



**Data Base Systems Mini Project**

**Smart Parking Management**

**System**

A report submitted to the  
Department of Electrical and Information Engineering  
Faculty of Engineering  
University of Ruhuna Sri Lanka

On 5th of April 2024  
In completing an assignment for the module  
EE4350 Data Base systems

By:  
Group 38

EG/2021/4620 Kularathna M.D.Y.B  
EG/2021/4622 Kumara A.R.M.D.D  
EG/2021/4623 Kumara D.G.A.C.D

# Contents

Chapter 1: Requirement Analysis.....	4
1.1 Functional Requirements.....	4
1.2 Data Requirements .....	5
Chapter 2: Conceptual Design.....	10
Chapter 3: Implementation.....	11
3.1 Table Implementation .....	11
3.2 Creating Foreign Key Constraint.....	20
3.3 Inserting Data for Tables .....	22
3.4 Update And Delete.....	30
Chapter 4: Transaction.....	38
.....	41
Complex Queries .....	42
Chapter 5: Tuning.....	48

## **List of figures**

Figure 1: ER diagram of the Smart Parking Management System.....	10
Figure 2: UML Class Diagram of the Smart Parking Management System .....	10

# **Chapter 1: Requirement Analysis**

## **1.1 Functional Requirements**

All of the program's desired operations are incorporated step-by-step in accordance with this chapter. This database accomplishes the following things in order to do that.

- Data retrieval
- Ability to alter the database's content at any time.
- Capable of completing potential data gaps or missing data.
- Capability to edit data that has already been entered into the database.
- The ability to add comments to data when the data is incomplete or ambiguous.
- The capacity to provide data with enhanced performance.

This schema is created to represent the smart parking management system and all the required entities of this schema have been covered. They are Employee, Employee Phone Number, Parking Lot, Maintenance Request, Parking Space, Security Camera, Parking Lot Maintenance, Vehicle, Parking Ticket, Payment, User, Reservation, User Phone Number, Spot History Date, Supervise. All the needed data for the schema shown in this database.

## **1.2 Data Requirements**

The attributes of each entity are shown below.

1. Employee:

- EmployeeId
- FirstName
- LastName
- Email
- JobTitle

2. Shift:

- ShiftId
- EmployeeId
- ShiftTime
- EndTime
- ShiftDate

3. EmployeePhoneNumber:

- PhoneId
- EmployeeId
- EmployeePhoneNumber

4. ParkingLot:

- ParkingLotId
- TotalSpace
- AvailableSpace
- Location
- Name
- OccupancyRate

5. MaintenanceRequest:

- RequestId
- EmployeeId
- RequestedDate
- ParkingLotId
- Status
- CompletedDate

6. ParkingSpace:

- ParkingSpaceId
- ParkingLotId
- SpaceNumber
- Status

7. Security Camera:

- SecurityCameraId
- ParkingLotId
- Location
- Status

8. ParkingLotMaintenance:

- ParkingLotMaintenanceId
- MaintenanceId
- RequestId
- EmployeeId

9. ParkingSpotHistory:

- ParkingSpotHistoryId
- ParkingSpaceId
- Status

10. Vehicle:

- VehicleId
- LicensePlate
- VehicleType

11. ParkingTicket:

- ParkingTicketId
- VehicleId
- ParkingLotId
- TicketStatus
- Reason
- IssueDate

12. Payment:

- PaymentId
- PaymentAmount
- ParkingTicketId
- PaymentDate
- PaymentMethod

13. User:

- UserId
- UserName
- Email
- Age
- BirthDay
- PassWord
- FirstName
- LastName

14. Reservation:

- ReservationId
- ParkingSpaceId
- UserId
- StartTime
- EndTime
- ReservationStatus
- Duration

15. UserPhoneNumber:

- UserPhoneNumberId
- UserId
- UserPhoneNumber

16. SpotHistoryDate:

- DateId
- ParkingSpotHistoryId
- StartTime
- EndTime

17. Supervise:

- SupervisorId
- EmployeeId
- SuperviseId

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

- Weak entities
- Recursive relationship

✓ Weak entities

There are two weak entities in our ER diagram shown, and we have shown them using a double-line rectangle. ParkingSpotHistory is one of them. It depends on the ParkingSpot entity. We used the spaceId and HistoryId as the keys for that entity. Another weak entity we have considered is the ParkingLotMaintenance. It depends on the ParkingLot and it does not have unique key. Therefore, we have considered two keys for identify that entity.

✓ Recursive relationship

We have created the one recursive relationship in our database. That is the scenario with the partnership between management and employees. As a result, a manager can only manage a limited number of employees, one of whom may also be the manager. So that's why this relationship was selected as a recursive relationship. In the UML Class Diagram, we have used another entity call supervise to represent the recursive relationship.

## **Chapter 2: Conceptual Design**

## 1. ER Diagram of the Project

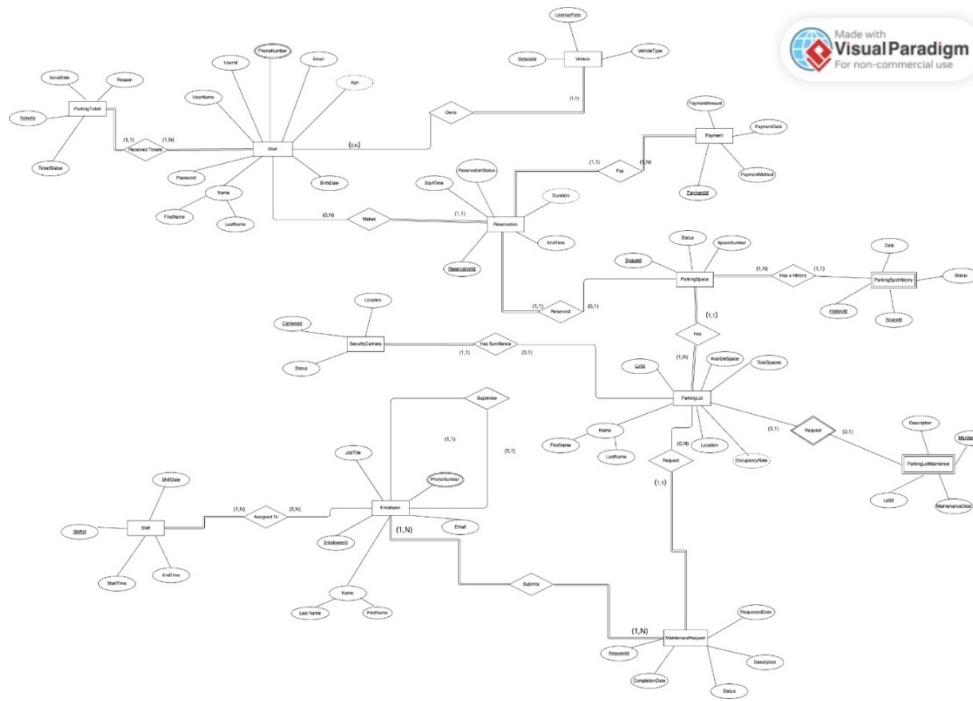


Figure 1: ER diagram of the Smart Parking Management System

## 2. UML Class Diagram of the Project

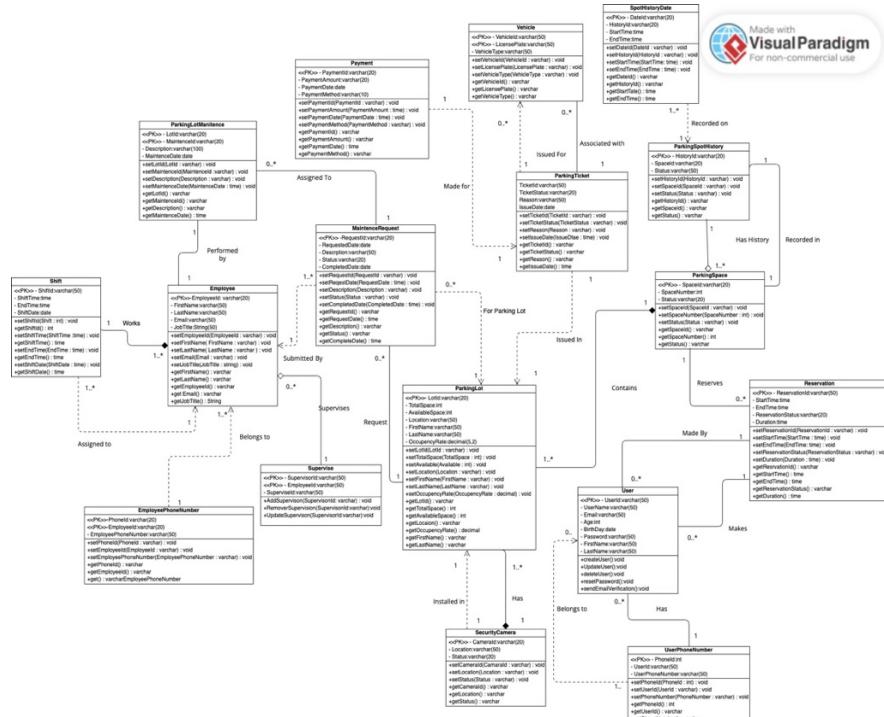


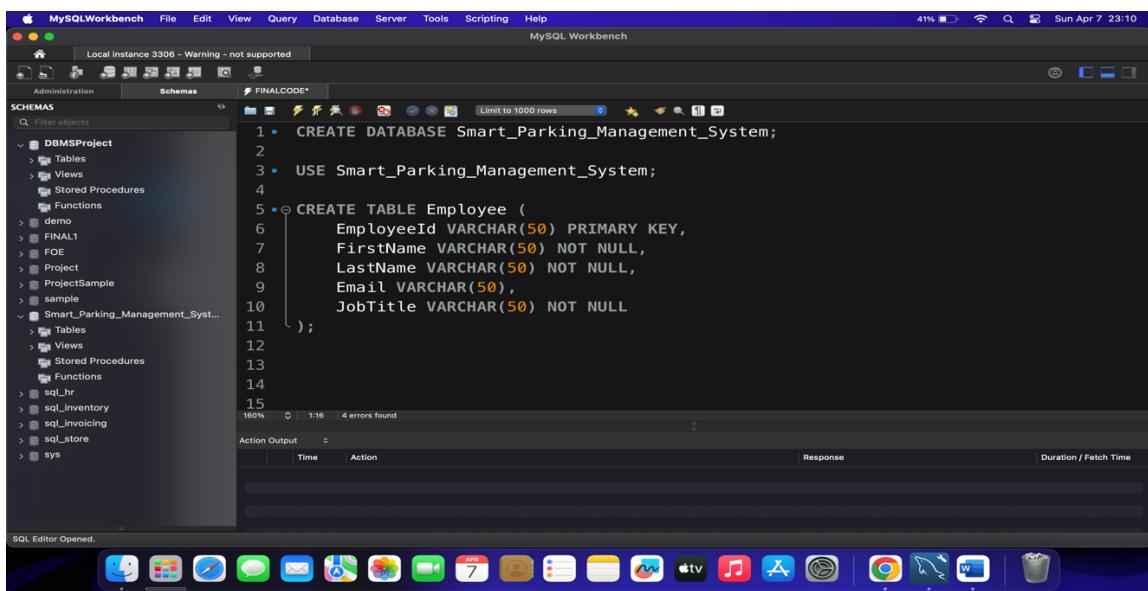
Figure 2: UML Class Diagram of the Smart Parking Management System

# Chapter 3: Implementation

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

## 3.1 Table Implementation

- ❖ Create table Employee.

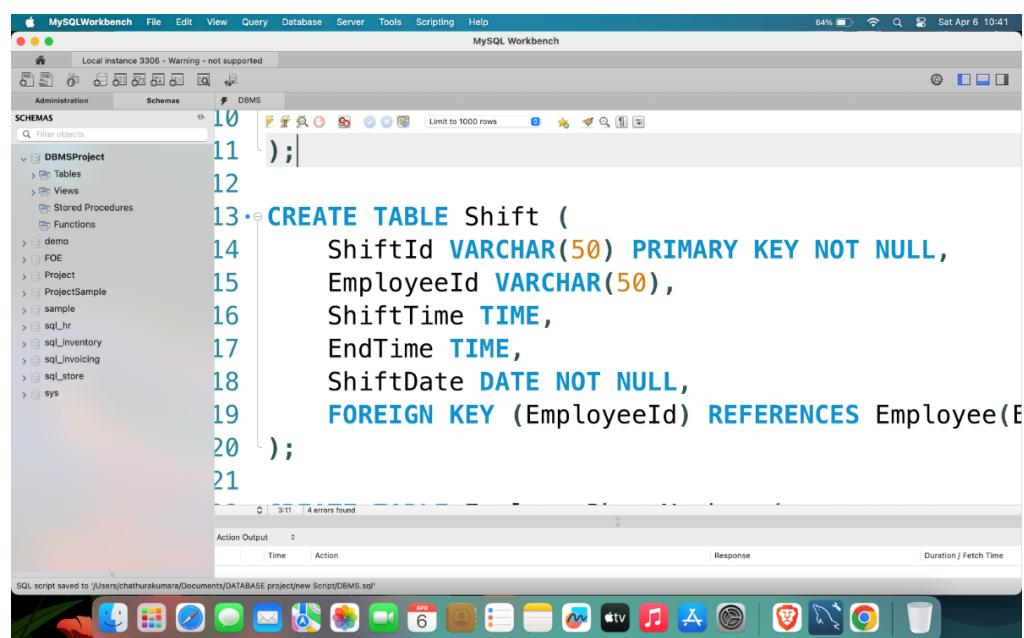


The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The title bar reads "MySQL Workbench". The main window has a dark theme. On the left, the "Schemas" tree view shows a database named "DBMSProject" containing tables, views, stored procedures, and functions. In the center, the SQL Editor tab is active, displaying the following SQL code:

```
1 • CREATE DATABASE Smart_Parking_Management_System;
2
3 • USE Smart_Parking_Management_System;
4
5 • CREATE TABLE Employee (
6     EmployeeId VARCHAR(50) PRIMARY KEY,
7     FirstName VARCHAR(50) NOT NULL,
8     LastName VARCHAR(50) NOT NULL,
9     Email VARCHAR(50),
10    JobTitle VARCHAR(50) NOT NULL
11 );
12
13
14
15
16
17
18
19
20
21
```

The status bar at the bottom indicates "Action Output" and shows "Time Action Response Duration / Fetch Time". The Mac OS X dock is visible at the bottom of the screen.

- ❖ Create table Shift.

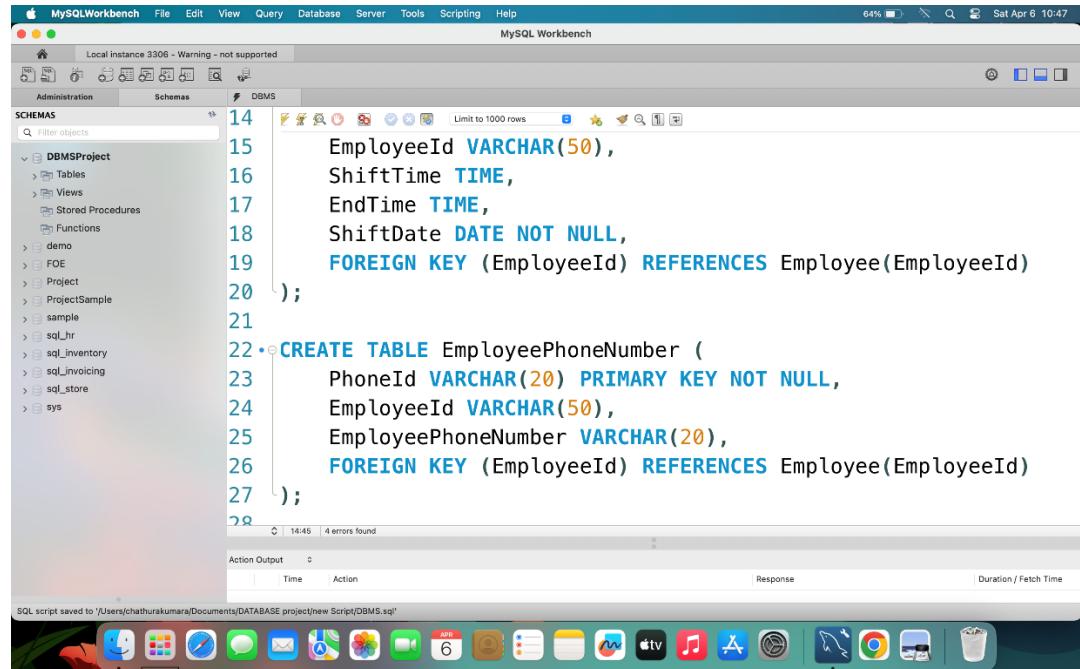


The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The title bar reads "MySQL Workbench". The main window has a dark theme. On the left, the "Schemas" tree view shows a database named "DBMSProject" containing tables, views, stored procedures, and functions. In the center, the SQL Editor tab is active, displaying the following SQL code:

```
10
11 );
12
13 • CREATE TABLE Shift (
14     ShiftId VARCHAR(50) PRIMARY KEY NOT NULL,
15     EmployeeId VARCHAR(50),
16     ShiftTime TIME,
17     EndTime TIME,
18     ShiftDate DATE NOT NULL,
19     FOREIGN KEY (EmployeeId) REFERENCES Employee(E
20 );
21
```

The status bar at the bottom indicates "Action Output" and shows "Time Action Response Duration / Fetch Time". The Mac OS X dock is visible at the bottom of the screen.

❖ Create table Employee Phone Number

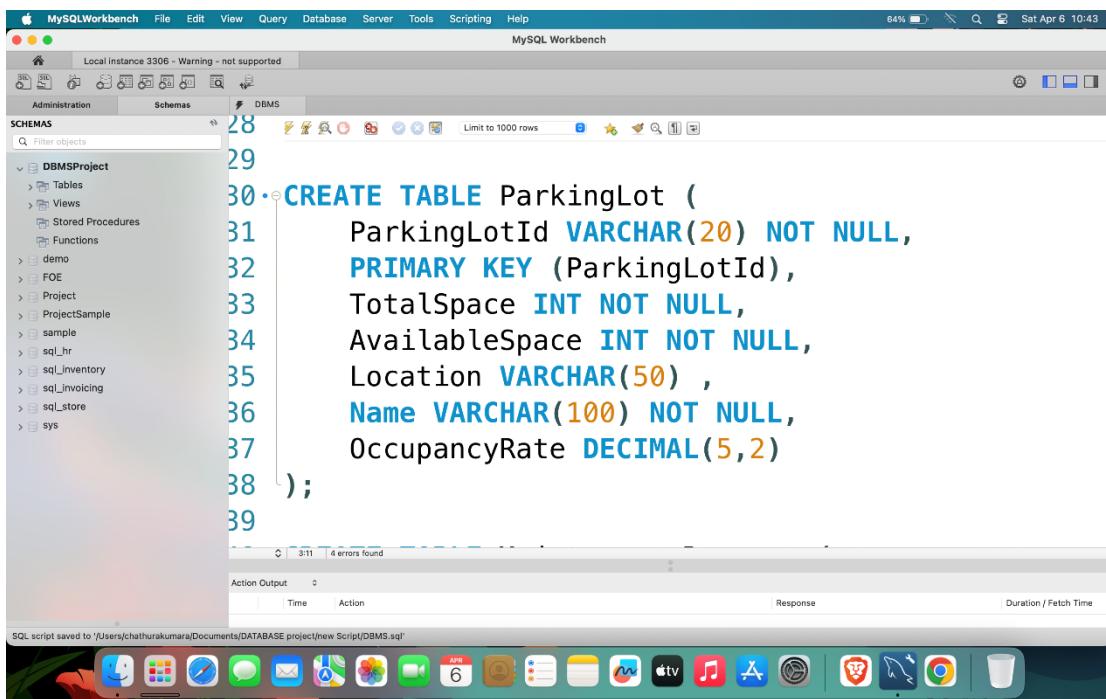


The screenshot shows the MySQL Workbench interface with the following SQL script:

```
14 EmployeeId VARCHAR(50),
15 ShiftTime TIME,
16 EndTime TIME,
17 ShiftDate DATE NOT NULL,
18 FOREIGN KEY (EmployeeId) REFERENCES Employee(EmployeeId)
19 );
20
21
22 CREATE TABLE EmployeePhoneNumber (
23 PhoneId VARCHAR(20) PRIMARY KEY NOT NULL,
24 EmployeeId VARCHAR(50),
25 EmployeePhoneNumber VARCHAR(20),
26 FOREIGN KEY (EmployeeId) REFERENCES Employee(EmployeeId)
27 );
```

The script creates two tables: `EmployeeShift` and `EmployeePhoneNumber`. The `EmployeeShift` table has columns for EmployeeId (VARCHAR(50)), ShiftTime (TIME), EndTime (TIME), and ShiftDate (DATE NOT NULL), with a foreign key reference to the `Employee` table. The `EmployeePhoneNumber` table has columns for PhoneId (VARCHAR(20) PRIMARY KEY NOT NULL), EmployeeId (VARCHAR(50)), and EmployeePhoneNumber (VARCHAR(20)), with a foreign key reference to the `Employee` table.

❖ Create table Parking Lot

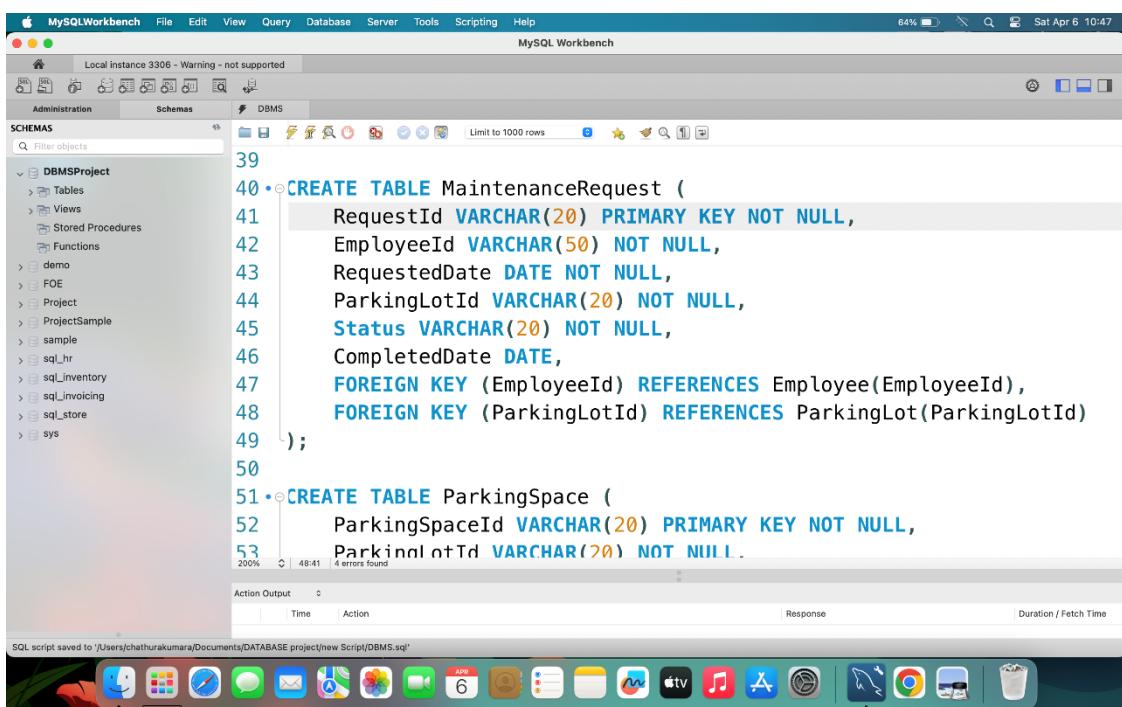


The screenshot shows the MySQL Workbench interface with the 'DBMS' tab selected. In the left sidebar, under the 'Schemas' section, the 'DBMSProject' schema is expanded, showing 'Tables', 'Views', 'Stored Procedures', 'Functions', and several other database objects. The main query editor window contains the following SQL code:

```
28
29
30 • CREATE TABLE ParkingLot (
31     ParkingLotId VARCHAR(20) NOT NULL,
32         PRIMARY KEY (ParkingLotId),
33     TotalSpace INT NOT NULL,
34     AvailableSpace INT NOT NULL,
35     Location VARCHAR(50) ,
36     Name VARCHAR(100) NOT NULL,
37     OccupancyRate DECIMAL(5,2)
38 );
39
```

The status bar at the bottom indicates '4 errors found'. The Mac OS X dock is visible at the bottom of the screen.

❖ Create table Maintenance Request

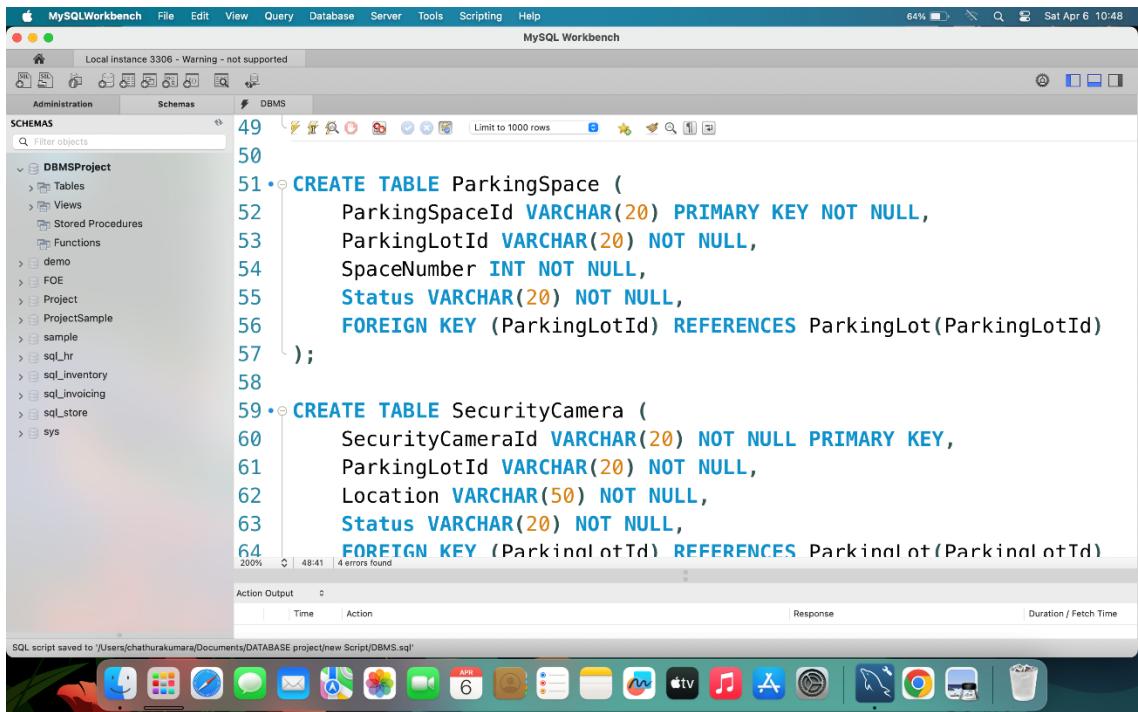


The screenshot shows the MySQL Workbench interface with the 'DBMS' tab selected. In the left sidebar, under the 'Schemas' section, the 'DBMSProject' schema is expanded, showing 'Tables', 'Views', 'Stored Procedures', 'Functions', and several other database objects. The main query editor window contains the following SQL code:

```
39
40 • CREATE TABLE MaintenanceRequest (
41     RequestId VARCHAR(20) PRIMARY KEY NOT NULL,
42     EmployeeId VARCHAR(50) NOT NULL,
43     RequestedDate DATE NOT NULL,
44     ParkingLotId VARCHAR(20) NOT NULL,
45     Status VARCHAR(20) NOT NULL,
46     CompletedDate DATE,
47     FOREIGN KEY (EmployeeId) REFERENCES Employee(EmployeeId),
48     FOREIGN KEY (ParkingLotId) REFERENCES ParkingLot(ParkingLotId)
49 );
50
51 • CREATE TABLE ParkingSpace (
52     ParkingSpaceId VARCHAR(20) PRIMARY KEY NOT NULL,
53     ParkingLotId VARCHAR(20) NOT NULL,
```

The status bar at the bottom indicates '4 errors found'. The Mac OS X dock is visible at the bottom of the screen.

## ❖ Create table Parking Space

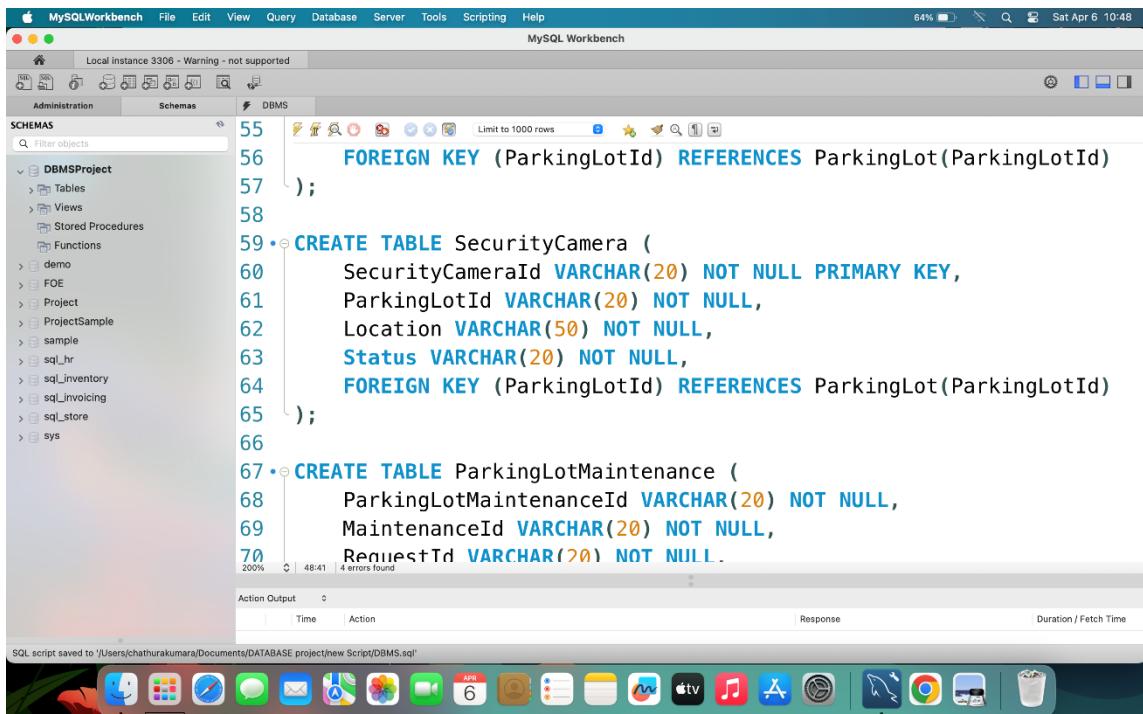


The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The title bar reads "MySQL Workbench" and "Local instance 3306 - Warning - not supported". The main window displays SQL code for creating the "ParkingSpace" table:

```
49
50
51 • CREATE TABLE ParkingSpace (
52     ParkingSpaceId VARCHAR(20) PRIMARY KEY NOT NULL,
53     ParkingLotId VARCHAR(20) NOT NULL,
54     SpaceNumber INT NOT NULL,
55     Status VARCHAR(20) NOT NULL,
56     FOREIGN KEY (ParkingLotId) REFERENCES ParkingLot(ParkingLotId)
57 );
58
59 • CREATE TABLE SecurityCamera (
60     SecurityCameraId VARCHAR(20) NOT NULL PRIMARY KEY,
61     ParkingLotId VARCHAR(20) NOT NULL,
62     Location VARCHAR(50) NOT NULL,
63     Status VARCHAR(20) NOT NULL,
64     FOREIGN KEY (ParkingLotId) REFERENCES ParkingLot(ParkingLotId)
```

The code is saved to a file named "DBMS.sql" at the bottom of the editor.

## ❖ Create table Security Camera

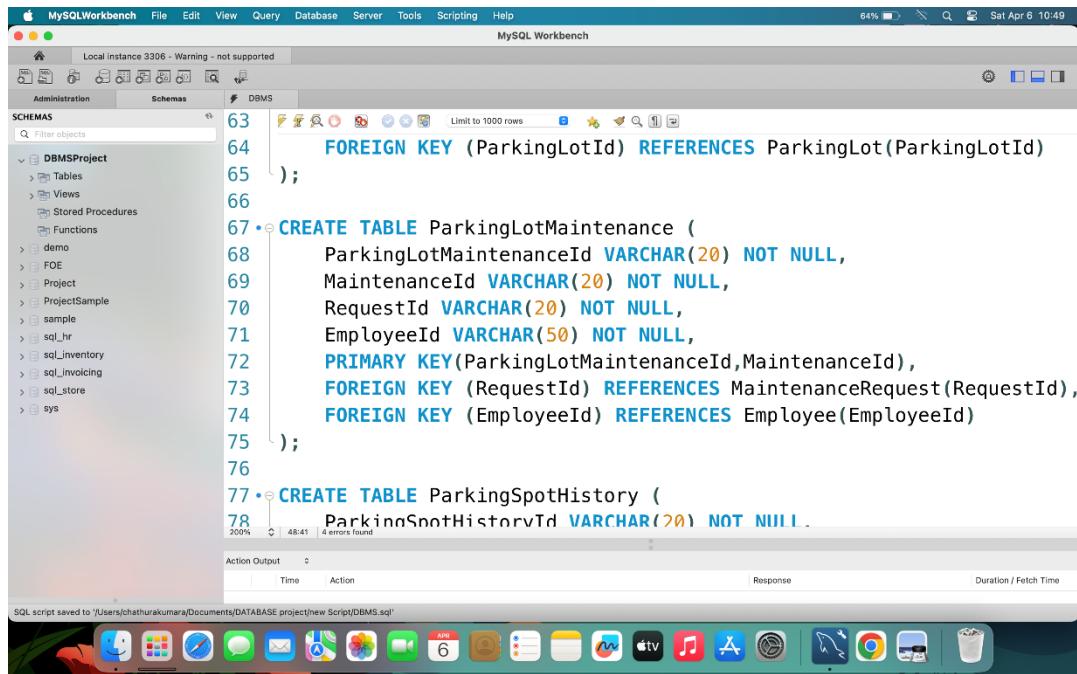


The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The title bar reads "MySQL Workbench" and "Local instance 3306 - Warning - not supported". The main window displays SQL code for creating the "SecurityCamera" and "ParkingLotMaintenance" tables:

```
55
56     FOREIGN KEY (ParkingLotId) REFERENCES ParkingLot(ParkingLotId)
57 );
58
59 • CREATE TABLE SecurityCamera (
60     SecurityCameraId VARCHAR(20) NOT NULL PRIMARY KEY,
61     ParkingLotId VARCHAR(20) NOT NULL,
62     Location VARCHAR(50) NOT NULL,
63     Status VARCHAR(20) NOT NULL,
64     FOREIGN KEY (ParkingLotId) REFERENCES ParkingLot(ParkingLotId)
65 );
66
67 • CREATE TABLE ParkingLotMaintenance (
68     ParkingLotMaintenanceId VARCHAR(20) NOT NULL,
69     MaintenanceId VARCHAR(20) NOT NULL,
70     RequestId VARCHAR(20) NOT NULL,
```

The code is saved to a file named "DBMS.sql" at the bottom of the editor.

