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

List of figures

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 doesnot 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

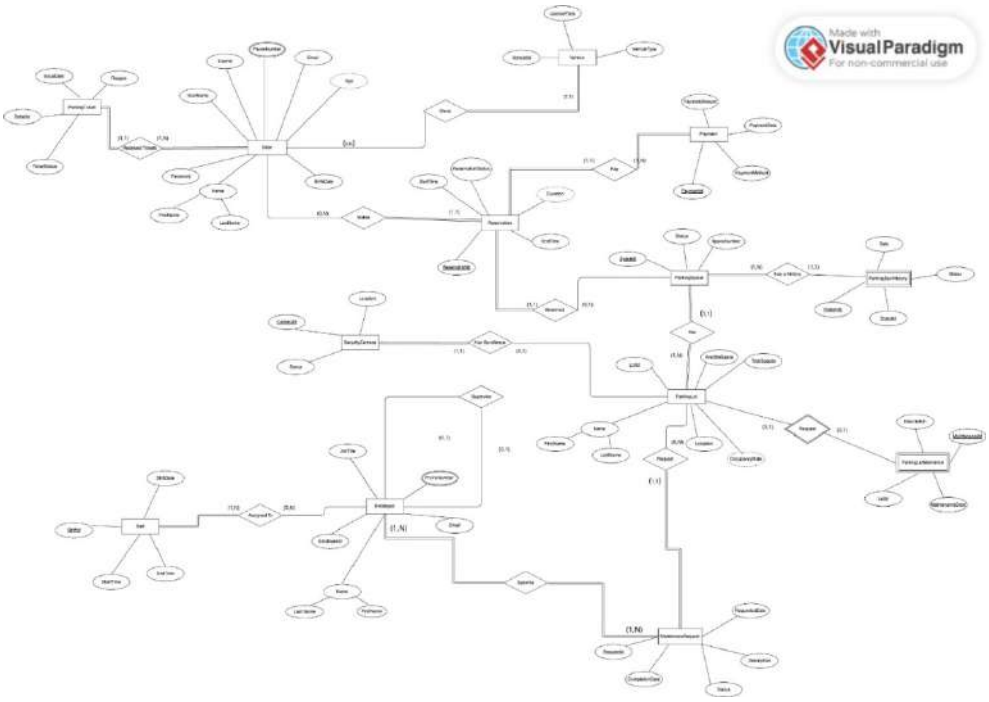


Figure 1: ER diagram of the Smart Parking Management System

2. UML Class Diagram of the Project

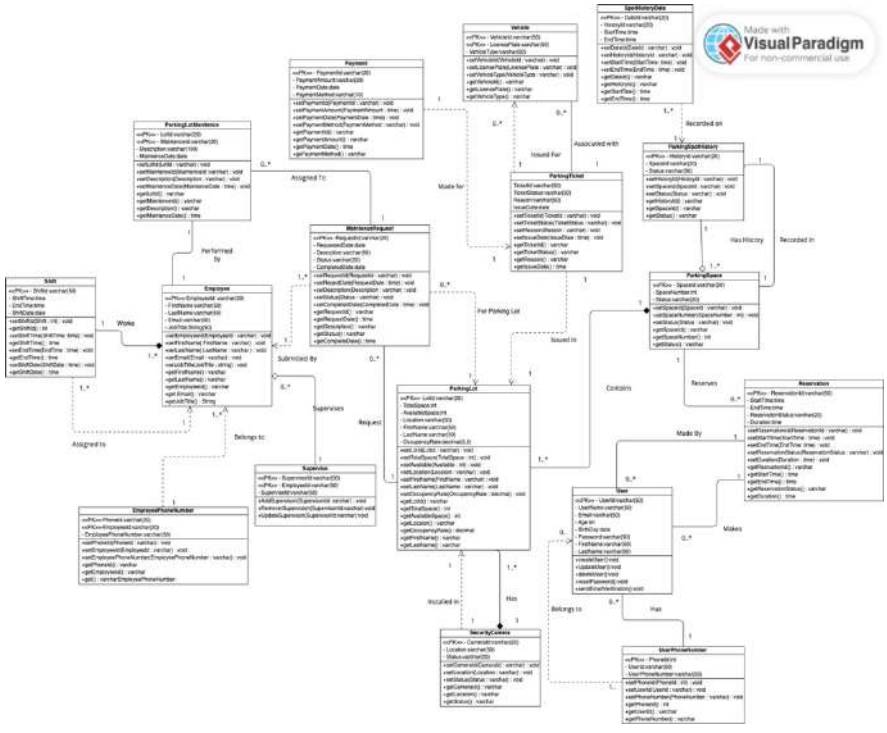


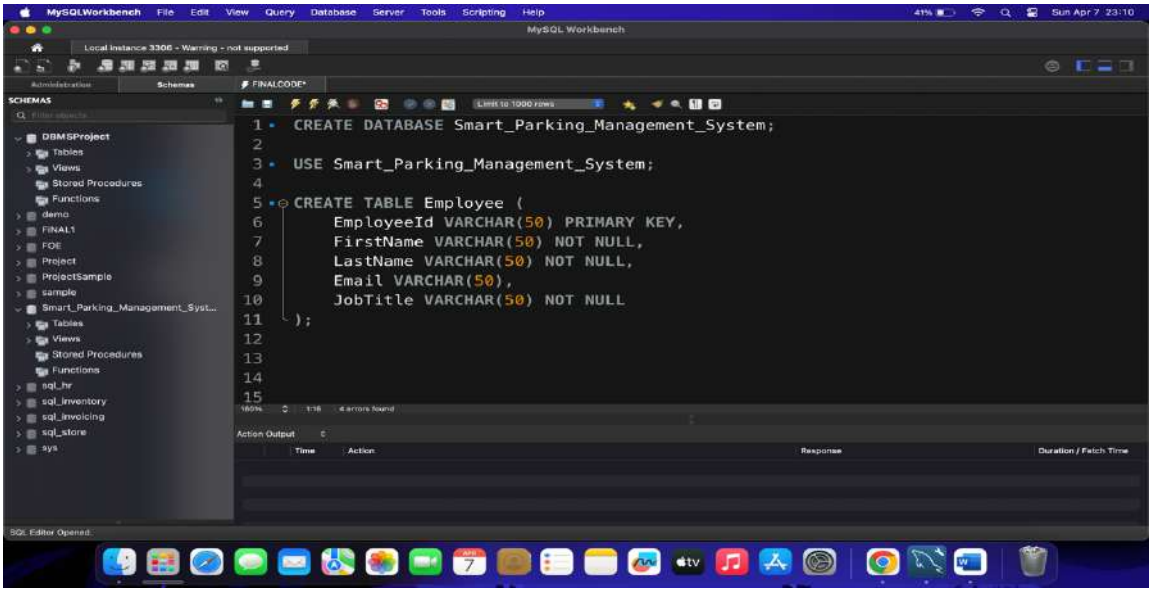
Figure 2: UML Class Diagram of the Smart Parking Management System

Chapter 3: Implementation

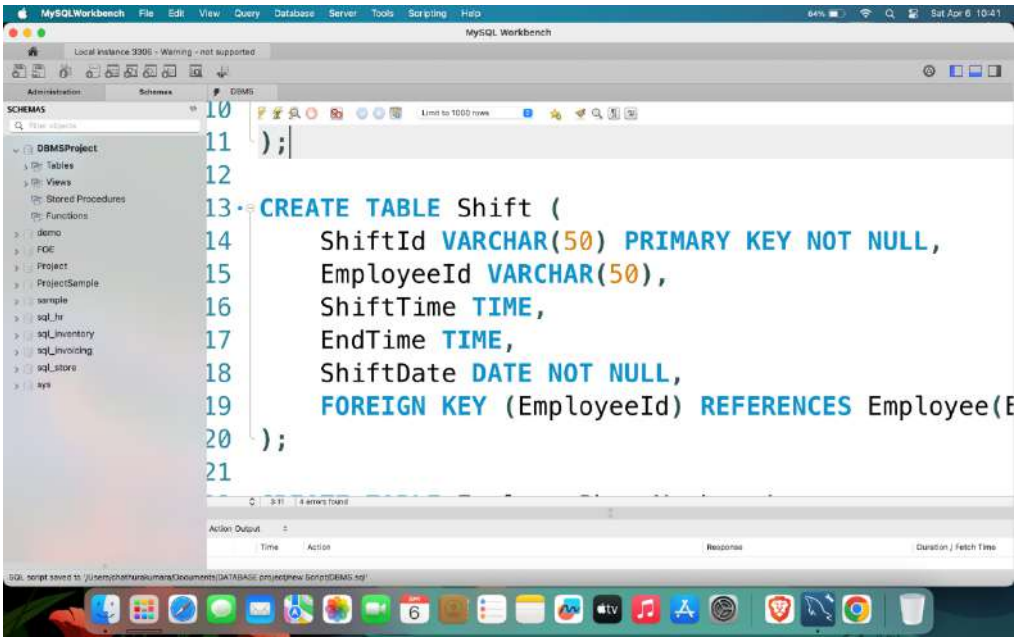
In this chapter, all the screenshots of implemented database model are included.

3.1 Table Implementation

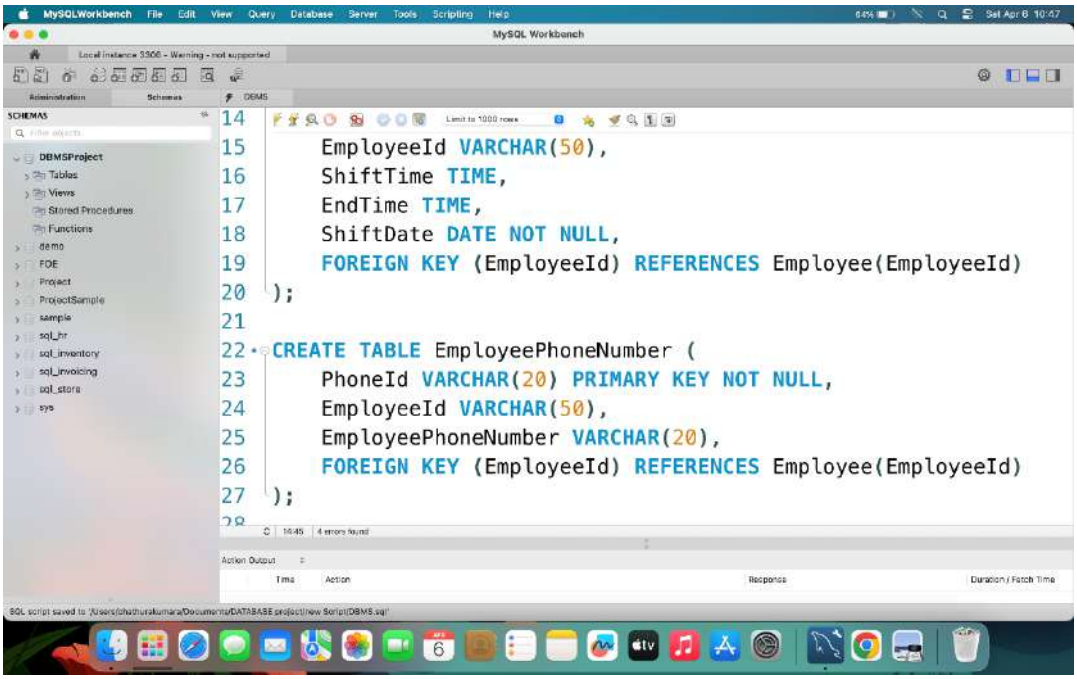
❖ Create table Employee.



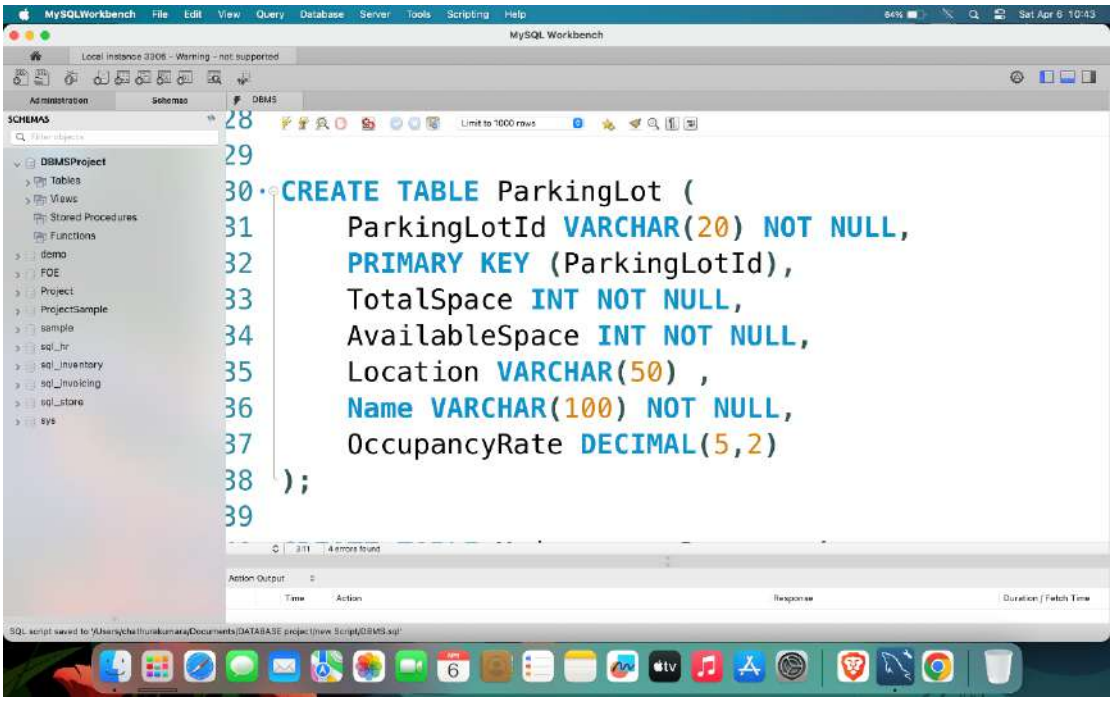
❖ Create table Shift.



❖ Create table Employee Phone Number

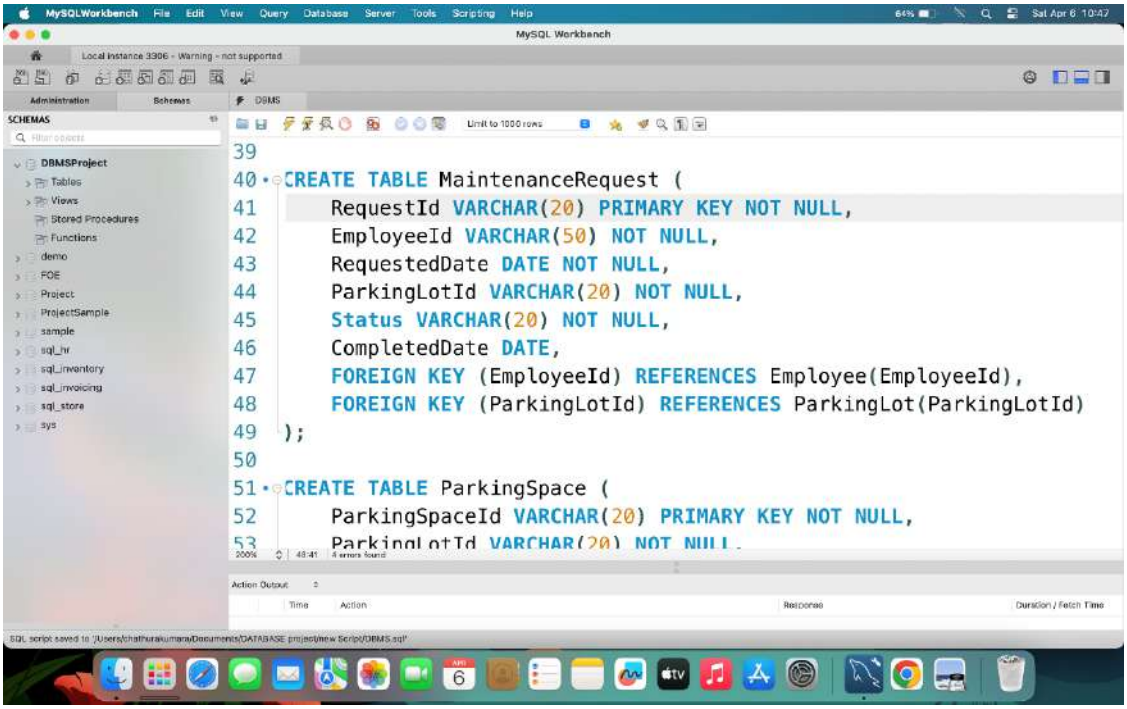


❖ Create table Parking Lot



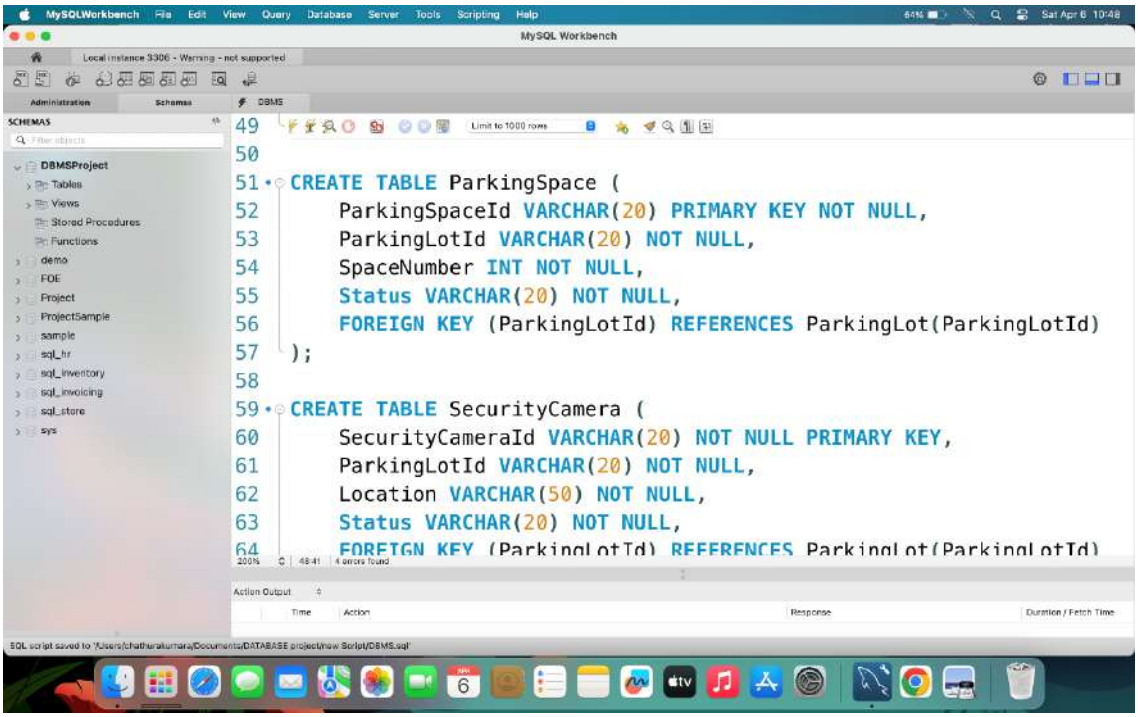
```
CREATE TABLE ParkingLot (
    ParkingLotId VARCHAR(20) NOT NULL,
    PRIMARY KEY (ParkingLotId),
    TotalSpace INT NOT NULL,
    AvailableSpace INT NOT NULL,
    Location VARCHAR(50) ,
    Name VARCHAR(100) NOT NULL,
    OccupancyRate DECIMAL(5,2)
);
```

❖ Create table Maintenance Request



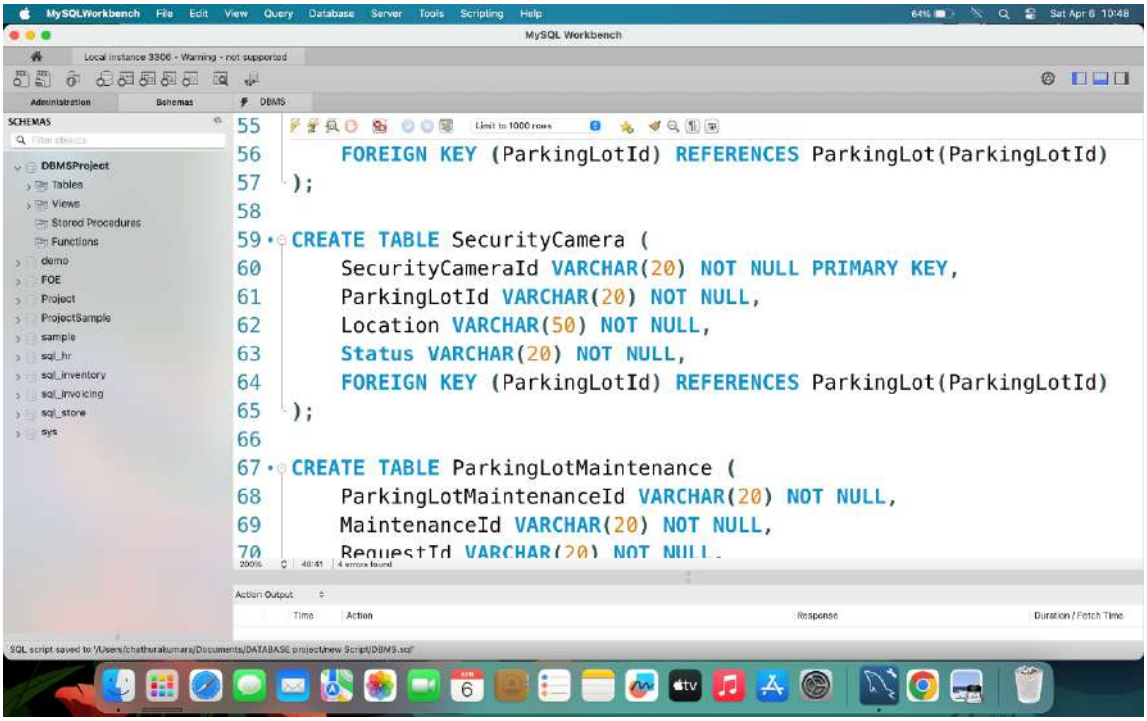
```
CREATE TABLE MaintenanceRequest (
    RequestId VARCHAR(20) PRIMARY KEY NOT NULL,
    EmployeeId VARCHAR(50) NOT NULL,
    RequestedDate DATE NOT NULL,
    ParkingLotId VARCHAR(20) NOT NULL,
    Status VARCHAR(20) NOT NULL,
    CompletedDate DATE,
    FOREIGN KEY (EmployeeId) REFERENCES Employee(EmployeeId),
    FOREIGN KEY (ParkingLotId) REFERENCES ParkingLot(ParkingLotId)
);

CREATE TABLE ParkingSpace (
    ParkingSpaceId VARCHAR(20) PRIMARY KEY NOT NULL,
    ParkingLotId VARCHAR(20) NOT NULL,
```

❖ Create table Parking Space



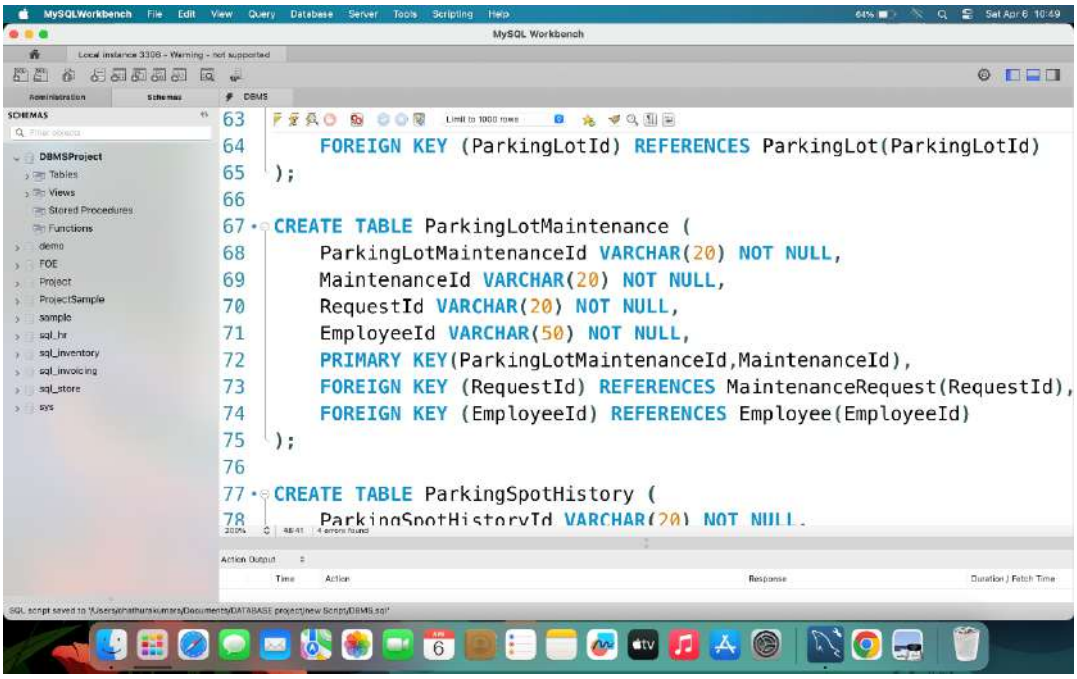
```
CREATE TABLE ParkingSpace (
    ParkingSpaceId VARCHAR(20) PRIMARY KEY NOT NULL,
    ParkingLotId VARCHAR(20) NOT NULL,
    SpaceNumber INT NOT NULL,
    Status VARCHAR(20) NOT NULL,
    FOREIGN KEY (ParkingLotId) REFERENCES ParkingLot(ParkingLotId)
);

CREATE TABLE SecurityCamera (
    SecurityCameraId VARCHAR(20) NOT NULL PRIMARY KEY,
    ParkingLotId VARCHAR(20) NOT NULL,
    Location VARCHAR(50) NOT NULL,
    Status VARCHAR(20) NOT NULL,
    FOREIGN KEY (ParkingLotId) REFERENCES ParkingLot(ParkingLotId)
```

❖ Create table Security Camera



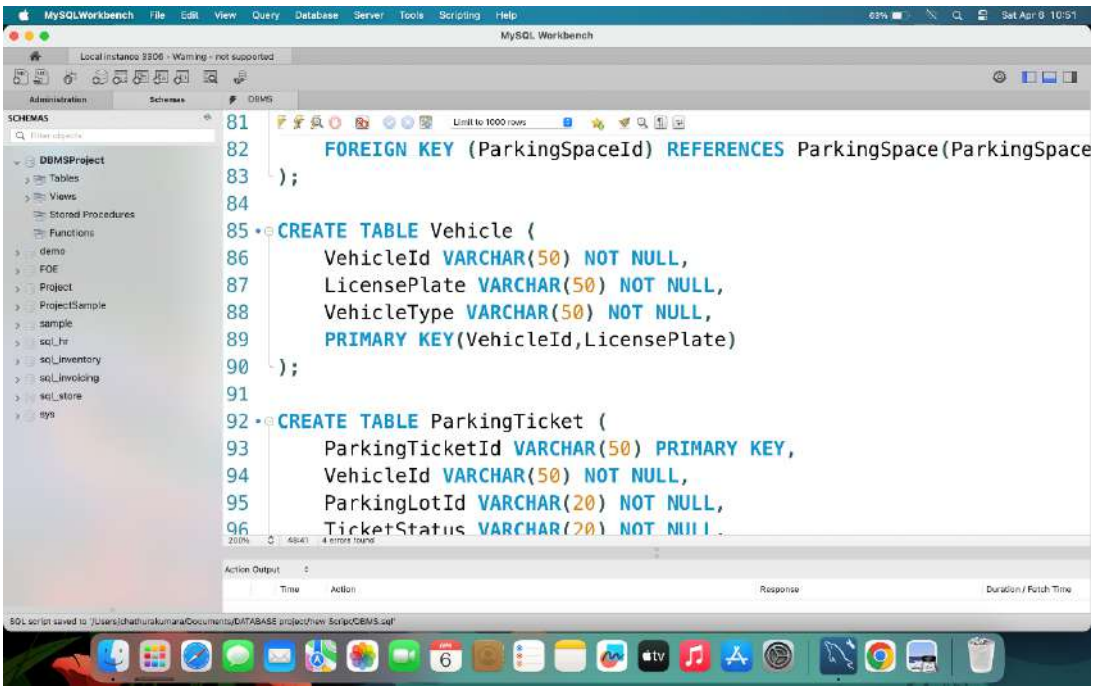
```
    FOREIGN KEY (ParkingLotId) REFERENCES ParkingLot(ParkingLotId)
);

CREATE TABLE SecurityCamera (
    SecurityCameraId VARCHAR(20) NOT NULL PRIMARY KEY,
    ParkingLotId VARCHAR(20) NOT NULL,
    Location VARCHAR(50) NOT NULL,
    Status VARCHAR(20) NOT NULL,
    FOREIGN KEY (ParkingLotId) REFERENCES ParkingLot(ParkingLotId)
);

CREATE TABLE ParkingLotMaintenance (
    ParkingLotMaintenanceId VARCHAR(20) NOT NULL,
    MaintenanceId VARCHAR(20) NOT NULL,
    RequestId VARCHAR(20) NOT NULL,
```

