



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

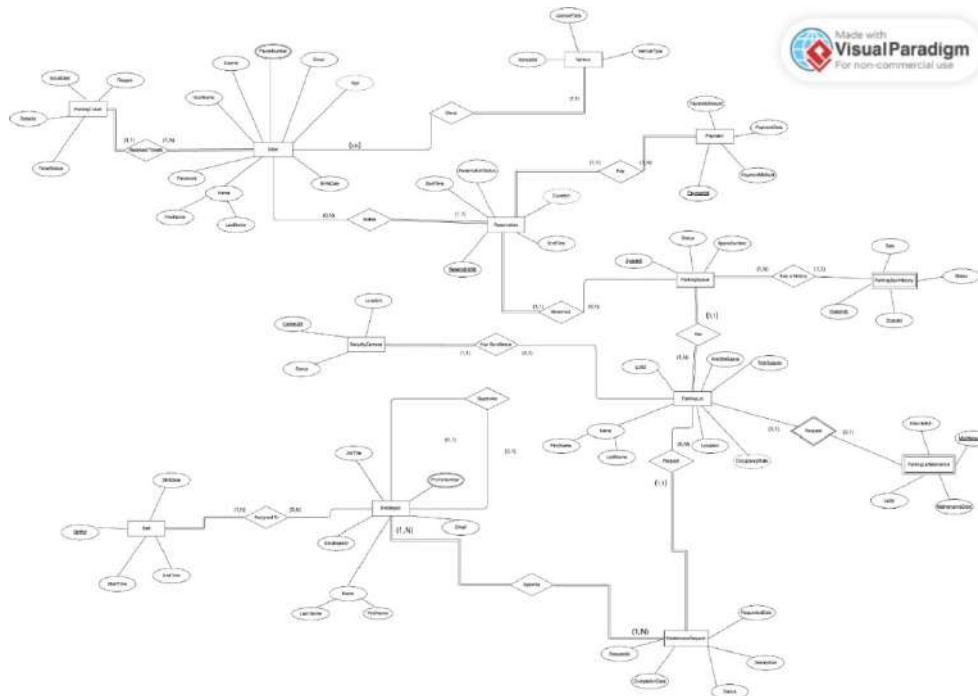


Figure 1: ER diagram of the Smart Parking Management System

2. UML Class Diagram of the Project

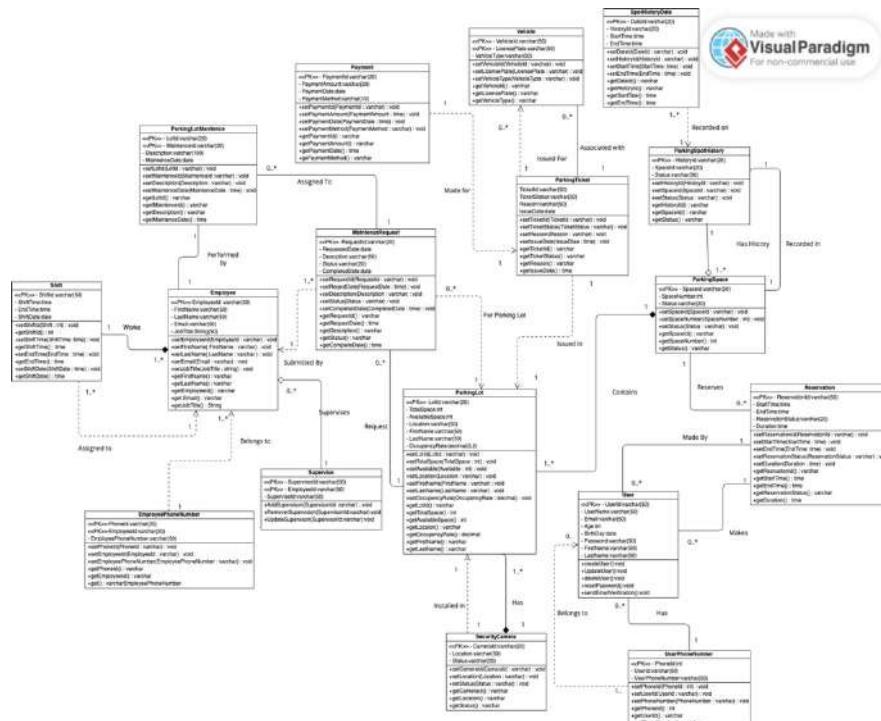


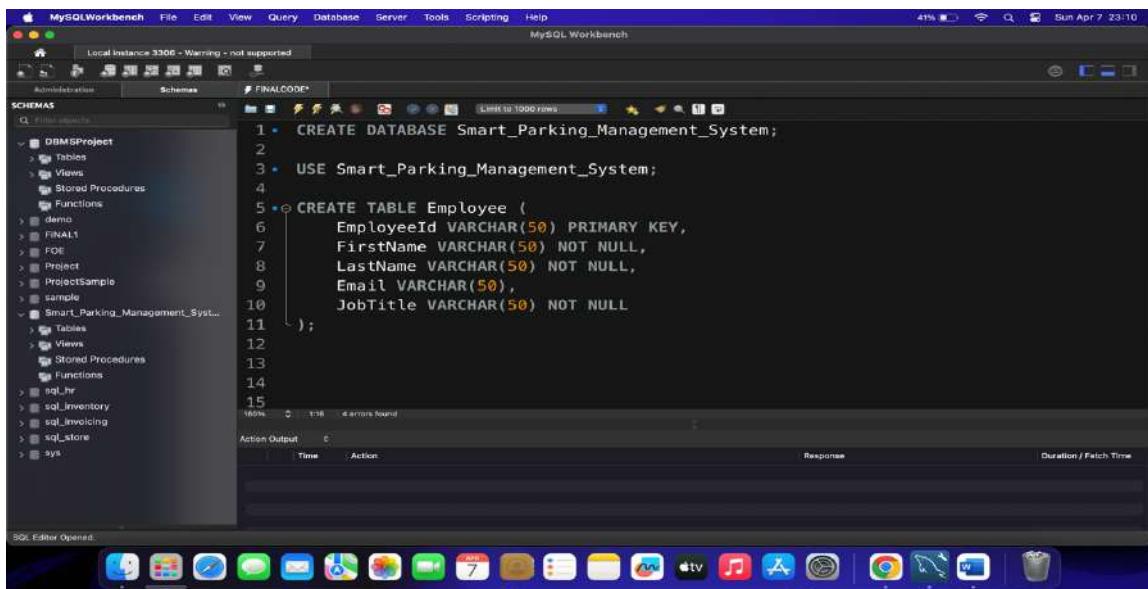
Figure 2: UML Class Diagram of the Smart Parking Management System

Chapter 3: Implementation

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

3.1 Table Implementation

- ❖ Create table Employee.

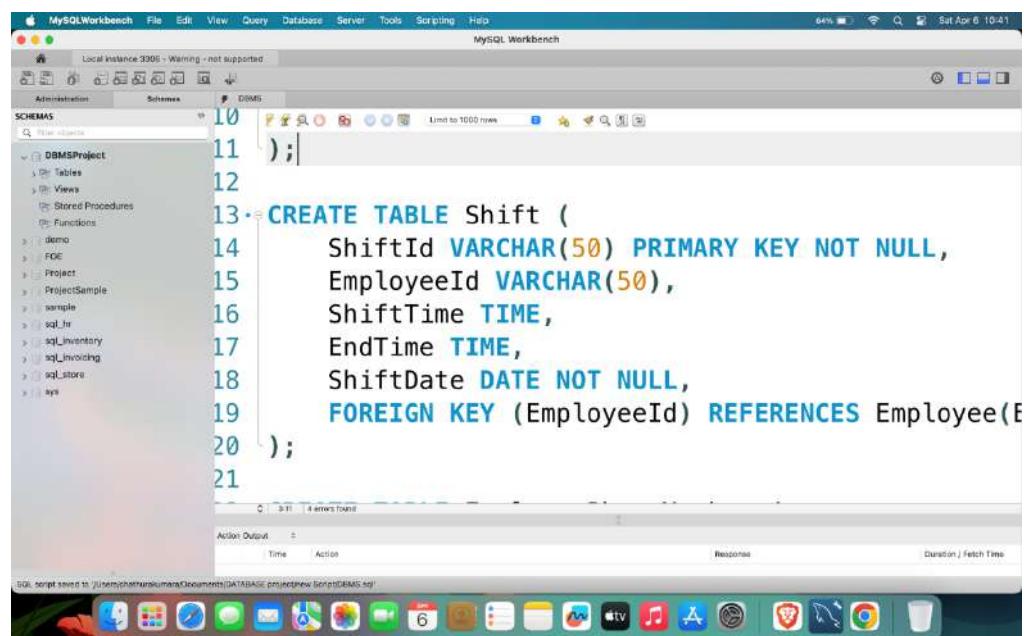


The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The title bar reads "MySQLWorkbench" and "Local Instance 3306 - Warning - not supported". The main window displays an SQL editor with the following code:

```
CREATE DATABASE Smart_Parking_Management_System;
USE Smart_Parking_Management_System;
CREATE TABLE Employee (
    EmployeeId VARCHAR(50) PRIMARY KEY,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    Email VARCHAR(50),
    JobTitle VARCHAR(50) NOT NULL
);
```

The code is numbered from 1 to 15. The status bar at the bottom indicates "0 rows" and "0 errors found". The MySQL Workbench toolbar is visible at the top, and the Mac OS X dock is visible at the bottom.

- ❖ Create table Shift.

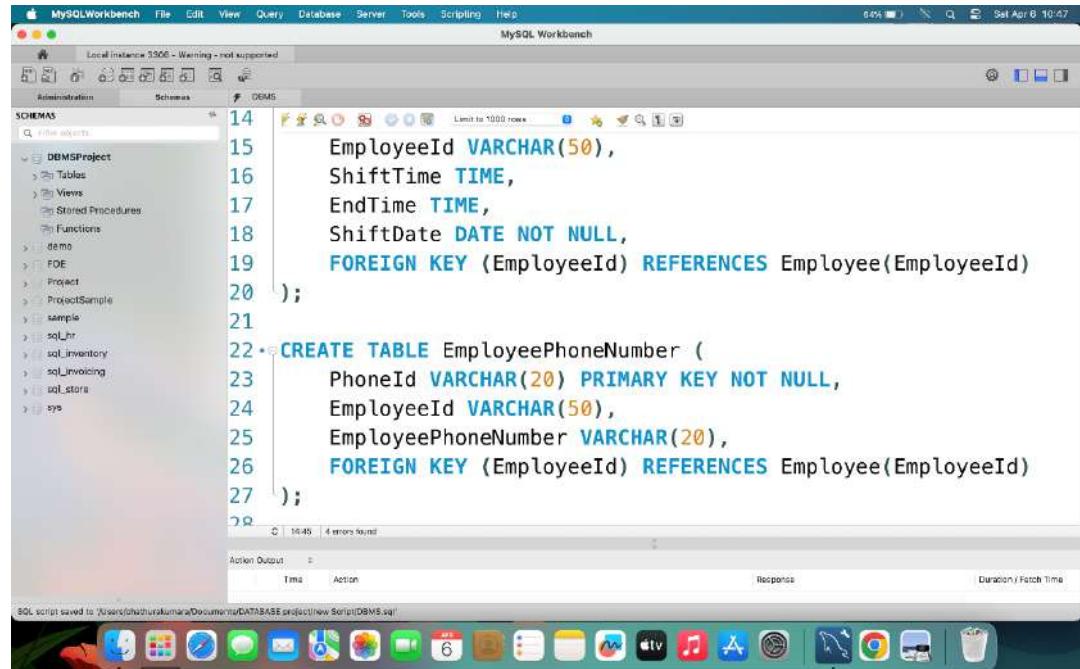


The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The title bar reads "MySQLWorkbench" and "Local Instance 3306 - Warning - not supported". The main window displays an SQL editor with the following code:

```
CREATE TABLE Shift (
    ShiftId VARCHAR(50) PRIMARY KEY NOT NULL,
    EmployeeId VARCHAR(50),
    ShiftTime TIME,
    EndTime TIME,
    ShiftDate DATE NOT NULL,
    FOREIGN KEY (EmployeeId) REFERENCES Employee(E)
);
```

The code is numbered from 11 to 21. The status bar at the bottom indicates "0 rows" and "0 errors found". The MySQL Workbench toolbar is visible at the top, and the Mac OS X dock is visible at the bottom.

❖ Create table Employee Phone Number

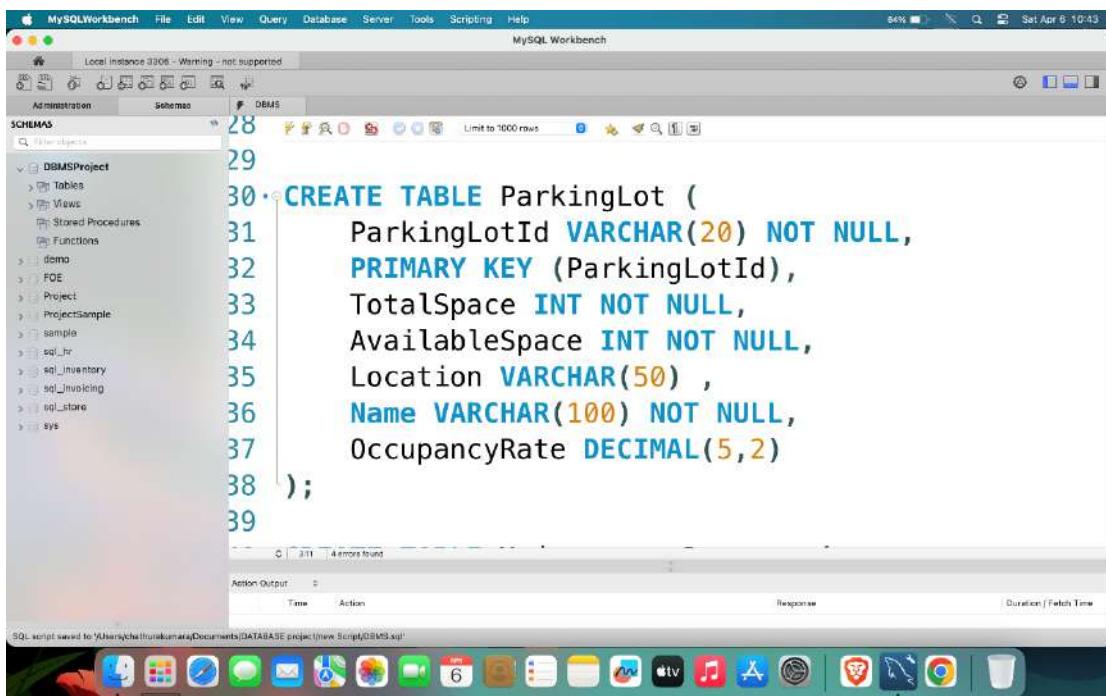


The screenshot shows the MySQL Workbench interface with the following SQL code:

```
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 code defines two tables: 'Shift' and 'EmployeePhoneNumber'. The 'Shift' table has columns for EmployeeId (VARCHAR(50)), ShiftTime (TIME), EndTime (TIME), and ShiftDate (DATE NOT NULL). It includes a FOREIGN KEY constraint referencing the Employee table. The 'EmployeePhoneNumber' table has columns for PhoneId (VARCHAR(20) PRIMARY KEY NOT NULL), EmployeeId (VARCHAR(50)), and EmployeePhoneNumber (VARCHAR(20)). It also includes a FOREIGN KEY constraint referencing the Employee table.

❖ Create table Parking Lot

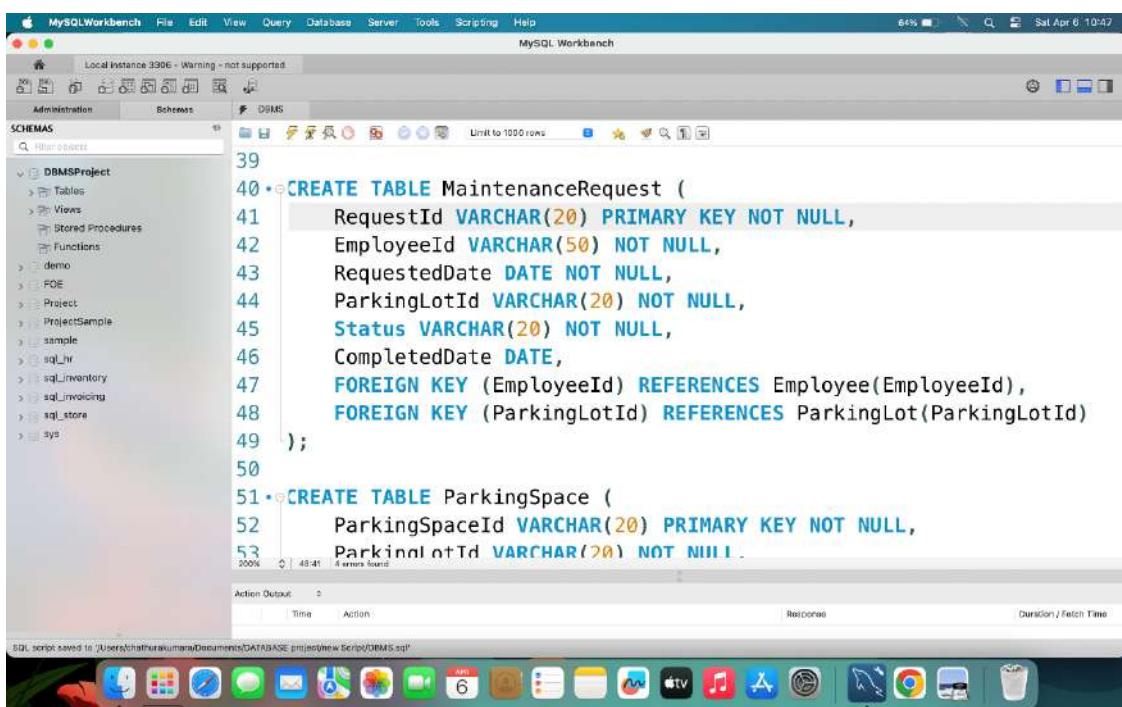


The screenshot shows the MySQL Workbench interface with the 'DBMS' tab selected. In the left sidebar, under the 'Schemas' section, the 'DBMSProject' schema is expanded, showing 'Tables', 'Views', 'Stored Procedures', and 'Functions'. Below these, several other schemas like 'demo', 'POE', 'Project', etc., are listed. 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 code is highlighted in blue and yellow, indicating syntax. The status bar at the bottom of the window shows 'SQL script saved to "/Users/charithraukumar/Documents/DATABASE project/New Script/DBMS.sql"'.

❖ Create table Maintenance Request

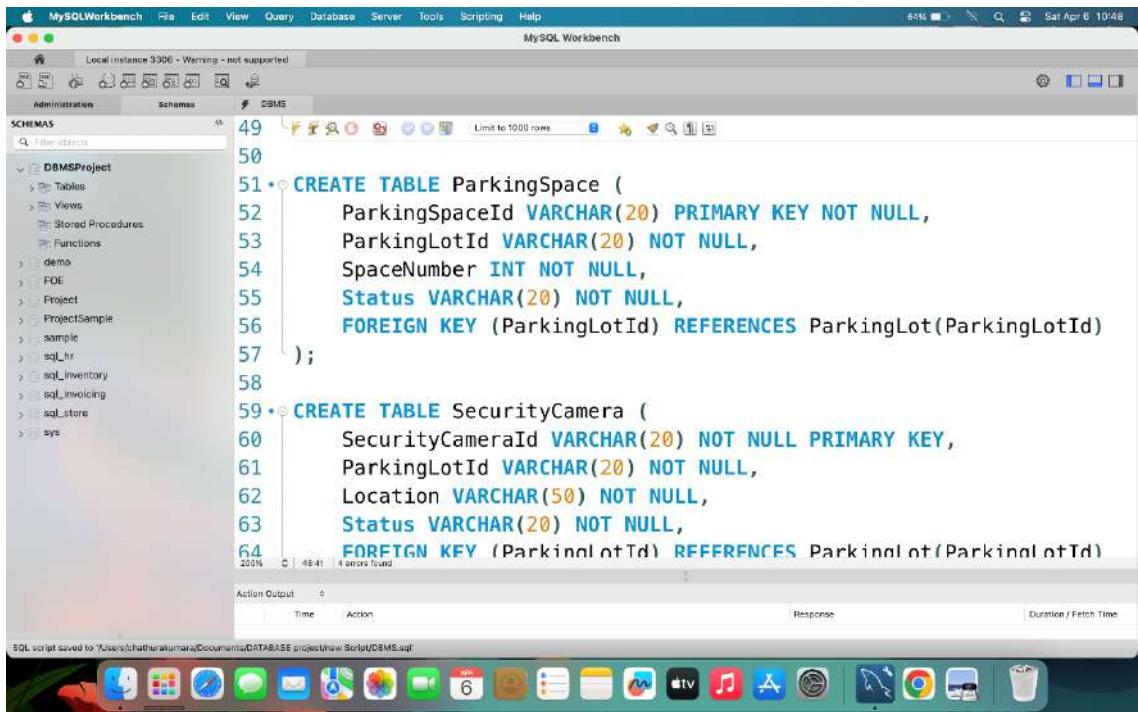


The screenshot shows the MySQL Workbench interface with the 'DBMS' tab selected. In the left sidebar, under the 'Schemas' section, the 'DBMSProject' schema is expanded, showing 'Tables', 'Views', 'Stored Procedures', and 'Functions'. Below these, several other schemas like 'demo', 'POE', 'Project', etc., are listed. 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 code is highlighted in blue and yellow, indicating syntax. The status bar at the bottom of the window shows 'SQL script saved to "/Users/charithraukumar/Documents/DATABASE project/New Script/DBMS.sql"'.

❖ Create table Parking Space

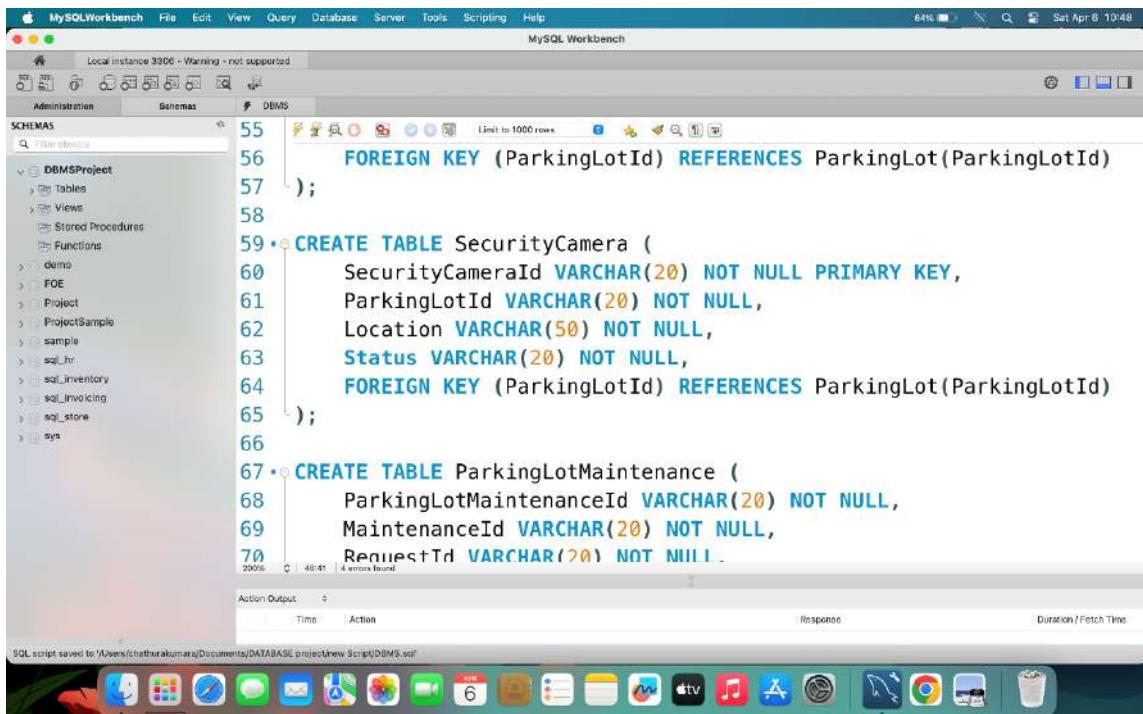


The screenshot shows the MySQL Workbench interface with the SQL tab active. The code area contains the following SQL script:

```
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 status bar at the bottom indicates "200% 48:41 4 errors found". The title bar shows "MySQL Workbench" and the system tray shows various application icons.

❖ Create table Security Camera

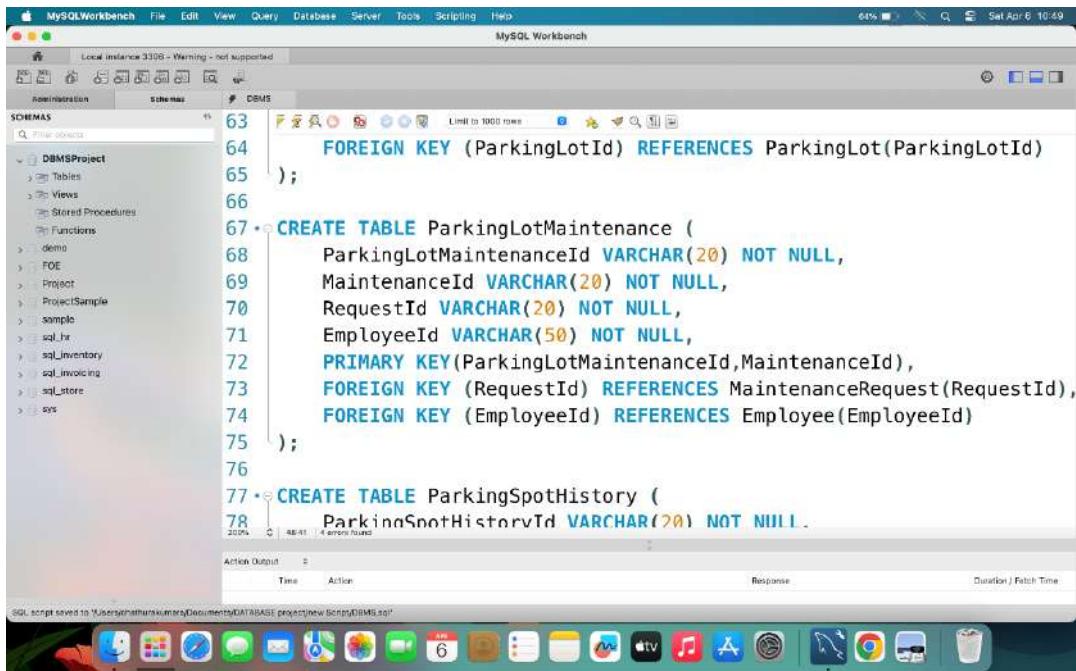


The screenshot shows the MySQL Workbench interface with the SQL tab active. The code area contains the following SQL script:

```
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 status bar at the bottom indicates "200% 48:41 4 errors found". The title bar shows "MySQL Workbench" and the system tray shows various application icons.

❖ Create table Parking Lot Maintenance

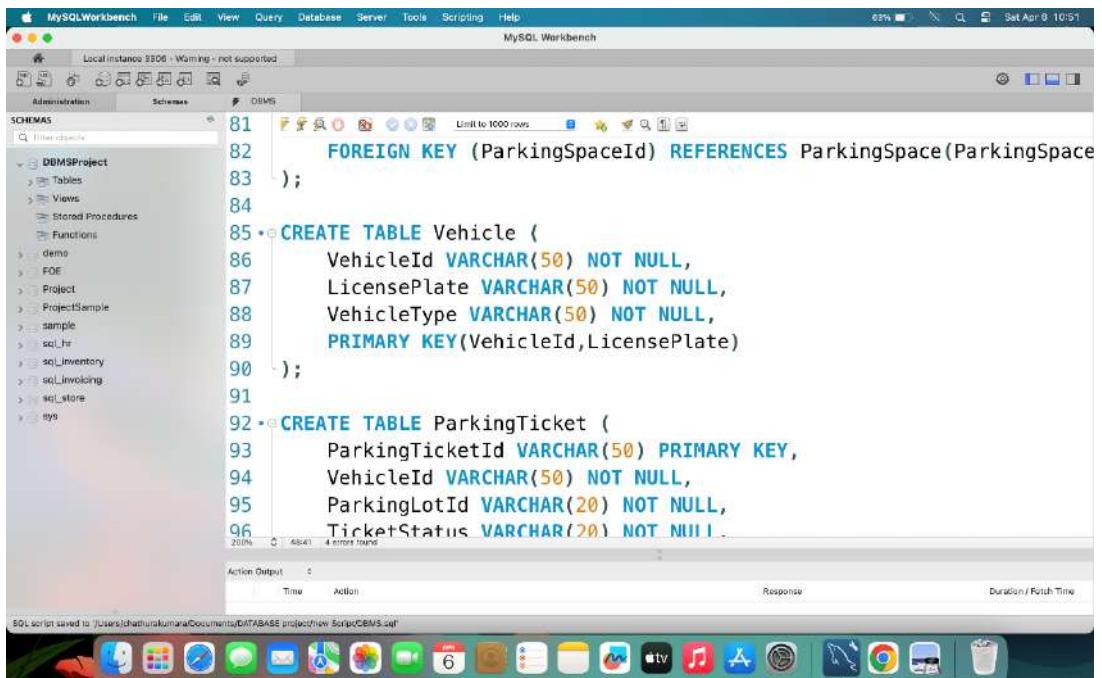


The screenshot shows the MySQL Workbench interface with the SQL editor tab active. The code being run is:

```
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 "4 errors found".

