289
290
```

In the bottom right corner of the code editor, there is a note: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."

Below the code editor, the "Output" pane displays the results of the executed statements:

Action	Time	Action	Message	Duration / Fetch
36	14:37:33	INSERT INTO MAINTENANCE (ServiceID, BusID, LastTireChange, LastBrakeChange, LastOilChange)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.031 sec
37	14:37:33	INSERT INTO SERVICE_COST (ServiceID, Amount, Payment_method) VALUES ('SERV001', 1500.00, 'Credit Card')	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
38	14:37:33	INSERT INTO FUEL (BillNo, BusID, FillDate, Volume) VALUES ('BILL001', 'BUS-001', '2023-05-10', 100.5), ('BILL002', 'BUS-002', '2023-05-11', 98.2), ('BILL003', 'BUS-003', '2023-05-12', 105.8), ('BILL004', 'BUS-004', '2023-05-13', 102.3), ('BILL005', 'BUS-005', '2023-05-14', 97.6), ('BILL006', 'BUS-006', '2023-05-15', 101.1), ('BILL007', 'BUS-007', '2023-05-16', 99.9), ('BILL008', 'BUS-008', '2023-05-17', 103.5), ('BILL009', 'BUS-009', '2023-05-18', 96.8), ('BILL010', 'BUS-010', '2023-05-19', 104.7)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.016 sec

The screenshot shows the MySQL Command Line Client interface. It displays the results of several SQL queries:

```
| FST1 |      5 | 270.00 | full      |
| FST2 |      5 | 160.00 | half      |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> select * from fuel;
+-----+-----+-----+-----+
| BillNo | BusID | FillDate | Volume |
+-----+-----+-----+-----+
| BILL001 | BUS-001 | 2023-05-10 | 100.5 |
| BILL002 | BUS-002 | 2023-05-11 | 98.2  |
| BILL003 | BUS-003 | 2023-05-12 | 105.8 |
| BILL004 | BUS-004 | 2023-05-13 | 102.3 |
| BILL005 | BUS-005 | 2023-05-14 | 97.6   |
| BILL006 | BUS-006 | 2023-05-15 | 101.1 |
| BILL007 | BUS-007 | 2023-05-16 | 99.9   |
| BILL008 | BUS-008 | 2023-05-17 | 103.5 |
| BILL009 | BUS-009 | 2023-05-18 | 96.8   |
| BILL010 | BUS-010 | 2023-05-19 | 104.7 |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> select * from fuel_cost;
+-----+-----+-----+
| BillNo | Amount | Payment_method |
+-----+-----+-----+
| BILL001 | 25000.00 | Cash           |
| BILL002 | 24000.00 | Credit Card   |
+-----+-----+-----+
```

15. Fuel_cost Table

The screenshot shows the MySQL Workbench interface. In the center, there is a SQL editor tab titled "SQL File 4*" containing the following SQL code:

```
287
288
289
290
291 -- Insert records into the FUEL_COST table
292 • INSERT INTO FUEL_COST (BillNo, Amount, Payment_method)
293 VALUES
294 ('BILL001', 25000.00, 'Cash'),
295 ('BILL002', 24000.00, 'Credit Card'),
296 ('BILL003', 26400.00, 'Cash'),
297 ('BILL004', 25500.00, 'Cash'),
298 ('BILL005', 24400.00, 'Credit Card'),
299 ('BILL006', 25200.00, 'Cash'),
300 ('BILL007', 24900.00, 'Credit Card'),
301 ('BILL008', 25800.00, 'Cash'),
302 ('BILL009', 24200.00, 'Credit Card'),
303 ('BILL010', 26100.00, 'Cash');
304
305
306
```

In the bottom right corner of the SQL editor, there is a note: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."

Below the SQL editor, there is an "Output" pane showing the results of the executed queries:

Action	Time	Action	Message	Duration / Fetch
36	14:37:33	INSERT INTO MAINTENANCE (ServiceID, BusID, LastTireChange, LastBrakeChange, LastOilChange, LastOilBrand, LastService)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.031 sec
37	14:37:33	INSERT INTO SERVICE_COST (ServiceID, Amount, Payment_method) VALUES ('SERV001', 1500.00, 'Credit Card')	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
38	14:37:33	INSERT INTO FUEL (BillNo, BusID, FillDate, Volume) VALUES ('BILL001', 'BUS-001', '2023-05-10', 100.5)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.016 sec

The status bar at the bottom right of the Workbench window shows the time as 4/7/2024 2:42 PM.

The screenshot shows the MySQL 8.0 Command Line Cli window. It displays the following SQL queries and their results:

```
| BILL005 | BUS-005 | 2023-05-14 | 97.6 |
| BILL006 | BUS-006 | 2023-05-15 | 101.1 |
| BILL007 | BUS-007 | 2023-05-16 | 99.9 |
| BILL008 | BUS-008 | 2023-05-17 | 103.5 |
| BILL009 | BUS-009 | 2023-05-18 | 96.8 |
| BILL010 | BUS-010 | 2023-05-19 | 104.7 |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql>
mysql>
mysql>
mysql>
mysql>
mysql> select * from fuel_cost;
+-----+-----+-----+
| BillNo | Amount | Payment_method |
+-----+-----+-----+
| BILL001 | 25000.00 | Cash |
| BILL002 | 24000.00 | Credit Card |
| BILL003 | 26400.00 | Cash |
| BILL004 | 25500.00 | Cash |
| BILL005 | 24400.00 | Credit Card |
| BILL006 | 25200.00 | Cash |
| BILL007 | 24900.00 | Credit Card |
| BILL008 | 25800.00 | Cash |
| BILL009 | 24200.00 | Credit Card |
| BILL010 | 26100.00 | Cash |
+-----+-----+-----+
10 rows in set (0.00 sec)

mysql>
mysql>
mysql>
mysql>
mysql>
mysql> select * from maintenance;
+-----+-----+-----+-----+-----+-----+-----+
| ServiceID | BusID | LastTireChange | LastBrakeChange | LastOilChange | LastOilBrand | LastService |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

The status bar at the bottom right of the Command Line Cli window shows the time as 4/7/2024 2:50 PM.

3.3. UPDATE AND DELETE

1. Employee Table

```
MySQL 8.0 Command Line Cli  +  ×

22 rows in set (0.00 sec)

mysql> UPDATE EMPLOYEE SET Salary = 65000.00 WHERE EmployeeID = 'EMP001';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE EMPLOYEE SET Position = 'Senior Driver' WHERE EmployeeID = 'EMP003';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> DELETE FROM EMPLOYEE WHERE EmployeeID = 'EMP020';
Query OK, 1 row affected (0.04 sec)

mysql> select * from employee;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| EmployeeID | Salary | Position | AddressNo | AddressStreet | AddressCity | FirstName | LastName | Birthday |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| EMP001 | 65000.00 | Manager | 123 | Galle Road | Colombo | Saman | Perera | 1980-05-15 |
| EMP002 | 62000.00 | Manager | 456 | Mahaweli Mawatha | Kandy | Malini | Silva | 1978-09-20 |
| EMP003 | 48000.00 | Senior Driver | 789 | Dutch Hospital Street | Galle | Nimali | Fernando | 1985-03-10 |
| EMP004 | 48000.00 | Driver | 1011 | Nupe Junction | Matara | Kumari | Ratnayake | 1987-07-08 |
| EMP005 | 48000.00 | Driver | 1213 | Hospital Road | Jaffna | Lalith | Bandara | 1989-12-25 |
| EMP006 | 48000.00 | Driver | 1415 | Temple Street | Trincomalee | Kusum | Fernando | 1986-08-30 |
| EMP007 | 48000.00 | Driver | 1617 | Main Street | Negombo | Sujatha | Silva | 1984-04-05 |
| EMP008 | 48000.00 | Driver | 1819 | Police Station Road | Anuradhapura | Sunil | Perera | 1990-06-12 |
| EMP009 | 48000.00 | Driver | 2021 | Bazar Street | Polonnaruwa | Nilmini | Fernando | 1992-02-18 |
| EMP010 | 48000.00 | Driver | 2223 | Railway Avenue | Batticaloa | Ranjan | Silva | 1983-10-07 |
| EMP011 | 55000.00 | Supervisor | 2425 | Church Road | Kalutara | Chandrika | Perera | 1982-06-25 |
| EMP012 | 57000.00 | Supervisor | 2627 | Station Road | Ratnapura | Sarath | Fernando | 1981-11-12 |
| EMP013 | 56000.00 | Supervisor | 2829 | Market Street | Kegalle | Malini | Silva | 1980-08-20 |
| EMP014 | 49000.00 | Driver | 3031 | Court Road | Hambantota | Nalaka | Ranasinghe | 1984-04-15 |
| EMP015 | 49000.00 | Driver | 3233 | Lake Road | Badulla | Saman | Jayawardena | 1986-09-30 |
| EMP016 | 63000.00 | Manager | 363 | Victoria Gardens | Colombo | Nishantha | Fernando | 1975-08-10 |
| EMP017 | 60000.00 | Manager | 3435 | Beach Road | Galle | Chaminda | Perera | 1979-02-28 |
| EMP018 | 62000.00 | Manager | 3637 | River View | Kandy | Anuradha | Silva | 1977-11-05 |
| EMP019 | 64000.00 | Manager | 3839 | Mountain View | Nuwara Eliya | Rukmal | Fernando | 1974-07-22 |
| EMP020 | 57000.00 | Supervisor | 4243 | Forest Avenue | Kandy | Indika | Rathnayake | 1981-01-18 |
| EMP021 | 59000.00 | Supervisor | 4445 | Lake Side | Anuradhapura | Dilhani | Silva | 1978-09-05 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
21 rows in set (0.00 sec)

mysql> select * from EMPLOYEE_CONTACT;
```

2. Employee_contact Table

```
MySQL 8.0 Command Line Cli  +  ×

mysql> UPDATE EMPLOYEE_CONTACT SET ContactNo = '0712345678' WHERE EmployeeID = 'EMP003' AND ContactNo = '0765678901';
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE EMPLOYEE_CONTACT SET ContactNo = '0765432109' WHERE EmployeeID = 'EMP016' AND ContactNo = '0745658121';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> DELETE FROM EMPLOYEE_CONTACT WHERE EmployeeID = 'EMP015' AND ContactNo = '0745677411';
Query OK, 1 row affected (0.02 sec)

mysql> select * from employee_contact;
+-----+-----+
| EmployeeID | ContactNo |
+-----+-----+
| EMP001 | 0711234567 |
| EMP001 | 072345678 |
| EMP002 | 0743456789 |
| EMP002 | 0774567896 |
| EMP003 | 0712345678 |
| EMP003 | 0717896123 |
| EMP003 | 0786789012 |
| EMP004 | 0728901234 |
| EMP004 | 07498012345 |
| EMP005 | 0778123456 |
| EMP006 | 0761234567 |
| EMP007 | 0782345678 |
| EMP008 | 0713456789 |
| EMP009 | 0724567896 |
| EMP010 | 0745678101 |
| EMP011 | 0745598901 |
| EMP011 | 0745628901 |
| EMP011 | 0745678901 |
| EMP012 | 0745679741 |
| EMP013 | 0745675971 |
| EMP014 | 0745673141 |
| EMP015 | 0745678001 |
| EMP015 | 0745675661 |
| EMP016 | 0765432109 |
| EMP017 | 0775678121 |
| EMP018 | 0755678121 |
| EMP019 | 0745672121 |
| EMP021 | 0745679121 |
| EMP022 | 0777899433 |
+-----+-----+
29 rows in set (0.00 sec)
```

3. Manager Table

```
MySQL 8.0 Command Line Cli + X
29 rows in set (0.00 sec)

mysql> UPDATE MANAGER SET Experience = 6 WHERE EmployeeID = 'EMP016';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE MANAGER SET Experience = Experience+1 WHERE EmployeeID = 'EMP017';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> DELETE FROM MANAGER WHERE EmployeeID = 'EMP019';
Query OK, 1 row affected (0.02 sec)

mysql> select * from manager;
+-----+-----+-----+
| EmployeeID | Area      | Experience |
+-----+-----+-----+
| EMP001     | Operations | 5          |
| EMP002     | Finance    | 6          |
| EMP016     | Human Resources | 6          |
| EMP017     | Transport   | 6          |
| EMP018     | Director   | 8          |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
mysql>
mysql>
mysql>
mysql>
mysql> UPDATE DRIVER SET LicenceType = 'Advanced' WHERE EmployeeID = 'EMP003';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> SET LeaderID = NULL WHERE EmployeeID = 'EMP004';
ERROR 1193 (HY000): Unknown system variable 'LeaderID'
mysql> UPDATE DRIVER SET LeaderID = NULL WHERE EmployeeID = 'EMP004';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> DELETE FROM DRIVER WHERE EmployeeID = 'EMP010';
Query OK, 1 row affected (0.01 sec)
```

4. Driver Table

```
MySQL 8.0 Command Line Cli + X
5 rows in set (0.00 sec)

mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> UPDATE DRIVER SET LicenceType = 'Advanced' WHERE EmployeeID = 'EMP003';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> SET LeaderID = NULL WHERE EmployeeID = 'EMP004';
ERROR 1193 (HY000): Unknown system variable 'LeaderID'
mysql> UPDATE DRIVER SET LeaderID = NULL WHERE EmployeeID = 'EMP004';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> DELETE FROM DRIVER WHERE EmployeeID = 'EMP010';
Query OK, 1 row affected (0.01 sec)

mysql> select * from driver;
+-----+-----+-----+-----+
| LicenceNo | EmployeeID | LicenceType | LeaderID |
+-----+-----+-----+-----+
| DL-001    | EMP003    | Advanced    | NULL     |
| DL-002    | EMP004    | Beginner    | NULL     |
| DL-003    | EMP005    | Beginner    | DL-001   |
| DL-004    | EMP006    | Expert      | NULL     |
| DL-005    | EMP007    | Intermediate | DL-004   |
| DL-006    | EMP008    | Intermediate | DL-004   |
| DL-007    | EMP009    | Intermediate | DL-004   |
| DL-009    | EMP014    | Intermediate | DL-001   |
| DL-010    | EMP015    | Beginner    | DL-001   |
+-----+-----+-----+-----+
9 rows in set (0.00 sec)

mysql>
mysql>
mysql>
mysql>
mysql>
mysql> UPDATE DRIVER_EXPERIENCE SET Experience = 6 WHERE EmployeeID = 'EMP014';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

5. Driver_experience Table

```
MySQL 8.0 Command Line Cli + X - □ ×

| DL-009 | EMP014 | Intermediate | DL-001 |
| DL-010 | EMP015 | Beginner     | DL-001 |
+-----+-----+-----+-----+
9 rows in set (0.00 sec)

mysql>
mysql>
mysql>
mysql>
mysql>
mysql> UPDATE DRIVER_EXPERIENCE SET Experience = 6 WHERE EmployeeID = 'EMP014';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE DRIVER_EXPERIENCE SET Experience = 7 WHERE EmployeeID = 'EMP015';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> DELETE FROM DRIVER_EXPERIENCE WHERE EmployeeID = 'EMP009';
Query OK, 1 row affected (0.02 sec)

mysql> select * from DRIVER_EXPERIENCE;
+-----+-----+
| EmployeeID | Experience |
+-----+-----+
| EMP003     |      10 |
| EMP004     |       1 |
| EMP005     |       1 |
| EMP006     |       6 |
| EMP007     |       2 |
| EMP008     |       3 |
| EMP010     |       4 |
| EMP014     |       6 |
| EMP015     |       7 |
+-----+-----+
9 rows in set (0.00 sec)

mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> UPDATE SUPERVISOR SET EmployeeID = 'EMP021' WHERE SupervisorID = 'SV-001';
ERROR 1062 (23000): Duplicate entry 'EMP021' for key 'supervisor.EmployeeID'
mysql>
```

6. Supervisor Table