## ❖ Create table Parking Lot Maintenance

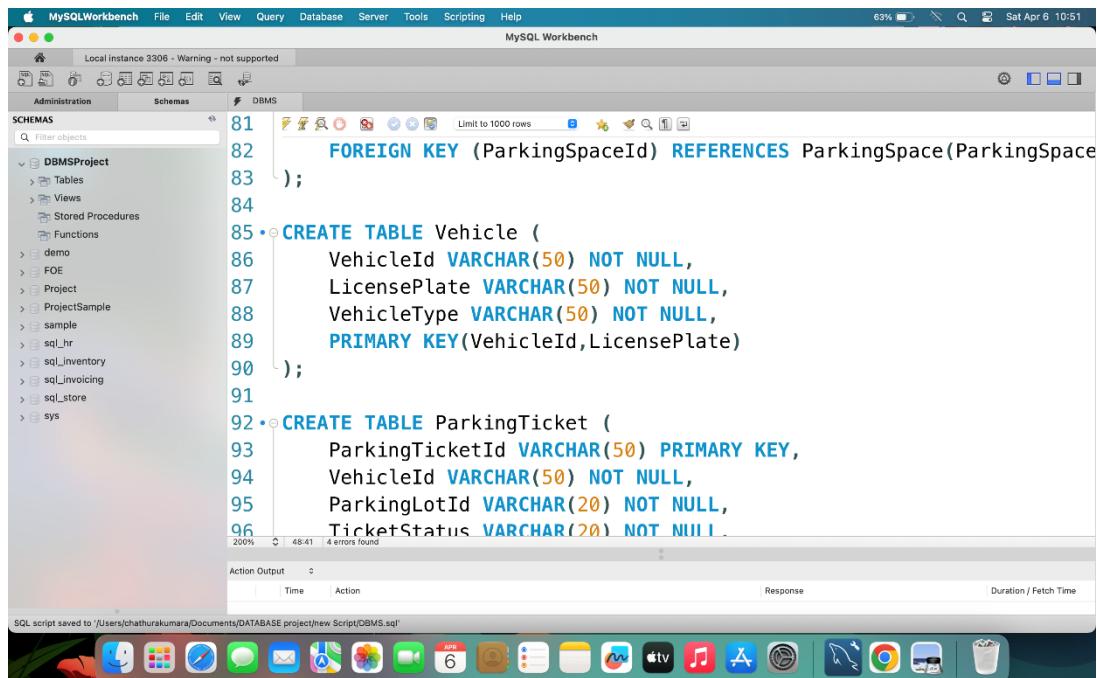


The screenshot shows the MySQL Workbench interface with the DBMS tab selected. In the center pane, SQL code is being typed into the editor:

```
63 FOREIGN KEY (ParkingLotId) REFERENCES ParkingLot(ParkingLotId)
64 );
65
66 CREATE TABLE ParkingLotMaintenance (
67     ParkingLotMaintenanceId VARCHAR(20) NOT NULL,
68     MaintenanceId VARCHAR(20) NOT NULL,
69     RequestId VARCHAR(20) NOT NULL,
70     EmployeeId VARCHAR(50) NOT NULL,
71     PRIMARY KEY(ParkingLotMaintenanceId,MaintenanceId),
72     FOREIGN KEY (RequestId) REFERENCES MaintenanceRequest(RequestId),
73     FOREIGN KEY (EmployeeId) REFERENCES Employee(EmployeeId)
74 );
75
76
77 CREATE TABLE ParkingSpotHistory (
78     ParkingSpotHistoryId VARCHAR(20) NOT NULL
79 );
```

The status bar at the bottom indicates "SQL script saved to '/Users/chathurakumara/Documents/DATABASE project/new Script/DBMS.sql'".

## ❖ Create Vehicle

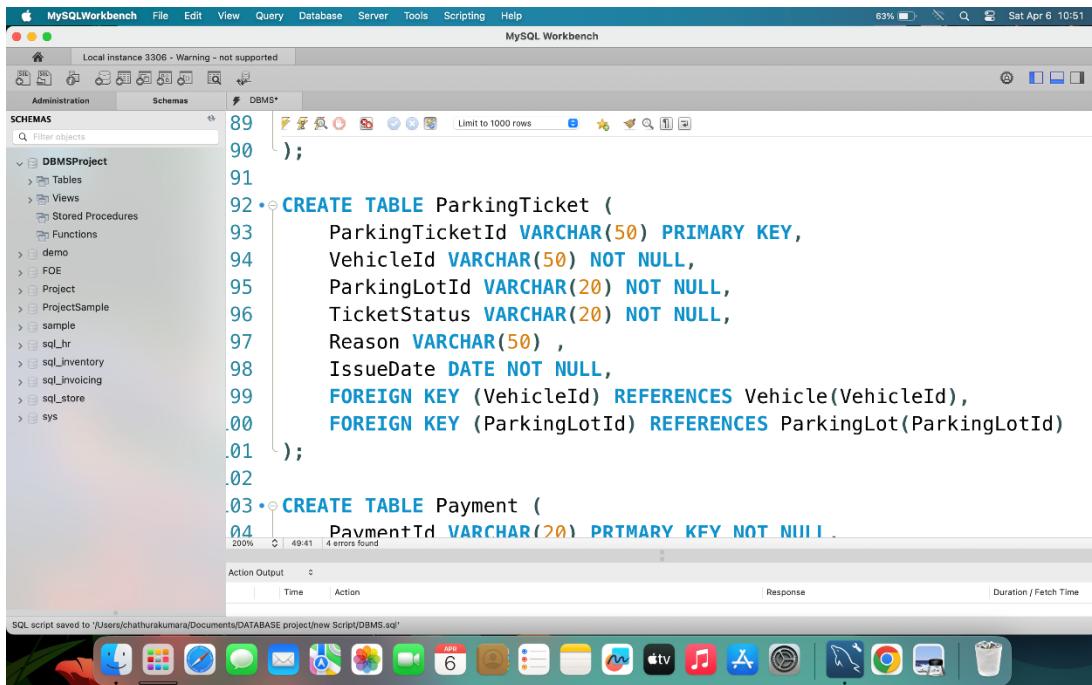


The screenshot shows the MySQL Workbench interface with the DBMS tab selected. In the center pane, SQL code is being typed into the editor:

```
81 FOREIGN KEY (ParkingSpaceId) REFERENCES ParkingSpace(ParkingSpace
82 );
83
84 CREATE TABLE Vehicle (
85     VehicleId VARCHAR(50) NOT NULL,
86     LicensePlate VARCHAR(50) NOT NULL,
87     VehicleType VARCHAR(50) NOT NULL,
88     PRIMARY KEY(VehicleId,LicensePlate)
89 );
90
91
92 CREATE TABLE ParkingTicket (
93     ParkingTicketId VARCHAR(50) PRIMARY KEY,
94     VehicleId VARCHAR(50) NOT NULL,
95     ParkingLotId VARCHAR(20) NOT NULL,
96     TicketStatus VARCHAR(20) NOT NULL
97 );
```

The status bar at the bottom indicates "SQL script saved to '/Users/chathurakumara/Documents/DATABASE project/new Script/DBMS.sql'".

❖ Create table ParkingTicket.

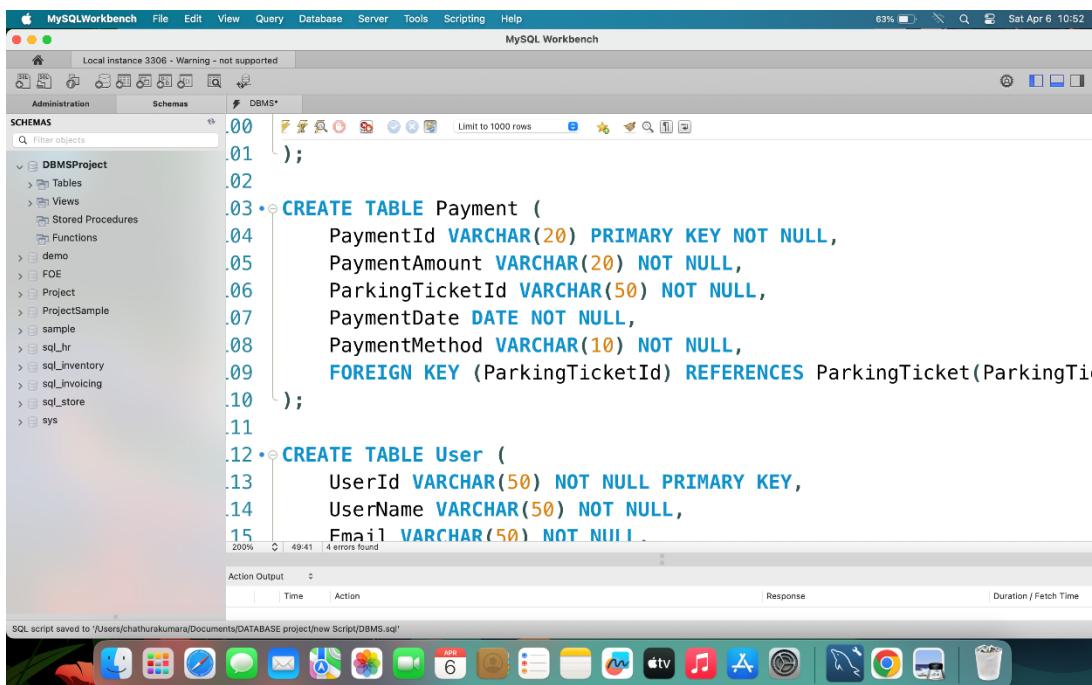


The screenshot shows the MySQL Workbench interface with the SQL editor tab active. The code being run is:

```
89 );
90 );
91
92 • CREATE TABLE ParkingTicket (
93     ParkingTicketId VARCHAR(50) PRIMARY KEY,
94     VehicleId VARCHAR(50) NOT NULL,
95     ParkingLotId VARCHAR(20) NOT NULL,
96     TicketStatus VARCHAR(20) NOT NULL,
97     Reason VARCHAR(50) ,
98     IssueDate DATE NOT NULL,
99     FOREIGN KEY (VehicleId) REFERENCES Vehicle(VehicleId),
100    FOREIGN KEY (ParkingLotId) REFERENCES ParkingLot(ParkingLotId)
101 );
102
103 • CREATE TABLE Payment (
104     PaymentId VARCHAR(20) PRIMARY KEY NOT NULL
105 );
```

The status bar at the bottom indicates "4 errors found".

❖ Create table Payment.

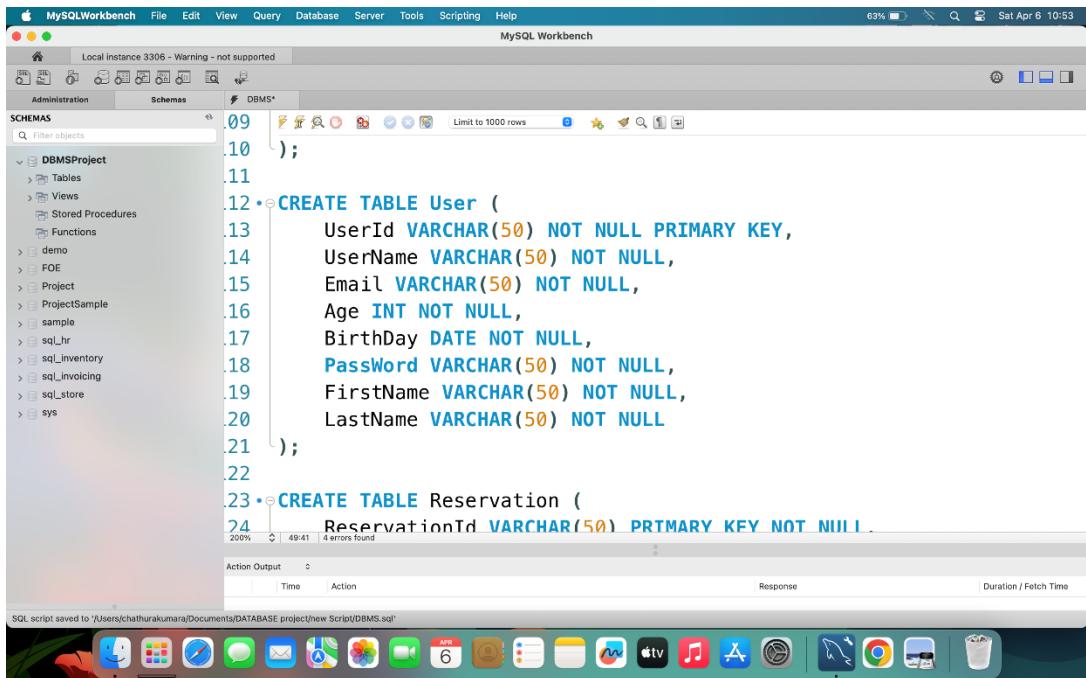


The screenshot shows the MySQL Workbench interface with the SQL editor tab active. The code being run is:

```
00 );
01 );
02
03 • CREATE TABLE Payment (
04     PaymentId VARCHAR(20) PRIMARY KEY NOT NULL,
05     PaymentAmount VARCHAR(20) NOT NULL,
06     ParkingTicketId VARCHAR(50) NOT NULL,
07     PaymentDate DATE NOT NULL,
08     PaymentMethod VARCHAR(10) NOT NULL,
09     FOREIGN KEY (ParkingTicketId) REFERENCES ParkingTicket(ParkingTicketId)
10 );
11
12 • CREATE TABLE User (
13     UserId VARCHAR(50) NOT NULL PRIMARY KEY,
14     UserName VARCHAR(50) NOT NULL,
15     Email VARCHAR(50) NOT NULL
16 );
```

The status bar at the bottom indicates "4 errors found".

❖ Create table User.

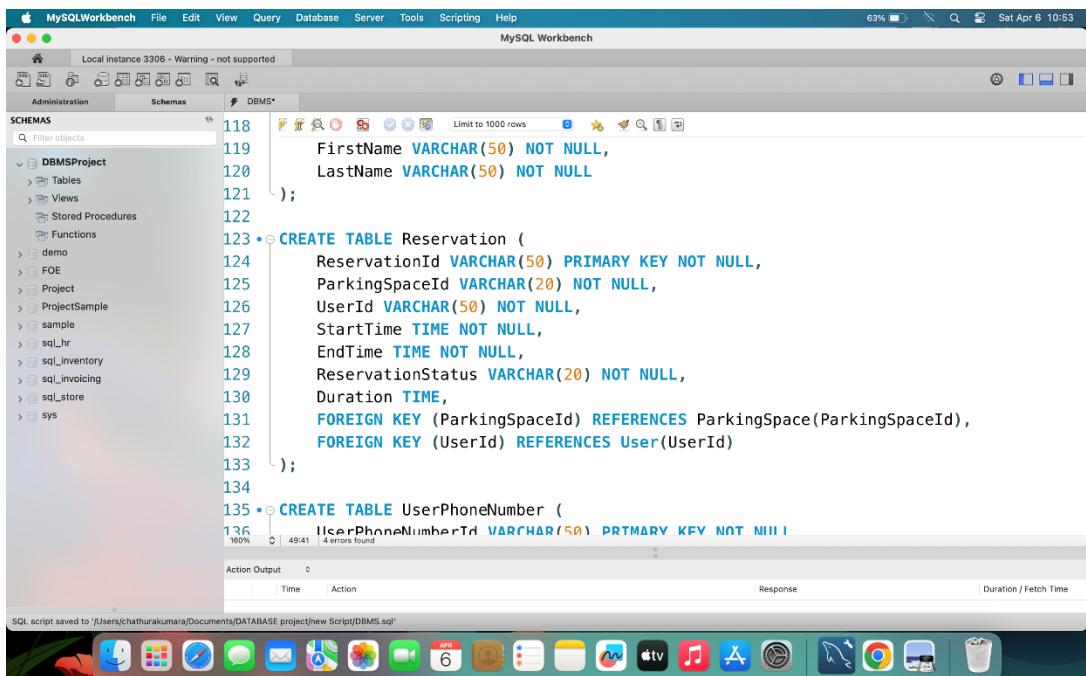


The screenshot shows the MySQL Workbench interface with the 'DBMS\*' tab selected. In the left sidebar, under the 'SCHEMAS' section, there is a tree view of database objects. The 'Tables' node under 'DBMSProject' is expanded, showing several table names. The main pane displays the SQL code for creating the 'User' table:

```
.09
.10 );
.11
.12 • CREATE TABLE User (
.13     UserId VARCHAR(50) NOT NULL PRIMARY KEY,
.14     UserName VARCHAR(50) NOT NULL,
.15     Email VARCHAR(50) NOT NULL,
.16     Age INT NOT NULL,
.17     BirthDay DATE NOT NULL,
.18     PassWord VARCHAR(50) NOT NULL,
.19     FirstName VARCHAR(50) NOT NULL,
.20     LastName VARCHAR(50) NOT NULL
.21 );
.22
.23 • CREATE TABLE Reservation (
.24     ReservationId VARCHAR(50) PRIMARY KEY NOT NULL,
```

The status bar at the bottom indicates that the script was saved to 'J:/Users/chathurakunara/Documents/DATABASE project/new Script/DBMS.sql'.

❖ Create table Reservation.



The screenshot shows the MySQL Workbench interface with the 'DBMS\*' tab selected. In the left sidebar, under the 'SCHEMAS' section, there is a tree view of database objects. The 'Tables' node under 'DBMSProject' is expanded, showing several table names. The main pane displays the SQL code for creating the 'Reservation' table:

```
118
119     FirstName VARCHAR(50) NOT NULL,
120     LastName VARCHAR(50) NOT NULL
121 );
122
123 • CREATE TABLE Reservation (
124     ReservationId VARCHAR(50) PRIMARY KEY NOT NULL,
125     ParkingSpaceId VARCHAR(20) NOT NULL,
126     UserId VARCHAR(50) NOT NULL,
127     StartTime TIME NOT NULL,
128     EndTime TIME NOT NULL,
129     ReservationStatus VARCHAR(20) NOT NULL,
130     Duration TIME,
131     FOREIGN KEY (ParkingSpaceId) REFERENCES ParkingSpace(ParkingSpaceId),
132     FOREIGN KEY (UserId) REFERENCES User(UserId)
133 );
134
135 • CREATE TABLE UserPhoneNumber (
136     UserPhoneNumberId VARCHAR(50) PRIMARY KEY NOT NULL,
```

The status bar at the bottom indicates that the script was saved to 'J:/Users/chathurakunara/Documents/DATABASE project/new Script/DBMS.sql'.

#### ❖ Create table UserPhoneNumber.

MySQLWorkbench

File Edit View Query Database Server Tools Scripting Help

Local instance 3306 - Warning - not supported MySQL Workbench

Administration Schemas DBMS\*

Schemas Filter objects .27

DBMSProject .28

Tables .29

Views .30

Stored Procedures .31

Functions .32

demo .33

FOE .34

Project .35

ProjectSample .36

sample .37

sql\_hr .38

sql\_inventory .39

sql\_invoicing .40

sql\_store .41

sys .42

EndTime **TIME NOT NULL**,  
ReservationStatus **VARCHAR(20) NOT NULL**,  
Duration **TIME**,  
**FOREIGN KEY** (ParkingSpaceId) **REFERENCES** ParkingSpace(ParkingSpace  
**FOREIGN KEY** (UserId) **REFERENCES** User(UserId)  
);

CREATE TABLE UserPhoneNumber (

UserPhoneNumberId **VARCHAR(50) PRIMARY KEY NOT NULL**,  
UserId **VARCHAR(50) NOT NULL**,  
UserPhoneNumber **VARCHAR(20) NOT NULL**,  
**FOREIGN KEY** (UserId) **REFERENCES** User(UserId)  
);

CREATE TABLE SnoHistoryDate (

Action Output

Time Action Response Duration / Fetch Time

SQL script saved to '/Users/chathurakumara/Documents/DATABASE project/new Script/DBMS.sql'

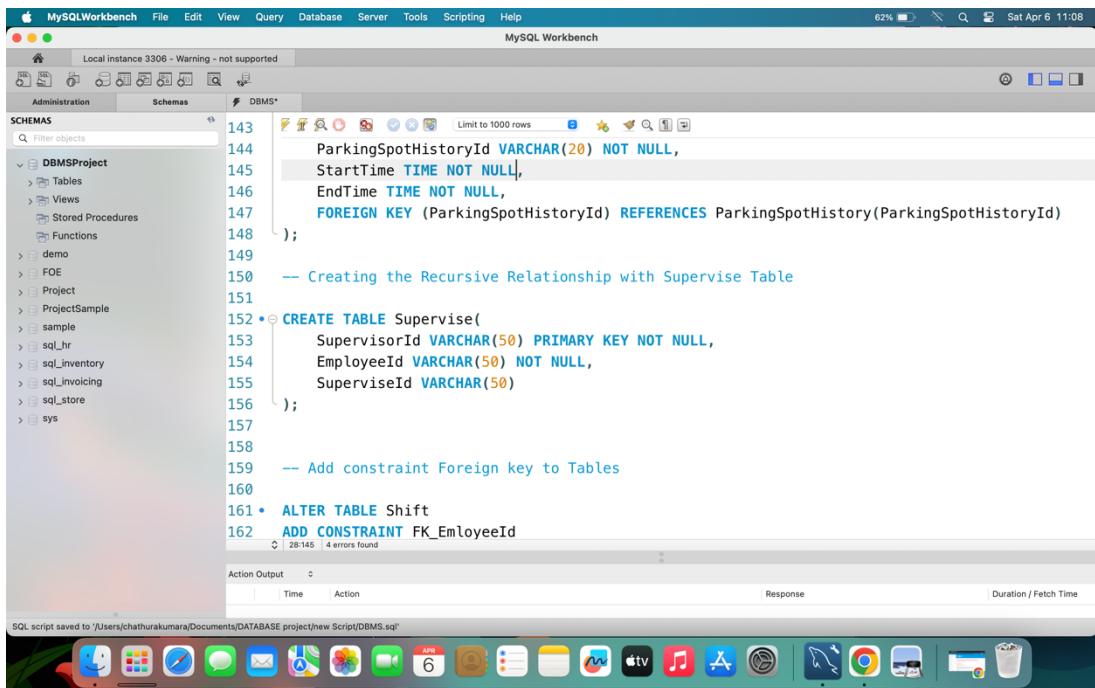
#### ❖ Create table SpotHistoryDate.

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

- Top Bar:** MySQLWorkbench, File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Title Bar:** Local Instance 3306 - Warning - not supported, MySQL Workbench.
- Schemas Tab:** Shows the current schema tree, including DBMSProject, demo, FOE, Project, ProjectSample, sample, sql\_hr, sql\_inventory, sql\_invoicing, sql\_store, and sys.
- SQL Editor:** Displays the following SQL code:

```
135 • CREATE TABLE UserPhoneNumbers (
136     UserId VARCHAR(50) PRIMARY KEY NOT NULL,
137     UserPhoneNumbersId VARCHAR(20) NOT NULL,
138     FOREIGN KEY (UserId) REFERENCES User(UserId)
139 );
140
141
142 • CREATE TABLE SpotHistoryDate (
143     DateId VARCHAR(20) PRIMARY KEY NOT NULL,
144     ParkingSpotHistoryId VARCHAR(20) NOT NULL,
145     StartTime TIME NOT NULL,
146     EndTime TIME NOT NULL,
147     FOREIGN KEY (ParkingSpotHistoryId) REFERENCES ParkingSpotHistory(ParkingSpotHistoryId)
148 );
149
150 -- Creating the Recursive Relationship with Supervise Table
151
152 • CREATE TABLE Supervise(
153     SupervisorId VARCHAR(50) PRIMARY KEY NOT NULL,
154     EmployeeId VARCHAR(50) NOT NULL,
```
- Action Output:** Shows the status "28:145 - 4 errors found".
- Bottom Status Bar:** SQL script saved to '/Users/chathurakumara/Documents/DATABASE project/New Script/DBMS.sql'.

❖ Create table Supervise.

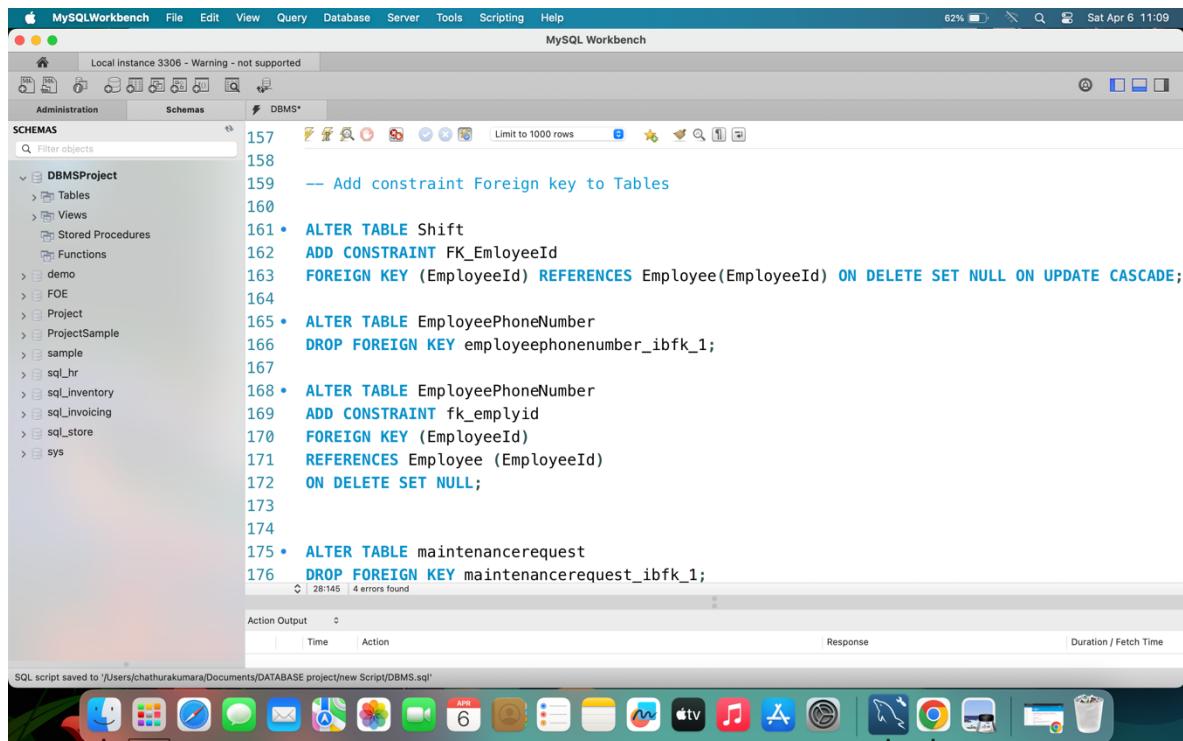


The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The main window displays an SQL script for creating a table named 'Supervise'. The script includes constraints for primary keys and foreign keys, as well as comments explaining the recursive relationship. The code is numbered from 143 to 162. The status bar at the bottom indicates the script was saved to a local file and shows the current date and time (Sat Apr 6 11:08).

```
143
144
145
146
147
148
149
150
151
152 • CREATE TABLE Supervise(
153     SupervisorId VARCHAR(50) PRIMARY KEY NOT NULL,
154     EmployeeId VARCHAR(50) NOT NULL,
155     SuperviseId VARCHAR(50)
156 );
157
158
159 -- Add constraint Foreign key to Tables
160
161 • ALTER TABLE Shift
162     ADD CONSTRAINT FK_EmployeeId
```

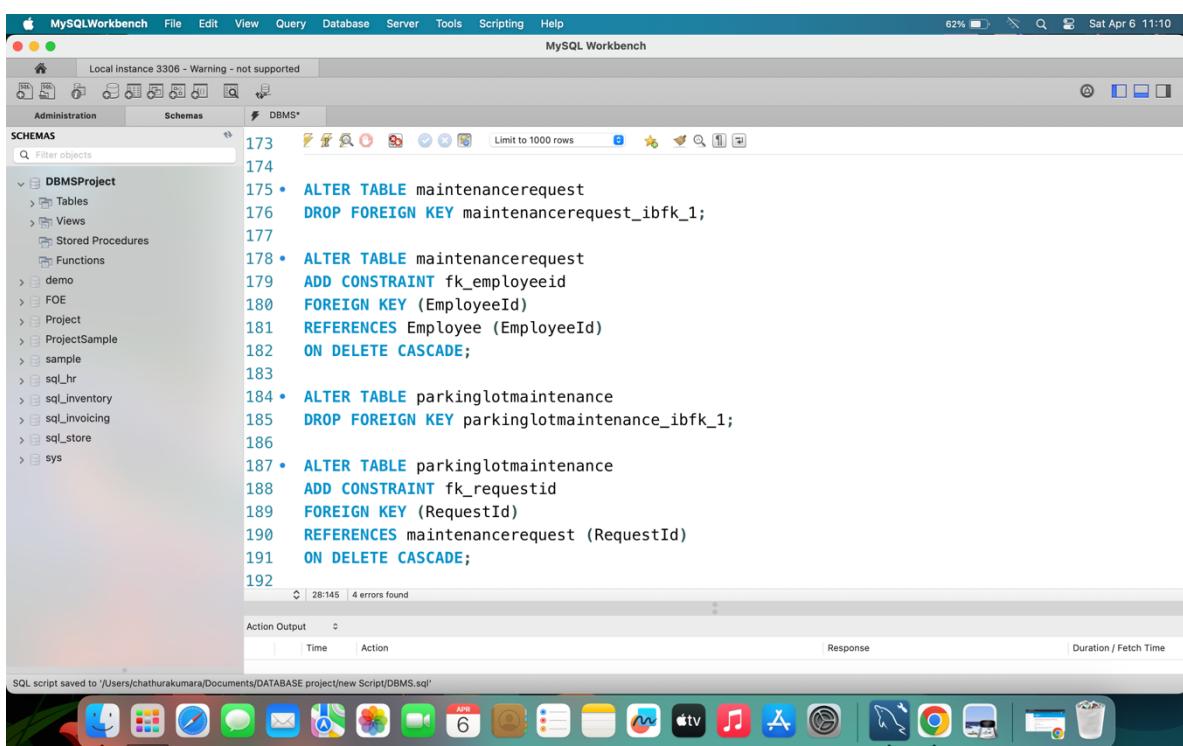
### 3.2 Creating Foreign Key Constraint

We have also inserted some of the foreign keys when implementing the table.