❖ Create Vehicle

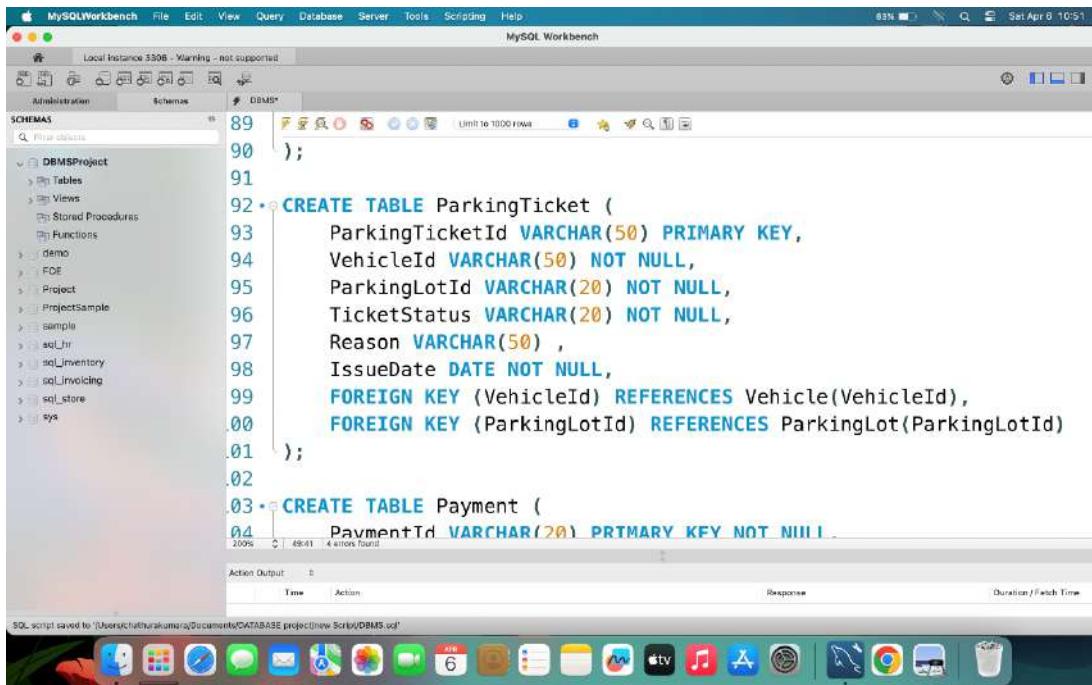


The screenshot shows the MySQL Workbench interface with the SQL editor tab active. The code being run is:

```
81 FOREIGN KEY (ParkingSpaceId) REFERENCES ParkingSpace(ParkingSpace
82 );
83
84
85 CREATE TABLE Vehicle (
86     VehicleId VARCHAR(50) NOT NULL,
87     LicensePlate VARCHAR(50) NOT NULL,
88     VehicleType VARCHAR(50) NOT NULL,
89     PRIMARY KEY(VehicleId,LicensePlate)
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 "4 errors found".

❖ Create table ParkingTicket.

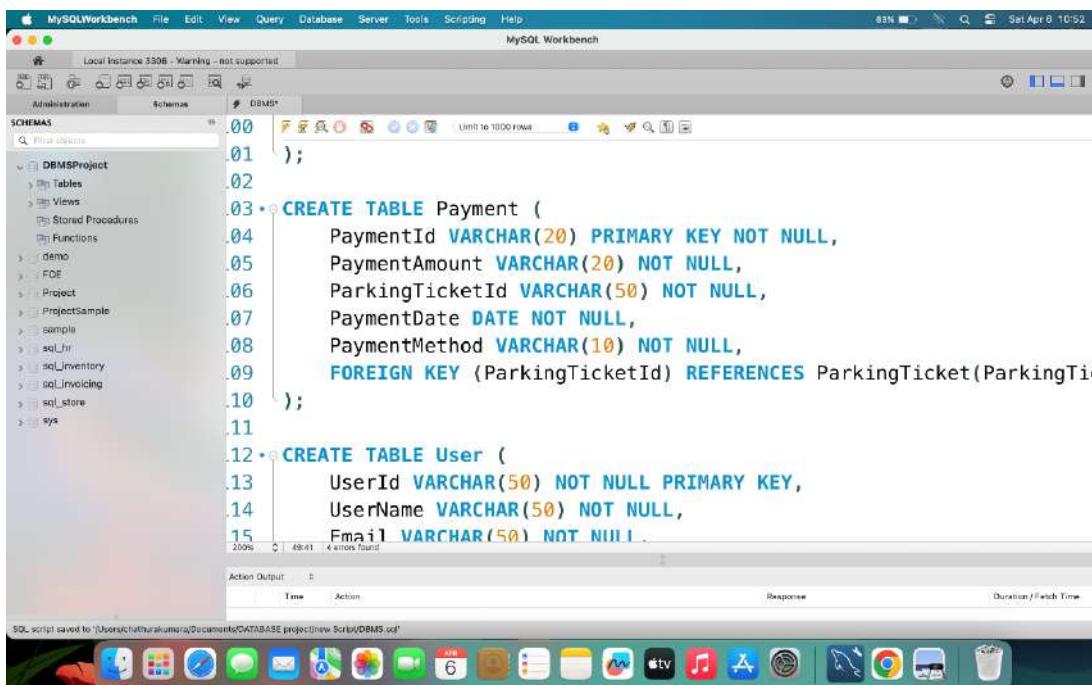


The screenshot shows the MySQL Workbench interface with the SQL tab active. The code editor contains the following SQL script:

```
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,
```

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

❖ Create table Payment.

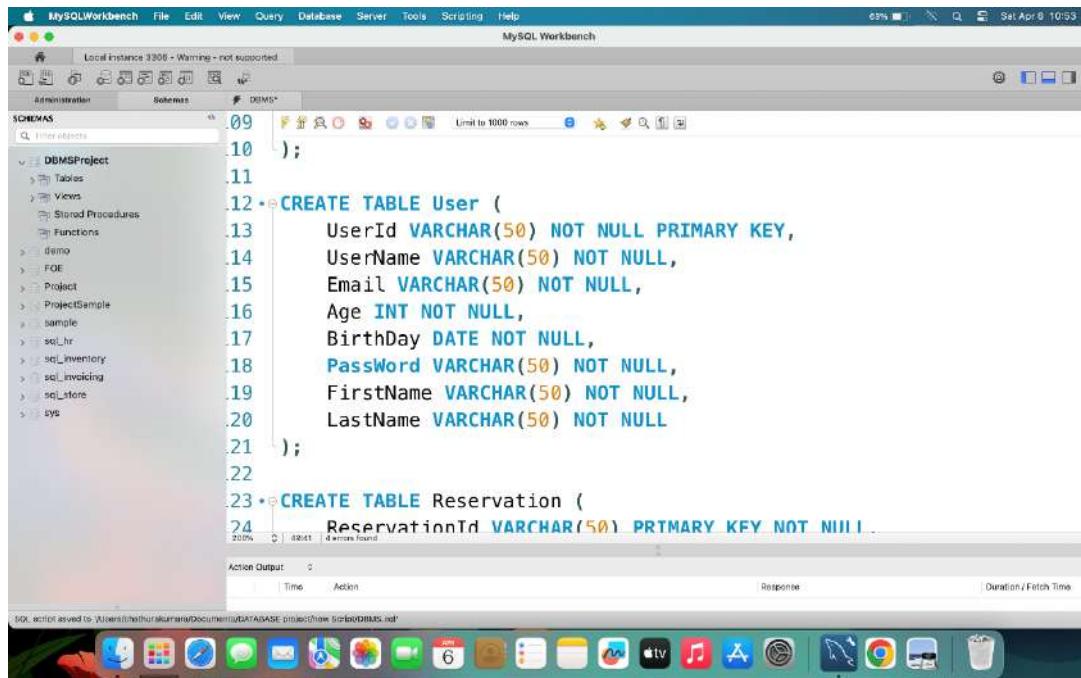


The screenshot shows the MySQL Workbench interface with the SQL tab active. The code editor contains the following SQL script:

```
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,
```

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

❖ Create table User.



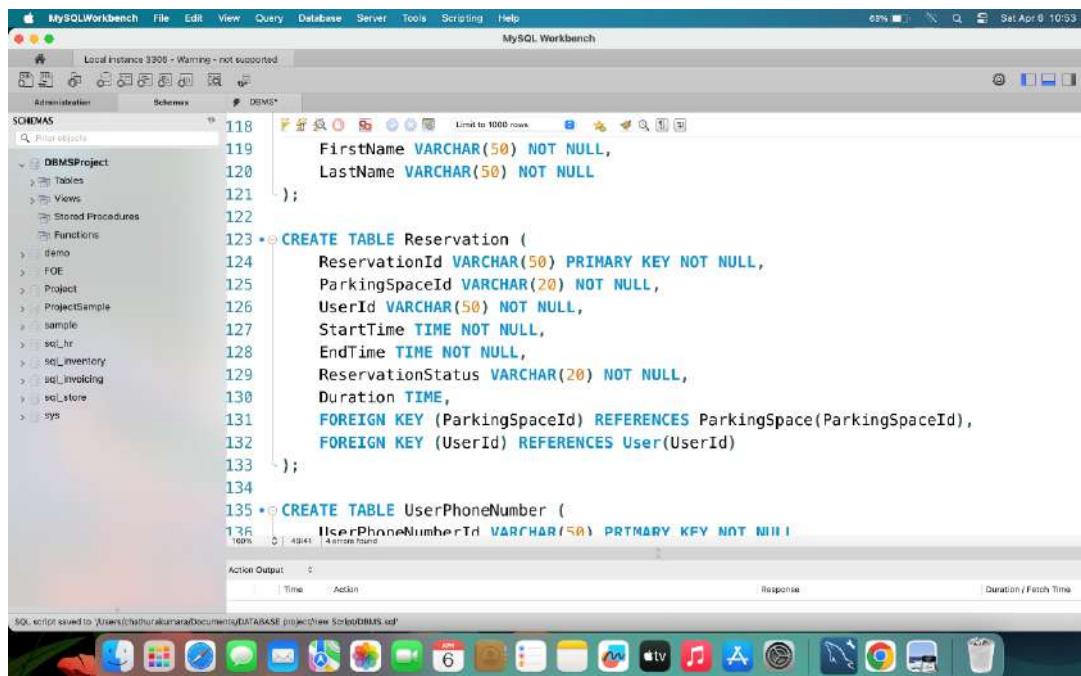
The screenshot shows the MySQL Workbench interface with the 'Schemas' tree on the left containing 'DBMSProject' and several other schemas. The main pane displays the following SQL code:

```
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 the SQL script was saved to 'C:\Users\thushararajana\Documents\DATABASE\project\view\ScriptDBMS.sql'.

❖ Create table Reservation.



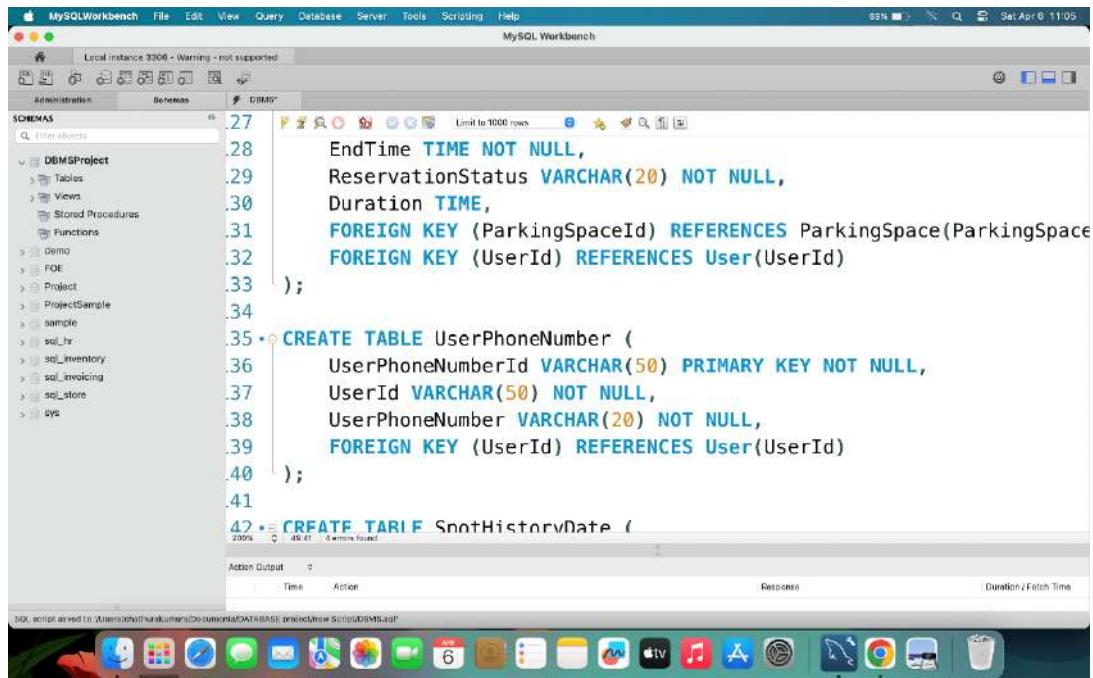
The screenshot shows the MySQL Workbench interface with the 'Schemas' tree on the left containing 'DBMSProject' and several other schemas. The main pane displays the following SQL code:

```
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 the SQL script was saved to 'C:\Users\thushararajana\Documents\DATABASE\project\view\ScriptDBMS.sql'.

❖ Create table UserPhoneNumber.



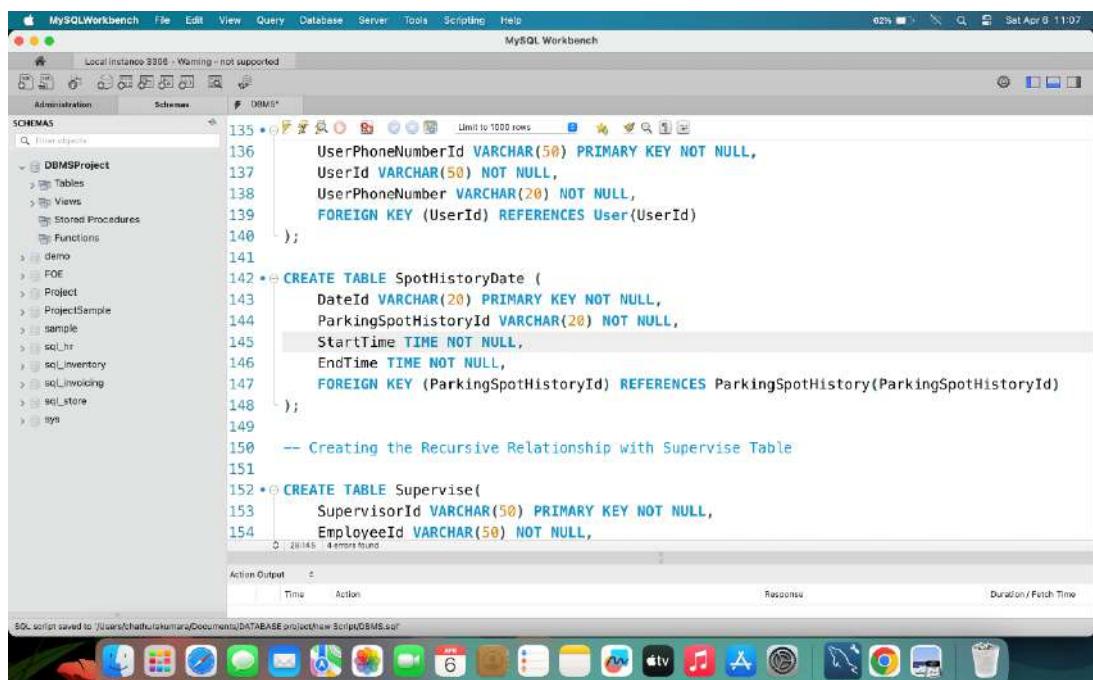
```

27 |   EndTime TIME NOT NULL,
28 |   ReservationStatus VARCHAR(20) NOT NULL,
29 |   Duration TIME,
30 |   FOREIGN KEY (ParkingSpaceId) REFERENCES ParkingSpace(ParkingSpaceId)
31 |   FOREIGN KEY (UserId) REFERENCES User(UserId)
32 | );
33 |
34 |
35 *+ CREATE TABLE UserPhoneNumber (
36 |   UserPhoneNumberId VARCHAR(50) PRIMARY KEY NOT NULL,
37 |   UserId VARCHAR(50) NOT NULL,
38 |   UserPhoneNumber VARCHAR(20) NOT NULL,
39 |   FOREIGN KEY (UserId) REFERENCES User(UserId)
40 | );
41 |
42 *+ CREATE TABLE SpotHistoryDate (

```

The screenshot shows the MySQL Workbench interface with a SQL editor window. The code being run is for creating the UserPhoneNumber table. The table has three columns: UserPhoneNumberId (primary key), UserId, and UserPhoneNumber. It also includes foreign key constraints linking to the User table. The code is numbered from 27 to 41. Below the code, there is a section titled 'CREATE TABLE SpotHistoryDate' which is partially visible. The status bar at the bottom indicates the script was saved to a local file.

❖ Create table SpotHistoryDate.



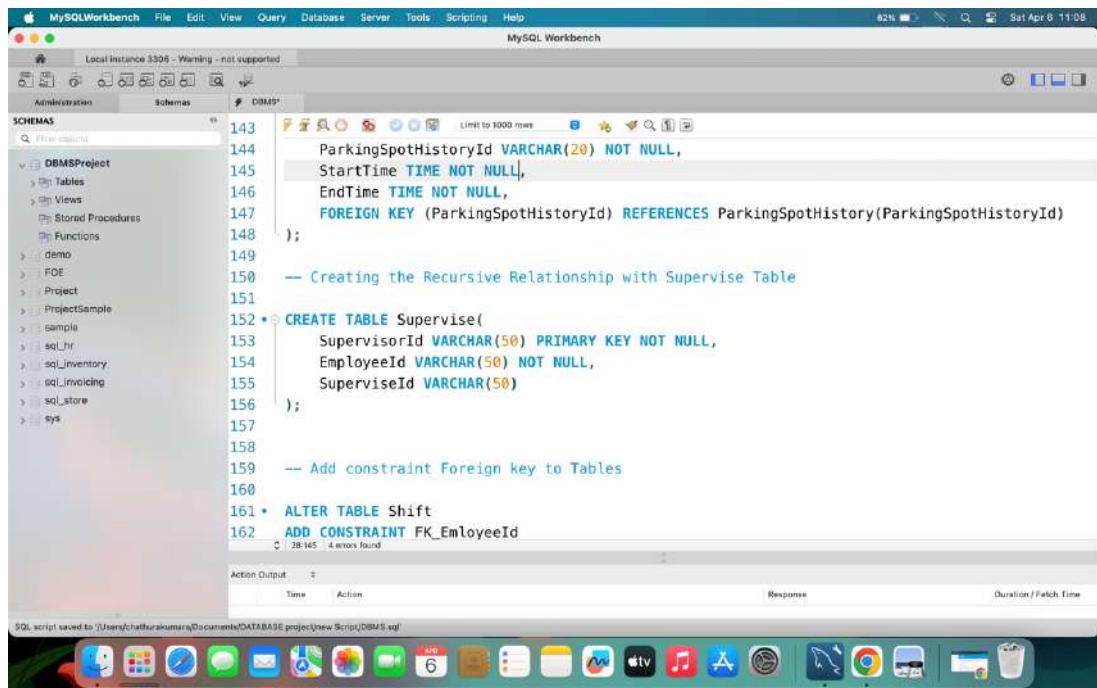
```

135 *+ CREATE TABLE UserPhoneNumber (
136 |   UserPhoneNumberId VARCHAR(50) PRIMARY KEY NOT NULL,
137 |   UserId VARCHAR(50) NOT NULL,
138 |   UserPhoneNumber VARCHAR(20) NOT NULL,
139 |   FOREIGN KEY (UserId) REFERENCES User(UserId)
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,

```

The screenshot shows the MySQL Workbench interface with a SQL editor window. The code being run is for creating the SpotHistoryDate table. The table has four columns: DateId (primary key), ParkingSpotHistoryId, StartTime, and EndTime. It also includes a foreign key constraint linking to the ParkingSpotHistory table. The code is numbered from 135 to 154. Below the code, there is a section titled 'Creating the Recursive Relationship with Supervise Table' which is partially visible. The status bar at the bottom indicates the script was saved to a local file.

❖ Create table Supervise.



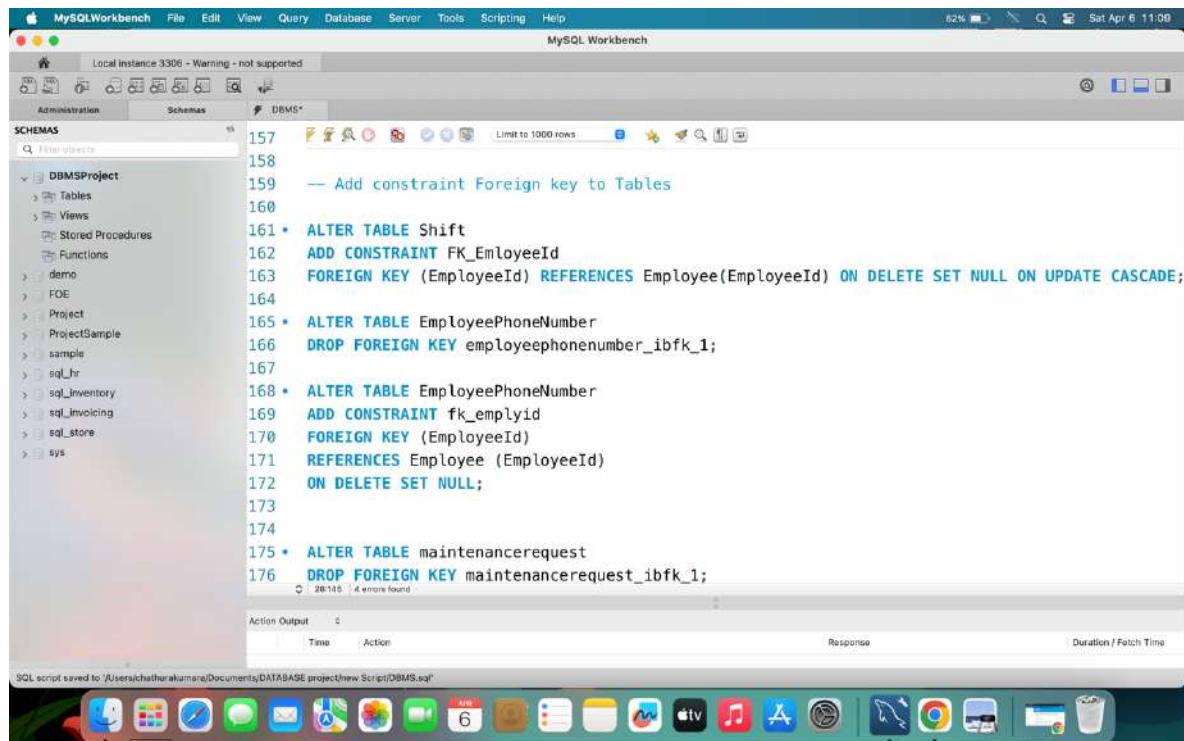
The screenshot shows the MySQL Workbench interface with a SQL editor window. The code being run is:

```
143     ParkingSpotHistoryId VARCHAR(20) NOT NULL,
144     StartTime TIME NOT NULL,
145     EndTime TIME NOT NULL,
146     FOREIGN KEY (ParkingSpotHistoryId) REFERENCES ParkingSpotHistory(ParkingSpotHistoryId)
147   );
148
149   -- Creating the Recursive Relationship with Supervise Table
150
151 * CREATE TABLE Supervise(
152     SupervisorId VARCHAR(50) PRIMARY KEY NOT NULL,
153     EmployeeId VARCHAR(50) NOT NULL,
154     SuperviseId VARCHAR(50)
155   );
156
157
158   -- Add constraint Foreign key to Tables
159
160 * ALTER TABLE Shift
161   ADD CONSTRAINT FK_EmployeeId
```

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

3.2 Creating Foreign Key Constraint

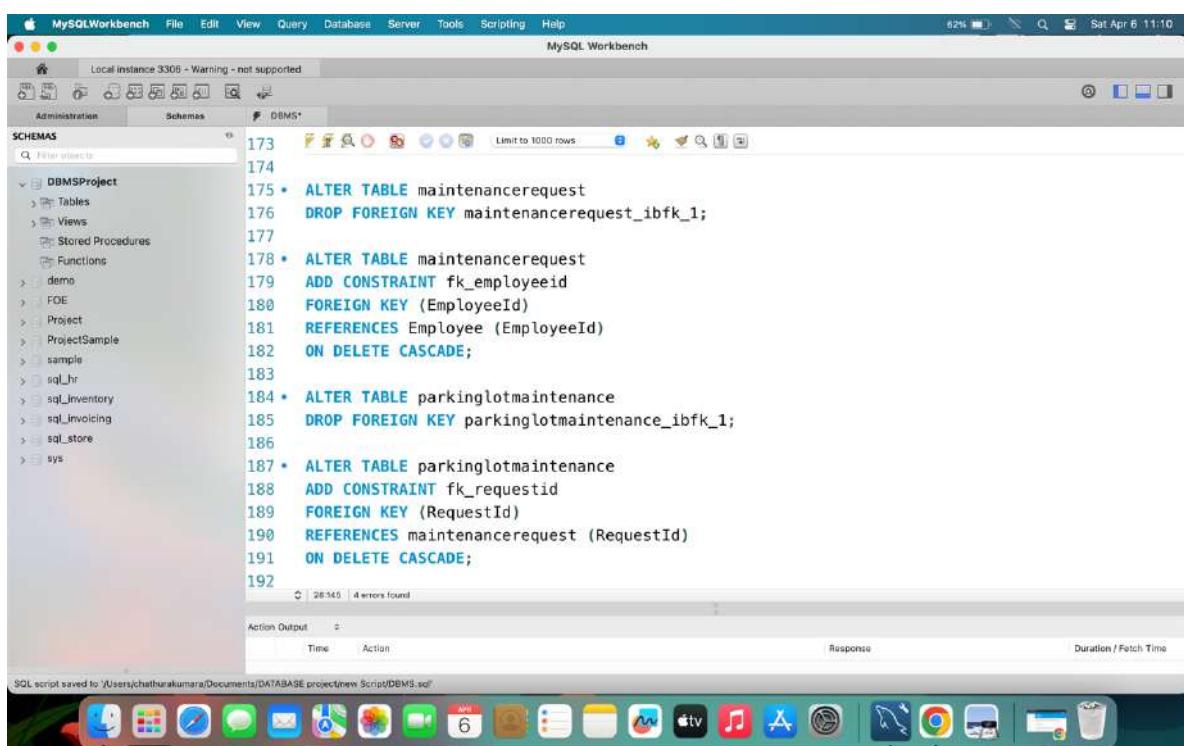
We have also inserted some of the foreign keys when implementing the table.



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