```
MySQL 8.0 Command Line Cli + X - □ ×

mysql>
mysql>
mysql>
mysql>
mysql>
mysql> UPDATE SUPERVISOR SET SupervisorID = 'SV-008' WHERE SupervisorID = 'SV-001';
Query OK, 1 row affected (0.03 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE SUPERVISOR SET SupervisorID = 'SV-009' WHERE SupervisorID = 'SV-002';
Query OK, 1 row affected (0.03 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> DELETE FROM SUPERVISOR WHERE SupervisorID = 'SV-006';
Query OK, 1 row affected (0.04 sec)

mysql> select * from supervisor;
+-----+-----+
| SupervisorID | EmployeeID |
+-----+-----+
| SV-008       | EMP011    |
| SV-009       | EMP012    |
| SV-003       | EMP013    |
| SV-005       | EMP021    |
+-----+-----+
4 rows in set (0.00 sec)

mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> UPDATE ROUTE SET SupervisorID = 'SV-003' WHERE RouteNo = 1;
Query OK, 1 row affected (0.03 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE ROUTE SET SupervisorID = 'SV-009' WHERE RouteNo = 2;
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1 Changed: 0 Warnings: 0

mysql> DELETE FROM ROUTE WHERE RouteNo = 6;
Query OK, 1 row affected (0.03 sec)

mysql> select * from route;
```

7. Route Table

```
MySQL 8.0 Command Line Cli x + - _   
mysql>  
mysql>  
mysql>  
mysql>  
mysql> UPDATE ROUTE SET SupervisorID = 'SV-003' WHERE RouteNo = 1;  
Query OK, 1 row affected (0.03 sec)  
Rows matched: 1 Changed: 1 Warnings: 0  
  
mysql> UPDATE ROUTE SET SupervisorID = 'SV-009' WHERE RouteNo = 2;  
Query OK, 0 rows affected (0.00 sec)  
Rows matched: 1 Changed: 0 Warnings: 0  
  
mysql> DELETE FROM ROUTE WHERE RouteNo = 6;  
Query OK, 1 row affected (0.03 sec)  
  
mysql> select * from route;  
+-----+-----+-----+-----+-----+-----+  
| RouteNo | SupervisorID | Distance | Duration | Start | End |  
+-----+-----+-----+-----+-----+-----+  
| 1 | SV-003 | 100 | 120 | Colombo | Kandy |  
| 2 | SV-009 | 150 | 150 | Colombo | Matara |  
| 3 | SV-003 | 80 | 90 | Gampaha | Polonnaruwa |  
| 4 | SV-008 | 200 | 180 | Kadawatha | Jaffna |  
| 5 | SV-009 | 120 | 130 | Colombo | Ratnapura |  
+-----+-----+-----+-----+-----+  
5 rows in set (0.00 sec)  
  
mysql>  
mysql>  
mysql>  
mysql>  
mysql>  
mysql>  
mysql>  
mysql> UPDATE BUS_SCHEDULE SET EndTime = '13:00:00' WHERE ScheduleID = 'SCH-003';  
Query OK, 1 row affected (0.02 sec)  
Rows matched: 1 Changed: 1 Warnings: 0  
  
mysql> UPDATE BUS_SCHEDULE SET StartTime = '08:30:00' WHERE ScheduleID = 'SCH-002';  
Query OK, 1 row affected (0.02 sec)  
Rows matched: 1 Changed: 1 Warnings: 0  
  
mysql> DELETE FROM BUS_SCHEDULE WHERE ScheduleID = 'SCH-010';  
Query OK, 1 row affected (0.02 sec)  
  
mysql> select * from BUS_SCHEDULE;
```

8. Bus_schedule Table

```
MySQL 8.0 Command Line Cli x + - _ X

mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> UPDATE BUS_SCHEDULE SET EndTime = '13:00:00' WHERE ScheduleID = 'SCH-003';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE BUS_SCHEDULE SET StartTime = '08:30:00' WHERE ScheduleID = 'SCH-002';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> DELETE FROM BUS_SCHEDULE WHERE ScheduleID = 'SCH-010';
Query OK, 1 row affected (0.02 sec)

mysql> select * from BUS_SCHEDULE;
+-----+-----+-----+-----+
| ScheduleID | RouteNo | StartTime | EndTime |
+-----+-----+-----+-----+
| SCH-001    |      1 | 08:00:00 | 10:00:00 |
| SCH-002    |      2 | 08:30:00 | 12:00:00 |
| SCH-003    |      3 | 10:00:00 | 13:00:00 |
| SCH-004    |      4 | 11:00:00 | 14:30:00 |
| SCH-005    |      5 | 12:00:00 | 14:00:00 |
| SCH-006    |      1 | 13:00:00 | 15:00:00 |
| SCH-007    |      2 | 14:00:00 | 17:00:00 |
| SCH-008    |      3 | 15:00:00 | 17:30:00 |
| SCH-009    |      4 | 16:00:00 | 19:30:00 |
+-----+-----+-----+-----+
9 rows in set (0.00 sec)

mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> UPDATE DEPOT SET Capacity = 15 WHERE DepotID = 'DEP001';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

9. Depot Table

```
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> UPDATE DEPOT SET Capacity = 15 WHERE DepotID = 'DEP001';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE DEPOT SET Capacity = 8 WHERE DepotID = 'DEP002';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> DELETE FROM DEPOT WHERE DepotID = 'DEP006';
Query OK, 1 row affected (0.02 sec)

mysql> select * from depot;
+-----+-----+-----+
| DepotID | Capacity | ContactNo |
+-----+-----+-----+
| DEP001 |      15 | 0711234567 |
| DEP002 |       8 | 0712345678 |
| DEP003 |       5 | 0762345678 |
| DEP004 |       3 | 0722345678 |
| DEP005 |       4 | 0742345678 |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
mysql>
mysql>
mysql>
mysql>
mysql> UPDATE BUS SET Efficiency = 92 WHERE BusID = 'BUS-001';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE BUS SET Efficiency = 90 WHERE BusID = 'BUS-002';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> DELETE FROM BUS WHERE BusID = 'BUS-010';
Query OK, 1 row affected (0.02 sec)
```

10. Bus Table

```
5 rows in set (0.00 sec)

mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> UPDATE BUS SET Efficiency = 92 WHERE BusID = 'BUS-001';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE BUS SET Efficiency = 90 WHERE BusID = 'BUS-002';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> DELETE FROM BUS WHERE BusID = 'BUS-010';
Query OK, 1 row affected (0.02 sec)

mysql> select * from bus;
+-----+-----+-----+-----+-----+-----+-----+-----+
| BusID | ScheduleID | DepotID | Brand | Efficiency | Capacity | Year | Category |
+-----+-----+-----+-----+-----+-----+-----+-----+
| BUS-001 | SCH-001 | DEP001 | TATA |      92 |      50 | 2018 | Express |
| BUS-002 | SCH-002 | DEP003 | TATA |      90 |      45 | 2017 | Standard |
| BUS-003 | SCH-003 | DEP002 | Leyland |      92 |      55 | 2019 | Luxury |
| BUS-004 | SCH-004 | DEP001 | Leyland |      88 |      40 | 2016 | Express |
| BUS-005 | SCH-005 | DEP002 | HINO |      85 |      50 | 2015 | Standard |
| BUS-006 | SCH-006 | DEP003 | HINO |      98 |      60 | 2020 | Luxury |
| BUS-007 | SCH-007 | DEP002 | TATA |      92 |      45 | 2017 | Express |
| BUS-008 | SCH-008 | DEP001 | TATA |      89 |      55 | 2018 | Standard |
| BUS-009 | SCH-009 | DEP002 | Leyland |      86 |      40 | 2019 | Luxury |
+-----+-----+-----+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)

mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> UPDATE FARE SET PriceRate = 280.00 WHERE FareID = 'F1T1';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

11. Fare Table

```
MySQL 8.0 Command Line Cli + ^

mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> UPDATE FARE SET PriceRate = 280.00 WHERE FareID = 'F1T1';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE FARE SET PriceRate = 320.00 WHERE FareID = 'F2T1';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> DELETE FROM FARE WHERE FareID = 'F5T2';
Query OK, 1 row affected (0.03 sec)

mysql> select * from fare;
+-----+-----+-----+-----+
| FareID | RouteNo | PriceRate | TicketType |
+-----+-----+-----+-----+
| F1T1   | 1      | 280.00  | full      |
| F1T2   | 1      | 150.00  | half      |
| F2T1   | 2      | 320.00  | full      |
| F2T2   | 2      | 180.00  | half      |
| F3T1   | 3      | 280.00  | full      |
| F3T2   | 3      | 160.00  | half      |
| F4T1   | 4      | 229.00  | full      |
| F4T2   | 4      | 139.00  | half      |
| F5T1   | 5      | 270.00  | full      |
+-----+-----+-----+-----+
9 rows in set (0.00 sec)

mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> UPDATE MAINTENANCE SET LastOilBrand = 'Shell' WHERE ServiceID = 'SERV001';
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1 Changed: 0 Warnings: 0
```

12. Maintenance Table

```
MySQL 8.0 Command Line Cli x + - _ X

mysql>
mysql>
mysql>
mysql>
mysql>
mysql> UPDATE MAINTENANCE SET LastOilBrand = 'Shell' WHERE ServiceID = 'SERV001';
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1 Changed: 0 Warnings: 0

mysql> UPDATE MAINTENANCE SET LastOilBrand = 'Castrol' WHERE ServiceID = 'SERV002';
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1 Changed: 0 Warnings: 0

mysql> DELETE FROM MAINTENANCE WHERE ServiceID = 'SERV007';
Query OK, 1 row affected (0.02 sec)

mysql> select * from maintenance;
+-----+-----+-----+-----+-----+-----+-----+-----+
| ServiceID | BusID | LastTireChange | LastTireBrand | LastBrakeChange | LastOilChange | LastOilBrand | LastService |
+-----+-----+-----+-----+-----+-----+-----+-----+
| SERV001   | BUS-001 | 2023-05-20 | Michelin      | 2023-04-15    | 2023-03-10    | Shell         | 2023-05-25  |
| SERV002   | BUS-002 | 2023-04-18 | Bridgestone   | 2023-03-12    | 2023-02-20    | Castrol       | 2023-05-28  |
| SERV003   | BUS-003 | 2023-06-05 | Goodyear      | 2023-05-02    | 2023-04-10    | Mobil         | 2023-05-30  |
| SERV004   | BUS-004 | 2023-05-25 | Continental   | 2023-04-20    | 2023-03-15    | Total         | 2023-06-01  |
| SERV005   | BUS-005 | 2023-04-30 | Pirelli       | 2023-03-25    | 2023-02-28    | BP            | 2023-06-02  |
| SERV006   | BUS-006 | 2023-06-10 | Firestone     | 2023-05-05    | 2023-04-20    | ExxonMobil   | 2023-06-03  |
| SERV008   | BUS-008 | 2023-06-20 | Hankook      | 2023-05-15    | 2023-04-10    | Shell         | 2023-06-07  |
| SERV009   | BUS-009 | 2023-05-28 | Dunlop        | 2023-04-25    | 2023-03-20    | Castrol       | 2023-06-10  |
| SERV010   | NULL    | 2023-06-02 | Michelin      | 2023-05-30    | 2023-04-25    | Mobil         | 2023-06-12  |
+-----+-----+-----+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)

mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> UPDATE SERVICE_COST SET Payment_method = 'Bank Transfer' WHERE ServiceID = 'SERV010';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE SERVICE_COST SET Payment_method = 'Cash' WHERE ServiceID = 'SERV009';
Query OK, 0 rows affected (0.00 sec)
```

13. Service_cost Table

```
MySQL 8.0 Command Line Cli + X

mysql>
mysql>
mysql>
mysql>
mysql> UPDATE SERVICE_COST SET Payment_method = 'Bank Transfer' WHERE ServiceID = 'SERV010';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE SERVICE_COST SET Payment_method = 'Cash' WHERE ServiceID = 'SERV009';
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1 Changed: 0 Warnings: 0

mysql> DELETE FROM SERVICE_COST WHERE ServiceID = 'SERV005';
Query OK, 1 row affected (0.02 sec)

mysql> select * from service_cost;
+-----+-----+-----+
| ServiceID | Amount | Payment_method |
+-----+-----+-----+
| SERV001 | 1500.00 | Credit Card |
| SERV002 | 1800.00 | Cash |
| SERV003 | 2000.00 | Bank Transfer |
| SERV004 | 1700.00 | Cash |
| SERV006 | 1600.00 | Cash |
| SERV008 | 1950.00 | Bank Transfer |
| SERV009 | 1850.00 | Cash |
| SERV010 | 2100.00 | Bank Transfer |
+-----+-----+-----+
8 rows in set (0.00 sec)

mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> UPDATE FUEL SET FillDate = '2023-05-12' WHERE BillNo = 'BILL001';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE FUEL SET FillDate = '2023-05-11' WHERE BillNo = 'BILL002';
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1 Changed: 0 Warnings: 0
```

14. Fuel Table

```
MySQL 8.0 Command Line Cli + X

mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> UPDATE FUEL SET FillDate = '2023-05-12' WHERE BillNo = 'BILL001';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE FUEL SET FillDate = '2023-05-11' WHERE BillNo = 'BILL002';
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1 Changed: 0 Warnings: 0

mysql> DELETE FROM FUEL WHERE BillNo = 'BILL005';
Query OK, 1 row affected (0.02 sec)

mysql> select * from fuel;
+-----+-----+-----+
| BillNo | BusID | FillDate | Volume |
+-----+-----+-----+
| BILL001 | BUS-001 | 2023-05-12 | 100.5 |
| BILL002 | BUS-002 | 2023-05-11 | 98.2 |
| BILL003 | BUS-003 | 2023-05-12 | 105.8 |
| BILL004 | BUS-004 | 2023-05-13 | 102.3 |
| BILL006 | BUS-006 | 2023-05-15 | 101.1 |
| BILL007 | BUS-007 | 2023-05-16 | 99.9 |
| BILL008 | BUS-008 | 2023-05-17 | 103.5 |
| BILL009 | BUS-009 | 2023-05-18 | 96.8 |
| BILL010 | NULL | 2023-05-19 | 104.7 |
+-----+-----+-----+
9 rows in set (0.00 sec)

mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> UPDATE FUEL_COST SET Amount = 24500.00 WHERE BillNo = 'BILL002';
Query OK, 1 row affected (0.04 sec)
```

15. Fuel_cost Table