The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'Schemas' tree, which includes the 'DBMSProject' schema containing tables like 'Shift', 'EmployeePhoneNumber', and 'maintenancerequest'. The main pane shows a SQL script with numbered lines. Lines 157 through 176 are visible, detailing the creation of a foreign key constraint named 'fk\_employeeid' on the 'EmployeeId' column of the 'maintenancerequest' table, which references the 'EmployeeId' column in the 'Employee' table. The status bar at the bottom indicates the script was saved to '/Users/chathurakumara/Documents/DATABASE project/new Script/DBMS.sql'.

```
157 -- Add constraint Foreign key to Tables
158
159 • ALTER TABLE Shift
160     ADD CONSTRAINT FK_EmployeeId
161     FOREIGN KEY (EmployeeId) REFERENCES Employee(EmployeeId) ON DELETE SET NULL ON UPDATE CASCADE;
162
163 • ALTER TABLE EmployeePhoneNumber
164     DROP FOREIGN KEY employeephonenumber_ibfk_1;
165
166 • ALTER TABLE EmployeePhoneNumber
167     ADD CONSTRAINT fk_employeeid
168     FOREIGN KEY (EmployeeId)
169     REFERENCES Employee (EmployeeId)
170     ON DELETE SET NULL;
171
172
173
174
175 • ALTER TABLE maintenancerequest
176     DROP FOREIGN KEY maintenancerequest_ibfk_1;
```



This screenshot shows the MySQL Workbench interface again. The left sidebar shows the 'DBMSProject' schema. The main pane displays a SQL script with lines 173 through 192. Lines 175, 176, 184, 185, 187, and 188 describe the creation of foreign key constraints. Line 175 creates 'fk\_employeeid' on the 'EmployeeId' column of 'maintenancerequest' referencing 'Employee'. Line 176 drops the previous foreign key on 'EmployeeId'. Line 184 creates 'fk\_requestid' on the 'RequestId' column of 'parkinglotmaintenance' referencing 'maintenancerequest'. Line 185 drops the previous foreign key on 'RequestId'. Line 187 creates 'fk\_userid' on the 'UserId' column of 'parkinglotmaintenance' referencing 'User'. Line 188 drops the previous foreign key on 'UserId'. The status bar at the bottom indicates the script was saved to '/Users/chathurakumara/Documents/DATABASE project/new Script/DBMS.sql'.

```
173
174
175 • ALTER TABLE maintenancerequest
176     DROP FOREIGN KEY maintenancerequest_ibfk_1;
177
178 • ALTER TABLE maintenancerequest
179     ADD CONSTRAINT fk_employeeid
180     FOREIGN KEY (EmployeeId)
181     REFERENCES Employee (EmployeeId)
182     ON DELETE CASCADE;
183
184 • ALTER TABLE parkinglotmaintenance
185     DROP FOREIGN KEY parkinglotmaintenance_ibfk_1;
186
187 • ALTER TABLE parkinglotmaintenance
188     ADD CONSTRAINT fk_userid
189     FOREIGN KEY (UserId)
190     REFERENCES User (UserId)
191     ON DELETE CASCADE;
192
```

MySQLWorkbench File Edit View Query Database Server Tools Scripting Help

Local instance 3306 - Warning - not supported

Administration Schemas DBMS\*

SCHEMAS Filter objects

DBMSProject

Tables Views Stored Procedures Functions

demo FOE Project ProjectSample sample sql\_hr sql\_inventory sql\_invoicing sql\_store sys

191 192  
193 • ALTER TABLE parkingticket  
194 ADD CONSTRAINT fk\_parkinglot\_id  
195 FOREIGN KEY (ParkingLotId)  
196 REFERENCES ParkingLot (ParkingLotId)  
197 ON DELETE CASCADE;  
198  
199 • ALTER TABLE ParkingTicket  
200 DROP FOREIGN KEY parkingticket\_ibfk\_1;  
201  
202 • ALTER TABLE ParkingTicket  
203 ADD CONSTRAINT fk\_vehicleid  
204 FOREIGN KEY (VehicleId)  
205 REFERENCES Vehicle(VehicleId)  
206 ON DELETE CASCADE;  
207  
208  
209 • ALTER TABLE payment  
210 ADD CONSTRAINT fk\_parkingticket\_id

SQL script saved to '/Users/chathurakumara/Documents/DATABASE project/new Script/DBMS.sql'

Action Output Time Action Response Duration / Fetch Time



MySQLWorkbench File Edit View Query Database Server Tools Scripting Help

Local instance 3306 - Warning - not supported

Administration Schemas DBMS\*

SCHEMAS Filter objects

DBMSProject

Tables Views Stored Procedures Functions

demo FINAL1 FINALCODE

Tables Views Stored Procedures Functions

FOE Project ProjectSample sample sql\_hr sql\_inventory sql\_invoicing sql\_store sys

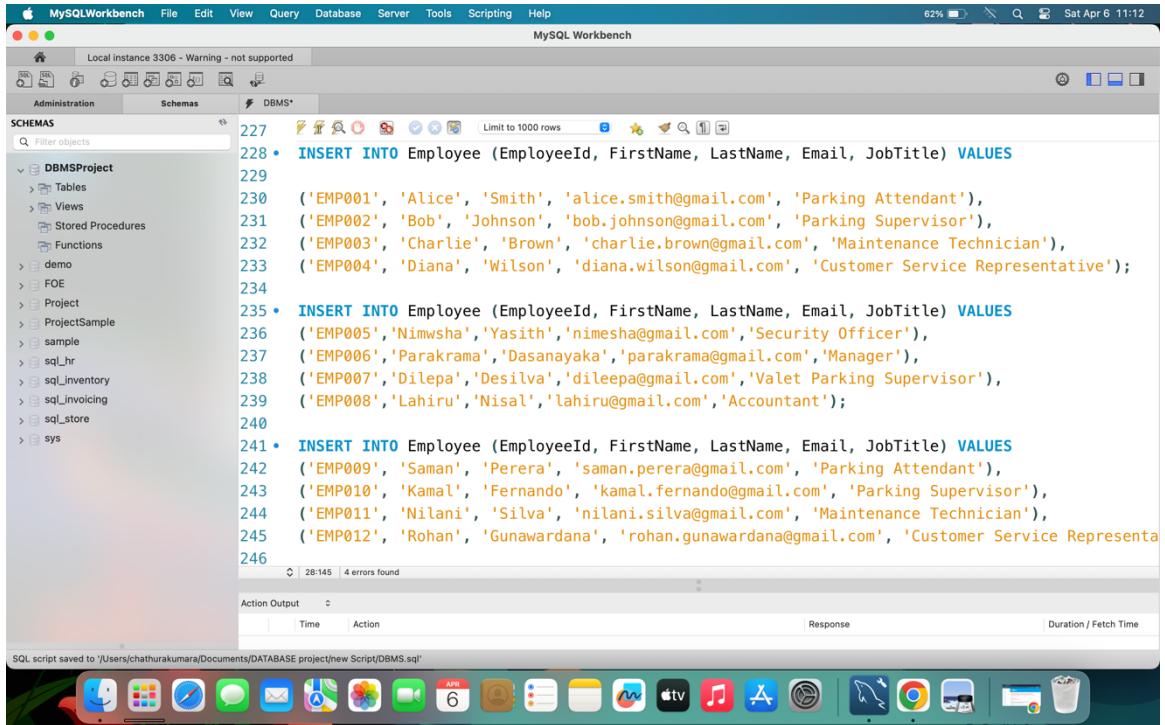
288 • ALTER TABLE ParkingTicket  
289 ADD CONSTRAINT fk\_vehicleid  
290 FOREIGN KEY (VehicleId)  
291 REFERENCES Vehicle(VehicleId)  
292 ON DELETE CASCADE;  
293  
294 — 6) payment  
295 • ALTER TABLE payment  
296 ADD CONSTRAINT fk\_parkingticket\_id  
297 FOREIGN KEY (ParkingTicketId)  
298 REFERENCES ParkingTicket (ParkingTicketId)  
299 ON DELETE CASCADE;  
300  
301 — Reservation  
302 • ALTER TABLE Reservation  
303 ADD CONSTRAINT fk\_parking\_space\_id  
304 FOREIGN KEY (ParkingSpaceId)  
305 REFERENCES ParkingSpace (ParkingSpaceId)  
306 ON DELETE CASCADE;  
307  
308 — 8) SpotHistoryDate  
309 • ALTER TABLE SpotHistoryDate  
310 DROP FOREIGN KEY spothistorydate\_ibfk\_1;  
311  
312 • ALTER TABLE SpotHistoryDate  
313 ADD CONSTRAINT fk\_parking\_spot\_history\_id  
314 FOREIGN KEY (ParkingSpotHistoryId)  
315 REFERENCES ParkingSpotHistory(ParkingSpotHistoryId)  
316 ON DELETE CASCADE;  
317  
318 — 9) UserPhoneNumber  
319 • ALTER TABLE UserPhoneNumber  
320 ADD CONSTRAINT fk\_user\_id  
321 FOREIGN KEY (Userid)  
322 REFERENCES User(UserId)  
323 ON DELETE CASCADE;

SQL script saved to '/Users/chathurakumara/Downloads/FINALCODE.sql'



### 3.3 Inserting Data for Tables

#### ❖ Employee Table

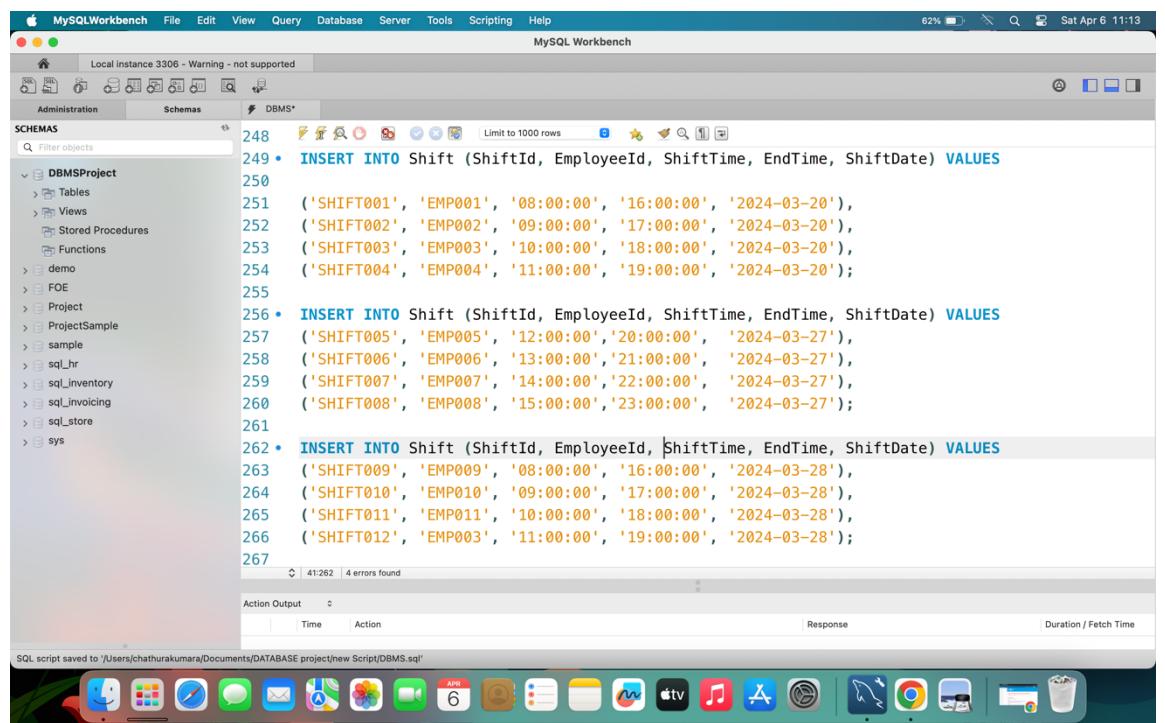


The screenshot shows the MySQL Workbench interface with the 'DBMS\*' tab selected. The left sidebar displays the 'Schemas' tree, which includes the 'DBMSProject' schema containing tables, views, stored procedures, functions, and demo data. The main pane shows a SQL script for inserting data into the 'Employee' table. The script contains 246 numbered lines of SQL code, starting with:

```
227 • INSERT INTO Employee (EmployeeId, FirstName, LastName, Email, JobTitle) VALUES
228 ('EMP001', 'Alice', 'Smith', 'alice.smith@gmail.com', 'Parking Attendant'),
229 ('EMP002', 'Bob', 'Johnson', 'bob.johnson@gmail.com', 'Parking Supervisor'),
230 ('EMP003', 'Charlie', 'Brown', 'charlie.brown@gmail.com', 'Maintenance Technician'),
231 ('EMP004', 'Diana', 'Wilson', 'diana.wilson@gmail.com', 'Customer Service Representative');
232
233
234
235 • INSERT INTO Employee (EmployeeId, FirstName, LastName, Email, JobTitle) VALUES
236 ('EMP005', 'Nimwscha', 'Yasith', 'nimeshsha@gmail.com', 'Security Officer'),
237 ('EMP006', 'Parakrama', 'Dasanayaka', 'parakrama@gmail.com', 'Manager'),
238 ('EMP007', 'Dilepa', 'Desilva', 'dileepa@gmail.com', 'Valet Parking Supervisor'),
239 ('EMP008', 'Lahiru', 'Nisal', 'lahiru@gmail.com', 'Accountant');
240
241 • INSERT INTO Employee (EmployeeId, FirstName, LastName, Email, JobTitle) VALUES
242 ('EMP009', 'Saman', 'Perera', 'saman.perera@gmail.com', 'Parking Attendant'),
243 ('EMP010', 'Kamal', 'Fernando', 'kamal.fernando@gmail.com', 'Parking Supervisor'),
244 ('EMP011', 'Nilani', 'Silva', 'nilani.silva@gmail.com', 'Maintenance Technician'),
245 ('EMP012', 'Rohan', 'Gunawardana', 'rohan.gunawardana@gmail.com', 'Customer Service Representa
246
```

The status bar at the bottom indicates '28:145 | 4 errors found'. The bottom navigation bar shows tabs for 'Action Output', 'Time', 'Action', 'Response', and 'Duration / Fetch Time'.

#### ❖ Shift

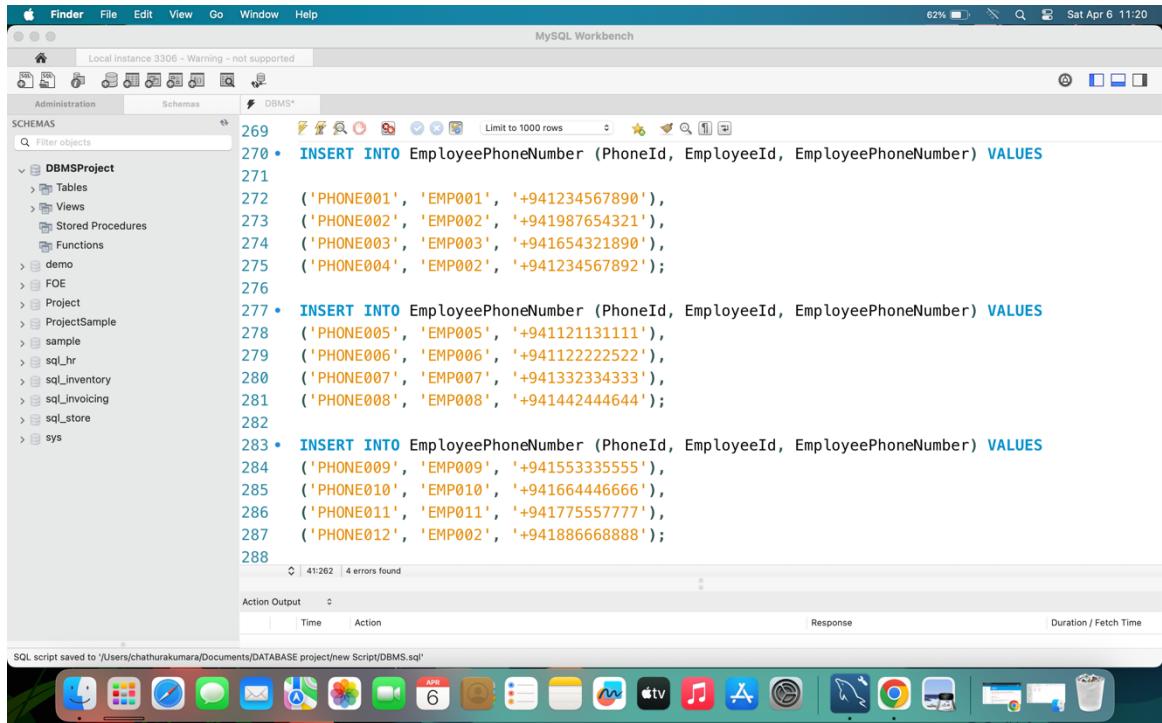


The screenshot shows the MySQL Workbench interface with the 'DBMS\*' tab selected. The left sidebar displays the 'Schemas' tree, which includes the 'DBMSProject' schema containing tables, views, stored procedures, functions, and demo data. The main pane shows a SQL script for inserting data into the 'Shift' table. The script contains 267 numbered lines of SQL code, starting with:

```
248 • INSERT INTO Shift (ShiftId, EmployeeId, ShiftTime, EndTime, ShiftDate) VALUES
249 ('SHIFT001', 'EMP001', '08:00:00', '16:00:00', '2024-03-20'),
250 ('SHIFT002', 'EMP002', '09:00:00', '17:00:00', '2024-03-20'),
251 ('SHIFT003', 'EMP003', '10:00:00', '18:00:00', '2024-03-20'),
252 ('SHIFT004', 'EMP004', '11:00:00', '19:00:00', '2024-03-20');
253
254
255
256 • INSERT INTO Shift (ShiftId, EmployeeId, ShiftTime, EndTime, ShiftDate) VALUES
257 ('SHIFT005', 'EMP005', '12:00:00', '20:00:00', '2024-03-27'),
258 ('SHIFT006', 'EMP006', '13:00:00', '21:00:00', '2024-03-27'),
259 ('SHIFT007', 'EMP007', '14:00:00', '22:00:00', '2024-03-27'),
260 ('SHIFT008', 'EMP008', '15:00:00', '23:00:00', '2024-03-27');
261
262 • INSERT INTO Shift (ShiftId, EmployeeId, ShiftTime, EndTime, ShiftDate) VALUES
263 ('SHIFT009', 'EMP009', '08:00:00', '16:00:00', '2024-03-28'),
264 ('SHIFT010', 'EMP010', '09:00:00', '17:00:00', '2024-03-28'),
265 ('SHIFT011', 'EMP011', '10:00:00', '18:00:00', '2024-03-28'),
266 ('SHIFT012', 'EMP003', '11:00:00', '19:00:00', '2024-03-28');
```

The status bar at the bottom indicates '41:262 | 4 errors found'. The bottom navigation bar shows tabs for 'Action Output', 'Time', 'Action', 'Response', and 'Duration / Fetch Time'.

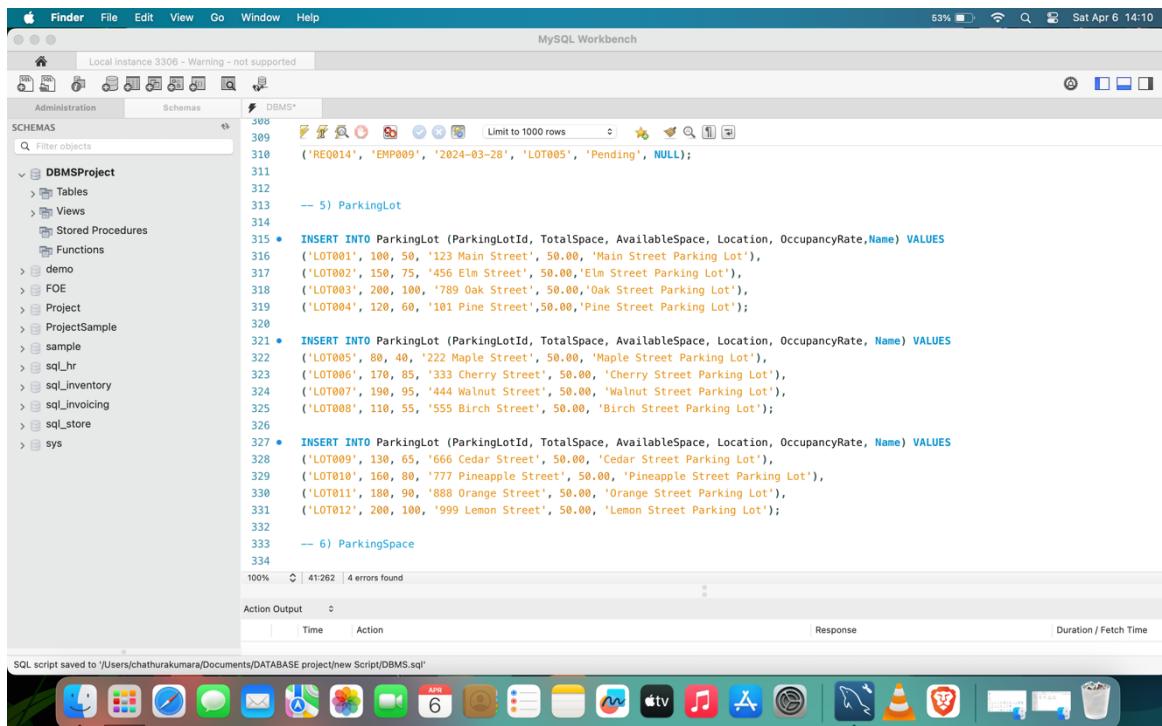
## ❖ Employee Phone Number



The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The title bar indicates "MySQL Workbench" and the date "Sat Apr 6 11:20". The main window displays an SQL script for inserting data into the "EmployeePhoneNumber" table. The script includes multiple INSERT INTO statements with values for columns like PhoneId, EmployeeId, and EmployeePhoneNumber. The code is color-coded for syntax highlighting. At the bottom of the script, it says "41:262 | 4 errors found". Below the script, there's a table titled "Action Output" with columns for Time, Action, Response, and Duration / Fetch Time. A status bar at the bottom shows various application icons.

```
269
270 • INSERT INTO EmployeePhoneNumber (PhoneId, EmployeeId, EmployeePhoneNumber) VALUES
271
272 ('PHONE001', 'EMP001', '+941234567890'),
273 ('PHONE002', 'EMP002', '+941987654321'),
274 ('PHONE003', 'EMP003', '+941654321890'),
275 ('PHONE004', 'EMP002', '+941234567892');
276
277 • INSERT INTO EmployeePhoneNumber (PhoneId, EmployeeId, EmployeePhoneNumber) VALUES
278 ('PHONE005', 'EMP005', '+941121131111'),
279 ('PHONE006', 'EMP006', '+941122222522'),
280 ('PHONE007', 'EMP007', '+941332334333'),
281 ('PHONE008', 'EMP008', '+941442444644');
282
283 • INSERT INTO EmployeePhoneNumber (PhoneId, EmployeeId, EmployeePhoneNumber) VALUES
284 ('PHONE009', 'EMP009', '+94155335555'),
285 ('PHONE010', 'EMP010', '+941664446666'),
286 ('PHONE011', 'EMP011', '+941775557777'),
287 ('PHONE012', 'EMP002', '+941886668888');
288
308
309 ('REQ014', 'EMP009', '2024-03-28', 'LOT005', 'Pending', NULL);
310
311
312
313 -- 5) ParkingLot
314
315 • INSERT INTO ParkingLot (ParkingLotId, TotalSpace, AvailableSpace, Location, OccupancyRate, Name) VALUES
316 ('LOT001', 100, 50, '123 Main Street', 50.00, 'Main Street Parking Lot'),
317 ('LOT002', 150, 75, '456 Elm Street', 50.00, 'Elm Street Parking Lot'),
318 ('LOT003', 200, 100, '789 Oak Street', 50.00, 'Oak Street Parking Lot'),
319 ('LOT004', 120, 60, '101 Pine Street', 50.00, 'Pine Street Parking Lot');
320
321 • INSERT INTO ParkingLot (ParkingLotId, TotalSpace, AvailableSpace, Location, OccupancyRate, Name) VALUES
322 ('LOT005', 80, 40, '222 Maple Street', 50.00, 'Maple Street Parking Lot'),
323 ('LOT006', 170, 85, '333 Cherry Street', 50.00, 'Cherry Street Parking Lot'),
324 ('LOT007', 190, 95, '444 Walnut Street', 50.00, 'Walnut Street Parking Lot'),
325 ('LOT008', 110, 55, '555 Birch Street', 50.00, 'Birch Street Parking Lot');
326
327 • INSERT INTO ParkingLot (ParkingLotId, TotalSpace, AvailableSpace, Location, OccupancyRate, Name) VALUES
328 ('LOT009', 130, 65, '666 Cedar Street', 50.00, 'Cedar Street Parking Lot'),
329 ('LOT010', 160, 80, '777 Pineapple Street', 50.00, 'Pineapple Street Parking Lot'),
330 ('LOT011', 180, 90, '888 Orange Street', 50.00, 'Orange Street Parking Lot'),
331 ('LOT012', 200, 100, '999 Lemon Street', 50.00, 'Lemon Street Parking Lot');
332
333 -- 6) ParkingSpace
334
```

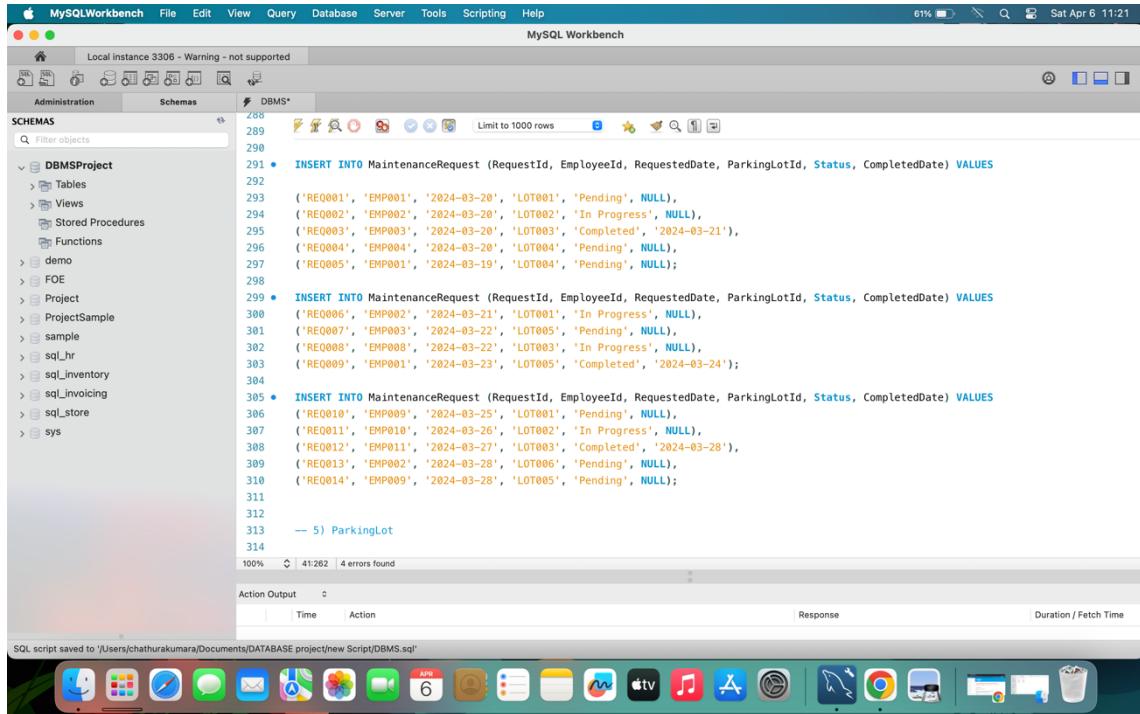
## ❖ Parking Lot



The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The title bar indicates "MySQL Workbench" and the date "Sat Apr 6 14:10". The main window displays an SQL script for inserting data into the "ParkingLot" table. The script includes multiple INSERT INTO statements with values for columns like ParkingLotId, TotalSpace, AvailableSpace, Location, OccupancyRate, and Name. The code is color-coded for syntax highlighting. At the bottom of the script, it says "41:262 | 4 errors found". Below the script, there's a table titled "Action Output" with columns for Time, Action, Response, and Duration / Fetch Time. A status bar at the bottom shows various application icons.

```
308
309 ('REQ014', 'EMP009', '2024-03-28', 'LOT005', 'Pending', NULL);
310
311
312
313 -- 5) ParkingLot
314
315 • INSERT INTO ParkingLot (ParkingLotId, TotalSpace, AvailableSpace, Location, OccupancyRate, Name) VALUES
316 ('LOT001', 100, 50, '123 Main Street', 50.00, 'Main Street Parking Lot'),
317 ('LOT002', 150, 75, '456 Elm Street', 50.00, 'Elm Street Parking Lot'),
318 ('LOT003', 200, 100, '789 Oak Street', 50.00, 'Oak Street Parking Lot'),
319 ('LOT004', 120, 60, '101 Pine Street', 50.00, 'Pine Street Parking Lot');
320
321 • INSERT INTO ParkingLot (ParkingLotId, TotalSpace, AvailableSpace, Location, OccupancyRate, Name) VALUES
322 ('LOT005', 80, 40, '222 Maple Street', 50.00, 'Maple Street Parking Lot'),
323 ('LOT006', 170, 85, '333 Cherry Street', 50.00, 'Cherry Street Parking Lot'),
324 ('LOT007', 190, 95, '444 Walnut Street', 50.00, 'Walnut Street Parking Lot'),
325 ('LOT008', 110, 55, '555 Birch Street', 50.00, 'Birch Street Parking Lot');
326
327 • INSERT INTO ParkingLot (ParkingLotId, TotalSpace, AvailableSpace, Location, OccupancyRate, Name) VALUES
328 ('LOT009', 130, 65, '666 Cedar Street', 50.00, 'Cedar Street Parking Lot'),
329 ('LOT010', 160, 80, '777 Pineapple Street', 50.00, 'Pineapple Street Parking Lot'),
330 ('LOT011', 180, 90, '888 Orange Street', 50.00, 'Orange Street Parking Lot'),
331 ('LOT012', 200, 100, '999 Lemon Street', 50.00, 'Lemon Street Parking Lot');
332
333 -- 6) ParkingSpace
334
```

## ❖ Maintenance Request

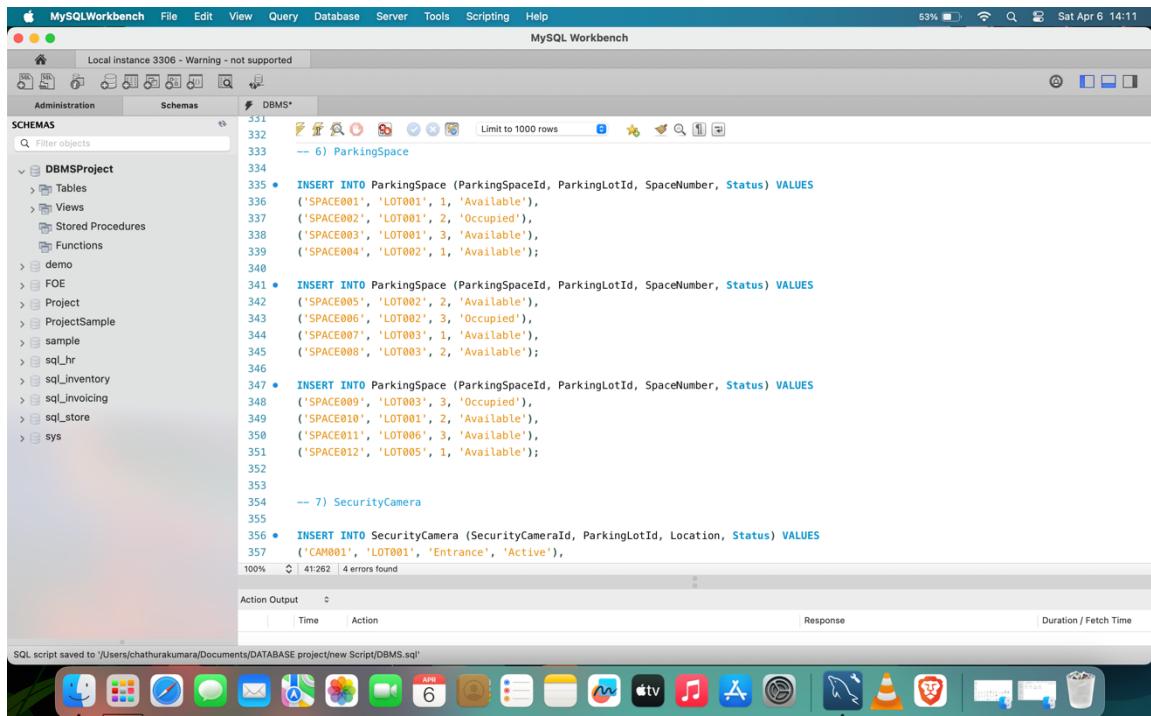


The screenshot shows the MySQL Workbench interface with the following SQL script being run:

```
288
289
290
291 • INSERT INTO MaintenanceRequest (RequestId, EmployeeId, RequestedDate, ParkingLotId, Status, CompletedDate) VALUES
292
293 ('REQ001', 'EMP001', '2024-03-20', 'LOT001', 'Pending', NULL),
294 ('REQ002', 'EMP002', '2024-03-20', 'LOT002', 'In Progress', NULL),
295 ('REQ003', 'EMP003', '2024-03-20', 'LOT003', 'Completed', '2024-03-21'),
296 ('REQ004', 'EMP004', '2024-03-20', 'LOT004', 'Pending', NULL),
297 ('REQ005', 'EMP001', '2024-03-19', 'LOT004', 'Pending', NULL);
298
299 • INSERT INTO MaintenanceRequest (RequestId, EmployeeId, RequestedDate, ParkingLotId, Status, CompletedDate) VALUES
300 ('REQ006', 'EMP002', '2024-03-21', 'LOT001', 'In Progress', NULL),
301 ('REQ007', 'EMP003', '2024-03-22', 'LOT005', 'Pending', NULL),
302 ('REQ008', 'EMP008', '2024-03-22', 'LOT003', 'In Progress', NULL),
303 ('REQ009', 'EMP001', '2024-03-23', 'LOT005', 'Completed', '2024-03-24');
304
305 • INSERT INTO MaintenanceRequest (RequestId, EmployeeId, RequestedDate, ParkingLotId, Status, CompletedDate) VALUES
306 ('REQ010', 'EMP009', '2024-03-25', 'LOT002', 'Pending', NULL),
307 ('REQ011', 'EMP010', '2024-03-26', 'LOT002', 'In Progress', NULL),
308 ('REQ012', 'EMP011', '2024-03-27', 'LOT003', 'Completed', '2024-03-28'),
309 ('REQ013', 'EMP002', '2024-03-28', 'LOT006', 'Pending', NULL),
310 ('REQ014', 'EMP009', '2024-03-28', 'LOT005', 'Pending', NULL);
311
312
313 --- 5) ParkingLot
314
```

The script is saving to '/Users/chathurakumara/Documents/DATABASE project/new Script/DBMS.sql'.