❖ Create table Parking Lot Maintenance



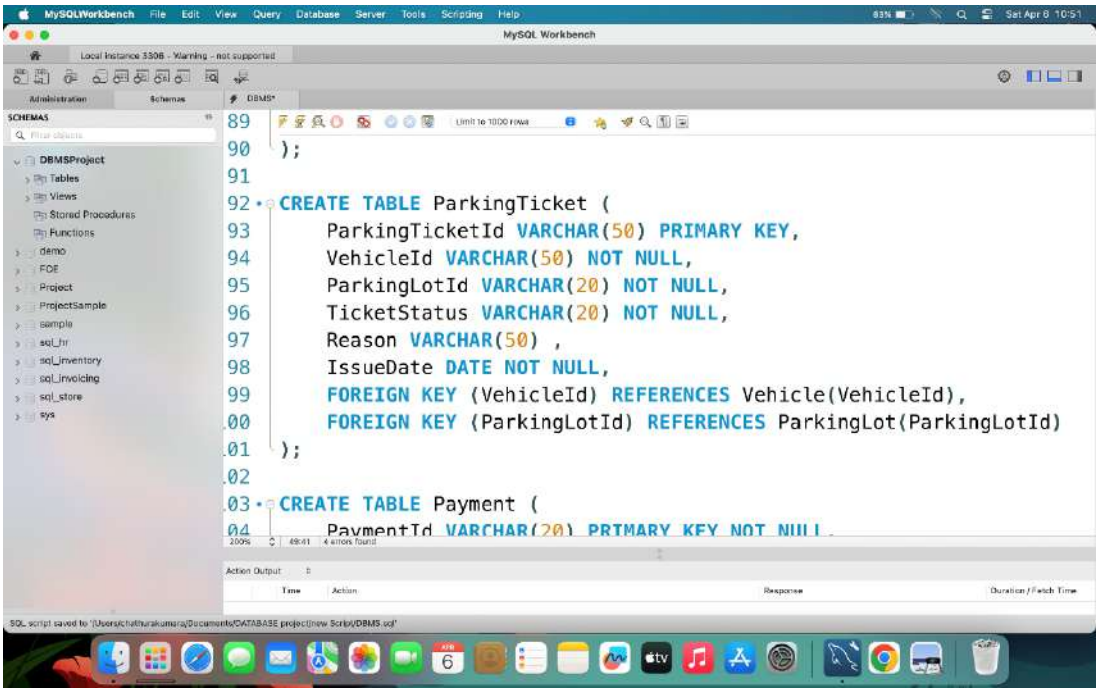
```
63
64      FOREIGN KEY (ParkingLotId) REFERENCES ParkingLot(ParkingLotId)
65  );
66
67  CREATE TABLE ParkingLotMaintenance (
68      ParkingLotMaintenanceId VARCHAR(20) NOT NULL,
69      MaintenanceId VARCHAR(20) NOT NULL,
70      RequestId VARCHAR(20) NOT NULL,
71      EmployeeId VARCHAR(50) NOT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

❖ Create Vehicle



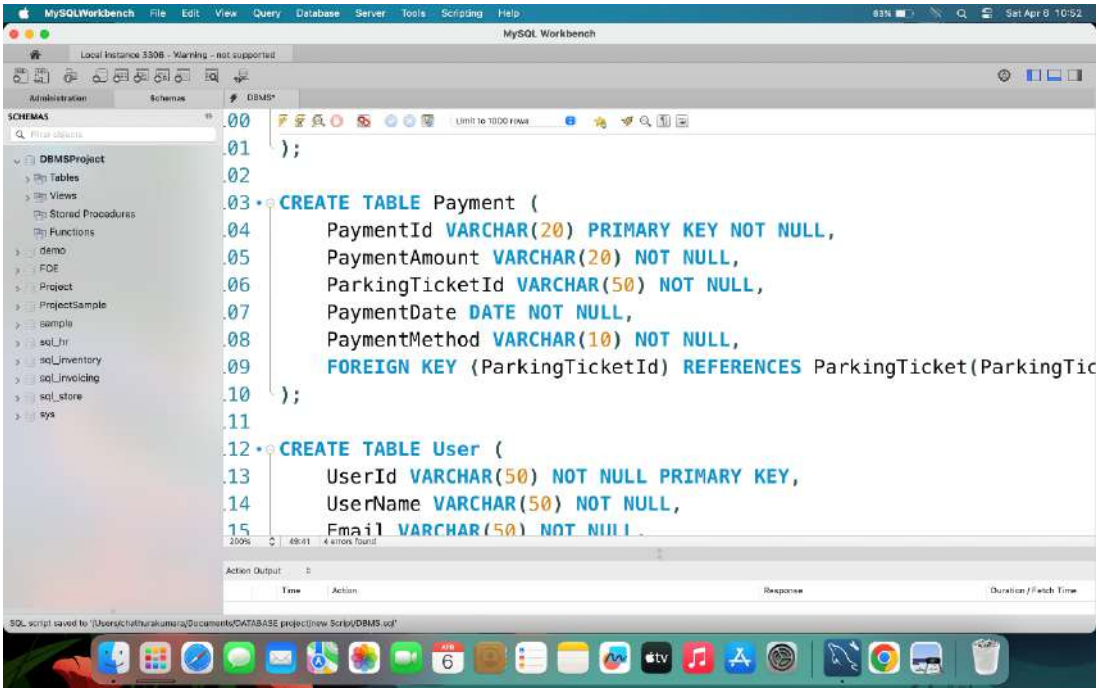
```
81
82      FOREIGN KEY (ParkingSpaceId) REFERENCES ParkingSpace(ParkingSpace
83  );
84
85  CREATE TABLE Vehicle (
86      VehicleId VARCHAR(50) NOT NULL,
87      LicensePlate VARCHAR(50) NOT NULL,
88      VehicleType VARCHAR(50) NOT NULL,
89      PRIMARY KEY(VehicleId,LicensePlate)
90  );
91
92  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,
```

❖ Create table ParkingTicket.



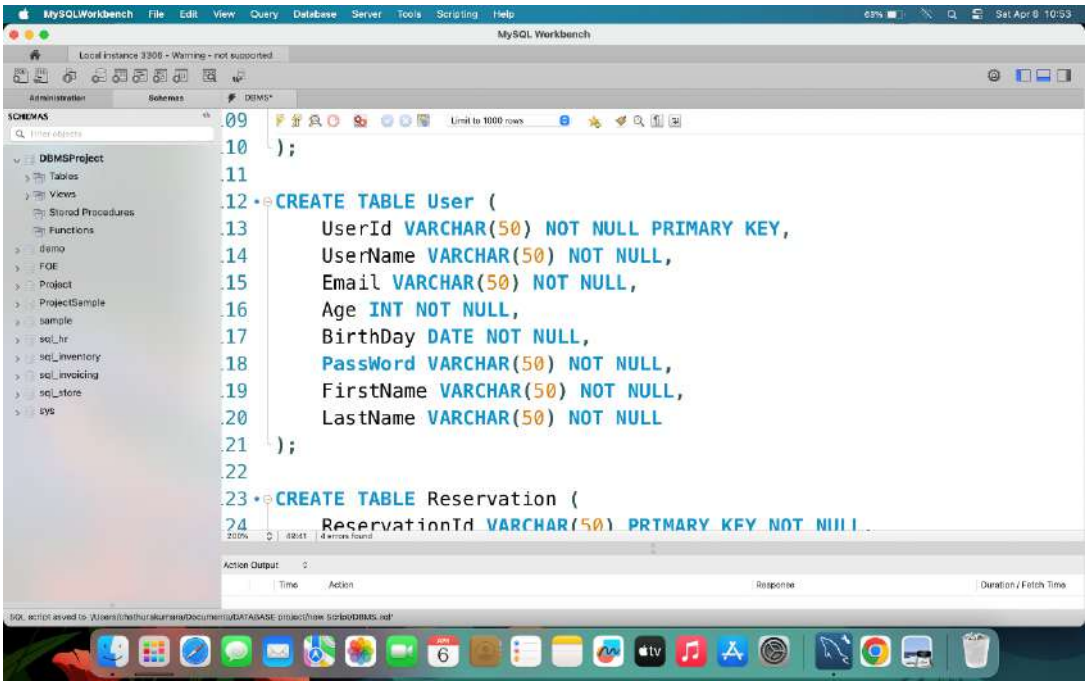
```
);

CREATE TABLE ParkingTicket (
    ParkingTicketId VARCHAR(50) PRIMARY KEY,
    VehicleId VARCHAR(50) NOT NULL,
    ParkingLotId VARCHAR(20) NOT NULL,
    TicketStatus VARCHAR(20) NOT NULL,
    Reason VARCHAR(50) ,
    IssueDate DATE NOT NULL,
    FOREIGN KEY (VehicleId) REFERENCES Vehicle(VehicleId),
    FOREIGN KEY (ParkingLotId) REFERENCES ParkingLot(ParkingLotId)
);

CREATE TABLE Payment (
    PaymentId VARCHAR(20) PRIMARY KEY NOT NULL,
```

❖ Create table Payment.



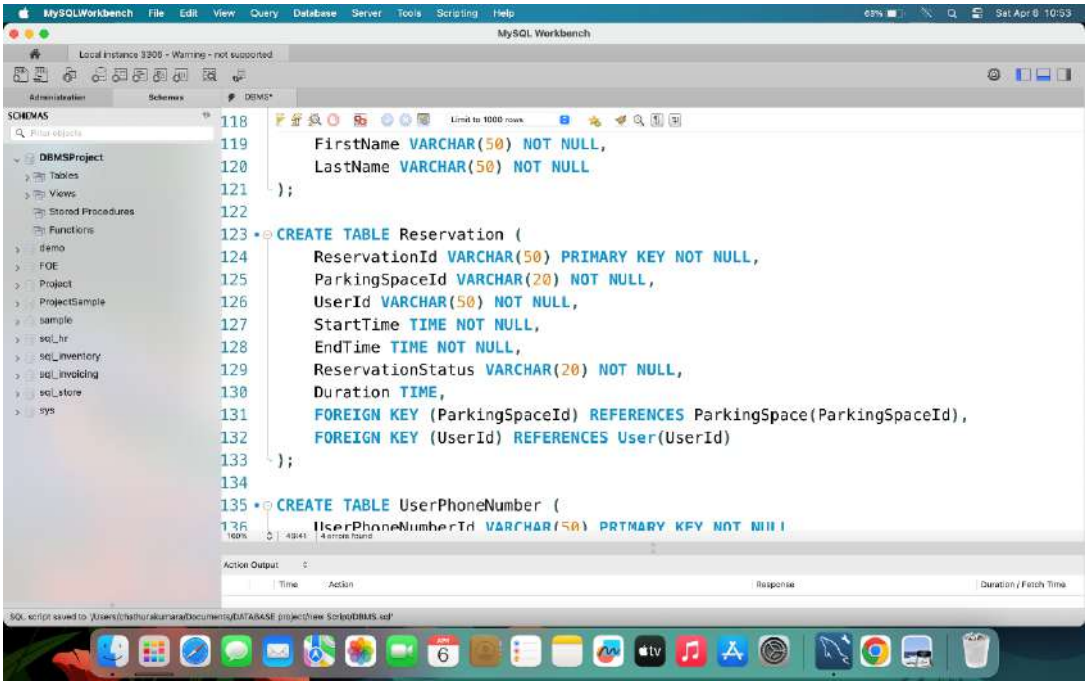
```
);

CREATE TABLE Payment (
    PaymentId VARCHAR(20) PRIMARY KEY NOT NULL,
    PaymentAmount VARCHAR(20) NOT NULL,
    ParkingTicketId VARCHAR(50) NOT NULL,
    PaymentDate DATE NOT NULL,
    PaymentMethod VARCHAR(10) NOT NULL,
    FOREIGN KEY (ParkingTicketId) REFERENCES ParkingTicket(ParkingTic
);

CREATE TABLE User (
    UserId VARCHAR(50) NOT NULL PRIMARY KEY,
    UserName VARCHAR(50) NOT NULL,
    Email VARCHAR(50) NOT NULL,
```

❖ Create table User.



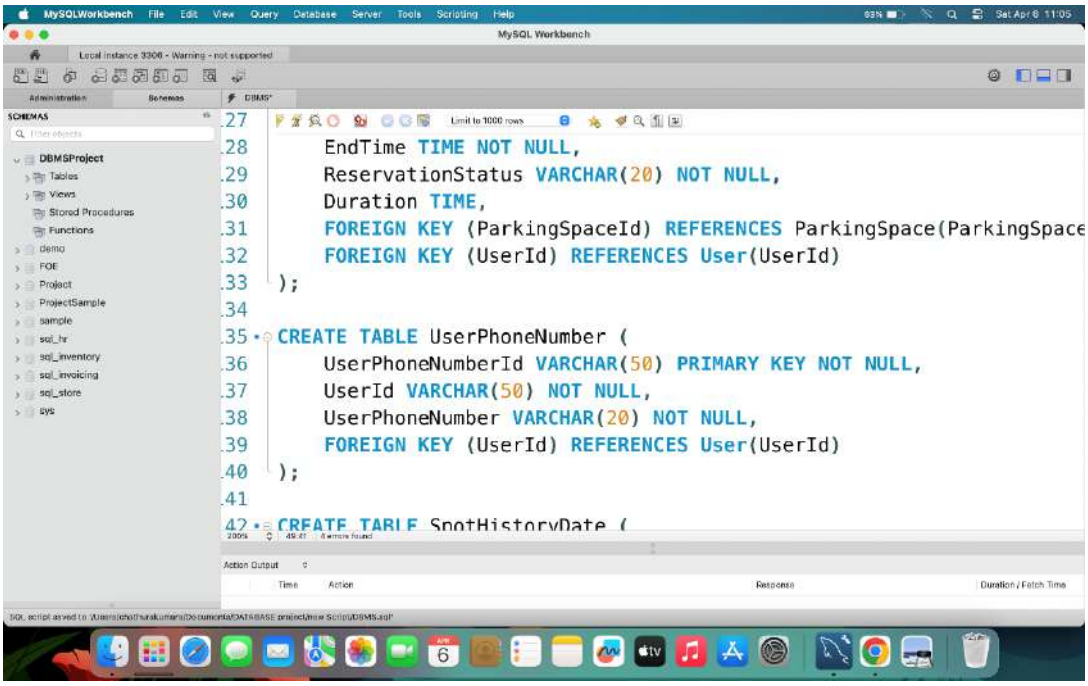
```
);

CREATE TABLE User (
    UserId VARCHAR(50) NOT NULL PRIMARY KEY,
    UserName VARCHAR(50) NOT NULL,
    Email VARCHAR(50) NOT NULL,
    Age INT NOT NULL,
    BirthDay DATE NOT NULL,
    PassWord VARCHAR(50) NOT NULL,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL
);

CREATE TABLE Reservation (
    ReservationId VARCHAR(50) PRIMARY KEY NOT NULL,
```

❖ Create table Reservation.



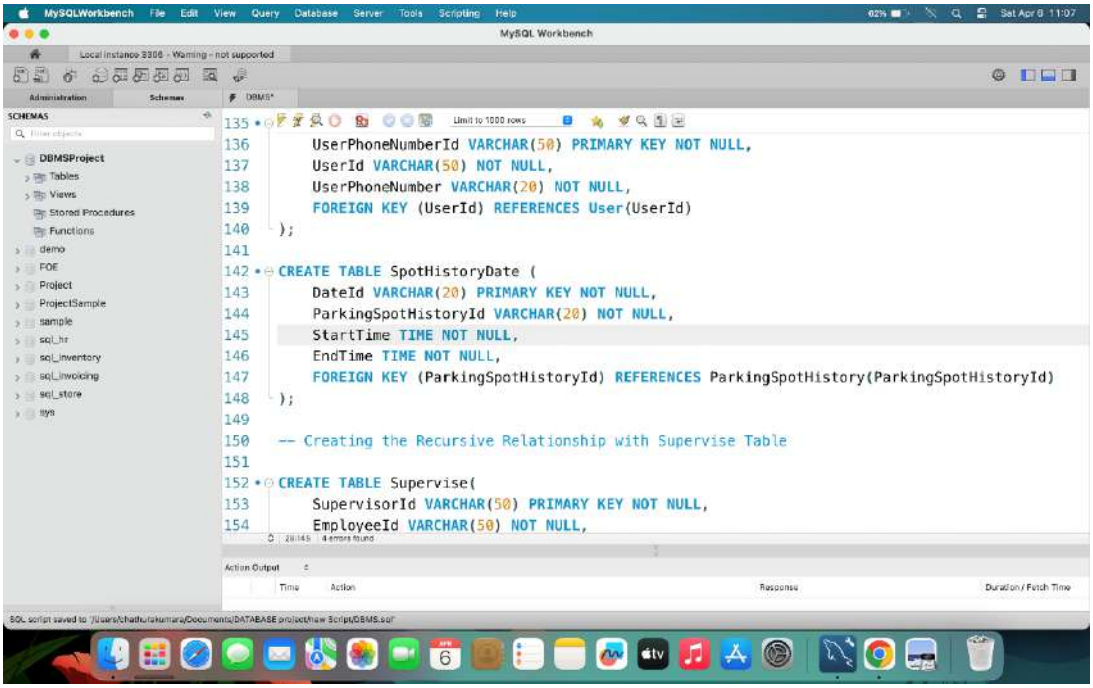
```
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL
);

CREATE TABLE Reservation (
    ReservationId VARCHAR(50) PRIMARY KEY NOT NULL,
    ParkingSpaceId VARCHAR(20) NOT NULL,
    UserId VARCHAR(50) NOT NULL,
    StartTime TIME NOT NULL,
    EndTime TIME NOT NULL,
    ReservationStatus VARCHAR(20) NOT NULL,
    Duration TIME,
    FOREIGN KEY (ParkingSpaceId) REFERENCES ParkingSpace(ParkingSpaceId),
    FOREIGN KEY (UserId) REFERENCES User(UserId)
);

CREATE TABLE UserPhoneNumber (
```

❖ Create table UserPhoneNumber.



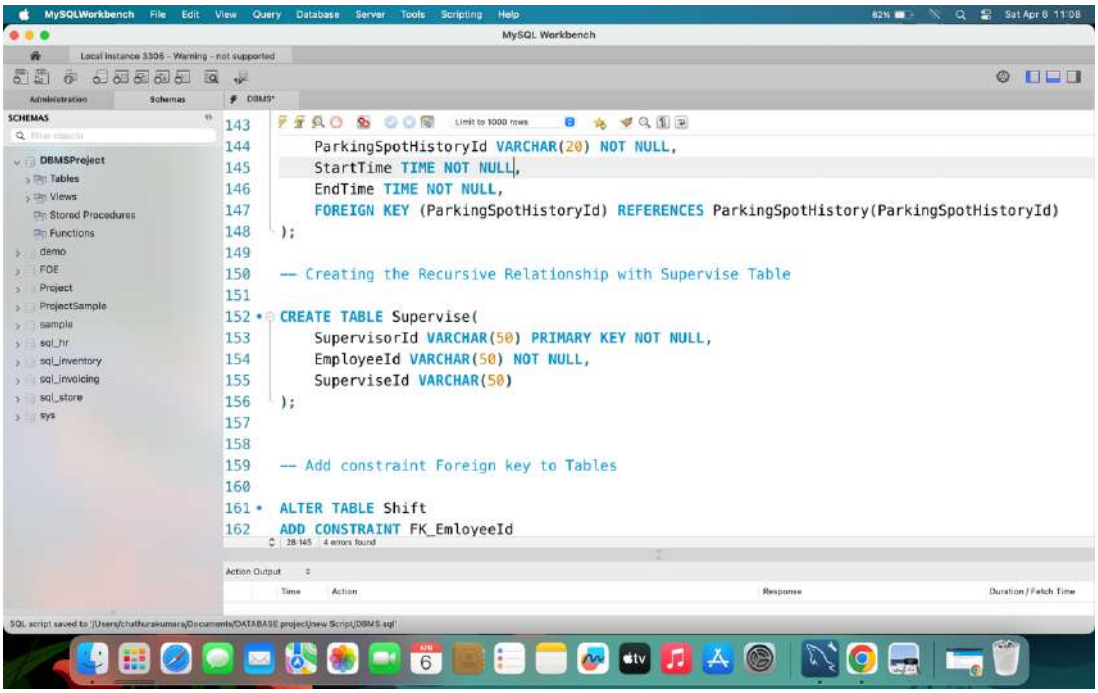
```
    EndTime TIME NOT NULL,
    ReservationStatus VARCHAR(20) NOT NULL,
    Duration TIME,
    FOREIGN KEY (ParkingSpaceId) REFERENCES ParkingSpace(ParkingSpace
    FOREIGN KEY (UserId) REFERENCES User(UserId)
);

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

❖ Create table SpotHistoryDate.



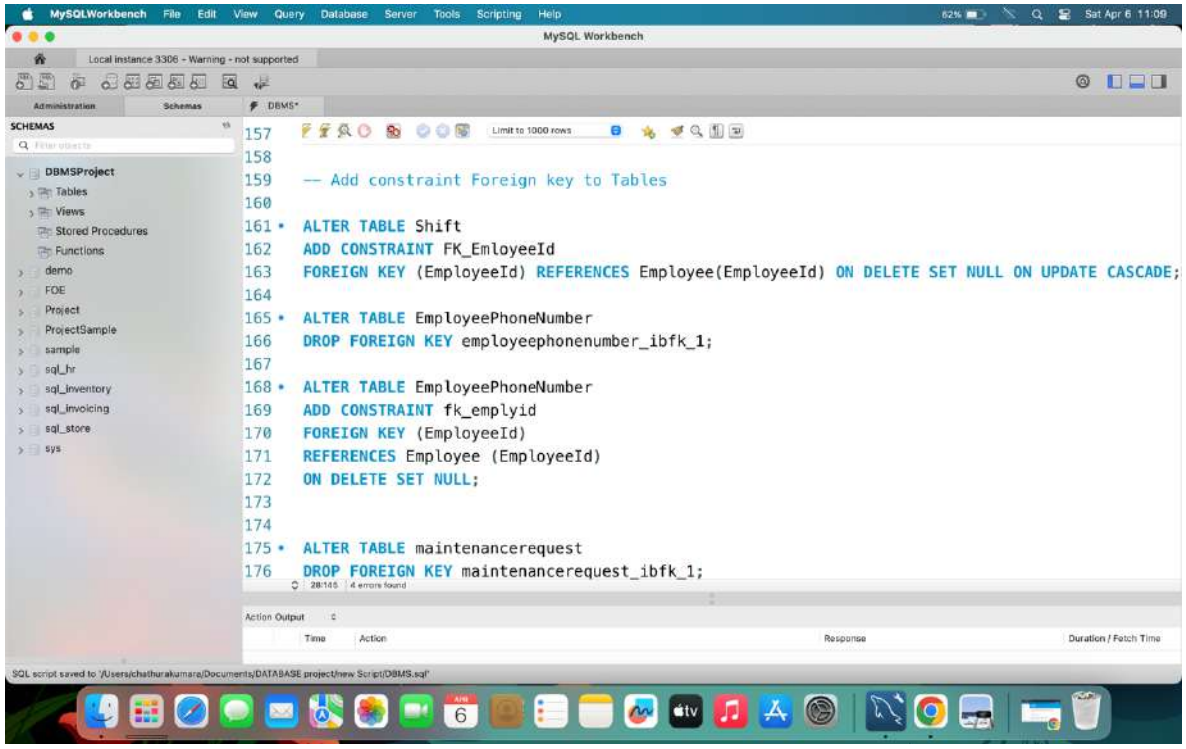
```
    UserPhoneNumberId VARCHAR(50) PRIMARY KEY NOT NULL,
    UserId VARCHAR(50) NOT NULL,
    UserPhoneNumber VARCHAR(20) NOT NULL,
    FOREIGN KEY (UserId) REFERENCES User(UserId)
);

CREATE TABLE SpotHistoryDate (
    DateId VARCHAR(20) PRIMARY KEY NOT NULL,
    ParkingSpotHistoryId VARCHAR(20) NOT NULL,
    StartTime TIME NOT NULL,
    EndTime TIME NOT NULL,
    FOREIGN KEY (ParkingSpotHistoryId) REFERENCES ParkingSpotHistory(ParkingSpotHistoryId)
);

-- Creating the Recursive Relationship with Supervise Table

CREATE TABLE Supervise(
    SupervisorId VARCHAR(50) PRIMARY KEY NOT NULL,
    EmployeeId VARCHAR(50) NOT NULL,
```

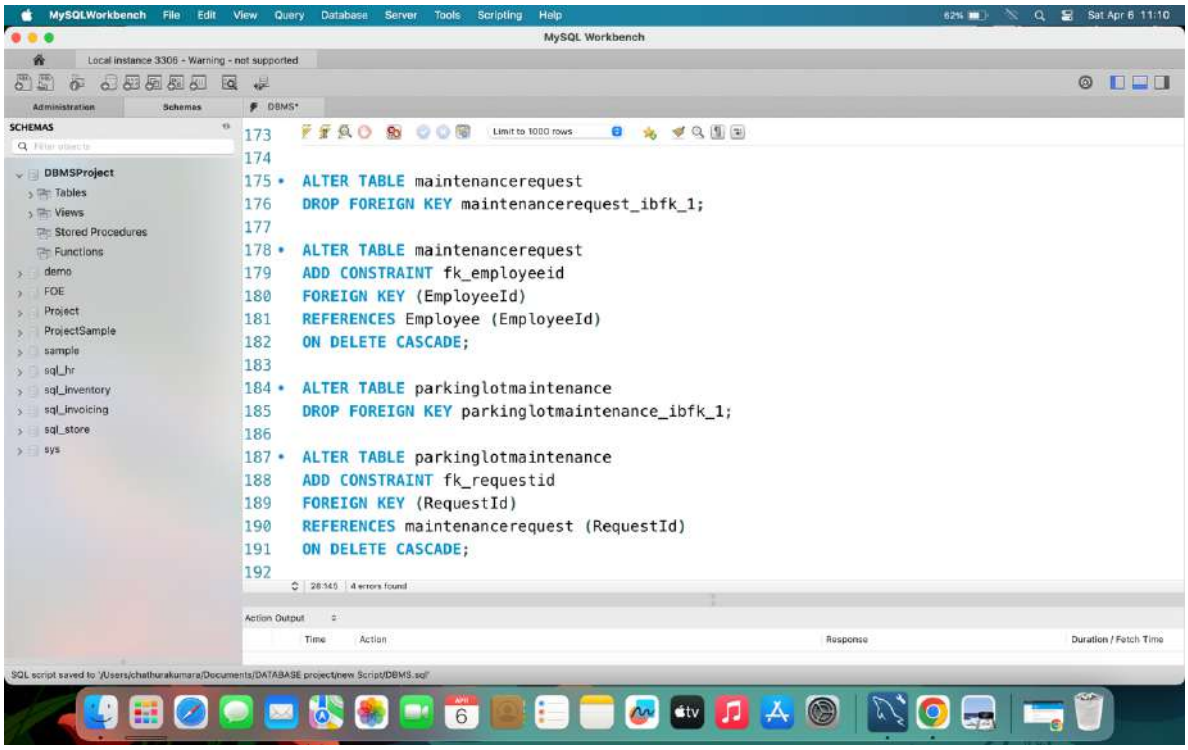
❖ Create table Supervise.