```
MySQL 8.0 Command Line Cli  + | x
9 rows in set (0.00 sec)

mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> UPDATE FUEL_COST SET Amount = 24500.00 WHERE BillNo = 'BILL002';
Query OK, 1 row affected (0.04 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE FUEL_COST SET Amount = 26000.00 WHERE BillNo = 'BILL003';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> DELETE FROM FUEL_COST WHERE BillNo = 'BILL006';
Query OK, 1 row affected (0.02 sec)

mysql> select * from fuel_cost;
+-----+-----+-----+
| BillNo | Amount | Payment_method |
+-----+-----+-----+
| BILL001 | 25000.00 | Cash          |
| BILL002 | 24500.00 | Credit Card   |
| BILL003 | 26000.00 | Cash          |
| BILL004 | 25500.00 | Cash          |
| BILL007 | 24900.00 | Credit Card   |
| BILL008 | 25800.00 | Cash          |
| BILL009 | 24200.00 | Credit Card   |
| BILL010 | 26100.00 | Cash          |
+-----+-----+-----+
8 rows in set (0.00 sec)

mysql>
mysql>
mysql>
mysql>
```

Chapter 4 - Transactions

➤ Simple Queries

- Select Operation

The screenshot shows the MySQL Workbench interface with a query editor window. The code entered is:

```
7  
8 -- Selecting all columns from the EMPLOYEE table  
9 • SELECT * FROM EMPLOYEE;  
10  
11  
12
```

The result grid displays the following data:

EmployeeID	Salary	Position	AddressNo	AddressStreet	AddressCity	FirstName	LastName	Birthday
EMP001	65000.00	Manager	123	Galle Road	Colombo	Saman	Perera	1980-05-15
EMP002	62000.00	Manager	456	Mahaveli Marawatha	Kandy	Malini	Silva	1978-09-20
EMP003	48000.00	Senior Driver	789	Dutch Hospital Street	Galle	Nimal	Fernando	1985-03-10
EMP004	48000.00	Driver	1011	Nuge Junction	Matara	Kumari	Rathnayake	1987-07-08
EMP005	48000.00	Driver	1213	Hospital Road	Jaffna	Lalith	Bandara	1999-12-25
EMP006	48000.00	Driver	1415	Temple Street	Trincomalee	Kusum	Fernando	1986-08-30
EMP007	48000.00	Driver	1617	Main Street	Negombo	Sujatha	Silva	1984-04-05
EMP008	48000.00	Driver	1819	Police Station Road	Anuradhapura	Sunil	Perera	1990-06-12
EMP009	48000.00	Driver	2021	Bazar Street	Polonnaruwa	Nimini	Fernando	1992-02-18
EMP010	48000.00	Driver	2223	Railway Avenue	Batticaloa	Ranjan	Silva	1983-10-07
EMP011	55000.00	Supervisor	2425	Church Road	Kalutara	Chandrika	Perera	1982-06-25
EMP012	57000.00	Supervisor	2627	Station Road	Ratnapura	Sarath	Fernando	1981-11-12
EMP013	56000.00	Supervisor	2829	Market Street	Kegalle	Malini	Silva	1980-08-20
EMP014	49000.00	Driver	3031	Court Road	Hambantota	Nalaka	Ranasinghe	1984-04-15
EMP015	49000.00	Driver	3233	Lake Road	Badulla	Sanan	Jayaward...	1986-09-30
EMP016	63000.00	Manager	363	Victoria Gardens	Colombo	Nishantha	Fernando	1975-08-10
EMP017	60000.00	Manager	3435	Beach Road	Galle	Chaminda	Perera	1979-02-28
EMP018	62000.00	Manager	3637	River View	Kandy	Anuradha	Silva	1977-11-05
EMP019	64000.00	Manager	3839	Mountain View	Nuwara Eliya	Rukmal	Fernando	1974-07-22
EMP021	57000.00	Supervisor	4243	Forest Avenue	Kandy	Indika	Rathnayake	1981-01-18
EMP022	59000.00	Supervisor	4445	Lake Side	Anuradhapura	Dilhani	Silva	1978-09-05

- Project Operation

The screenshot shows the MySQL Workbench interface with a query editor window. The code entered is:

```
16  
17  
18  
19 -- Selecting specific columns (EmployeeID, FirstName, LastName) from the EMPLOYEE table  
20 • SELECT EmployeeID, FirstName, LastName FROM EMPLOYEE;  
21  
22
```

The result grid displays the following data:

EmployeeID	FirstName	LastName
EMP001	Saman	Perera
EMP002	Malini	Silva
EMP003	Nimal	Fernando
EMP004	Kumari	Rathnayake
EMP005	Lalith	Bandara
EMP006	Kusum	Fernando
EMP007	Sujatha	Silva
EMP008	Suni	Perera
EMP009	Nimini	Fernando
EMP010	Ranjan	Silva
EMP011	Chandrika	Perera
EMP012	Sarath	Fernando
EMP013	Malini	Silva
EMP014	Nalaka	Ranasinghe
EMP015	Sanan	Jayaward...
EMP016	Nishantha	Fernando
EMP017	Chaminda	Perera
EMP018	Anuradha	Silva
EMP019	Rukmal	Fernando
EMP021	Indika	Rathnayake
EMP022	Dilhani	Silva

- Cartesian product

The screenshot shows the MySQL Workbench interface with a query editor window. The code entered is:

```

22
23
24 -- Performing a Cartesian product of the EMPLOYEE and DRIVER tables
25 • SELECT * FROM EMPLOYEE, DRIVER;
26
27
28

```

The results grid displays the Cartesian product of the EMPLOYEE and DRIVER tables. The columns are EmployeeID, Salary, Position, AddressNo, AddressStreet, AddressCity, FirstName, LastName, Birthday, LicenceNo, EmployeeID, LicenceType, and LeaderID. The data shows multiple rows for each employee, with the EmployeeID from the first table and the DriverID from the second table appearing in the EmployeeID column.

EmployeeID	Salary	Position	AddressNo	AddressStreet	AddressCity	FirstName	LastName	Birthday	LicenceNo	EmployeeID	LicenceType	LeaderID
EMP001	65000.00	Manager	123	Galle Road	Colombo	Saman	Perera	1980-05-15	DL-010	EMP015	Beginer	DL-001
EMP001	65000.00	Manager	123	Galle Road	Colombo	Saman	Perera	1980-05-15	DL-009	EMP014	Intermediate	DL-001
EMP001	65000.00	Manager	123	Galle Road	Colombo	Saman	Perera	1980-05-15	DL-007	EMP009	Intermediate	DL-004
EMP001	65000.00	Manager	123	Galle Road	Colombo	Saman	Perera	1980-05-15	DL-006	EMP008	Intermediate	DL-004
EMP001	65000.00	Manager	123	Galle Road	Colombo	Saman	Perera	1980-05-15	DL-005	EMP007	Intermediate	DL-004
EMP001	65000.00	Manager	123	Galle Road	Colombo	Saman	Perera	1980-05-15	DL-004	EMP006	Expert	HULL
EMP001	65000.00	Manager	123	Galle Road	Colombo	Saman	Perera	1980-05-15	DL-003	EMP005	Beginer	DL-001
EMP001	65000.00	Manager	123	Galle Road	Colombo	Saman	Perera	1980-05-15	DL-002	EMP004	Beginer	HULL
EMP001	65000.00	Manager	123	Galle Road	Colombo	Saman	Perera	1980-05-15	DL-001	EMP003	Advanced	HULL
EMP002	62000.00	Manager	456	Mahaweli Mawatha	Kandy	Malini	Silva	1978-09-20	DL-010	EMP015	Beginer	DL-001
EMP002	62000.00	Manager	456	Mahaweli Mawatha	Kandy	Malini	Silva	1978-09-20	DL-009	EMP014	Intermediate	DL-001
EMP002	62000.00	Manager	456	Mahaweli Mawatha	Kandy	Malini	Silva	1978-09-20	DL-007	EMP009	Intermediate	DL-004
EMP002	62000.00	Manager	456	Mahaweli Mawatha	Kandy	Malini	Silva	1978-09-20	DL-006	EMP008	Intermediate	DL-004
EMP002	62000.00	Manager	456	Mahaweli Mawatha	Kandy	Malini	Silva	1978-09-20	DL-005	EMP007	Intermediate	DL-004
EMP002	62000.00	Manager	456	Mahaweli Mawatha	Kandy	Malini	Silva	1978-09-20	DL-004	EMP006	Expert	HULL
EMP002	62000.00	Manager	456	Mahaweli Mawatha	Kandy	Malini	Silva	1978-09-20	DL-003	EMP005	Beginer	DL-001
EMP002	62000.00	Manager	456	Mahaweli Mawatha	Kandy	Malini	Silva	1978-09-20	DL-002	EMP004	Beginer	HULL
EMP002	62000.00	Manager	456	Mahaweli Mawatha	Kandy	Malini	Silva	1978-09-20	DL-001	EMP003	Advanced	HULL
EMP003	48000.00	Senior ...	789	Dutch Hospital Str...	Galle	Nimal	Fernando	1985-03-10	DL-010	EMP015	Beginer	DL-001
EMP003	48000.00	Senior ...	789	Dutch Hospital Str...	Galle	Nimal	Fernando	1985-03-10	DL-009	EMP014	Intermediate	DL-001
EMP003	48000.00	Senior ...	789	Dutch Hospital Str...	Galle	Nimal	Fernando	1985-03-10	DL-007	EMP009	Intermediate	DL-004
EMP003	48000.00	Senior ...	789	Dutch Hospital Str...	Galle	Nimal	Fernando	1985-03-10	DL-006	EMP008	Intermediate	DL-004
EMP003	48000.00	Senior ...	789	Dutch Hospital Str...	Galle	Nimal	Fernando	1985-03-10	DL-005	EMP007	Intermediate	DL-004

- Creating a user view

The screenshot shows the MySQL Workbench interface with a query editor window. The code entered is:

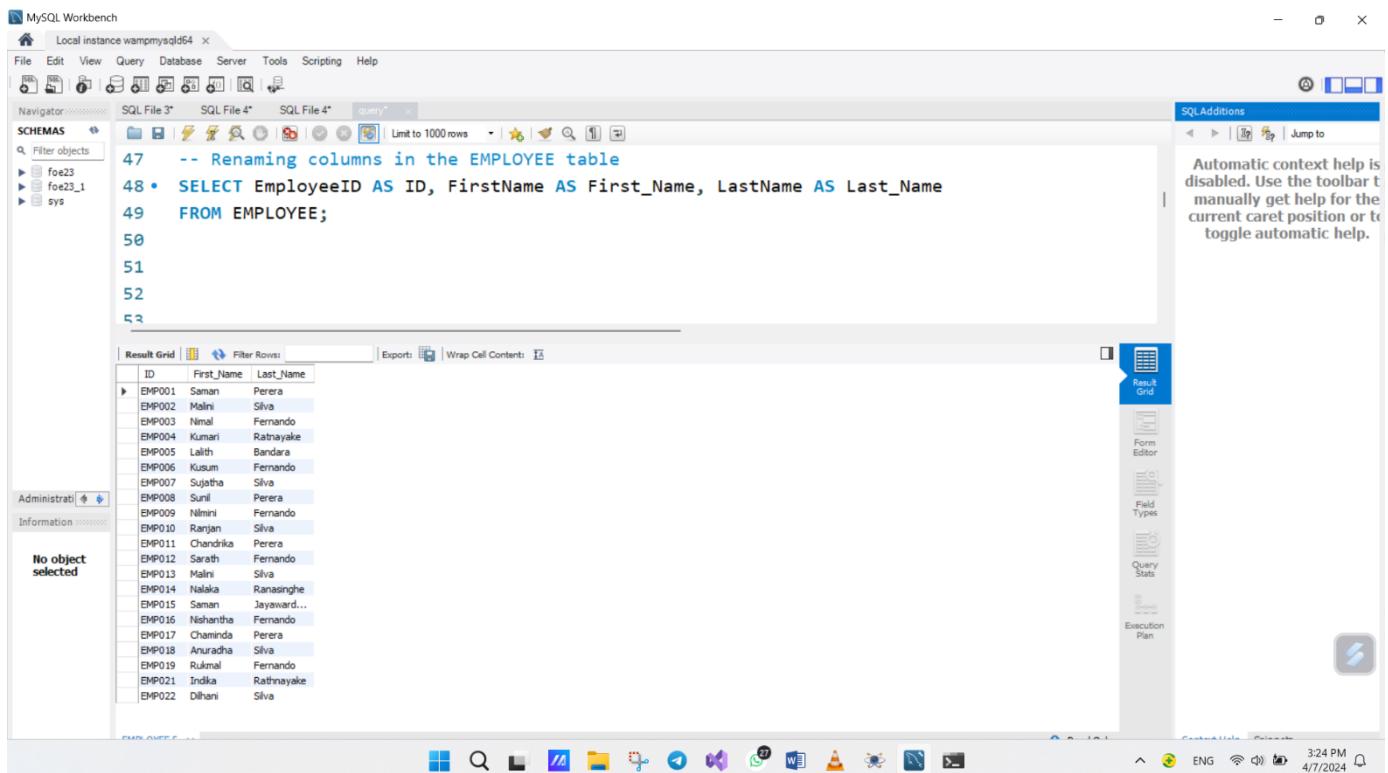
```

27
28
29
30
31
32 -- Creating a view named Manager_View to select details of managers
33 • CREATE VIEW Manager_View AS
34 SELECT EmployeeID, Position, Salary
35 FROM EMPLOYEE
36 WHERE Position = 'Manager';
37
38
39
40
41
42
43
44
45
46
47 -- Renaming columns in the EMPLOYEE table
48 • SELECT EmployeeID AS ID, FirstName AS First_Name, LastName AS Last_Name

```

3:24 PM 4/7/2024

- Renaming Operation



The screenshot shows the MySQL Workbench interface with a query editor window. The query is:

```

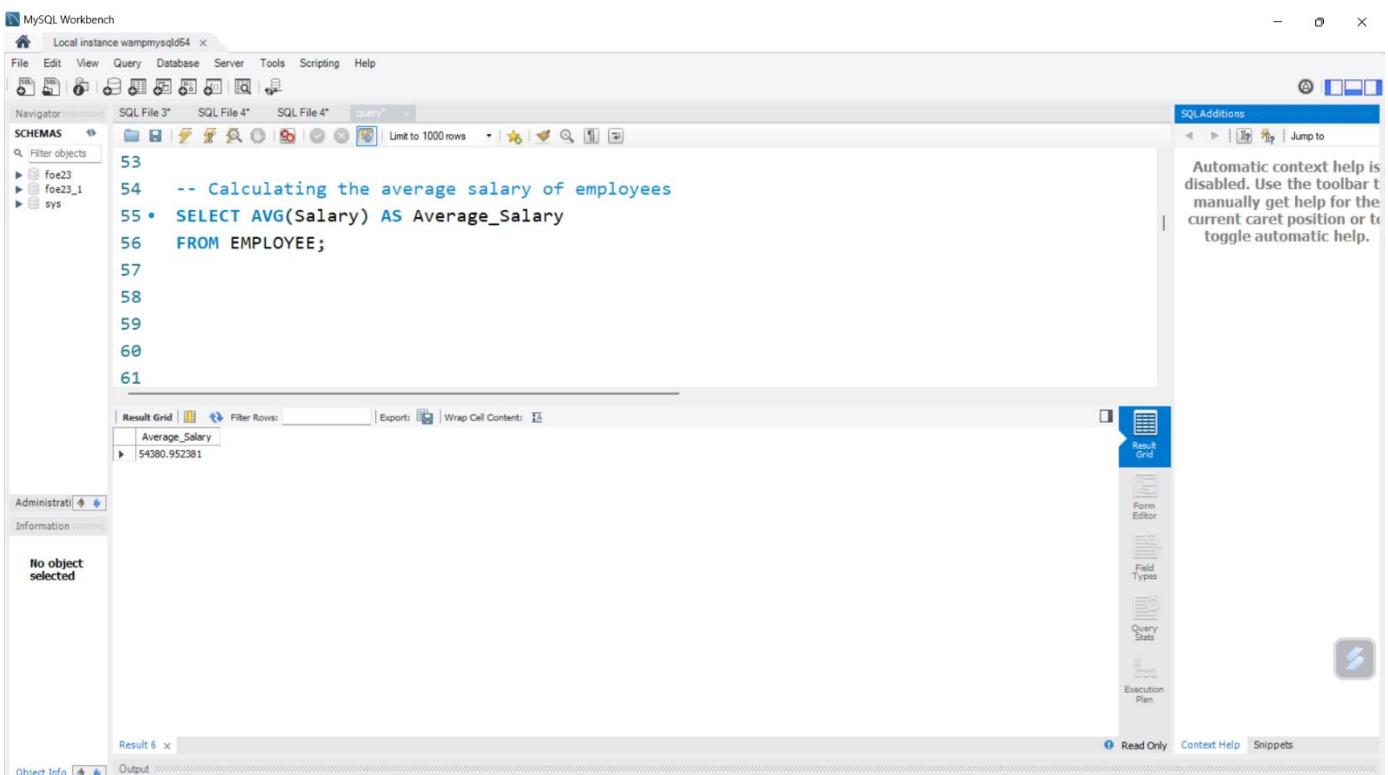
47 -- Renaming columns in the EMPLOYEE table
48 • SELECT EmployeeID AS ID, FirstName AS First_Name, LastName AS Last_Name
49 FROM EMPLOYEE;
50
51
52
53

```

The result grid shows the following data:

ID	First_Name	Last_Name
EMP001	Saman	Perera
EMP002	Mallin	Silva
EMP003	Nimal	Fernando
EMP004	Kumari	Rathnayake
EMP005	Lalith	Bandara
EMP006	Kusum	Fernando
EMP007	Sujatha	Silva
EMP008	Sunil	Perera
EMP009	Nimini	Fernando
EMP010	Ranjan	Silva
EMP011	Chandrika	Perera
EMP012	Sarith	Fernando
EMP013	Mallin	Silva
EMP014	Nalaka	Ranasinghe
EMP015	Saman	Jayaward...
EMP016	Nishantha	Fernando
EMP017	Chaminda	Perera
EMP018	Anuradha	Silva
EMP019	Rukmal	Fernando
EMP020	Indika	Rathnayake
EMP022	Dihan	Silva

- Aggregation (Average function)



The screenshot shows the MySQL Workbench interface with a query editor window. The query is:

```

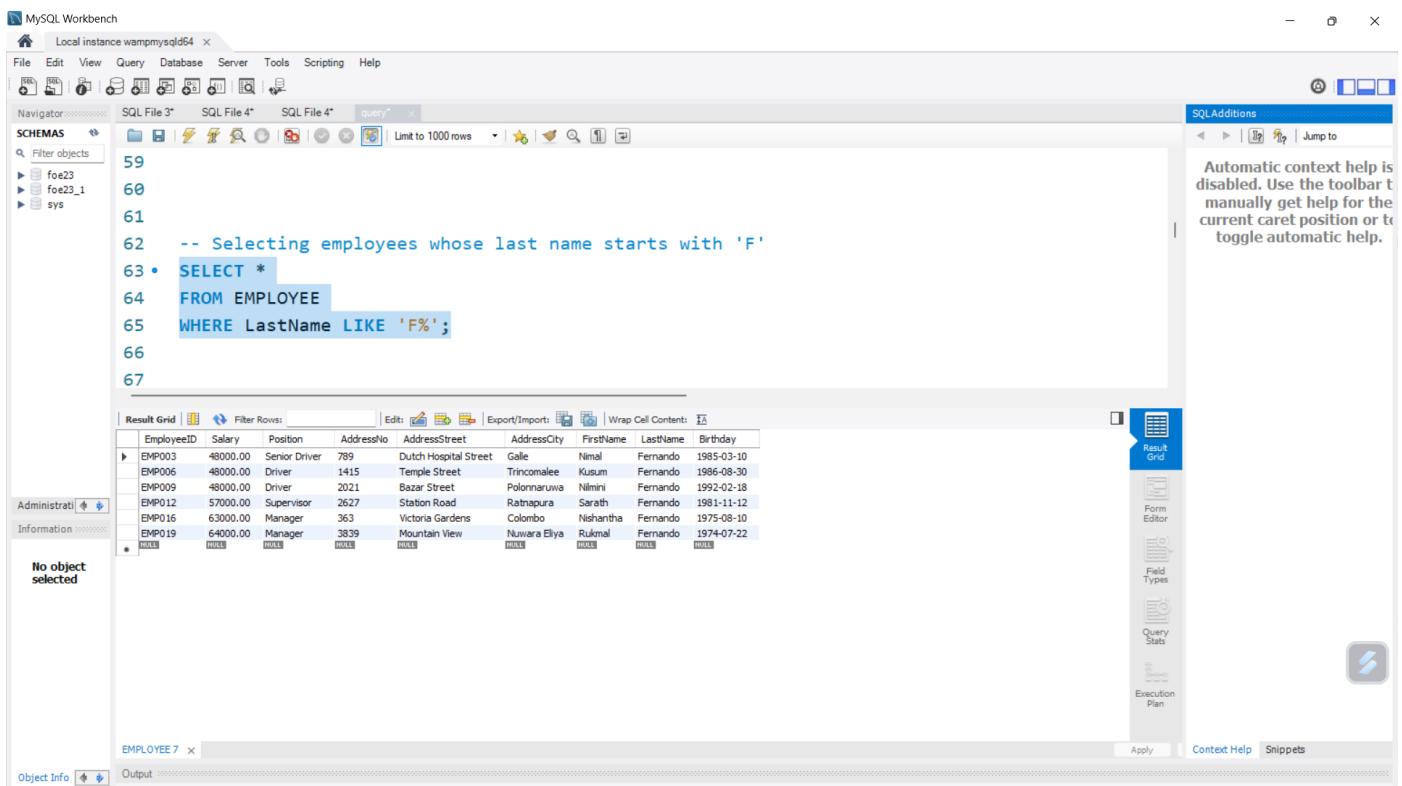
53
54 -- Calculating the average salary of employees
55 • SELECT AVG(Salary) AS Average_Salary
56 FROM EMPLOYEE;
57
58
59
60
61

```

The result grid shows the following data:

Average_Salary
54380.952381

- Like key word



The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```

59
60
61
62 -- Selecting employees whose last name starts with 'F'
63 • SELECT *
64   FROM EMPLOYEE
65 WHERE LastName LIKE 'F%';
66
67

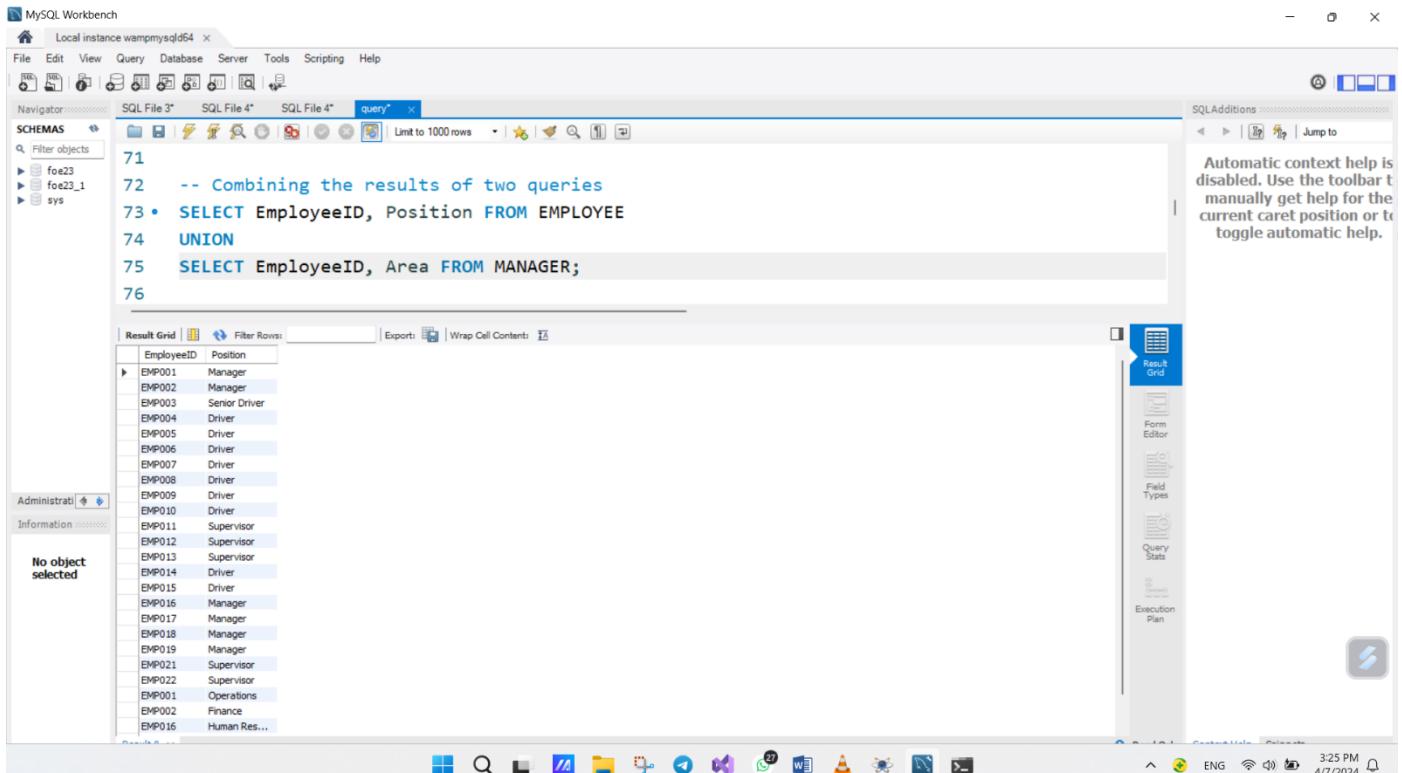
```

The result grid displays the following data:

EmployeeID	Salary	Position	AddressNo	AddressStreet	AddressCity	FirstName	LastName	Birthday
EMP003	48000.00	Senior Driver	789	Dutch Hospital Street	Galle	Nimal	Fernando	1985-03-10
EMP005	48000.00	Driver	1415	Temple Street	Trincomalee	Kusum	Fernando	1986-08-30
EMP009	48000.00	Driver	2021	Bazar Street	Polonnaruwa	Nimini	Fernando	1992-02-18
EMP012	57000.00	Supervisor	2627	Station Road	Ratnapura	Sarith	Fernando	1981-11-12
EMP016	63000.00	Manager	363	Victoria Gardens	Colombo	Nishantha	Fernando	1975-08-10
EMP019	64000.00	Manager	3839	Mountain View	Nuwara Eliya	Rukmal	Fernando	1974-07-22

➤ Complex Queries

- Union



The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```

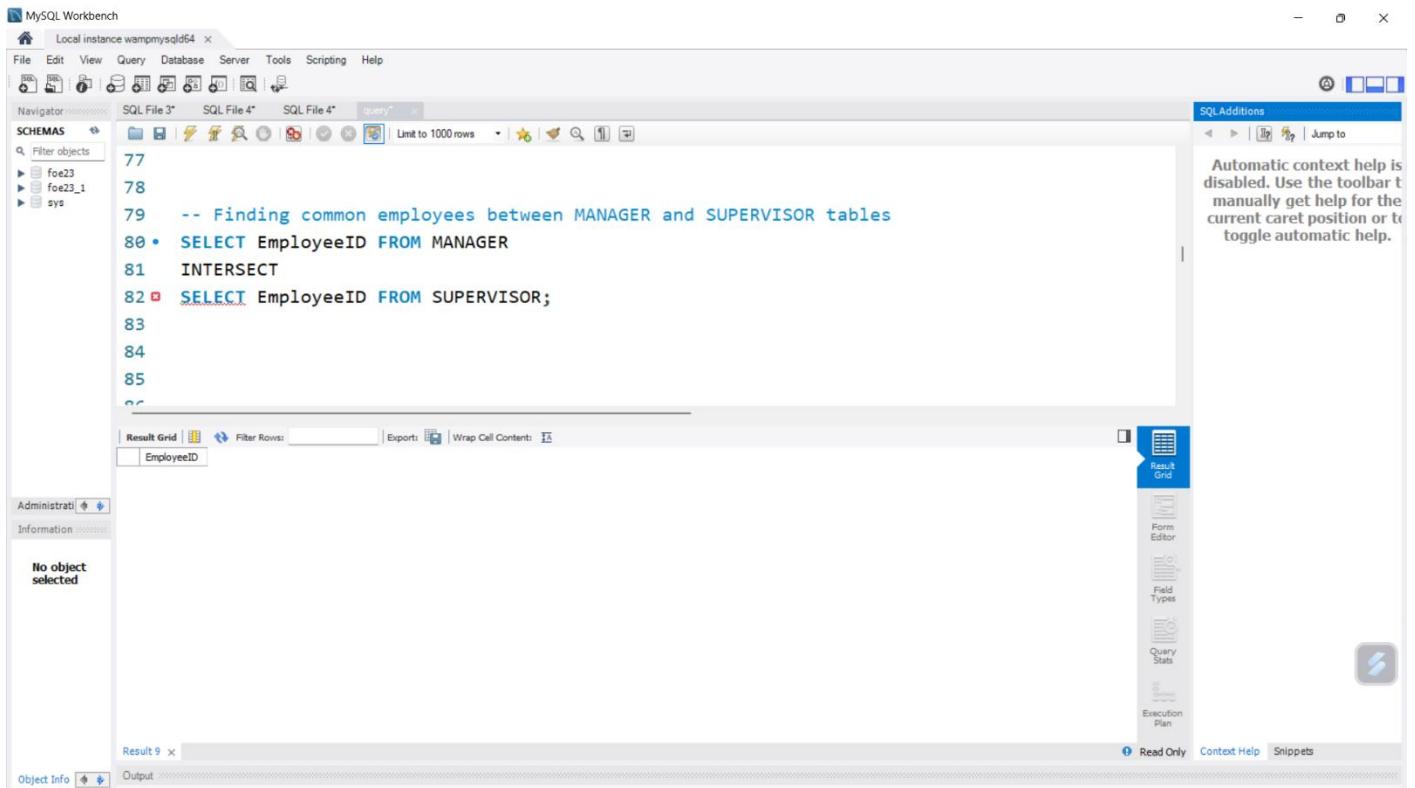
71
72 -- Combining the results of two queries
73 • SELECT EmployeeID, Position FROM EMPLOYEE
74 UNION
75   SELECT EmployeeID, Area FROM MANAGER;
76

```

The result grid displays the combined data from both tables:

EmployeeID	Position
EMP001	Manager
EMP002	Manager
EMP003	Senior Driver
EMP004	Driver
EMP005	Driver
EMP006	Driver
EMP007	Driver
EMP008	Driver
EMP009	Driver
EMP010	Driver
EMP011	Supervisor
EMP012	Supervisor
EMP013	Supervisor
EMP014	Driver
EMP015	Driver
EMP016	Manager
EMP017	Manager
EMP018	Manager
EMP019	Manager
EMP021	Supervisor
EMP022	Supervisor
EMP001	Operations
EMP002	Finance
EMP016	Human Res...

- Intersection



MySQL Workbench - Local instance wampmysqld54