## ❖ Parking Space

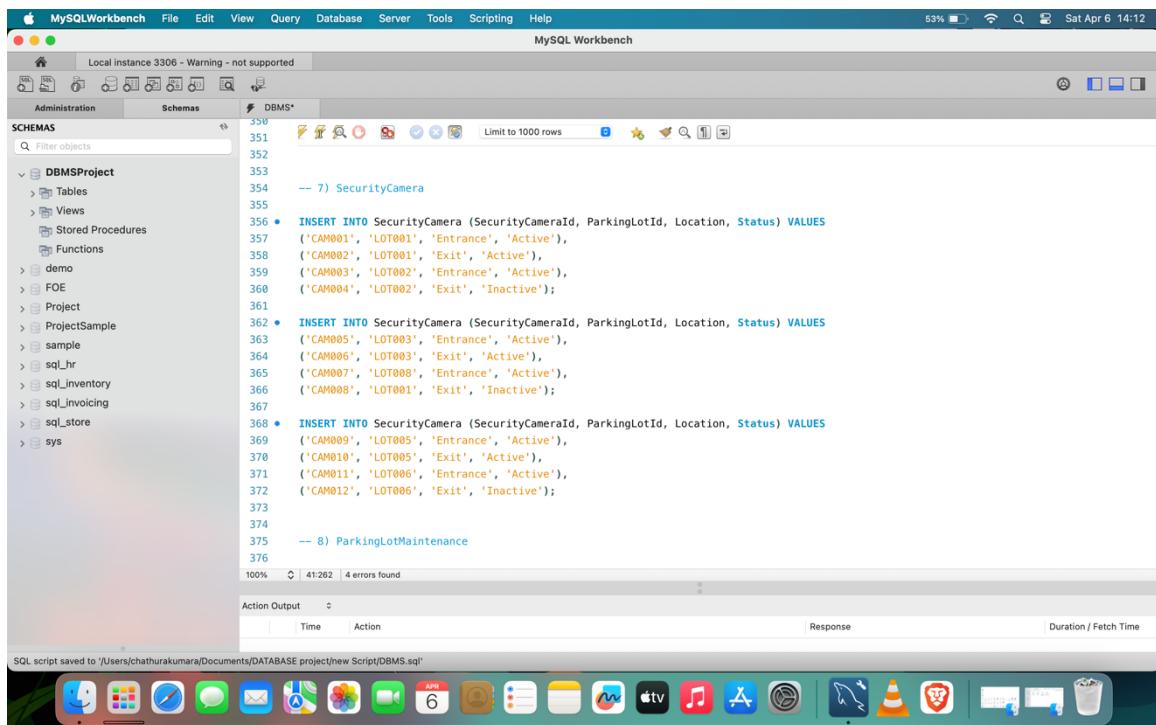


The screenshot shows the MySQL Workbench interface with the following SQL script being run:

```
331
332
333 --- 6) ParkingSpace
334
335 • INSERT INTO ParkingSpace (ParkingSpaceId, ParkingLotId, SpaceNumber, Status) VALUES
336 ('SPACE001', 'LOT001', 1, 'Available'),
337 ('SPACE002', 'LOT001', 2, 'Occupied'),
338 ('SPACE003', 'LOT001', 3, 'Available'),
339 ('SPACE004', 'LOT002', 1, 'Available');
340
341 • INSERT INTO ParkingSpace (ParkingSpaceId, ParkingLotId, SpaceNumber, Status) VALUES
342 ('SPACE005', 'LOT002', 2, 'Available'),
343 ('SPACE006', 'LOT002', 3, 'Occupied'),
344 ('SPACE007', 'LOT003', 1, 'Available'),
345 ('SPACE008', 'LOT003', 2, 'Available');
346
347 • INSERT INTO ParkingSpace (ParkingSpaceId, ParkingLotId, SpaceNumber, Status) VALUES
348 ('SPACE009', 'LOT003', 3, 'Occupied'),
349 ('SPACE010', 'LOT001', 2, 'Available'),
350 ('SPACE011', 'LOT006', 3, 'Available'),
351 ('SPACE012', 'LOT005', 1, 'Available');
352
353
354 --- 7) SecurityCamera
355
356 • INSERT INTO SecurityCamera (SecurityCameraId, ParkingLotId, Location, Status) VALUES
357 ('CAM001', 'LOT001', 'Entrance', 'Active'),
```

The script is saving to '/Users/chathurakumara/Documents/DATABASE project/new Script/DBMS.sql'.

## ❖ Security Camera



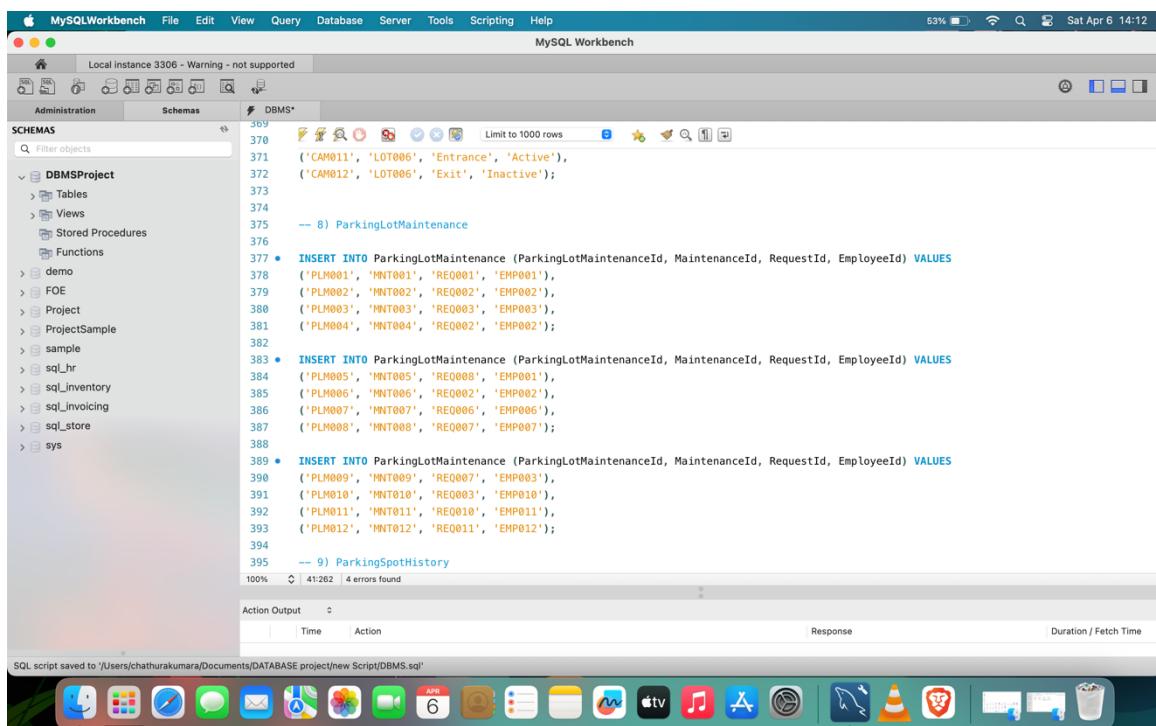
The screenshot shows the MySQL Workbench interface with the following details:

- Toolbar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Header:** Local instance 3306 - Warning - not supported, MySQL Workbench, Sat Apr 6 14:12.
- Schemas:** DBMSProject, demo, FOE, Project, ProjectSample, sample, sql\_hr, sql\_inventory, sql\_invoicing, sql\_store, sys.
- Code Editor:** Displays SQL scripts for Security Camera tables. The code includes:
  - Script 1 (SecurityCamera):

```
350
351
352
353
354 -- 7) SecurityCamera
355
356 • INSERT INTO SecurityCamera (SecurityCameraId, ParkingLotId, Location, Status) VALUES
('CAM001', 'LOT001', 'Entrance', 'Active'),
('CAM002', 'LOT001', 'Exit', 'Active'),
('CAM003', 'LOT002', 'Entrance', 'Active'),
('CAM004', 'LOT002', 'Exit', 'Inactive');
361
362 • INSERT INTO SecurityCamera (SecurityCameraId, ParkingLotId, Location, Status) VALUES
('CAM005', 'LOT003', 'Entrance', 'Active'),
('CAM006', 'LOT003', 'Exit', 'Active'),
('CAM007', 'LOT008', 'Entrance', 'Active'),
('CAM008', 'LOT001', 'Exit', 'Inactive');
367
374
375 -- 8) ParkingLotMaintenance
376
```
  - Script 2 (ParkingLotMaintenance):

```
368 • INSERT INTO SecurityCamera (SecurityCameraId, ParkingLotId, Location, Status) VALUES
('CAM009', 'LOT005', 'Entrance', 'Active'),
('CAM010', 'LOT005', 'Exit', 'Active'),
('CAM011', 'LOT006', 'Entrance', 'Active'),
('CAM012', 'LOT006', 'Exit', 'Inactive');
373
374
375 -- 8) ParkingLotMaintenance
376
```
- Status Bar:** 100%, 41:262, 4 errors found.
- Action Output:** Shows columns for Time, Action, Response, and Duration / Fetch Time.
- Bottom:** SQL script saved to '/Users/chathurakumara/Documents/DATABASE project/new Script/DBMS.sql'.

## ❖ Parking Lot Maintenance



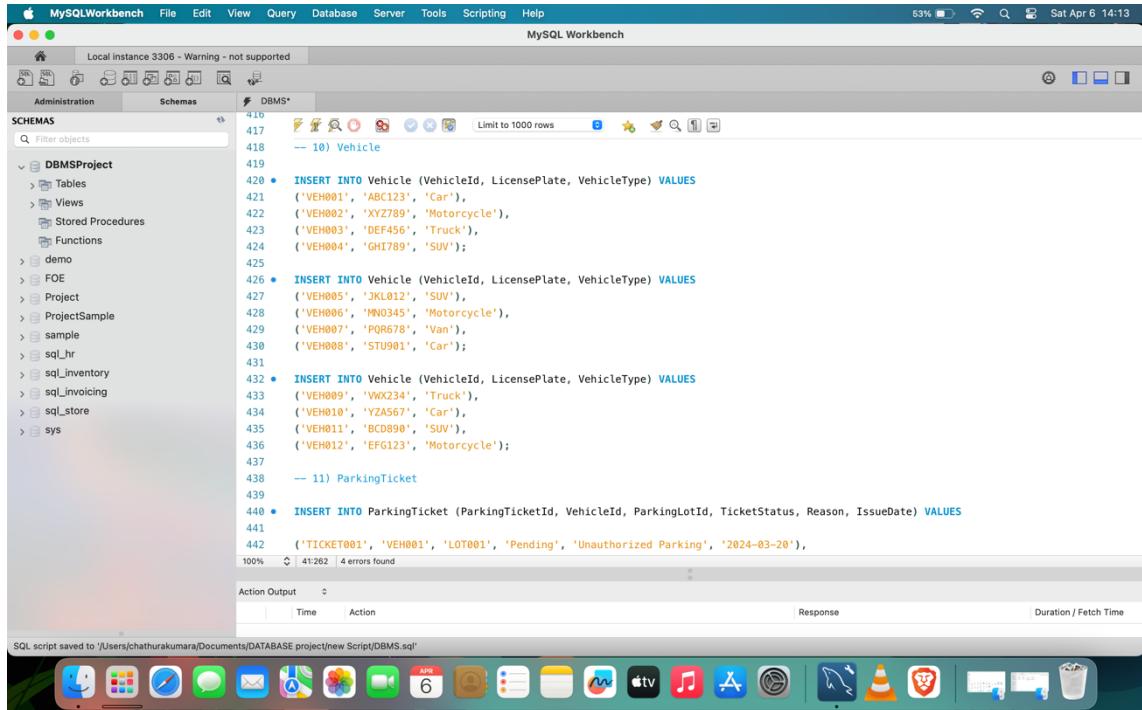
The screenshot shows the MySQL Workbench interface with the following details:

- Toolbar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Header:** Local instance 3306 - Warning - not supported, MySQL Workbench, Sat Apr 6 14:12.
- Schemas:** DBMSProject, demo, FOE, Project, ProjectSample, sample, sql\_hr, sql\_inventory, sql\_invoicing, sql\_store, sys.
- Code Editor:** Displays SQL scripts for Parking Lot Maintenance tables. The code includes:
  - Script 1 (ParkingLotMaintenance):

```
369
370
371 ('CAM011', 'LOT006', 'Entrance', 'Active'),
372 ('CAM012', 'LOT006', 'Exit', 'Inactive');
373
374
375 -- 8) ParkingLotMaintenance
376
```
  - Script 2 (ParkingLotMaintenance):

```
377 • INSERT INTO ParkingLotMaintenance (ParkingLotMaintenanceId, MaintenanceId, RequestId, EmployeeId) VALUES
('PLM001', 'MNT001', 'REQ001', 'EMP001'),
('PLM002', 'MNT002', 'REQ002', 'EMP002'),
('PLM003', 'MNT003', 'REQ003', 'EMP003'),
('PLM004', 'MNT004', 'REQ002', 'EMP002');
382
383 • INSERT INTO ParkingLotMaintenance (ParkingLotMaintenanceId, MaintenanceId, RequestId, EmployeeId) VALUES
('PLM005', 'MNT005', 'REQ008', 'EMP001'),
('PLM006', 'MNT006', 'REQ002', 'EMP002'),
('PLM007', 'MNT007', 'REQ006', 'EMP006'),
('PLM008', 'MNT008', 'REQ007', 'EMP007');
388
389 • INSERT INTO ParkingLotMaintenance (ParkingLotMaintenanceId, MaintenanceId, RequestId, EmployeeId) VALUES
('PLM009', 'MNT009', 'REQ007', 'EMP003'),
('PLM010', 'MNT010', 'REQ003', 'EMP010'),
('PLM011', 'MNT011', 'REQ010', 'EMP011'),
('PLM012', 'MNT012', 'REQ011', 'EMP012');
394
395 -- 9) ParkingSpotHistory
```
- Status Bar:** 100%, 41:262, 4 errors found.
- Action Output:** Shows columns for Time, Action, Response, and Duration / Fetch Time.
- Bottom:** SQL script saved to '/Users/chathurakumara/Documents/DATABASE project/new Script/DBMS.sql'.

## ❖ Vehicle

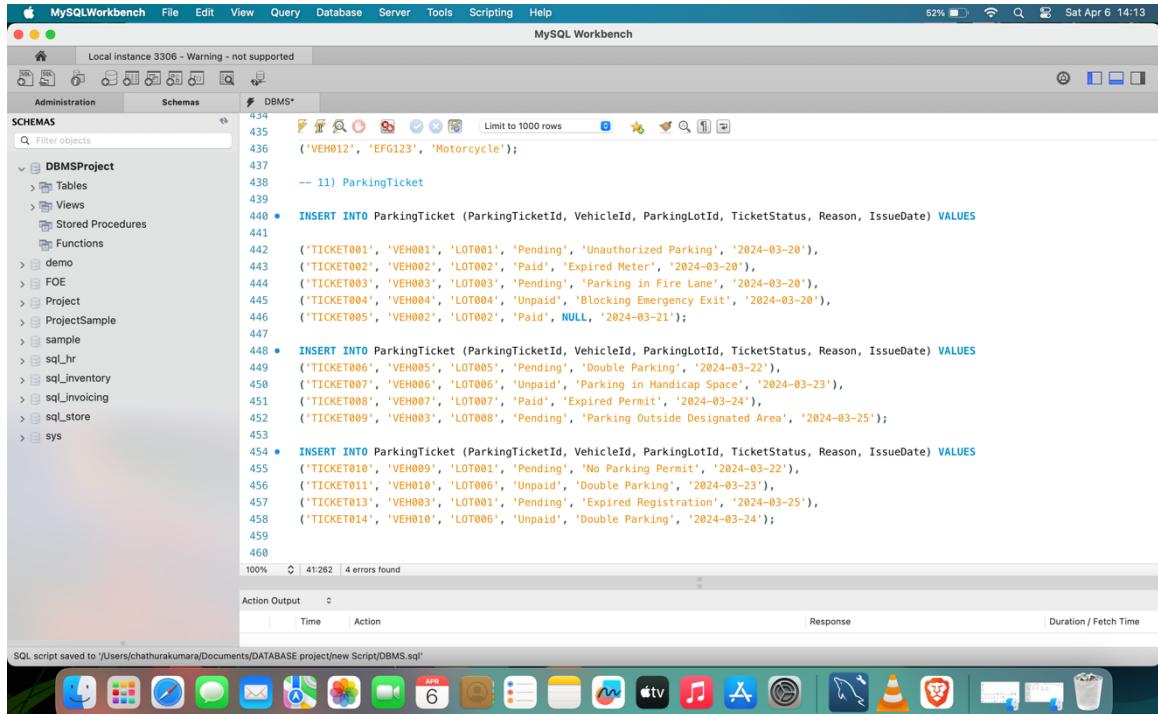


The screenshot shows the MySQL Workbench interface with the 'Vehicle' schema selected. The left sidebar lists various database objects like Tables, Views, and Procedures. The main pane displays a SQL script for populating the 'Vehicle' table with 10 entries. The status bar at the bottom indicates the script was saved to a local file.

```
416
417
418 -- 10) Vehicle
419
420 • INSERT INTO Vehicle (VehicleId, LicensePlate, VehicleType) VALUES
421 ('VEH001', 'ABC123', 'Car'),
422 ('VEH002', 'XYZ789', 'Motorcycle'),
423 ('VEH003', 'DEF456', 'Truck'),
424 ('VEH004', 'GHI789', 'SUV');
425
426 • INSERT INTO Vehicle (VehicleId, LicensePlate, VehicleType) VALUES
427 ('VEH005', 'JKL012', 'SUV'),
428 ('VEH006', 'MNO345', 'Motorcycle'),
429 ('VEH007', 'PQR678', 'Van'),
430 ('VEH008', 'STU901', 'Car');
431
432 • INSERT INTO Vehicle (VehicleId, LicensePlate, VehicleType) VALUES
433 ('VEH010', 'VWX234', 'Truck'),
434 ('VEH011', 'YZA567', 'Car'),
435 ('VEH012', 'BCD890', 'SUV'),
436 ('VEH013', 'EFG123', 'Motorcycle');
437
438 -- 11) ParkingTicket
439
440 • INSERT INTO ParkingTicket (ParkingTicketId, VehicleId, ParkingLotId, TicketStatus, Reason, IssueDate) VALUES
441
442 ('TICKET001', 'VEH001', 'LOT001', 'Pending', 'Unauthorized Parking', '2024-03-20'),
443
444
445
446
447
448 • INSERT INTO ParkingTicket (ParkingTicketId, VehicleId, ParkingLotId, TicketStatus, Reason, IssueDate) VALUES
449
450
451
452
453
454 • INSERT INTO ParkingTicket (ParkingTicketId, VehicleId, ParkingLotId, TicketStatus, Reason, IssueDate) VALUES
455
456
457
458
459
460
```

SQL script saved to '/Users/chathurakumara/Documents/DATABASE project/new Script/DBMS.sql'

## ❖ Parking Ticket

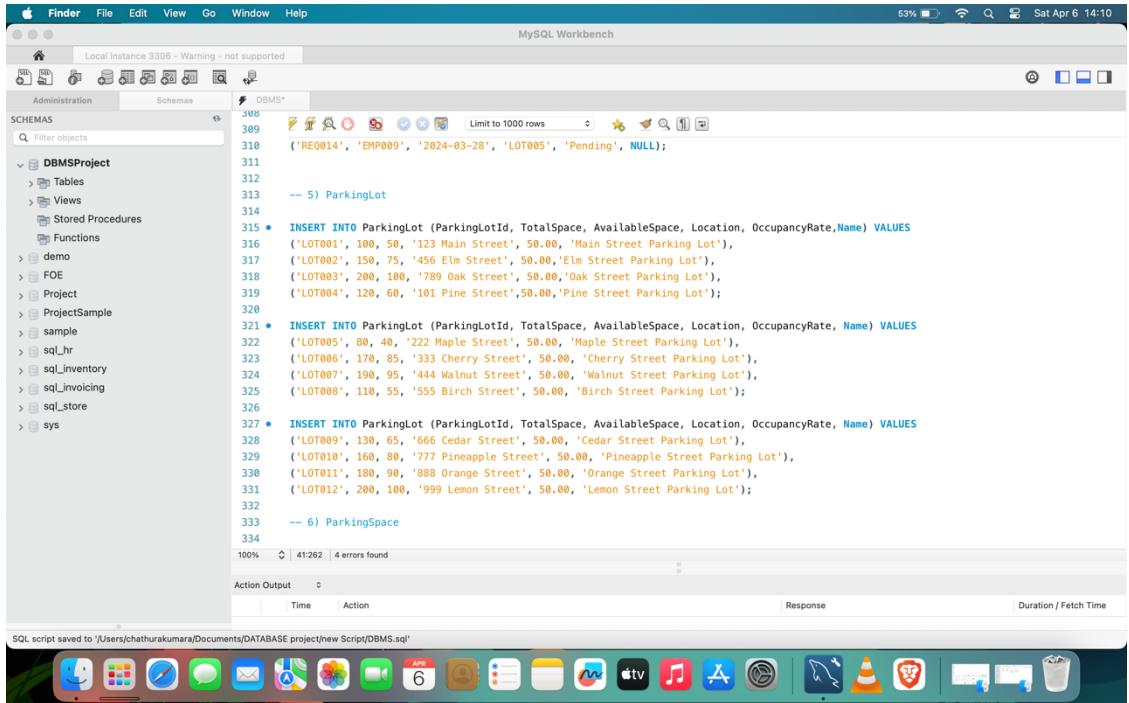


The screenshot shows the MySQL Workbench interface with the 'Parking Ticket' schema selected. The left sidebar lists various database objects. The main pane displays a SQL script for populating the 'ParkingTicket' table with 11 entries. The status bar at the bottom indicates the script was saved to a local file.

```
434
435
436 ('VEH012', 'EFG123', 'Motorcycle');
437
438 -- 11) ParkingTicket
439
440 • INSERT INTO ParkingTicket (ParkingTicketId, VehicleId, ParkingLotId, TicketStatus, Reason, IssueDate) VALUES
441
442 ('TICKET001', 'VEH001', 'LOT001', 'Pending', 'Unauthorized Parking', '2024-03-20'),
443 ('TICKET002', 'VEH002', 'LOT002', 'Paid', 'Expired Meter', '2024-03-20'),
444 ('TICKET003', 'VEH003', 'LOT003', 'Pending', 'Parking in Fire Lane', '2024-03-20'),
445 ('TICKET004', 'VEH004', 'LOT004', 'Unpaid', 'Blocking Emergency Exit', '2024-03-20'),
446 ('TICKET005', 'VEH002', 'LOT002', 'Paid', 'NULL', '2024-03-21');
447
448 • INSERT INTO ParkingTicket (ParkingTicketId, VehicleId, ParkingLotId, TicketStatus, Reason, IssueDate) VALUES
449
450
451
452
453
454 • INSERT INTO ParkingTicket (ParkingTicketId, VehicleId, ParkingLotId, TicketStatus, Reason, IssueDate) VALUES
455
456
457
458
459
460
```

SQL script saved to '/Users/chathurakumara/Documents/DATABASE project/new Script/DBMS.sql'

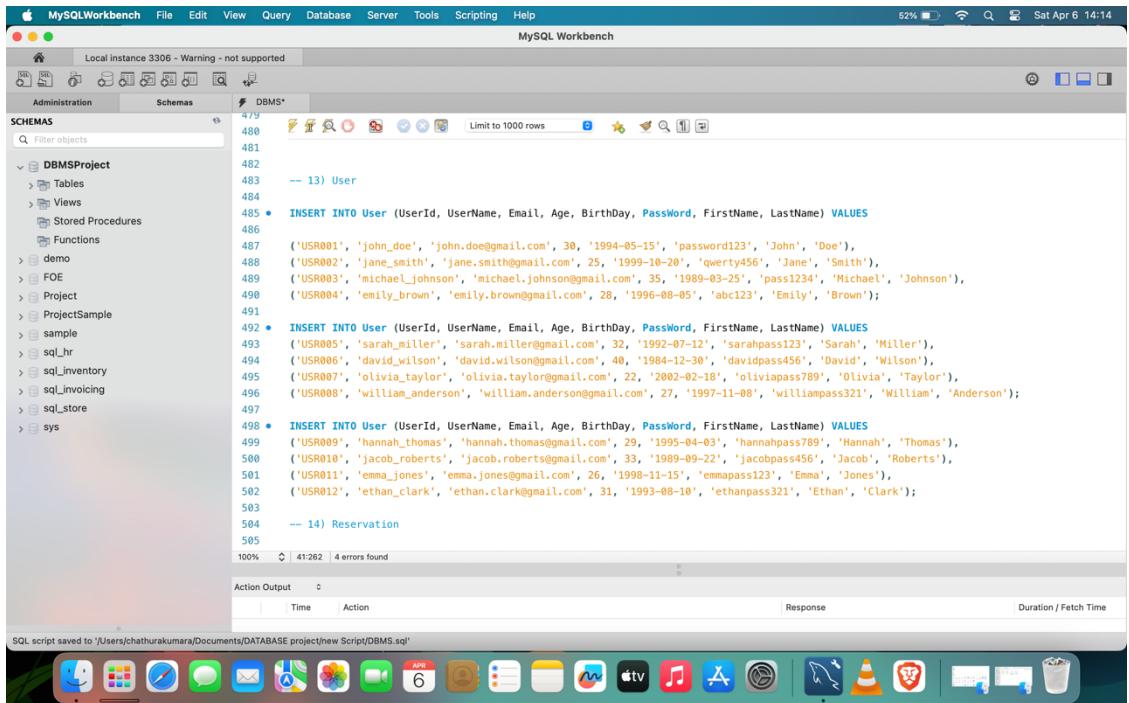
## ❖ Payment



The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The title bar reads "MySQL Workbench" and "Local instance 3306 - Warning - not supported". The main window displays a SQL script for creating a "ParkingLot" table. The script includes several INSERT statements with values for columns like ParkingLotId, TotalSpace, AvailableSpace, Location, OccupancyRate, and Name. The code is color-coded for syntax highlighting. At the bottom of the script, it says "100% 41:262 4 errors found". Below the script, there's a table header for "Action Output" with columns "Time", "Action", "Response", and "Duration / Fetch Time". A status bar at the bottom of the screen shows various application icons.

```
308
309
310 ('RE0014', 'EMP009', '2024-03-28', 'LOT005', 'Pending', NULL);
311
312 -- 5) ParkingLot
313
314
315 • INSERT INTO ParkingLot (ParkingLotId, TotalSpace, AvailableSpace, Location, OccupancyRate, Name) VALUES
316 ('LOT001', 100, 50, '123 Main Street', 50.00, 'Main Street Parking Lot'),
317 ('LOT002', 150, 75, '456 Elm Street', 50.00, 'Elm Street Parking Lot'),
318 ('LOT003', 200, 100, '789 Oak Street', 50.00, 'Oak Street Parking Lot'),
319 ('LOT004', 120, 60, '101 Pine Street', 50.00, 'Pine Street Parking Lot');
320
321 • INSERT INTO ParkingLot (ParkingLotId, TotalSpace, AvailableSpace, Location, OccupancyRate, Name) VALUES
322 ('LOT005', 80, 40, '222 Maple Street', 50.00, 'Maple Street Parking Lot'),
323 ('LOT006', 170, 85, '333 Cherry Street', 50.00, 'Cherry Street Parking Lot'),
324 ('LOT007', 190, 95, '444 Walnut Street', 50.00, 'Walnut Street Parking Lot'),
325 ('LOT008', 110, 55, '555 Birch Street', 50.00, 'Birch Street Parking Lot');
326
327 • INSERT INTO ParkingLot (ParkingLotId, TotalSpace, AvailableSpace, Location, OccupancyRate, Name) VALUES
328 ('LOT009', 130, 65, '666 Cedar Street', 50.00, 'Cedar Street Parking Lot'),
329 ('LOT010', 160, 80, '777 Pineapple Street', 50.00, 'Pineapple Street Parking Lot'),
330 ('LOT011', 180, 90, '888 Orange Street', 50.00, 'Orange Street Parking Lot'),
331 ('LOT012', 200, 100, '999 Lemon Street', 50.00, 'Lemon Street Parking Lot');
332
333 -- 6) ParkingSpace
334
```

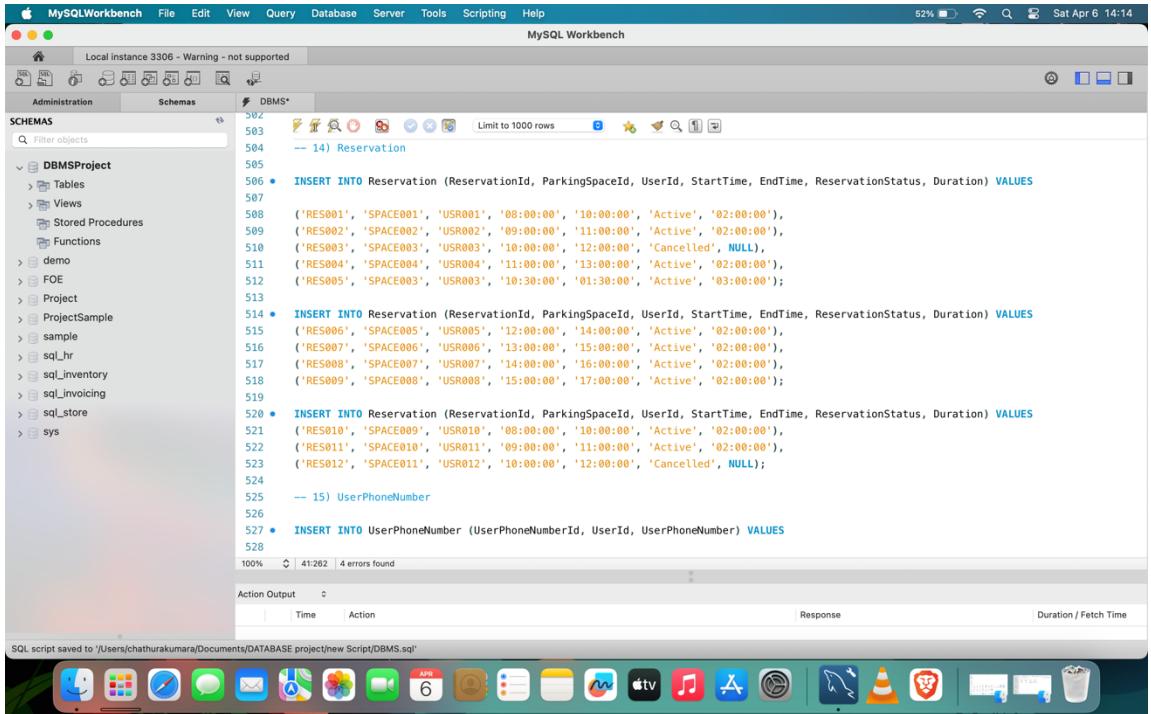
## ❖ User



The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The title bar reads "MySQL Workbench" and "Local instance 3306 - Warning - not supported". The main window displays a SQL script for creating a "User" table. The script includes several INSERT statements with values for columns like UserId, UserName, Email, Age, BirthDay, PassWord, FirstName, and LastName. The code is color-coded for syntax highlighting. At the bottom of the script, it says "100% 41:262 4 errors found". Below the script, there's a table header for "Action Output" with columns "Time", "Action", "Response", and "Duration / Fetch Time". A status bar at the bottom of the screen shows various application icons.

```
479
480
481
482
483 -- 13) User
484
485 • INSERT INTO User (UserId, UserName, Email, Age, BirthDay, PassWord, FirstName, LastName) VALUES
486
487 ('USR001', 'john_doe', 'john.doe@gmail.com', 30, '1994-05-15', 'password123', 'John', 'Doe'),
488 ('USR002', 'jane_smith', 'jane.smith@gmail.com', 25, '1999-10-20', 'qwerty456', 'Jane', 'Smith'),
489 ('USR003', 'michael_johnson', 'michael.johnson@gmail.com', 35, '1989-03-25', 'pass1234', 'Michael', 'Johnson'),
490 ('USR004', 'emily_brown', 'emily.brown@gmail.com', 28, '1996-08-05', 'abc123', 'Emily', 'Brown');
491
492 • INSERT INTO User (UserId, UserName, Email, Age, BirthDay, PassWord, FirstName, LastName) VALUES
493 ('USR005', 'sarah_miller', 'sarah.miller@gmail.com', 32, '1992-07-12', 'sarahpass123', 'Sarah', 'Miller'),
494 ('USR006', 'david_wilson', 'david.wilson@gmail.com', 40, '1984-12-30', 'davidpass456', 'David', 'Wilson'),
495 ('USR007', 'olivia_taylor', 'olivia.taylor@gmail.com', 22, '2002-02-18', 'oliviapass789', 'Olivia', 'Taylor'),
496 ('USR008', 'william_anderson', 'william.anderson@gmail.com', 27, '1997-11-08', 'williampass321', 'William', 'Anderson');
497
498 • INSERT INTO User (UserId, UserName, Email, Age, BirthDay, PassWord, FirstName, LastName) VALUES
499 ('USR009', 'hannah_thomas', 'hannah.thomas@gmail.com', 29, '1995-04-03', 'hannahpass789', 'Hannah', 'Thomas'),
500 ('USR010', 'jacob_roberts', 'jacob.roberts@gmail.com', 33, '1988-09-22', 'jacobjpass456', 'Jacob', 'Roberts'),
501 ('USR011', 'emma_jones', 'emma.jones@gmail.com', 26, '1998-11-15', 'emmapass123', 'Emma', 'Jones'),
502 ('USR012', 'ethan_clark', 'ethan.clark@gmail.com', 31, '1993-08-10', 'ethanpass321', 'Ethan', 'Clark');
503
504 -- 14) Reservation
505
```