```
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;
```

The script is saved to `/Users/charithakumara/Documents/DATABASE project/new Script/DBMS.sql`.



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

```
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_requestid
189 FOREIGN KEY (RequestId)
190 REFERENCES maintenancerequest (RequestId)
191 ON DELETE CASCADE;
192
```

The script is saved to `/Users/charithakumara/Documents/DATABASE project/new Script/DBMS.sql`.

MySQLWorkbench

File Edit View Query Database Server Tools Scripting Help

MySQL Workbench

Local instance 3305 - Warning - not supported

Administration Schemas DBMS*

Limit to 1000 rows

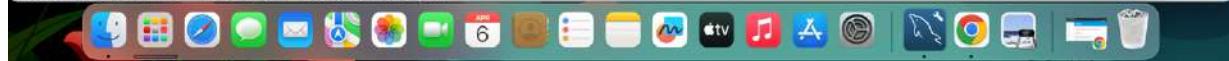
191 192
193 * ALTER TABLE parkingticket
194 ADD CONSTRAINT fk_parkinglot_id
195 FOREIGN KEY (ParkingLotId)
196 REFERENCES ParkingLot (ParkingLotId)
197 ON DELETE CASCADE;
198
199 * ALTER TABLE ParkingTicket
200 DROP FOREIGN KEY parkingticket_ibfk_1;
201
202 * ALTER TABLE ParkingTicket
203 ADD CONSTRAINT 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

28:145 : 4 errors found

Action Output

Time Action Response Duration / Fetch Time

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



MySQLWorkbench

File Edit View Query Database Server Tools Scripting Help

MySQL Workbench

Local instance 3306 - Warning - not supported

Administration Schemas FINALCODE

Limit to 1000 rows

288 * ALTER TABLE ParkingTicket
289 ADD CONSTRAINT fk_vehicleid
290 FOREIGN KEY (VehicleId)
291 REFERENCES Vehicle(VehicleId)
292 ON DELETE CASCADE;
293
294 -- At 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
302
303 * ALTER TABLE Reservation
304 ADD CONSTRAINT fk_parking_space_id
305 FOREIGN KEY (ParkingSpaceId)
306 REFERENCES ParkingSpace (ParkingSpaceId)
307 ON DELETE CASCADE;
308
309 -- At Spottistorydate
310 * ALTER TABLE Spottistorydate
311 DEFERRED FOREIGN KEY spottistorydate_ibfk_1
312
313 * ALTER TABLE Spottistorydate
314 ADD CONSTRAINT fk_parkingSpotHistory_id
315 FOREIGN KEY (ParkingSpotHistoryId)
316 REFERENCES ParkingSpotHistory (ParkingSpotHistoryId)
317 ON DELETE CASCADE;
318
319
320 -- At UserHeader
321 * ALTER TABLE UserHeader
322 ADD CONSTRAINT fk_user_id
323 FOREIGN KEY (Userid)
324 REFERENCES User (Userid)
325 ON DELETE CASCADE;

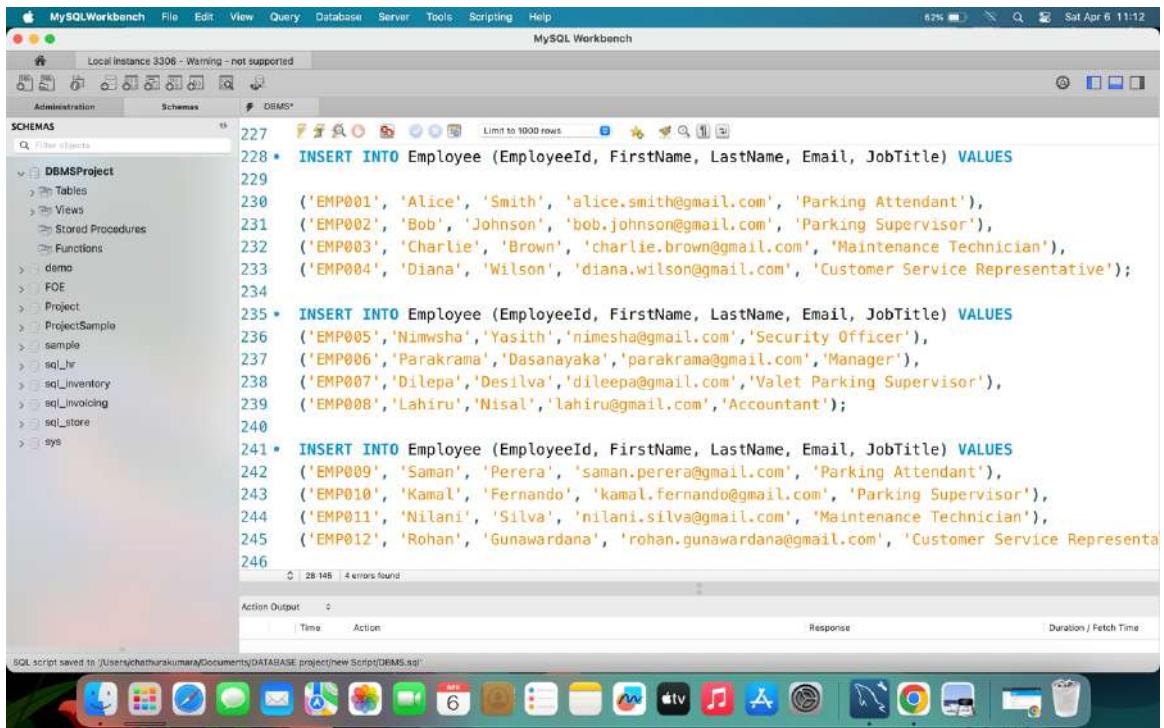
1:1220 : 4 errors found

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



3.3 Inserting Data for Tables

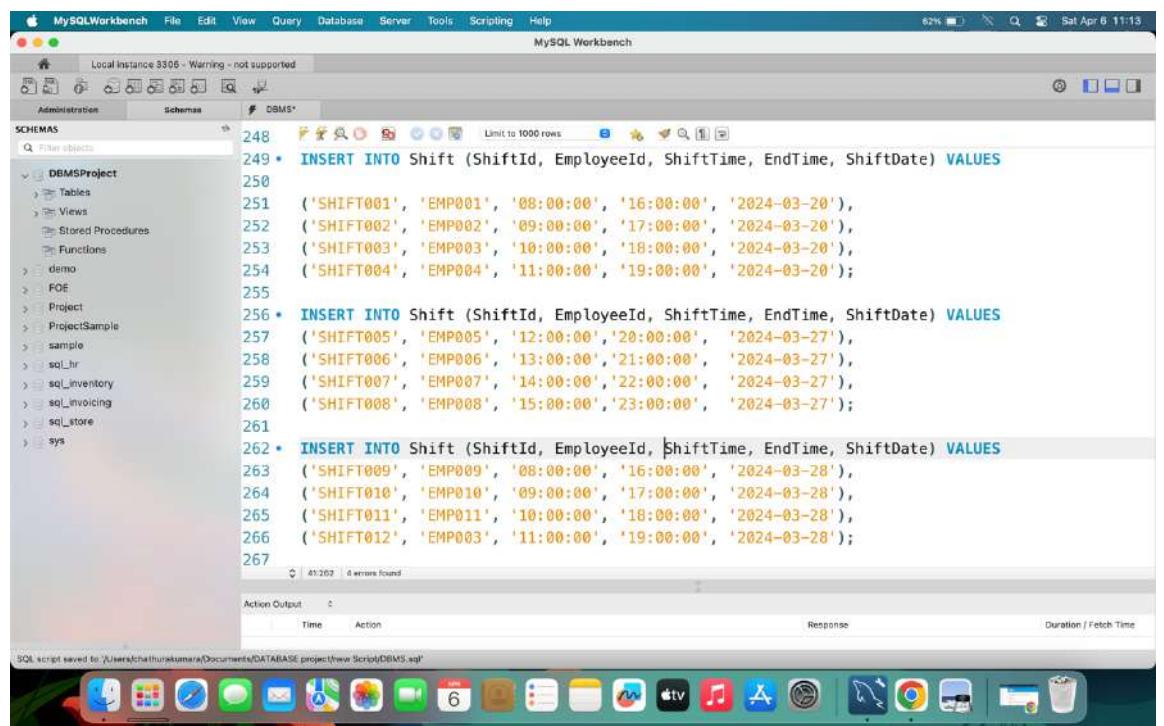
❖ Employee Table



The screenshot shows the MySQL Workbench interface with the 'Schemas' tab selected. In the central query editor window, there is a large block of SQL code for inserting data into the 'Employee' table. The code consists of approximately 246 lines of SQL, starting with line 227 and ending at line 246. The code includes several INSERT statements with multiple rows of employee data, such as ('EMP001', 'Alice', 'Smith', 'alice.smith@gmail.com', 'Parking Attendant') and ('EMP002', 'Bob', 'Johnson', 'bob.johnson@gmail.com', 'Parking Supervisor'). The interface also shows a status bar at the bottom indicating '28:146 | 4 errors found'.

```
227 228 • INSERT INTO Employee (EmployeeId, FirstName, LastName, Email, JobTitle) VALUES
229
230 ('EMP001', 'Alice', 'Smith', 'alice.smith@gmail.com', 'Parking Attendant'),
231 ('EMP002', 'Bob', 'Johnson', 'bob.johnson@gmail.com', 'Parking Supervisor'),
232 ('EMP003', 'Charlie', 'Brown', 'charlie.brown@gmail.com', 'Maintenance Technician'),
233 ('EMP004', 'Diana', 'Wilson', 'diana.wilson@gmail.com', 'Customer Service Representative');
234
235 • INSERT INTO Employee (EmployeeId, FirstName, LastName, Email, JobTitle) VALUES
236 ('EMP005', 'Nimwsha', 'Yasith', 'nimwsha@gmail.com', 'Security Officer'),
237 ('EMP006', 'Parakrama', 'Dasanayaka', 'parakrama@gmail.com', 'Manager'),
238 ('EMP007', 'Dileepa', '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 Representative');
246
```

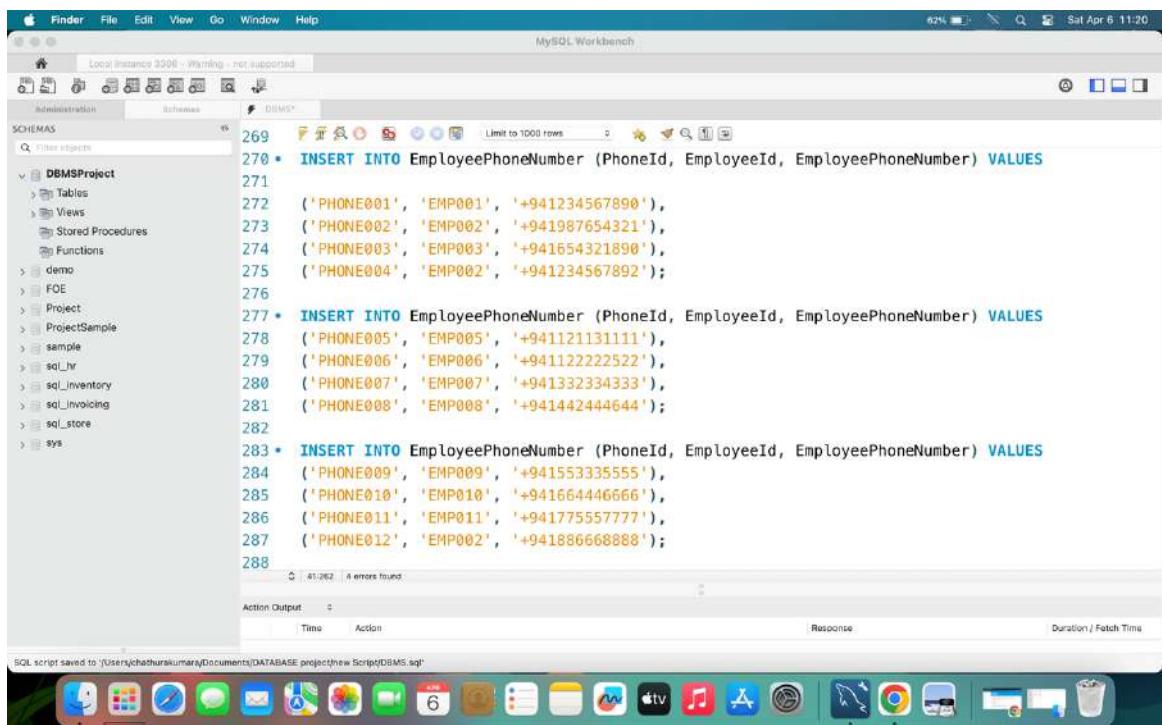
❖ Shift



The screenshot shows the MySQL Workbench interface with the 'Schemas' tab selected. In the central query editor window, there is a large block of SQL code for inserting data into the 'Shift' table. The code consists of approximately 267 lines of SQL, starting with line 248 and ending at line 267. The code includes several INSERT statements with multiple rows of shift data, such as ('SHIFT001', 'EMP001', '08:00:00', '16:00:00', '2024-03-20'), ('SHIFT002', 'EMP002', '09:00:00', '17:00:00', '2024-03-20'), and so on. The interface also shows a status bar at the bottom indicating '41:267 | 4 errors found'.

```
248 249 • INSERT INTO Shift (ShiftId, EmployeeId, ShiftTime, EndTime, ShiftDate) VALUES
250
251 ('SHIFT001', 'EMP001', '08:00:00', '16:00:00', '2024-03-20'),
252 ('SHIFT002', 'EMP002', '09:00:00', '17:00:00', '2024-03-20'),
253 ('SHIFT003', 'EMP003', '10:00:00', '18:00:00', '2024-03-20'),
254 ('SHIFT004', 'EMP004', '11:00:00', '19:00:00', '2024-03-20');
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');
```

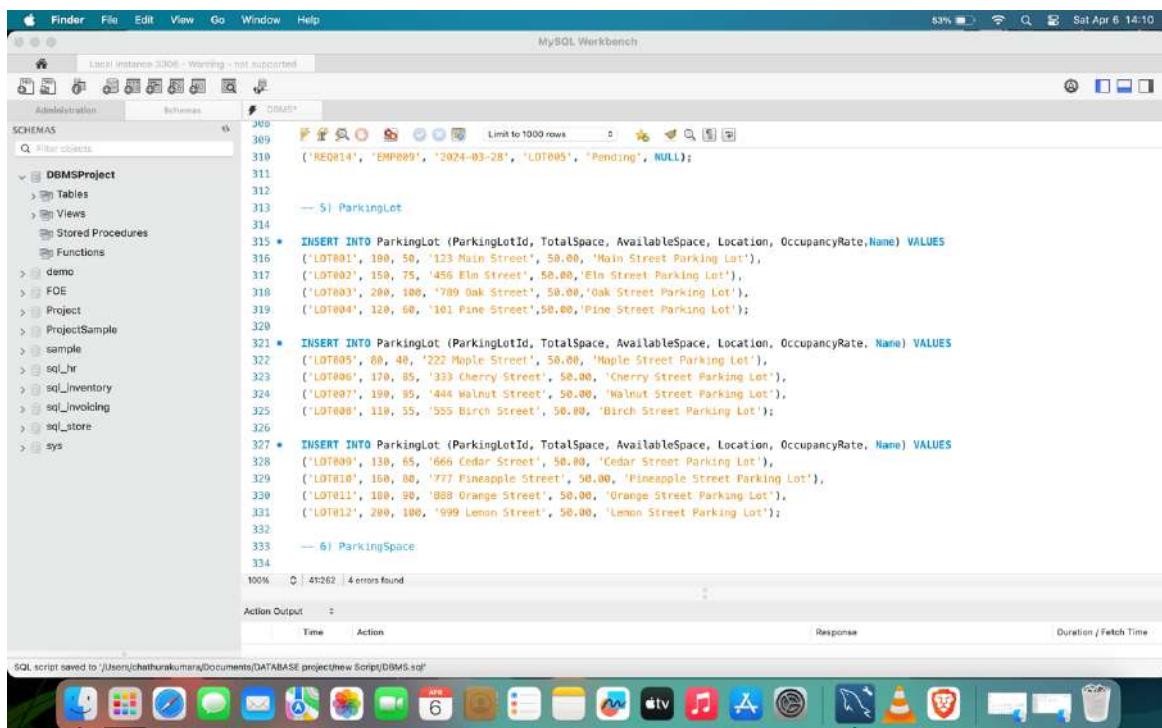
❖ Employee Phone Number



The screenshot shows the MySQL Workbench interface with an open SQL editor window. The code is an SQL script for inserting data into the 'EmployeePhoneNumber' table. The script includes several INSERT INTO statements with values for columns like PhoneId, EmployeeId, and EmployeePhoneNumber. The code spans from line 269 to 288. A status bar at the bottom indicates 41:262 rows affected and 4 errors found.

```
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', '+941553335555'),
285 ('PHONE010', 'EMP010', '+941664446666'),
286 ('PHONE011', 'EMP011', '+941775557777'),
287 ('PHONE012', 'EMP002', '+941886668888');
288
41:262 | 4 errors found.
```

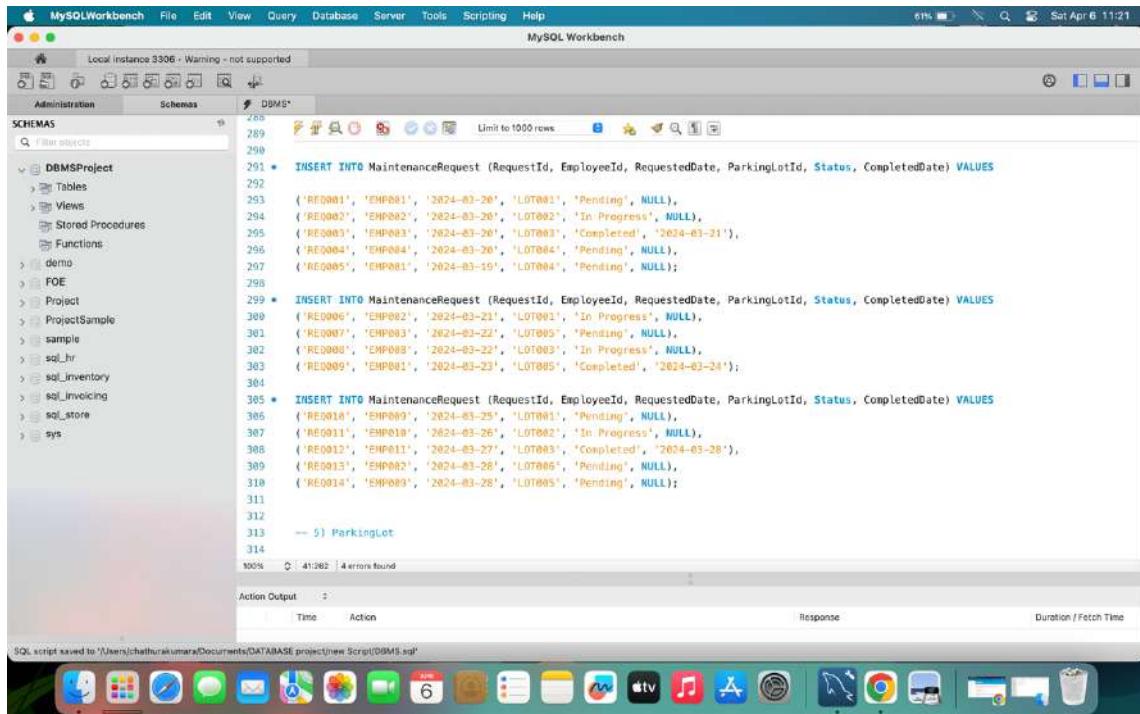
❖ Parking Lot



The screenshot shows the MySQL Workbench interface with an open SQL editor window. The code is an SQL script for inserting data into the 'ParkingLot' table. It includes several INSERT INTO statements with values for columns like ParkingLotId, TotalSpace, AvailableSpace, Location, OccupancyRate, and Name. The code spans from line 300 to 334. A status bar at the bottom indicates 41:262 rows affected and 4 errors found.

```
300
301
302
303
304
305 • INSERT INTO ParkingLot (ParkingLotId, TotalSpace, AvailableSpace, Location, OccupancyRate, Name) VALUES
306 ('LOT001', 'EMP009', '2024-03-28', 'LOT005', 'Pending', NULL);
307
308
309 ('REQM14', '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
100% 41:262 | 4 errors found.
```

❖ Maintenance Request



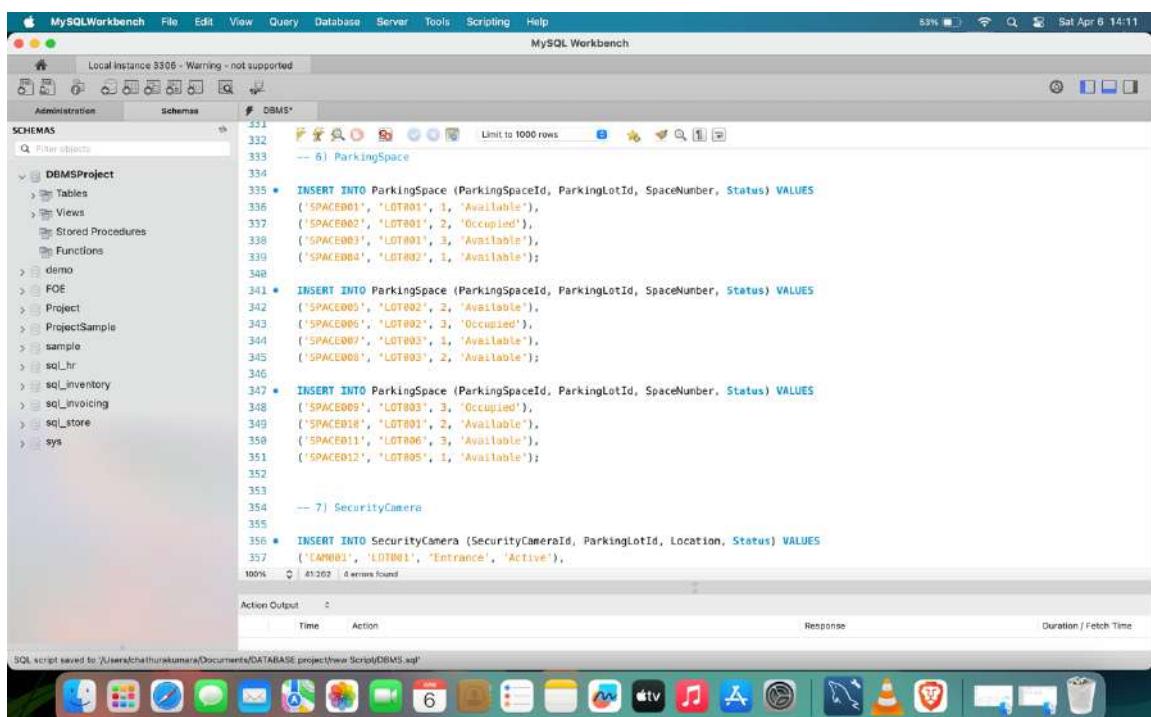
The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** DBMSProject
- Code Editor:** SQL script for creating Maintenance Requests.
- Script Content (Excerpt):**

```
286 • INSERT INTO MaintenanceRequest (RequestId, EmployeeId, RequestedDate, ParkingLotId, Status, CompletedDate) VALUES
287 ('REQ001', 'EMP001', '2024-03-20', 'LOT001', 'Pending', NULL),
288 ('REQ002', 'EMP002', '2024-03-20', 'LOT002', 'In Progress', NULL),
289 ('REQ003', 'EMP003', '2024-03-20', 'LOT003', 'Completed', '2024-03-21'),
290 ('REQ004', 'EMP004', '2024-03-20', 'LOT004', 'Pending', NULL),
291 ('REQ005', 'EMP001', '2024-03-19', 'LOT004', 'Pending', NULL);
292
293 • INSERT INTO MaintenanceRequest (RequestId, EmployeeId, RequestedDate, ParkingLotId, Status, CompletedDate) VALUES
294 ('REQ006', 'EMP002', '2024-03-21', 'LOT001', 'In Progress', NULL),
295 ('REQ007', 'EMP003', '2024-03-22', 'LOT005', 'Pending', NULL),
296 ('REQ008', 'EMP008', '2024-03-22', 'LOT003', 'In Progress', NULL),
297 ('REQ009', 'EMP001', '2024-03-23', 'LOT005', 'Completed', '2024-03-24');
298
299 • INSERT INTO MaintenanceRequest (RequestId, EmployeeId, RequestedDate, ParkingLotId, Status, CompletedDate) VALUES
300 ('REQ010', 'EMP009', '2024-03-25', 'LOT001', 'Pending', NULL),
301 ('REQ011', 'EMP010', '2024-03-26', 'LOT002', 'In Progress', NULL),
302 ('REQ012', 'EMP011', '2024-03-27', 'LOT003', 'Completed', '2024-03-28'),
303 ('REQ013', 'EMP002', '2024-03-28', 'LOT005', 'Pending', NULL),
304 ('REQ014', 'EMP009', '2024-03-28', 'LOT005', 'Pending', NULL);
305
306 • INSERT INTO MaintenanceRequest (RequestId, EmployeeId, RequestedDate, ParkingLotId, Status, CompletedDate) VALUES
307 ('REQ015', 'EMP001', '2024-03-29', 'LOT002', 'In Progress', NULL),
308 ('REQ016', 'EMP008', '2024-03-29', 'LOT003', 'Completed', '2024-03-30'),
309 ('REQ017', 'EMP009', '2024-03-29', 'LOT005', 'Pending', NULL),
310 ('REQ018', 'EMP001', '2024-03-29', 'LOT005', 'Pending', NULL);
311
312 --- 5) Parkinglot
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
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'),
```

- Action Output:** 41/262 | 4 errors found
- SQL script saved to:** /Users/chaithrakumara/Documents/DATABASE project/New Script/DBMS.sql

❖ Parking Space



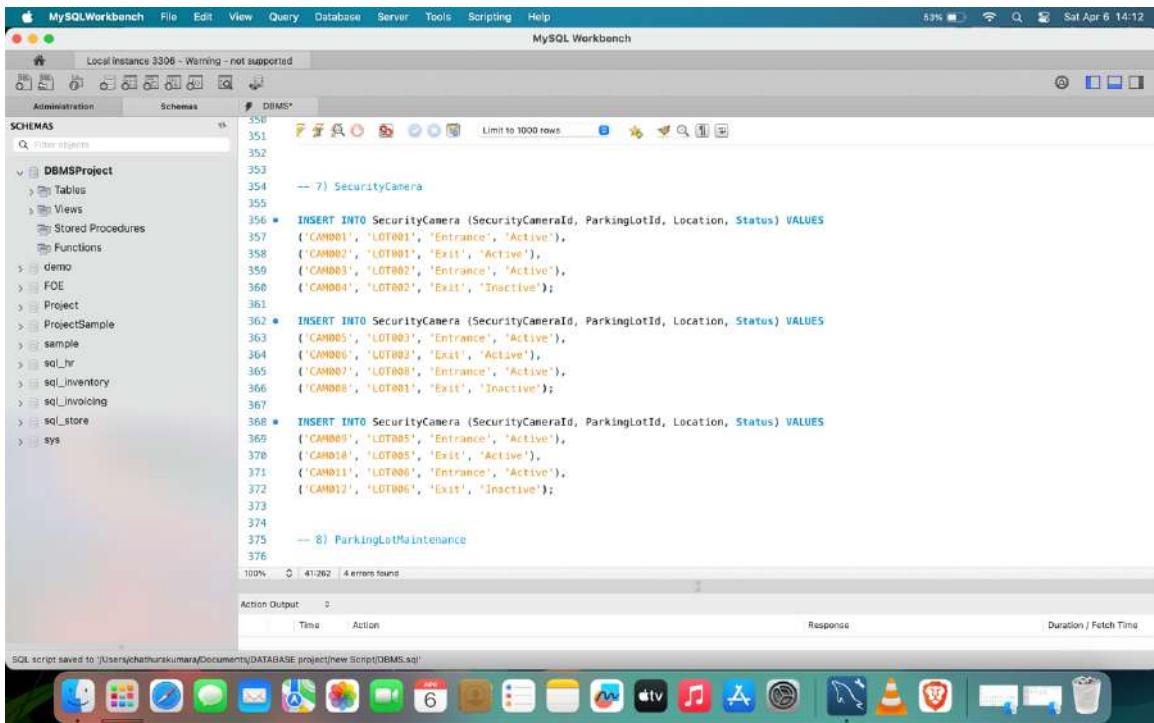
The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** DBMSProject
- Code Editor:** SQL script for creating Parking Spaces.
- Script Content (Excerpt):**