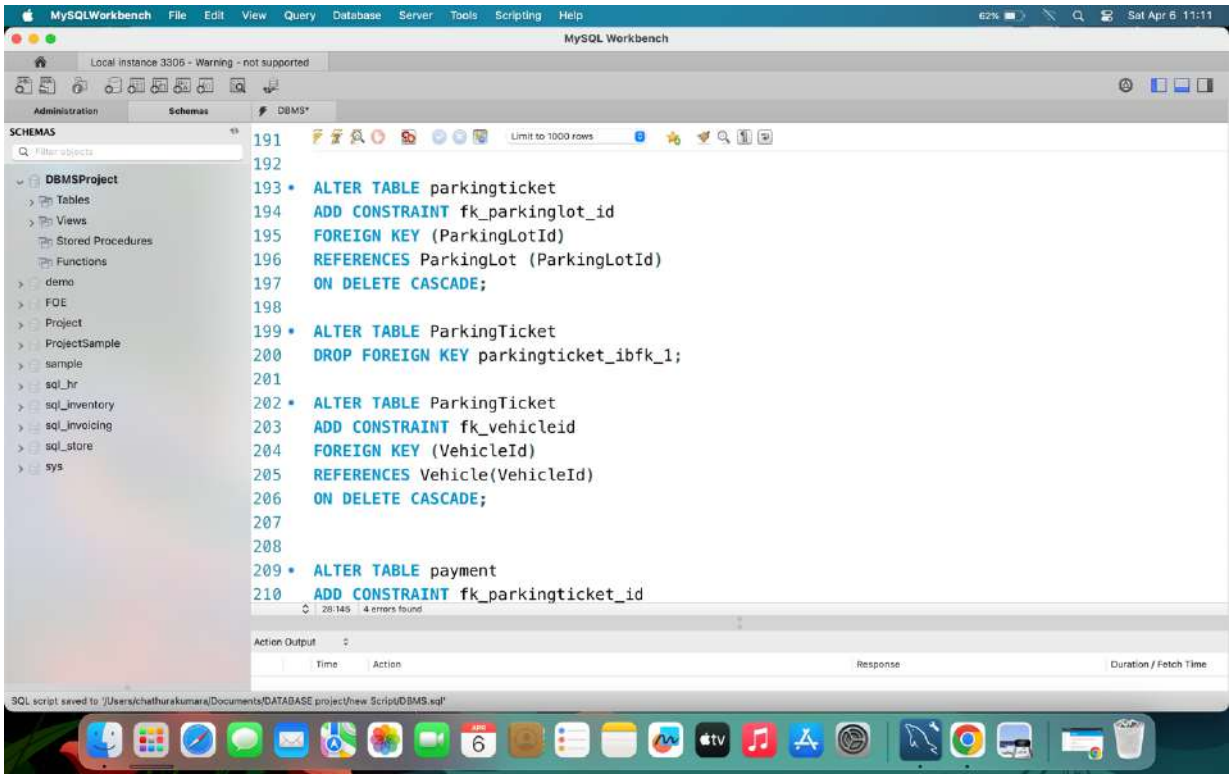


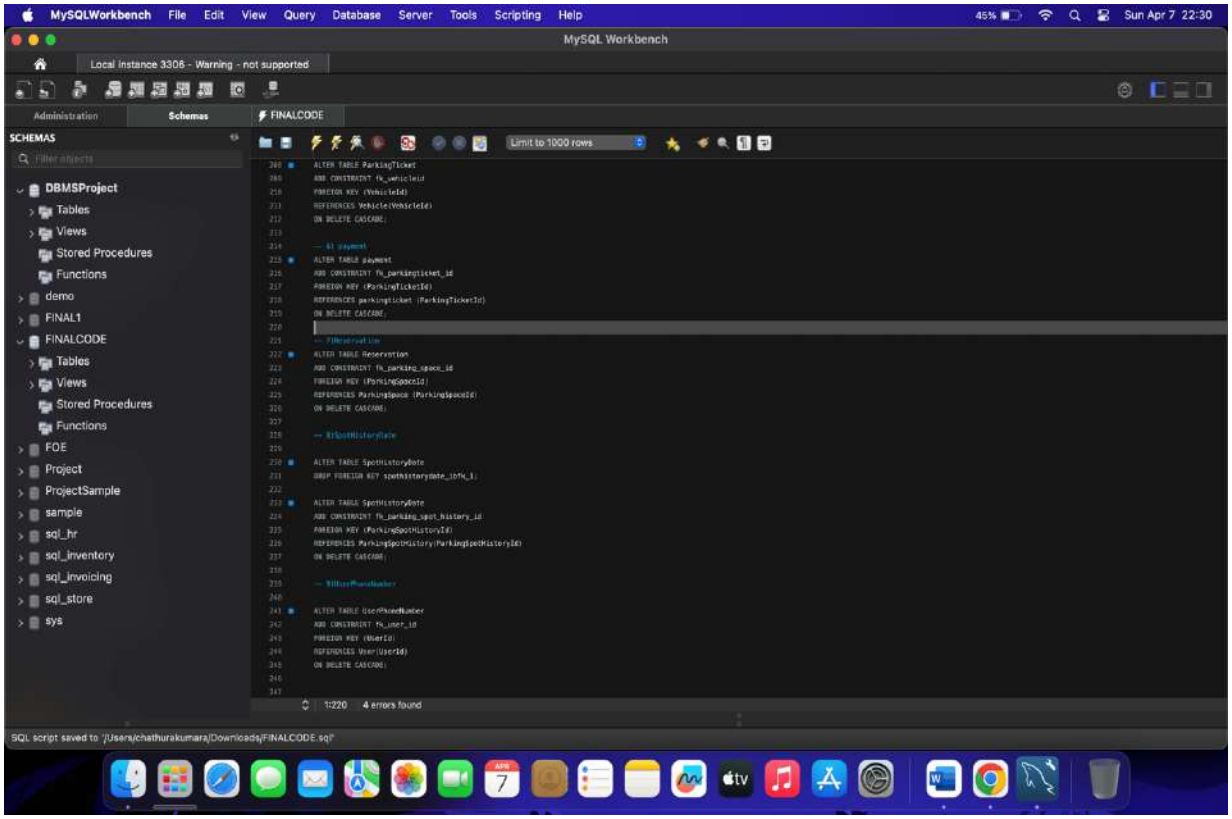
3.2 Creating Foreign Key Constraint

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



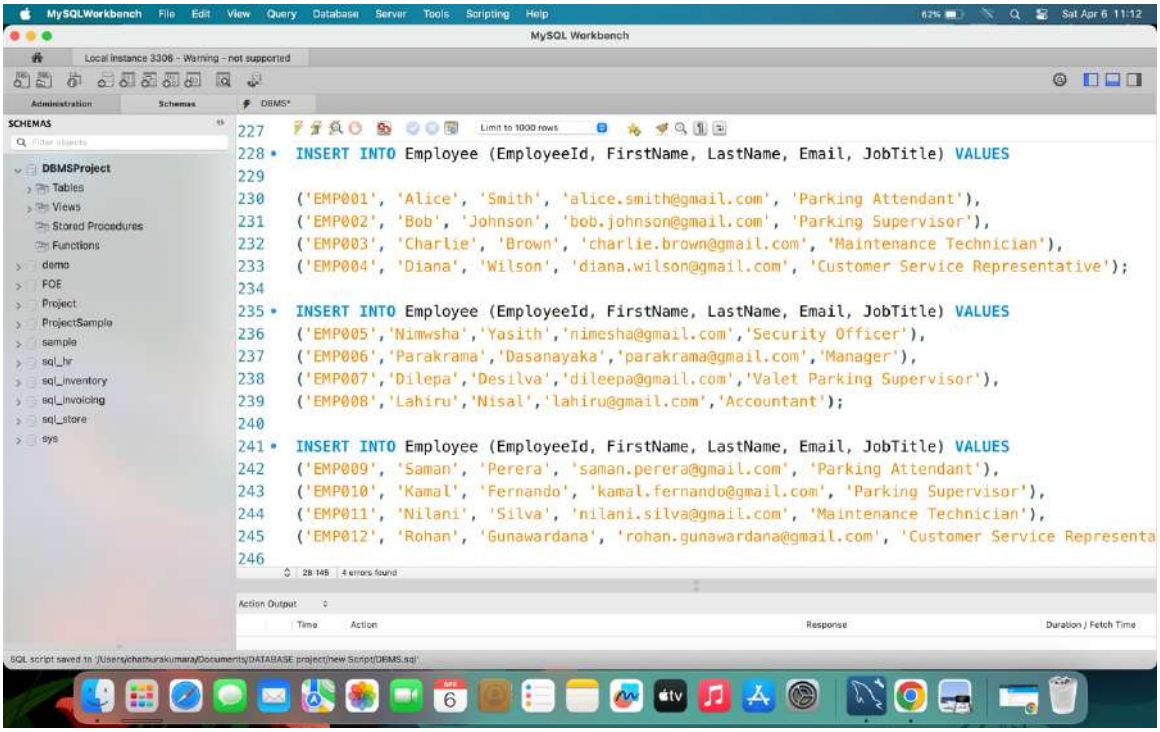



MySQLWorkbench File Edit View Query Database Server Tools Scripting Help
MySQL Workbench
Local instance 3306 - Warning - not supported
Administration Schemas
SCHEMAS
DBMSProject
Tables
Views
Stored Procedures
Functions
demo
FOE
Project
ProjectSample
sample
sql_hr
sql_inventory
sql_invoicing
sql_store
sys
DBMS*
Limit to 1000 rows
ALTER TABLE parkingticket
ADD CONSTRAINT fk_parkinglot_id
FOREIGN KEY (ParkingLotId)
REFERENCES ParkingLot (ParkingLotId)
ON DELETE CASCADE;
ALTER TABLE ParkingTicket
DROP FOREIGN KEY parkingticket_ibfk_1;
ALTER TABLE ParkingTicket
ADD CONSTRAINT fk_vehicleid
FOREIGN KEY (VehicleId)
REFERENCES Vehicle(VehicleId)
ON DELETE CASCADE;
ALTER TABLE payment
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'

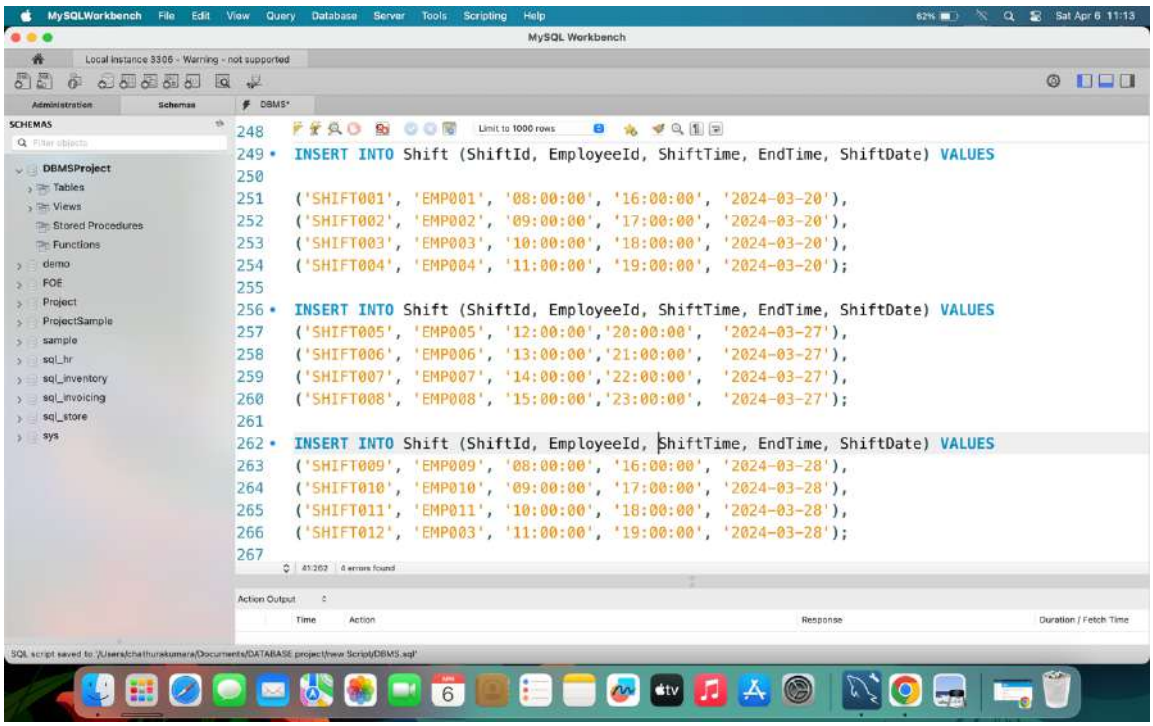

MySQLWorkbench File Edit View Query Database Server Tools Scripting Help
MySQL Workbench
Local instance 3306 - Warning - not supported
Administration Schemas
SCHEMAS
DBMSProject
Tables
Views
Stored Procedures
Functions
demo
FINAL1
FINALCODE
Tables
Views
Stored Procedures
Functions
FOE
Project
ProjectSample
sample
sql_hr
sql_inventory
sql_invoicing
sql_store
sys
FINALCODE
Limit to 1000 rows
1:220 4 errors found
SQL script saved to '/Users/chathurakumara/Downloads/FINALCODE.sql'

3.3 Inserting Data for Tables

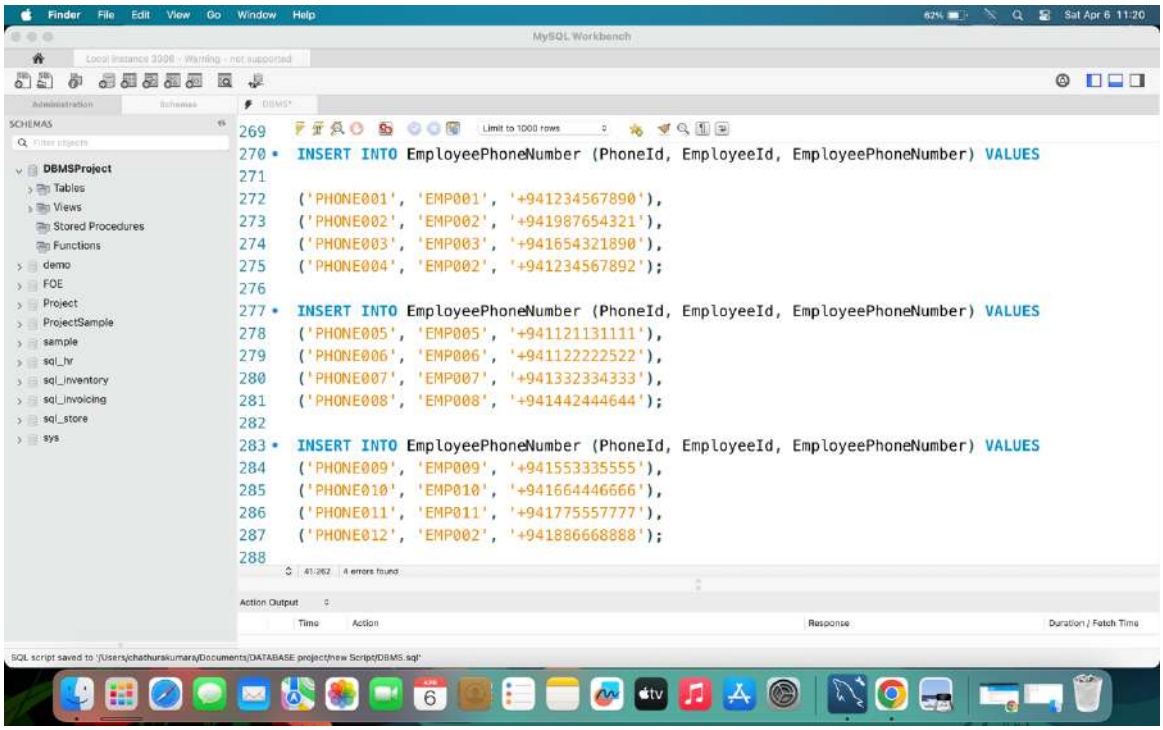
❖ Employee Table



❖ Shift

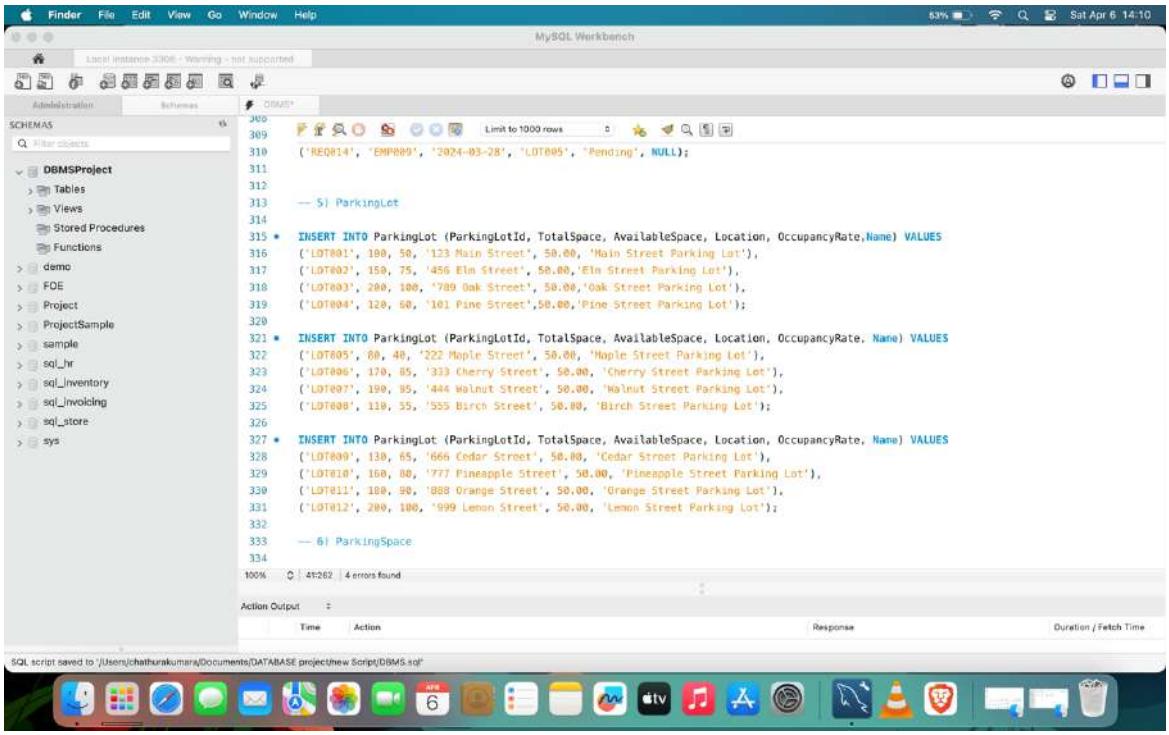


❖ Employee Phone Number



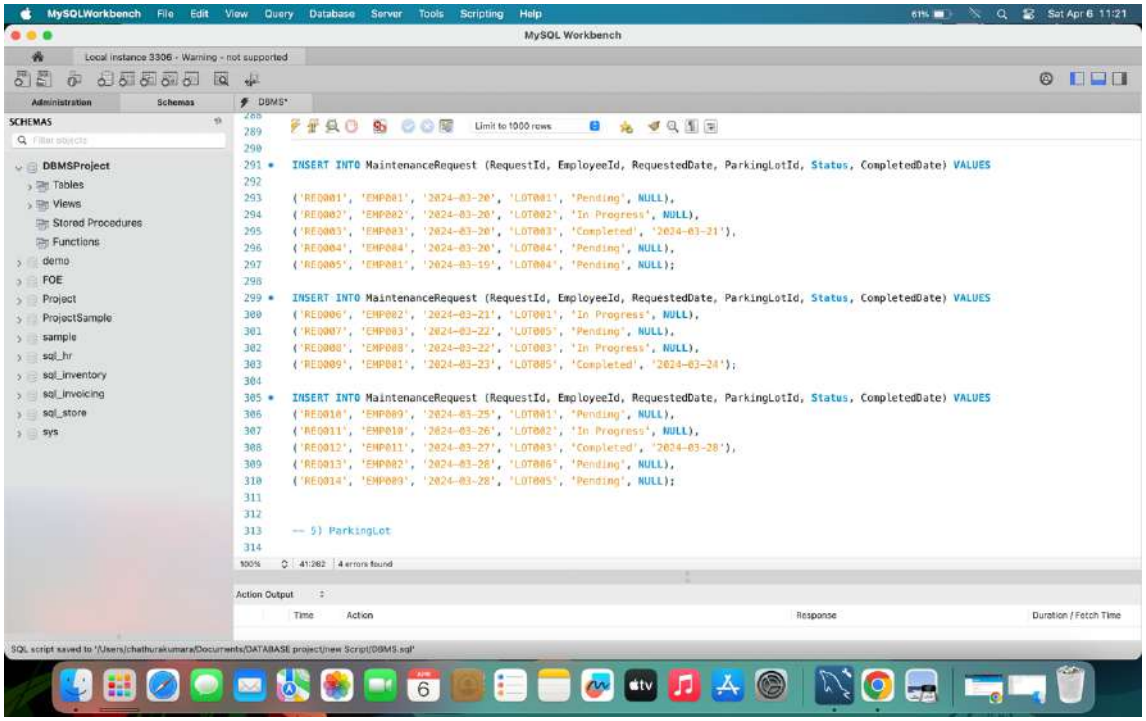
```
INSERT INTO EmployeePhoneNumber (PhoneId, EmployeeId, EmployeePhoneNumber) VALUES
('PHONE001', 'EMP001', '+941234567890'),
('PHONE002', 'EMP002', '+941987654321'),
('PHONE003', 'EMP003', '+941654321890'),
('PHONE004', 'EMP002', '+941234567892');

INSERT INTO EmployeePhoneNumber (PhoneId, EmployeeId, EmployeePhoneNumber) VALUES
('PHONE005', 'EMP005', '+941121131111'),
('PHONE006', 'EMP006', '+941122222522'),
('PHONE007', 'EMP007', '+941332334333'),
('PHONE008', 'EMP008', '+941442444644');

INSERT INTO EmployeePhoneNumber (PhoneId, EmployeeId, EmployeePhoneNumber) VALUES
('PHONE009', 'EMP009', '+941553335555'),
('PHONE010', 'EMP010', '+941664446666'),
('PHONE011', 'EMP011', '+941775557777'),
('PHONE012', 'EMP002', '+941886668888');
```

❖ Parking Lot

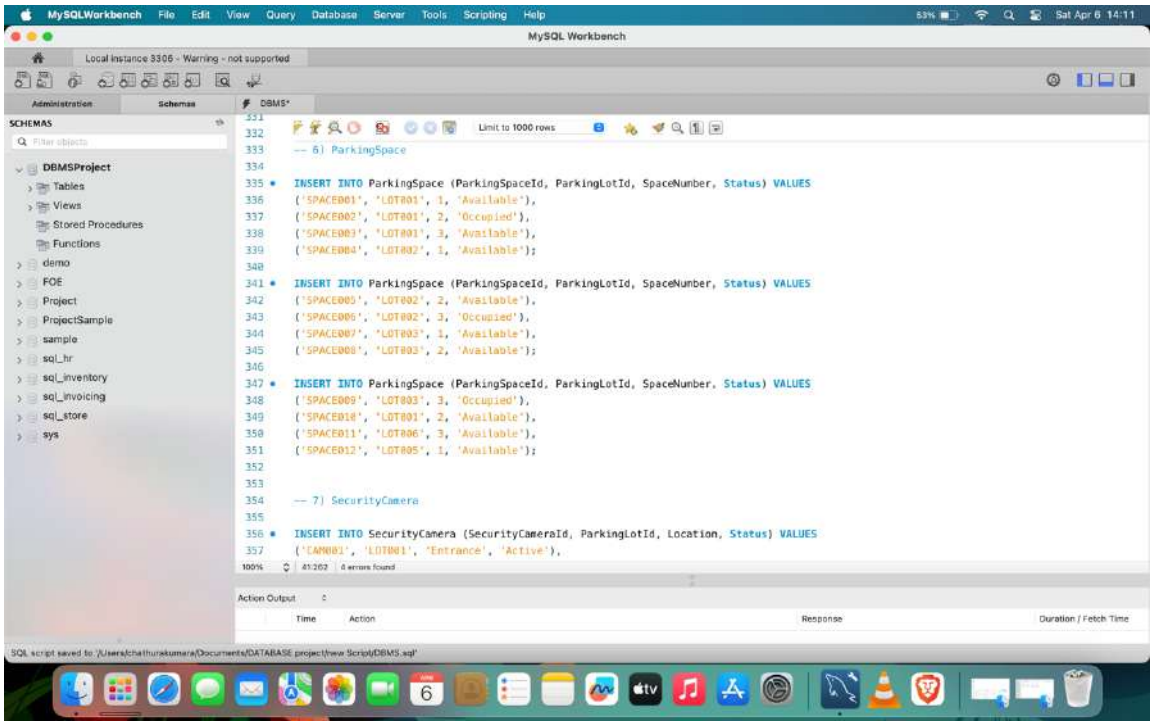


```
('REQ014', 'EMP009', '2024-03-28', 'LOT005', 'Pending', NULL);


-- 5) ParkingLot

INSERT INTO ParkingLot (ParkingLotId, TotalSpace, AvailableSpace, Location, OccupancyRate,Name) VALUES
('LOT001', 100, 50, '123 Main Street', 50.00, 'Main Street Parking Lot'),
('LOT002', 150, 75, '456 Elm Street', 50.00,'Elm Street Parking Lot'),
('LOT003', 200, 100, '789 Oak Street', 50.00,'Oak Street Parking Lot'),
('LOT004', 120, 60, '101 Pine Street',50.00,'Pine Street Parking Lot');

INSERT INTO ParkingLot (ParkingLotId, TotalSpace, AvailableSpace, Location, OccupancyRate, Name) VALUES
('LOT005', 80, 40, '222 Maple Street', 50.00, 'Maple Street Parking Lot'),
('LOT006', 170, 85, '333 Cherry Street', 50.00, 'Cherry Street Parking Lot'),
('LOT007', 190, 95, '444 Walnut Street', 50.00, 'Walnut Street Parking Lot'),
('LOT008', 110, 55, '555 Birch Street', 50.00, 'Birch Street Parking Lot');

INSERT INTO ParkingLot (ParkingLotId, TotalSpace, AvailableSpace, Location, OccupancyRate, Name) VALUES
('LOT009', 130, 65, '666 Cedar Street', 50.00, 'Cedar Street Parking Lot'),
('LOT010', 160, 80, '777 Pineapple Street', 50.00, 'Pineapple Street Parking Lot'),
('LOT011', 180, 90, '888 Orange Street', 50.00, 'Orange Street Parking Lot'),
('LOT012', 200, 100, '999 Lemon Street', 50.00, 'Lemon Street Parking Lot');

-- 6) ParkingSpace
```