## ❖ Reservation



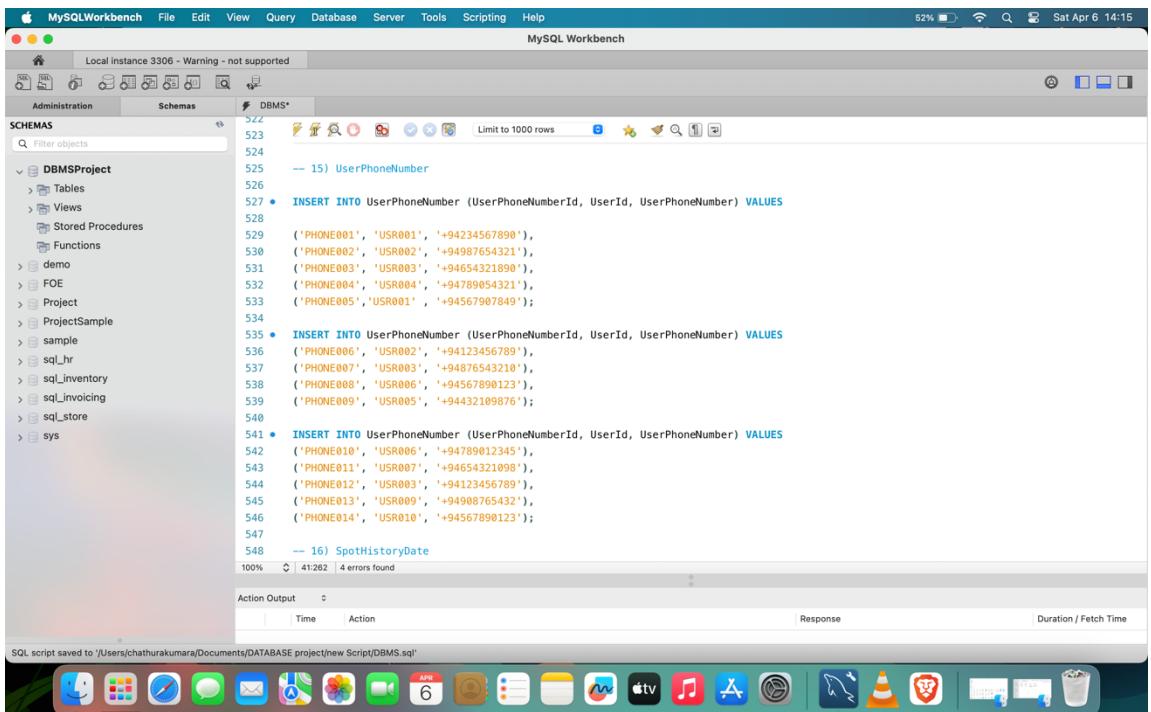
The screenshot shows the MySQL Workbench interface with the following details:

- Title Bar:** MySQL Workbench, Local instance 3306 - Warning - not supported, Sat Apr 6 14:14.
- Schemas:** DBMSProject is selected.
- DBMS\* Tab:** Shows a list of SQL statements:

  - 502 -- 14) Reservation
  - 503 504 505 506 • INSERT INTO Reservation (ReservationId, ParkingSpaceId, UserId, StartTime, EndTime, ReservationStatus, Duration) VALUES ('RES001', 'SPACE001', 'USR001', '08:00:00', '10:00:00', 'Active', '02:00:00'), ('RES002', 'SPACE002', 'USR002', '09:00:00', '11:00:00', 'Active', '02:00:00'), ('RES003', 'SPACE003', 'USR003', '10:00:00', '12:00:00', 'Cancelled', NULL), ('RES004', 'SPACE004', 'USR004', '11:00:00', '13:00:00', 'Active', '02:00:00'), ('RES005', 'SPACE005', 'USR003', '10:30:00', '01:30:00', 'Active', '03:00:00');
  - 513 514 • INSERT INTO Reservation (ReservationId, ParkingSpaceId, UserId, StartTime, EndTime, ReservationStatus, Duration) VALUES ('RES006', 'SPACE005', 'USR005', '12:00:00', '14:00:00', 'Active', '02:00:00'), ('RES007', 'SPACE006', 'USR006', '13:00:00', '15:00:00', 'Active', '02:00:00'), ('RES008', 'SPACE007', 'USR007', '14:00:00', '16:00:00', 'Active', '02:00:00'), ('RES009', 'SPACE008', 'USR008', '15:00:00', '17:00:00', 'Active', '02:00:00);
  - 519 520 • INSERT INTO Reservation (ReservationId, ParkingSpaceId, UserId, StartTime, EndTime, ReservationStatus, Duration) VALUES ('RES010', 'SPACE009', 'USR010', '08:00:00', '10:00:00', 'Active', '02:00:00'), ('RES011', 'SPACE010', 'USR011', '09:00:00', '11:00:00', 'Active', '02:00:00'), ('RES012', 'SPACE011', 'USR012', '10:00:00', '12:00:00', 'Cancelled', NULL);
  - 524 525 -- 15) UserPhoneNumber
  - 526 527 • INSERT INTO UserPhoneNumber (UserPhoneNumberId, UserId, UserPhoneNumber) VALUES
  - 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 -- 16) SpotHistoryDate

- Bottom Status Bar:** SQL script saved to '/Users/chathurakumara/Documents/DATABASE project/new Script/DBMS.sql'

## ❖ Phone Number



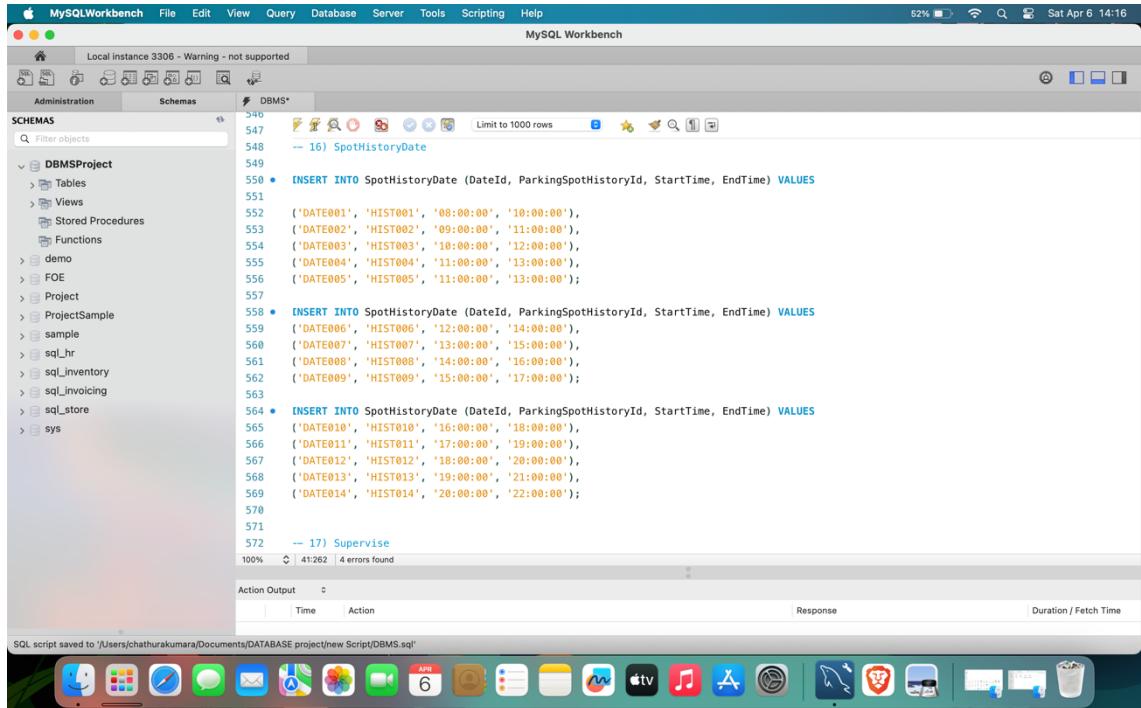
The screenshot shows the MySQL Workbench interface with the following details:

- Title Bar:** MySQL Workbench, Local instance 3306 - Warning - not supported, Sat Apr 6 14:15.
- Schemas:** DBMSProject is selected.
- DBMS\* Tab:** Shows a list of SQL statements:

  - 522 -- 15) UserPhoneNumber
  - 523 524 525 526 527 • INSERT INTO UserPhoneNumber (UserPhoneNumberId, UserId, UserPhoneNumber) VALUES ('PHONE001', 'USR001', '+94234567890'), ('PHONE002', 'USR002', '+94987654321'), ('PHONE003', 'USR003', '+94654321890'), ('PHONE004', 'USR004', '+94789054321'), ('PHONE005', 'USR001', '+94567987649');
  - 534 535 • INSERT INTO UserPhoneNumber (UserPhoneNumberId, UserId, UserPhoneNumber) VALUES ('PHONE006', 'USR002', '+94123456789'), ('PHONE007', 'USR003', '+94876543210'), ('PHONE008', 'USR006', '+94567890123'), ('PHONE009', 'USR005', '+94432109876');
  - 540 541 • INSERT INTO UserPhoneNumber (UserPhoneNumberId, UserId, UserPhoneNumber) VALUES ('PHONE010', 'USR006', '+94789012345'), ('PHONE011', 'USR007', '+94654321998'), ('PHONE012', 'USR003', '+94123456789'), ('PHONE013', 'USR009', '+94980765432'), ('PHONE014', 'USR010', '+94567890123');
  - 547 548 -- 16) SpotHistoryDate

- Bottom Status Bar:** SQL script saved to '/Users/chathurakumara/Documents/DATABASE project/new Script/DBMS.sql'

## ❖ SpotHistoryDate.

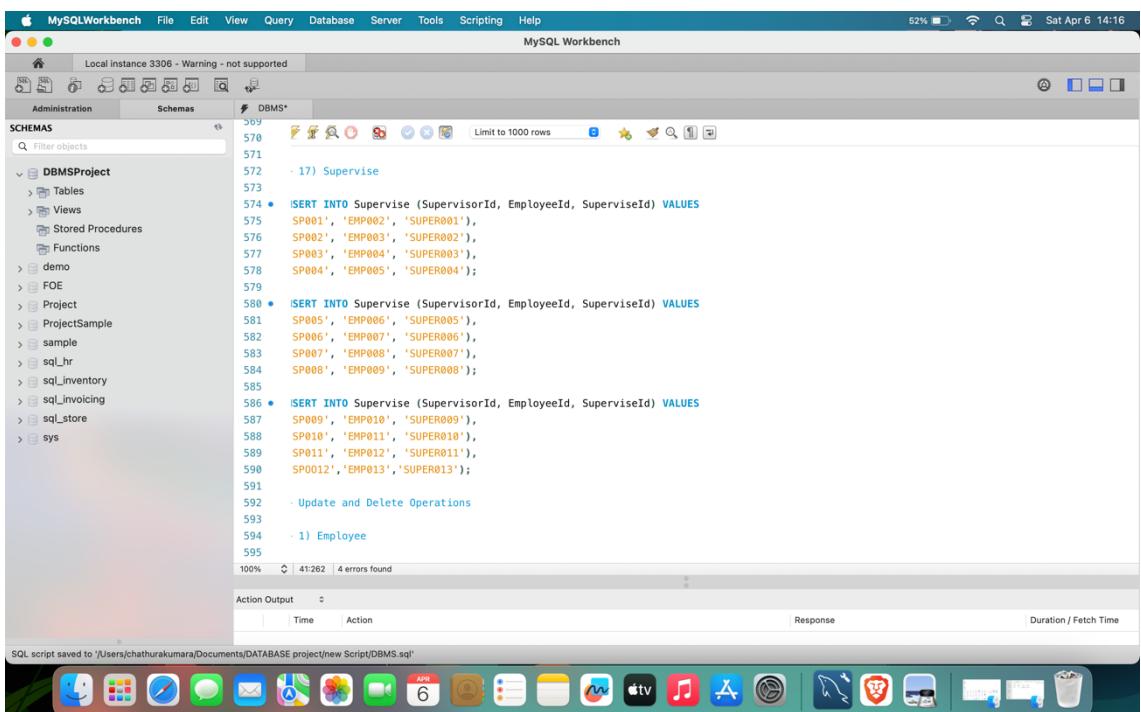


The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The title bar reads "MySQLWorkbench" and "Local instance 3306 - Warning - not supported". The main window displays a SQL editor with the following code:

```
540
541
542
543
544
545
546
547
548 -- 16) SpotHistoryDate
549
550 • INSERT INTO SpotHistoryDate (DateId, ParkingSpotHistoryId, StartTime, EndTime) VALUES
551
552 ('DATE001', 'HIST001', '08:00:00', '10:00:00'),
553 ('DATE002', 'HIST002', '09:00:00', '11:00:00'),
554 ('DATE003', 'HIST003', '10:00:00', '12:00:00'),
555 ('DATE004', 'HIST004', '11:00:00', '13:00:00'),
556 ('DATE005', 'HIST005', '11:00:00', '13:00:00');
557
558 • INSERT INTO SpotHistoryDate (DateId, ParkingSpotHistoryId, StartTime, EndTime) VALUES
559 ('DATE006', 'HIST006', '12:00:00', '14:00:00'),
560 ('DATE007', 'HIST007', '13:00:00', '15:00:00'),
561 ('DATE008', 'HIST008', '14:00:00', '16:00:00'),
562 ('DATE009', 'HIST009', '15:00:00', '17:00:00');
563
564 • INSERT INTO SpotHistoryDate (DateId, ParkingSpotHistoryId, StartTime, EndTime) VALUES
565 ('DATE010', 'HIST010', '16:00:00', '18:00:00'),
566 ('DATE011', 'HIST011', '17:00:00', '19:00:00'),
567 ('DATE012', 'HIST012', '18:00:00', '20:00:00'),
568 ('DATE013', 'HIST013', '19:00:00', '21:00:00'),
569 ('DATE014', 'HIST014', '20:00:00', '22:00:00');
570
571
572 -- 17) Supervise
```

The status bar at the bottom indicates "100%" completion, "41:262" seconds, and "4 errors found". The toolbar above the editor includes icons for Home, New, Open, Save, Import, Export, Find, Replace, and Help. The bottom dock shows various Mac OS X application icons.

## ❖ Supervise



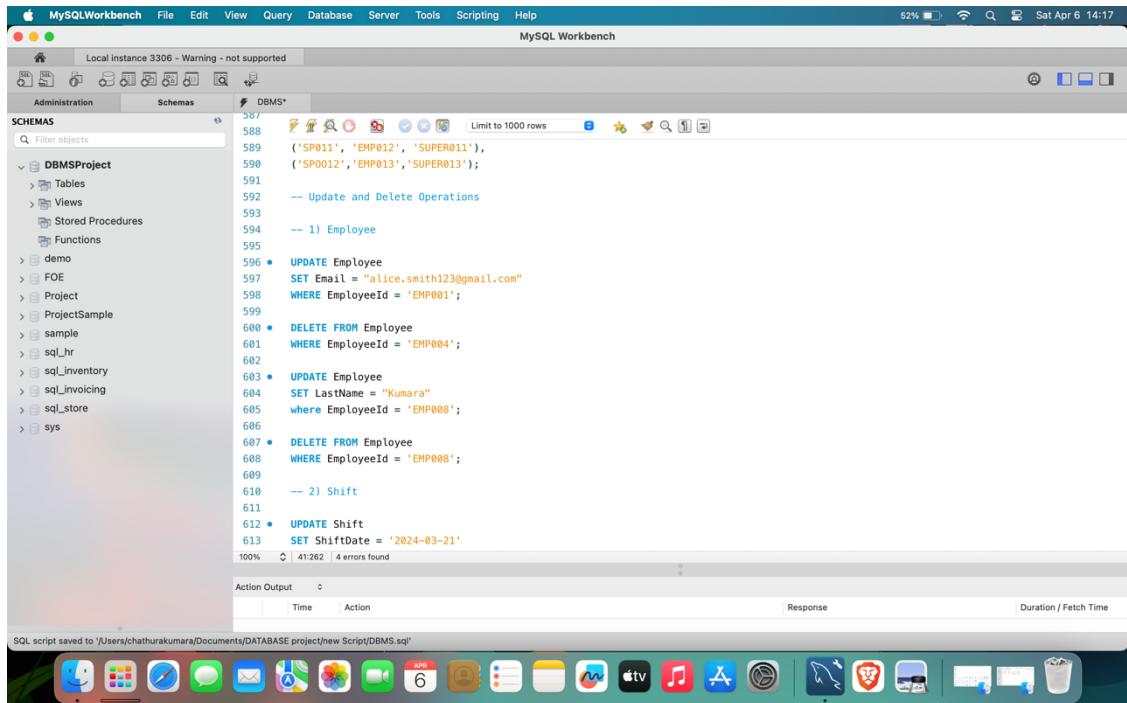
The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The title bar reads "MySQLWorkbench" and "Local instance 3306 - Warning - not supported". The main window displays a SQL editor with the following code:

```
569
570
571
572 -- 17) Supervise
573
574 • INSERT INTO Supervise (SupervisorId, EmployeeId, SuperviseId) VALUES
575 'SP001', 'EMP002', 'SUPER001',
576 'SP002', 'EMP003', 'SUPER002',
577 'SP003', 'EMP004', 'SUPER003',
578 'SP004', 'EMP005', 'SUPER004';
579
580 • INSERT INTO Supervise (SupervisorId, EmployeeId, SuperviseId) VALUES
581 'SP005', 'EMP006', 'SUPER005',
582 'SP006', 'EMP007', 'SUPER006',
583 'SP007', 'EMP008', 'SUPER007',
584 'SP008', 'EMP009', 'SUPER008';
585
586 • INSERT INTO Supervise (SupervisorId, EmployeeId, SuperviseId) VALUES
587 'SP009', 'EMP010', 'SUPER009',
588 'SP010', 'EMP011', 'SUPER010',
589 'SP011', 'EMP012', 'SUPER011',
590 'SP0012', 'EMP013', 'SUPER013';
591
592 -- Update and Delete Operations
593
594 -- 1) Employee
595
```

The status bar at the bottom indicates "100%" completion, "41:262" seconds, and "4 errors found". The toolbar above the editor includes icons for Home, New, Open, Save, Import, Export, Find, Replace, and Help. The bottom dock shows various Mac OS X application icons.

## 3.4 Update And Delete

### ❖ Employee

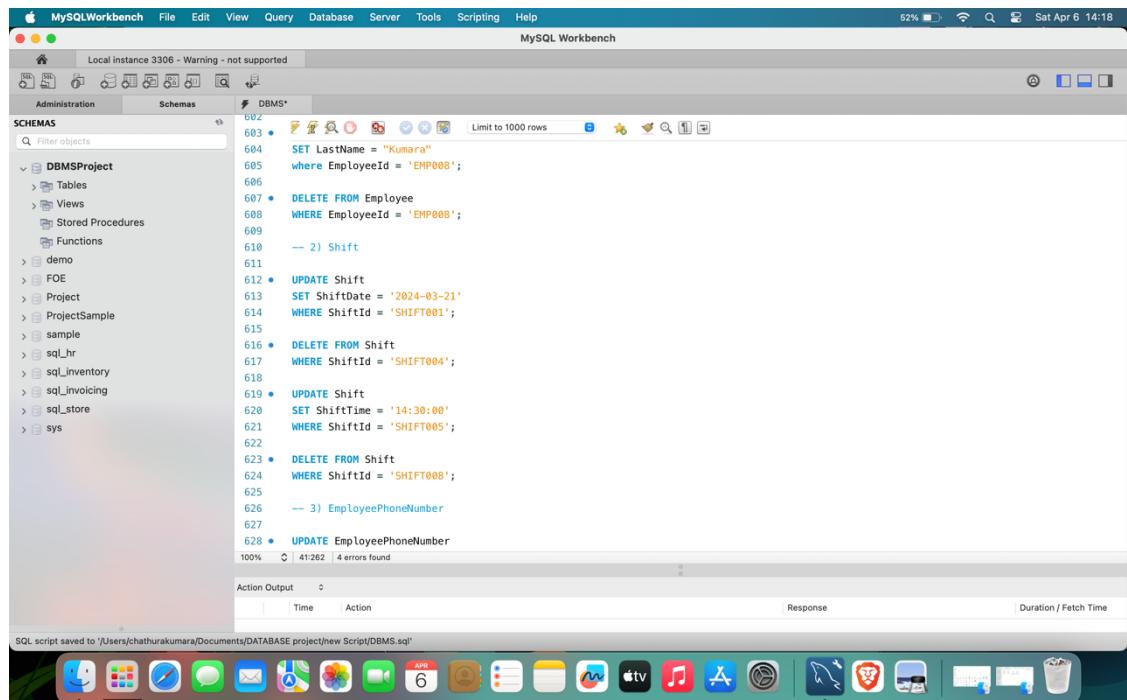


The screenshot shows the MySQL Workbench interface with the DBMS\* tab selected. The left sidebar displays the schema structure under 'DBMSProject'. The main pane contains the following SQL script:

```
587
588 ('SP011', 'EMP012', 'SUPER011'),
589 ('SP0012', 'EMP013', 'SUPER013');
590
591 -- Update and Delete Operations
592
593 --- 1) Employee
594
595
596 • UPDATE Employee
597 SET Email = "alice.smith123@gmail.com"
598 WHERE EmployeeId = 'EMP001';
599
600 • DELETE FROM Employee
601 WHERE EmployeeId = 'EMP004';
602
603 • UPDATE Employee
604 SET LastName = "Kumara"
605 where EmployeeId = 'EMP008';
606
607 • DELETE FROM Employee
608 WHERE EmployeeId = 'EMP008';
609
610 --- 2) Shift
611
612 • UPDATE Shift
613 SET ShiftDate = '2024-03-21'
100% 41/262 | 4 errors found
```

The status bar at the bottom indicates the script was saved to '/Users/chathurakumara/Documents/DATABASE project/new Script/DBMS.sql'.

### ❖ Shift

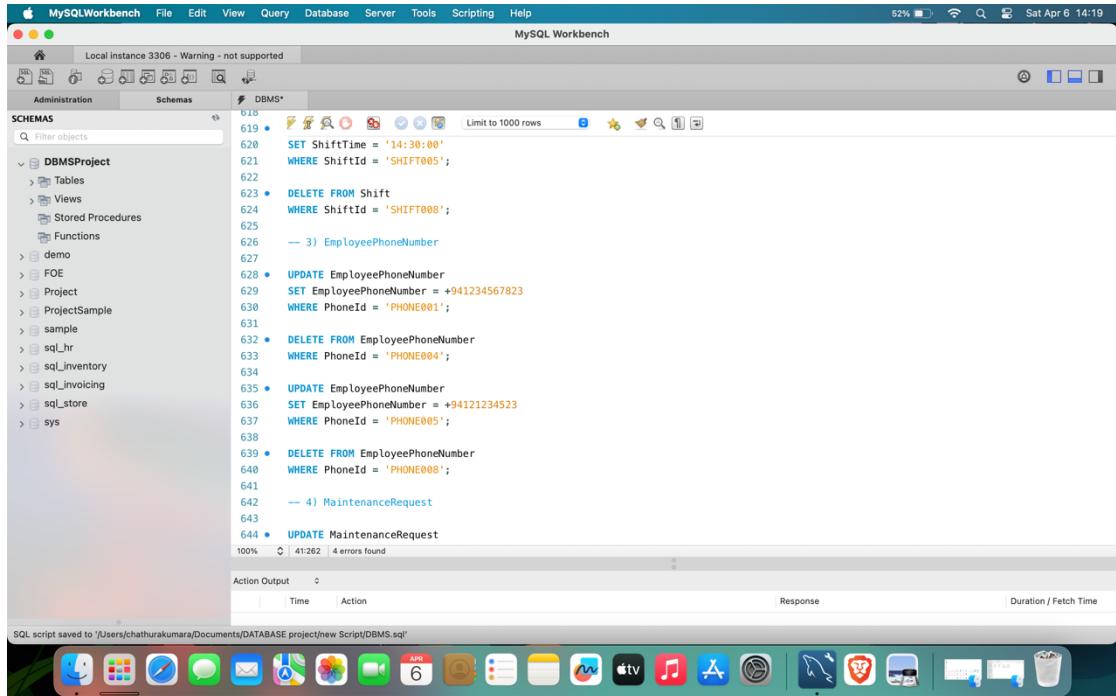


The screenshot shows the MySQL Workbench interface with the DBMS\* tab selected. The left sidebar displays the schema structure under 'DBMSProject'. The main pane contains the following SQL script:

```
602
603 • SET LastName = "Kumara"
604
605 where EmployeeId = 'EMP008';
606
607 • DELETE FROM Employee
608 WHERE EmployeeId = 'EMP008';
609
610 --- 2) Shift
611
612 • UPDATE Shift
613 SET ShiftDate = '2024-03-21'
614 WHERE ShiftId = 'SHIFT001';
615
616 • DELETE FROM Shift
617 WHERE ShiftId = 'SHIFT004';
618
619 • UPDATE Shift
620 SET ShiftTime = '14:30:00'
621 WHERE ShiftId = 'SHIFT005';
622
623 • DELETE FROM Shift
624 WHERE ShiftId = 'SHIFT008';
625
626 --- 3) EmployeePhoneNumber
627
628 • UPDATE EmployeePhoneNumber
100% 41/262 | 4 errors found
```

The status bar at the bottom indicates the script was saved to '/Users/chathurakumara/Documents/DATABASE project/new Script/DBMS.sql'.

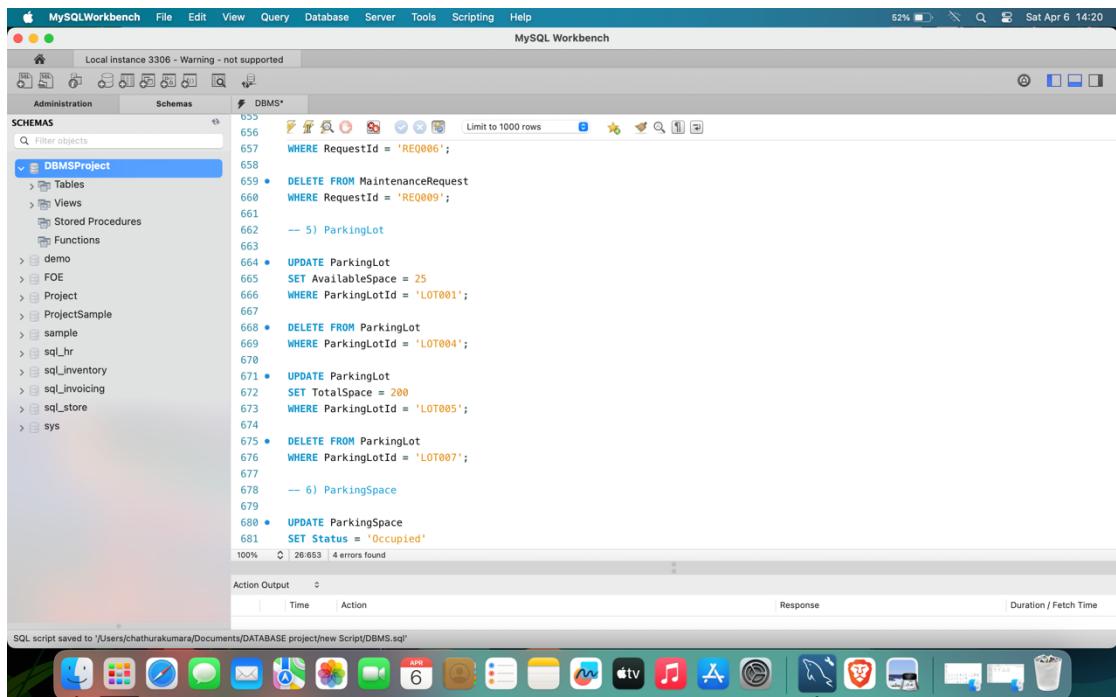
## ❖ Employee Phone Number



The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The title bar reads "MySQL Workbench" and "Local instance 3306 - Warning - not supported". The main window displays a SQL script titled "DBMS\*". The script contains several SQL statements, mostly UPDATE and DELETE queries, related to the "EmployeePhoneNumber" table. The code is numbered from 618 to 643. A message at the bottom indicates "4 errors found". Below the script, there's a table header for "Action Output" with columns "Time", "Action", "Response", and "Duration / Fetch Time". The status bar at the bottom shows "SQL script saved to '/Users/chathurakumara/Documents/DATABASE project/new Script/DBMS.sql'".

```
618
619 • SET ShiftTime = '14:30:00'
620 WHERE ShiftId = 'SHIFT005';
621
622 • DELETE FROM Shift
623 WHERE ShiftId = 'SHIFT008';
624
625 -- 3) EmployeePhoneNumber
626
627 • UPDATE EmployeePhoneNumber
628 SET EmployeePhoneNumber = '+941234567823'
629 WHERE PhoneId = 'PHONE001';
630
631 • UPDATE EmployeePhoneNumber
632 SET EmployeePhoneNumber = '+94121234523'
633 WHERE PhoneId = 'PHONE004';
634
635 • UPDATE EmployeePhoneNumber
636 SET EmployeePhoneNumber = '+94121234523'
637 WHERE PhoneId = 'PHONE005';
638
639 • DELETE FROM EmployeePhoneNumber
640 WHERE PhoneId = 'PHONE008';
641
642 -- 4) MaintenanceRequest
643
644 • UPDATE MaintenanceRequest
```

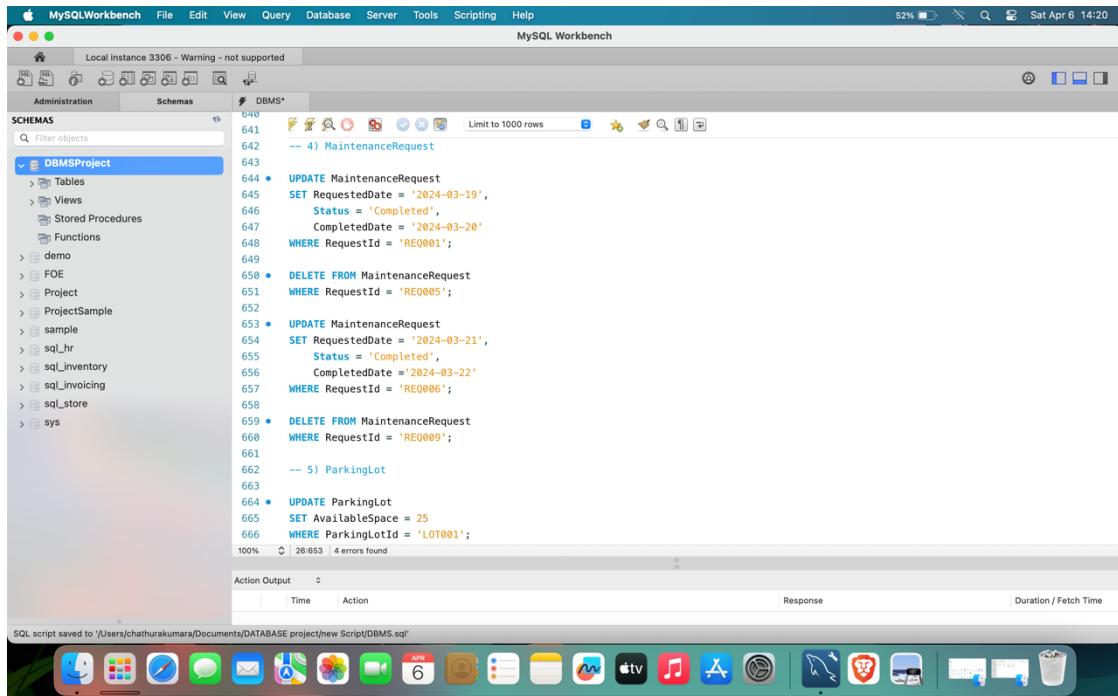
## ❖ Parking Lot



The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The title bar reads "MySQL Workbench" and "Local instance 3306 - Warning - not supported". The main window displays a SQL script titled "DBMS\*". The script contains several SQL statements, mostly UPDATE and DELETE queries, related to the "ParkingLot" and "ParkingSpace" tables. The code is numbered from 655 to 688. A message at the bottom indicates "4 errors found". Below the script, there's a table header for "Action Output" with columns "Time", "Action", "Response", and "Duration / Fetch Time". The status bar at the bottom shows "SQL script saved to '/Users/chathurakumara/Documents/DATABASE project/new Script/DBMS.sql'".

```
655
656 WHERE RequestId = 'REQ006';
657
658 • DELETE FROM MaintenanceRequest
659 WHERE RequestId = 'REQ009';
660
661 -- 5) ParkingLot
662
663
664 • UPDATE ParkingLot
665 SET AvailableSpace = 25
666 WHERE ParkingLotId = 'LOT001';
667
668 • DELETE FROM ParkingLot
669 WHERE ParkingLotId = 'LOT004';
670
671 • UPDATE ParkingLot
672 SET TotalSpace = 200
673 WHERE ParkingLotId = 'LOT005';
674
675 • DELETE FROM ParkingLot
676 WHERE ParkingLotId = 'LOT007';
677
678 -- 6) ParkingSpace
679
680 • UPDATE ParkingSpace
681 SET Status = 'Occupied'
```

## ❖ Maintenance Request

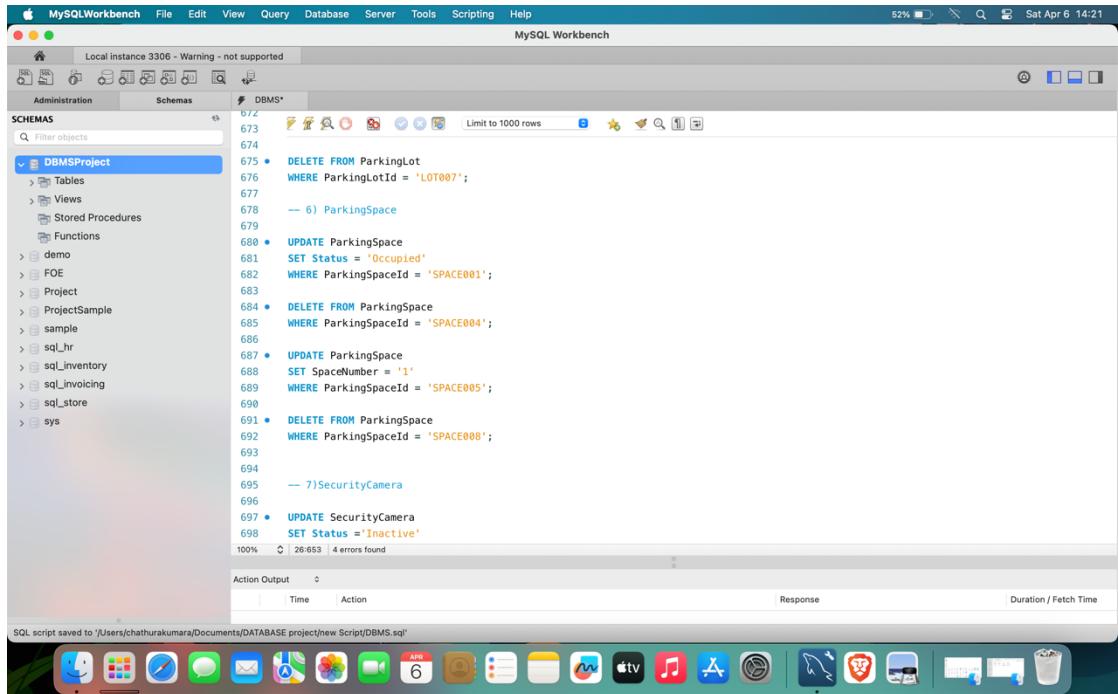


The screenshot shows the MySQL Workbench interface with the title bar "MySQL Workbench" and the status bar "Sat Apr 6 14:20". The main window displays a SQL editor with the following code:

```
640
641
642 --- 4) MaintenanceRequest
643
644 • UPDATE MaintenanceRequest
645     SET RequestedDate = '2024-03-19',
646         Status = 'Completed',
647         CompletedDate = '2024-03-20'
648     WHERE RequestId = 'REQ001';
649
650 • DELETE FROM MaintenanceRequest
651     WHERE RequestId = 'REQ005';
652
653 • UPDATE MaintenanceRequest
654     SET RequestedDate = '2024-03-21',
655         Status = 'Completed',
656         CompletedDate = '2024-03-22'
657     WHERE RequestId = 'REQ006';
658
659 • DELETE FROM MaintenanceRequest
660     WHERE RequestId = 'REQ009';
661
662 --- 5) ParkingLot
663
664 • UPDATE ParkingLot
665     SET AvailableSpace = 25
666     WHERE ParkingLotId = 'LOT001';
100% 26:653 | 4 errors found
```

The SQL script was saved to "/Users/chathurakumara/Documents/DATABASE project/new Script/DBMS.sql".