```
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'),
```

- Action Output:** 41/262 | 4 errors found
- SQL script saved to:** /Users/chaithrakumara/Documents/DATABASE project/New Script/DBMS.sql

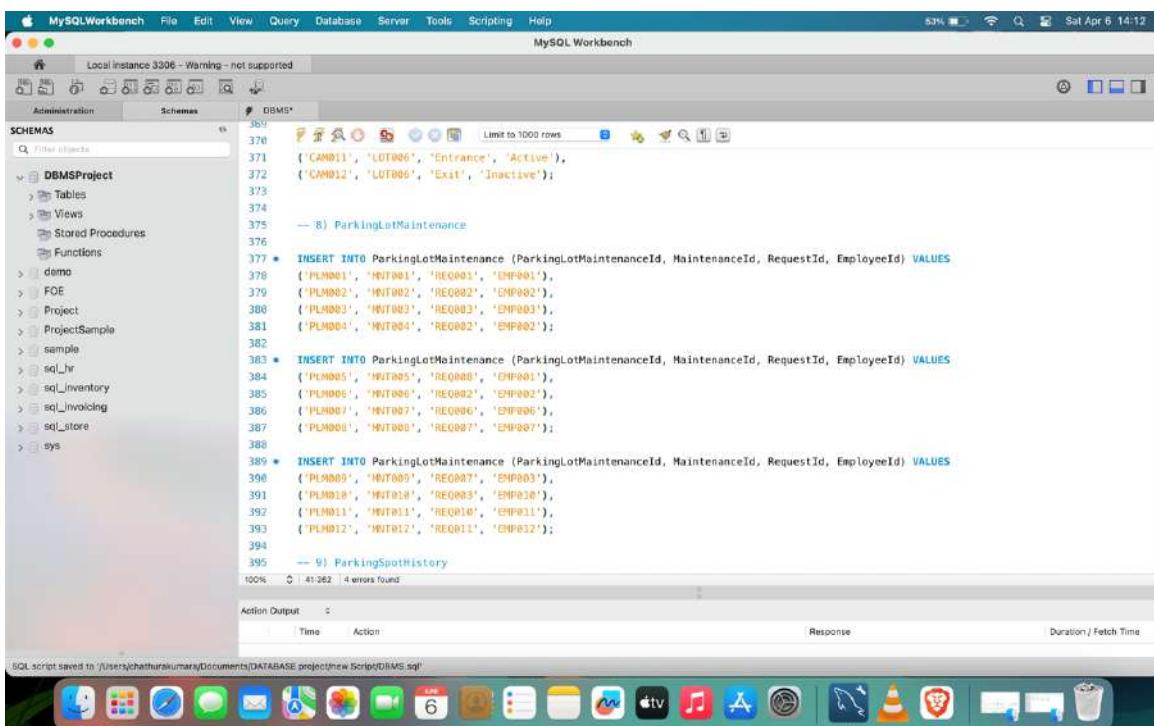
❖ Security Camera



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

- Title Bar:** MySQL Workbench, Local Instance 3306 - Warning - not supported, Sat Apr 6 14:12.
- Schemas:** DBMSProject is selected.
- Code Editor:** Displays SQL scripts for creating SecurityCamera and ParkingLotMaintenance tables. The SecurityCamera table has columns: SecurityCameraId, ParkinglotId, Location, Status. The ParkingLotMaintenance table has columns: ParkingLotMaintenanceId, MaintenanceId, RequestId, EmployeeId.
- Status Bar:** 100%, 41/262 | 4 errors found.
- Action Output:** Shows a table with columns: Time, Action, Response, Duration / Fetch Time.
- Bottom:** SQL script saved to 'Users/jchathurakumara/Documents/DATABASE project/new Script/DBMS.sql'.

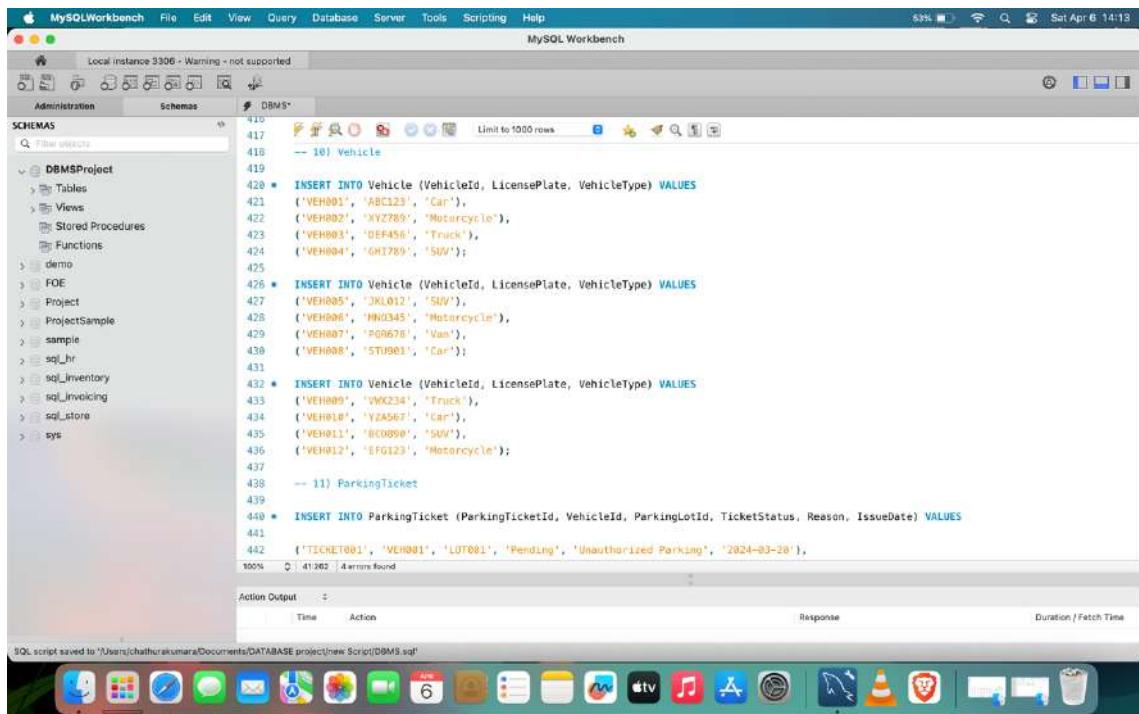
❖ Parking Lot Maintenance



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

- Title Bar:** MySQL Workbench, Local Instance 3306 - Warning - not supported, Sat Apr 6 14:12.
- Schemas:** DBMSProject is selected.
- Code Editor:** Displays SQL scripts for creating ParkingLotMaintenance and ParkingSpotHistory tables. The ParkingLotMaintenance table has columns: ParkingLotMaintenanceId, MaintenanceId, RequestId, EmployeeId. The ParkingSpotHistory table has columns: PLHMaintenanceId, PLHStatus, PLHDate.
- Status Bar:** 100%, 41/262 | 4 errors found.
- Action Output:** Shows a table with columns: Time, Action, Response, Duration / Fetch Time.
- Bottom:** SQL script saved to 'Users/jchathurakumara/Documents/DATABASE project/new Script/PLM.sql'.

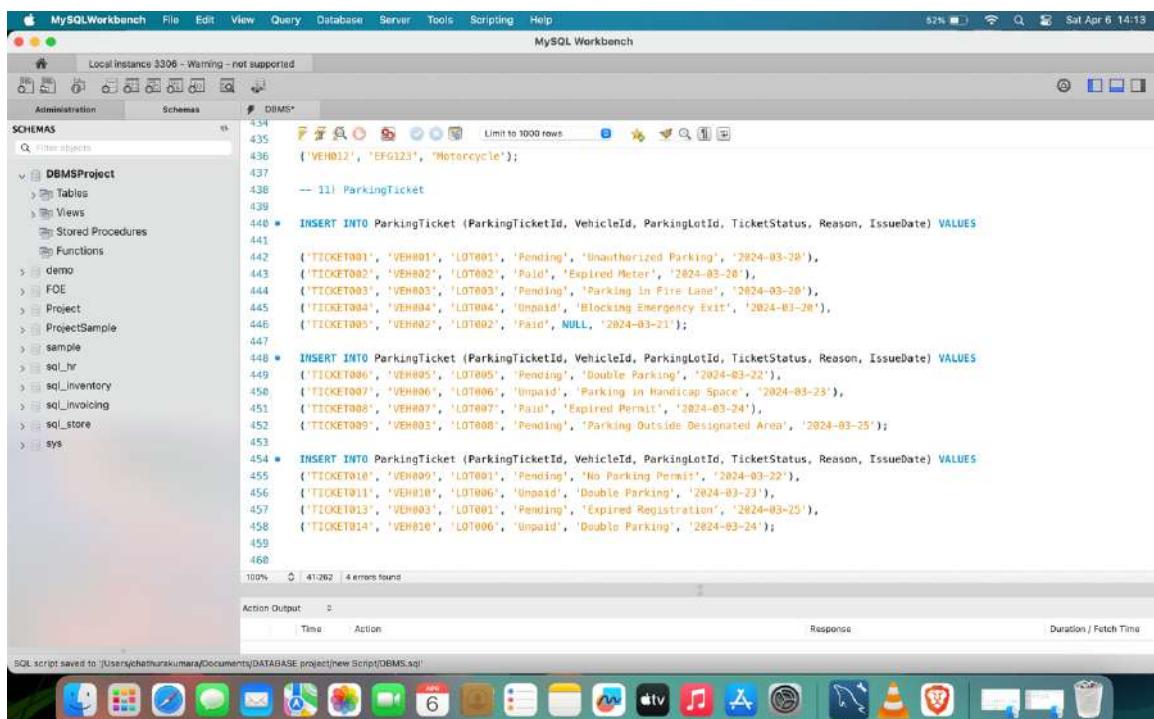
❖ Vehicle



The screenshot shows the MySQL Workbench interface with the 'Vehicle' schema selected. The SQL editor pane displays a script for creating the Vehicle table and inserting 10 rows of data. The status bar at the bottom indicates the script was saved to 'DBMS.sql'.

```
417 -- 10) Vehicle
418
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 ('VEH009', 'VWX234', 'Truck'),
434 ('VEH010', 'YZA567', 'Car'),
435 ('VEH011', 'BCD890', 'SUV'),
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
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
```

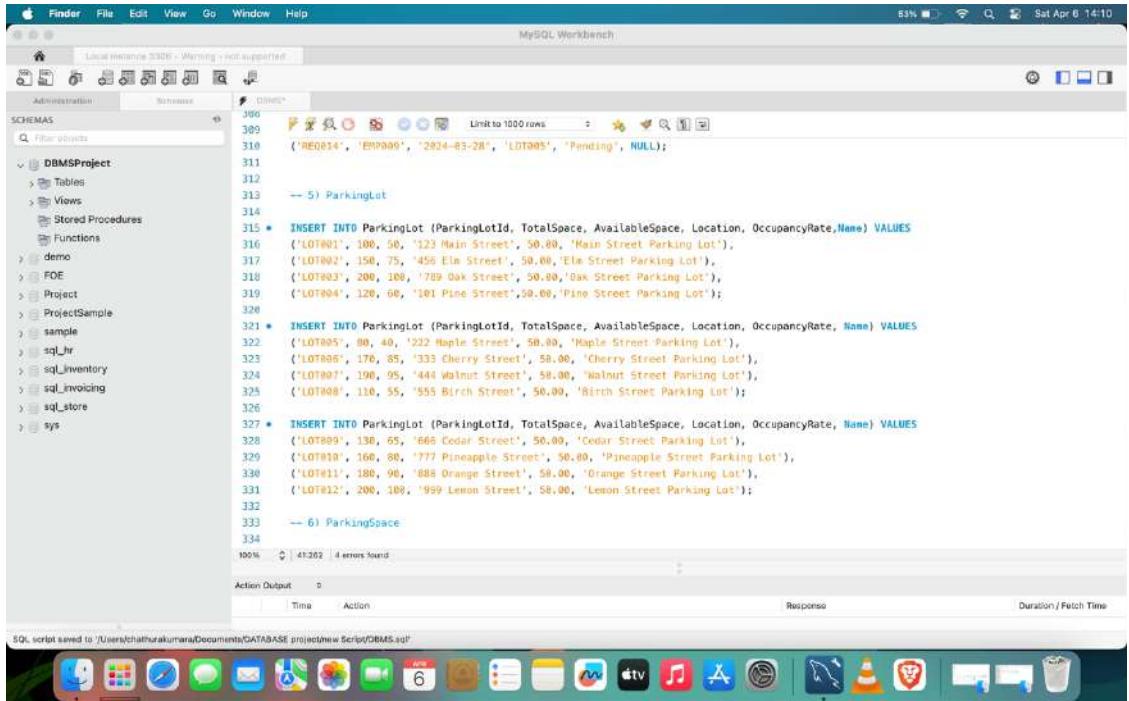
❖ Parking Ticket



The screenshot shows the MySQL Workbench interface with the 'ParkingTicket' schema selected. The SQL editor pane displays a script for creating the ParkingTicket table and inserting 11 rows of data. The status bar at the bottom indicates the script was saved to 'DBMS.sql'.

```
434
435 ('VEH012', 'EFG123', 'Motorcycle');
436
437 -- 11) ParkingTicket
438
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, '2024-03-20');
447
448 * INSERT INTO ParkingTicket (ParkingTicketId, VehicleId, ParkingLotId, TicketStatus, Reason, IssueDate) VALUES
449 ('TICKET006', 'VEH005', 'LOT005', 'Pending', 'Double Parking', '2024-03-22'),
450 ('TICKET007', 'VEH006', 'LOT006', 'Unpaid', 'Parking in Handicap Space', '2024-03-23'),
451 ('TICKET008', 'VEH007', 'LOT007', 'Paid', 'Expired Permit', '2024-03-24'),
452 ('TICKET009', 'VEH003', 'LOT008', 'Pending', 'Parking Outside Designated Area', '2024-03-25');
453
454 * INSERT INTO ParkingTicket (ParkingTicketId, VehicleId, ParkingLotId, TicketStatus, Reason, IssueDate) VALUES
455 ('TICKET010', 'VEH009', 'LOT001', 'Pending', 'No Parking Permit', '2024-03-22'),
456 ('TICKET011', 'VEH010', 'LOT006', 'Unpaid', 'Double Parking', '2024-03-23'),
457 ('TICKET012', 'VEH003', 'LOT001', 'Pending', 'Expired Registration', '2024-03-25'),
458 ('TICKET013', 'VEH010', 'LOT006', 'Unpaid', 'Double Parking', '2024-03-24');
459
460
```

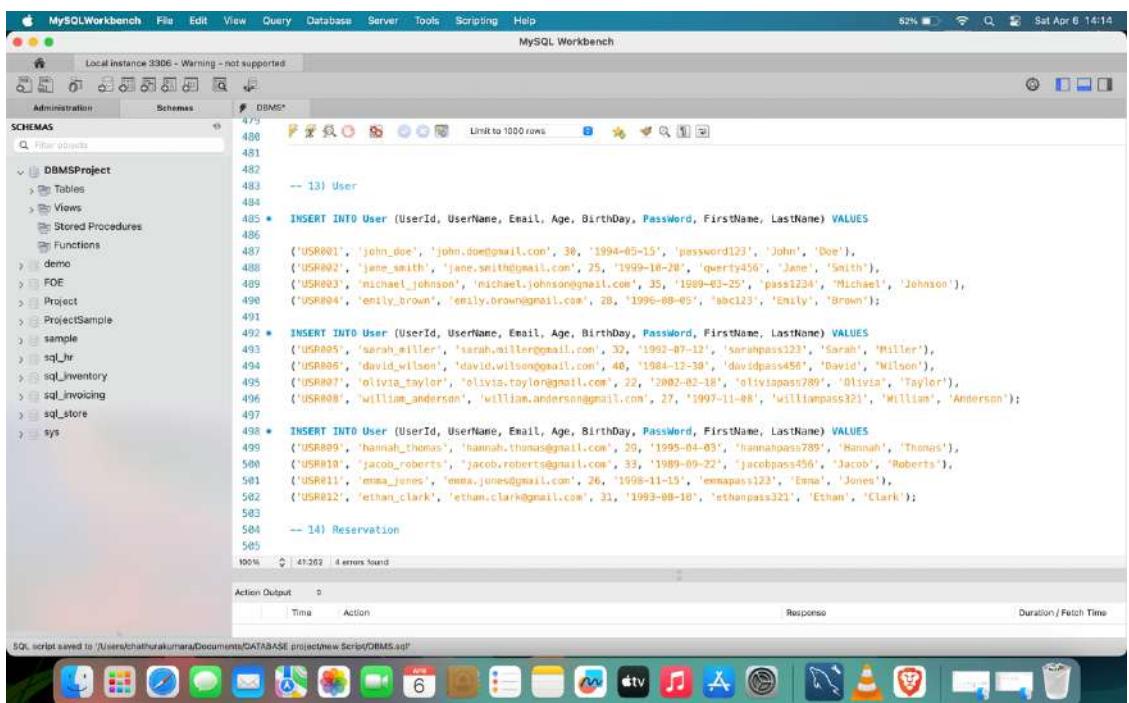
❖ Payment



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 a SQL script for creating parking lots in a schema named 'DBMSProject'. The script includes several INSERT statements for tables like 'ParkingLot' and 'ParkingSpace'. The status bar at the bottom shows '100%' completion and '4 errors found'. The Mac OS X dock is visible at the bottom of the screen.

```
309
310 ('REQ014', 'EMP009', '2924-03-28', 'LOT005', 'Pending', NULL);
311
312 -- 5) ParkingLot
313
314 * INSERT INTO ParkingLot (ParkingLotId, TotalSpace, AvailableSpace, Location, OccupancyRate, Name) VALUES
315 ('LOT001', 100, 50, '123 Main Street', 50.00, 'Main Street Parking Lot'),
316 ('LOT002', 150, 75, '456 Elm Street', 50.00, 'Elm Street Parking Lot'),
317 ('LOT003', 200, 100, '789 Oak Street', 50.00, 'Oak Street Parking Lot'),
318 ('LOT004', 120, 60, '101 Pine Street', 50.00, 'Pine Street Parking Lot'),
319
320 * INSERT INTO ParkingLot (ParkingLotId, TotalSpace, AvailableSpace, Location, OccupancyRate, Name) VALUES
321 ('LOT005', 80, 40, '222 Maple Street', 50.00, 'Maple Street Parking Lot'),
322 ('LOT006', 170, 85, '333 Cherry Street', 50.00, 'Cherry Street Parking Lot'),
323 ('LOT007', 190, 95, '444 Walnut Street', 50.00, 'Walnut Street Parking Lot'),
324 ('LOT008', 110, 55, '555 Birch Street', 50.00, 'Birch Street Parking Lot'),
325
326 * INSERT INTO ParkingLot (ParkingLotId, TotalSpace, AvailableSpace, Location, OccupancyRate, Name) VALUES
327 ('LOT009', 130, 65, '666 Cedar Street', 50.00, 'Cedar Street Parking Lot'),
328 ('LOT010', 160, 80, '777 Pineapple Street', 50.00, 'Pineapple Street Parking Lot'),
329 ('LOT011', 180, 90, '888 Orange Street', 50.00, 'Orange Street Parking Lot'),
330 ('LOT012', 200, 100, '999 Lemon Street', 50.00, 'Lemon Street Parking Lot'),
331
332 -- 6) ParkingSpace
333
334
```

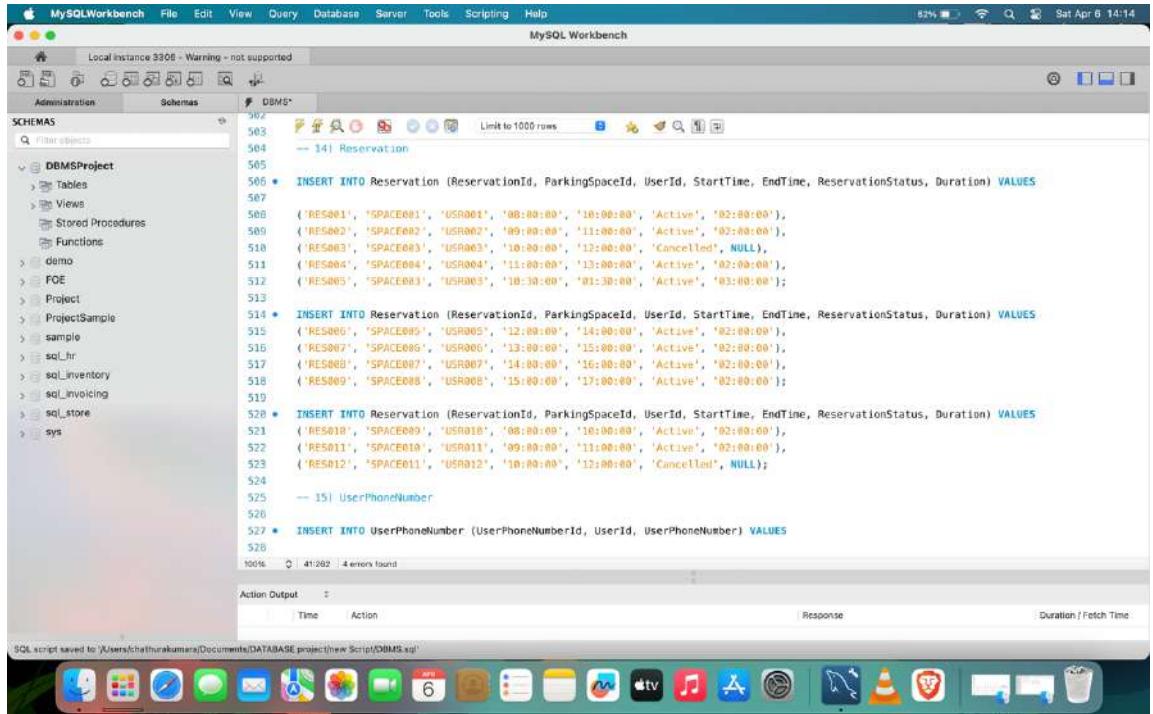
❖ User



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 14:14'. The main window displays a SQL script for creating users in a schema named 'DBMSProject'. The script includes several INSERT statements for the 'User' table. The status bar at the bottom shows '100%' completion and '4 errors found'. The Mac OS X dock is visible at the bottom of the screen.