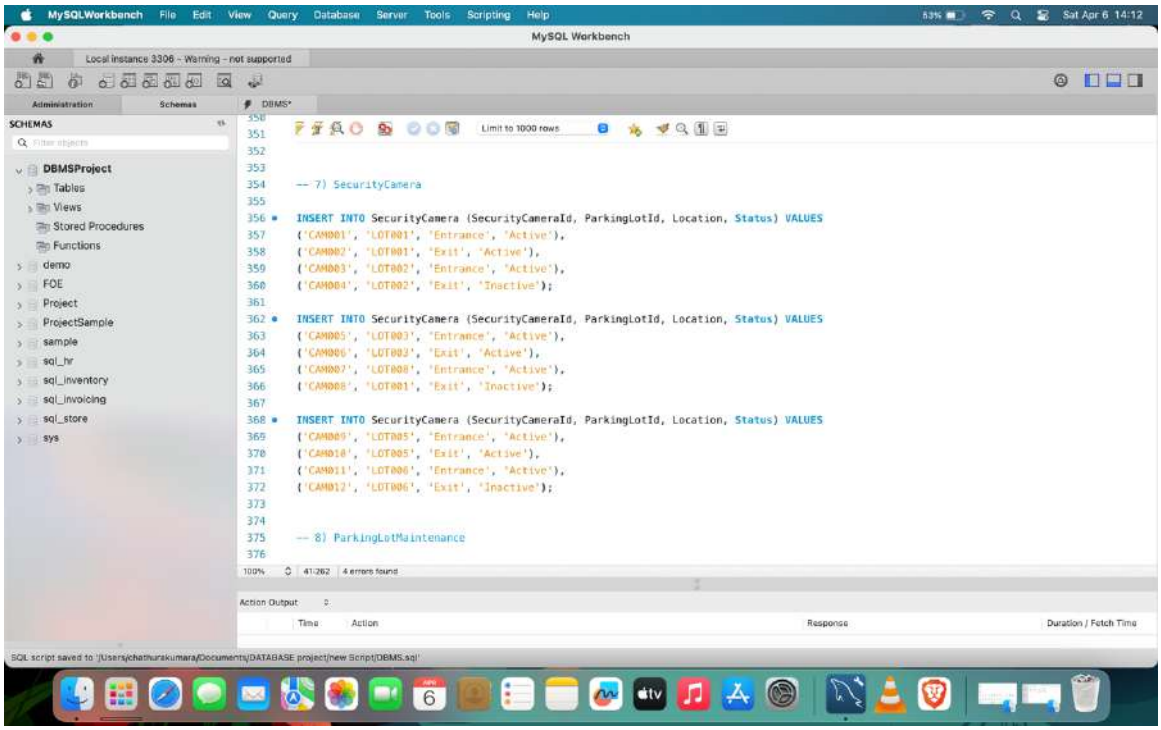
❖ Maintenance Request



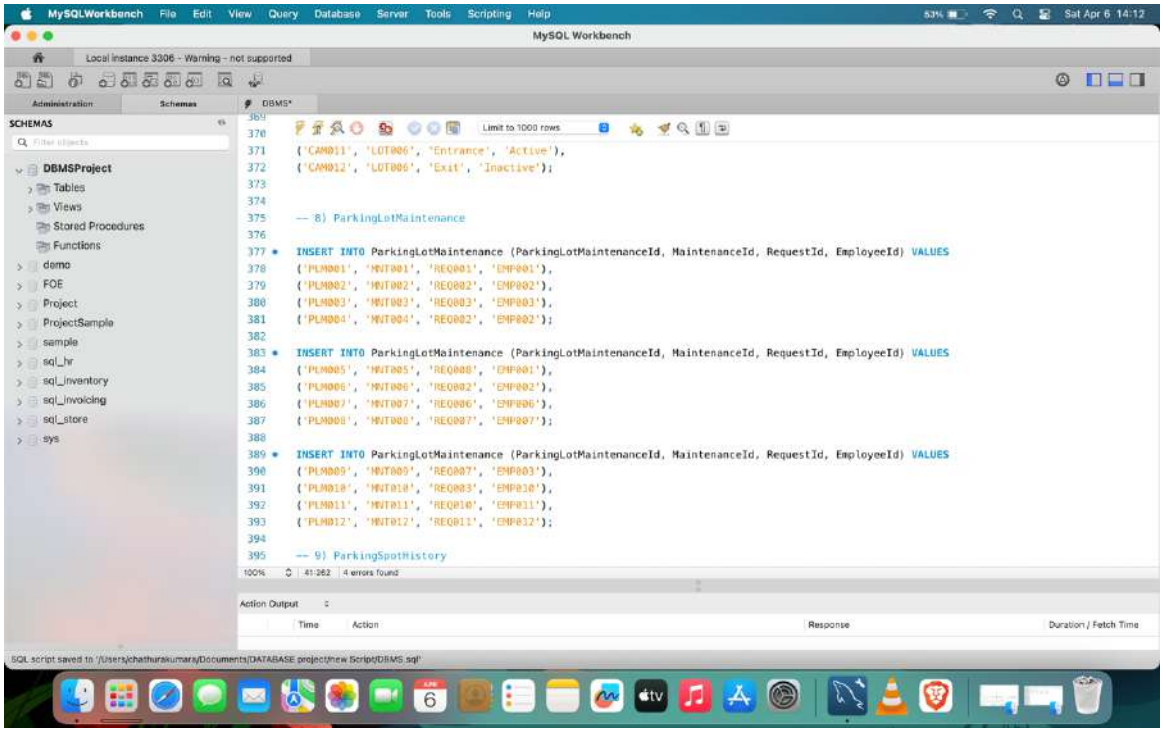
❖ Parking Space



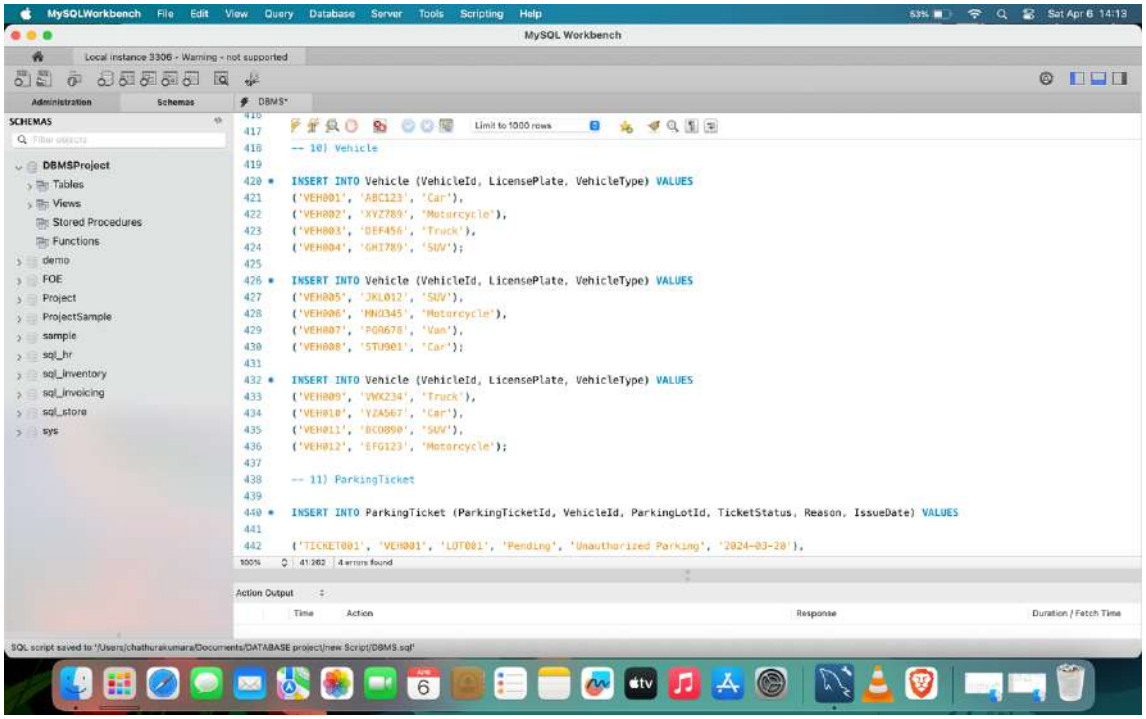
❖ Security Camera



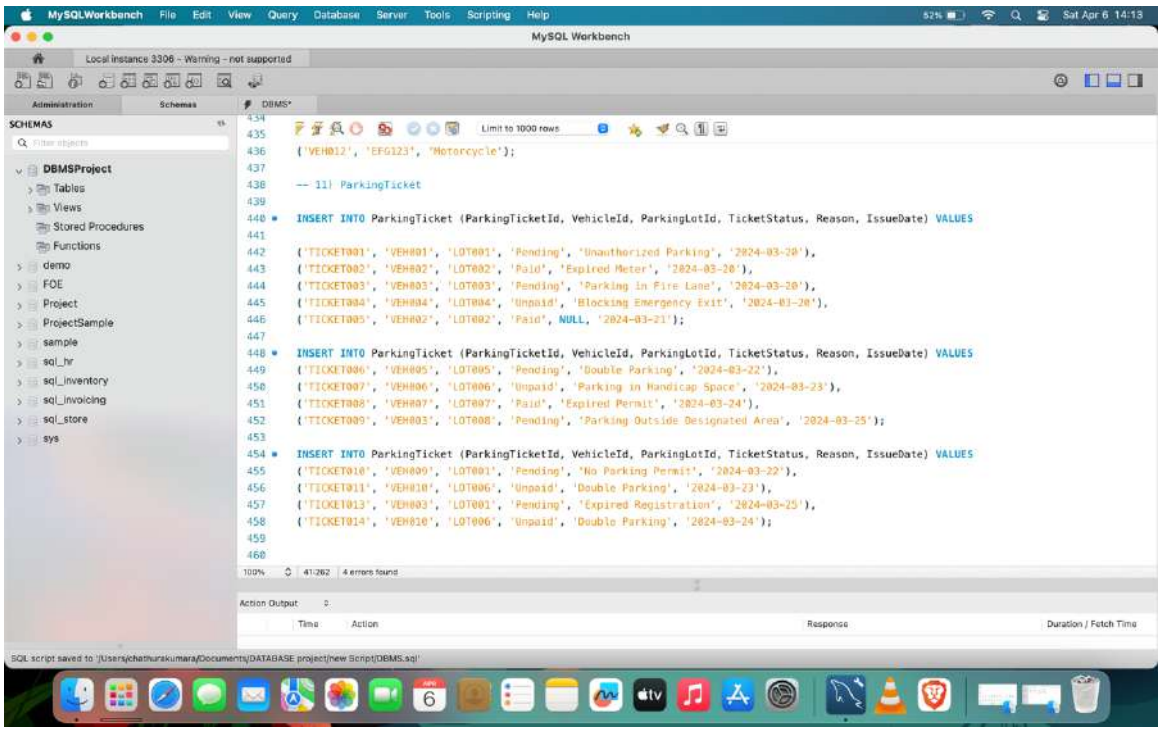
❖ Parking Lot Maintenance



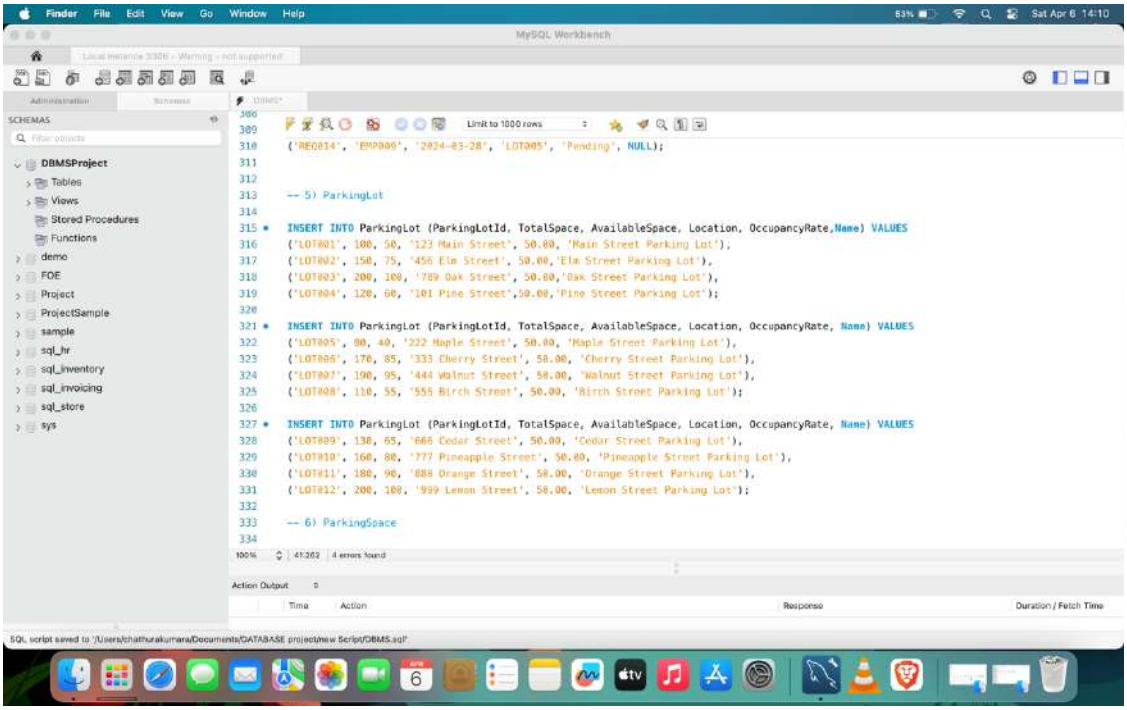
❖ Vehicle



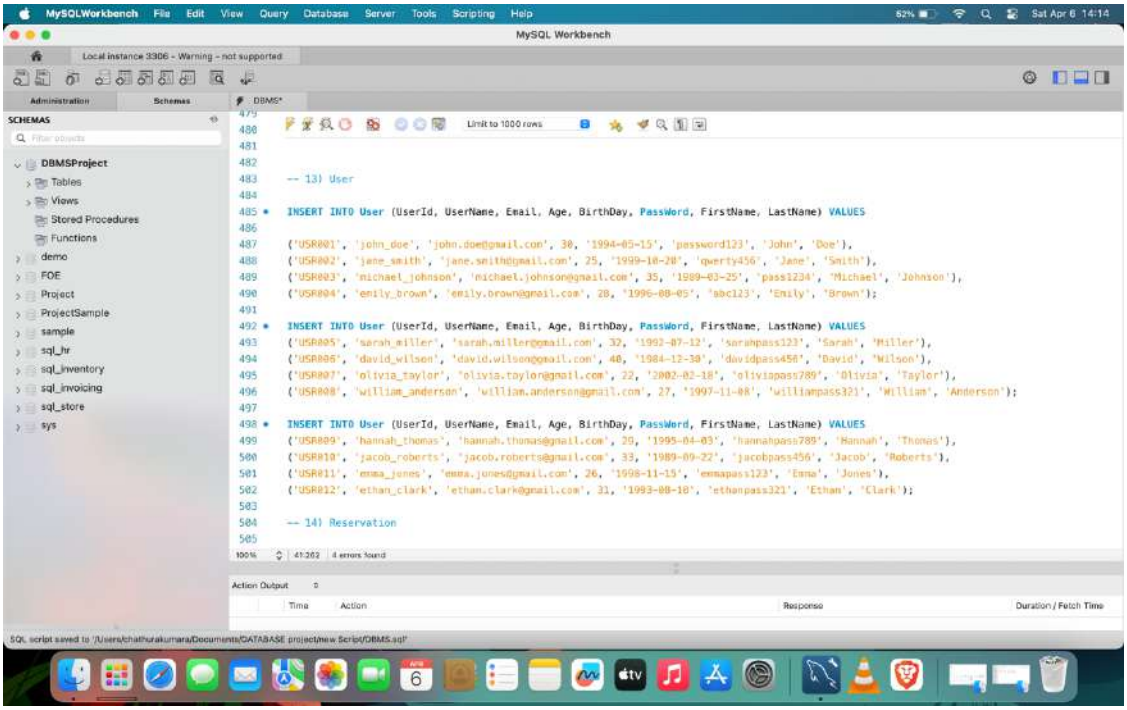
❖ Parking Ticket



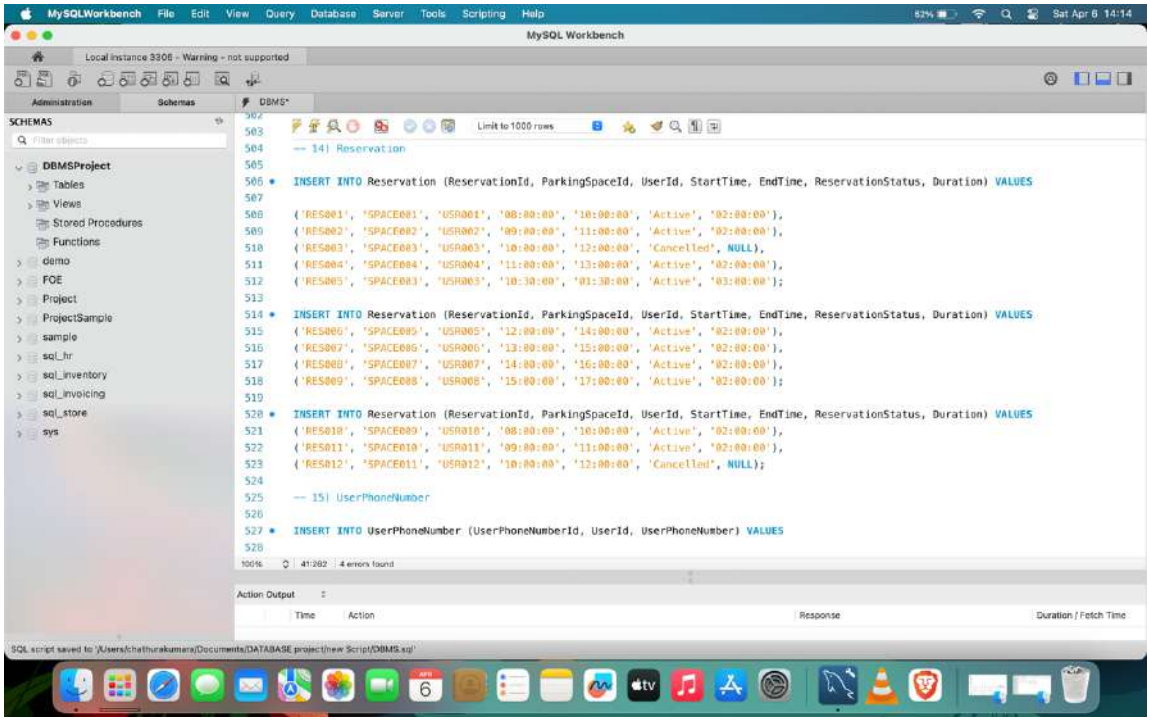
❖ Payment



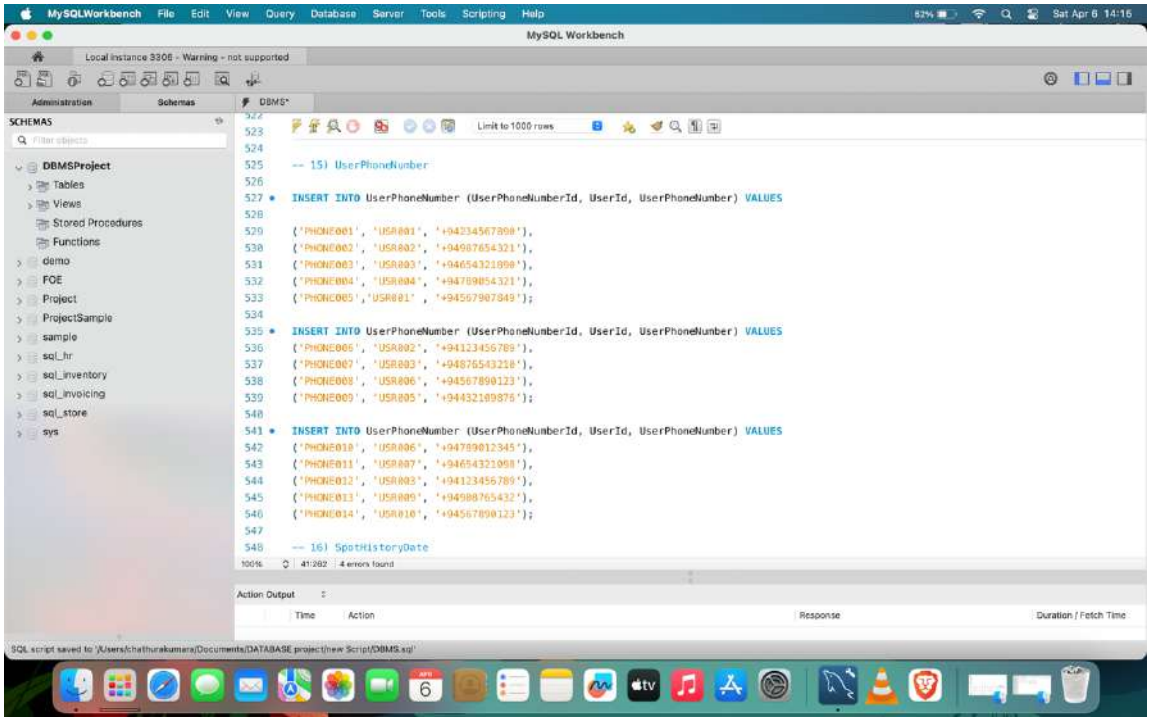
❖ User



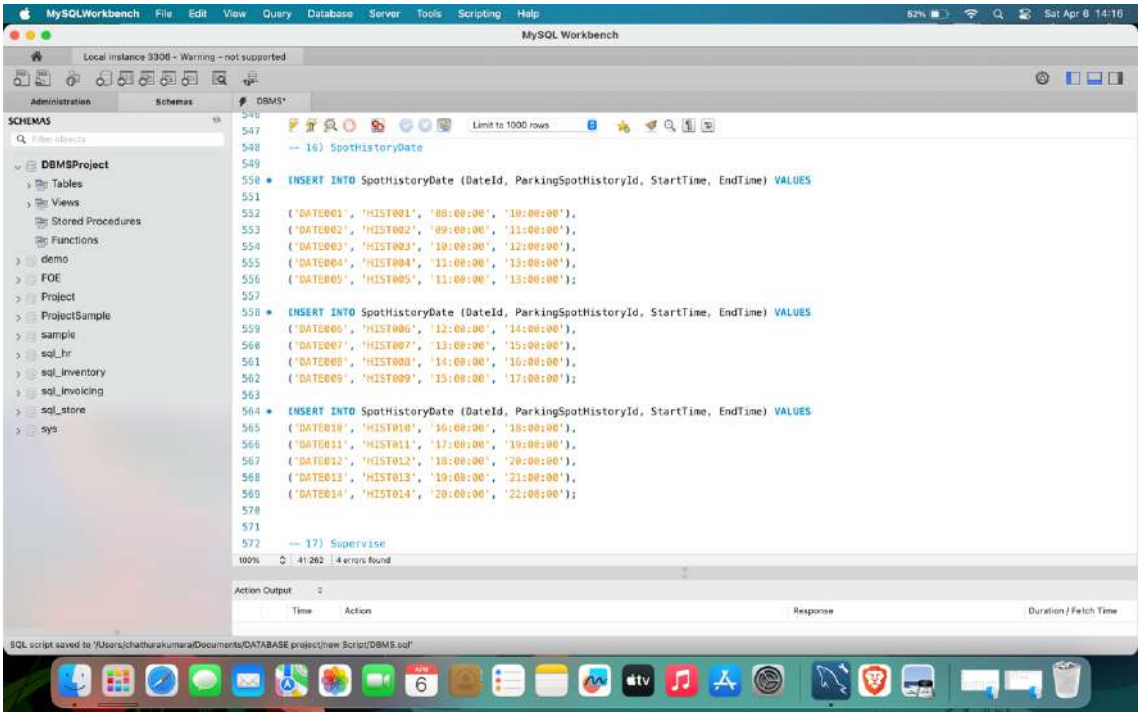
❖ Reservation



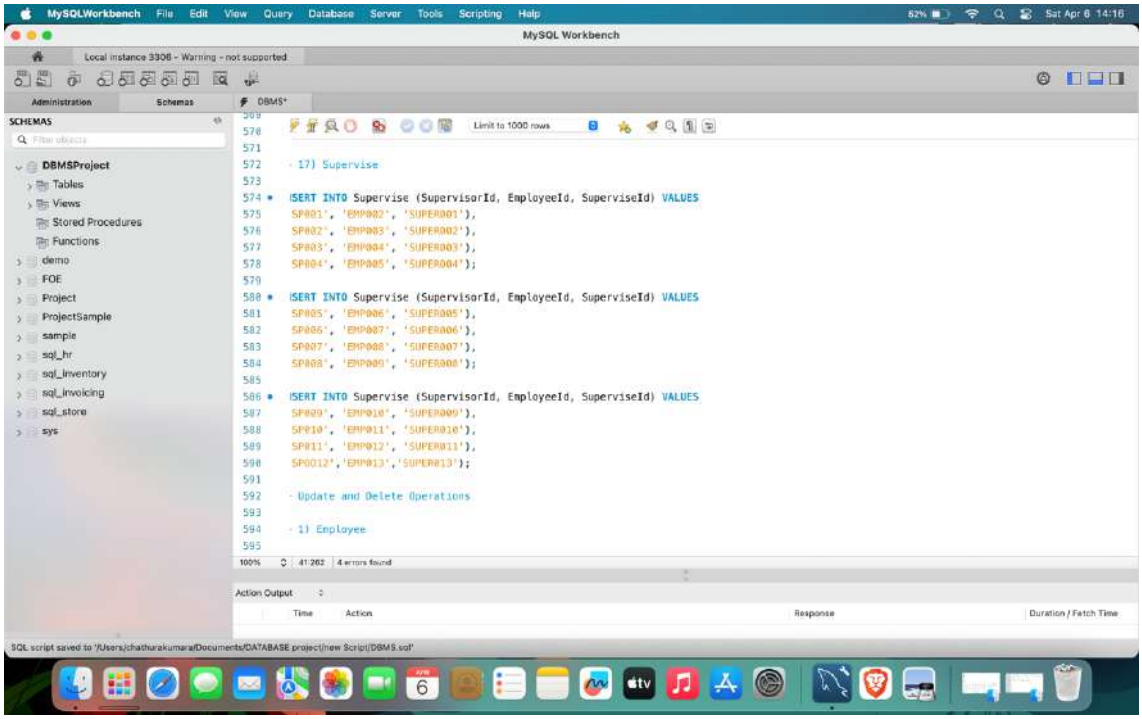
❖ Phone Number



❖ SpotHistoryDate.