## ❖ Parking Space

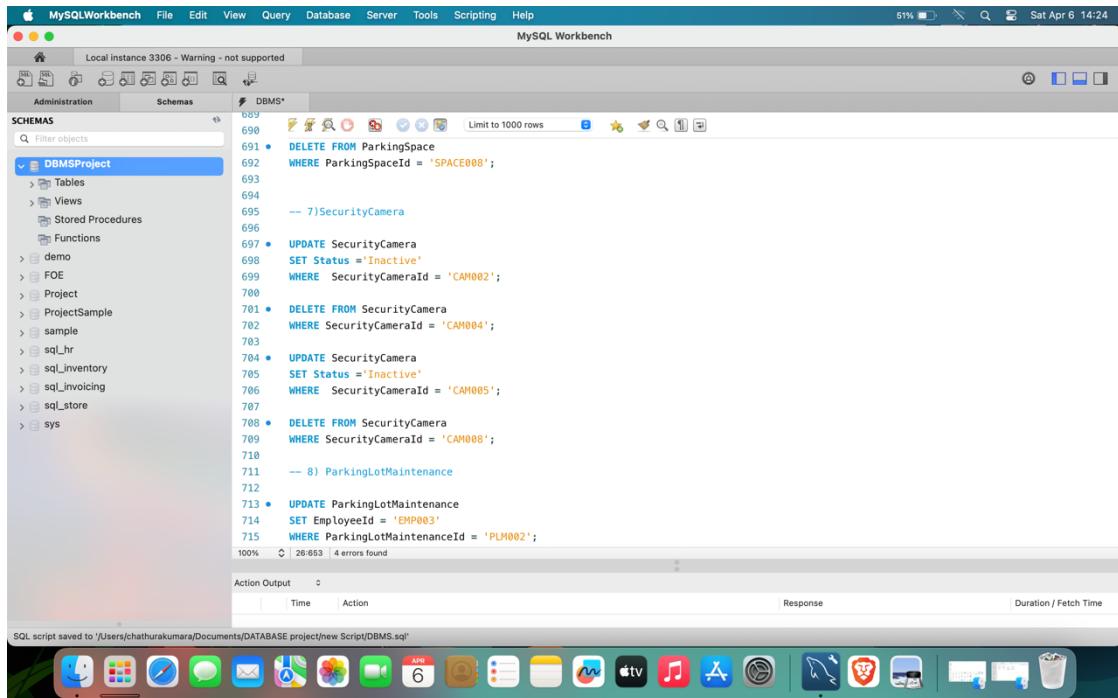


The screenshot shows the MySQL Workbench interface with the title bar "MySQL Workbench" and the status bar "Sat Apr 6 14:21". The main window displays a SQL editor with the following code:

```
673
674
675 • DELETE FROM ParkingLot
676     WHERE ParkingLotId = 'LOT007';
677
678 --- 6) ParkingSpace
679
680 • UPDATE ParkingSpace
681     SET Status = 'Occupied'
682     WHERE ParkingSpaceId = 'SPACE001';
683
684 • DELETE FROM ParkingSpace
685     WHERE ParkingSpaceId = 'SPACE004';
686
687 • UPDATE ParkingSpace
688     SET SpaceNumber = '1'
689     WHERE ParkingSpaceId = 'SPACE005';
690
691 • DELETE FROM ParkingSpace
692     WHERE ParkingSpaceId = 'SPACE008';
693
694
695 --- 7) SecurityCamera
696
697 • UPDATE SecurityCamera
698     SET Status = 'Inactive'
```

The SQL script was saved to "/Users/chathurakumara/Documents/DATABASE project/new Script/DBMS.sql".

## ❖ Security Camera

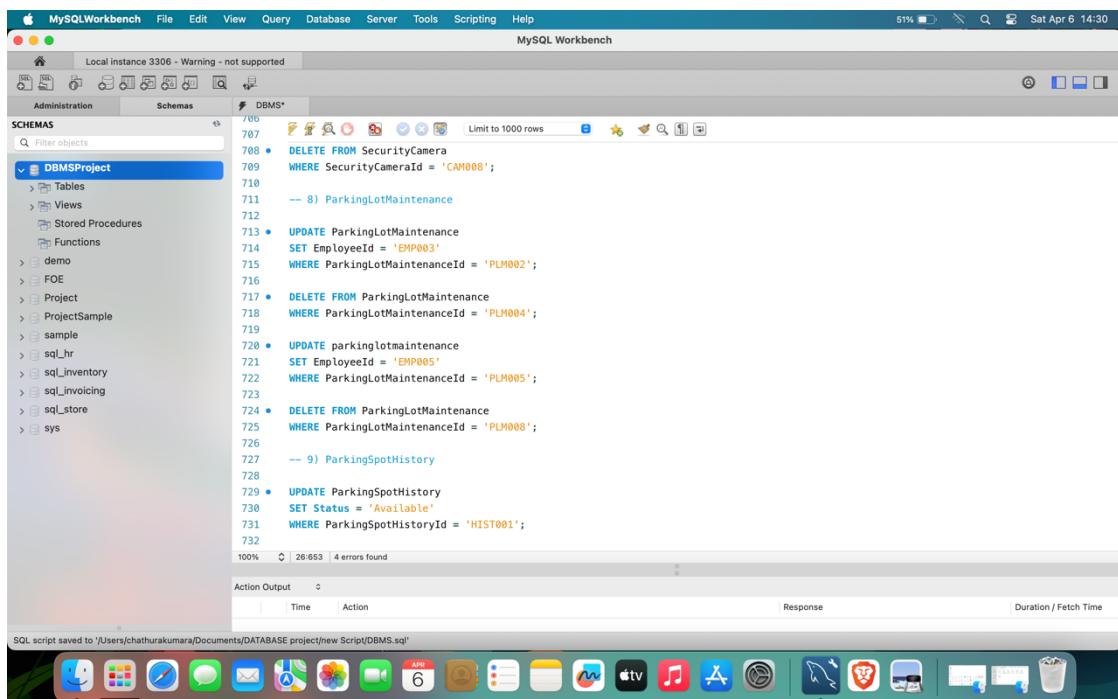


The screenshot shows the MySQL Workbench interface with the following SQL script:

```
689 690 691 • DELETE FROM ParkingSpace  
WHERE ParkingSpaceId = 'SPACE008';  
692  
693  
694 -- 7)SecurityCamera  
695  
696 697 • UPDATE SecurityCamera  
SET Status ='Inactive'  
WHERE SecurityCameraId = 'CAM002';  
698  
699  
700 • DELETE FROM SecurityCamera  
WHERE SecurityCameraId = 'CAM004';  
701  
702  
703 704 • UPDATE SecurityCamera  
SET Status ='Inactive'  
WHERE SecurityCameraId = 'CAM005';  
705  
706  
707 708 • DELETE FROM SecurityCamera  
WHERE SecurityCameraId = 'CAM008';  
709  
710  
711 -- 8) ParkingLotMaintenance  
712  
713 • UPDATE ParkingLotMaintenance  
SET EmployeeId = 'EMP003'  
WHERE ParkingLotMaintenanceId = 'PLM002';  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729 • UPDATE ParkingSpotHistory  
SET Status = 'Available'  
WHERE ParkingSpotHistoryId = 'HIST001';  
730  
731  
732
```

The script includes several DELETE and UPDATE statements for tables like ParkingSpace, SecurityCamera, and ParkingLotMaintenance. It also contains comments indicating sections for Security Camera (lines 694-695) and Parking Lot Maintenance (line 711).

## ❖ Parking Lot Maintenance

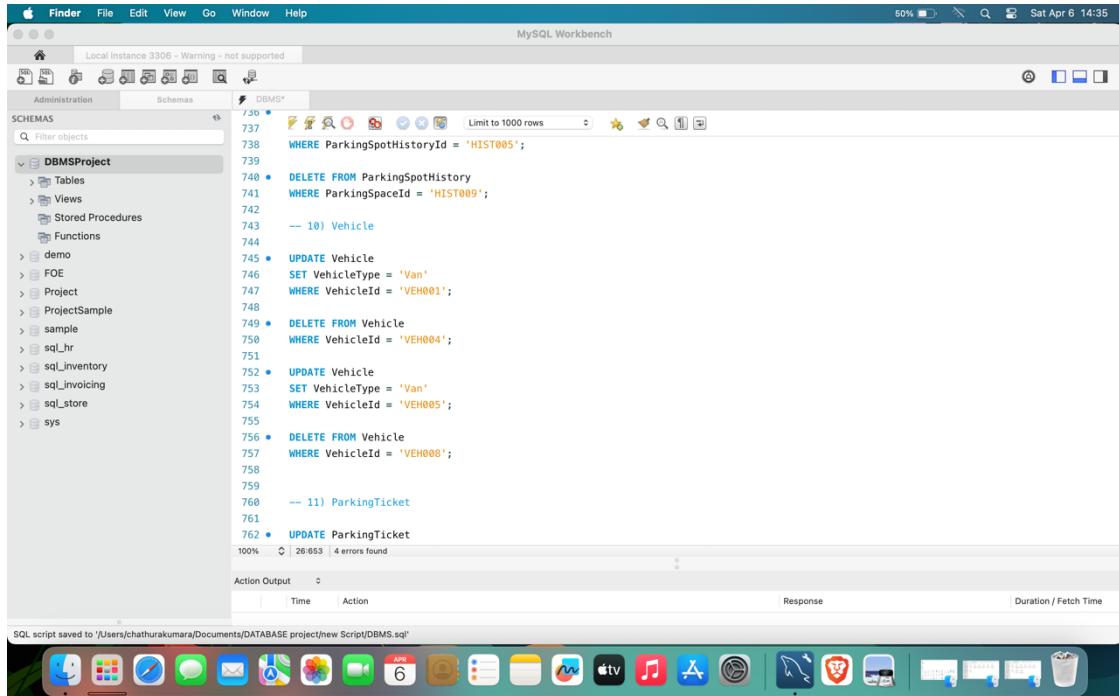


The screenshot shows the MySQL Workbench interface with the following SQL script:

```
700 701 702 703 704 705 706 707 708 • DELETE FROM SecurityCamera  
WHERE SecurityCameraId = 'CAM008';  
709  
710  
711 -- 8) ParkingLotMaintenance  
712  
713 • UPDATE ParkingLotMaintenance  
SET EmployeeId = 'EMP003'  
WHERE ParkingLotMaintenanceId = 'PLM002';  
714  
715  
716  
717 • DELETE FROM ParkingLotMaintenance  
WHERE ParkingLotMaintenanceId = 'PLM004';  
718  
719  
720 • UPDATE parkinglotmaintenance  
SET EmployeeId = 'EMP005'  
WHERE ParkingLotMaintenanceId = 'PLM005';  
721  
722  
723  
724 • DELETE FROM ParkingLotMaintenance  
WHERE ParkingLotMaintenanceId = 'PLM008';  
725  
726  
727 -- 9) ParkingSpotHistory  
728  
729 • UPDATE ParkingSpotHistory  
SET Status = 'Available'  
WHERE ParkingSpotHistoryId = 'HIST001';  
730  
731  
732
```

This script continues the theme of maintenance, specifically for parking lots. It includes several DELETE and UPDATE statements for the ParkingLotMaintenance table, along with a section for ParkingSpotHistory (lines 727-732).

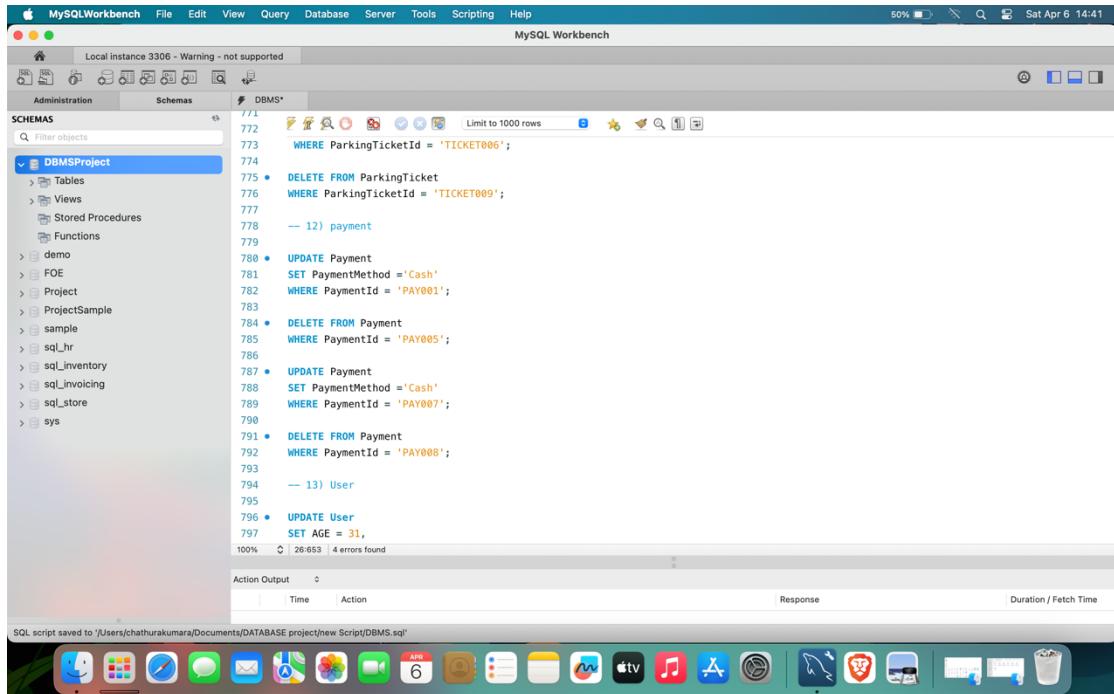
## ❖ Vehicle



The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The title bar reads "MySQL Workbench" and "Sat Apr 6 14:35". The main window displays a SQL editor with the following code:

```
730 • WHERE ParkingSpotHistoryId = 'HIST005';
731
732 • DELETE FROM ParkingSpotHistory
733 WHERE ParkingSpaceId = 'HIST009';
734
735 -- 10) Vehicle
736
737 • UPDATE Vehicle
738 SET VehicleType = 'Van'
739 WHERE VehicleId = 'VEH001';
740
741 • DELETE FROM Vehicle
742 WHERE VehicleId = 'VEH004';
743
744 • UPDATE Vehicle
745 SET VehicleType = 'Van'
746 WHERE VehicleId = 'VEH005';
747
748 • DELETE FROM Vehicle
749 WHERE VehicleId = 'VEH008';
750
751
752 -- 11) ParkingTicket
753
754 • UPDATE ParkingTicket
755 WHERE VehicleId = 'VEH005';
756
757 -- 11) ParkingTicket
758
759
760 -- 12) payment
761
762 • UPDATE Payment
763 SET PaymentMethod = 'Cash'
764 WHERE PaymentId = 'PAYMENT001';
765
766 • DELETE FROM Payment
767 WHERE PaymentId = 'PAYMENT005';
768
769 • UPDATE Payment
770 SET PaymentMethod = 'Credit'
771 WHERE PaymentId = 'PAYMENT006';
772
773 • DELETE FROM Payment
774 WHERE PaymentId = 'PAYMENT009';
775
776 -- 12) payment
777
778 • UPDATE Payment
779 SET PaymentMethod = 'Cash'
780 WHERE PaymentId = 'PAYMENT001';
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2819
2820
2821
```

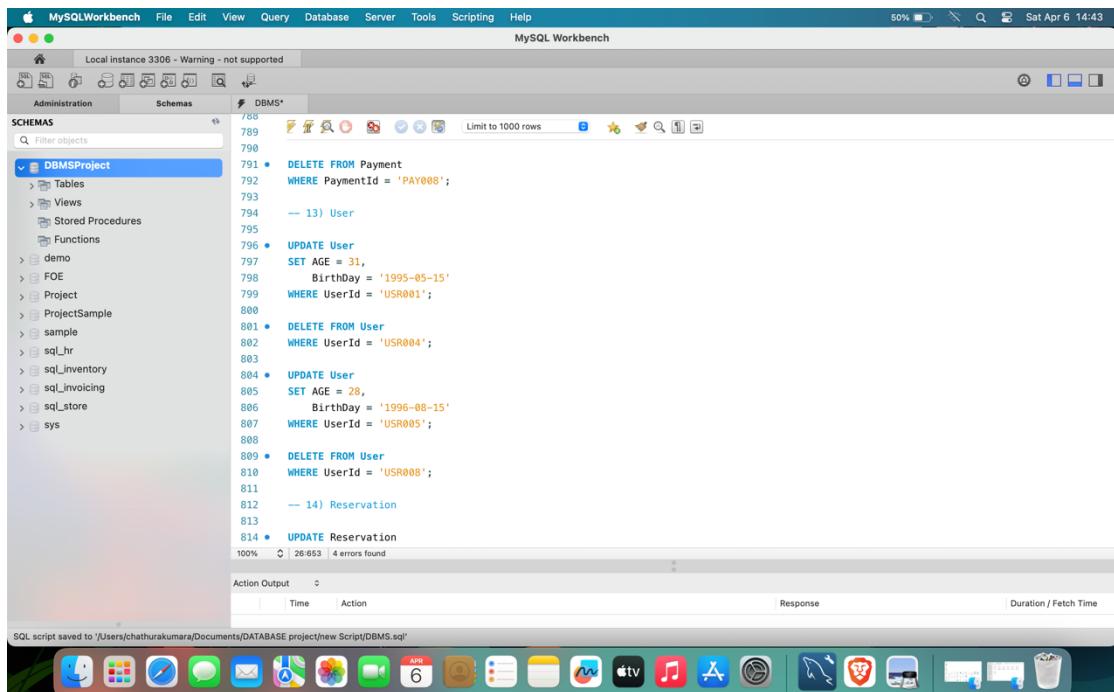
## ❖ Payment



The screenshot shows the MySQL Workbench interface with a script editor containing SQL code. The code is organized into sections: 1) Deletion of parking tickets, 2) Updates to the Payment table, and 3) Updates to the User table. The code includes various WHERE clauses and SET statements. A message at the bottom indicates 4 errors found.

```
772 WHERE ParkingTicketId = 'TICKET006';
773
774 • DELETE FROM ParkingTicket
775 WHERE ParkingTicketId = 'TICKET009';
776
777 -- 12) payment
778
779
780 • UPDATE Payment
781 SET PaymentMethod = 'Cash'
782 WHERE PaymentId = 'PAY001';
783
784 • DELETE FROM Payment
785 WHERE PaymentId = 'PAY005';
786
787 • UPDATE Payment
788 SET PaymentMethod = 'Cash'
789 WHERE PaymentId = 'PAY007';
790
791 • DELETE FROM Payment
792 WHERE PaymentId = 'PAY008';
793
794 -- 13) User
795
796 • UPDATE User
797 SET AGE = 31,
100% 26-653 | 4 errors found
```

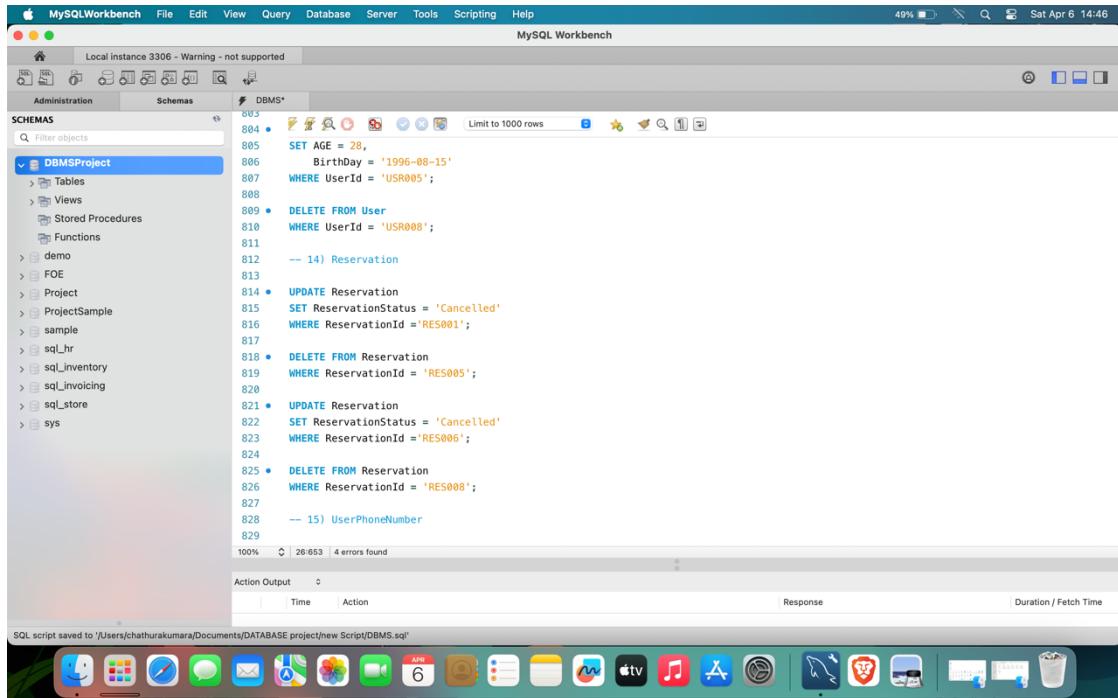
## ❖ User



The screenshot shows the MySQL Workbench interface with a script editor containing SQL code. The code is organized into sections: 1) Deletion from the Payment table, 2) Updates to the User table (including BirthDay), 3) Deletion from the User table, 4) Updates to the User table (including BirthDay), and 5) Deletion from the Reservation table. The code includes various WHERE clauses and SET statements. A message at the bottom indicates 4 errors found.

```
789 WHERE PaymentId = 'PAY008';
790
791 • DELETE FROM Payment
792 WHERE PaymentId = 'PAY008';
793
794 -- 13) User
795
796 • UPDATE User
797 SET AGE = 31,
798 BirthDay = '1995-05-15'
799 WHERE UserId = 'USR001';
800
801 • DELETE FROM User
802 WHERE UserId = 'USR004';
803
804 • UPDATE User
805 SET AGE = 28,
806 BirthDay = '1996-08-15'
807 WHERE UserId = 'USR005';
808
809 • DELETE FROM User
810 WHERE UserId = 'USR008';
811
812 -- 14) Reservation
813
814 • UPDATE Reservation
100% 26-653 | 4 errors found
```

## ❖ Reservation

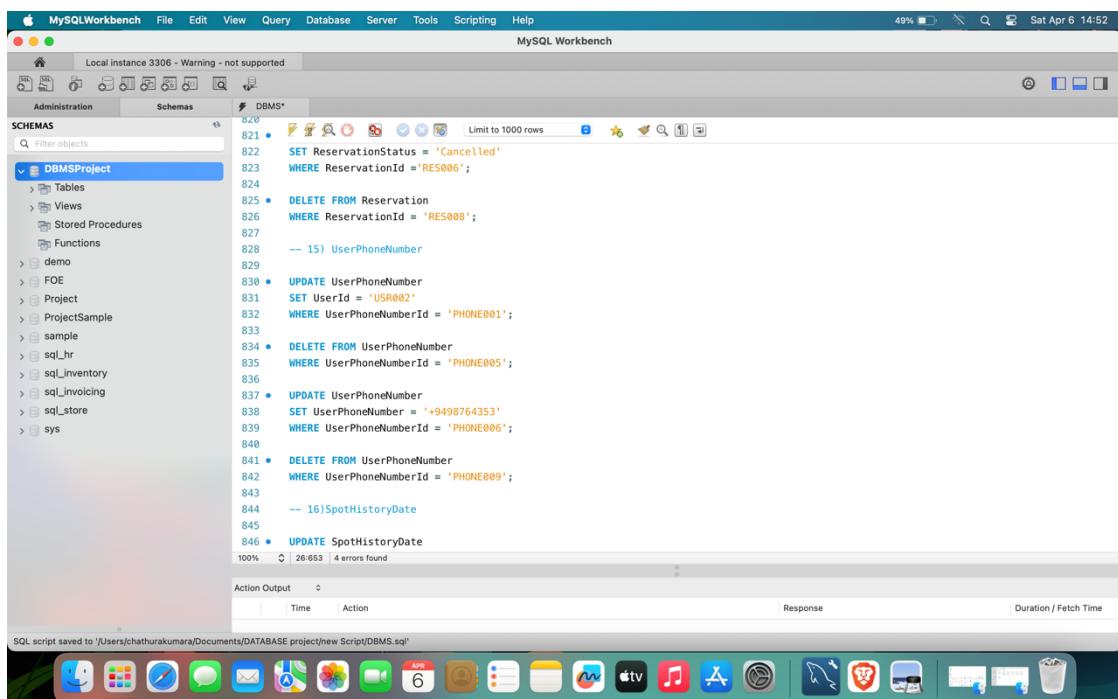


The screenshot shows the MySQL Workbench interface with the following SQL script:

```
803
804 • SET AGE = 28,
805     BirthDay = '1996-08-15'
806     WHERE UserId = 'USR005';
807
808 • DELETE FROM User
809     WHERE UserId = 'USR008';
810
811 -- 14) Reservation
812
813
814 • UPDATE Reservation
815     SET ReservationStatus = 'Cancelled'
816     WHERE ReservationId = 'RES001';
817
818 • DELETE FROM Reservation
819     WHERE ReservationId = 'RES005';
820
821 • UPDATE Reservation
822     SET ReservationStatus = 'Cancelled'
823     WHERE ReservationId = 'RES006';
824
825 • DELETE FROM Reservation
826     WHERE ReservationId = 'RES008';
827
828 -- 15) UserPhoneNumber
829
```

SQL script saved to '/Users/chathurakumara/Documents/DATABASE project/new Script/DBMS.sql'

## ❖ Phone Number

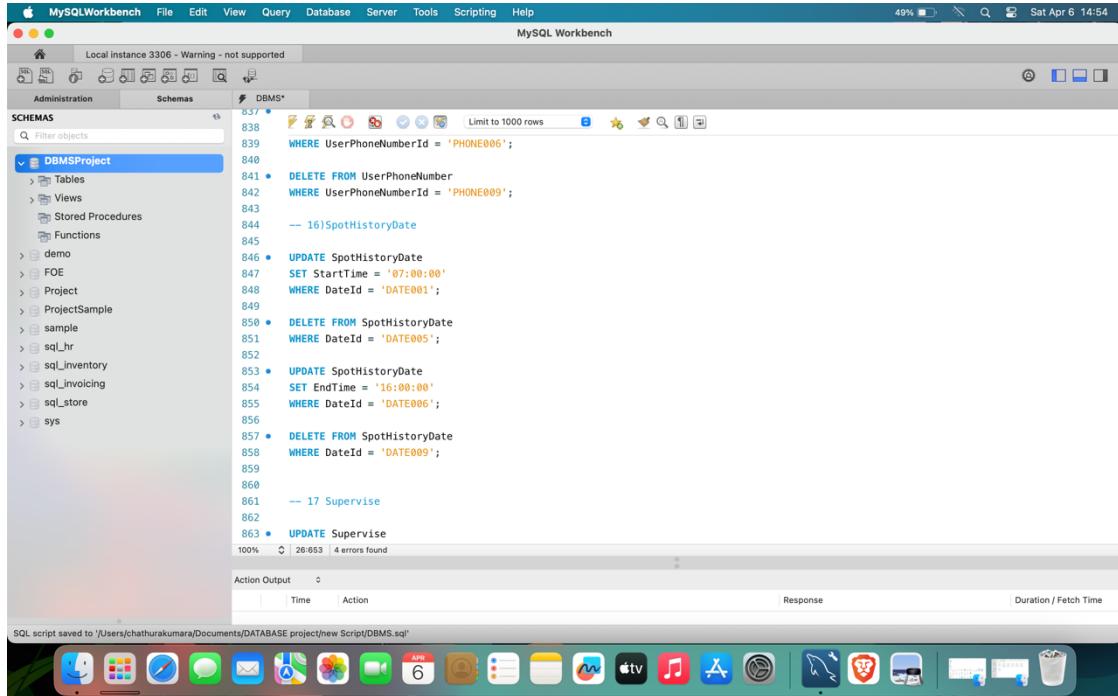


The screenshot shows the MySQL Workbench interface with the following SQL script:

```
820
821 • SET ReservationStatus = 'Cancelled'
822     WHERE ReservationId = 'RES006';
823
824
825 • DELETE FROM Reservation
826     WHERE ReservationId = 'RES008';
827
828 -- 15) UserPhoneNumber
829
830 • UPDATE UserPhoneNumber
831     SET UserId = 'USR002'
832     WHERE UserPhoneNumberId = 'PHONE001';
833
834 • DELETE FROM UserPhoneNumber
835     WHERE UserPhoneNumberId = 'PHONE005';
836
837 • UPDATE UserPhoneNumber
838     SET UserPhoneNumber = '+9498764353'
839     WHERE UserPhoneNumberId = 'PHONE006';
840
841 • DELETE FROM UserPhoneNumber
842     WHERE UserPhoneNumberId = 'PHONE009';
843
844 -- 16) SpotHistoryDate
845
846 • UPDATE SpotHistoryDate
```

SQL script saved to '/Users/chathurakumara/Documents/DATABASE project/new Script/DBMS.sql'

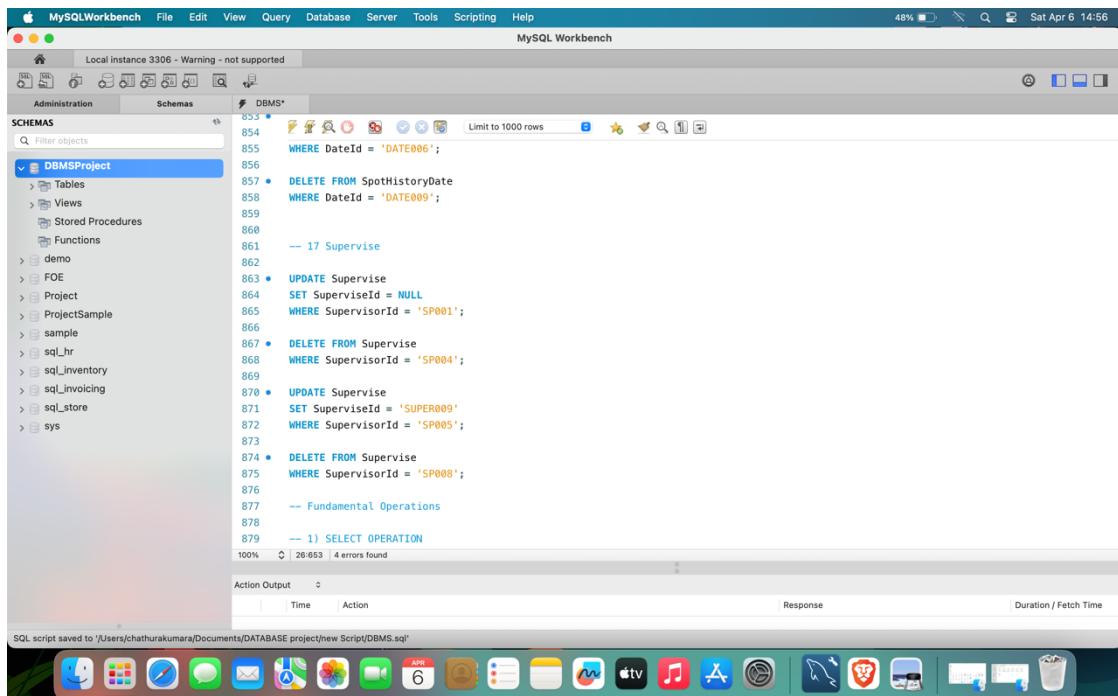
## ❖ SpotHistoryDate



The screenshot shows the MySQL Workbench interface with the title bar "MySQL Workbench" and the status bar "Sat Apr 6 14:54". The main window displays a SQL script in the "DBMS\*" tab. The script is for a schema named "DBMSProject" and includes several DELETE and UPDATE statements for tables like "UserPhoneNumber" and "SpotHistoryDate". The code is color-coded for syntax highlighting. The bottom status bar indicates "100%" completion and "26:653 | 4 errors found". The Mac OS X Dock is visible at the bottom.

```
837 • WHERE UserPhoneNumberId = 'PHONE006';
838
839 WHERE UserPhoneNumberId = 'PHONE005';
840
841 • DELETE FROM UserPhoneNumber
842 WHERE UserPhoneNumberId = 'PHONE005';
843
844 --- 16)SpotHistoryDate
845
846 • UPDATE SpotHistoryDate
847 SET StartTime = '07:00:00'
848 WHERE DateId = 'DATE001';
849
850 • DELETE FROM SpotHistoryDate
851 WHERE DateId = 'DATE005';
852
853 • UPDATE SpotHistoryDate
854 SET EndTime = '16:00:00'
855 WHERE DateId = 'DATE006';
856
857 • DELETE FROM SpotHistoryDate
858 WHERE DateId = 'DATE009';
859
860
861 --- 17 Supervise
862
863 • UPDATE Supervise
100% 26:653 | 4 errors found
```