```
479
480
481
482 -- 13) User
483
484 * INSERT INTO User (UserId, UserName, Email, Age, BirthDay, PassWord, FirstName, LastName) VALUES
485 ('USR001', 'john_doe', 'john.doe@gmail.com', 30, '1994-05-15', 'password123', 'John', 'Doe'),
486 ('USR002', 'jane_smith', 'jane.smith@gmail.com', 25, '1999-10-28', 'qwertyp456', 'Jane', 'Smith'),
487 ('USR003', 'michael_johnson', 'michael.johnson@gmail.com', 35, '1989-03-25', 'pass1234', 'Michael', 'Johnson'),
488 ('USR004', 'emily_brown', 'emily.brown@gmail.com', 20, '1996-08-05', 'abc123', 'Emily', 'Brown');
489
490 * INSERT INTO User (UserId, UserName, Email, Age, BirthDay, PassWord, FirstName, LastName) VALUES
491 ('USR005', 'sarah_miller', 'sarah.miller@gmail.com', 32, '1993-07-12', 'sarahpass123', 'Sarah', 'Miller'),
492 ('USR006', 'david_wilson', 'david.wilson@gmail.com', 40, '1984-12-30', 'davidpass4567', 'David', 'Wilson'),
493 ('USR007', 'olivia_taylor', 'olivia.taylor@gmail.com', 22, '2002-02-18', 'oliviapass789', 'Olivia', 'Taylor'),
494 ('USR008', 'william_anderson', 'william.anderson@gmail.com', 27, '1997-11-08', 'williampass321', 'William', 'Anderson');
495
496 * INSERT INTO User (UserId, UserName, Email, Age, BirthDay, PassWord, FirstName, LastName) VALUES
497 ('USR009', 'hammad_thomas', 'hammad.thomas@gmail.com', 20, '1995-04-05', 'hammadpass789', 'Hammad', 'Thomas'),
498 ('USR010', 'jacob_roberts', 'jacob.roberts@gmail.com', 33, '1989-09-22', 'jacobpass456', 'Jacob', 'Roberts'),
499 ('USR011', 'emma_jones', 'emma.jones@gmail.com', 26, '1998-11-15', 'emmapass123', 'Emma', 'Jones'),
500 ('USR012', 'ethan_clark', 'ethan.clark@gmail.com', 31, '1993-08-10', 'ethanpass321', 'Ethan', 'Clark');
501
502 -- 14) Reservation
503
504
```

❖ Reservation



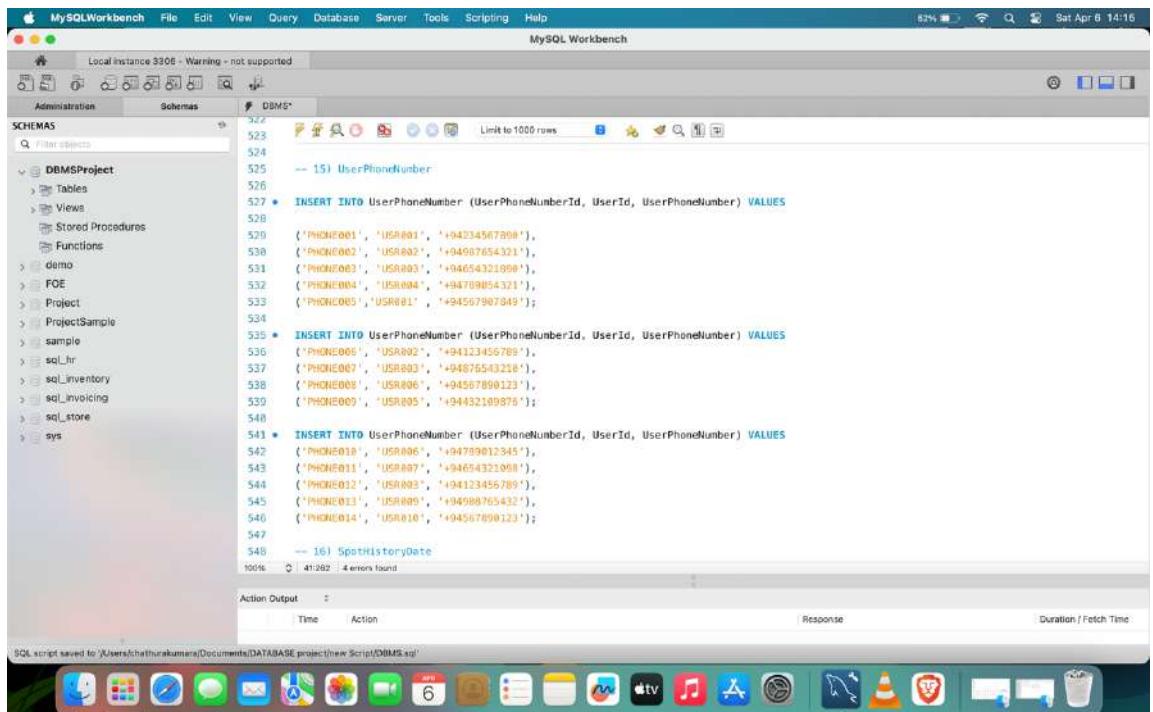
The screenshot shows the MySQL Workbench interface with the following SQL script:

```
502
503
504 -- 141 Reservation
505
506 • INSERT INTO Reservation (ReservationId, ParkingSpaceId, UserId, StartTime, EndTime, ReservationStatus, Duration) VALUES
507
508 ('RES001', 'SPACE001', 'USR001', '08:00:00', '10:00:00', 'Active', '02:00:00'),
509 ('RES002', 'SPACE002', 'USR002', '09:00:00', '11:00:00', 'Active', '02:00:00'),
510 ('RES003', 'SPACE003', 'USR003', '10:00:00', '12:00:00', 'Cancelled', NULL),
511 ('RES004', 'SPACE004', 'USR004', '11:00:00', '13:00:00', 'Active', '02:00:00'),
512 ('RES005', 'SPACE005', 'USR005', '10:30:00', '01:30:00', 'Active', '00:30:00');
513
514 • INSERT INTO Reservation (ReservationId, ParkingSpaceId, UserId, StartTime, EndTime, ReservationStatus, Duration) VALUES
515 ('RES006', 'SPACE006', 'USR006', '12:00:00', '14:00:00', 'Active', '02:00:00'),
516 ('RES007', 'SPACE007', 'USR007', '13:00:00', '15:00:00', 'Active', '02:00:00'),
517 ('RES008', 'SPACE008', 'USR008', '14:00:00', '16:00:00', 'Active', '02:00:00'),
518 ('RES009', 'SPACE009', 'USR009', '15:00:00', '17:00:00', 'Active', '02:00:00');
519
520 • INSERT INTO Reservation (ReservationId, ParkingSpaceId, UserId, StartTime, EndTime, ReservationStatus, Duration) VALUES
521 ('RES010', 'SPACE010', 'USR010', '08:00:00', '10:00:00', 'Active', '02:00:00'),
522 ('RES011', 'SPACE011', 'USR011', '09:00:00', '11:00:00', 'Active', '02:00:00),
523 ('RES012', 'SPACE012', 'USR012', '10:00:00', '12:00:00', 'Cancelled', NULL);
524
525 -- 151 UserPhoneNumber
526
527 • INSERT INTO UserPhoneNumber (UserPhoneNumberId, UserId, UserPhoneNumber) VALUES
528
1006 0 | 41/262 | 4 errors found
```

Action Output Time Action Response Duration / Fetch Time

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

❖ Phone Number



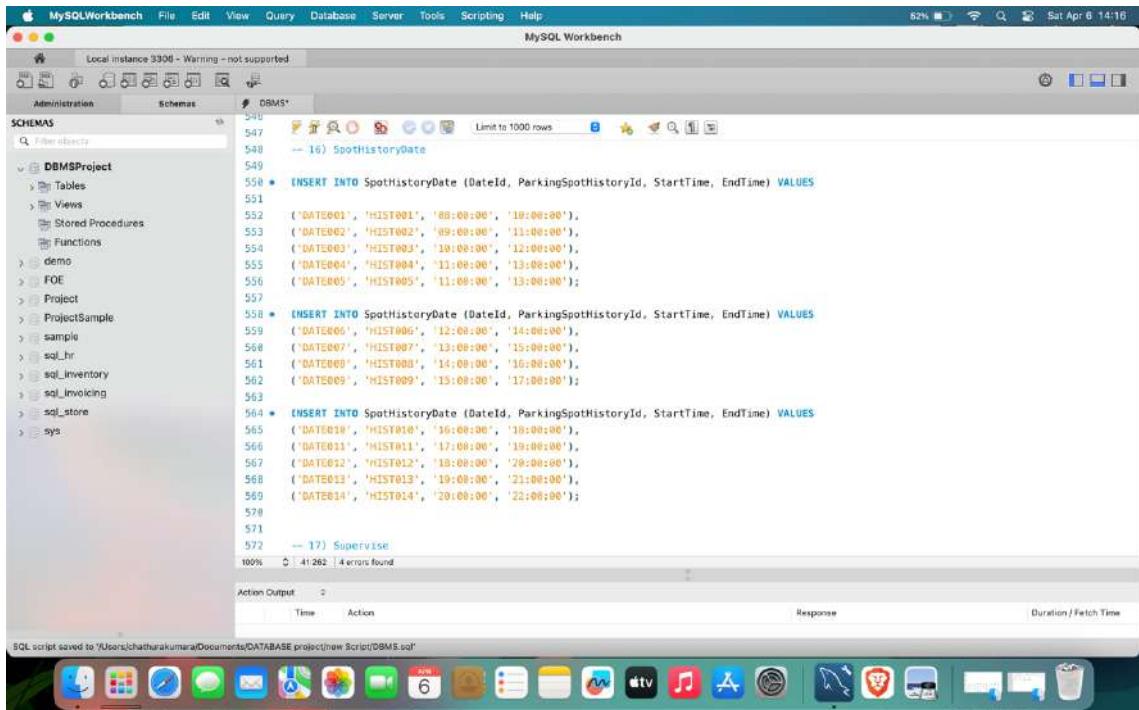
The screenshot shows the MySQL Workbench interface with the following SQL script:

```
522
523
524 -- 151 UserPhoneNumber
525
526
527 • INSERT INTO UserPhoneNumber (UserPhoneNumberId, UserId, UserPhoneNumber) VALUES
528
529 ('PHONE001', 'USR001', '+94234567899'),
530 ('PHONE002', 'USR002', '+94907654321'),
531 ('PHONE003', 'USR003', '+94654321099'),
532 ('PHONE004', 'USR004', '+94789054321'),
533 ('PHONE005', 'USR005', '+94567987649');
534
535 • INSERT INTO UserPhoneNumber (UserPhoneNumberId, UserId, UserPhoneNumber) VALUES
536 ('PHONE006', 'USR006', '+94123456789'),
537 ('PHONE007', 'USR007', '+94876543210'),
538 ('PHONE008', 'USR008', '+94567899123'),
539 ('PHONE009', 'USR009', '+94432109876');
540
541 • INSERT INTO UserPhoneNumber (UserPhoneNumberId, UserId, UserPhoneNumber) VALUES
542 ('PHONE010', 'USR006', '+9479012345'),
543 ('PHONE011', 'USR007', '+94654321098'),
544 ('PHONE012', 'USR008', '+94123456789'),
545 ('PHONE013', 'USR009', '+94989765432'),
546 ('PHONE014', 'USR010', '+94567890123');
547
548 -- 161 SpotHistoryDate
1006 0 | 41/262 | 4 errors found
```

Action Output Time Action Response Duration / Fetch Time

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

❖ SpotHistoryDate.

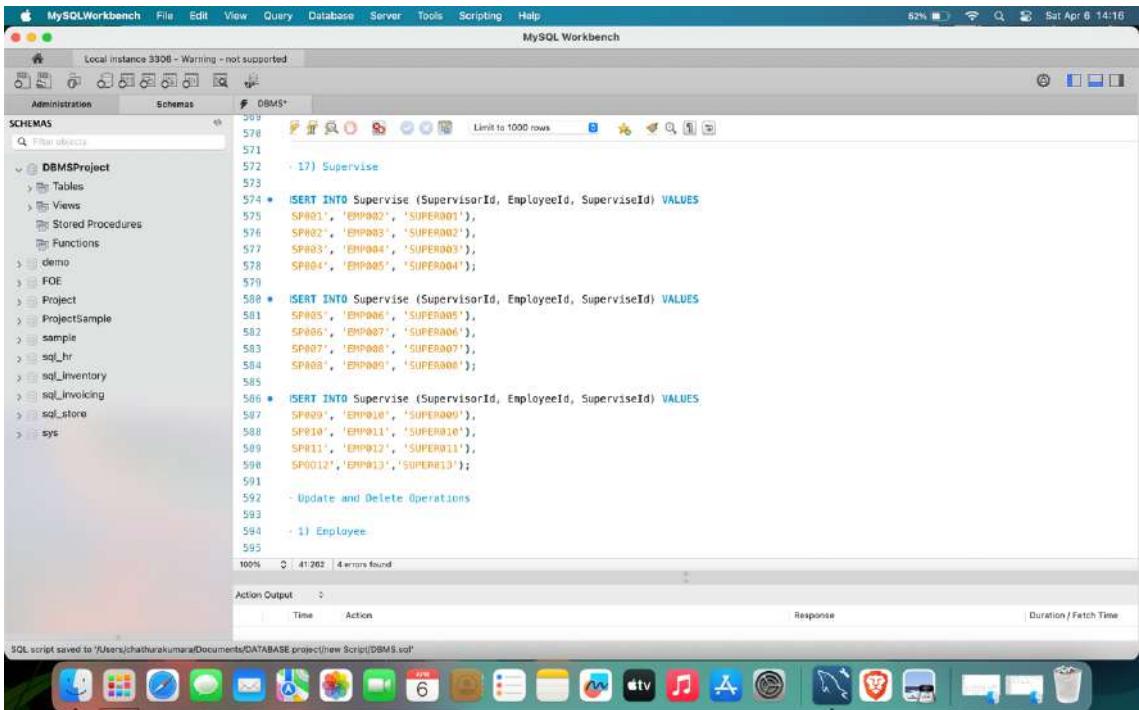


```

MySQLWorkbench  File  Edit  View  Query  Database  Server  Tools  Scripting  Help
MySQL Workbench
Local instance 3306 - Warning - not supported
MySQL Workbench
Administration  Schemas  DBMS*
SCHEMAS
Q Filter objects
DBMSProject
Tables
Views
Stored Procedures
Functions
demo
FOE
Project
ProjectSample
sample
sql_hr
sql_inventory
sql_invoicing
sql_store
sys
Limit to 1000 rows
547
548 -- 16) SpotHistoryDate
549
550 • INSERT INTO SpotHistoryDate (DateId, ParkingSpotHistoryId, StartTime, EndTime) VALUES
551
552 ('DATEB01', 'HISTB01', '08:00:00', '10:00:00'),
553 ('DATEB02', 'HISTB02', '09:00:00', '11:00:00'),
554 ('DATEB03', 'HISTB03', '10:00:00', '12:00:00'),
555 ('DATEB04', 'HISTB04', '11:00:00', '13:00:00'),
556 ('DATEB05', 'HISTB05', '11:00:00', '13:00:00');
557
558 • INSERT INTO SpotHistoryDate (DateId, ParkingSpotHistoryId, StartTime, EndTime) VALUES
559 ('DATEB06', 'HISTB06', '12:00:00', '14:00:00'),
560 ('DATEB07', 'HISTB07', '13:00:00', '15:00:00'),
561 ('DATEB08', 'HISTB08', '14:00:00', '16:00:00'),
562 ('DATEB09', 'HISTB09', '15:00:00', '17:00:00');
563
564 • INSERT INTO SpotHistoryDate (DateId, ParkingSpotHistoryId, StartTime, EndTime) VALUES
565 ('DATEB10', 'HISTB10', '16:00:00', '18:00:00'),
566 ('DATEB11', 'HISTB11', '17:00:00', '19:00:00'),
567 ('DATEB12', 'HISTB12', '18:00:00', '20:00:00'),
568 ('DATEB13', 'HISTB13', '19:00:00', '21:00:00'),
569 ('DATEB14', 'HISTB14', '20:00:00', '22:00:00');
570
571
572 -- 17) Supervise
100%  41:262  4 errors found
Action Output
Time Action Response Duration / Fetch Time
SQL script saved to '/Users/chathurakumar/Documents/DATABASE project/new Script/DBMS5.sql'

```

❖ Supervise



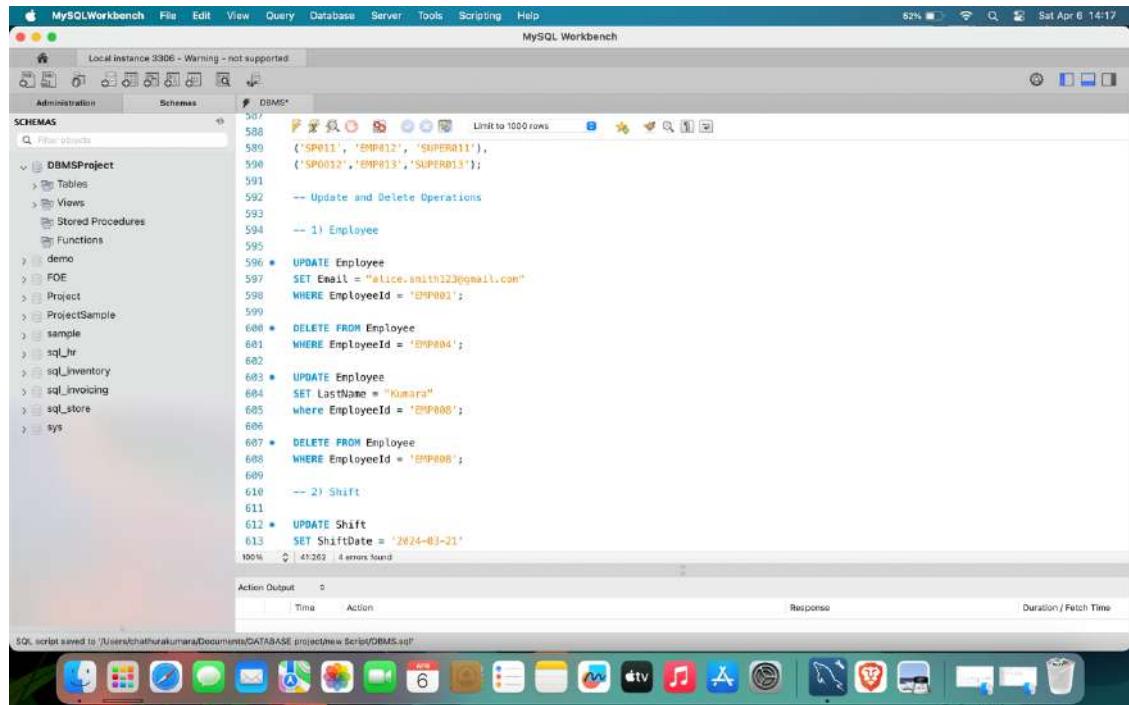
```

MySQLWorkbench  File  Edit  View  Query  Database  Server  Tools  Scripting  Help
MySQL Workbench
Local instance 3306 - Warning - not supported
MySQL Workbench
Administration  Schemas  DBMS*
SCHEMAS
Q Filter objects
DBMSProject
Tables
Views
Stored Procedures
Functions
demo
FOE
Project
ProjectSample
sample
sql_hr
sql_inventory
sql_invoicing
sql_store
sys
Limit to 1000 rows
573
574 -- 17) Supervise
575
576 • INSERT INTO Supervise (SupervisorId, EmployeeId, SupervisorId) VALUES
577 SP001, 'EMP002', 'SUPER001'),
578 SP002, 'EMP003', 'SUPER002'),
579 SP003, 'EMP004', 'SUPER003'),
580 SP004, 'EMP005', 'SUPER004');
581
582 • INSERT INTO Supervise (SupervisorId, EmployeeId, SupervisorId) VALUES
583 SP005, 'EMP006', 'SUPER005'),
584 SP006, 'EMP007', 'SUPER006'),
585 SP007, 'EMP008', 'SUPER007'),
586 SP008, 'EMP009', 'SUPER008');
587
588 • INSERT INTO Supervise (SupervisorId, EmployeeId, SupervisorId) VALUES
589 SP009, 'EMP010', 'SUPER009'),
590 SP010, 'EMP011', 'SUPER010'),
591 SP011, 'EMP012', 'SUPER011'),
592 SP012, 'EMP013', 'SUPER012');
593
594 - Update and Delete Operations
595
596 - 1) Employee
597
598
100%  41:262  4 errors found
Action Output
Time Action Response Duration / Fetch Time
SQL script saved to '/Users/chathurakumar/Documents/DATABASE project/new Script/DBMS5.sql'

```

3.4 Update And Delete

❖ Employee



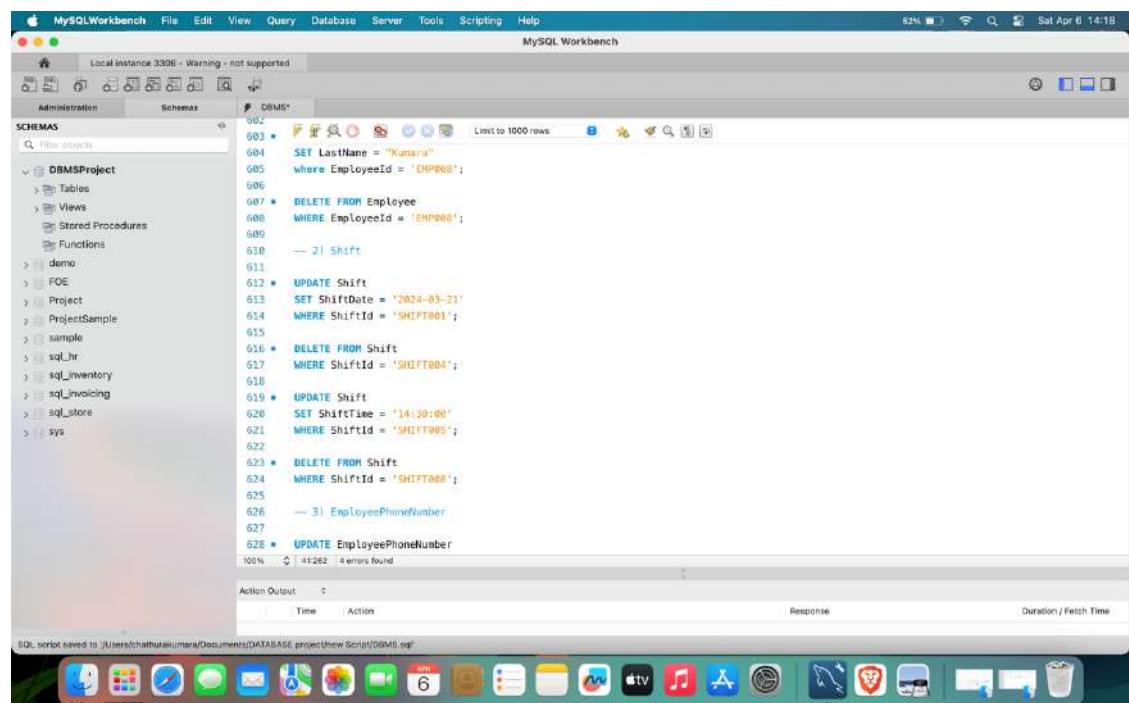
The screenshot shows the MySQL Workbench interface with the DBMS* schema selected. The code editor displays several SQL statements:

```
507
508  ('SP011', 'EMP012', 'SUPER011'),
509  ('SP002', 'EMP013', 'SUPER013');
510  -- Update and Delete Operations
511
512  --- 1) Employee
513
514  UPDATE Employee
515  SET Email = "alice.smith123@gmail.com"
516  WHERE EmployeeId = 'EMP001';
517
518  * DELETE FROM Employee
519  WHERE EmployeeId = 'EMP004';
520
521  --- 2) Shift
522
523  * UPDATE Shift
524  SET ShiftDate = '2024-03-21'
525
526  100% 41/262 4 errors found
```

Action Output: Time Action Response Duration / Fetch Time

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

❖ Shift



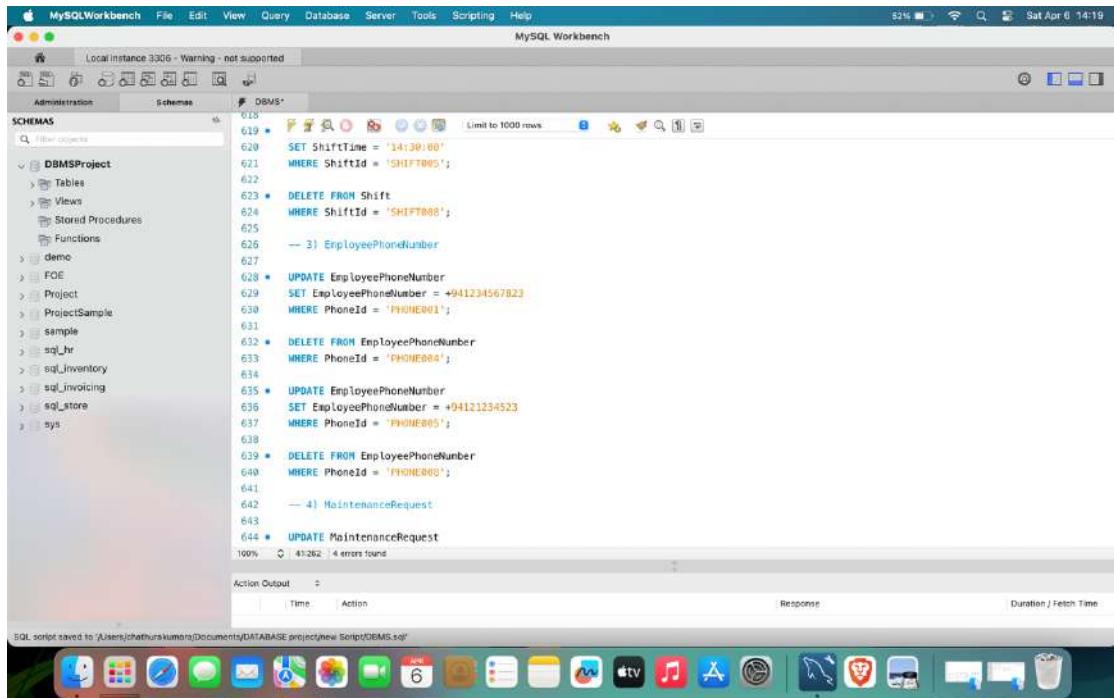
The screenshot shows the MySQL Workbench interface with the DBMS* schema selected. The code editor displays several SQL statements:

```
602
603  * UPDATE Shift
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'
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
629
630  100% 41/262 4 errors found
```

Action Output: Time Action Response Duration / Fetch Time

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

❖ Employee Phone Number



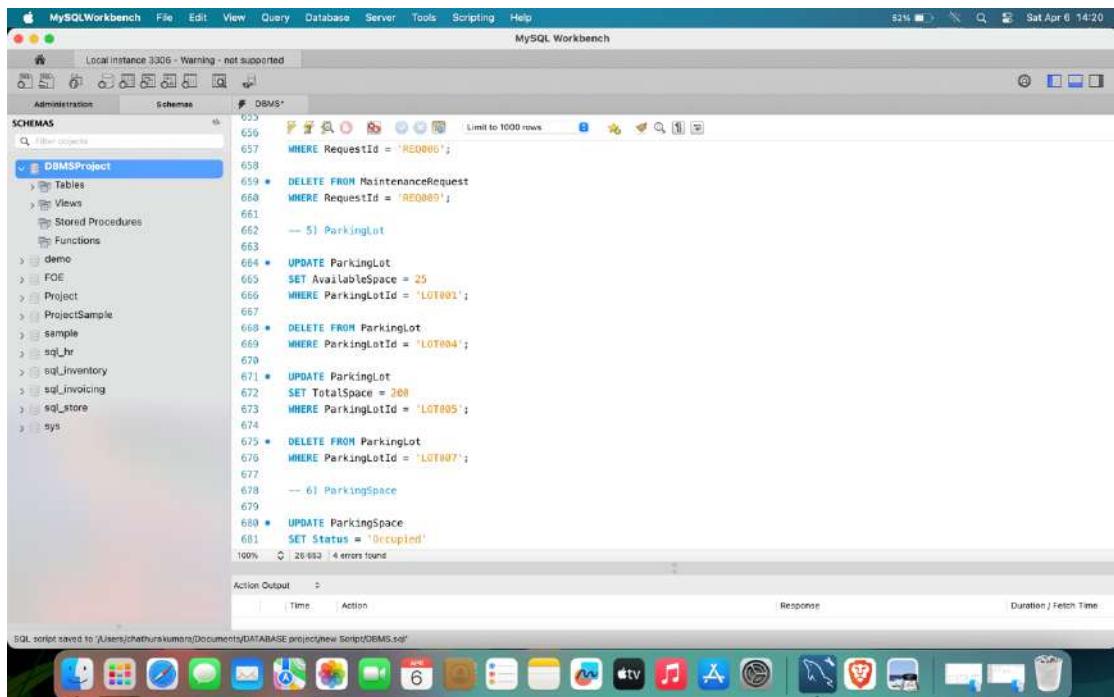
The screenshot shows the MySQL Workbench interface with the following SQL script:

```
018 * 
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 • DELETE FROM EmployeePhoneNumber;
632 WHERE PhoneId = 'PHONE004';
633 
634 • UPDATE EmployeePhoneNumber;
635 SET EmployeePhoneNumber = '+94111234523'
636 WHERE PhoneId = 'PHONE005';
637 
638 • DELETE FROM EmployeePhoneNumber;
639 WHERE PhoneId = 'PHONE006';
640 
641 -- 4) MaintenanceRequest
642 
643 • UPDATE MaintenanceRequest;
644 
100% 41-262 | 4 errors found
```

Action Output: Time Action Response Duration / Fetch Time

SQL script saved to '/Users/chathurika/Documents/DATABASE project/new ScriptDBMS.sql'

❖ Parking Lot



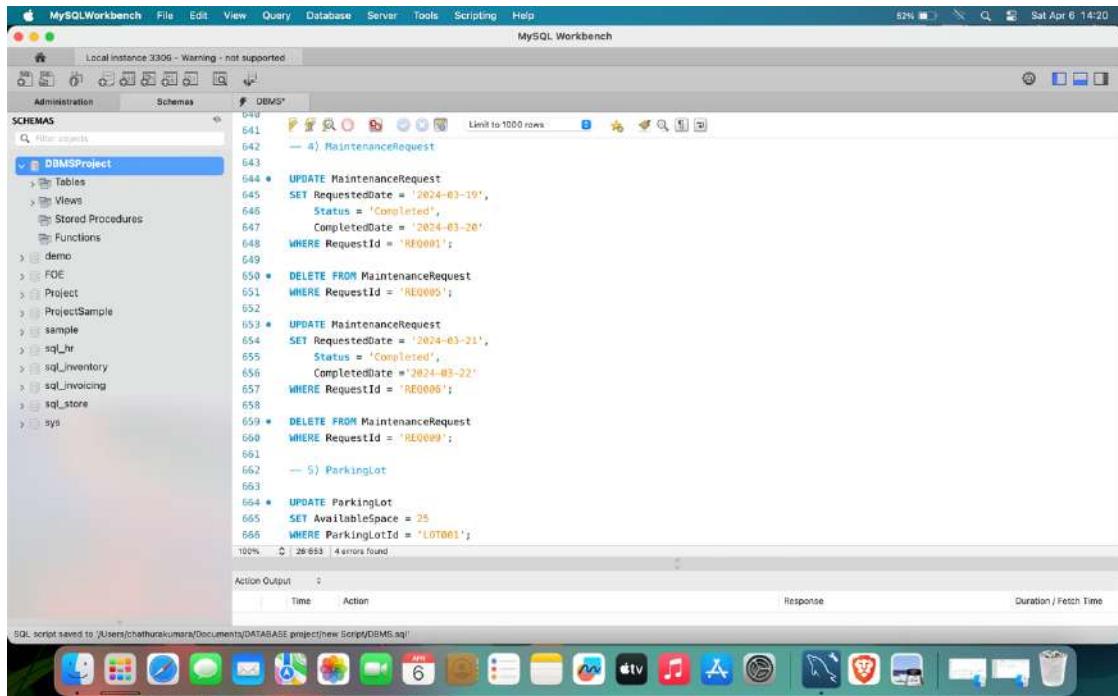
The screenshot shows the MySQL Workbench interface with the following SQL script:

```
033 * 
556 WHERE RequestId = 'REQ006';
557 
558 • DELETE FROM MaintenanceRequest;
559 WHERE RequestId = 'REQ009';
560 
561 -- 5) ParkingLot
562 
563 • UPDATE ParkingLot;
564 SET AvailableSpace = 25
565 WHERE ParkingLotId = 'LOT001';
566 
567 • DELETE FROM ParkingLot;
568 WHERE ParkingLotId = 'LOT004';
569 
570 • UPDATE ParkingLot;
571 SET TotalSpace = 208
572 WHERE ParkingLotId = 'LOT005';
573 
574 • DELETE FROM ParkingLot;
575 WHERE ParkingLotId = 'LOT007';
576 
577 -- 6) ParkingSpace
578 
579 • UPDATE ParkingSpace;
580 SET Status = 'Occupied';
581 
100% 26-682 | 4 errors found
```

Action Output: Time Action Response Duration / Fetch Time

SQL script saved to '/Users/chathurika/Documents/DATABASE project/new ScriptDBMS.sql'

❖ Maintenance Request



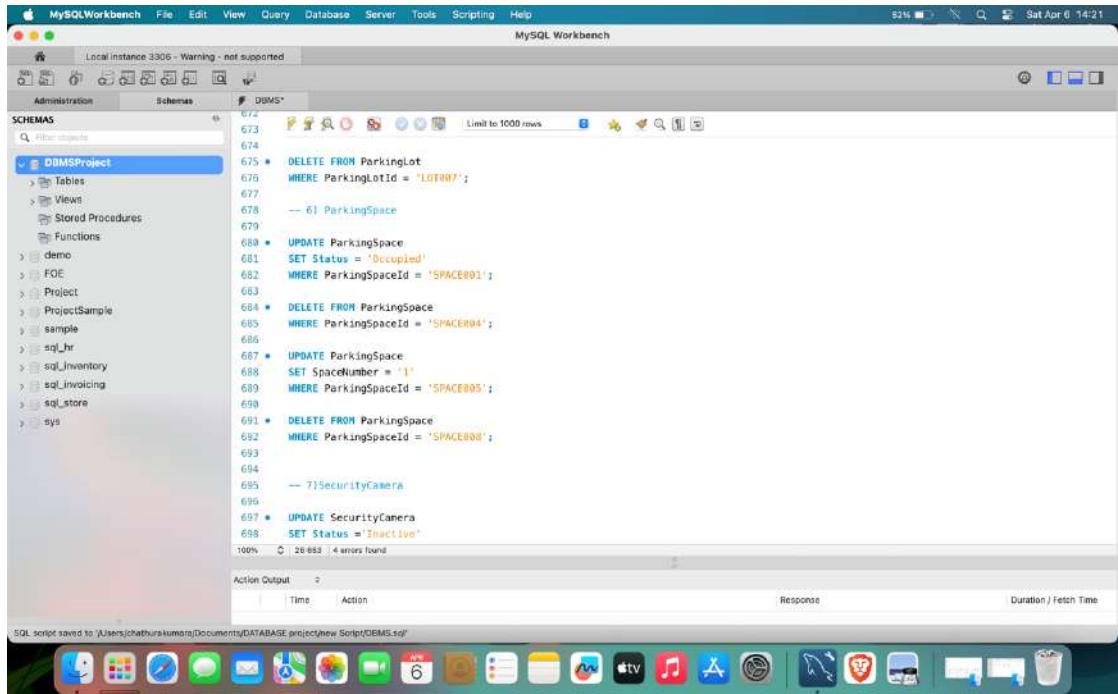
The screenshot shows the MySQL Workbench interface with a script editor window. The script contains several SQL statements for updating and deleting rows in the 'MaintenanceRequest' table. The statements set RequestedDate to '2024-03-19', Status to 'Completed', and CompletedDate to '2024-03-20' for RequestId 'RE0001'. They also set RequestedDate to '2024-03-21', Status to 'Completed', and CompletedDate to '2024-03-22' for RequestId 'RE0002'. Finally, they delete rows where RequestId is 'REQ009'.

```
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 = 'RE0001';
649
650 • DELETE FROM MaintenanceRequest
651   WHERE RequestId = 'RE0001';
652
653 • UPDATE MaintenanceRequest
654   SET RequestedDate = '2024-03-21',
655     Status = 'Completed',
656     CompletedDate = '2024-03-22'
657   WHERE RequestId = 'RE0002';
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% 0 26/63 | 4 errors found
```

Action Output: Time Action Response Duration / Fetch Time

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

❖ Parking Space



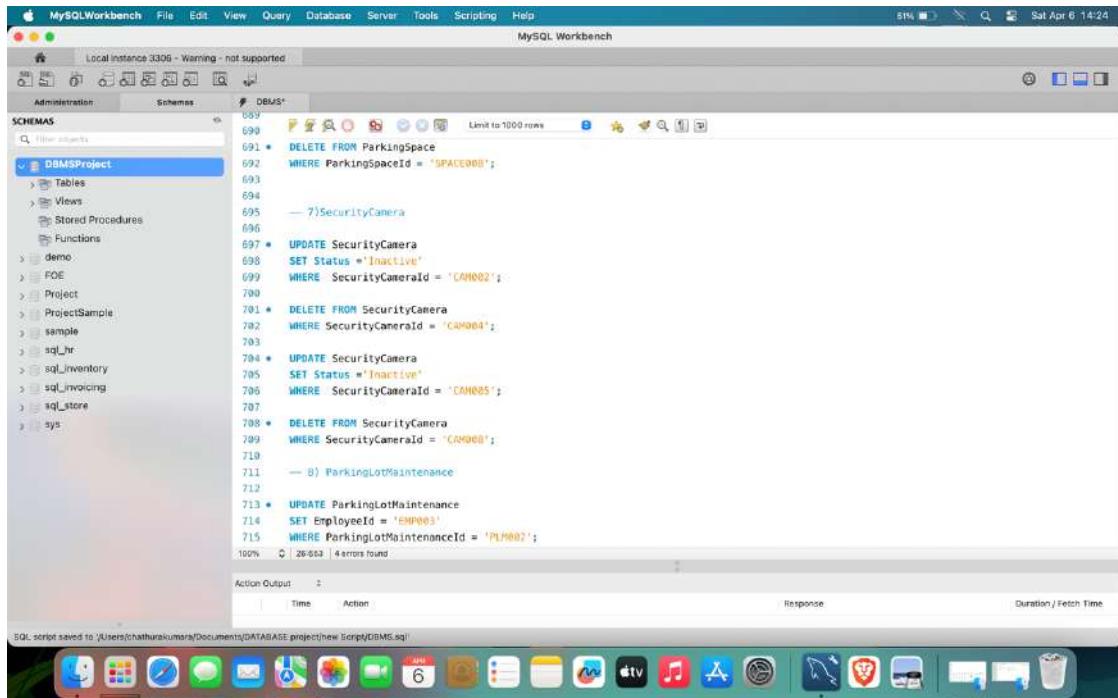
The screenshot shows the MySQL Workbench interface with a script editor window. The script contains several SQL statements for managing parking spaces. It includes deleting rows from the 'ParkingLot' table where ParkingLotId is 'LOT007', updating 'ParkingSpace' status to 'Occupied' for ParkingSpaceId 'SPACE001', deleting rows from 'ParkingSpace' where ParkingSpaceId is 'SPACE004', updating 'ParkingSpace' SpaceNumber to '1' for ParkingSpaceId 'SPACE005', deleting rows from 'ParkingSpace' where ParkingSpaceId is 'SPACE008', and updating 'SecurityCamera' status to 'Inactive' for SecurityCameraId 'SEC001'.

```
672
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'
```

Action Output: Time Action Response Duration / Fetch Time

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

❖ Security Camera

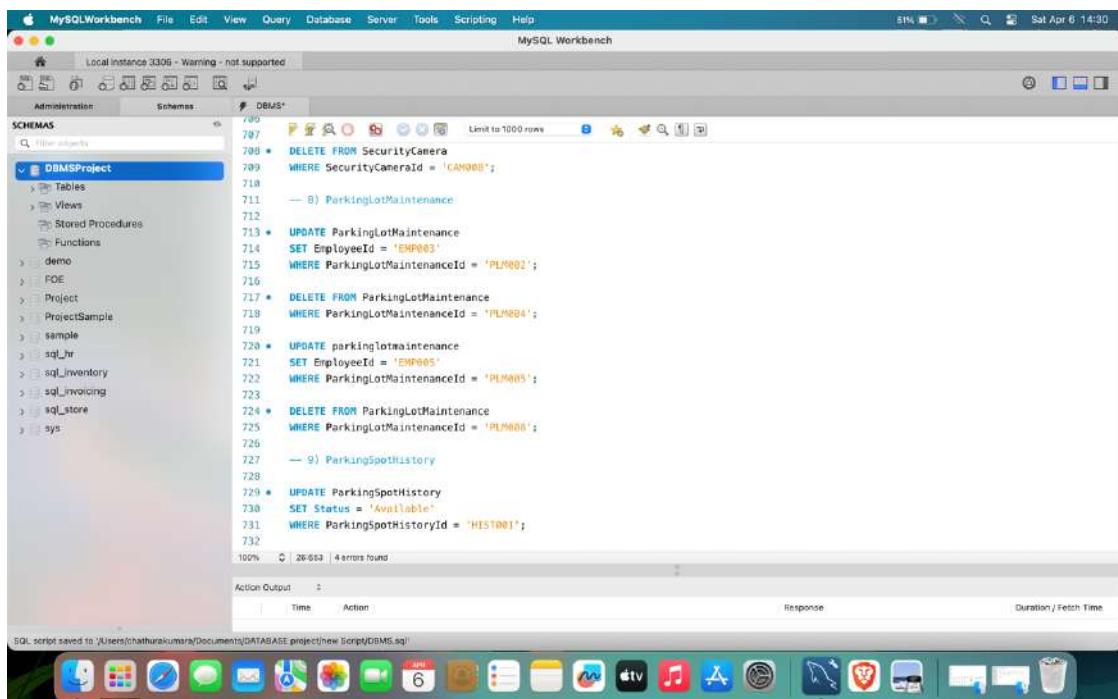


The screenshot shows the MySQL Workbench interface with a SQL editor window. The title bar reads "MySQL Workbench" and "Local Instance 3306 - Warning - not supported". The SQL editor contains the following code:

```
689
690 *   DELETE FROM ParkingSpace
691 WHERE ParkingSpaceId = 'SPACE008';
692
693 — 7)SecurityCamera
694
695 *   UPDATE SecurityCamera
696 SET Status = 'Inactive'
697 WHERE SecurityCameraId = 'CAM002';
698
699 *   DELETE FROM SecurityCamera
700 WHERE SecurityCameraId = 'CAM004';
701
702 *   UPDATE SecurityCamera
703 SET Status = 'Inactive'
704 WHERE SecurityCameraId = 'CAM005';
705
706 *   DELETE FROM SecurityCamera
707 WHERE SecurityCameraId = 'CAM008';
708
709 — 8) ParkingLotMaintenance
710
711 *   UPDATE ParkingLotMaintenance
712 SET EmployeeId = 'EMP003'
713 WHERE ParkingLotMaintenanceId = 'PLM002';
714
715 — 8) ParkingLotMaintenance
716
717 *   DELETE FROM ParkingLotMaintenance
718 WHERE ParkingLotMaintenanceId = 'PLM004';
719
720 *   UPDATE parkinglotmaintenance
721 SET EmployeeId = 'EMP005'
722 WHERE ParkingLotMaintenanceId = 'PLM005';
723
724 *   DELETE FROM ParkingLotMaintenance
725 WHERE ParkingLotMaintenanceId = 'PLM006';
726
727 — 9) ParkingSpotHistory
728
729 *   UPDATE ParkingSpotHistory
730 SET Status = 'Available'
731 WHERE ParkingSpotHistoryId = 'HIST001';
732
```

The status bar at the bottom indicates "100% 26/63 | 4 errors found". The toolbar below the editor includes icons for Save, Undo, Redo, Copy, Paste, Find, Replace, and others.

❖ Parking Lot Maintenance



The screenshot shows the MySQL Workbench interface with a SQL editor window. The title bar reads "MySQL Workbench" and "Local Instance 3306 - Warning - not supported". The SQL editor contains the following code:

```
780
781 *   DELETE FROM SecurityCamera
782 WHERE SecurityCameraId = 'CAM008';
783
784 — 8) ParkingLotMaintenance
785
786 *   UPDATE ParkingLotMaintenance
787 SET EmployeeId = 'EMP003'
788 WHERE ParkingLotMaintenanceId = 'PLM002';
789
790 *   DELETE FROM ParkingLotMaintenance
791 WHERE ParkingLotMaintenanceId = 'PLM004';
792
793 *   UPDATE parkinglotmaintenance
794 SET EmployeeId = 'EMP005'
795 WHERE ParkingLotMaintenanceId = 'PLM005';
796
797 *   DELETE FROM ParkingLotMaintenance
798 WHERE ParkingLotMaintenanceId = 'PLM006';
799
800 — 9) ParkingSpotHistory
801
802 *   UPDATE ParkingSpotHistory
803 SET Status = 'Available'
804 WHERE ParkingSpotHistoryId = 'HIST001';
805
```

The status bar at the bottom indicates "100% 26/63 | 4 errors found". The toolbar below the editor includes icons for Save, Undo, Redo, Copy, Paste, Find, Replace, and others.

❖ Vehicle

The screenshot shows the MySQL Workbench interface. The left sidebar displays the database schema with a tree view of tables, views, stored procedures, and functions under the DBMSProject database. The main pane shows the DBMS* schema with a table named History. A SQL query is being run against this table:

```
WHERE ParkingSpotHistoryId = 'HIST005';
DELETE FROM ParkingSpotHistory
WHERE ParkingSpaceId = 'HIST005';
-- 10) Vehicle
UPDATE Vehicle
SET VehicleType = 'Van'
WHERE VehicleId = 'VEH001';
DELETE FROM Vehicle
WHERE VehicleId = 'VEH004';
UPDATE Vehicle
SET VehicleType = 'Van'
WHERE VehicleId = 'VEH005';
-- 11) ParkingTicket
UPDATE ParkingTicket
```

The status bar at the bottom indicates 100% completion with 26,853 rows affected and 4 errors found.

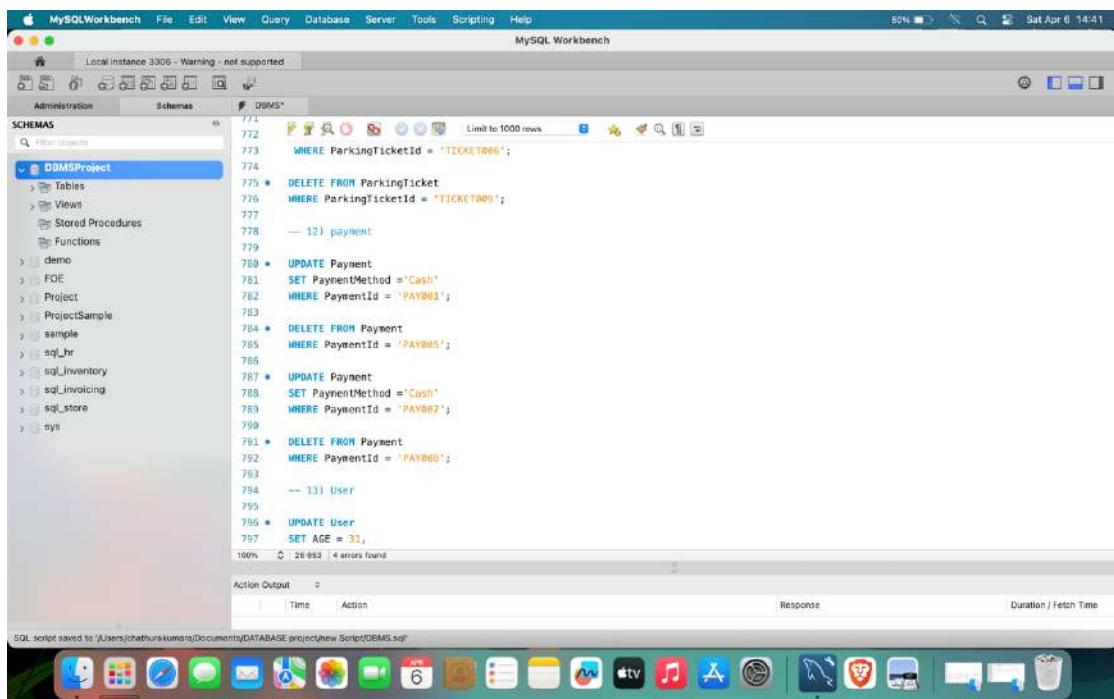
◆ ParkingTicket.

The screenshot shows the MySQL Workbench application window. The title bar indicates "MySQLWorkbench" and "Local instance 3306 - Warning - not supported". The main interface includes a toolbar with various icons, a navigation pane on the left listing databases like "DBMSProject", "Tables", "Views", etc., and a large central area for writing and executing SQL scripts. A scrollable text area displays the following SQL code:

```
755 • WHERE VehicleId = 'VEH0001';
756 *
757 WHERE VehicleId = 'VEH0001';
758
759 --- 11) ParkingTicket
760
761 • UPDATE ParkingTicket
762 SET TicketStatus = 'Paid',
763 Reason = NULL
764 WHERE ParkingTicketId = 'TICKET001';
765
766 • DELETE FROM ParkingTicket
767 WHERE ParkingTicketId = 'TICKET005';
768
769 • UPDATE ParkingTicket
770 SET TicketStatus = 'Pending',
771 Reason = NULL
772 WHERE ParkingTicketId = 'TICKET006';
773
774 • DELETE FROM ParkingTicket
775 WHERE ParkingTicketId = 'TICKET009';
776
777 --- 12) payment
778
779 • UPDATE Payment
780 SET PaymentMethod = 'Cash'
781
```

At the bottom, there are tabs for "Action Output", "Time", "Action", "Response", and "Duration / Fetch Time". A status bar at the bottom shows "SQL script saved to: /Users/charithura/Downloads/DATABASE.project/new Script/DEBM.sql".

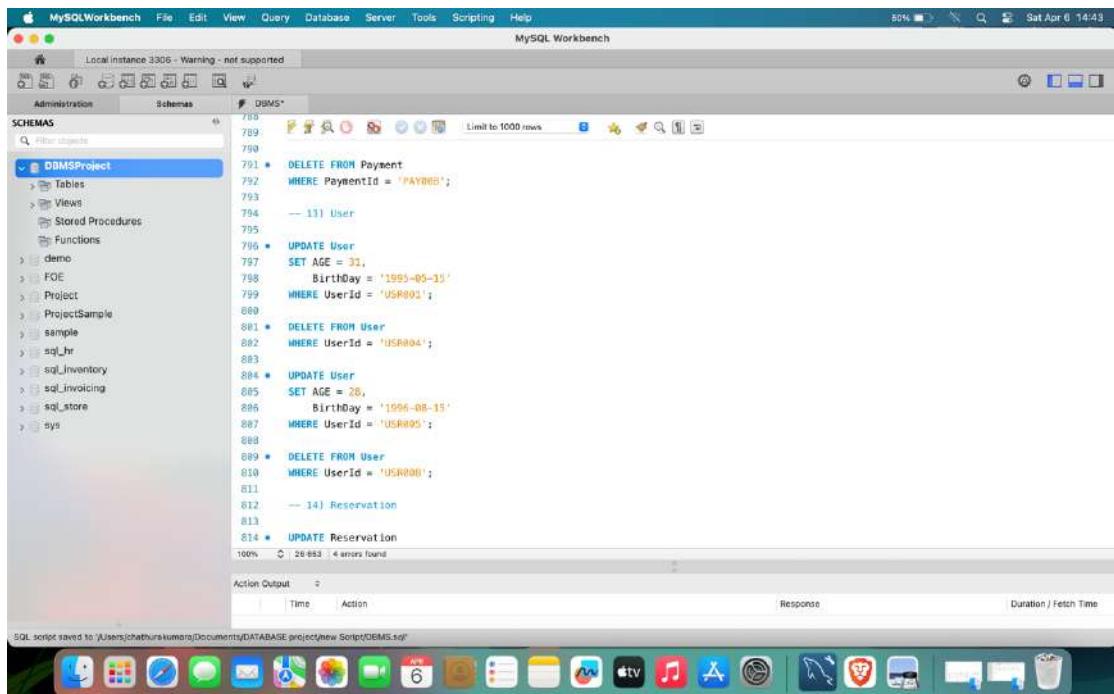
❖ Payment



The screenshot shows the MySQL Workbench interface with a script editor containing SQL code. The code is performing various operations on the 'Payment' table, including DELETE and UPDATE statements. The script is part of a larger project named 'DMSPProject'. The interface includes a navigation pane on the left, a toolbar at the top, and a status bar at the bottom indicating the date and time.

```
771  WHERE ParkingTicketId = 'TICKET006';
772
773 *  DELETE FROM ParkingTicket
774   WHERE ParkingTicketId = 'TICKET009';
775
776   -- 12) payment
777
778 *  UPDATE Payment
779   SET PaymentMethod = 'Cash'
780   WHERE PaymentId = 'PAY001';
781
782 *  UPDATE Payment
783   WHERE PaymentId = 'PAY005';
784
785 *  UPDATE Payment
786   SET PaymentMethod = 'Cash'
787   WHERE PaymentId = 'PAY007';
788
789 *  UPDATE Payment
790   WHERE PaymentId = 'PAY009';
791
792   -- 13) User
793
794 *  UPDATE User
795   SET AGE = 33;
796
797
100%  26 653 | 4 errors found
```

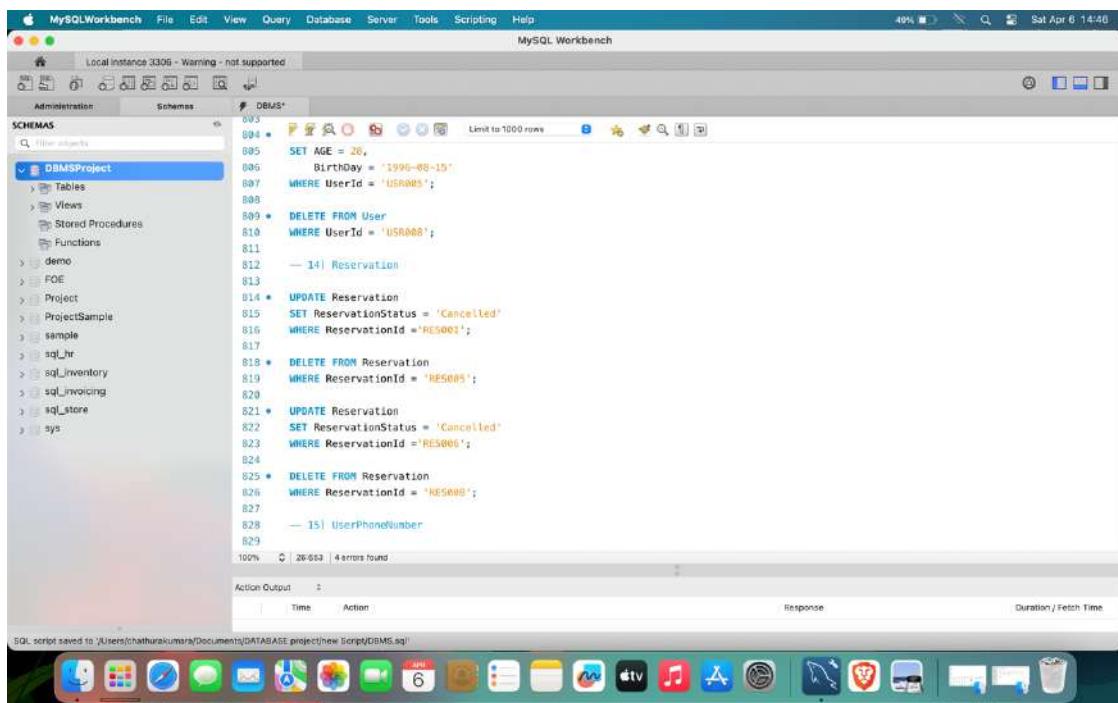
❖ User



The screenshot shows the MySQL Workbench interface with a script editor containing SQL code for User operations. The code includes DELETE and UPDATE statements on the 'User' table. Similar to the previous screenshot, it's part of the 'DMSPProject'. The interface and toolbars are consistent with the first screenshot.