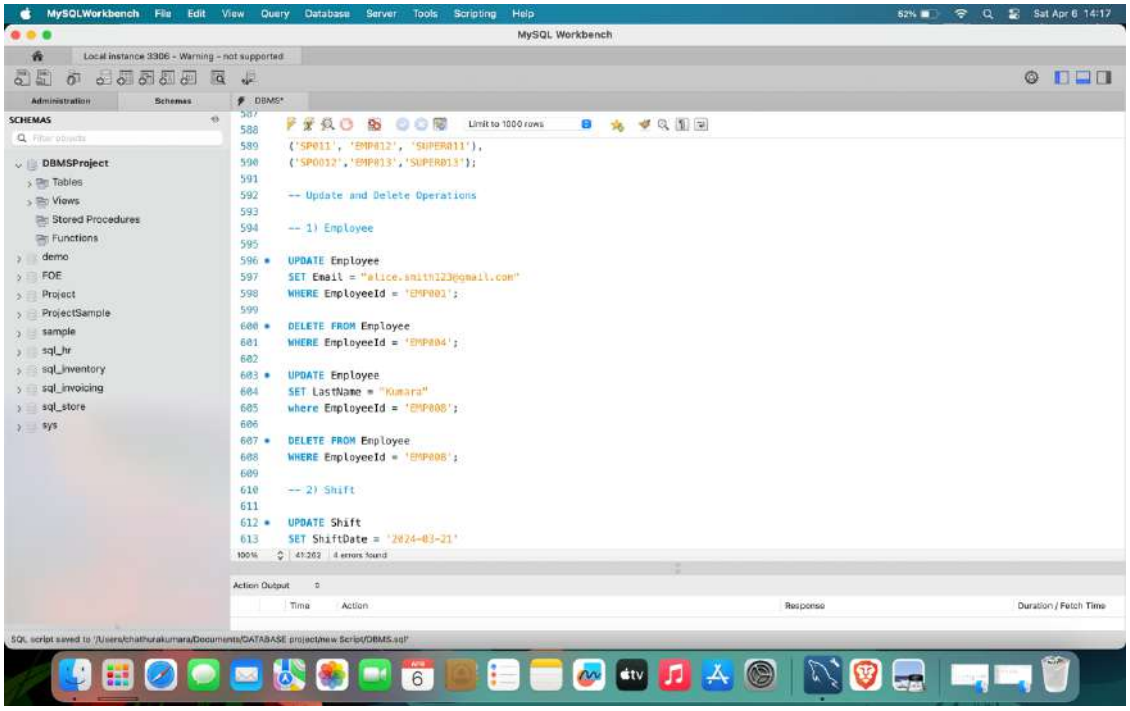


❖ Supervise

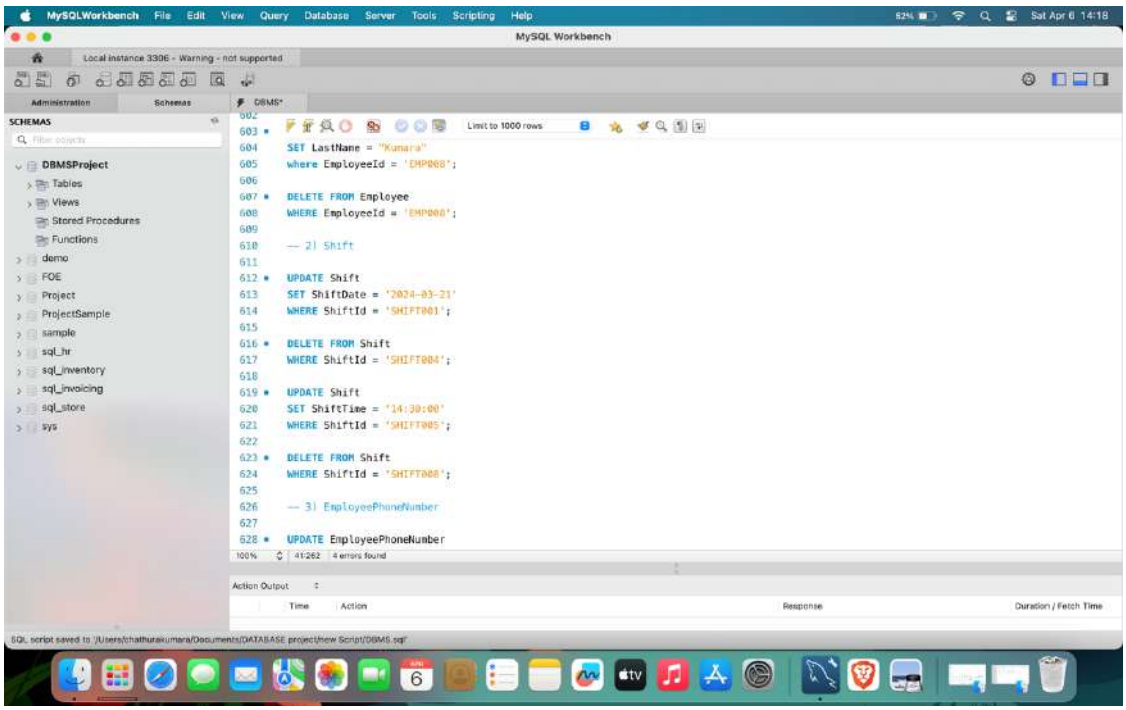


3.4 Update And Delete

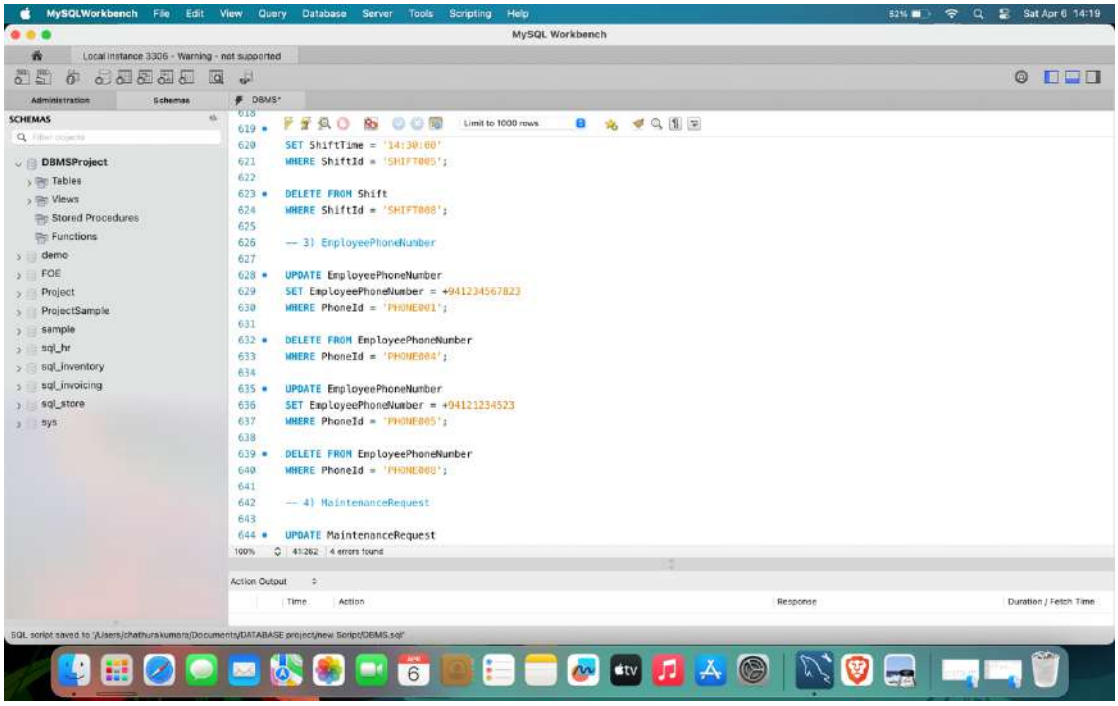
❖ Employee



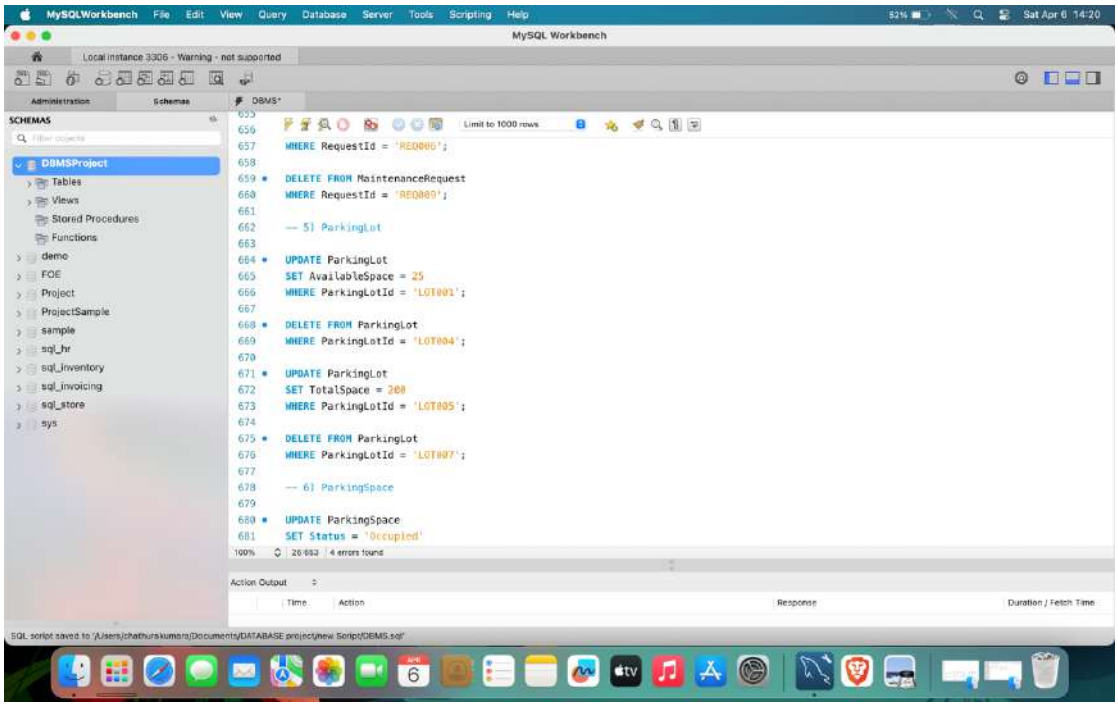
❖ Shift



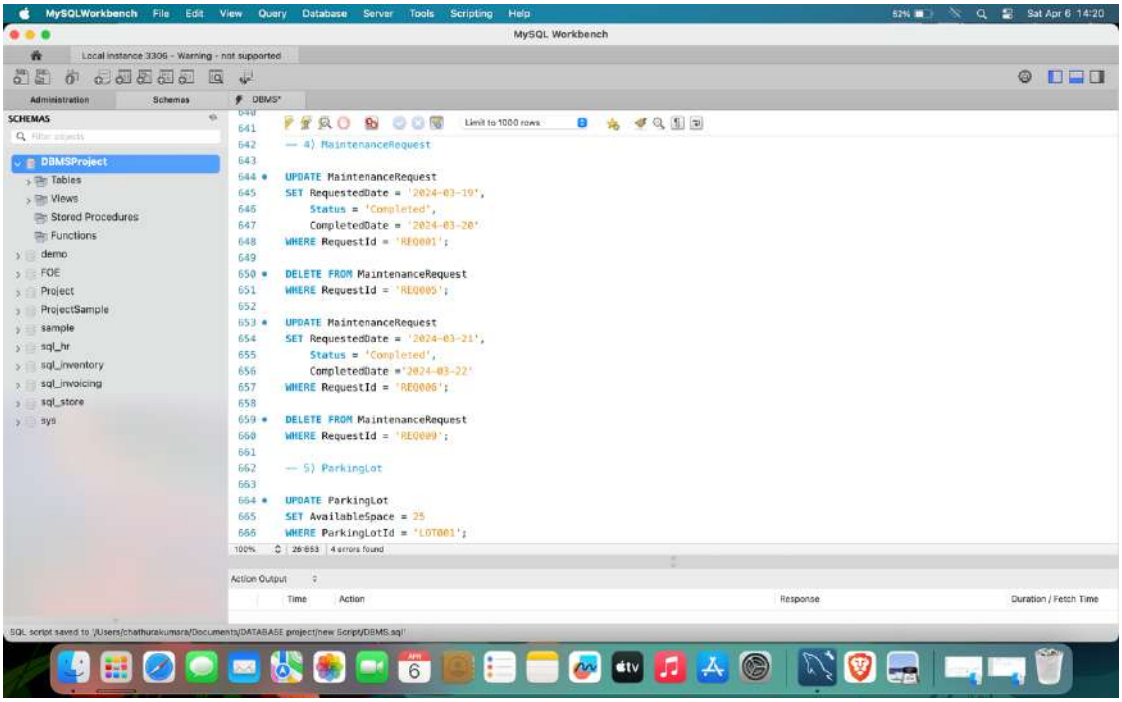
❖ Employee Phone Number



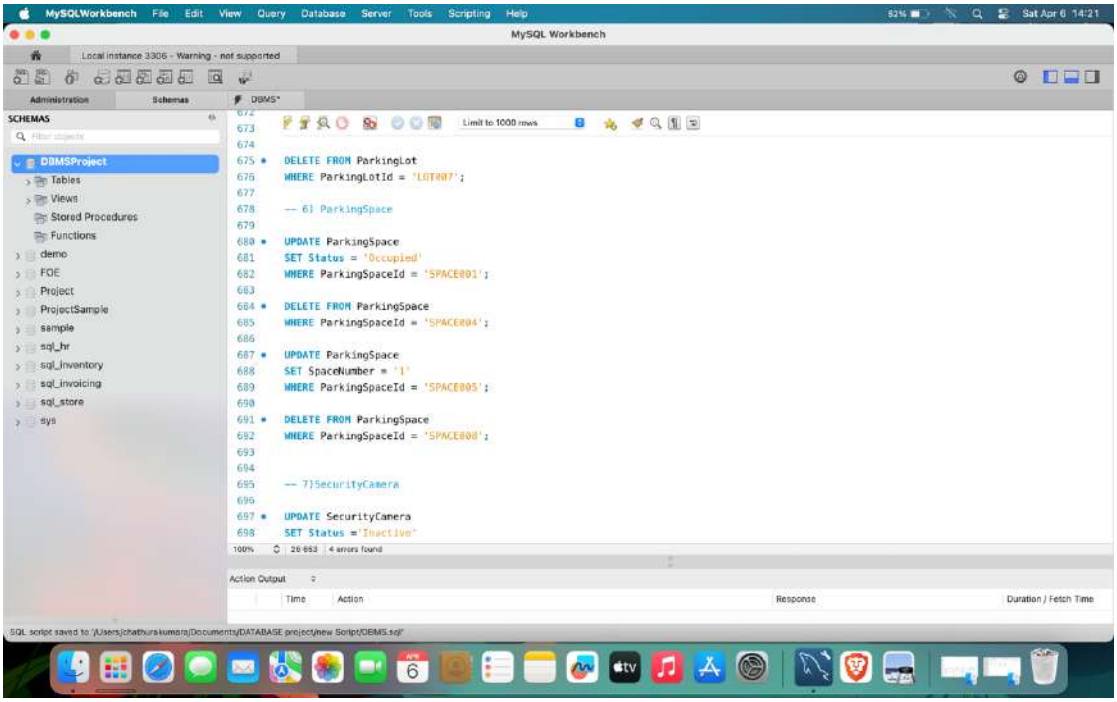
❖ Parking Lot



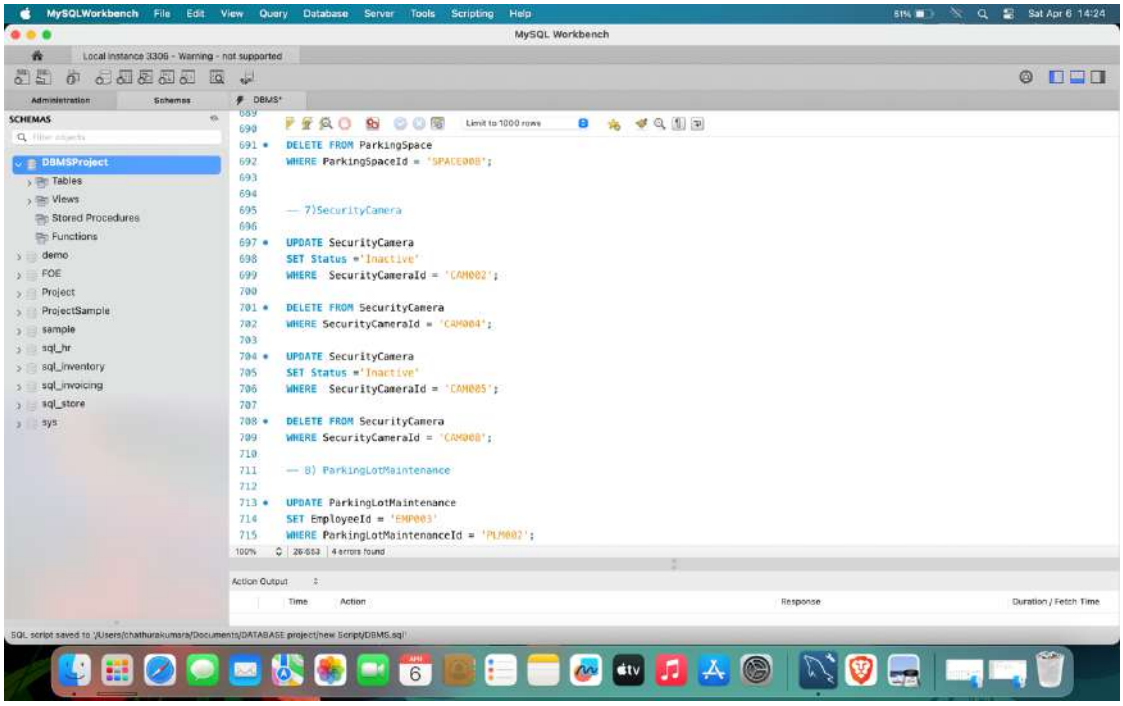
❖ Maintenance Request



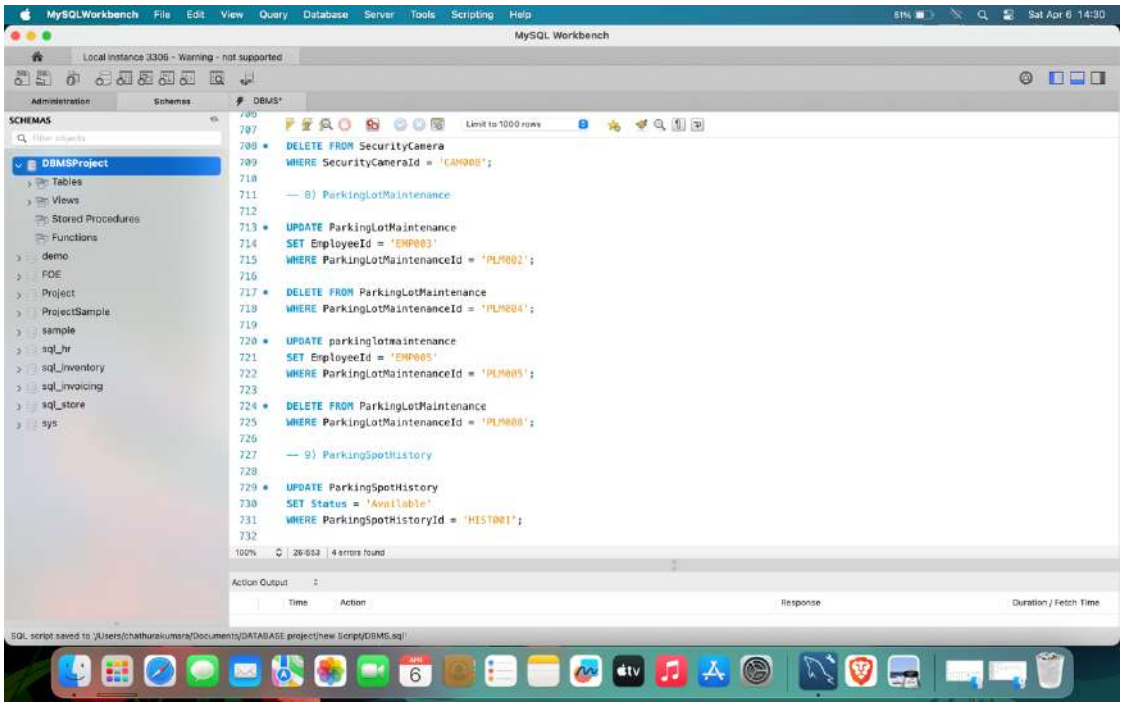
❖ Parking Space



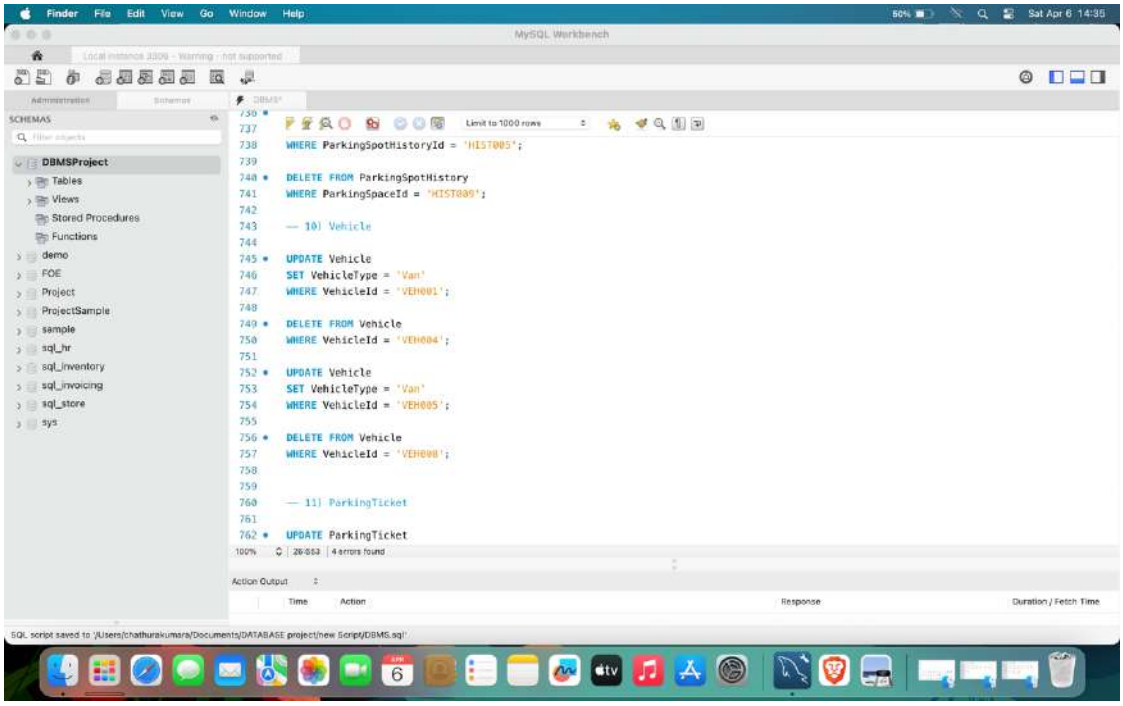
❖ Security Camera



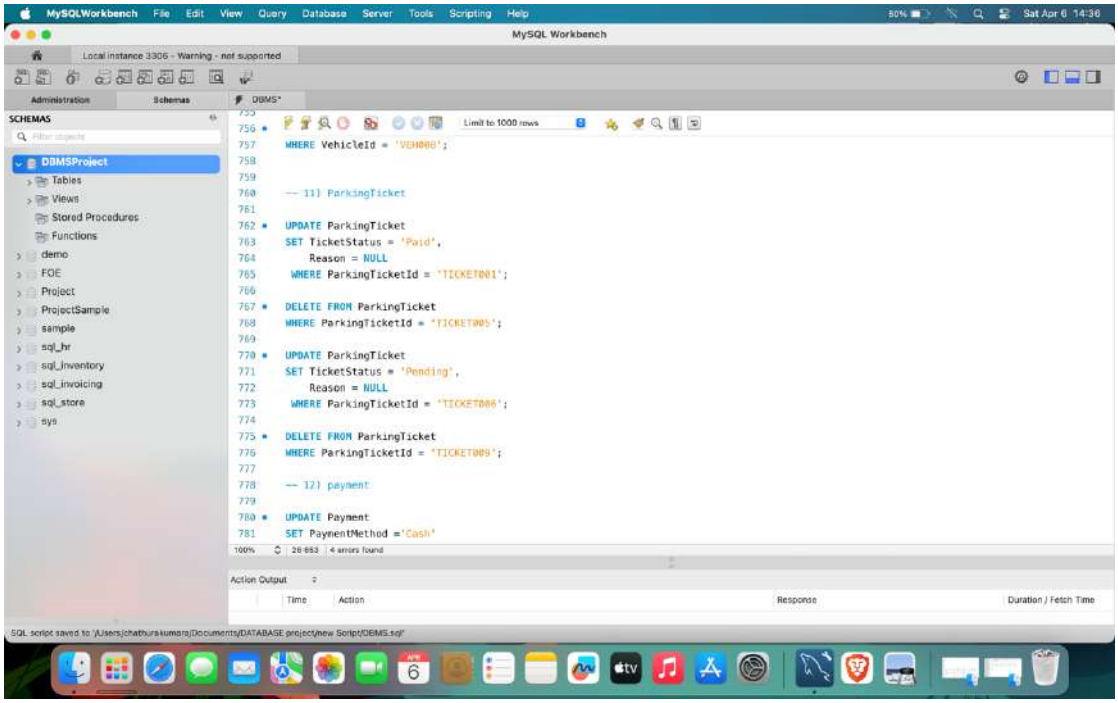
❖ Parking Lot Maintenance



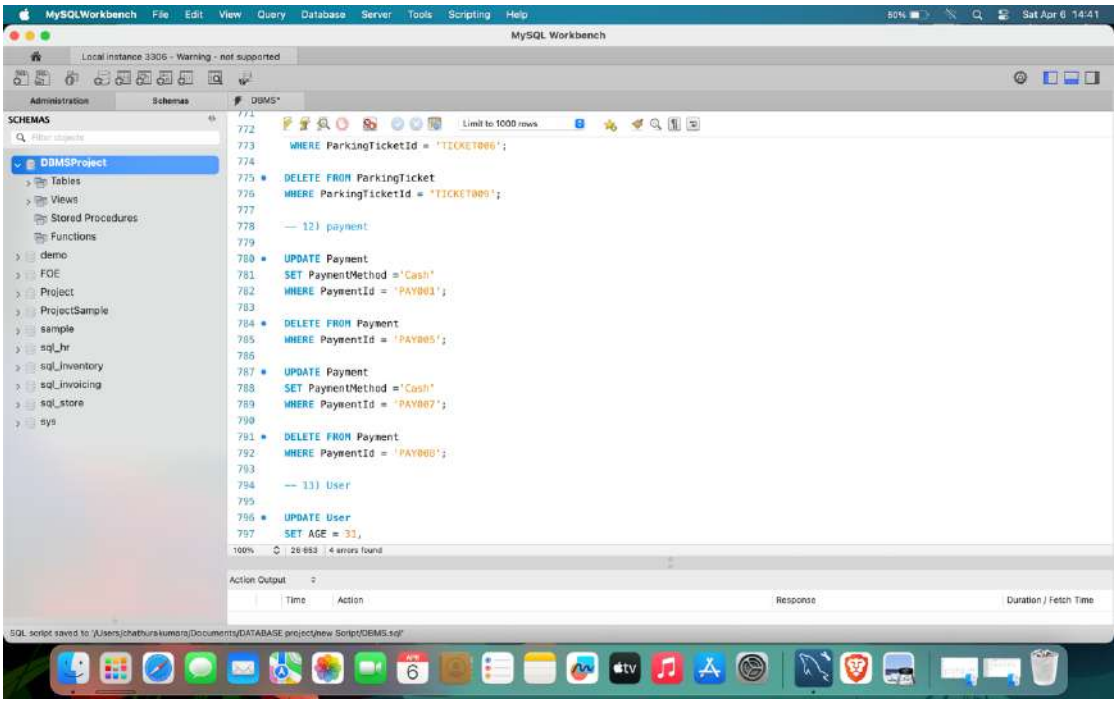
❖ Vehicle



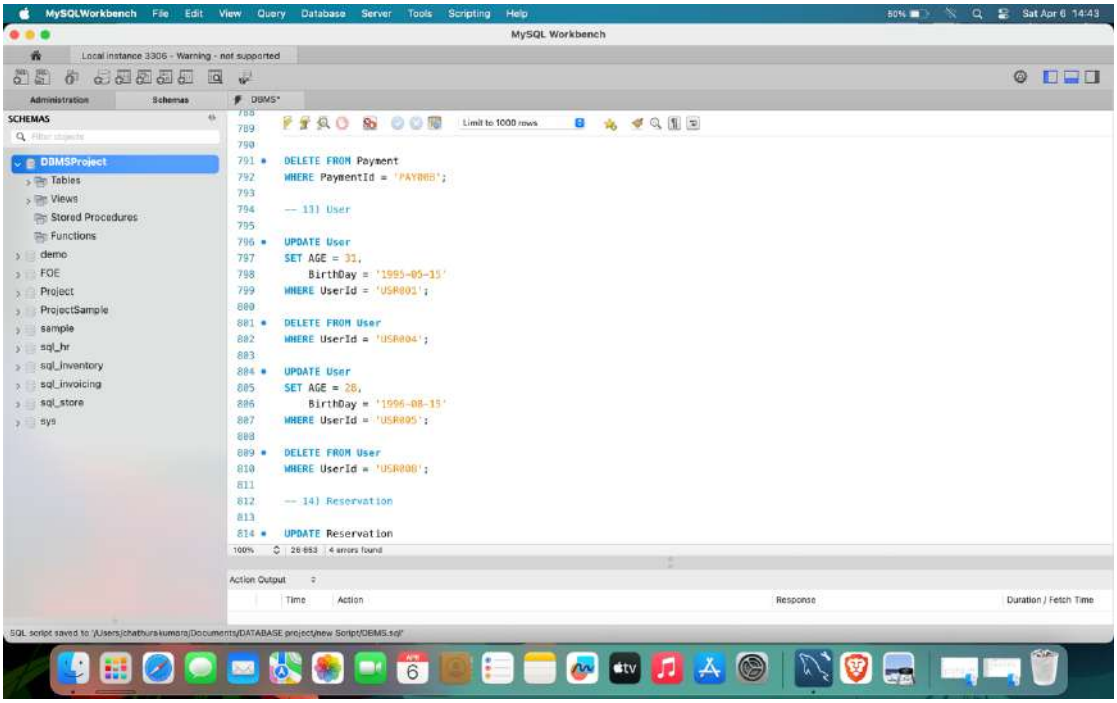
❖ ParkingTicket.