## ❖ Supervise

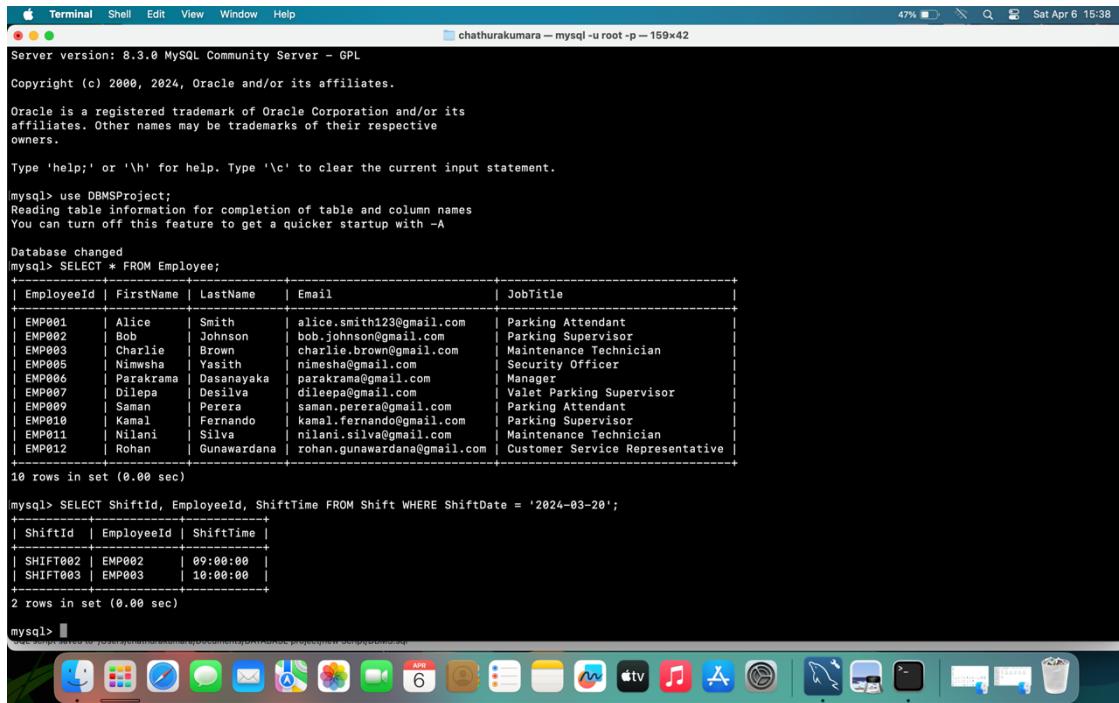


The screenshot shows the MySQL Workbench interface with the title bar "MySQL Workbench" and the status bar "Sat Apr 6 14:56". The main window displays a SQL script in the "DBMS\*" tab. The script is for a schema named "DBMSProject" and includes several DELETE and UPDATE statements for tables like "SpotHistoryDate" and "Supervise". The code is color-coded for syntax highlighting. The bottom status bar indicates "100%" completion and "26:653 | 4 errors found". The Mac OS X Dock is visible at the bottom.

```
853 • WHERE DateId = 'DATE006';
854
855 WHERE DateId = 'DATE009';
856
857 • DELETE FROM SpotHistoryDate
858 WHERE DateId = 'DATE009';
859
860
861 --- 17 Supervise
862
863 • UPDATE Supervise
864 SET SupervisorId = NULL
865 WHERE SupervisorId = 'SP001';
866
867 • DELETE FROM Supervise
868 WHERE SupervisorId = 'SP004';
869
870 • UPDATE Supervise
871 SET SupervisorId = 'SUPER009'
872 WHERE SupervisorId = 'SP005';
873
874 • DELETE FROM Supervise
875 WHERE SupervisorId = 'SP008';
876
877 --- Fundamental Operations
878
879 --- 1) SELECT OPERATION
100% 26:653 | 4 errors found
```

# Chapter 4: Transaction

## ❖ Select Operation



A screenshot of a macOS desktop environment. In the center is a Terminal window titled "chathurakumara -- mysql -u root -p - 159x42". The window displays MySQL command-line interface output. The user has connected to a database named "DBMSProject". They run a query to select all columns from the "Employee" table, which returns 10 rows of data. Then, they run a query to select ShiftId, EmployeeId, and ShiftTime for shifts on March 20, 2024, which returns 2 rows of data. The MySQL prompt "mysql>" appears at the bottom of the terminal window. The desktop bar at the bottom shows various application icons.

```
Terminal Shell Edit View Window Help
chathurakumara -- mysql -u root -p - 159x42
Server version: 8.3.0 MySQL Community Server - GPL
Copyright (c) 2000, 2024, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

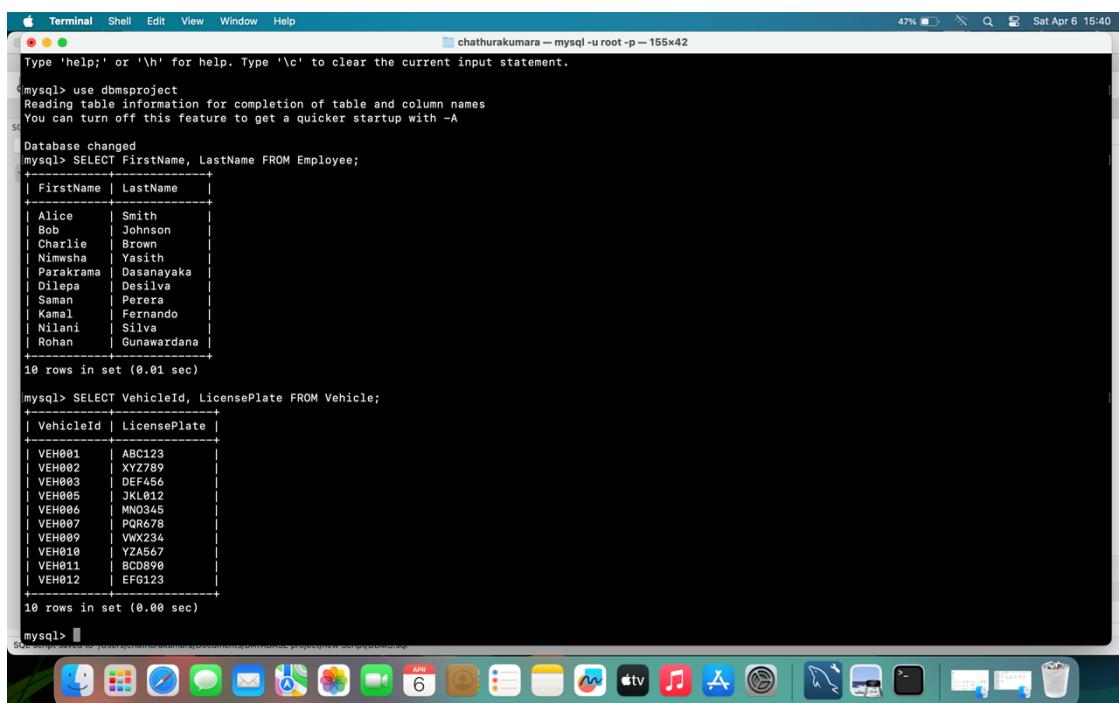
mysql> use DBMSProject;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT * FROM Employee;
+-----+-----+-----+-----+
| EmployeeId | FirstName | LastName | Email | JobTitle
+-----+-----+-----+-----+
| EMP001 | Alice | Smith | alice.smith123@gmail.com | Parking Attendant
| EMP002 | Bob | Johnson | bob.johnson@gmail.com | Parking Supervisor
| EMP003 | Charlie | Brown | charlie.brown@gmail.com | Maintenance Technician
| EMP005 | Nimwshe | Yasith | nimwshe@gmail.com | Security Officer
| EMP006 | Parakkrama | Dasanayaka | parakkrama@gmail.com | Manager
| EMP007 | Dilepa | Desilva | dileepa@gmail.com | Valet Parking Supervisor
| EMP009 | Saman | Perera | saman.perera@gmail.com | Parking Attendant
| EMP10 | Kamal | Fernando | kamal.fernando@gmail.com | Parking Supervisor
| EMP011 | Nilani | Silva | nilani.silva@gmail.com | Maintenance Technician
| EMP012 | Rohan | Gunawardana | rohan.gunawardana@gmail.com | Customer Service Representative
+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql> SELECT ShiftId, EmployeeId, ShiftTime FROM Shift WHERE ShiftDate = '2024-03-20';
+-----+-----+-----+
| ShiftId | EmployeeId | ShiftTime
+-----+-----+-----+
| SHIFT002 | EMP002 | 09:00:00
| SHIFT003 | EMP003 | 10:00:00
+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

## ❖ Project Operation



A screenshot of a macOS desktop environment. In the center is a Terminal window titled "chathurakumara -- mysql -u root -p - 155x42". The window displays MySQL command-line interface output. The user has connected to a database named "dbmsproject". They run a query to select FirstName and LastName from the "Employee" table, which returns 10 rows of data. Then, they run a query to select VehicleId and LicensePlate from the "Vehicle" table, which returns 12 rows of data. The MySQL prompt "mysql>" appears at the bottom of the terminal window. The desktop bar at the bottom shows various application icons.

```
Terminal Shell Edit View Window Help
chathurakumara -- mysql -u root -p - 155x42
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use dbmsproject;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT FirstName, LastName FROM Employee;
+-----+-----+
| FirstName | LastName
+-----+-----+
| Alice | Smith
| Bob | Johnson
| Charlie | Brown
| Nimwshe | Yasith
| Parakkrama | Dasanayaka
| Dilepa | Desilva
| Saman | Perera
| Kamal | Fernando
| Nilani | Silva
| Rohan | Gunawardana
+-----+-----+
10 rows in set (0.01 sec)

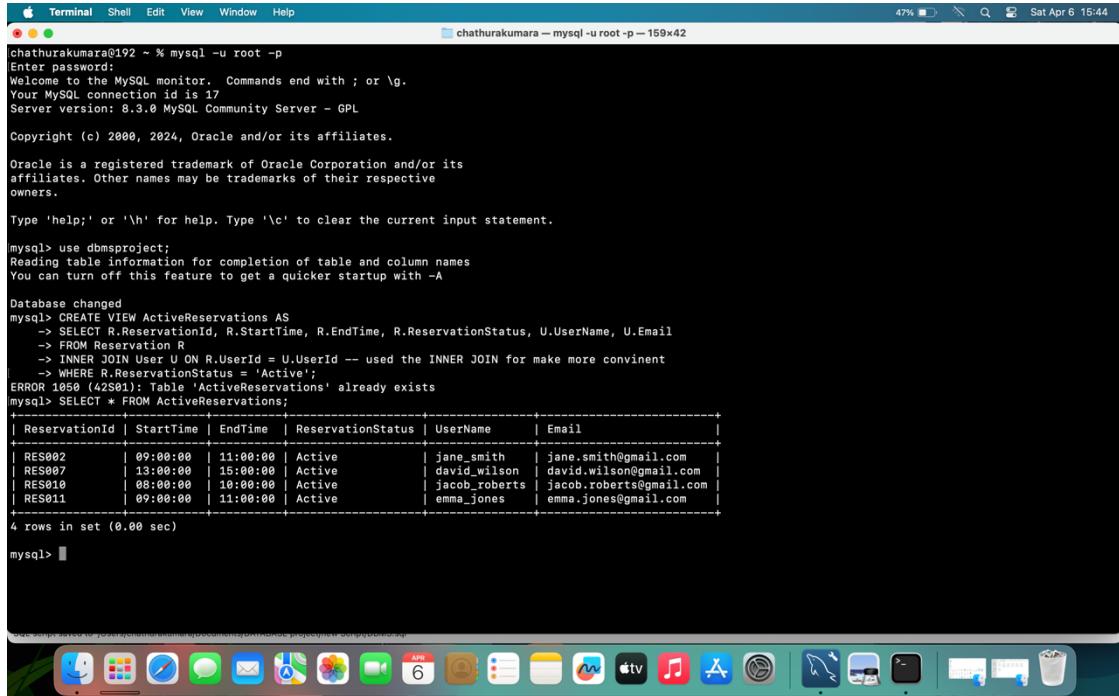
mysql> SELECT VehicleId, LicensePlate FROM Vehicle;
+-----+-----+
| VehicleId | LicensePlate
+-----+-----+
| VEH001 | ABC123
| VEH002 | XYZ789
| VEH003 | DEF456
| VEH005 | JK1012
| VEH006 | MNO345
| VEH007 | PQR678
| VEH009 | VWX234
| VEH10 | YZA567
| VEH11 | BCD890
| VEH12 | EFG123
+-----+-----+
12 rows in set (0.00 sec)

mysql>
```

#### ❖ Cartesian product 1

#### ❖ Cartesian product 2

## ❖ Creating a user view



```
chathurakumara@192 ~ % mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 17
Server version: 8.3.0 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

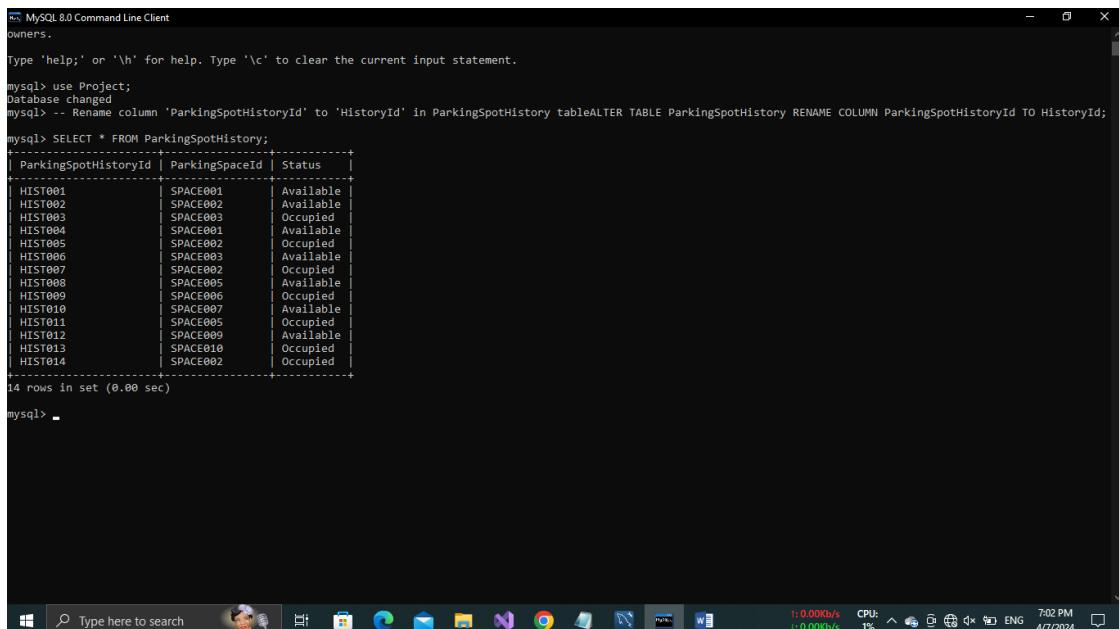
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use dbmsproject;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> CREATE VIEW ActiveReservations AS
    >     SELECT R.ReservationId, R.StartTime, R.EndTime, R.ReservationStatus, U.UserName, U.Email
    >     FROM Reservation R
    >     INNER JOIN User U ON R.UserId = U.UserId -- used the INNER JOIN for make more convinient
    >     WHERE R.ReservationStatus = 'Active';
ERROR 1050 (42S01): Table 'ActiveReservations' already exists
mysql> SELECT * FROM ActiveReservations;
+-----+-----+-----+-----+-----+-----+
| ReservationId | StartTime | EndTime | ReservationStatus | UserName | Email
+-----+-----+-----+-----+-----+-----+
| RES002 | 09:00:00 | 11:00:00 | Active | jane_smith | jane.smith@gmail.com
| RES007 | 13:00:00 | 15:00:00 | Active | david_wilson | david.wilson@gmail.com
| RES010 | 08:00:00 | 10:00:00 | Active | jacob_roberts | jacob.roberts@gmail.com
| RES011 | 09:00:00 | 11:00:00 | Active | emma_jones | emma.jones@gmail.com
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

## ❖ Renaming Operation



```
MySQL 8.0 Command Line Client
owners.

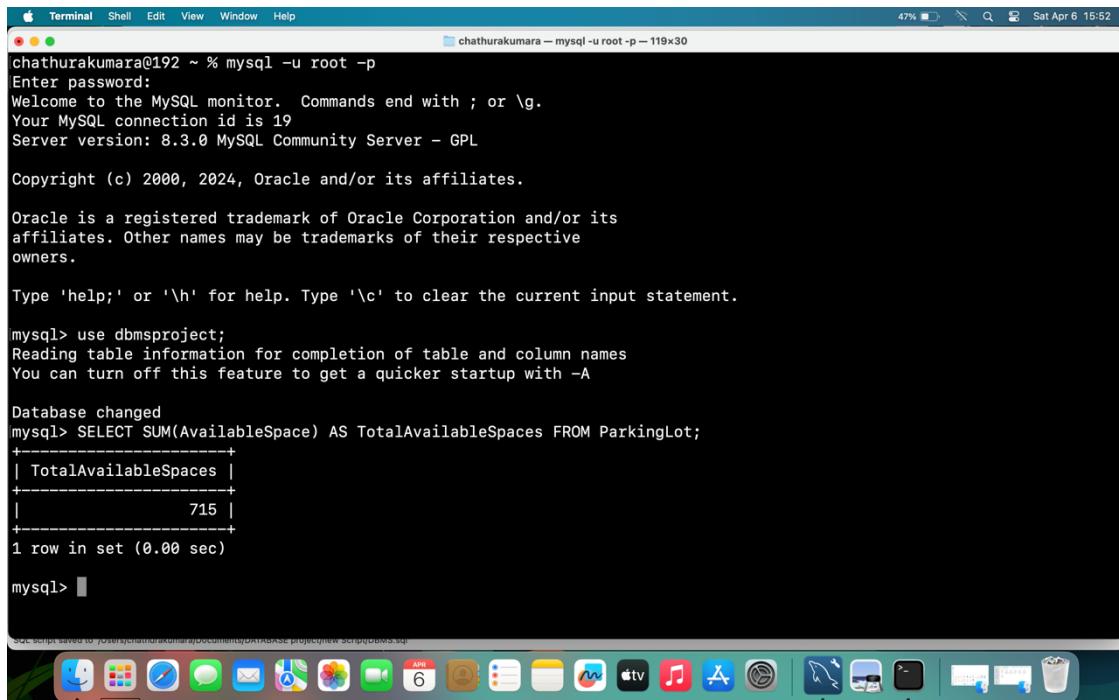
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use Project;
Database changed
mysql> -- Rename column 'ParkingSpotHistoryId' to 'HistoryId' in 'ParkingSpotHistory' table
ALTER TABLE ParkingSpotHistory RENAME COLUMN ParkingSpotHistoryId TO HistoryId;

mysql> SELECT * FROM ParkingSpotHistory;
+-----+-----+-----+
| ParkingSpotHistoryId | ParkingSpaceId | Status
+-----+-----+-----+
| HIST001 | SPACE001 | Available
| HIST002 | SPACE002 | Available
| HIST003 | SPACE003 | Occupied
| HIST004 | SPACE001 | Available
| HIST005 | SPACE002 | Occupied
| HIST006 | SPACE003 | Available
| HIST007 | SPACE002 | Occupied
| HIST008 | SPACE005 | Available
| HIST009 | SPACE004 | Occupied
| HIST010 | SPACE007 | Available
| HIST011 | SPACE005 | Occupied
| HIST012 | SPACE009 | Available
| HIST013 | SPACE010 | Occupied
| HIST014 | SPACE002 | Occupied
+-----+-----+-----+
14 rows in set (0.00 sec)

mysql>
```

❖ Aggregation (Sum function)



```
Terminal Shell Edit View Window Help
chathurakumara@192 ~ % mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 19
Server version: 8.3.0 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

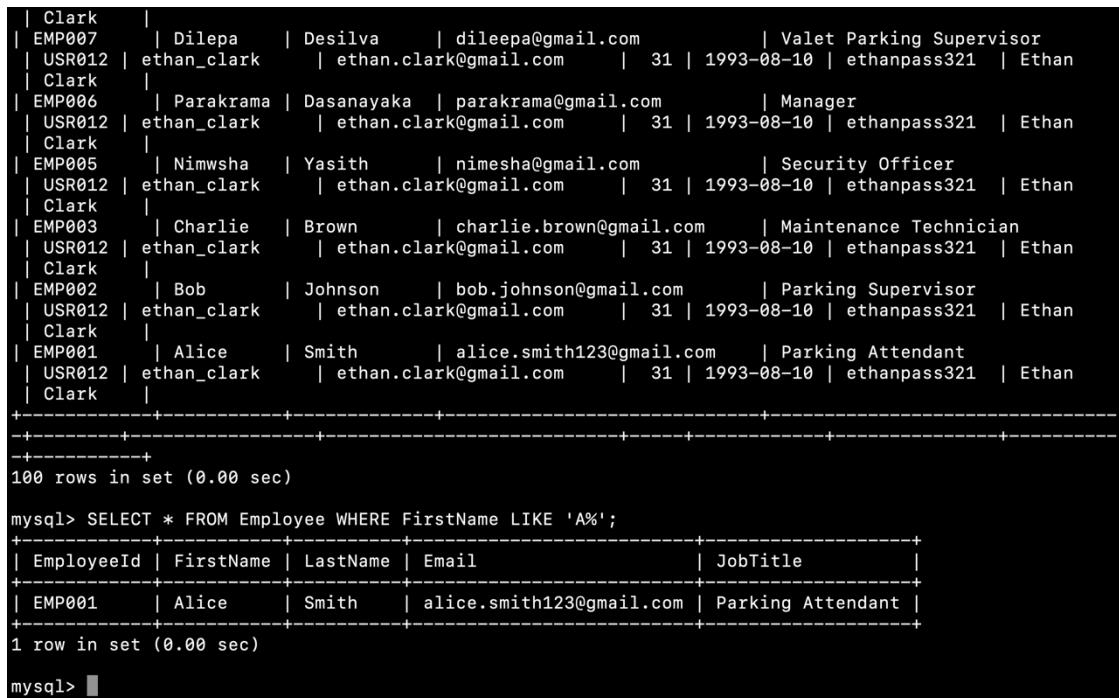
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use dbmsproject;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT SUM(AvailableSpace) AS TotalAvailableSpaces FROM ParkingLot;
+-----+
| TotalAvailableSpaces |
+-----+
| 715 |
+-----+
1 row in set (0.00 sec)

mysql>
```

❖ Like operator 1



```
| Clark      |
| EMP007    | Dilepa     | Desilva      | dileepa@gmail.com          | Valet Parking Supervisor
| USR012   | ethan_clark | ethan.clark@gmail.com | 31 | 1993-08-10 | ethanpass321 | Ethan
| Clark      |
| EMP006    | Parakrama   | Dasanayaka   | parakrama@gmail.com        | Manager
| USR012   | ethan_clark | ethan.clark@gmail.com | 31 | 1993-08-10 | ethanpass321 | Ethan
| Clark      |
| EMP005    | Nimwsha     | Yasith       | nimesha@gmail.com         | Security Officer
| USR012   | ethan_clark | ethan.clark@gmail.com | 31 | 1993-08-10 | ethanpass321 | Ethan
| Clark      |
| EMP003    | Charlie     | Brown        | charlie.brown@gmail.com  | Maintenance Technician
| USR012   | ethan_clark | ethan.clark@gmail.com | 31 | 1993-08-10 | ethanpass321 | Ethan
| Clark      |
| EMP002    | Bob          | Johnson      | bob.johnson@gmail.com   | Parking Supervisor
| USR012   | ethan_clark | ethan.clark@gmail.com | 31 | 1993-08-10 | ethanpass321 | Ethan
| Clark      |
| EMP001    | Alice        | Smith        | alice.smith123@gmail.com | Parking Attendant
| USR012   | ethan_clark | ethan.clark@gmail.com | 31 | 1993-08-10 | ethanpass321 | Ethan
| Clark      |
+-----+
100 rows in set (0.00 sec)

mysql> SELECT * FROM Employee WHERE FirstName LIKE 'A%';
+-----+-----+-----+-----+-----+
| EmployeeId | FirstName | LastName | Email           | JobTitle      |
+-----+-----+-----+-----+-----+
| EMP001     | Alice     | Smith    | alice.smith123@gmail.com | Parking Attendant |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

## Complex Queries

### ❖ Union

```
Terminal Shell Edit View Window Help chathurakumara — mysql -u root -p — 119x30
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use dbmsproject;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT FirstName, LastName, JobTitle
    > FROM Employee
    > JOIN Shift ON Employee.EmployeeId = Shift.EmployeeId
    > WHERE ShiftDate = '2024-03-20'
    >
    > UNION
    >
    > SELECT FirstName, LastName, JobTitle
    > FROM Employee
    > JOIN Shift ON Employee.EmployeeId = Shift.EmployeeId
    > WHERE ShiftDate = '2024-03-27';
+-----+-----+-----+
| FirstName | LastName | JobTitle |
+-----+-----+-----+
| Bob       | Johnson  | Parking Supervisor |
| Charlie   | Brown    | Maintenance Technician |
| Nimwska   | Yasith   | Security Officer |
| Parakrama | Dasanayaka | Manager |
| Dilepa    | Desilva  | Valet Parking Supervisor |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

SQl script saved to /Users/chathurakumara/Documents/DBASE project/new Script/USERS3.sql

A screenshot of a Mac OS X desktop. A MySQL terminal window is open in the foreground, showing the results of a UNION query. The desktop background is a green landscape. The Dock at the bottom contains various application icons.

### ❖ Intersection

```
Terminal Shell Edit View Window Help chathurakumara — mysql -u root -p — 119x30
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use dbmsproject1
ERROR 1049 (42000): Unknown database 'dbmsproject1'
mysql> use dbmsproject;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

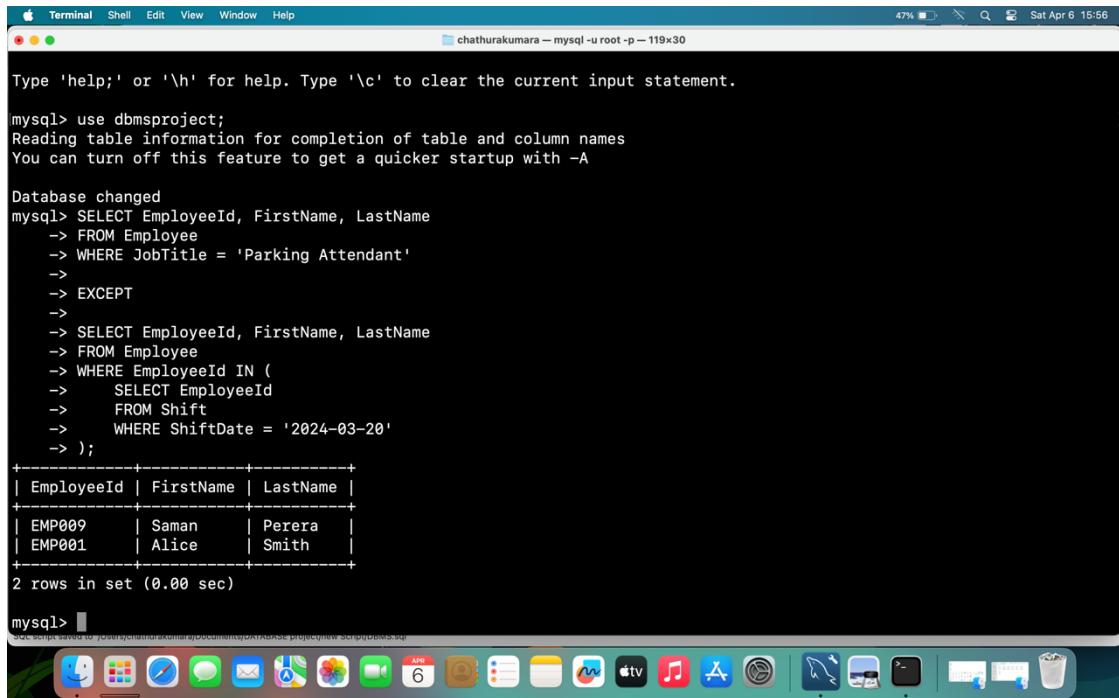
Database changed
mysql> SELECT EmployeeId, FirstName, LastName
    > FROM Employee
    > WHERE JobTitle = 'Parking Attendant'
    >
    > INTERSECT
    >
    > SELECT EmployeeId, FirstName, LastName
    > FROM Employee
    > WHERE EmployeeId IN (
    >     SELECT EmployeeId
    >     FROM Shift
    >     WHERE ShiftDate = '2024-03-28'
    > );
+-----+-----+-----+
| EmployeeId | FirstName | LastName |
+-----+-----+-----+
| EMP009     | Saman    | Perera  |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

SQl script saved to /Users/chathurakumara/Documents/DBASE project/new Script/USERS3.sql

A screenshot of a Mac OS X desktop. A MySQL terminal window is open in the foreground, showing the results of an intersection query using the INTERSECT operator. The desktop background is a green landscape. The Dock at the bottom contains various application icons.

## ❖ Set Difference



```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

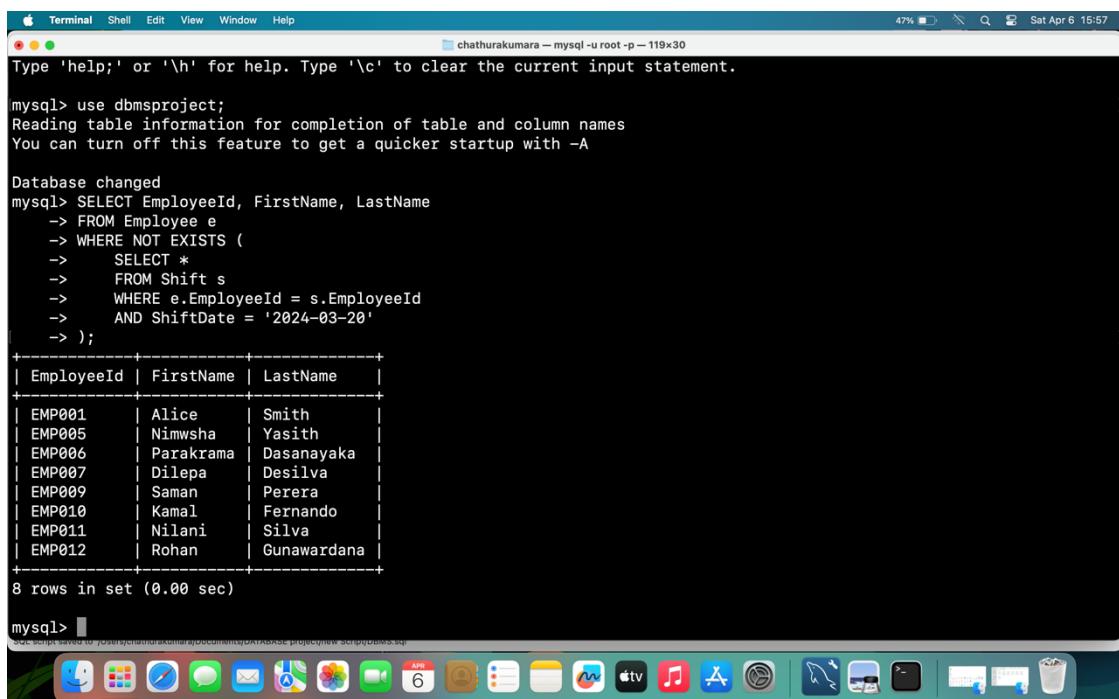
mysql> use dbmsproject;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT EmployeeId, FirstName, LastName
   > FROM Employee
   > WHERE JobTitle = 'Parking Attendant'
   >
   > EXCEPT
   >
   > SELECT EmployeeId, FirstName, LastName
   > FROM Employee
   > WHERE EmployeeId IN (
   >     SELECT EmployeeId
   >         FROM Shift
   >         WHERE ShiftDate = '2024-03-20'
   > );
+-----+-----+-----+
| EmployeeId | FirstName | LastName |
+-----+-----+-----+
| EMP009     | Saman    | Perera   |
| EMP001     | Alice    | Smith    |
+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

The screenshot shows a macOS desktop with a Terminal window open. The window title is "chathurakumara — mysql -u root -p — 119x30". The terminal displays a MySQL session. The user runs a query to select employees from the 'Employee' table where their job title is 'Parking Attendant'. This result is then combined with another query that selects employees from the 'Employee' table where their EmployeeID is present in the 'Shift' table for the date '2024-03-20'. The 'EXCEPT' keyword is used to remove the rows from the second query from the first, resulting in two rows: 'EMP009' (Saman, Perera) and 'EMP001' (Alice, Smith).