```
765
766 *  DELETE FROM Payment
767   WHERE PaymentId = 'PAY008';
768
769   -- 13) User
770
771 *  UPDATE User
772   SET AGE = 33,
773     BirthDay = '1995-05-15'
774   WHERE UserId = 'USR001';
775
776 *  DELETE FROM User
777   WHERE UserId = 'USR004';
778
779 *  UPDATE User
780   SET AGE = 28,
781     BirthDay = '1996-08-15'
782   WHERE UserId = 'USR005';
783
784 *  DELETE FROM User
785   WHERE UserId = 'USR000';
786
787   -- 14) Reservation
788
789 *  UPDATE Reservation
790   WHERE UserId = 'USR000';
791
792
100%  26 653 | 4 errors found
```

❖ Reservation

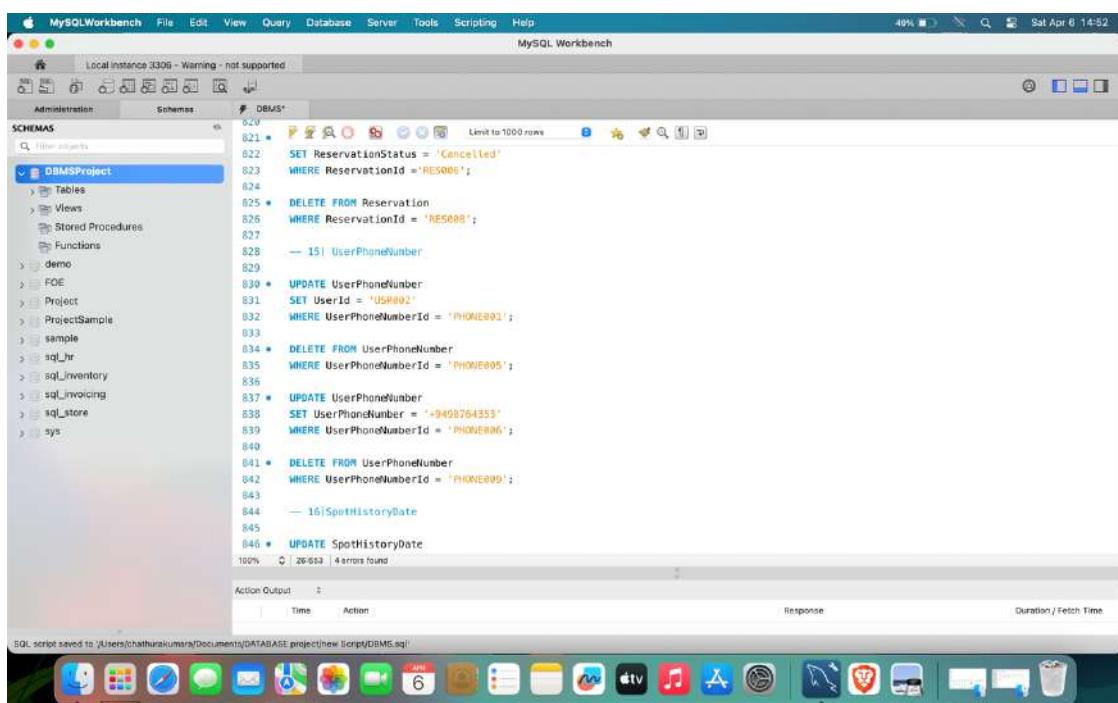


The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The main window displays a script editor with several SQL statements. The schema 'DBMSProject' is selected in the left sidebar. The SQL code includes:

```
803
804 • SET AGE = 20,
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
830
831
832
833
834 • UPDATE UserPhoneNumber
835     SET UserId = 'USR002'
836     WHERE UserPhoneNumberId = 'PHONE001';
837
838 • UPDATE UserPhoneNumber
839     SET UserPhoneNumber = '+91987654321'
840     WHERE UserPhoneNumberId = 'PHONE005';
841
842 • UPDATE UserPhoneNumber
843     WHERE UserPhoneNumberId = 'PHONE009';
844
845
846 • UPDATE SpotHistoryDate
```

The status bar at the bottom indicates '100% 26/63 | 4 errors found'. The Mac OS X dock is visible at the bottom of the screen.

❖ Phone Number

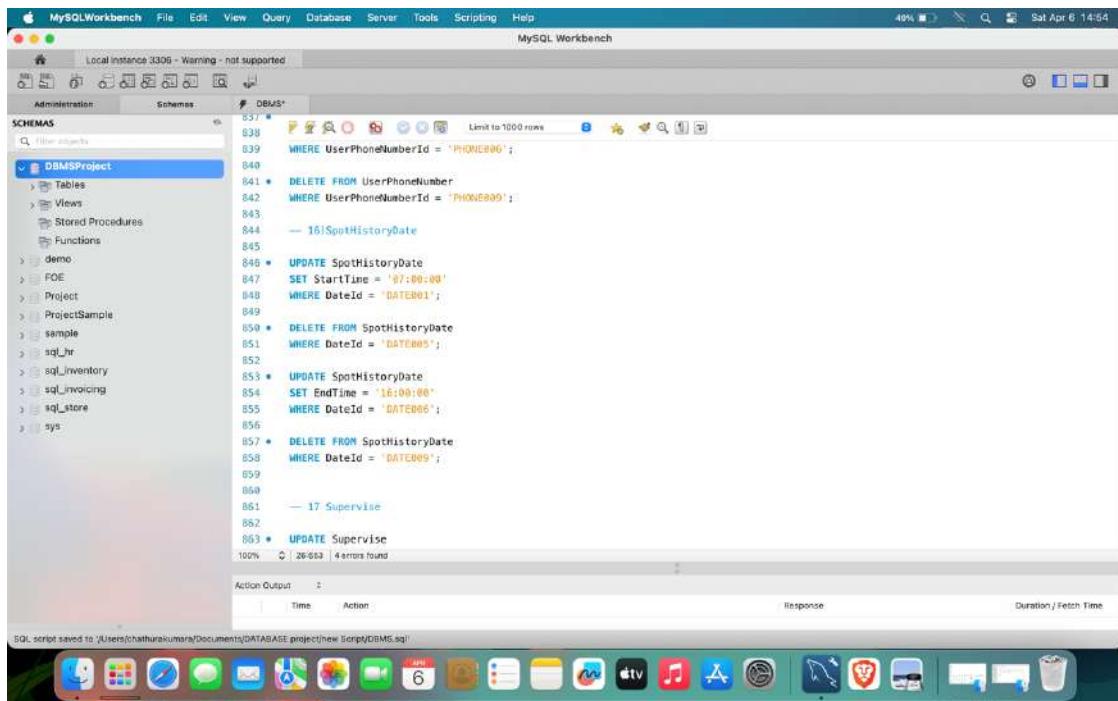


The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The main window displays a script editor with several SQL statements. The schema 'DBMSProject' is selected in the left sidebar. The SQL code includes:

```
820
821 • SET ReservationStatus = 'Cancelled'
822     WHERE ReservationId = 'RES006';
823
824 • DELETE FROM Reservation
825     WHERE ReservationId = 'RES008';
826
827 — 15) UserPhoneNumber
828
829
830
831
832
833
834 • UPDATE UserPhoneNumber
835     SET UserId = 'USR002'
836     WHERE UserPhoneNumberId = 'PHONE001';
837
838 • UPDATE UserPhoneNumber
839     SET UserPhoneNumber = '+91987654321'
840     WHERE UserPhoneNumberId = 'PHONE005';
841
842 • UPDATE UserPhoneNumber
843     WHERE UserPhoneNumberId = 'PHONE009';
844
845
846 • UPDATE SpotHistoryDate
```

The status bar at the bottom indicates '100% 26/63 | 4 errors found'. The Mac OS X dock is visible at the bottom of the screen.

❖ SpotHistoryDate

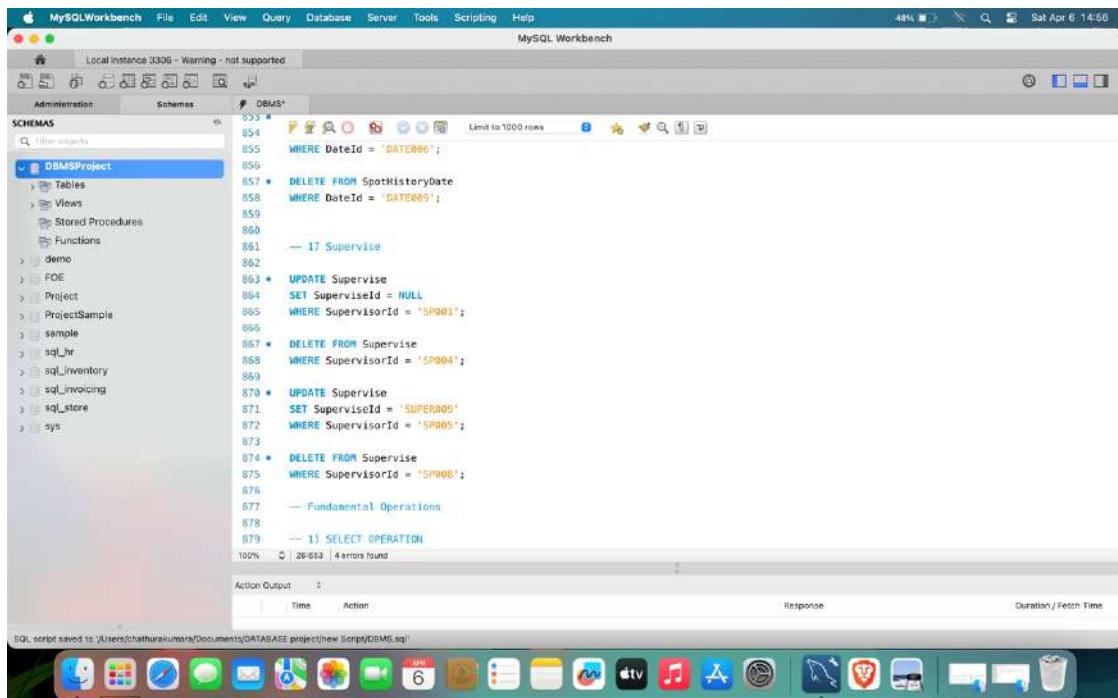


The screenshot shows the MySQL Workbench interface with the DBMS* tab selected. The left sidebar displays the schema tree under the DBMSProject schema. The main pane contains the following SQL script:

```
837 * WHERE UserPhoneNumberId = 'PHONE000';
838
839 WHERE UserPhoneNumberId = 'PHONE001';
840
841 * DELETE FROM UserPhoneNumber
842 WHERE UserPhoneNumberId = 'PHONE001';
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 = 'DATE005';
856
857 * DELETE FROM SpotHistoryDate
858 WHERE DateId = 'DATE009';
859
860
861 --- 17 Supervise
862
863 * UPDATE Supervise
100% 26-63 | 4 errors found
```

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

❖ Supervise



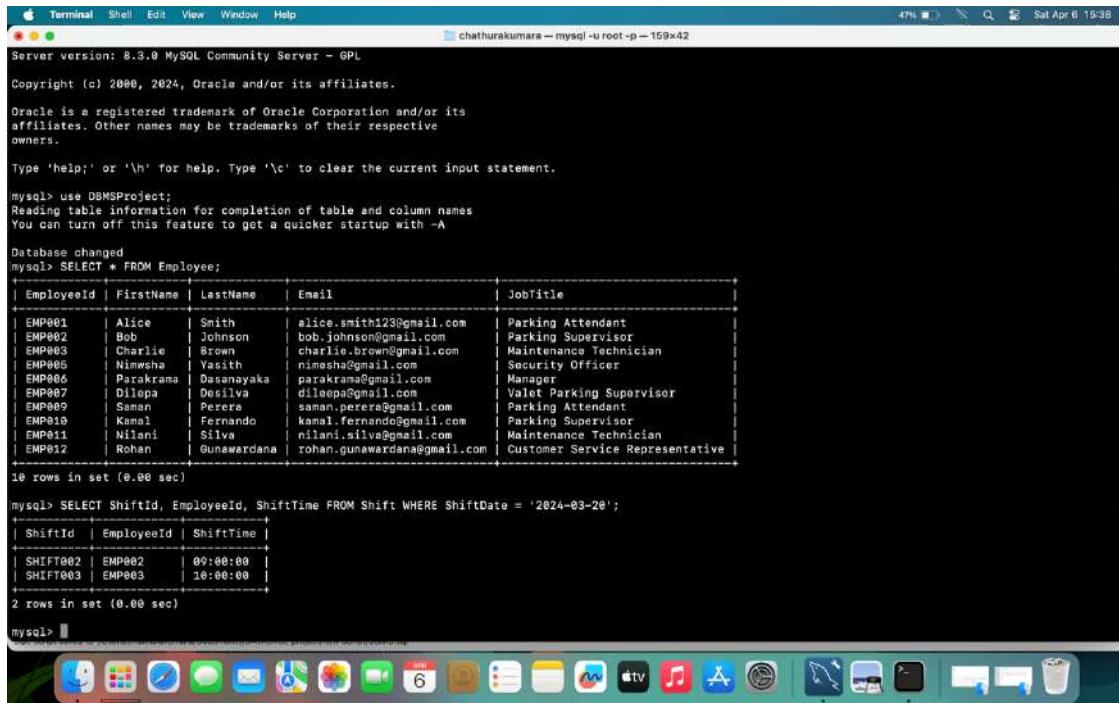
The screenshot shows the MySQL Workbench interface with the DBMS* tab selected. The left sidebar displays the schema tree under the DBMSProject schema. The main pane contains the following SQL script:

```
833 * WHERE DateId = 'DATE006';
834
835 WHERE DateId = 'DATE009';
836
837 * DELETE FROM SpotHistoryDate
838 WHERE DateId = 'DATE009';
839
840
841 --- 17 Supervise
842
843 * UPDATE Supervise
844 SET SupervisorId = NULL
845 WHERE SupervisorId = 'SP001';
846
847 * DELETE FROM Supervise
848 WHERE SupervisorId = 'SP004';
849
850 * UPDATE Supervise
851 SET SupervisorId = 'SUPER009'
852 WHERE SupervisorId = 'SP005';
853
854 * DELETE FROM Supervise
855 WHERE SupervisorId = 'SP008';
856
857 --- Fundamental Operations
858
859 --- 1) SELECT OPERATION
100% 26-63 | 4 errors found
```

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

Chapter 4: Transaction

❖ Select Operation



```
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.

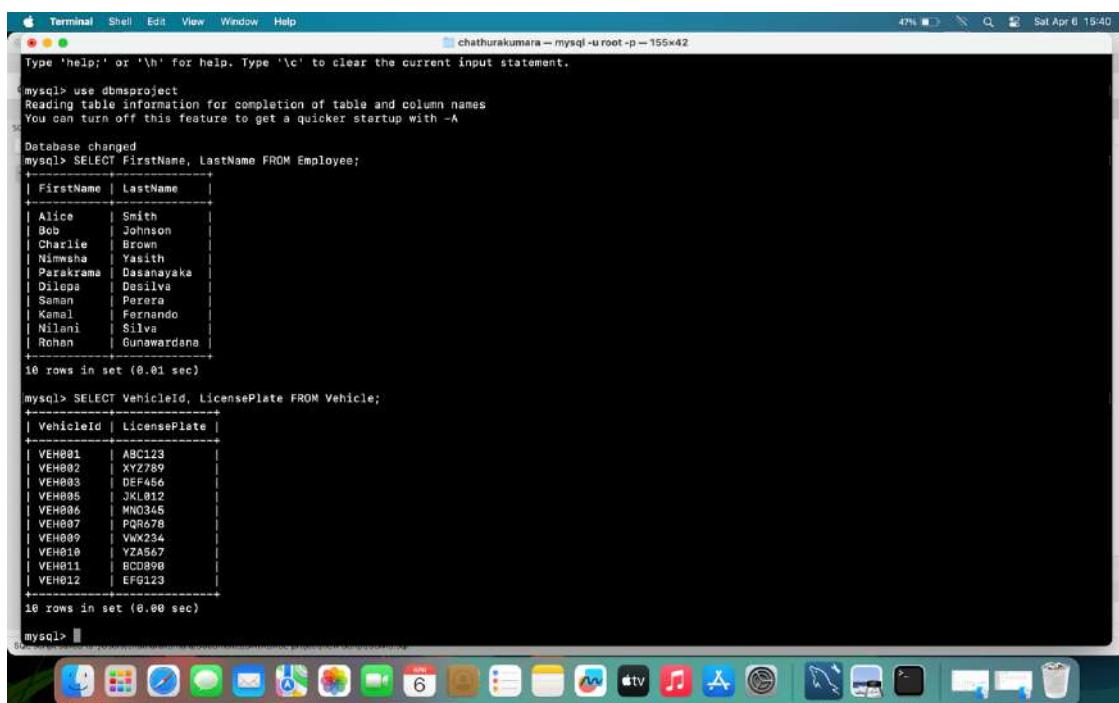
mysql> use DBMSProject;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

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 | Dilupa | Desilva | dilupa@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



```
Terminal Shell Edit View Window Help chathurakumara -- mysql -u root -p - 159x42
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

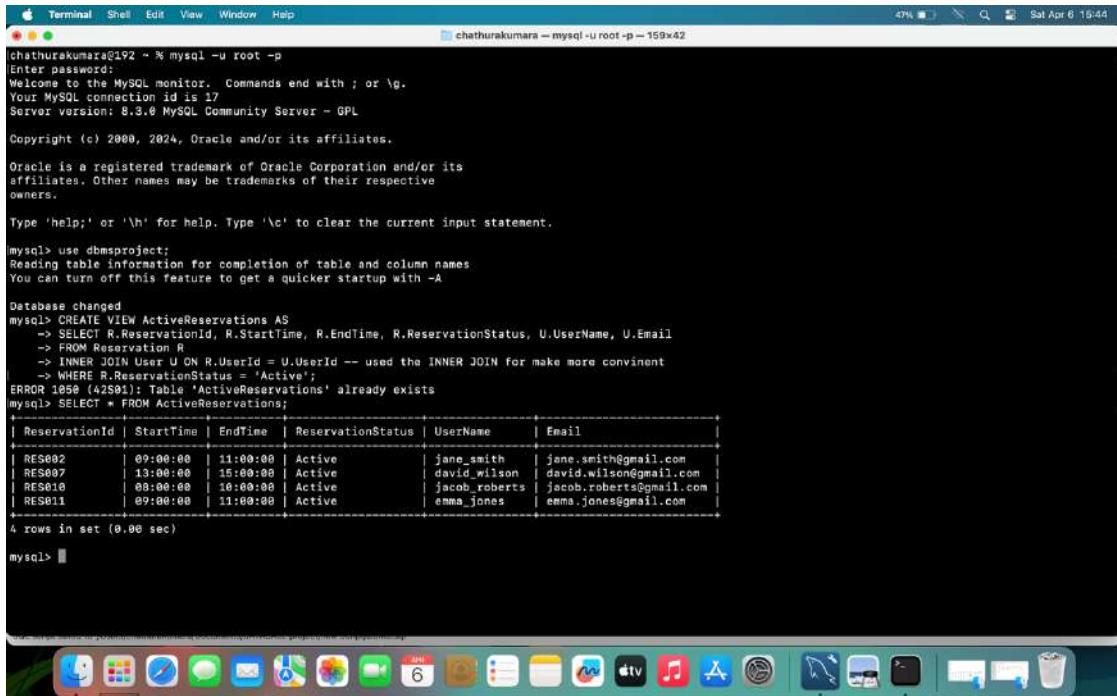
mysql> use dbmsproject;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT FirstName, LastName FROM Employee;
+-----+-----+
| FirstName | LastName |
+-----+-----+
| Alice | Smith |
| Bob | Johnson |
| Charlie | Brown |
| Nimwshe | Yasith |
| Parakkrama | Dasanayaka |
| Dilupa | 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 |
+-----+-----+
10 rows in set (0.00 sec)

mysql>
```


❖ 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.

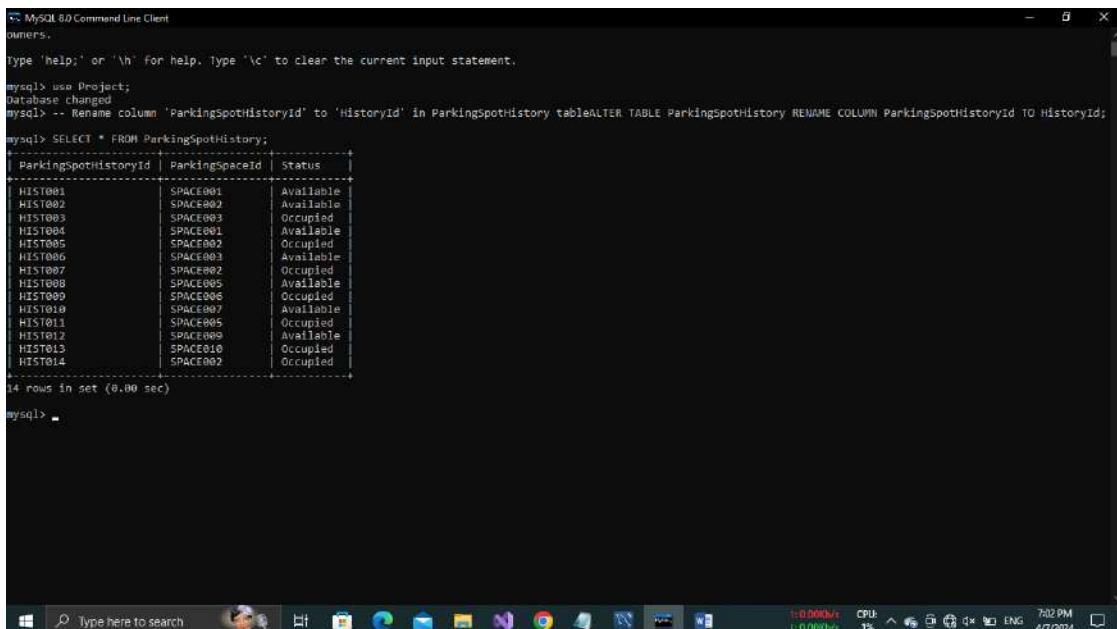
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

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 convenient
-> WHERE R.ReservationStatus = 'Active';
ERROR 1054 (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>

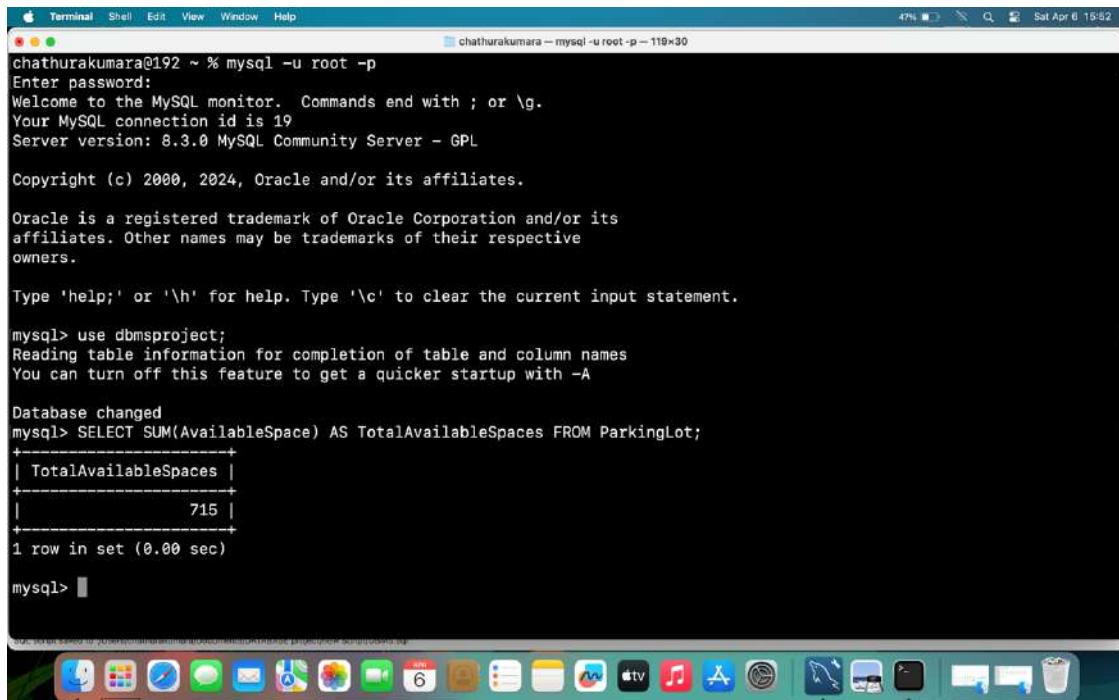
❖ Renaming Operation



MySQL 8.0 Command Line Client
D:\
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 | SPACE006 | 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)



```
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.

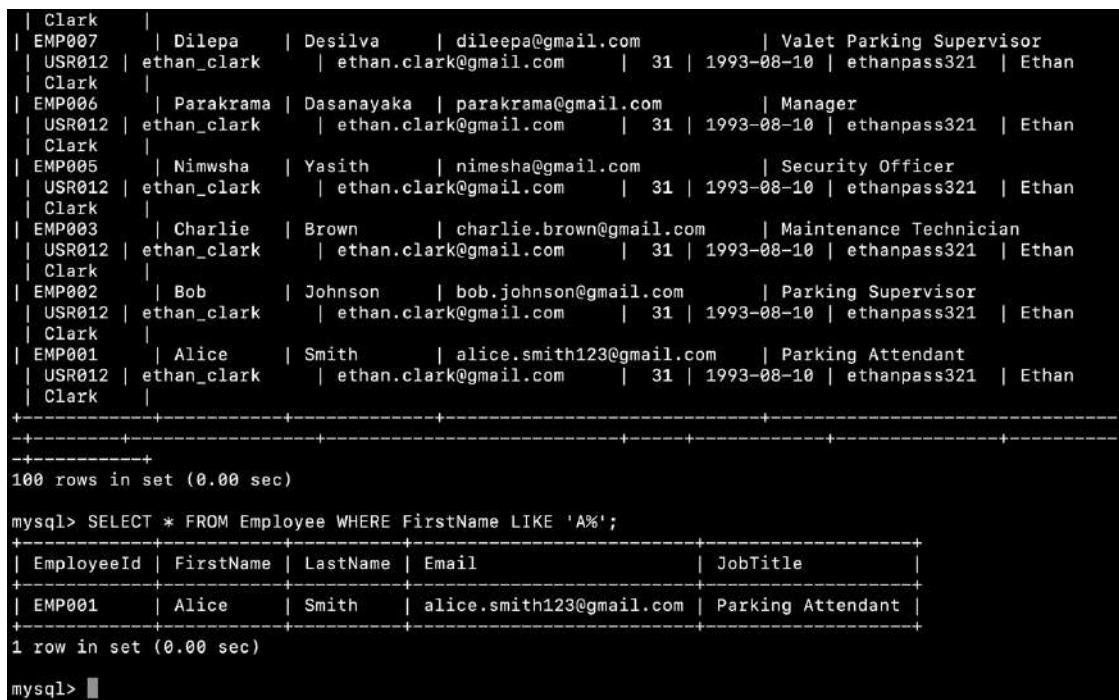
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

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

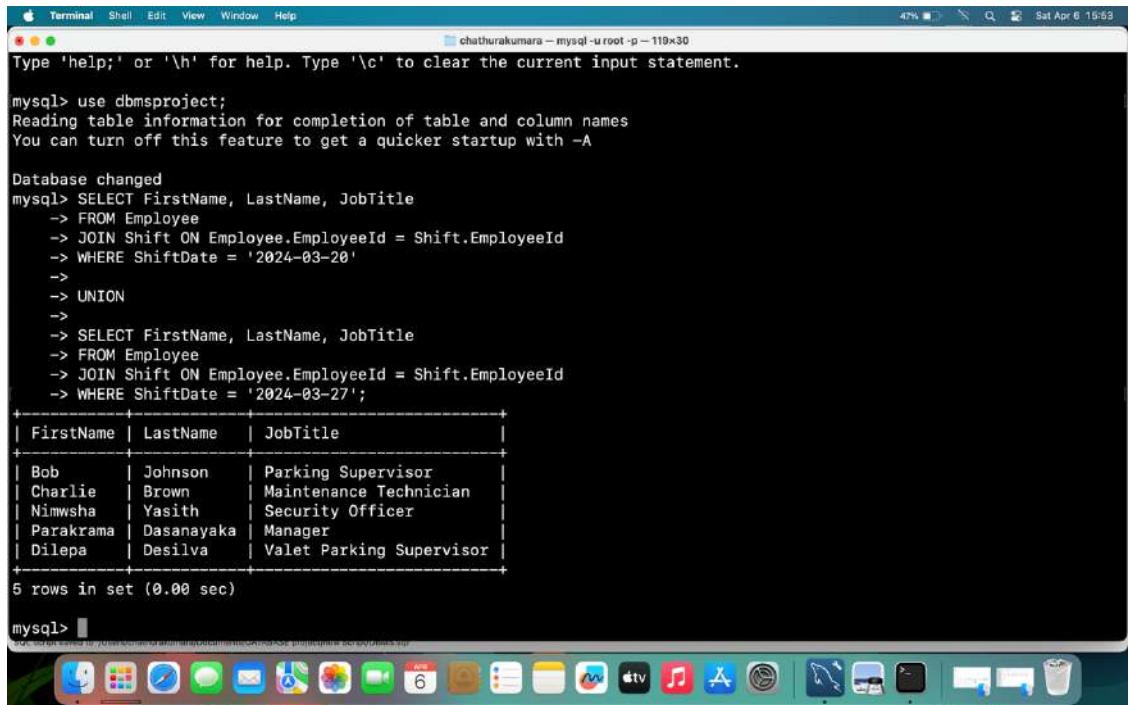
❖ Union