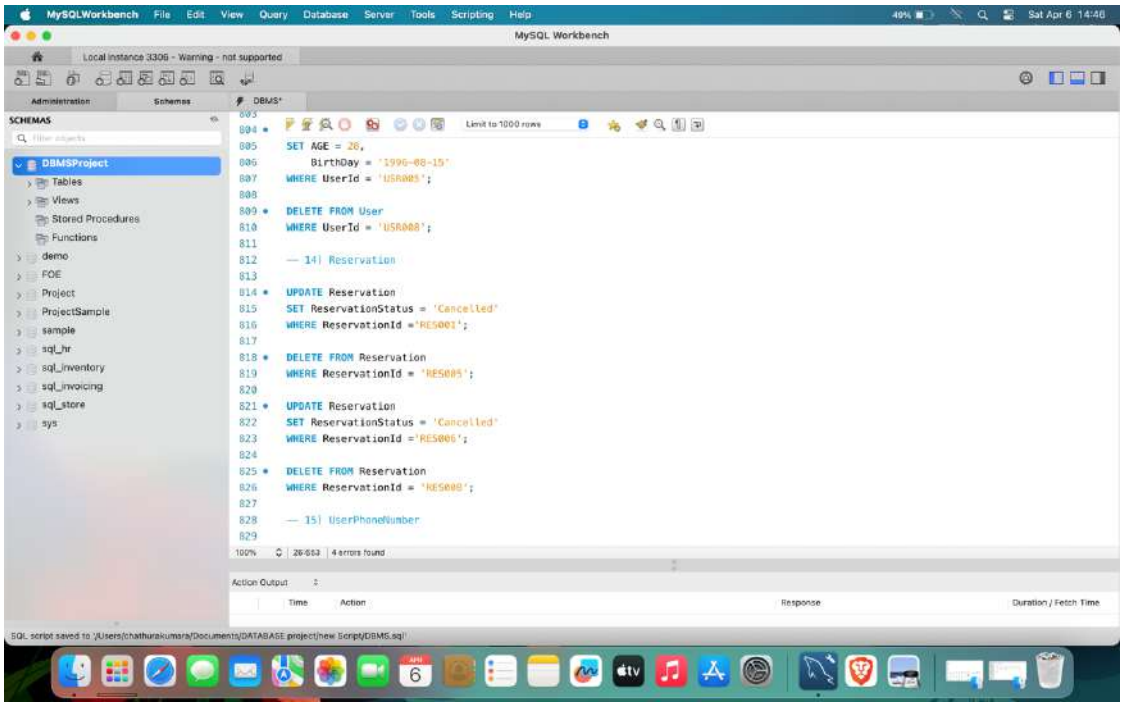
❖ Payment



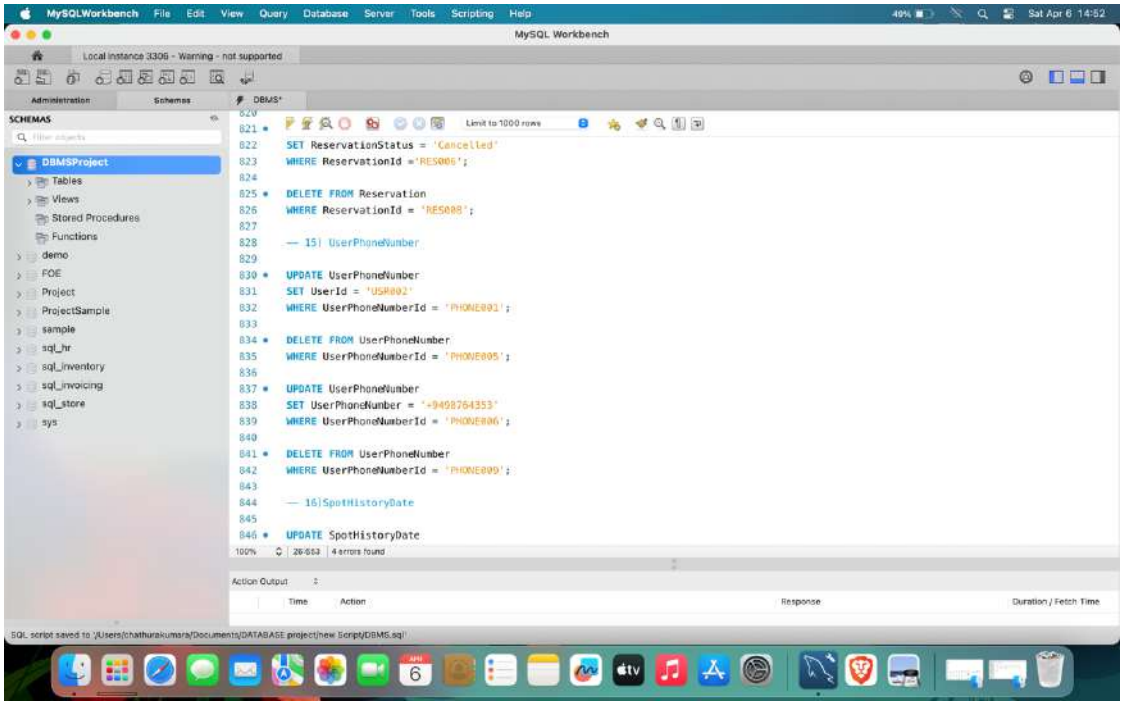
❖ User



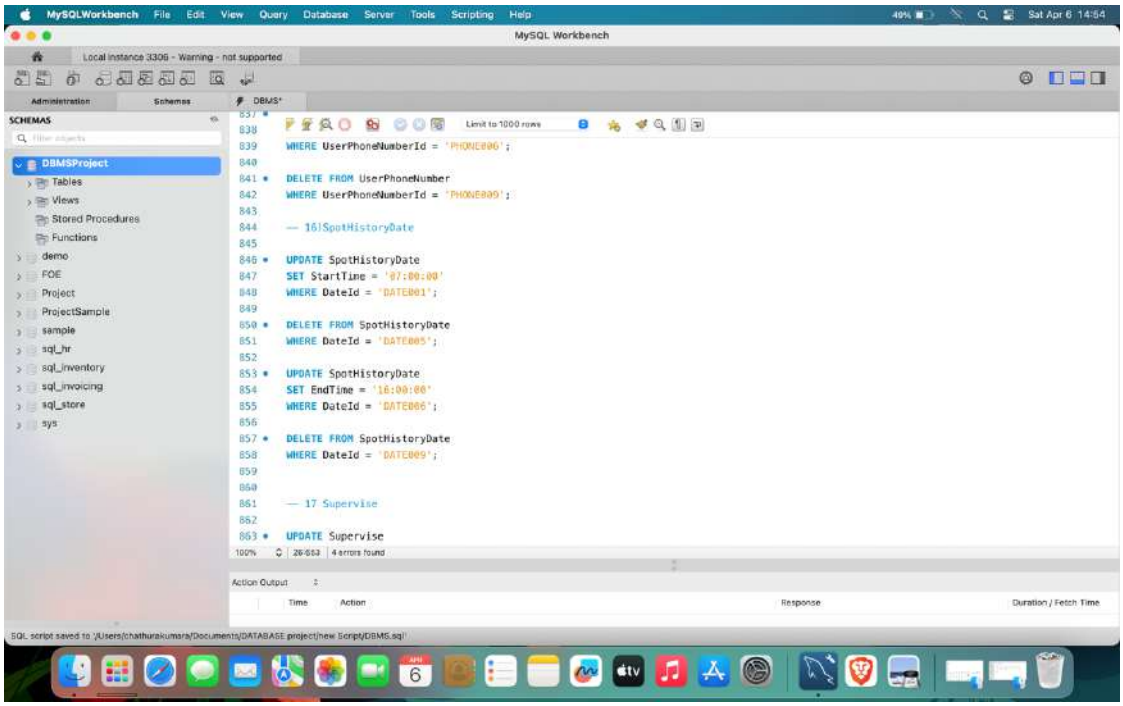
❖ Reservation



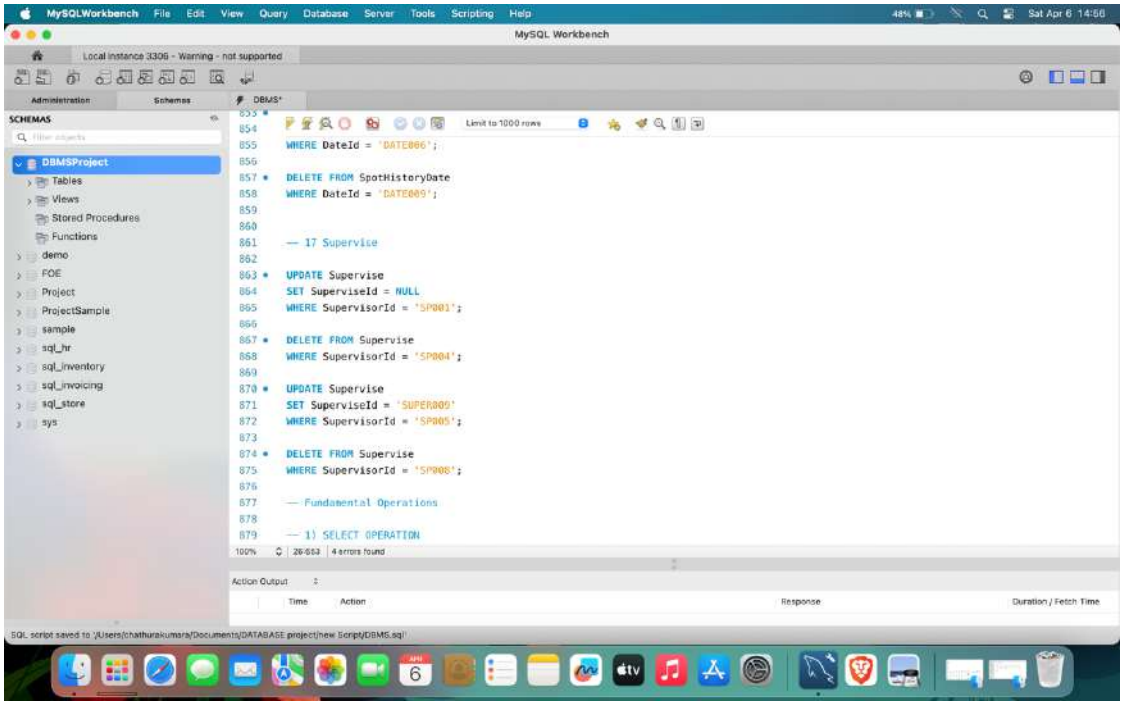
❖ Phone Number



❖ SpotHistoryDate

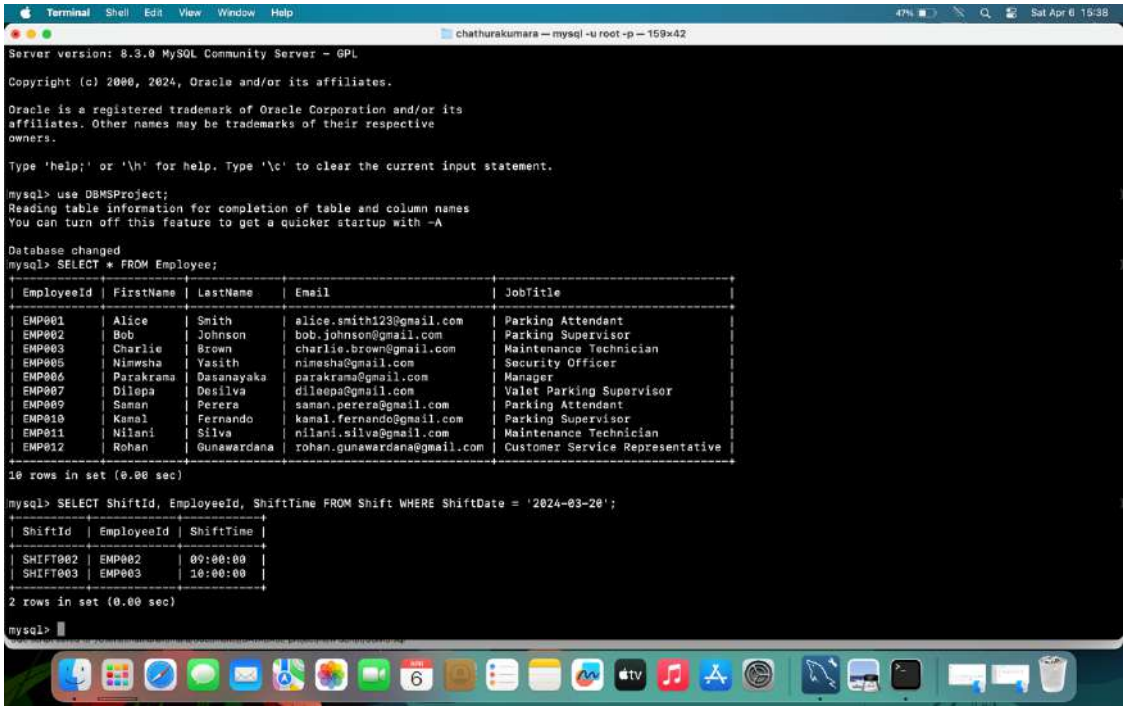


❖ Supervise

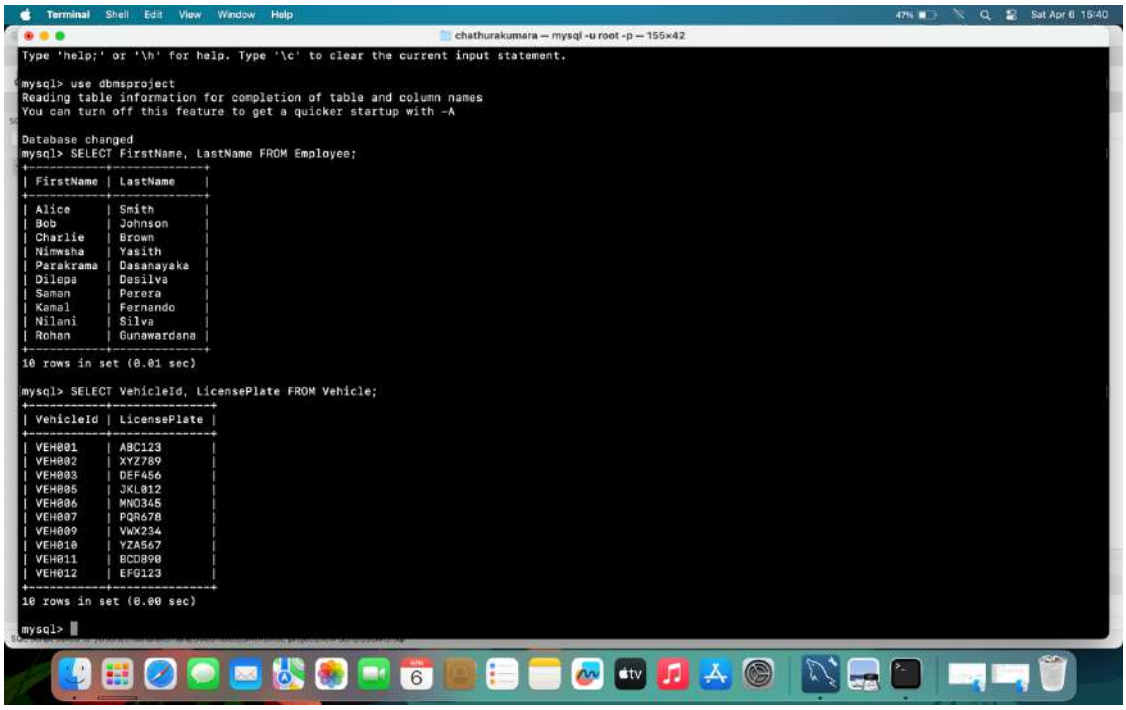


Chapter 4: Transaction

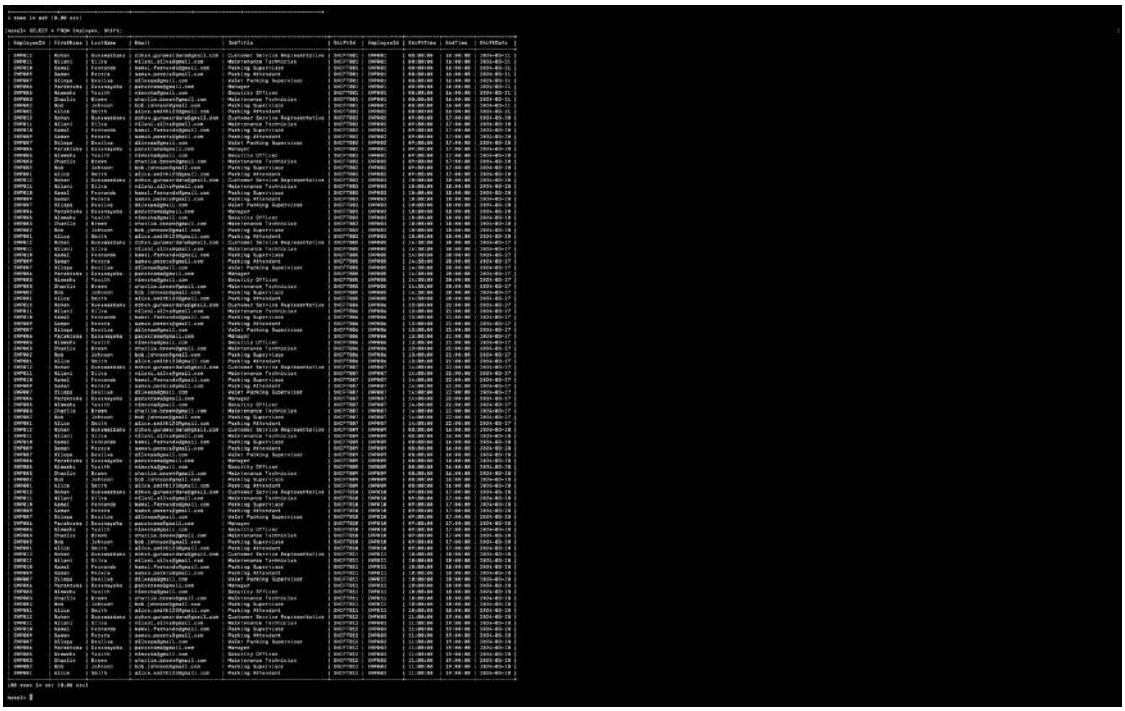
❖ Select Operation



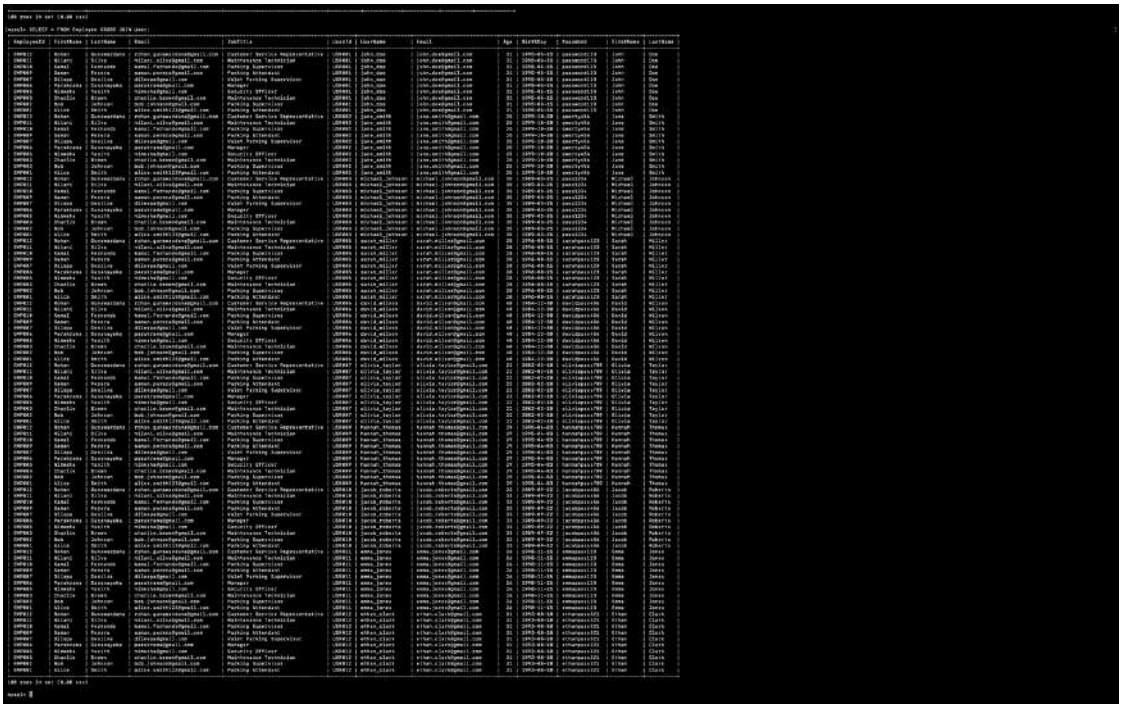
❖ Project Operation



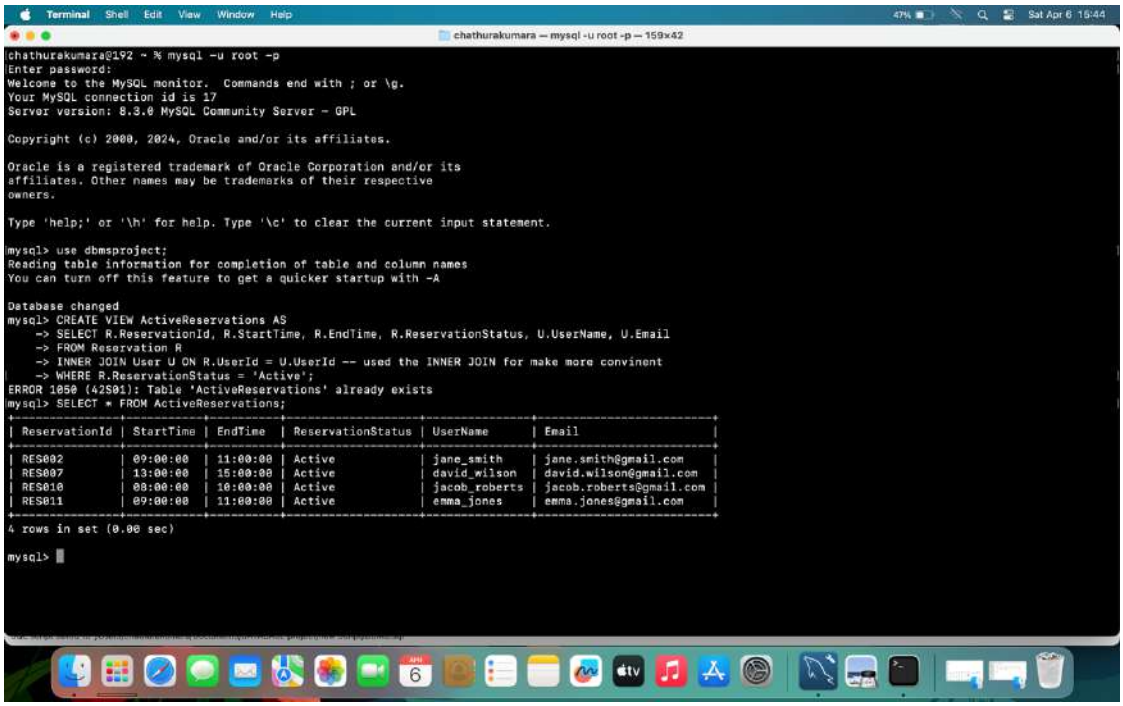
❖ Cartesian product 1



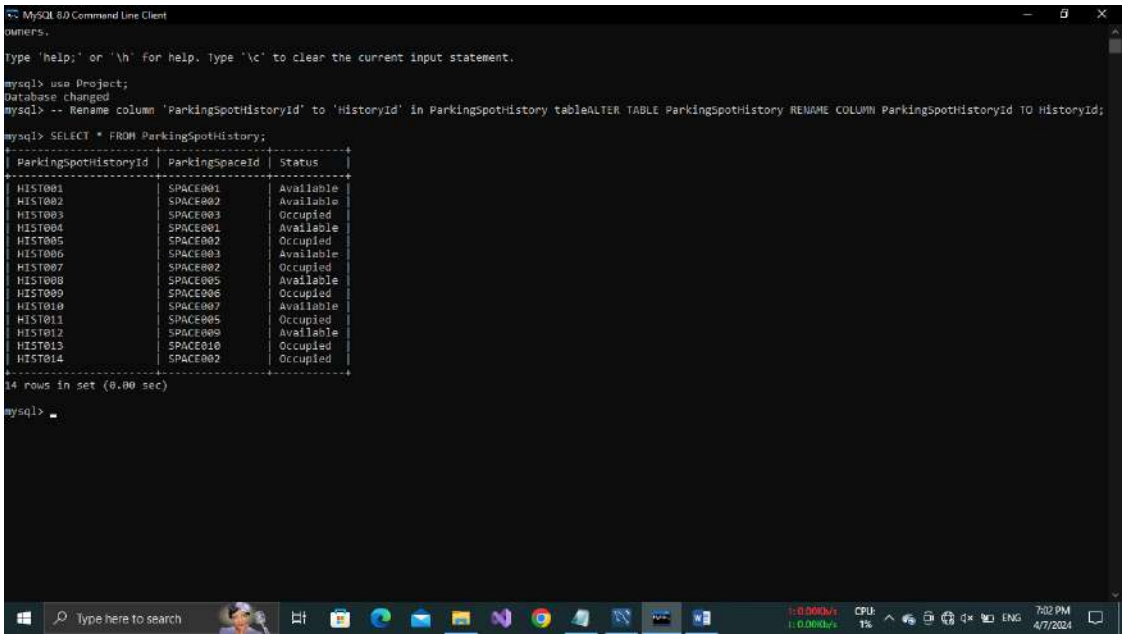
❖ Cartesian product 2



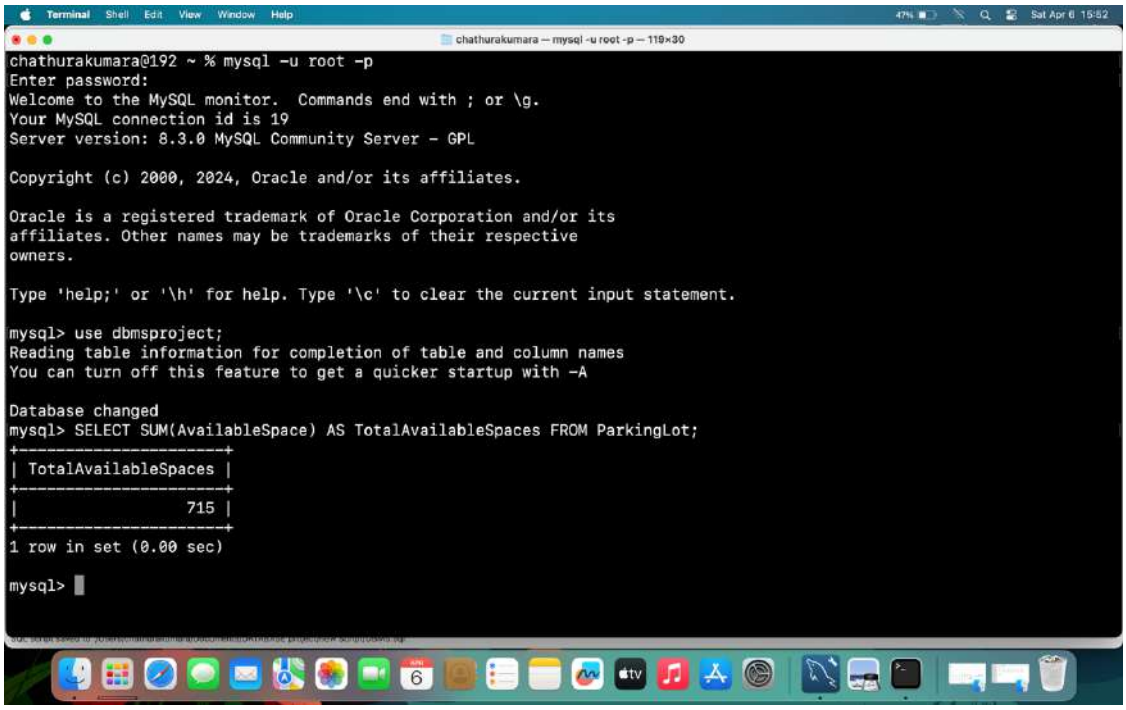
❖ Creating a user view



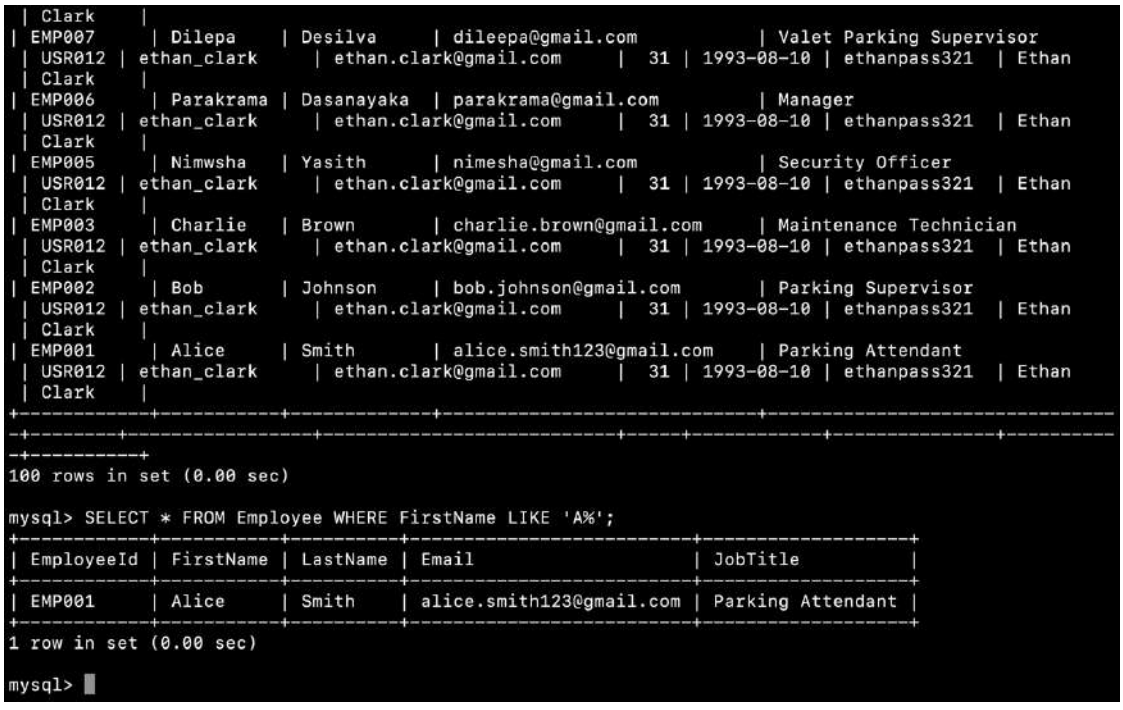
❖ Renaming Operation



❖ Aggregation (Sum function)