## ❖ Division



```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use dbmsproject;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT EmployeeId, FirstName, LastName
   > FROM Employee e
   > WHERE NOT EXISTS (
   >     SELECT *
   >         FROM Shift s
   >         WHERE e.EmployeeId = s.EmployeeId
   >         AND ShiftDate = '2024-03-20'
   > );
+-----+-----+-----+
| EmployeeId | FirstName | LastName |
+-----+-----+-----+
| EMP001     | Alice    | Smith   |
| EMP005     | Nimwsha  | Yasith  |
| EMP006     | Parakrama | Dasanayaka |
| EMP007     | Dilepa   | Desilva |
| EMP009     | Saman    | Perera  |
| EMP010     | Kamal    | Fernando |
| EMP011     | Nilani   | Silva   |
| EMP012     | Rohan    | Gunawardana |
+-----+-----+-----+
8 rows in set (0.00 sec)

mysql>
```

The screenshot shows a macOS desktop with a Terminal window open. The window title is "chathurakumara — mysql -u root -p — 119x30". The terminal displays a MySQL session. The user runs a query to select employees from the 'Employee' table. The query includes a subquery that checks if there is a corresponding entry in the 'Shift' table for each employee's ID and the date '2024-03-20'. The 'NOT EXISTS' clause ensures that only employees who do not have a shift on that specific date are selected. The result is eight rows of employees: EMP001 (Alice, Smith), EMP005 (Nimwsha, Yasith), EMP006 (Parakrama, Dasanayaka), EMP007 (Dilepa, Desilva), EMP009 (Saman, Perera), EMP010 (Kamal, Fernando), EMP011 (Nilani, Silva), and EMP012 (Rohan, Gunawardana).

❖ Inner join

```
Terminal Shell Edit View Window Help
chathurakumara — mysql -u root -p — 119x30
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use dbmsproject;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT e.EmployeeId, e.FirstName, e.LastName, s.ShiftTime, s.EndTime
   -> FROM Employee e
   -> INNER JOIN
   -> Shift s ON e.EmployeeId = s.EmployeeId;
+-----+-----+-----+-----+-----+
| EmployeeId | FirstName | LastName | ShiftTime | EndTime |
+-----+-----+-----+-----+-----+
| EMP001 | Alice | Smith | 08:00:00 | 16:00:00 |
| EMP002 | Bob | Johnson | 09:00:00 | 17:00:00 |
| EMP003 | Charlie | Brown | 10:00:00 | 18:00:00 |
| EMP003 | Charlie | Brown | 11:00:00 | 19:00:00 |
| EMP005 | Nimwsha | Yasith | 14:30:00 | 20:00:00 |
| EMP006 | Parakrama | Dasanayaka | 13:00:00 | 21:00:00 |
| EMP007 | Dilepa | Desilva | 14:00:00 | 22:00:00 |
| EMP009 | Saman | Perera | 08:00:00 | 16:00:00 |
| EMP010 | Kamal | Fernando | 09:00:00 | 17:00:00 |
| EMP011 | Nilani | Silva | 10:00:00 | 18:00:00 |
+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql>
```



❖ Natural join

```
Terminal Shell Edit View Window Help
chathurakumara — mysql -u root -p — 119x30
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use dbmsproject;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT e.EmployeeId, e.FirstName, e.LastName, s.ShiftTime, s.EndTime
   -> FROM Employee e
   -> NATURAL JOIN
   -> Shift s;
+-----+-----+-----+-----+-----+
| EmployeeId | FirstName | LastName | ShiftTime | EndTime |
+-----+-----+-----+-----+-----+
| EMP001 | Alice | Smith | 08:00:00 | 16:00:00 |
| EMP002 | Bob | Johnson | 09:00:00 | 17:00:00 |
| EMP003 | Charlie | Brown | 10:00:00 | 18:00:00 |
| EMP003 | Charlie | Brown | 11:00:00 | 19:00:00 |
| EMP005 | Nimwsha | Yasith | 14:30:00 | 20:00:00 |
| EMP006 | Parakrama | Dasanayaka | 13:00:00 | 21:00:00 |
| EMP007 | Dilepa | Desilva | 14:00:00 | 22:00:00 |
| EMP009 | Saman | Perera | 08:00:00 | 16:00:00 |
| EMP010 | Kamal | Fernando | 09:00:00 | 17:00:00 |
| EMP011 | Nilani | Silva | 10:00:00 | 18:00:00 |
+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql>
```



## ❖ Left Outer Join

```
Terminal Shell Edit View Window Help
chathurakumara — mysql -u root -p — 119x30
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use dbmsproject;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT e.EmployeeId, e.FirstName, e.LastName, s.ShiftTime, s.EndTime
   > FROM Employee e
   > LEFT OUTER JOIN
   > Shift s ON e.EmployeeId = s.EmployeeId;
+-----+-----+-----+-----+
| EmployeeId | FirstName | LastName | ShiftTime | EndTime |
+-----+-----+-----+-----+
| EMP001 | Alice | Smith | 08:00:00 | 16:00:00 |
| EMP002 | Bob | Johnson | 09:00:00 | 17:00:00 |
| EMP003 | Charlie | Brown | 10:00:00 | 18:00:00 |
| EMP003 | Charlie | Brown | 11:00:00 | 19:00:00 |
| EMP005 | Nimsinha | Yasith | 14:30:00 | 20:00:00 |
| EMP006 | Parakrama | Dasanayaka | 13:00:00 | 21:00:00 |
| EMP007 | Dilepa | Desilva | 14:00:00 | 22:00:00 |
| EMP009 | Saman | Perera | 08:00:00 | 16:00:00 |
| EMP010 | Kamal | Fernando | 09:00:00 | 17:00:00 |
| EMP011 | Nilani | Silva | 10:00:00 | 18:00:00 |
| EMP012 | Rohan | Gunawardana | NULL | NULL |
+-----+-----+-----+-----+
11 rows in set (0.00 sec)

mysql>
```

## ❖ Right outer join

```
Terminal Shell Edit View Window Help
chathurakumara — mysql -u root -p — 119x30
owners.

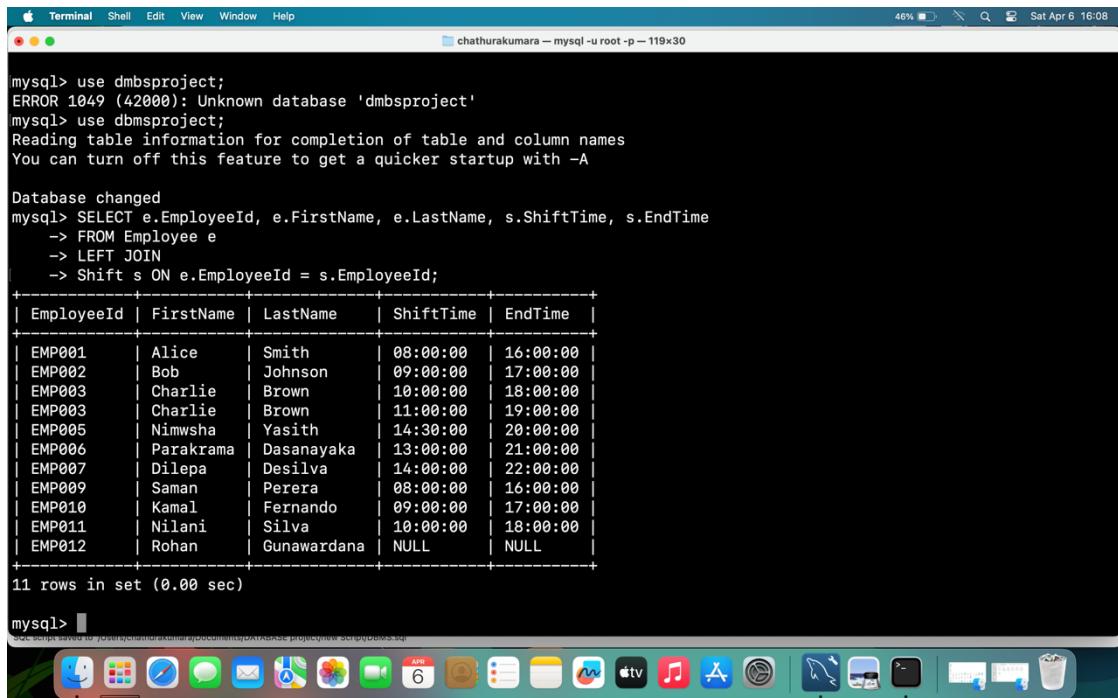
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use dbmsproject;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT e.EmployeeId, e.FirstName, e.LastName, s.ShiftTime, s.EndTime
   > FROM Employee e
   > RIGHT OUTER JOIN
   > Shift s ON e.EmployeeId = s.EmployeeId;
+-----+-----+-----+-----+
| EmployeeId | FirstName | LastName | ShiftTime | EndTime |
+-----+-----+-----+-----+
| EMP001 | Alice | Smith | 08:00:00 | 16:00:00 |
| EMP002 | Bob | Johnson | 09:00:00 | 17:00:00 |
| EMP003 | Charlie | Brown | 10:00:00 | 18:00:00 |
| EMP005 | Nimsinha | Yasith | 14:30:00 | 20:00:00 |
| EMP006 | Parakrama | Dasanayaka | 13:00:00 | 21:00:00 |
| EMP007 | Dilepa | Desilva | 14:00:00 | 22:00:00 |
| EMP009 | Saman | Perera | 08:00:00 | 16:00:00 |
| EMP010 | Kamal | Fernando | 09:00:00 | 17:00:00 |
| EMP011 | Nilani | Silva | 10:00:00 | 18:00:00 |
| EMP003 | Charlie | Brown | 11:00:00 | 19:00:00 |
+-----+-----+-----+-----+
10 rows in set (0.01 sec)

mysql>
```

❖ Full Outer join



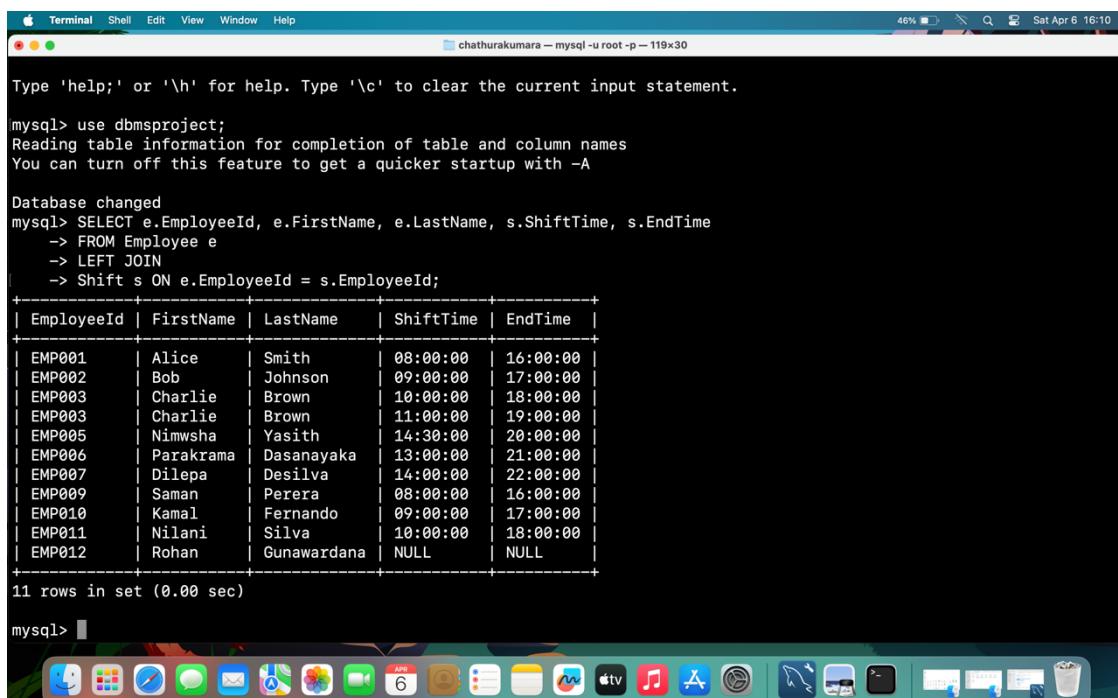
MySQL terminal window showing a full outer join query:

```
mysql> use dbmsproject;
ERROR 1049 (42000): Unknown database 'dbmsproject'
mysql> use dbmsproject;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT e.EmployeeId, e.FirstName, e.LastName, s.ShiftTime, s.EndTime
   > FROM Employee e
   > LEFT JOIN
   > Shift s ON e.EmployeeId = s.EmployeeId;
+-----+-----+-----+-----+-----+
| EmployeeId | FirstName | LastName | ShiftTime | EndTime |
+-----+-----+-----+-----+-----+
| EMP001 | Alice | Smith | 08:00:00 | 16:00:00 |
| EMP002 | Bob | Johnson | 09:00:00 | 17:00:00 |
| EMP003 | Charlie | Brown | 10:00:00 | 18:00:00 |
| EMP003 | Charlie | Brown | 11:00:00 | 19:00:00 |
| EMP005 | Nimwsha | Yasith | 14:30:00 | 20:00:00 |
| EMP006 | Parakrama | Dasanayaka | 13:00:00 | 21:00:00 |
| EMP007 | Dilepa | Desilva | 14:00:00 | 22:00:00 |
| EMP009 | Saman | Perera | 08:00:00 | 16:00:00 |
| EMP010 | Kamal | Fernando | 09:00:00 | 17:00:00 |
| EMP011 | Nilani | Silva | 10:00:00 | 18:00:00 |
| EMP012 | Rohan | Gunawardana | NULL | NULL |
+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)

mysql>
```

❖ Outer Join



MySQL terminal window showing an outer join query:

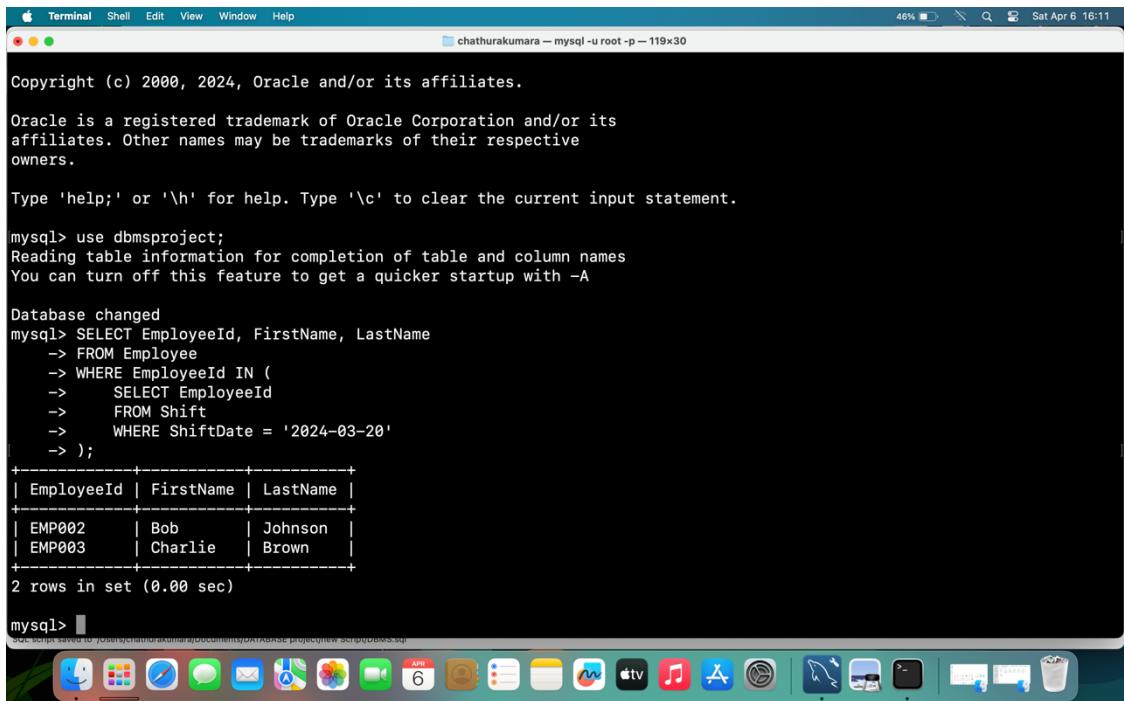
```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use dbmsproject;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT e.EmployeeId, e.FirstName, e.LastName, s.ShiftTime, s.EndTime
   > FROM Employee e
   > LEFT JOIN
   > Shift s ON e.EmployeeId = s.EmployeeId;
+-----+-----+-----+-----+-----+
| EmployeeId | FirstName | LastName | ShiftTime | EndTime |
+-----+-----+-----+-----+-----+
| EMP001 | Alice | Smith | 08:00:00 | 16:00:00 |
| EMP002 | Bob | Johnson | 09:00:00 | 17:00:00 |
| EMP003 | Charlie | Brown | 10:00:00 | 18:00:00 |
| EMP003 | Charlie | Brown | 11:00:00 | 19:00:00 |
| EMP005 | Nimwsha | Yasith | 14:30:00 | 20:00:00 |
| EMP006 | Parakrama | Dasanayaka | 13:00:00 | 21:00:00 |
| EMP007 | Dilepa | Desilva | 14:00:00 | 22:00:00 |
| EMP009 | Saman | Perera | 08:00:00 | 16:00:00 |
| EMP010 | Kamal | Fernando | 09:00:00 | 17:00:00 |
| EMP011 | Nilani | Silva | 10:00:00 | 18:00:00 |
| EMP012 | Rohan | Gunawardana | NULL | NULL |
+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)

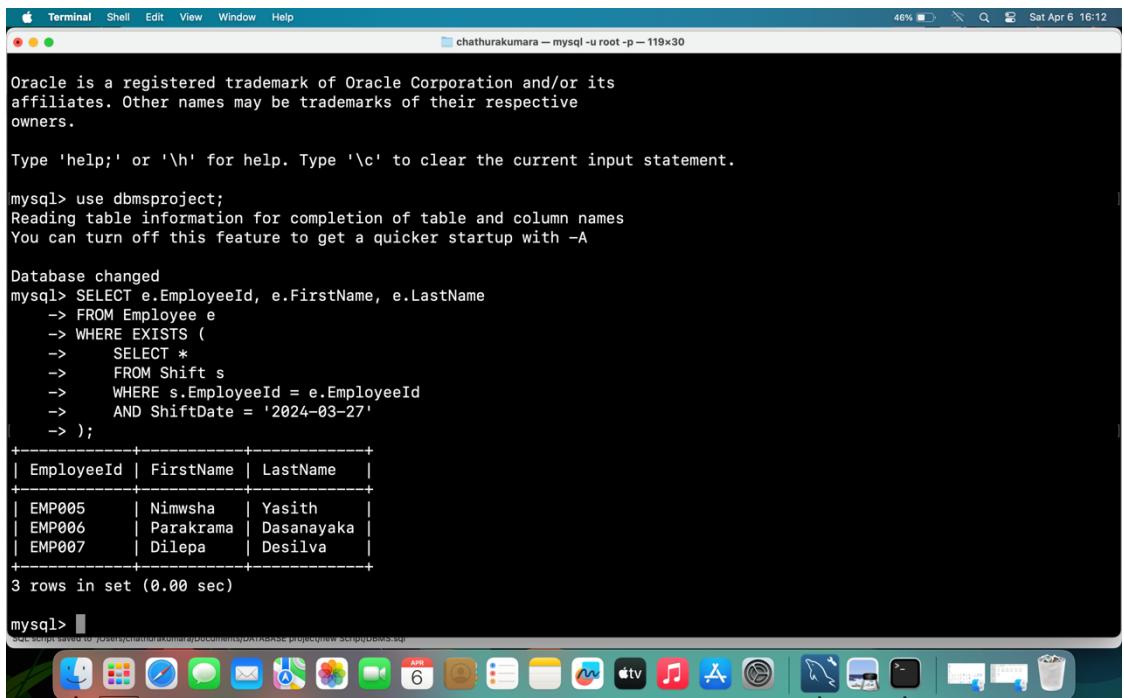
mysql>
```

## ❖ Independent query



Copyright (c) 2000, 2024, Oracle and/or its affiliates.  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql> use dbmsproject;  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
Database changed  
mysql> SELECT EmployeeId, FirstName, LastName  
 -> FROM Employee  
 -> WHERE EmployeeId IN (  
 -> SELECT EmployeeId  
 -> FROM Shift  
 -> WHERE ShiftDate = '2024-03-20'  
 -> );  
+-----+-----+-----+  
| EmployeeId | FirstName | LastName |  
+-----+-----+-----+  
| EMP002 | Bob | Johnson |  
| EMP003 | Charlie | Brown |  
+-----+-----+-----+  
2 rows in set (0.00 sec)  
mysql>

## ❖ Correlated Nested query.



Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql> use dbmsproject;  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
Database changed  
mysql> SELECT e.EmployeeId, e.FirstName, e.LastName  
 -> FROM Employee e  
 -> WHERE EXISTS (  
 -> SELECT \*  
 -> FROM Shift s  
 -> WHERE s.EmployeeId = e.EmployeeId  
 -> AND ShiftDate = '2024-03-27'  
 -> );  
+-----+-----+-----+  
| EmployeeId | FirstName | LastName |  
+-----+-----+-----+  
EMP005	Nimwsha	Yasith
EMP006	Parakrama	Dasanayaka
EMP007	Dilepa	Desilva
+-----+-----+-----+  
3 rows in set (0.00 sec)  
mysql>

## Chapter 5: Tuning

Query optimization involves assessing the efficacy of a query by analyzing the data accessed before and after implementing an index. When the data accessed diminishes post-index implementation, it indicates successful query tuning. All screenshots listed adhere to this standard. To optimize each challenging query, the following steps are undertaken to ensure comprehensive comprehension:

- Remove any externally built indexes that are currently present in utilized tables.
- After creating a proper index, display the number of accessed rows. (using EXPLAIN command)
- Display the number of rows accessed before creating a suitable index. (using EXPLAIN command)
- Display table indexes before creating a suitable index. (using SHOW INDEX command)
- Show table indexes after constructing an appropriate index. (using SHOW INDEX command)
- Union tuning

```
mysql> EXPLAIN
--> SELECT e.FirstName, e.LastName, s.ShiftTime, s.EndTime, pr.Name AS ParkingLotName
--> FROM Employee e, Shift s, MaintenanceRequest m, ParkingLot pr
--> WHERE
-->     e.EmployeeId = s.EmployeeId
-->     AND e.EmployeeId = m.EmployeeId
-->     AND m.ParkingLotId = pr.ParkingLotId
-->     AND e.JobTitle = 'Parking Attendant';
-->
--> UNION
-->
--> SELECT e.FirstName, e.LastName, s.ShiftTime, s.EndTime, pr.Name AS ParkingLotName
--> FROM Employee e, Shift s, MaintenanceRequest m, ParkingLot pr
--> WHERE
-->     e.EmployeeId = s.EmployeeId
-->     AND e.EmployeeId = m.EmployeeId
-->     AND m.ParkingLotId = pr.ParkingLotId
-->     AND e.JobTitle = 'Parking Supervisor';
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY | e | NULL | ALL | PRIMARY | NULL | NULL | NULL | 8 | 12.60 | Using where |
| 1 | PRIMARY | s | NULL | ref | FK_EmployeeId | FK_EmployeeId | 283 | dbmsproject.e.EmployeeId | 1 | 100.00 | NULL |
| 1 | PRIMARY | m | NULL | ref | ParkingLotId,fk_employeeeid | fk_employeeeid | 282 | dbmsproject.e.EmployeeId | 1 | 100.00 | NULL |
| 1 | PRIMARY | pr | NULL | eq_ref | PRIMARY | PRIMARY | 82 | dbmsproject.m.ParkingLotId | 1 | 100.00 | NULL |
| 2 | UNION | e | NULL | ALL | PRIMARY | NULL | NULL | NULL | 8 | 12.60 | NULL |
| 2 | UNION | s | NULL | ref | FK_EmployeeId | FK_EmployeeId | 283 | dbmsproject.e.EmployeeId | 1 | 100.00 | Using where |
| 2 | UNION | m | NULL | ref | ParkingLotId,fk_employeeeid | fk_employeeeid | 282 | dbmsproject.e.EmployeeId | 1 | 100.00 | NULL |
| 2 | UNION | pr | NULL | eq_ref | PRIMARY | PRIMARY | 82 | dbmsproject.m.ParkingLotId | 1 | 100.00 | NULL |
| 3 | UNION RESULT | <union1,2> | NULL | ALL | NULL | NULL | NULL | NULL | NULL | NULL | Using temporary |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
9 rows in set, 1 warning (0.00 sec)

mysql>
```

- Inner Join tuning

- Left outer join tuning

```
chathurakumara@192 ~ % mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 34
Server version: 8.3.0 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use dbmsproject;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> EXPLAIN
--> SELECT e.FirstName, e.LastName, s.ShiftTime, s.EndTime, s.ShiftDate
--> FROM Employee e
--> LEFT JOIN Shift s ON e.EmployeeId = s.EmployeeId;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE    | e     | NULL      | ALL  | NULL          | NULL | NULL    | NULL | 8   | 100.00  | NULL       |
| 1 | SIMPLE    | s     | NULL      | ALL  | FK_EmployeeId | NULL | NULL    | NULL | 16  | 100.00  | Using where; Using join buffer (hash join) |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql>
```

- Right outer join tuning

- Outer union tuning

```
Terminal Shell Edit View Window Help chathurakumara — mysql -u root -p — 204x51
chathurakumara@192 ~ % mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 37
Server version: 8.3.0 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use dbmsproject;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> EXPLAIN
--> SELECT ParkingLotId, Name, Location, OccupancyRate
--> FROM Parkinglot p1
--> WHERE EXISTS (
-->   SELECT 1
-->   FROM ParkingLot p2
-->   WHERE p2.ParkingLotId = p1.ParkingLotId
-->   GROUP BY p2.ParkingLotId
-->   HAVING MAX(p2.OccupancyRate) > 50
--> )
--> UNION ALL
--> SELECT ParkingLotId, Name, Location, OccupancyRate
--> FROM Parkinglot p3
--> WHERE NOT EXISTS (
-->   SELECT 1
-->   FROM ParkingLot p4
-->   WHERE p4.ParkingLotId = p3.ParkingLotId
-->   GROUP BY p4.ParkingLotId
-->   HAVING MAX(p4.OccupancyRate) > 50
--> );
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY | p1 | NULL | ALL | NULL | NULL | NULL | dbmsproject.p1.ParkingLotId | 10 | 100.00 | Using where |
| 2 | DEPENDENT SUBQUERY | p2 | NULL | eq_ref | PRIMARY | p2 | 82 | dbmsproject.p1.ParkingLotId | 1 | 100.00 | NULL |
| 3 | UNION | p3 | NULL | ALL | NULL | NULL | NULL | dbmsproject.p3.ParkingLotId | 10 | 100.00 | Using where |
| 4 | DEPENDENT SUBQUERY | p4 | NULL | eq_ref | PRIMARY | p4 | 82 | dbmsproject.p3.ParkingLotId | 1 | 100.00 | NULL |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set, 3 warnings (0.00 sec)

mysql> ||


```

- Division tuning

```
Terminal Shell Edit View Window Help chathurakumara — mysql -u root -p — 204x55
chathurakumara@192 ~ % mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 49
Server version: 8.3.0 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use dbmsproject;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW INDEX FROM Reservation;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_Index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Reservation | 0 | PRIMARY | 1 | ReservationId | A | 8 | NULL | NULL | BTREE | | | YES | NULL |
| Reservation | 1 | Userid | 1 | Userid | A | 8 | NULL | NULL | BTREE | | | YES | NULL |
| Reservation | 1 | fk_parking_space_id | 1 | ParkingSpaceId | A | 8 | NULL | NULL | BTREE | | | YES | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)

mysql> EXPLAIN
--> SELECT DISTINCT r.ReservationId, r.UserId
--> FROM Reservation r
--> WHERE NOT EXISTS (
-->   SELECT ps.ParkingSpaceId
-->   FROM ParkingSpace ps
-->   WHERE NOT EXISTS (
-->     SELECT pt.ParkingTicketId
-->     FROM ParkingTicket pt
-->     WHERE pt.VehicleId = 'VEMB01'
-->     AND pt.ParkingLotId = 'LOT001'
-->     AND pt.ParkingLotId = ps.ParkingLotId
-->   )
--> )
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY | NULL | 10 | 100.00 | Impossible WHERE | | | | | | |
| 2 | SUBQUERY | ps | NULL | index | index | NULL | NULL | ps.ParkingLotId | 82 | 100.00 | Using index |
| 2 | SUBQUERY | <subquery> | NULL | eq_ref | <auto_distinct_key> | <auto_distinct_key> | 83 | dbmsproject.ps.ParkingLotId | 1 | 100.00 | Using where; Not exists |
| 3 | MATERIALIZED | pt | NULL | ref | fk_parkinglot_id,fk_vehicleid | fk_vehicleid | 202 | const | 1 | 100.00 | Using where |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set, 2 warnings (0.01 sec)

mysql> ||


```

- Correlated Nested query.

```
Terminal Shell Edit View Window Help chathurakumara -- mysql -u root -p - 204x51
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 43
Server version: 8.3.0 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type 'c' to clear the current input statement.

mysql> use dbmsproject;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> CREATE INDEX idx_user_id ON User(UserId);
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> CREATE INDEX idx_parking_space_id ON ParkingSpace(ParkingSpaceId);
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> EXPLAIN
--> SELECT DISTINCT r.ReservationId, r.UserId
-->   FROM Reservation r
--> WHERE NOT EXISTS (
-->   -->   SELECT ps.ParkingSpaceId
-->   -->     FROM ParkingSpace ps
-->   -->   WHERE NOT EXISTS (
-->   -->     SELECT pt.ParkingTicketId
-->   -->       FROM ParkingTicket pt
-->   -->      WHERE pt.VehicleId = 'VH0001'
-->   -->      AND pt.ParkingLotId = 'LOT001'
-->   -->      AND pt.ParkingLotId = ps.ParkingLotId
-->   );
--> );

+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY | NULL | Impossible WHERE | | | | | | | | |
| 2 | SUBQUERY | ps | NULL | index | NULL | ParkingLotId | 82 | NULL | 10 | 100.00 | Using index |
| 2 | SUBQUERY | <subquery> | NULL | eq_ref | <auto_distinct_key> | <auto_distinct_key> | 83 | dbmsproject.ps.ParkingLotId | 1 | 100.00 | Using where; Not exists |
| 3 | MATERIALIZED | pt | NULL | ref | fk_parkinglot_id,fk_vehicleid | fk_parkinglot_id,fk_vehicleid | 262 | const | 1 | 100.00 | Using where |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set, 2 warnings (0.01 sec)

mysql>
mysql> ||
```

- Intersection tuning

- Set difference tuning.

```

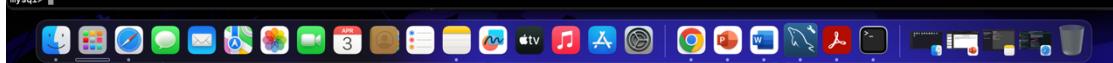
Terminal Shell Edit View Window Help
chathurakumara — mysql -u root -p — 204x61
Server version: 8.3.0 MySQL Community Server - GPL
Copyright (c) 2000, 2024, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> use dbmsproject;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed
mysql> EXPLAIN
--> SELECT EmployeeId, FirstName, LastName
-->   FROM Employee
-->  EXCEPT
-->    SELECT e.EmployeeId, e.FirstName, e.LastName
-->      FROM Employee e
-->     JOIN Shift s ON e.EmployeeId = s.EmployeeId;
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY | Employee | NULL | ALL | NULL | NULL | NULL | 283 | 100.00 | NULL | |
| 2 | EXCEPT | e | NULL | index | FK_EmployeeId | FK_EmployeeId | 283 | NULL | 1 | 100.00 | Using where; Using index |
| 3 | EXCEPT | e | NULL | eq_ref | PRIMARY, idx_employee_id | PRIMARY | 282 | <dbmsproject.s.EmployeeId | 1 | 100.00 | NULL |
| 4 | EXCEPT RESULT | <except1,2> | NULL | ALL | NULL | NULL | NULL | NULL | NULL | NULL | Using temporary |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set, 1 warning (0.00 sec)

mysql> SHOW INDEX FROM Employee;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Employee | 0 | PRIMARY | 1 | EmployeeId | A | 8 | NULL | NULL | BTREE | | YES | NULL |
| Employee | 1 | idx_employee_id | 1 | EmployeeId | A | 8 | NULL | NULL | BTREE | | YES | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> SHOW INDEX FROM Shift;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Shift | 0 | PRIMARY | 1 | ShiftId | A | 18 | NULL | NULL | BTREE | | YES | NULL |
| Shift | 1 | FK_EmployeeId | 1 | EmployeeId | A | 9 | NULL | NULL | YES | BTREE | | YES | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)

mysql> 

```



- Independent nested query Tuning.

```

Terminal Shell Edit View Window Help
chathurakumara@192 ~ % mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 44
Server version: 8.3.0 MySQL Community Server - GPL
Copyright (c) 2000, 2024, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> use dbmsproject;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed
mysql> EXPLAIN
--> SELECT DISTINCT e.FirstName, e.LastName
-->   FROM Employee e
--> WHERE e.EmployeeID IN (
-->   -->   SELECT DISTINCT m.EmployeeId
-->   -->     FROM MaintenanceRequest m
-->   -->     WHERE m.Status = 'Pending'
-->   );
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | <subquery2> | NULL | ALL | NULL | NULL | NULL | NULL | 100.00 | Using temporary | |
| 1 | SIMPLE | e | NULL | eq_ref | PRIMARY, idx_employee_id | PRIMARY | 202 | <subquery2>.EmployeeId | 1 | 100.00 | NULL |
| 2 | MATERIALIZED | m | NULL | ALL | fk_employeeid | NULL | NULL | NULL | 10 | 10.00 | Using where |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set, 1 warning (0.00 sec)

mysql> 

```