```

77
78
79 -- Finding common employees between MANAGER and SUPERVISOR tables
80 • SELECT EmployeeID FROM MANAGER
81 INTERSECT
82 ◻ SELECT EmployeeID FROM SUPERVISOR;
83
84
85
86

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | EmployeeID

No object selected

Result 9 ×

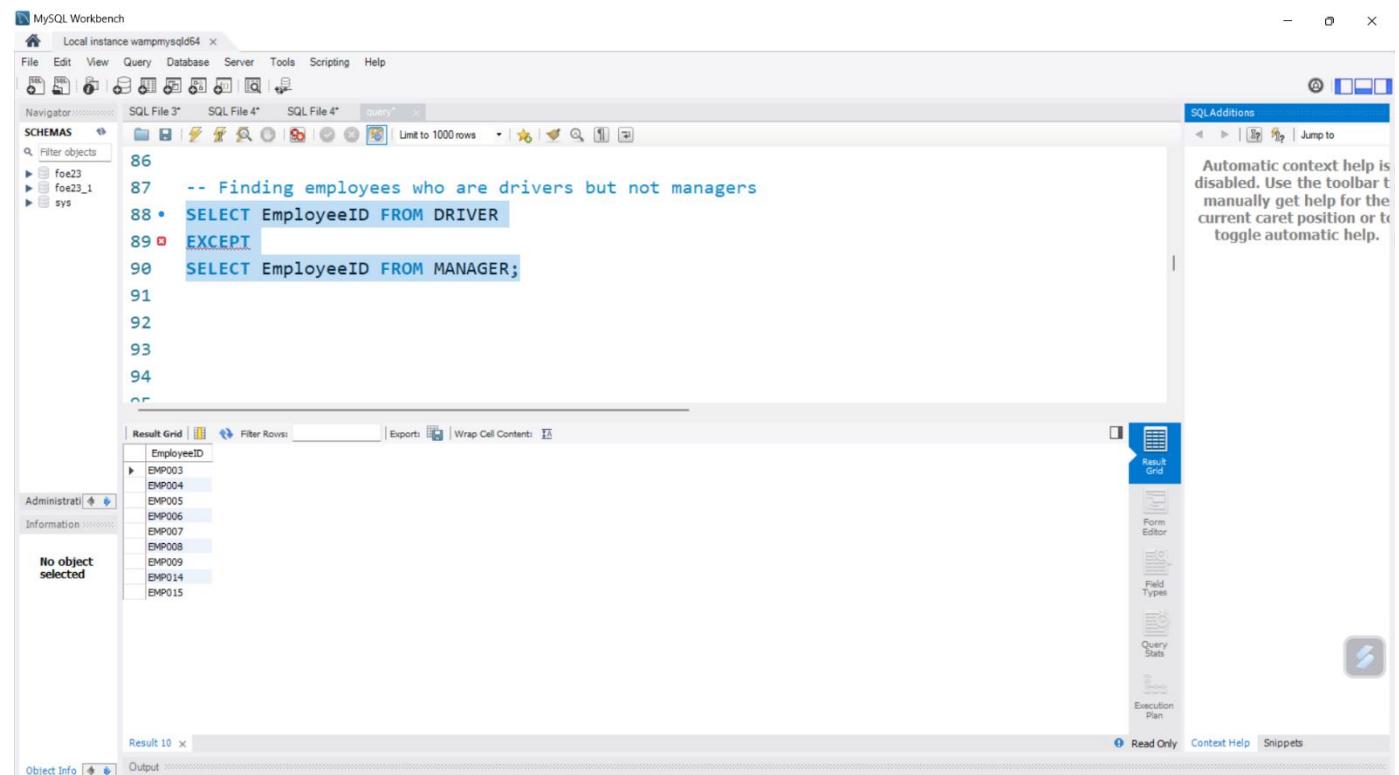
Object Info Output

Result 10 ×

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

- Set Difference



MySQL Workbench - Local instance wampmysqld54

```

86
87 -- Finding employees who are drivers but not managers
88 • SELECT EmployeeID FROM DRIVER
89 ◻ EXCEPT
90 SELECT EmployeeID FROM MANAGER;
91
92
93
94

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | EmployeeID

EMP003
EMP004
EMP005
EMP006
EMP007
EMP008
EMP009
EMP014
EMP015

No object selected

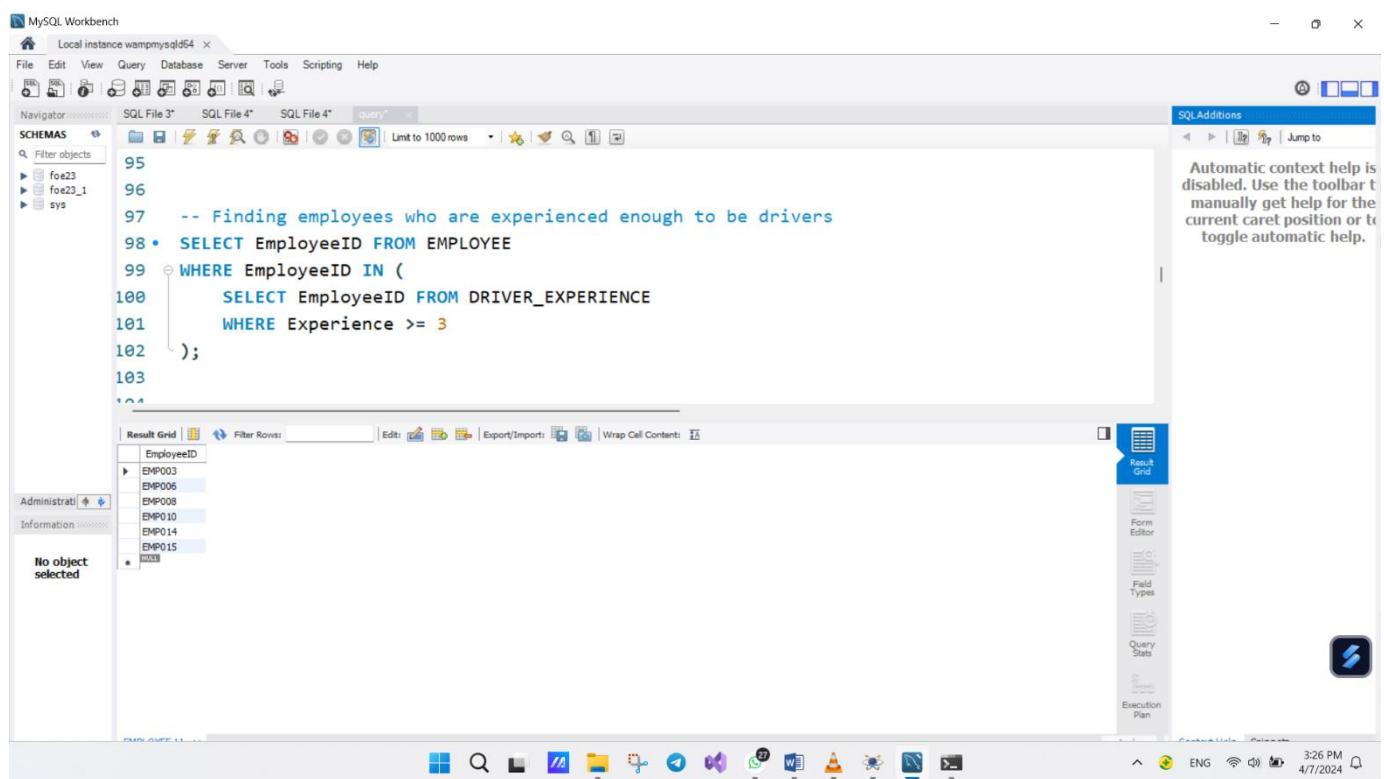
Result 10 ×

Object Info Output

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

- Division



MySQL Workbench - Local instance wampmysqld64

```

95
96
97 -- Finding employees who are experienced enough to be drivers
98 • SELECT EmployeeID FROM EMPLOYEE
99 WHERE EmployeeID IN (
100     SELECT EmployeeID FROM DRIVER_EXPERIENCE
101     WHERE Experience >= 3
102 );
103
104

```

Result Grid | Filter Rows: _____ | Edit: _____ | Export/Imports: _____ | Wrap Cell Content: _____

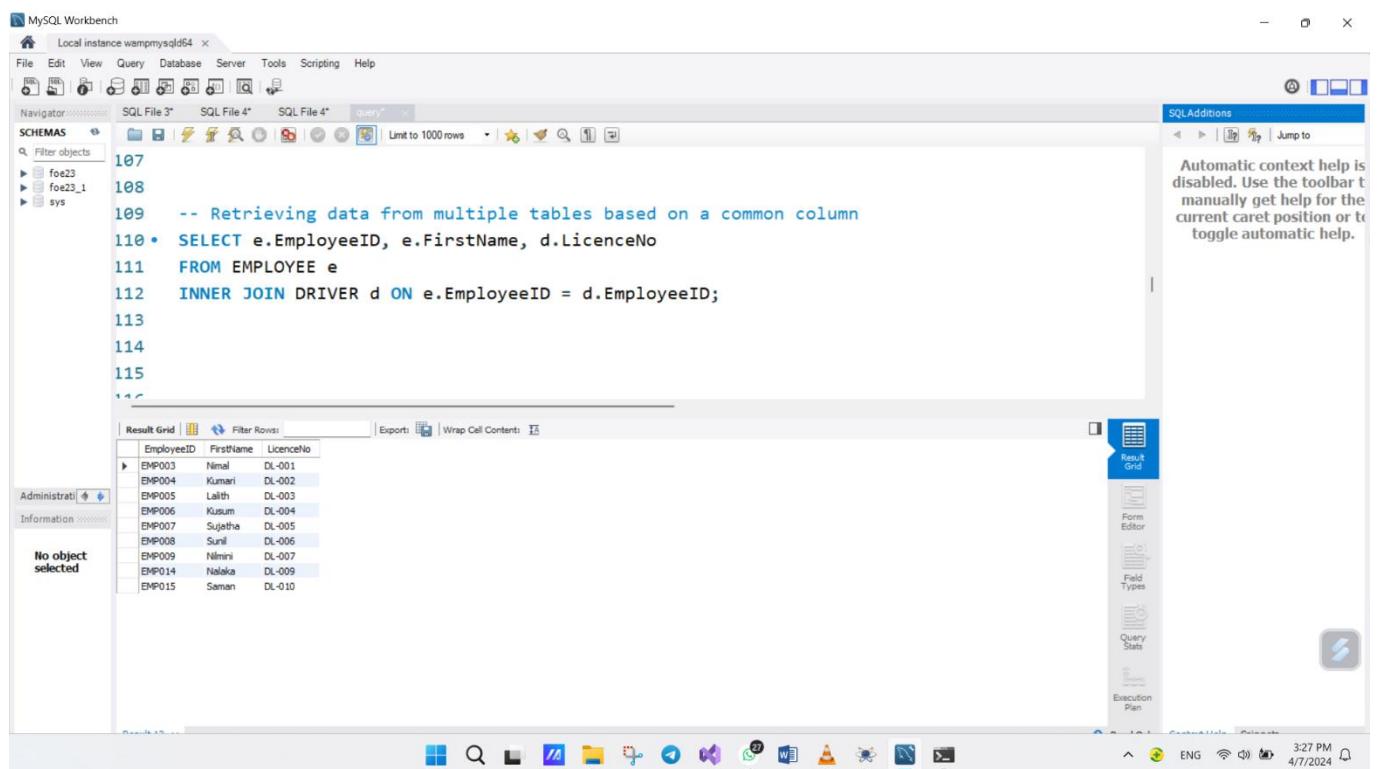
EmployeeID
EMP003
EMP005
EMP008
EMP010
EMP014
EMP015
NULL

No object selected

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

- Inner join



MySQL Workbench - Local instance wampmysqld64

```

107
108
109 -- Retrieving data from multiple tables based on a common column
110 • SELECT e.EmployeeID, e.FirstName, d.LicenceNo
111   FROM EMPLOYEE e
112   INNER JOIN DRIVER d ON e.EmployeeID = d.EmployeeID;
113
114
115
116

```

Result Grid | Filter Rows: _____ | Export: _____ | Wrap Cell Content: _____

EmployeeID	FirstName	LicenceNo
EMP003	Nimal	DL-001
EMP004	Kumari	DL-002
EMP005	Lalith	DL-003
EMP006	Kusum	DL-004
EMP007	Sujatha	DL-005
EMP008	Sunil	DL-006
EMP009	Nilmini	DL-007
EMP014	Nalaka	DL-009
EMP015	Salman	DL-010

No object selected

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

- Natural join

The screenshot shows the MySQL Workbench interface with a query editor window. The code is as follows:

```

116
117
118
119 -- Joining tables based on columns with the same name
120 • SELECT e.EmployeeID, e.FirstName, d.LicenceNo
121   FROM EMPLOYEE e
122 NATURAL JOIN DRIVER d;
123
124
125

```

The results grid displays the following data:

EmployeeID	FirstName	LicenceNo
EMP003	Nimal	DL-001
EMP004	Kumari	DL-002
EMP005	Lalith	DL-003
EMP006	Kusum	DL-004
EMP007	Sujatha	DL-005
EMP008	Sunil	DL-006
EMP009	Nilmini	DL-007
EMP014	Nalaka	DL-009
EMP015	Saman	DL-010

- Left Outer Join

The screenshot shows the MySQL Workbench interface with a query editor window. The code is as follows:

```

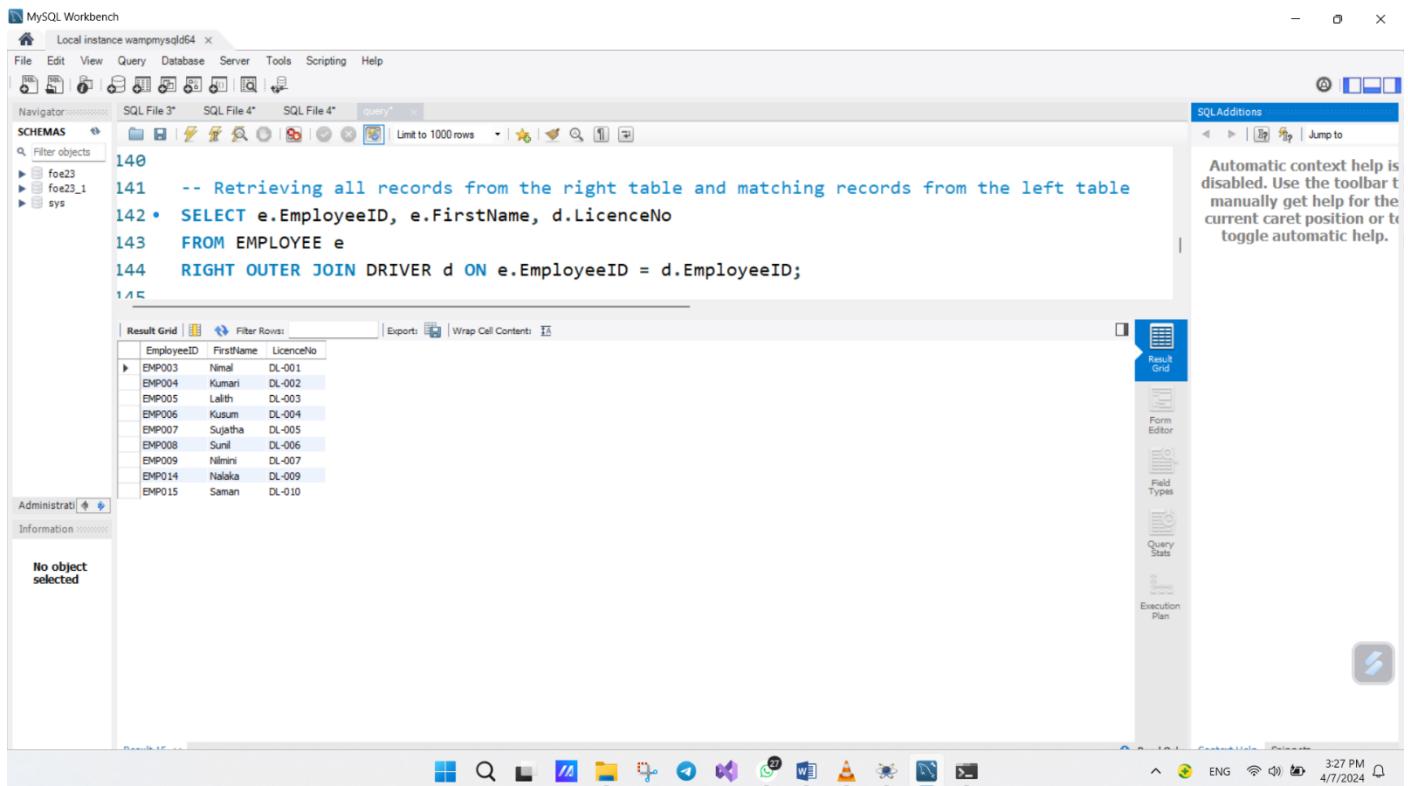
131
132 -- Retrieving all records from the left table and matching records from the right table
133 • SELECT e.EmployeeID, e.FirstName, d.LicenceNo
134   FROM EMPLOYEE e
135 LEFT OUTER JOIN DRIVER d ON e.EmployeeID = d.EmployeeID;
136

```

The results grid displays the following data:

EmployeeID	FirstName	LicenceNo
EMP001	Saman	NULL
EMP002	Malini	NULL
EMP003	Nimal	DL-001
EMP004	Kumari	DL-002
EMP005	Lalith	DL-003
EMP006	Kusum	DL-004
EMP007	Sujatha	DL-005
EMP008	Sunil	DL-006
EMP009	Nilmini	DL-007
EMP010	Ranjan	NULL
EMP011	Chandrika	NULL
EMP012	Sarath	NULL
EMP013	Malini	NULL
EMP014	Nalaka	DL-009
EMP015	Saman	DL-010
EMP016	Nishantha	NULL
EMP017	Chaminda	NULL
EMP018	Anuradha	NULL
EMP019	Rukmal	NULL
EMP021	Indika	NULL
EMP022	Dihani	NULL

- Right outer join



The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```

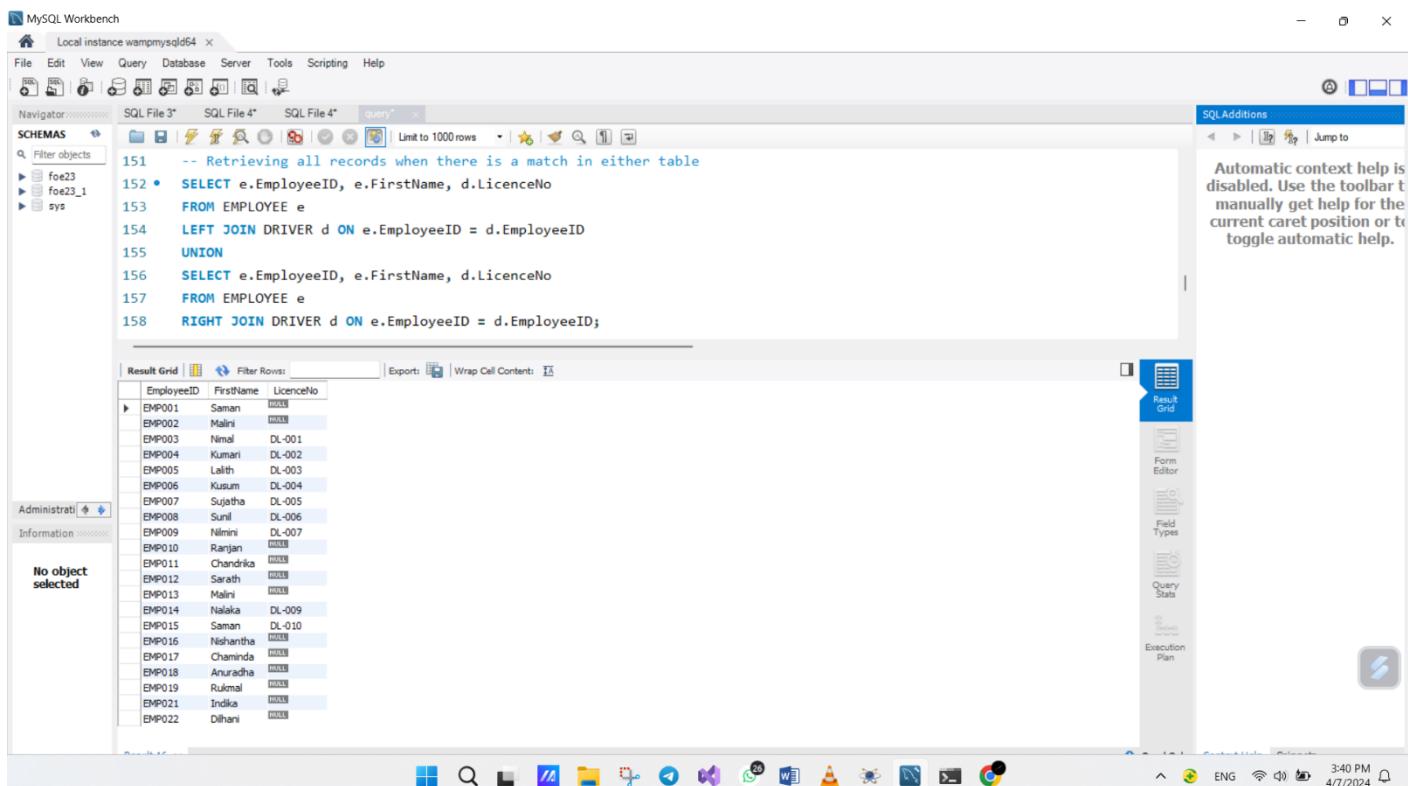
140
141 -- Retrieving all records from the right table and matching records from the left table
142 • SELECT e.EmployeeID, e.FirstName, d.LicenceNo
143   FROM EMPLOYEE e
144     RIGHT OUTER JOIN DRIVER d ON e.EmployeeID = d.EmployeeID;
145

```

The result grid displays the following data:

EmployeeID	FirstName	LicenceNo
EMP003	Nimal	DL-001
EMP004	Kumari	DL-002
EMP005	Lalith	DL-003
EMP006	Kusum	DL-004
EMP007	Sujatha	DL-005
EMP008	Sunil	DL-006
EMP009	Nimini	DL-007
EMP014	Nalaka	DL-009
EMP015	Saman	DL-010

- Full Outer join



The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```

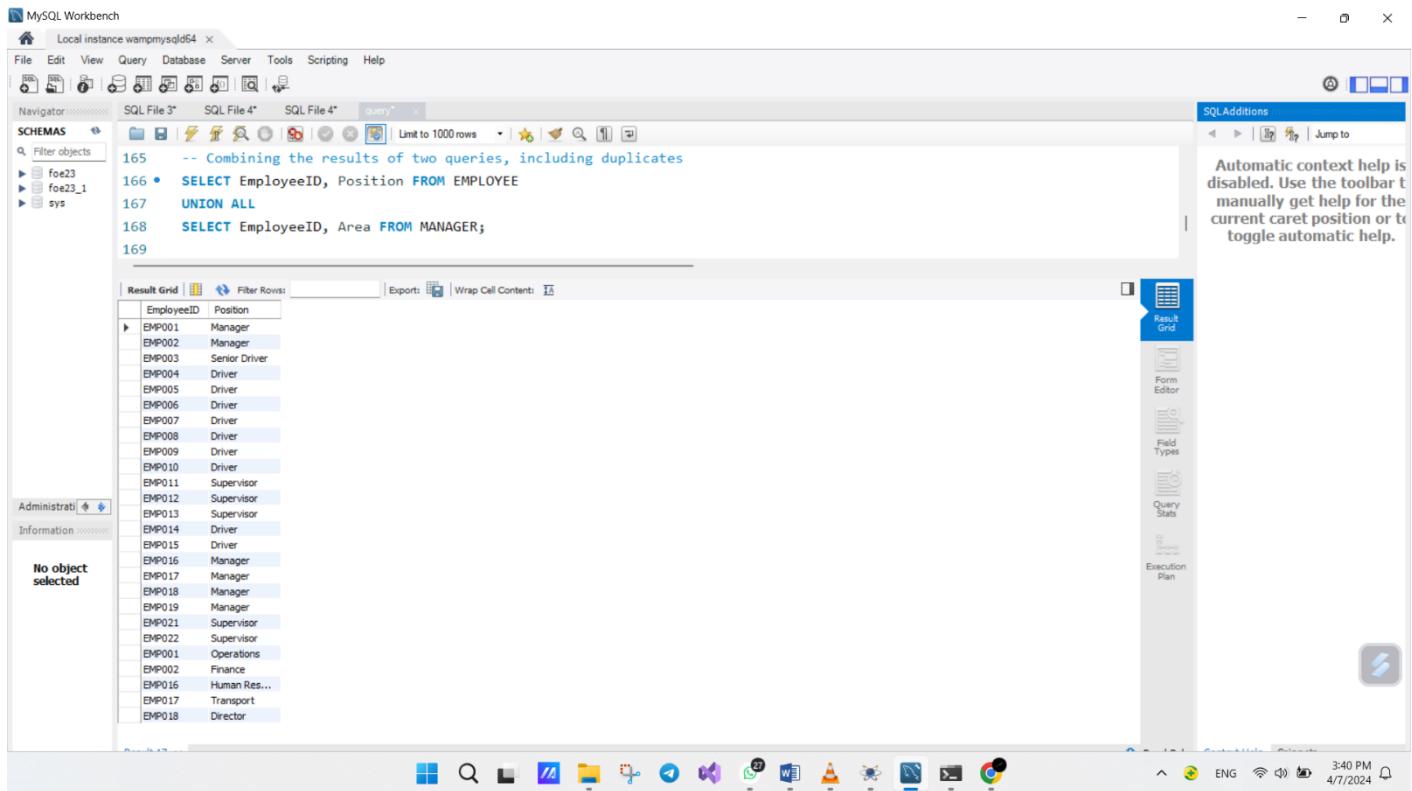
151 -- Retrieving all records when there is a match in either table
152 • SELECT e.EmployeeID, e.FirstName, d.LicenceNo
153   FROM EMPLOYEE e
154     LEFT JOIN DRIVER d ON e.EmployeeID = d.EmployeeID
155   UNION
156   SELECT e.EmployeeID, e.FirstName, d.LicenceNo
157   FROM EMPLOYEE e
158     RIGHT JOIN DRIVER d ON e.EmployeeID = d.EmployeeID;
159

```

The result grid displays the following data:

EmployeeID	FirstName	LicenceNo
EMP001	Saman	NULL
EMP002	Malini	NULL
EMP003	Nimal	DL-001
EMP004	Kumari	DL-002
EMP005	Lalith	DL-003
EMP006	Kusum	DL-004
EMP007	Sujatha	DL-005
EMP008	Sunil	DL-006
EMP009	Nimini	DL-007
EMP010	Ranjan	NULL
EMP011	Chandrika	NULL
EMP012	Sarath	NULL
EMP013	Malini	NULL
EMP014	Nalaka	DL-009
EMP015	Saman	DL-010
EMP016	Nishantha	NULL
EMP017	Chaminda	NULL
EMP018	Anuradha	NULL
EMP019	Rukmal	NULL
EMP021	Indika	NULL
EMP022	Dihani	NULL

- Outer union



The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```

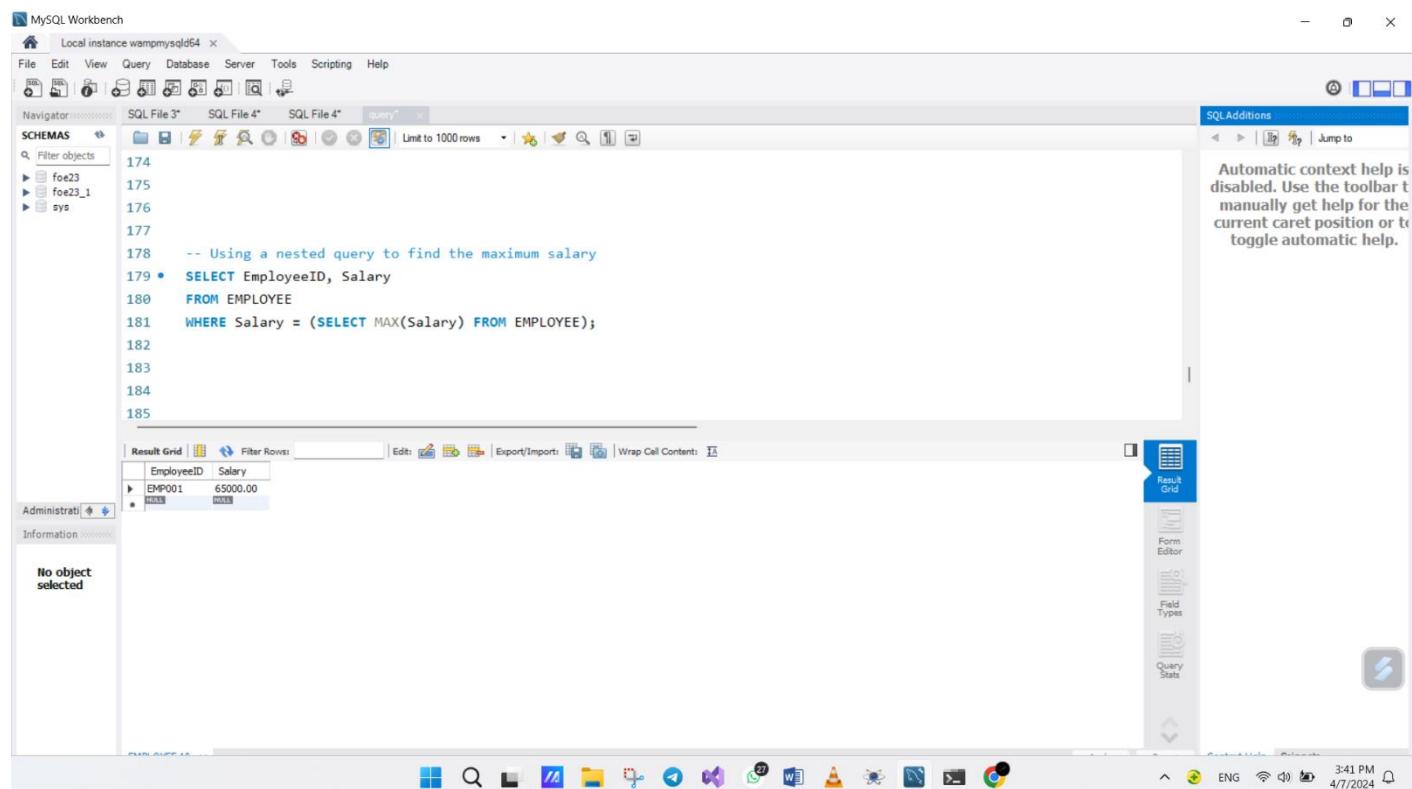
165 -- Combining the results of two queries, including duplicates
166 • SELECT EmployeeID, Position FROM EMPLOYEE
167 UNION ALL
168 SELECT EmployeeID, Area FROM MANAGER;
169