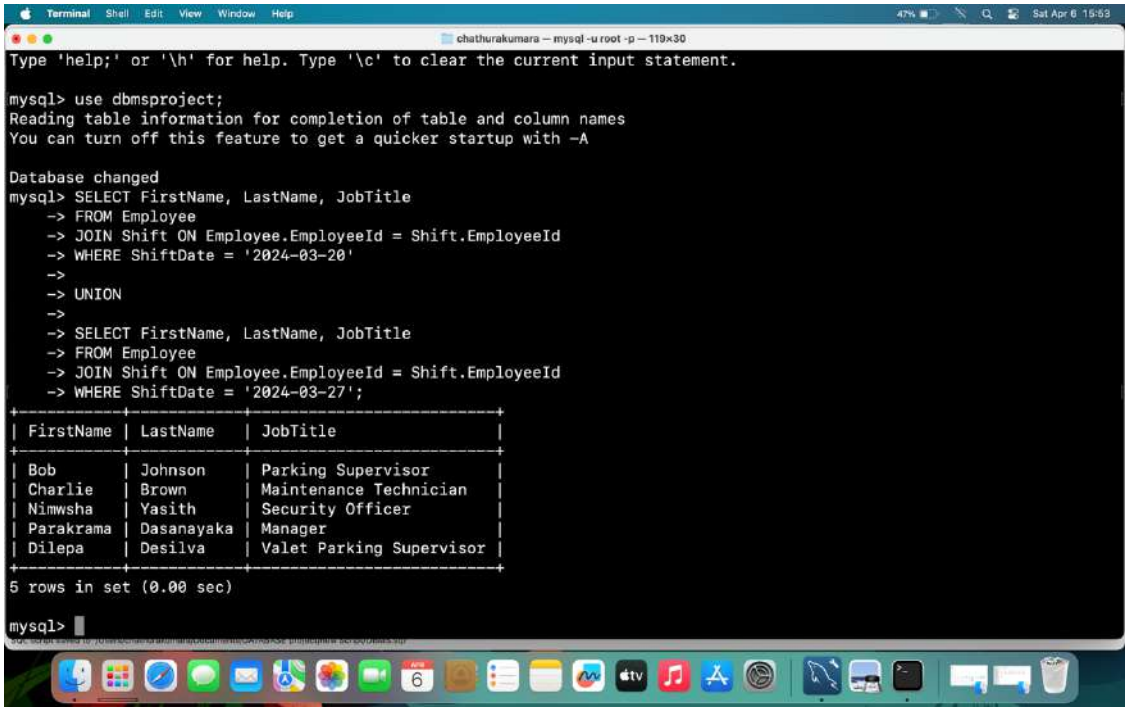


❖ Like operator 1



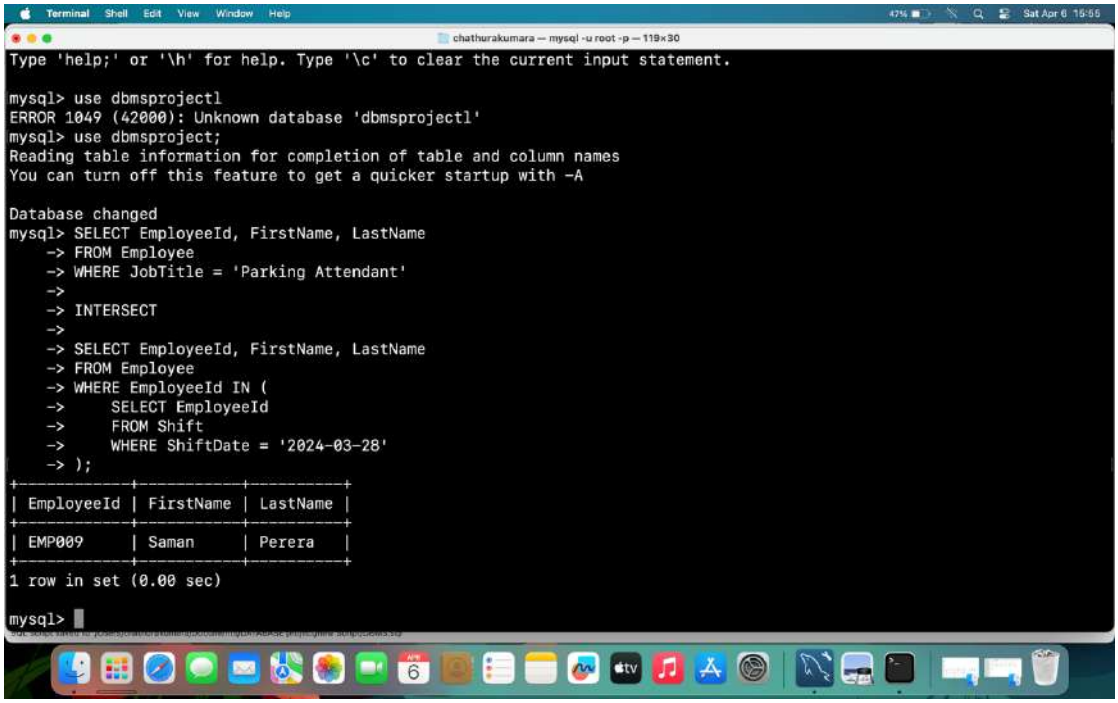
Complex Queries

- Union



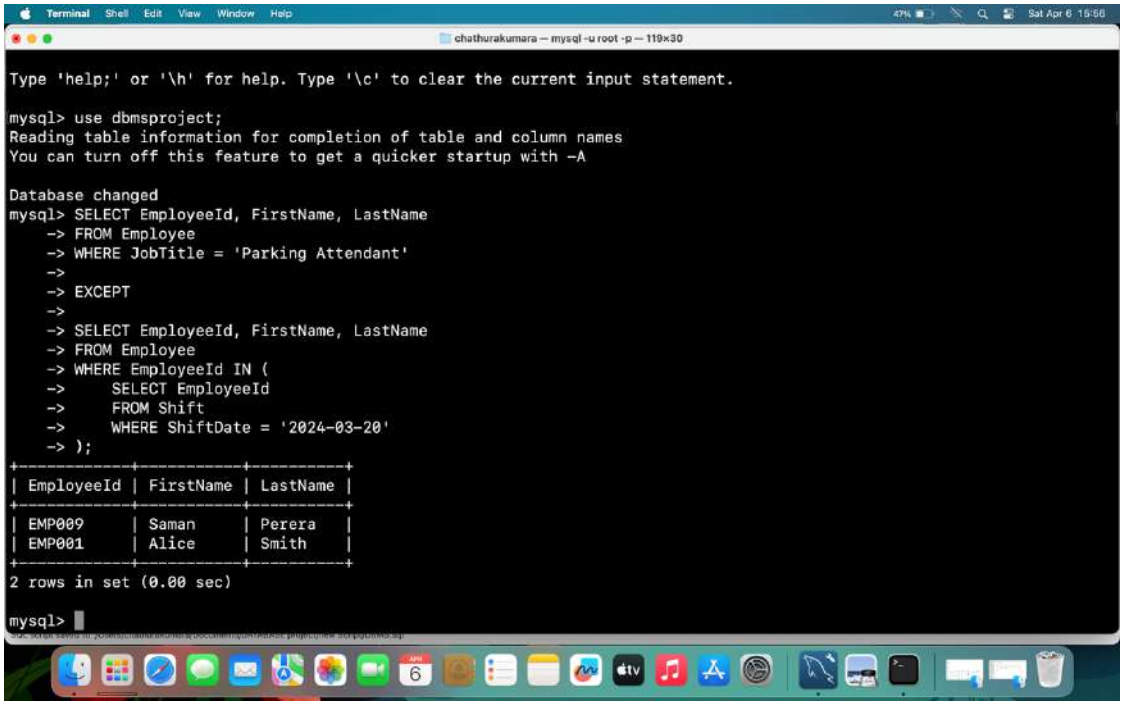
```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

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

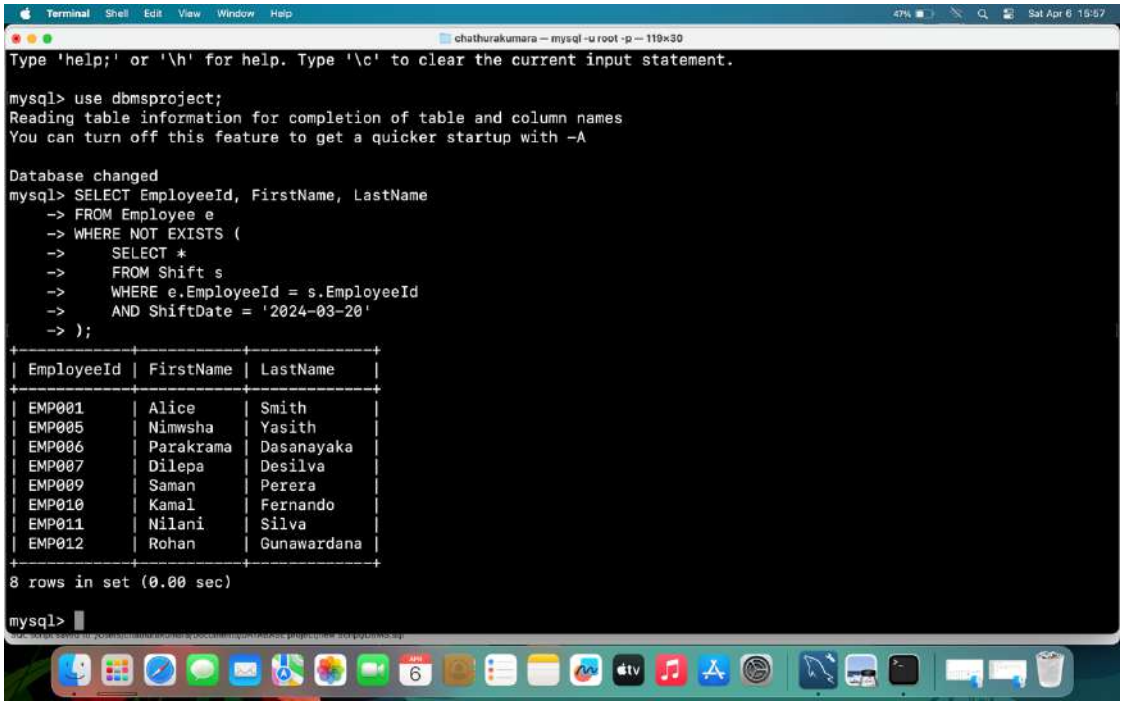


```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use dbmsprojectl
ERROR 1049 (42000): Unknown database 'dbmsprojectl'
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>
```

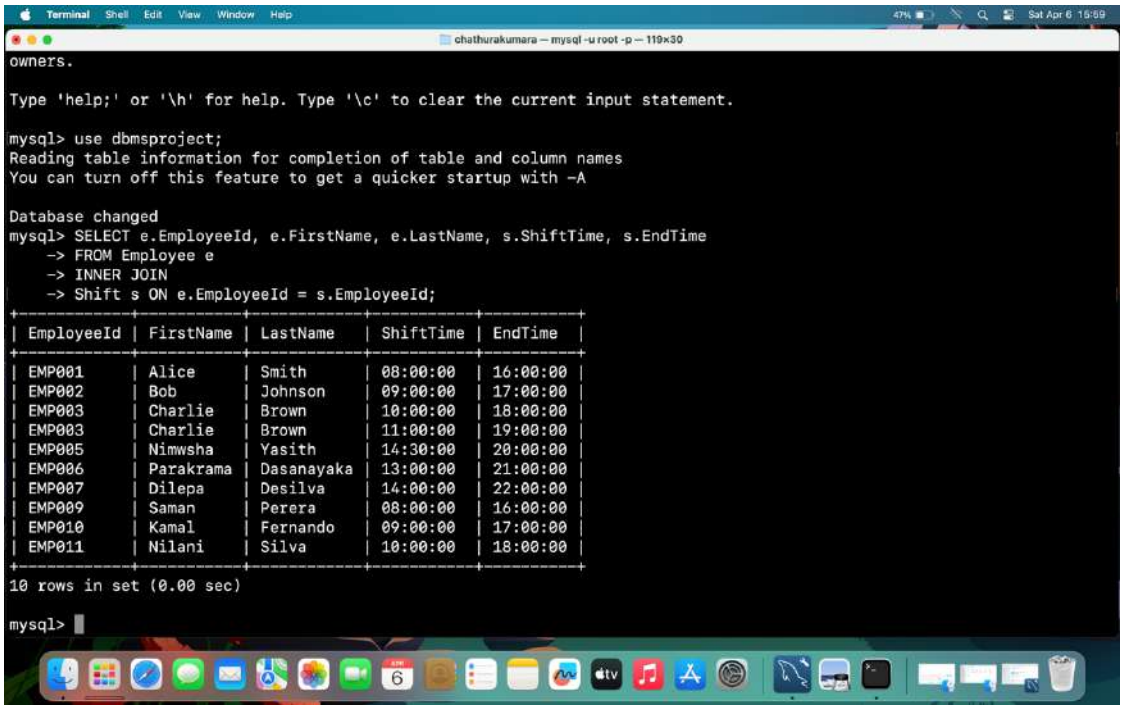
❖ Set Difference



❖ Division

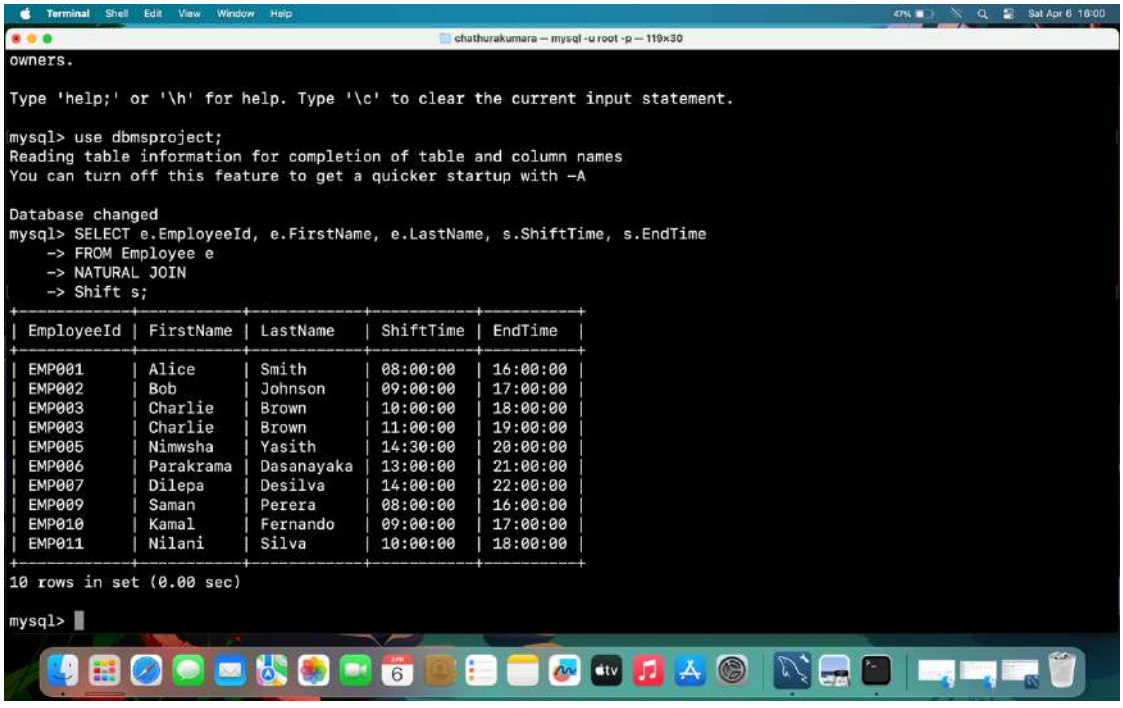


❖ Inner join



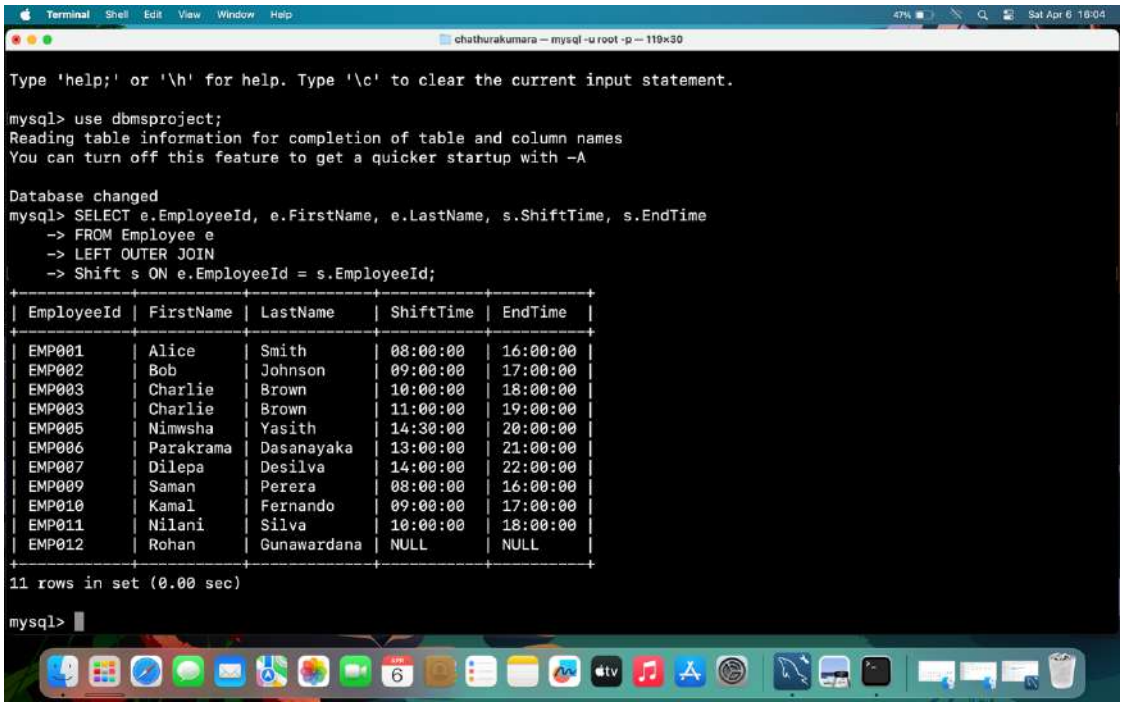
```
owners.

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



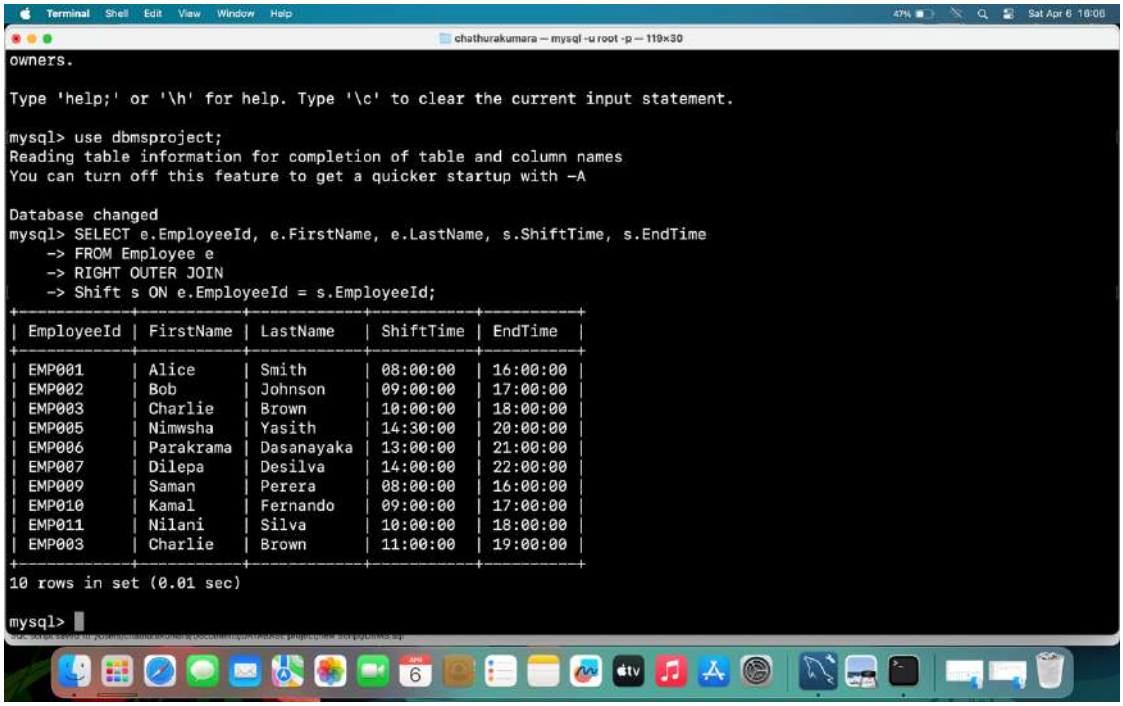
```
owners.

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



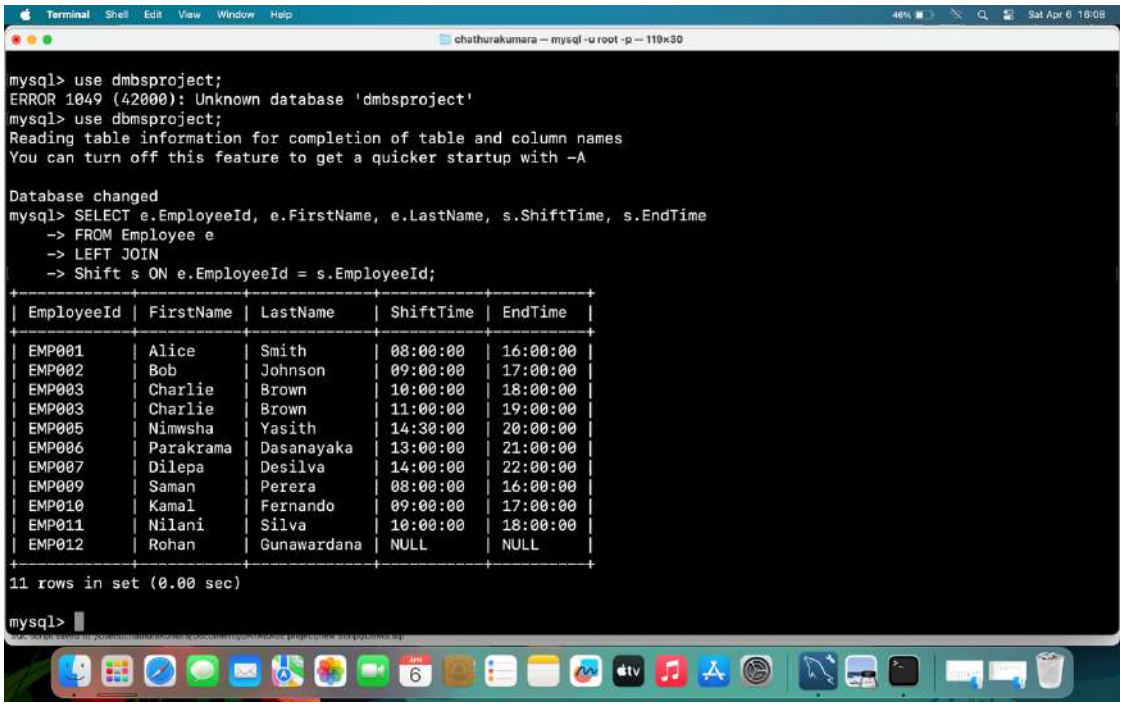
```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

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



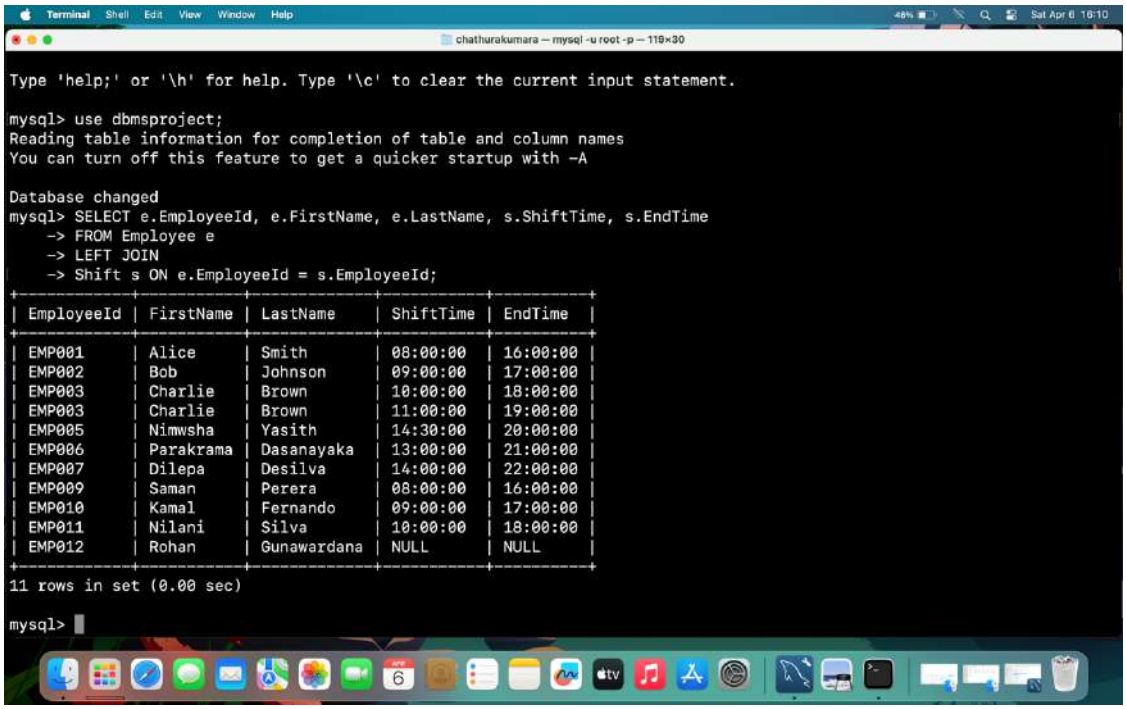
```
owners.

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



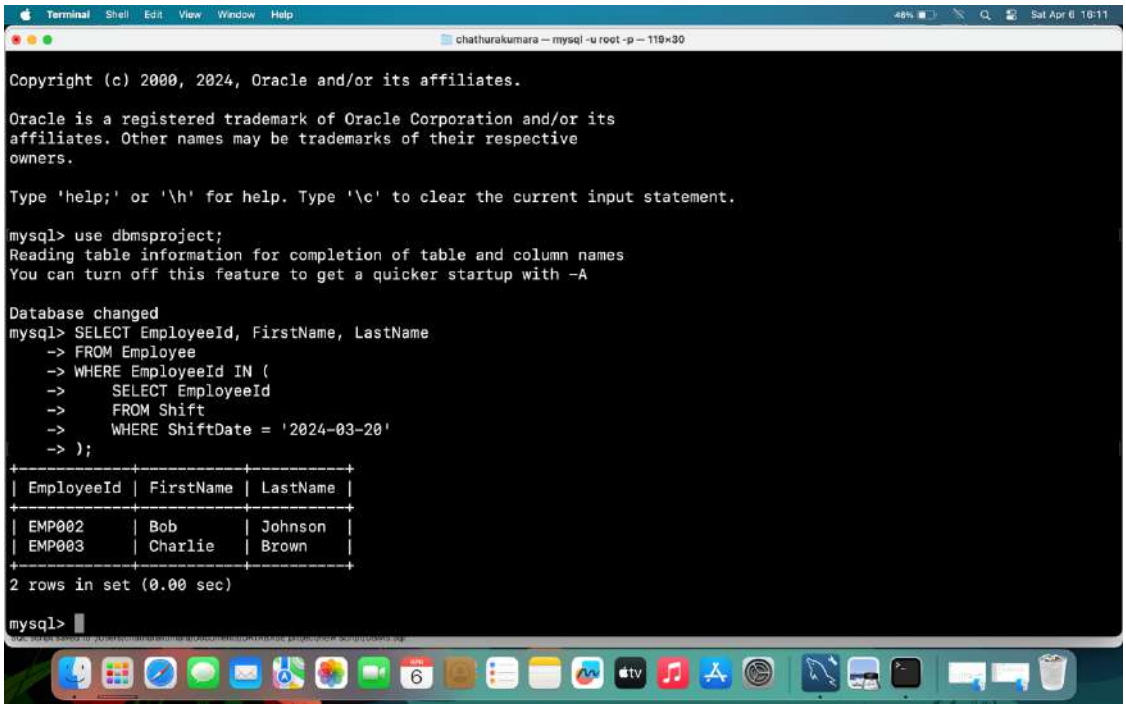
```
mysql> use dmbsproject;
ERROR 1049 (42000): Unknown database 'dmbsproject'
mysql> use dbmsproject;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