```
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 |
| Nimwsha   | Yasith   | Security Officer |
| Parakrama | Dasanayaka | Manager |
| Dilepa    | Desilva  | Valet Parking Supervisor |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```



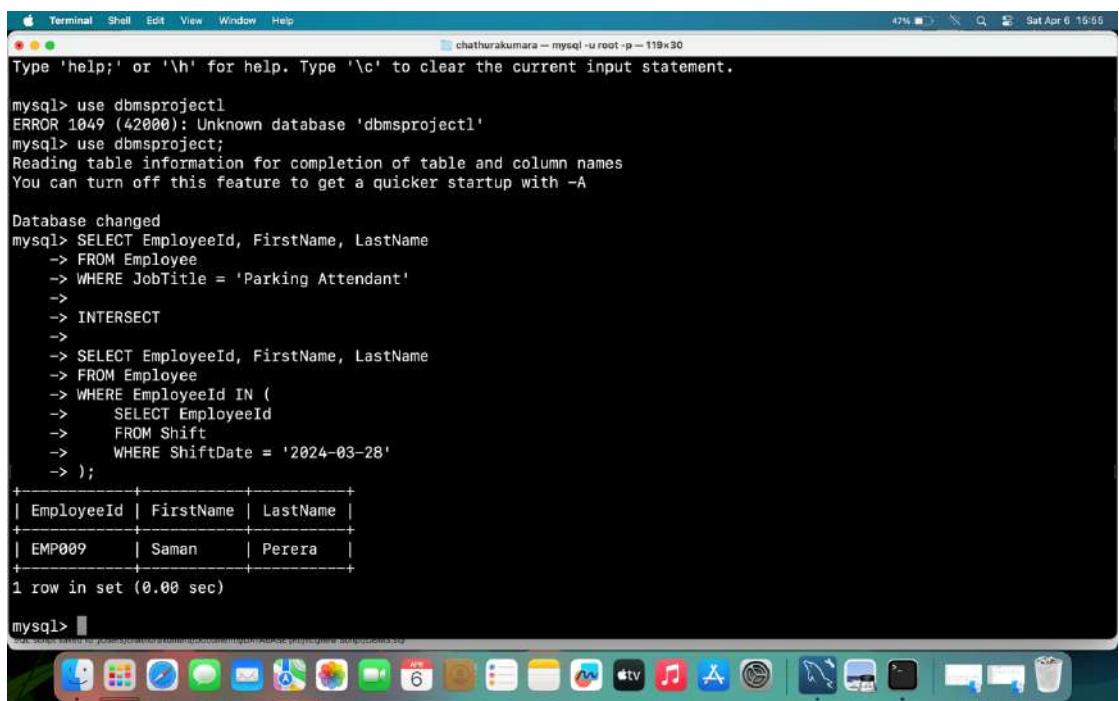
❖ Intersection

```
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.

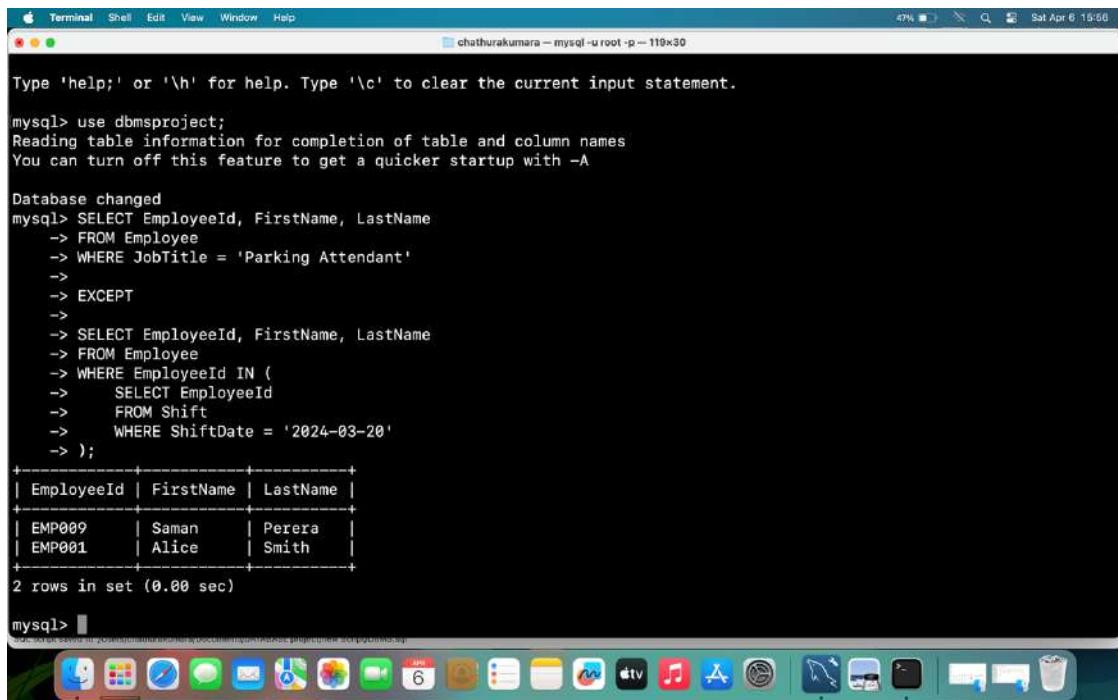
mysql> use dbmsproject1
ERROR 1049 (42000): Unknown database 'dbmsproject1'
mysql> use dbmsproject;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

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>
```



❖ Set Difference



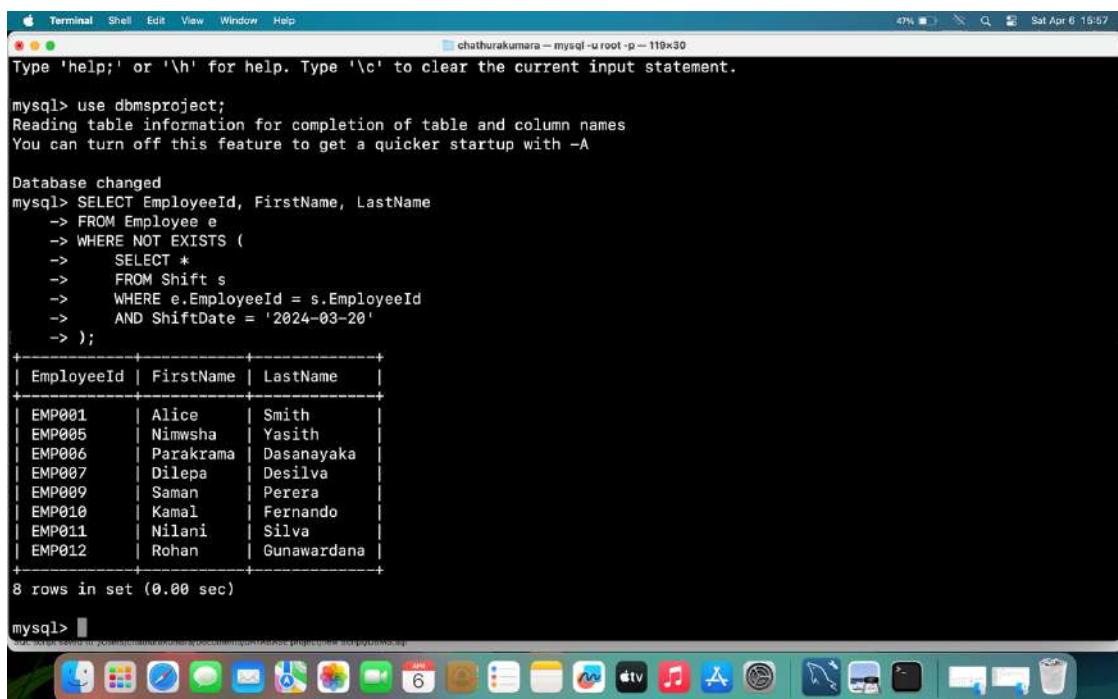
```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

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>
```

❖ 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>
```

❖ 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 | 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>
```

❖ Right outer join

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

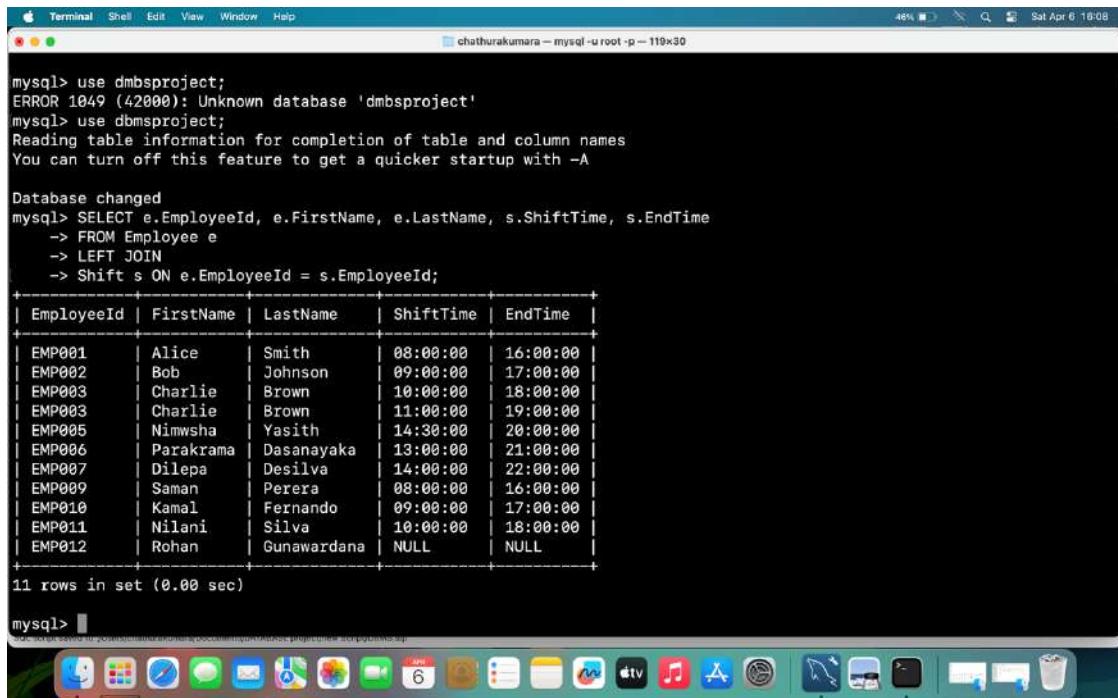
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

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 | 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 |
| EMP003 | Charlie | Brown | 11:00:00 | 19:00:00 |
+-----+-----+-----+-----+-----+
10 rows in set (0.01 sec)

mysql>
```

❖ Full Outer join



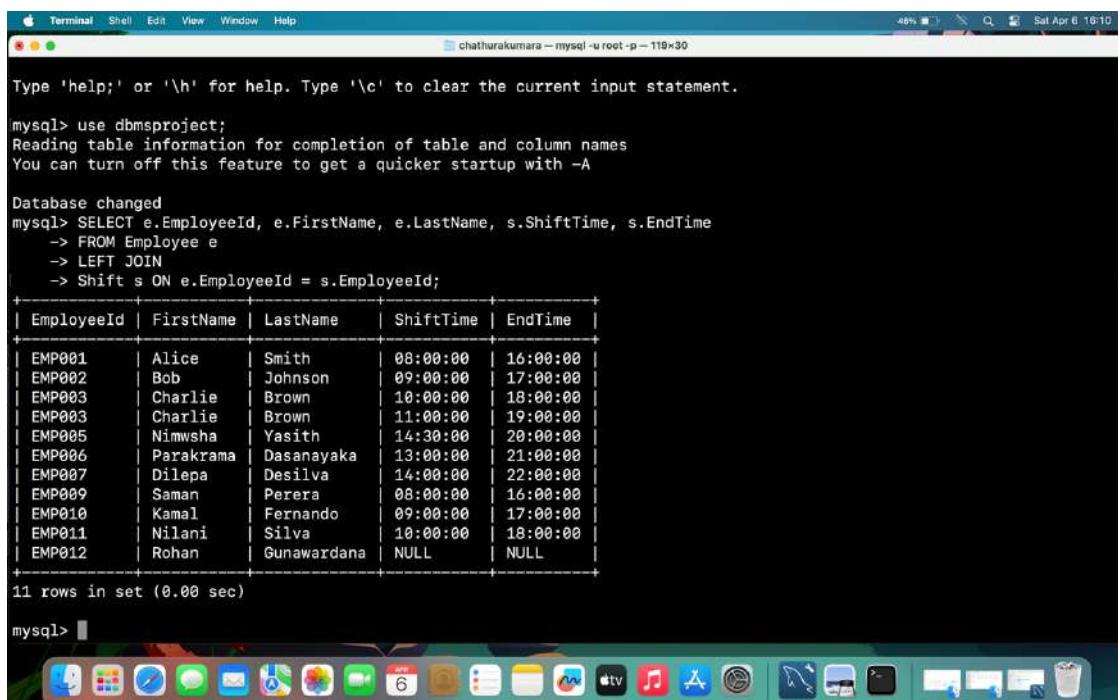
MySQL terminal window showing a full outer join query:

```
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:

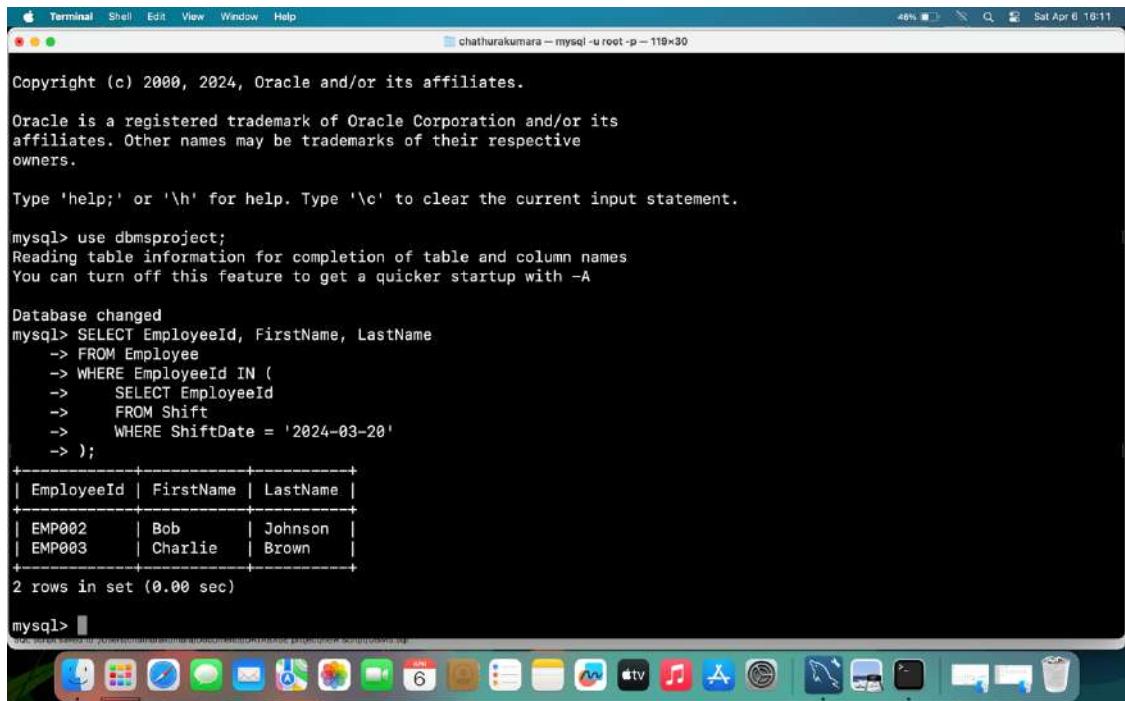
```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

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.

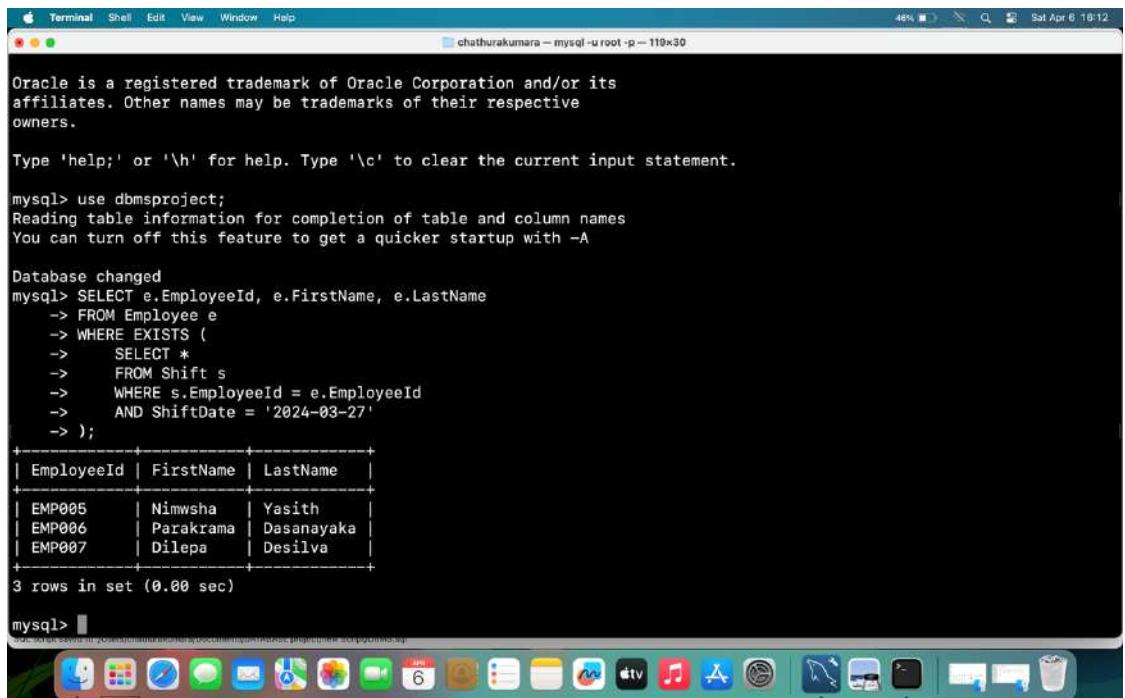
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

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 | 12.58 | Using where |
| 1 | PRIMARY | s | NULL | ref | FK_EmployeeId | FK_EmployeeId | 203 | dbmsproject.e.EmployeeId | 1 | 100.00 | NULL |
| 1 | PRIMARY | m | NULL | ref | ParkingLotId,fk_employeeeid | fk_employeeeid | 202 | 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 | 12.58 | Using where |
| 2 | UNION | s | NULL | ref | FX_EmployeeId | FX_EmployeeId | 203 | dbmsproject.e.EmployeeId | 1 | 100.00 | NULL |
| 2 | UNION | m | NULL | ref | ParkingLotId,fk_employeeeid | fk_employeeeid | 202 | 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@i9P2 ~ % mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 34
Server version: 8.3.8 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 | 18  | 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

```
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: 8.0.24 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 1
--> 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 | NULL | 10 | 100.00 | Using where |
| 2 | DEPENDENT SUBQUERY | p2 | NULL | eq_ref | PRIMARY | PRIMARY | NULL | dbmsproject.p1.ParkingLotId | 1 | 100.00 | NULL |
| 3 | UNION | p3 | NULL | ALL | NULL | NULL | NULL | NULL | 10 | 100.00 | Using where |
| 4 | DEPENDENT SUBQUERY | p4 | NULL | eq_ref | PRIMARY | PRIMARY | 02 | dbmsproject.p3.ParkingLotId | 1 | 100.00 | NULL |
+----+
4 rows in set, 3 warnings (0.00 sec)

mysql>
```

- Division tuning

```
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: 8.0.24 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.

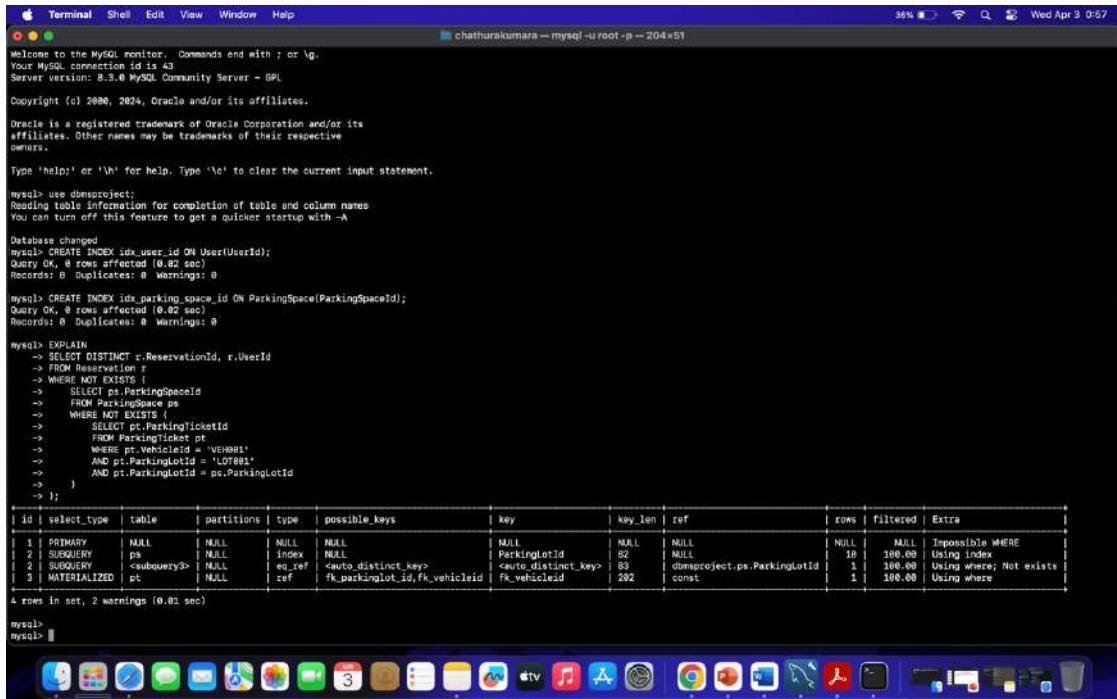
mysql> use dbmsproject;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

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 1
-->     FROM ParkingSpace ps
-->     WHERE NOT EXISTS (
-->         SELECT rt.ParkingTicketId
-->         FROM ParkingTicket pt
-->         WHERE pt.VehicleId = 'VEH001'
-->         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 | NULL | 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)

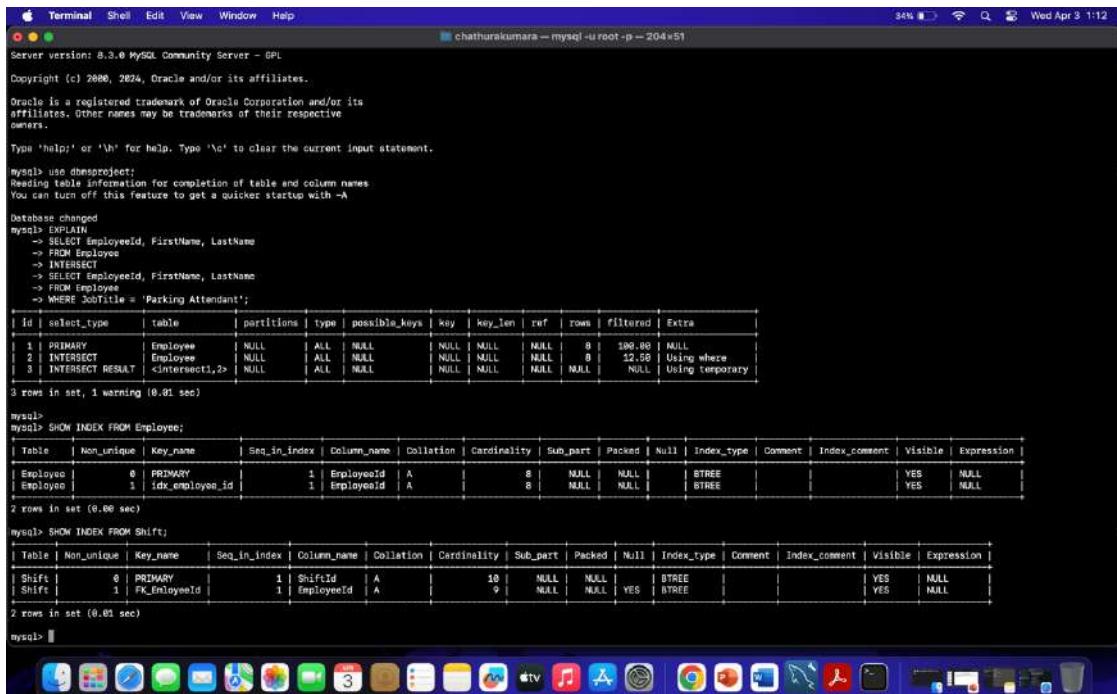
mysql>
```

- Correlated Nested query.



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 dmsproject;
Reading tables 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 c.ReservationId, r.UserId
--> FROM Reservation r
--> WHERE NOT EXISTS (
--> SELECT ps.ParkingSpaceId
--> FROM ParkingSpace ps
--> WHERE ps.Id =
--> 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	ps_id	ps_id	NULL	NULL	18	100.00	Using index
2	SUBQUERY	<subquery3>	NULL	eq_ref	<auto_distinct_key>	<auto_distinct_key>	89	dmsproject.ps.ParkingLotId	1	100.00	Using where; Not exists
3	MATERIALIZED	pt	NULL	ref	fk_parkinglot_id,fk_vehicleid	fk_vehicleid	292	const	1	100.00	Using where
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set, 2 warnings (0.02 sec)
mysql>
mysql>

- Intersection tuning



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 dmsproject;
Reading tables 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
--> INTERSECT
--> SELECT EmployeeId, FirstName, LastName
--> FROM Employee
--> WHERE JobTitle = 'Parking Attendant';
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1	PRIMARY	Employee	NULL	ALL	NULL	NULL	NULL	8	100.00	NULL	
2	INTERSECT	Employee	NULL	ALL	NULL	NULL	NULL	8	100.00	NULL	Using where
3	INTERSECT RESULT	<intersect1,2>	NULL	ALL	NULL	NULL	NULL	NULL	NULL	NULL	Using temporary
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+											
3 rows in set, 1 warning (0.01 sec)											
mysql>											
mysql> SHOW INDEX FROM Employee;											
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+											
Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+											
Employee	0	PRIMARY	1	EmployeeId	A	8	NULL	NULL	BTREE		YES
Employee	1	idx_employee_id	1	EmployeeId	A	8	NULL	NULL	BTREE		YES
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+											
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
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+											
Shift	0	PRIMARY	1	ShiftId	A	10	NULL	NULL	BTREE		YES
Shift	1	FK_EmployeeId	1	EmployeeId	A	9	NULL	NULL	BTREE		YES
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)
mysql>

- Set difference tuning.

```

Terminal Shell Edit View Window Help
chathurakumara - mysql -u root -p - 204x51
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 dbnspjroject;
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 Shift s ON e.EmployeeId = s.EmployeeId;
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref |
+----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY | Employee | NULL | ALL | NULL | NULL | NULL | |
| 2 | EXCEPT | e | NULL | index | FK_EmployeeId | FK_EmployeeId | 202 | NULL |
| 2 | EXCEPT | e | NULL | eq_ref | PRIMARY,`idx_employee_id` | PRIMARY | 202 | dbnspjroject.s.EmployeeId |
| 3 | EXCEPT RESULT | except3_2s | NULL | ALL | NULL | NULL | NULL |
+----+-----+-----+-----+-----+-----+-----+-----+
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 | 10 | 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 dbnspjroject;
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 | m | 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> 
```