```

The result grid displays the combined data from both tables:

EmployeeID	Position
EMP001	Manager
EMP002	Manager
EMP003	Senior Driver
EMP004	Driver
EMP005	Driver
EMP006	Driver
EMP007	Driver
EMP008	Driver
EMP009	Driver
EMP010	Driver
EMP011	Supervisor
EMP012	Supervisor
EMP013	Supervisor
EMP014	Driver
EMP015	Driver
EMP016	Manager
EMP017	Manager
EMP018	Manager
EMP019	Manager
EMP021	Supervisor
EMP022	Supervisor
EMP001	Operations
EMP002	Finance
EMP016	Human Res...
EMP017	Transport
EMP018	Director

- Nested query -1



The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```

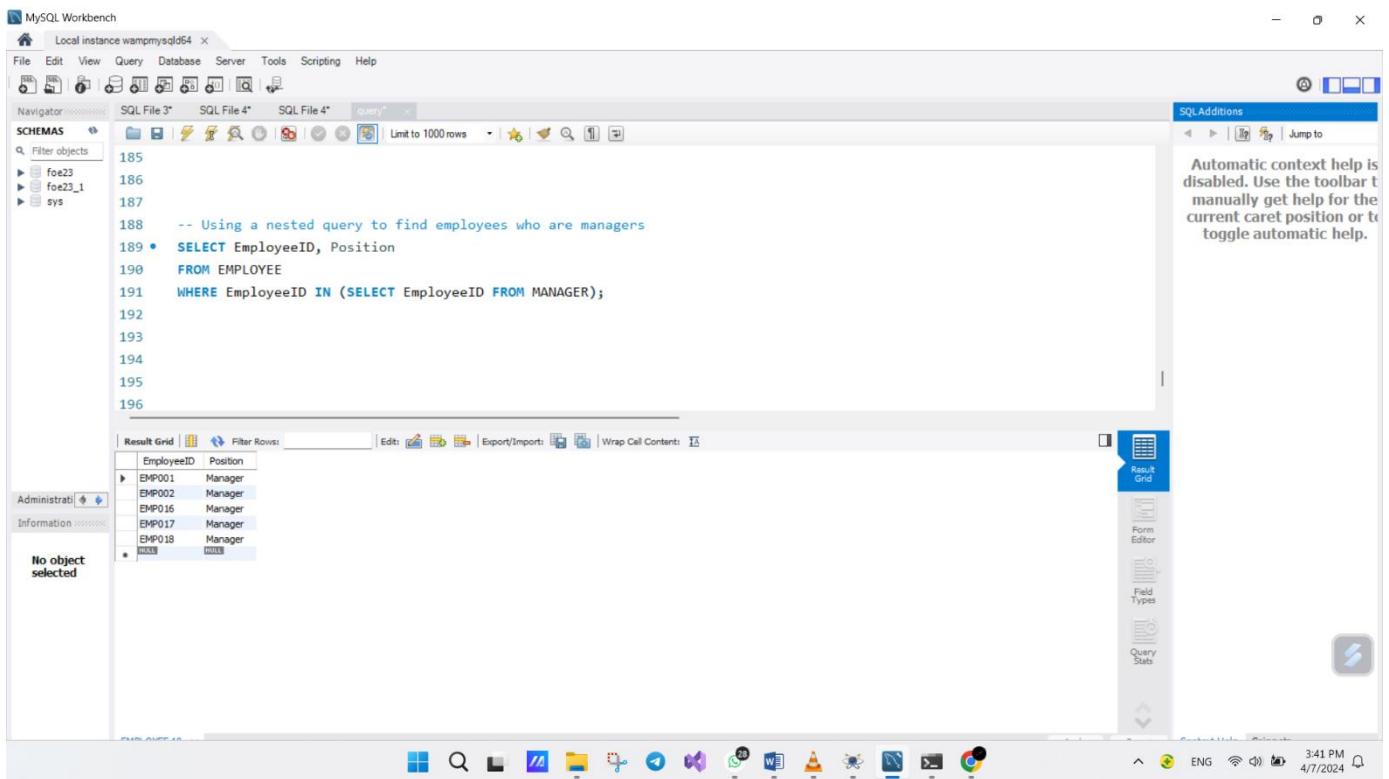
174
175
176
177
178 -- Using a nested query to find the maximum salary
179 • SELECT EmployeeID, Salary
180   FROM EMPLOYEE
181 WHERE Salary = (SELECT MAX(Salary) FROM EMPLOYEE);
182
183
184
185

```

The result grid displays the maximum salary:

EmployeeID	Salary
EMP001	65000.00

- Nested query -2



The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```

185
186
187
188 -- Using a nested query to find employees who are managers
189 • SELECT EmployeeID, Position
190   FROM EMPLOYEE
191 WHERE EmployeeID IN (SELECT EmployeeID FROM MANAGER);
192
193
194
195
196

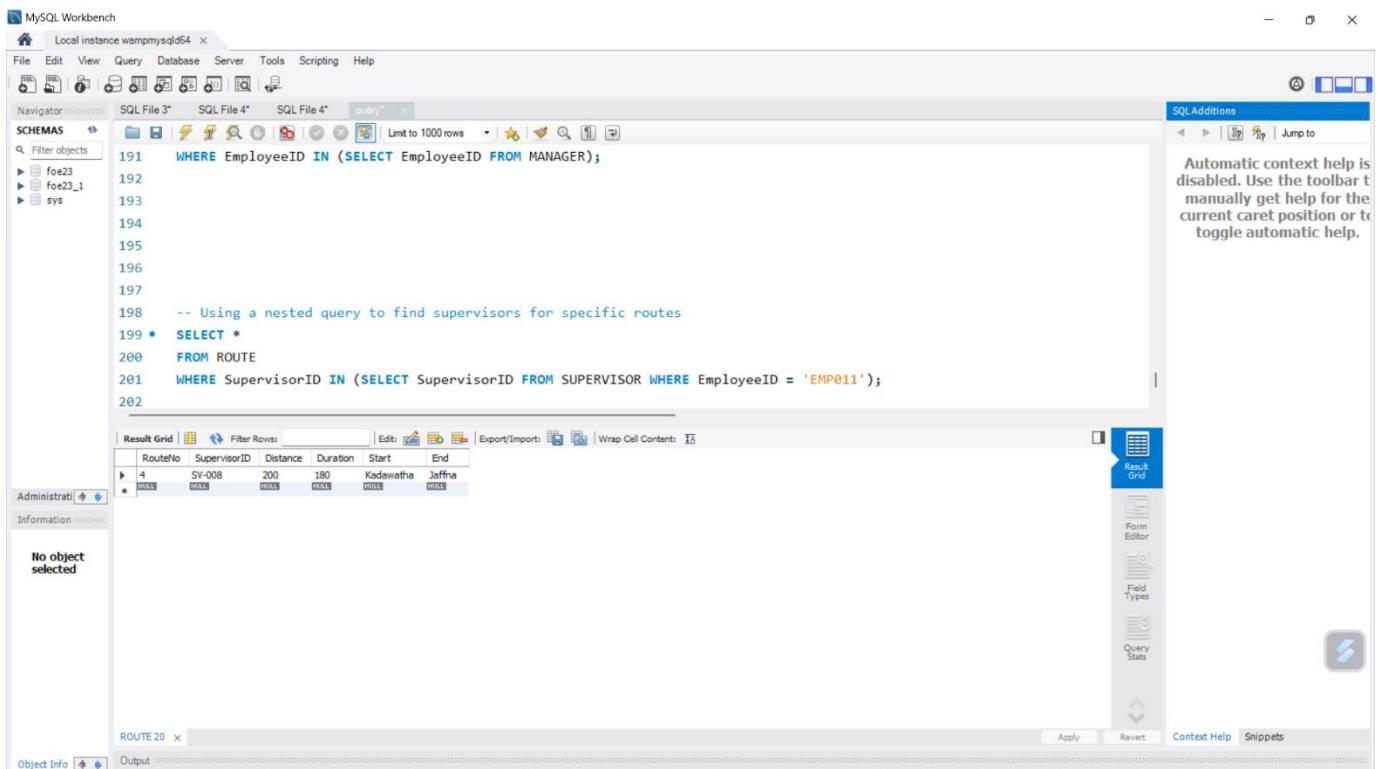
```

The results grid displays the following data:

EmployeeID	Position
EMP001	Manager
EMP002	Manager
EMP016	Manager
EMP017	Manager
EMP018	Manager

A message in the status bar indicates: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."

- Nested query -3



The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```

191 WHERE EmployeeID IN (SELECT EmployeeID FROM MANAGER);
192
193
194
195
196
197
198 -- Using a nested query to find supervisors for specific routes
199 • SELECT *
200   FROM ROUTE
201 WHERE SupervisorID IN (SELECT SupervisorID FROM SUPERVISOR WHERE EmployeeID = 'EMP011');
202

```

The results grid displays the following data:

RouteNo	SupervisorID	Distance	Duration	Start	End
4	SV-008	200	180	Kadawatha	Jaffna
*	NULL	NULL	NULL	NULL	NULL

A message in the status bar indicates: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."

Chapter 5 - Tuning

- Nested Query-1 Tuning

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

- Schemas:** Local instance wampmysqld64
- Query Editor:** Contains the following SQL code:

```
L94
L95 -- Using a nested query to find the maximum salary
L96 • explain(SELECT EmployeeID, Salary
L97   FROM EMPLOYEE
L98   WHERE Salary = (SELECT MAX(Salary) FROM EMPLOYEE));
L99
L100
L101
L102
L103
L104
L105
```
- Result Grid:** Shows the execution plan for the query. The first row is highlighted.

ID	Select_Type	Table	Partitions	Type	Possible_Keys	Key	Key_Len	Ref	Rows	Filtered	Extra
1	PRIMARY	EMPLOYEE	HASH	ALL	HASH	HASH	HASH	HASH	21	10.00	Using where
2	SUBQUERY	EMPLOYEE	HASH	ALL	HASH	HASH	HASH	HASH	21	100.00	HASH

- Information Panel:** Shows "No object selected".
- Toolbar:** Includes icons for File, Edit, View, Database, Server, Tools, Scripting, Help, and various database management functions.
- System Tray:** Shows system status including battery level, signal strength, and date/time (8:03 PM, 4/7/2024).

Before Tuning

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

- Schemas:** Local instance wampmysqld64
- Query Editor:** Contains the following SQL code:

```
203
204
205
206 -- tuned the nested query to find the maximum salary
207 • explain(SELECT EmployeeID, Salary
208   FROM EMPLOYEE
209   ORDER BY Salary DESC
210   LIMIT 1);
211
212
213
214
```
- Result Grid:** Shows the execution plan for the tuned query. The first row is highlighted.

ID	Select_Type	Table	Partitions	Type	Possible_Keys	Key	Key_Len	Ref	Rows	Filtered	Extra
1	SIMPLE	EMPLOYEE	HASH	ALL	HASH	HASH	HASH	HASH	21	100.00	Using filesort

- Information Panel:** Shows "No object selected".
- Toolbar:** Includes icons for File, Edit, View, Database, Server, Tools, Scripting, Help, and various database management functions.
- System Tray:** Shows system status including battery level, signal strength, and date/time (8:04 PM, 4/7/2024).

After Tuning

- Nested Query-2 Tuning

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

- Navigator:** Shows the schema `private_bus_company` selected.
- SQL File 4*** contains the following SQL code:

```

1
2 * use private_bus_company;
3
4 * EXPLAIN(SELECT *
5   FROM ROUTE
6   WHERE SupervisorID IN (SELECT SupervisorID FROM SUPERVISOR WHERE EmployeeID = 'EMP011'));
7

```
- Result Grid:** Displays the execution plan for the EXPLAIN command. The output is:

ID	Select Type	Table	Partitions	Type	Possible Keys	Key	Key Len	Ref	Rows	Filtered	Extra
1	SIMPLE	SUPERVISOR	NULL	const	PRIMARY,EmployeeID	EmployeeID	42	const	1	100.00	Using index
1	SIMPLE	ROUTE	NULL	ref	SupervisorID	SupervisorID	82	const	2	100.00	NULL
- Output:** Shows the results of the EXPLAIN command:

#	Time	Action	Message	Duration / Fetch
1	14:12:24	EXPLAIN(SELECT * FROM ROUTE WHERE SupervisorID IN (SELECT SupervisorID FROM SUPERVISOR W...	2 row(s) returned	0.000 sec / 0.000 sec

Before Tuning

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

- Navigator:** Shows the schema `private_bus_company` selected.
- SQL File 4*** contains the following SQL code:

```

1
2 * use private_bus_company;
3
4 * EXPLAIN(SELECT r.* 
5   FROM ROUTE r
6   JOIN SUPERVISOR s ON r.SupervisorID = s.SupervisorID
7   WHERE s.EmployeeID = 'EMP011');
7

```
- Result Grid:** Displays the execution plan for the EXPLAIN command. The output is:

ID	Select Type	Table	Partitions	Type	Possible Keys	Key	Key Len	Ref	Rows	Filtered	Extra
1	SIMPLE	s	NULL	const	PRIMARY,EmployeeID	EmployeeID	42	const	1	100.00	Using index
1	SIMPLE	r	NULL	ref	SupervisorID	SupervisorID	82	const	2	100.00	NULL
- Output:** Shows the results of the EXPLAIN command:

#	Time	Action	Message	Duration / Fetch
1	14:08:05	use private_bus_company	0 row(s) affected	0.000 sec
2	14:08:10	SELECT r.* FROM ROUTE r JOIN SUPERVISOR s ON r.SupervisorID = s.SupervisorID WHERE s.EmployeeID ...	2 row(s) returned	0.000 sec / 0.000 sec
3	14:10:57	EXPLAIN(SELECT r.* FROM ROUTE r JOIN SUPERVISOR s ON r.SupervisorID = s.SupervisorID WHERE s.EmployeeID = 'EMP011');	2 row(s) returned	0.016 sec / 0.000 sec

After Tuning

- Nested Query-3 Tuning

```

1
2 * use private_bus_company;
3
4 * EXPLAIN(SELECT EmployeeID, Position
5   FROM EMPLOYEE
6 WHERE EmployeeID IN (SELECT EmployeeID FROM MANAGER));
7

```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	MANAGER		index	EmployeeID	EmployeeID	42	NULL	2	100.00	Using index
1	SIMPLE	EMPLOYEE		eq_ref	PRIMARY	PRIMARY	42	private_bus_company.MANAGER.EmployeeID	1	100.00	

Result Grid | Filter Rows: Export: Wrap Cell Content: Result Grid | Form Editor | Field Types

Output

#	Time	Action	Message	Duration / Fetch
1	14:16:36	EXPLAIN(SELECT EmployeeID, Position FROM EMPLOYEE WHERE EmployeeID IN (SELECT EmployeeID FROM MANAGER))	2 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Before Tuning