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

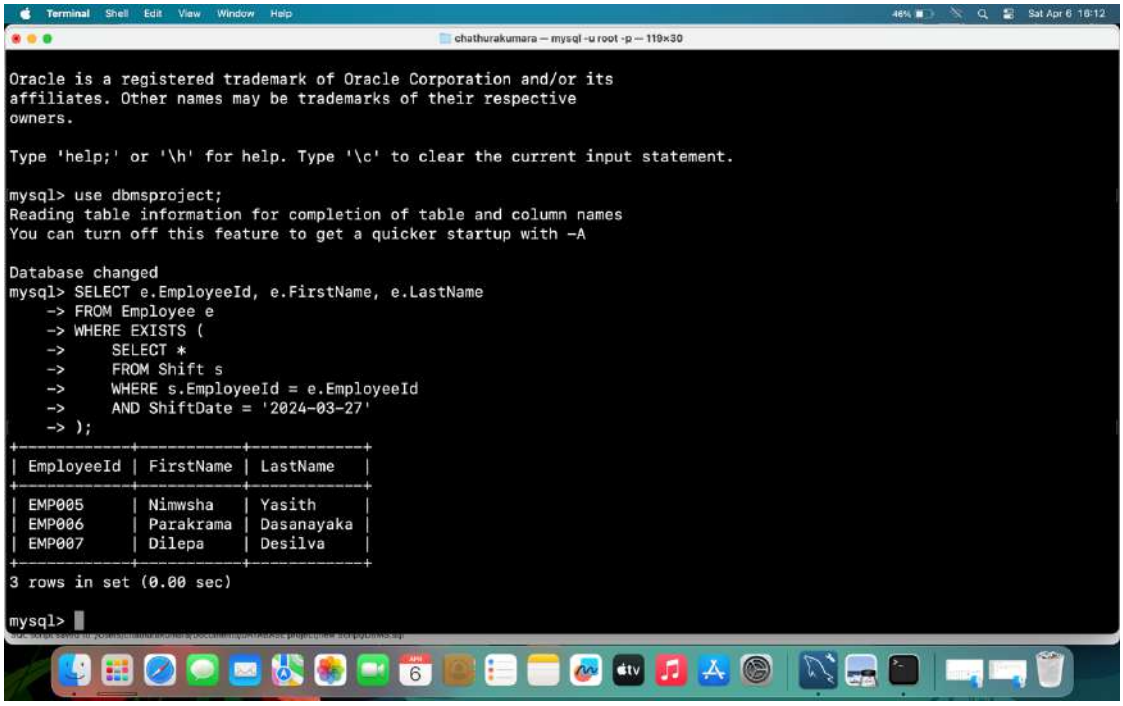


```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

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



❖ Correlated Nested query.

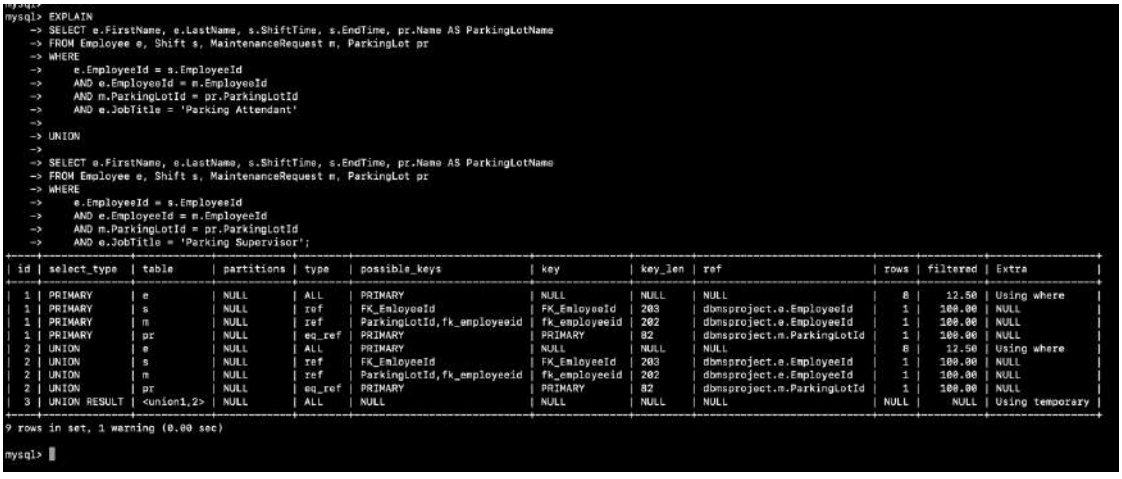


Chapter 5: Tuning

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:

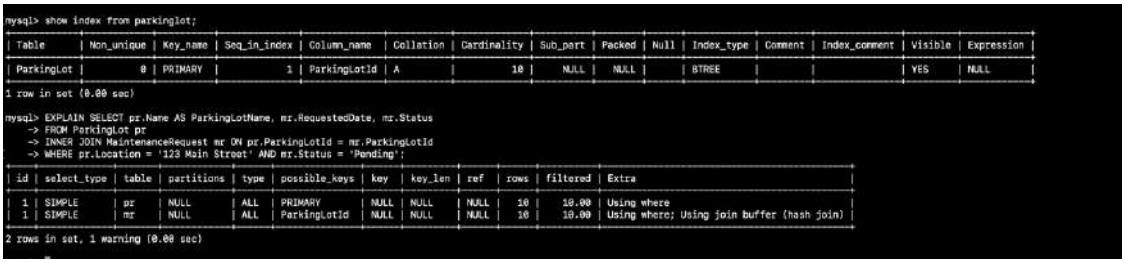
- Remove any externally built indexes that are currently present in utilized tables. o
- After creating a proper index, display the number of accessed rows. (using EXPLAIN command)
- Display the number of rows accessed before creating a suitable index. (using EXPLAIN command)
- Display table indexes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

- Union tuning



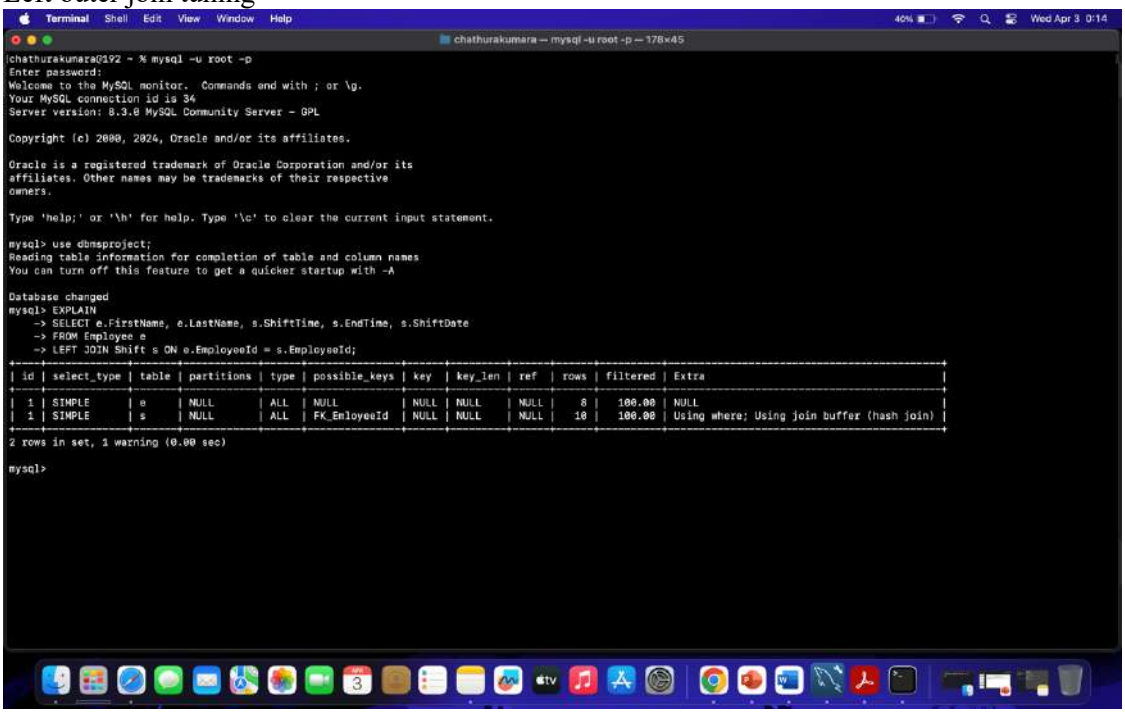
```
mysql> EXPLAIN
    -> SELECT e.FirstName, e.LastName, s.ShiftTime, s.EndTime, pr.Name AS ParkingLotName
    -> FROM Employee e, Shift s, MaintenanceRequest m, ParkingLot pr
    -> WHERE
    ->     e.EmployeeId = s.EmployeeId
    ->     AND e.EmployeeId = m.EmployeeId
    ->     AND m.ParkingLotId = 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 |    12.50 | 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_employeeid | fk_employeeid | 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 |    12.50 | Using where     |
|  2 | UNION        | s            | NULL       | ref    | FK_EmployeeId         | FK_EmployeeId   | 203     | dbmsproject.e.EmployeeId      |    1 |   100.00 | NULL            |
|  2 | UNION        | m            | NULL       | ref    | ParkingLotId,fk_employeeid | fk_employeeid | 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



```
mysql> show index from parkinglot;
+------------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Table      | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+------------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| ParkingLot |          0 | PRIMARY  |            1 | ParkingLotId | A         |          10 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
+------------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
1 row in set (0.00 sec)

mysql> EXPLAIN SELECT pr.Name AS ParkingLotName, mr.RequestedDate, mr.Status
    -> FROM ParkingLot pr
    -> INNER JOIN MaintenanceRequest mr ON pr.ParkingLotId = mr.ParkingLotId
    -> WHERE pr.Location = '123 Main Street' AND mr.Status = 'Pending';
+----+-------------+-------+------------+------+---------------+------+---------+------+------+----------+-----------------------------------------------------+
| id | select_type | table | partitions | type | possible_keys | key  | key_len | ref  | rows | filtered | Extra                                               |
+----+-------------+-------+------------+------+---------------+------+---------+------+------+----------+-----------------------------------------------------+
|  1 | SIMPLE      | pr    | NULL       | ALL  | PRIMARY       | NULL | NULL    | NULL |   10 |    10.00 | Using where                                         |
|  1 | SIMPLE      | mr    | NULL       | ALL  | ParkingLotId  | NULL | NULL    | NULL |   10 |    10.00 | Using where; Using join buffer (hash join)          |
+----+-------------+-------+------------+------+---------------+------+---------+------+------+----------+-----------------------------------------------------+
2 rows in set, 1 warning (0.00 sec)
```

- Left outer join tuning

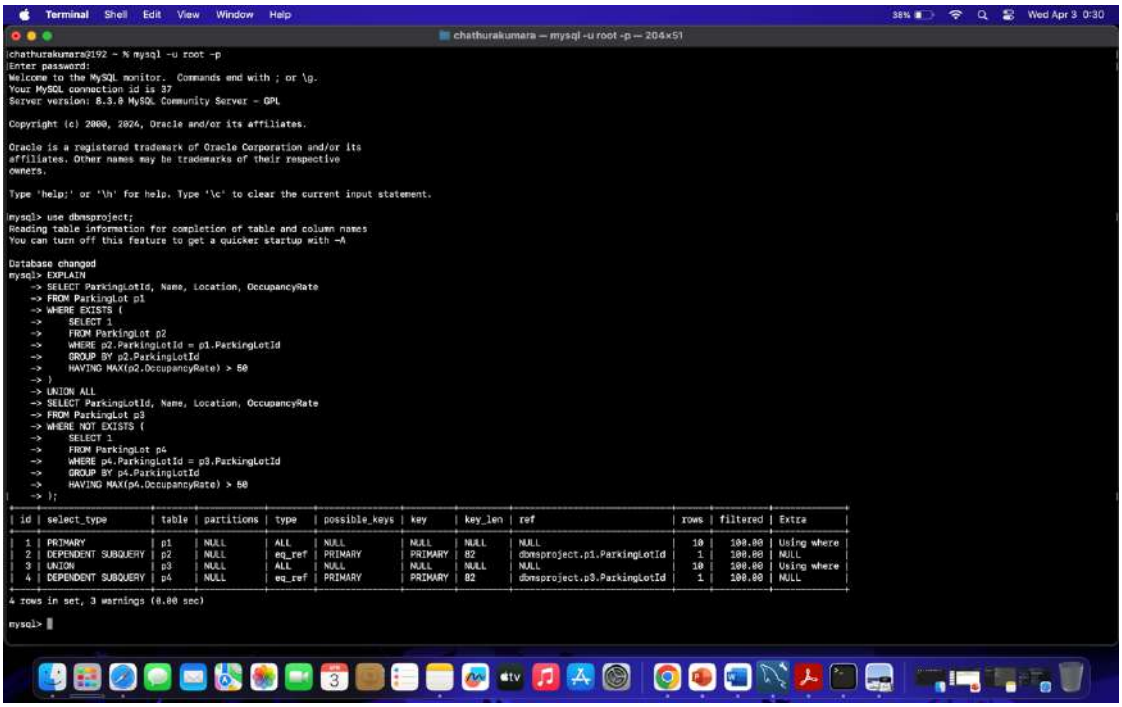


```
Terminal  Shell  Edit  View  Window  Help                                   40%  Wed Apr 3  0:14
chathurakumara — mysql -u root -p — 178×45

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

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

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

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

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

Database changed
mysql> EXPLAIN
    -> SELECT e.FirstName, e.LastName, s.ShiftTime, s.EndTime, s.ShiftDate
    -> FROM Employee e
    -> LEFT JOIN Shift s ON e.EmployeeId = s.EmployeeId;
+----+-------------+-------+------------+------+---------------+------+---------+------+------+----------+-----------------------------------------------------+
| id | select_type | table | partitions | type | possible_keys | key  | key_len | ref  | rows | filtered | Extra                                               |
+----+-------------+-------+------------+------+---------------+------+---------+------+------+----------+-----------------------------------------------------+
|  1 | SIMPLE      | e     | NULL       | ALL  | NULL          | NULL | NULL    | NULL |    8 |   100.00 | NULL                                                |
|  1 | SIMPLE      | s     | NULL       | ALL  | FK_EmployeeId | NULL | NULL    | NULL |   10 |   100.00 | Using where; Using join buffer (hash join)          |
+----+-------------+-------+------------+------+---------------+------+---------+------+------+----------+-----------------------------------------------------+
2 rows in set, 1 warning (0.00 sec)

mysql>
```

- Right outer join tuning

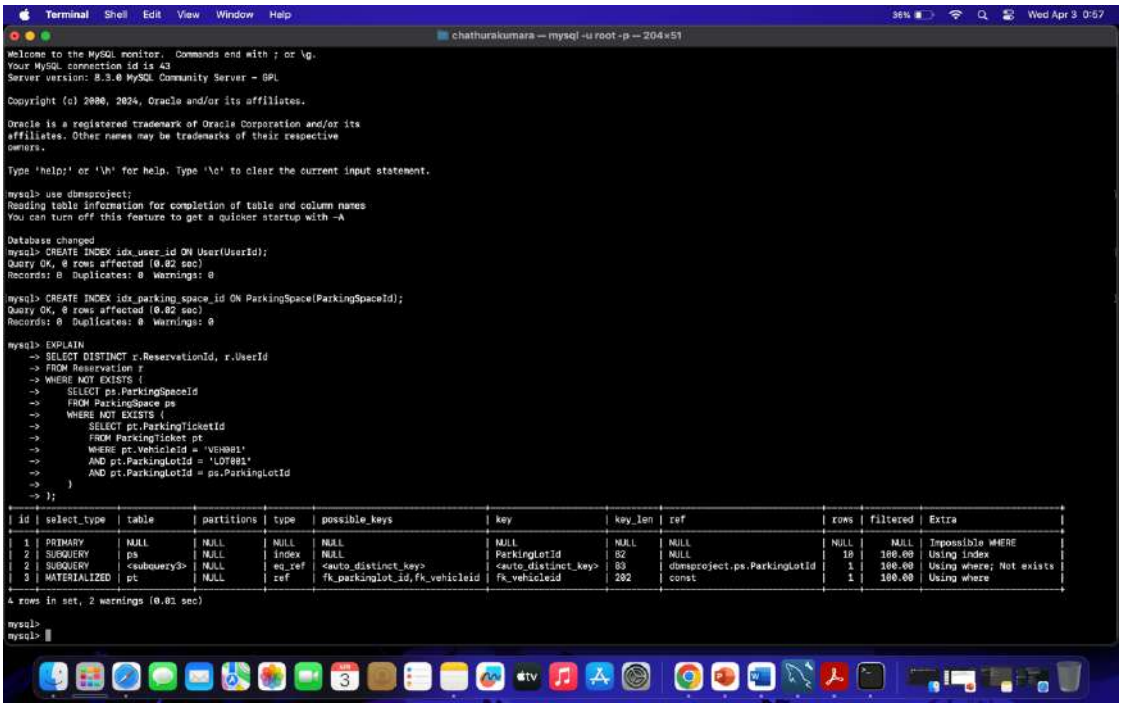
```
Terminal  Shell  Edit  View  Window  Help                                   39%  Wed Apr 3  0:21
chathurakumara — mysql -u root -p — 204×51

Database changed
mysql> CREATE INDEX idx_employee_id ON EmployeePhoneNumber(EmployeeId);
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> CREATE INDEX idx_employee_id ON Employee(EmployeeId);
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> EXPLAIN SELECT e.FirstName, e.LastName, ep.EmployeePhoneNumber
    -> FROM EmployeePhoneNumber ep
    -> RIGHT JOIN Employee e ON e.EmployeeId = ep.EmployeeId;
+----+-------------+-------+------------+------+---------------+----------------+---------+-------------------------+------+----------+-------+
| id | select_type | table | partitions | type | possible_keys | key            | key_len | ref                     | rows | filtered | Extra |
+----+-------------+-------+------------+------+---------------+----------------+---------+-------------------------+------+----------+-------+
|  1 | SIMPLE      | e     | NULL       | ALL  | NULL          | NULL           | NULL    | NULL                    |    8 |   100.00 | NULL  |
|  1 | SIMPLE      | ep    | NULL       | ref  | idx_employee_id | idx_employee_id | 203    | dbmsproject.e.EmployeeId |    1 |   100.00 | NULL  |
+----+-------------+-------+------------+------+---------------+----------------+---------+-------------------------+------+----------+-------+
2 rows in set, 1 warning (0.01 sec)

mysql> EXPLAIN SELECT e.FirstName, e.LastName, ep.EmployeePhoneNumber
    -> FROM EmployeePhoneNumber ep
    -> RIGHT JOIN Employee e ON e.EmployeeId = ep.EmployeeId;
+----+-------------+-------+------------+------+---------------+----------------+---------+-------------------------+------+----------+-------+
| id | select_type | table | partitions | type | possible_keys | key            | key_len | ref                     | rows | filtered | Extra |
+----+-------------+-------+------------+------+---------------+----------------+---------+-------------------------+------+----------+-------+
|  1 | SIMPLE      | e     | NULL       | ALL  | NULL          | NULL           | NULL    | NULL                    |    8 |   100.00 | NULL  |
|  1 | SIMPLE      | ep    | NULL       | ref  | idx_employee_id | idx_employee_id | 203    | dbmsproject.e.EmployeeId |    1 |   100.00 | NULL  |
+----+-------------+-------+------------+------+---------------+----------------+---------+-------------------------+------+----------+-------+
2 rows in set, 1 warning (0.00 sec)

mysql> SHOW INDEX FROM EmployeePhoneNumber;
+---------------------+------------+-----------------+--------------+---------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Table               | Non_unique | Key_name        | Seq_in_index | Column_name   | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+---------------------+------------+-----------------+--------------+---------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| EmployeePhoneNumber |          0 | PRIMARY         |            1 | PhoneId       | A         |          10 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| EmployeePhoneNumber |          1 | idx_employee_id |            1 | EmployeeId    | A         |           9 |     NULL |   NULL | YES  | BTREE      |         |               | YES     | NULL       |
+---------------------+------------+-----------------+--------------+---------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
2 rows in set (0.01 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 (0.01 sec)

mysql>
```

- Outer union tuning



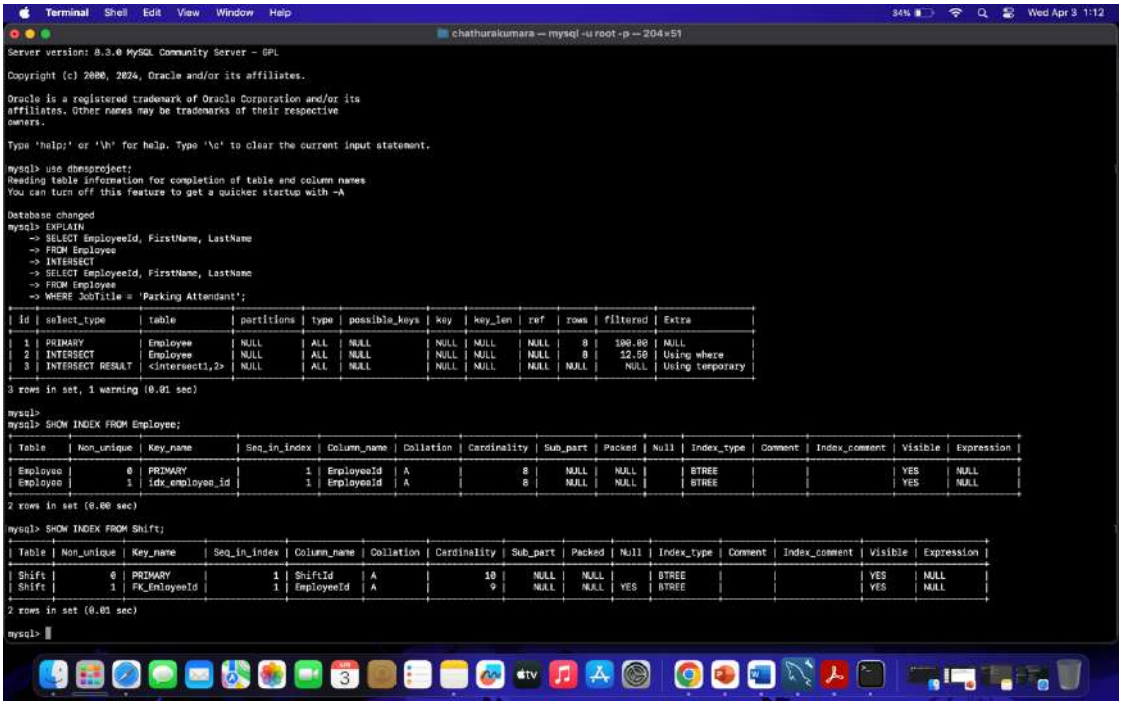
- Division tuning

- Correlated Nested query.



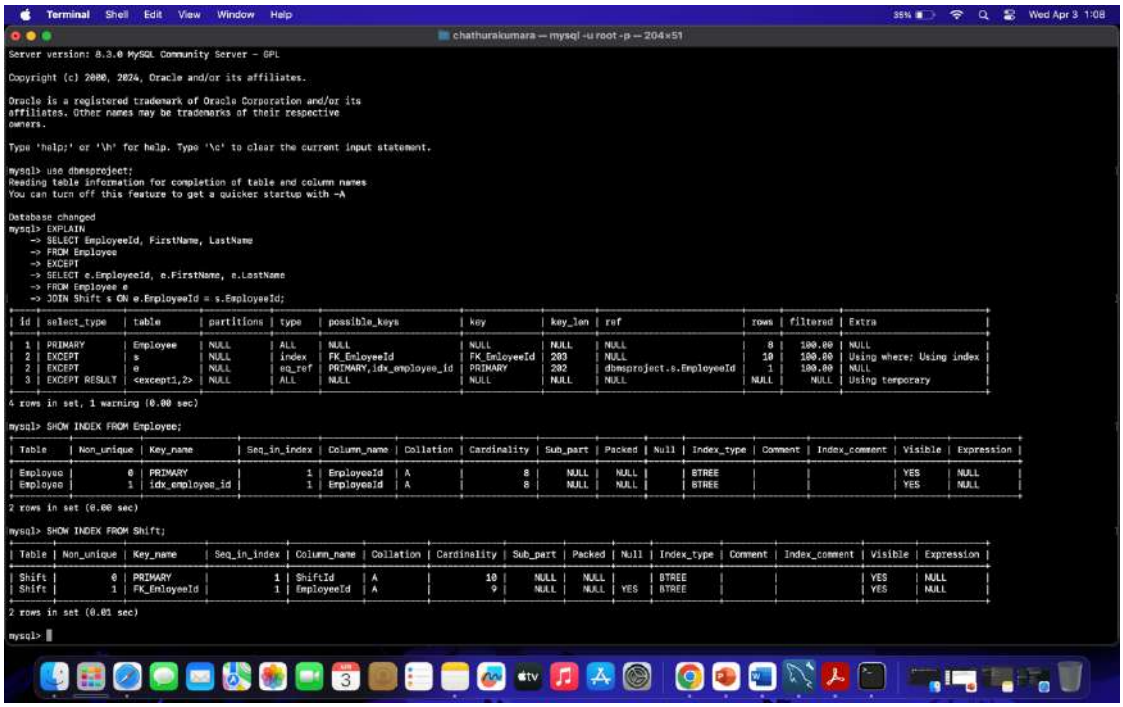
```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 43
Server version: 8.3.0 MySQL Community Server - GPL

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

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

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

mysql> EXPLAIN
    -> SELECT DISTINCT r.ReservationId, r.UserId
    -> FROM Reservation r
    -> WHERE NOT EXISTS (
    ->     SELECT ps.ParkingSpaceId
    ->     FROM ParkingSpace ps
    ->     WHERE NOT EXISTS (
    ->         SELECT pt.ParkingTicketId
    ->         FROM ParkingTicket pt
    ->         WHERE pt.VehicleId = '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         | NULL       | NULL   | NULL                | NULL                | NULL    | NULL                          | NULL |     NULL | Impossible WHERE         |
|  2 | SUBQUERY    | ps           | NULL       | index  | NULL                | ParkingLotId        | 82      | NULL                          |   10 |   100.00 | Using index              |
|  2 | SUBQUERY    | <subquery3>  | 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>
mysql>
```

- Intersection tuning



```
Server version: 8.3.0 MySQL Community Server - GPL

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

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

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

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

Database changed
mysql> EXPLAIN
    -> SELECT EmployeeId, FirstName, LastName
    -> FROM Employee
    -> 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    | NULL |    8 |   100.00 | NULL            |
|  2 | INTERSECT    | Employee      | NULL       | ALL  | NULL          | NULL | NULL    | NULL |    8 |    12.50 | 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 | 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>
```

- Set difference tuning.



- Independent nested query Tuning.