```

1
2 * use private_bus_company;
3
4 * EXPLAIN(SELECT e.EmployeeID, e.Position
5   FROM EMPLOYEE e
6   JOIN MANAGER m ON e.EmployeeID = m.EmployeeID);
7

```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	m		index	EmployeeID	EmployeeID	42	NULL	2	100.00	Using index
1	SIMPLE	e		eq_ref	PRIMARY	PRIMARY	42	private_bus_company.m.EmployeeID	1	100.00	

Result Grid | Filter Rows: Export: Wrap Cell Content: Result Grid | Form Editor | Field Types

Output

#	Time	Action	Message	Duration / Fetch
1	14:14:59	EXPLAIN(SELECT e.EmployeeID, e.Position FROM EMPLOYEE e JOIN MANAGER m ON e.EmployeeID = m.E...)	2 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

After Tuning

- Full outer join tuning

The screenshot shows the MySQL Workbench interface with the Explain plan for a query. The query is:

```

158
159
160 -- Retrieving all records when there is a match in either table
161 • explain(SELECT e.EmployeeID, e.FirstName, d.LicenceNo
162   FROM EMPLOYEE e
163   LEFT JOIN DRIVER d ON e.EmployeeID = d.EmployeeID
164 UNION
165   SELECT e.EmployeeID, e.FirstName, d.LicenceNo
166   FROM EMPLOYEE e
167   RIGHT JOIN DRIVER d ON e.EmployeeID = d.EmployeeID);
168
169
170
171
172
173

```

The Explain plan output is:

ID	Select Type	Table	Partitions	Type	Possible Keys	Key	Key Len	Ref	Rows	Filtered	Extra
1	PRIMARY	e		ALL	NULL	NULL	NULL	NULL	21	100.00	NULL
1	PRIMARY	d		ALL	eq_ref	EmployeeID	42	private_bus_company.e.EmployeeID	1	100.00	Using index
2	UNION	d		index	NULL	EmployeeID	42	NULL	9	100.00	Using index
2	UNION	e		ALL	eq_ref	PRIMARY	42	private_bus_company.d.EmployeeID	1	100.00	NULL
3	UNION RESULT	<union1,2>		ALL	NULL	NULL	NULL	NULL	1	100.00	Using temporary

Result Grid: Read Only | Context Help | Snippets

Information: Result Grid | Form Editor

Administrators: No object selected

File Edit View Query Database Server Tools Scripting Help

Navigator: Local instance wampmysqld64

SQL File 3* SQL File 4* update delete query* Limit to 1000 rows

158
159
160 -- Retrieving all records when there is a match in either table
161 • explain(SELECT e.EmployeeID, e.FirstName, d.LicenceNo
162 FROM EMPLOYEE e
163 LEFT JOIN DRIVER d ON e.EmployeeID = d.EmployeeID
164 UNION
165 SELECT e.EmployeeID, e.FirstName, d.LicenceNo
166 FROM EMPLOYEE e
167 RIGHT JOIN DRIVER d ON e.EmployeeID = d.EmployeeID);
168
169
170
171
172
173

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Before Tuning

The screenshot shows the MySQL Workbench interface with the Explain plan for a query. The query is:

```

148
149
150 -- Retrieving all records from the right table and matching records from the left table
151 • EXPLAIN(SELECT e.EmployeeID, e.FirstName, d.LicenceNo
152   FROM EMPLOYEE e
153   RIGHT OUTER JOIN DRIVER d ON e.EmployeeID = d.EmployeeID);
154
155
156
157
158
159
160
161
162
163

```

The Explain plan output is:

ID	Select Type	Table	Partitions	Type	Possible Keys	Key	Key Len	Ref	Rows	Filtered	Extra
1	SIMPLE	d		ALL	index	NULL	42	EmployeeID	9	100.00	Using index
1	SIMPLE	e		ALL	eq_ref	PRIMARY	42	private_bus_company.d.EmployeeID	1	100.00	NULL

Result Grid: Read Only | Context Help | Snippets

Information: Result Grid | Form Editor

Administrators: No object selected

File Edit View Query Database Server Tools Scripting Help

Navigator: Local instance wampmysqld64

SQL File 3* SQL File 4* update delete query* Limit to 1000 rows

148
149
150 -- Retrieving all records from the right table and matching records from the left table
151 • EXPLAIN(SELECT e.EmployeeID, e.FirstName, d.LicenceNo
152 FROM EMPLOYEE e
153 RIGHT OUTER JOIN DRIVER d ON e.EmployeeID = d.EmployeeID);
154
155
156
157
158
159
160
161
162
163

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

After Tuning

- Outer union tuning

```

148
149
150 -- Retrieving all records from the right table and matching records from the left table
151 • EXPLAIN(SELECT e.EmployeeID, e.FirstName, d.LicenceNo
152   FROM EMPLOYEE e
153     RIGHT OUTER JOIN DRIVER d ON e.EmployeeID = d.EmployeeID);
154
155
156
157
158
159
160
161
162
163

```

No object selected

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶	1	SIMPLE	d	HOLI	index	HOLI	EmployeeID	42	HOLI	9	100.00	Using index
	1	SIMPLE	e	HOLI	eq_ref	PRIMARY	PRIMARY	42	private_bus_company.d.EmployeeID	1	100.00	HOLI

Result 48 x

Output:

Action Output

Result Grid Form Editor

Read Only Context Help Snippets

8:14 PM 4/7/2024

Before Tuning

This query is tune using indexing

```

157
158 -- Ensure indexes on EmployeeID columns
159 • CREATE INDEX idx_employee_id ON EMPLOYEE(EmployeeID);
160 • CREATE INDEX idx_driver_employee_id ON DRIVER(EmployeeID);
161
162 -- then try the query
163 • EXPLAIN SELECT e.EmployeeID, e.FirstName, d.LicenceNo
164   FROM EMPLOYEE e
165     RIGHT OUTER JOIN DRIVER d ON e.EmployeeID = d.EmployeeID;
166
167
168
169
170
171
172

```

No object selected

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶	1	SIMPLE	d	HOLI	index	HOLI	EmployeeID	42	HOLI	10	100.00	Using index
	1	SIMPLE	e	HOLI	eq_ref	PRIMARY,idx_employee_id	PRIMARY	42	private_bus_company.d.EmployeeID	1	100.00	HOLI

Result 49 x

Output:

Action Output

Result Grid Form Editor

Read Only Context Help Snippets

8:16 PM 4/7/2024

After Tuning

- Left Outer Join Tuning

```

138
139
140
141 -- Retrieving all records from the left table and matching records from the right table
142 • EXPLAIN SELECT e.EmployeeID, e.FirstName, d.LicenceNo
143 FROM EMPLOYEE e
144 LEFT OUTER JOIN DRIVER d ON e.EmployeeID = d.EmployeeID;
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165

```

No object selected

Result Grid | Filter Rows: Export: Wrap Cell Content:

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶	1	SIMPLE	e	<input type="checkbox"/>	ALL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	21	100.00	<input type="checkbox"/>
	1	SIMPLE	d	<input type="checkbox"/>	eq_ref	EmployeeID, idx_driver_employee_id	EmployeeID	42	private_bus_company.e.EmployeeID	1	100.00	Using index

Result 50 x

Output

Action Output

Object Info

Message

100 20:15:55 EXPLAIN SELECT e.EmployeeID, e.FirstName, d.LicenceNo FROM EMPLOYEE e RIGHT OUTER JOIN DRIVER d ON... 2 row(s) returned

Duration / Fetch 0.000 sec / 0.000 sec

Before Tuning

```

150
151
152
153 -- create indexes
154 • CREATE INDEX idx_employee_id ON EMPLOYEE(EmployeeID);
155 • CREATE INDEX idx_driver_employee_id ON DRIVER(EmployeeID);
156
157
158
159 -- then try the query
160 • EXPLAIN SELECT e.EmployeeID, e.FirstName, d.LicenceNo
161 FROM EMPLOYEE e
162 LEFT OUTER JOIN DRIVER d ON e.EmployeeID = d.EmployeeID;
163
164
165

```

No object selected

Result Grid | Filter Rows: Export: Wrap Cell Content:

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶	1	SIMPLE	e	<input type="checkbox"/>	ALL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	21	100.00	<input type="checkbox"/>
	1	SIMPLE	d	<input type="checkbox"/>	eq_ref	EmployeeID, idx_driver_employee_id	EmployeeID	42	private_bus_company.e.EmployeeID	1	100.00	Using index

Result 56 x

Output

Action Output

Object Info

Message

8:25 PM 4/7/2024

After Tuning

- Set Difference -1 Tuning

The screenshot shows the MySQL Workbench interface with the SQL Editor tab open. The code is:

```

1
2 • use private_bus_company;
3
4 • EXPLAIN(SELECT EmployeeID FROM EMPLOYEE
5 WHERE EmployeeID IN (
6     SELECT EmployeeID FROM DRIVER_EXPERIENCE
7     WHERE Experience >= 3
8 ) );
9

```

The Result Grid shows the execution plan:

ID	Select Type	Table	Partitions	Type	Possible Keys	Key	Key Len	Ref	Rows	Filtered
1	SIMPLE	DRIVER_EXPERIENCE	NULL	ALL	PRIMARY	NULL	NULL	NULL	8	33.33
1	SIMPLE	EMPLOYEE	NULL	eq_ref	PRIMARY	PRIMARY	42	private_bus_company.DRIVER_EXPERIENCE.E...	1	100.00

The Output pane shows the command and its execution details:

```

# Time Action Message Duration / Fetch
1 14:25:13 EXPLAIN(SELECT EmployeeID FROM EMPLOYEE WHERE EmployeeID IN ( SELECT EmployeeID FROM D... 2 row(s) returned 0.000 sec / 0.000 sec

```

Before Tuning

The screenshot shows the MySQL Workbench interface with the SQL Editor tab open. The code is:

```

5 FROM EMPLOYEE
6
7 EXCEPT
8
9
10 SELECT e.EmployeeID
11 FROM EMPLOYEE e
12 JOIN DRIVER_EXPERIENCE de ON e.EmployeeID = de.EmployeeID
13 WHERE de.Experience < 3;
14

```

The Result Grid shows the execution plan:

ID	Select Type	Table	Partitions	Type	Possible Keys	Key	Key Len	Ref	Rows	Filtered	Extra
1	PRIMARY	EMPLOYEE	NULL	INDEX	NULL	PRIMARY	42	NULL	15	100.00	Using index
2	EXCEPT	de	NULL	ALL	PRIMARY	NULL	NULL	NULL	8	33.33	Using where
2	EXCEPT	e	NULL	eq_ref	PRIMARY	PRIMARY	42	private_bus_company.de.EmployeeID	1	100.00	Using index
3	EXCEPT RESULT	<except1,2>	NULL	ALL	NULL	NULL	NULL	NULL			Using temporary

The Output pane shows two commands and their execution details:

```

# Time Action Message Duration / Fetch
1 14:25:13 EXPLAIN(SELECT EmployeeID FROM EMPLOYEE WHERE EmployeeID IN ( SELECT EmployeeID FROM D... 2 row(s) returned 0.000 sec / 0.000 sec
2 14:26:03 EXPLAIN(SELECT EmployeeID FROM EMPLOYEE EXCEPT SELECT e.EmployeeID FROM EMPLOYEE e JOI... 4 row(s) returned 0.000 sec / 0.000 sec

```

After Tuning

- Set Difference -2 Tuning

```

1
2 • use private_bus_company;
3
4 • EXPLAIN(
5   SELECT EmployeeID, Position FROM EMPLOYEE
6   UNION ALL
7   SELECT EmployeeID, Area FROM MANAGER
8 );
9

```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	EMPLOYEE	HASH	ALL	HASH	HASH	HASH	HASH	15	100.00	HASH
2	UNION	MANAGER	HASH	ALL	HASH	HASH	HASH	HASH	2	100.00	HASH

Result 10 ×

Action Output

#	Time	Action	Message	Duration / Fetch
1	14:30:05	EXPLAIN(SELECT EmployeeID, Position FROM EMPLOYEE UNION ALL SELECT EmployeeID, Area FROM M...)	2 row(s) returned	0.000 sec / 0.000 sec

Before Tuning

```

3
4 • EXPLAIN(
5   SELECT EmployeeID, Position FROM EMPLOYEE
6
7   EXCEPT
8
9   SELECT EmployeeID, Area AS Position FROM MANAGER
10 )
11

```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	EMPLOYEE	HASH	ALL	HASH	HASH	HASH	HASH	15	100.00	HASH
2	EXCEPT	MANAGER	HASH	ALL	HASH	HASH	HASH	HASH	2	100.00	HASH
3	EXCEPT RESULT	<except1,2>	HASH	ALL	HASH	HASH	HASH	HASH	Using temporary		

Result 12 ×

Action Output

#	Time	Action	Message	Duration / Fetch
1	14:31:04	EXPLAIN(SELECT EmployeeID, Position FROM EMPLOYEE EXCEPT SELECT EmployeeID, Area AS Position...)	3 row(s) returned	0.016 sec / 0.000 sec

After Tuning

- Intersection Tuning

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Schemas

private_bus_company

Result Grid

ID	Select Type	Table	Partitions	Type	Possible Keys	Key	Key Len	Ref	Rows	Filtered	Extra
1	PRIMARY	MANAGER		index		EmployeeID	42	NULL	2	100.00	Using index
2	INTERSECT	SUPERVISOR		index		EmployeeID	42	NULL	2	100.00	Using index
3	INTERSECT RESULT	<intersect1,2>		ALL			NULL	NULL	NULL	NULL	Using temporary

Result 13 x

Action Output

#	Time	Action	Message	Duration / Fetch
1	14:36:17	EXPLAIN(SELECT EmployeeID FROM MANAGER INTERSECT SELECT EmployeeID FROM SUPERVISOR)	3 row(s) returned	0.000 sec / 0.000 sec

No object selected

Object Info Session

32°C Partly sunny

2:36 PM 4/8/2024

Before Tuning

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Schemas

private_bus_company

Result Grid

ID	Select Type	Table	Partitions	Type	Possible Keys	Key	Key Len	Ref	Rows	Filtered	Extra
1	SIMPLE	m		index		EmployeeID	EmployeeID	42	2	100.00	Using index
1	SIMPLE	s		eq_ref		EmployeeID	EmployeeID	42	1	100.00	Using index

Result 14 x

Action Output

#	Time	Action	Message	Duration / Fetch
1	14:36:17	EXPLAIN(SELECT EmployeeID FROM MANAGER INTERSECT SELECT EmployeeID FROM SUPERVISOR)	3 row(s) returned	0.000 sec / 0.000 sec
2	14:37:08	EXPLAIN(SELECT m.EmployeeID FROM MANAGER m JOIN SUPERVISOR s ON m.EmployeeID = s.EmployeeID)	2 row(s) returned	0.000 sec / 0.000 sec

No object selected

Object Info Session

32°C Partly sunny

2:37 PM 4/8/2024

After Tuning