



AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH (AIUB)

DEPARTMENT OF ENGINEERING

INTRODUCTION TO DATABASE

Section: N, Group: 7

Project Name

PARKING MANAGEMENT SYSTEM

Supervised By

RIFAT IBN ALAM

Submitted By

Name	ID
1. SAZID-AL-ABEDIN	22-45999-1
2. SADMAN SAMIR RAFITH	22-46018-1
3. MOHAMMAD ARIYAN PATHAN	22-46011-1
4. MD. SADMAN HOSSAIN	22-46061-1

Date of Submission: December 26, 2022

TABLE OF CONTENT

Topics	Page No
I Title Page	1
II Table of content	2
1. Scenario	3-4
2. ER Diagram	5
3. Normalization	6-16
4. Finalization	17
5. Table Creation/sequence	18-35
6. Data Insertion	36-52
7. Query	53-57
8. View	58-59

ER Diagram Case Study:

In the present day. A vehicle is become a must, As Urbanization is an ongoing process everywhere. People nowadays like to go to a lot of recreational areas for relaxation. Therefore, each of these locations requires a parking area where visitors may park their cars conveniently and safely. A system that keeps track of vehicle information is necessary for every parking lot to provide the facility. A system that automatically delivers data processing at a high rate of speed in a structured manner is a vehicle parking management system.

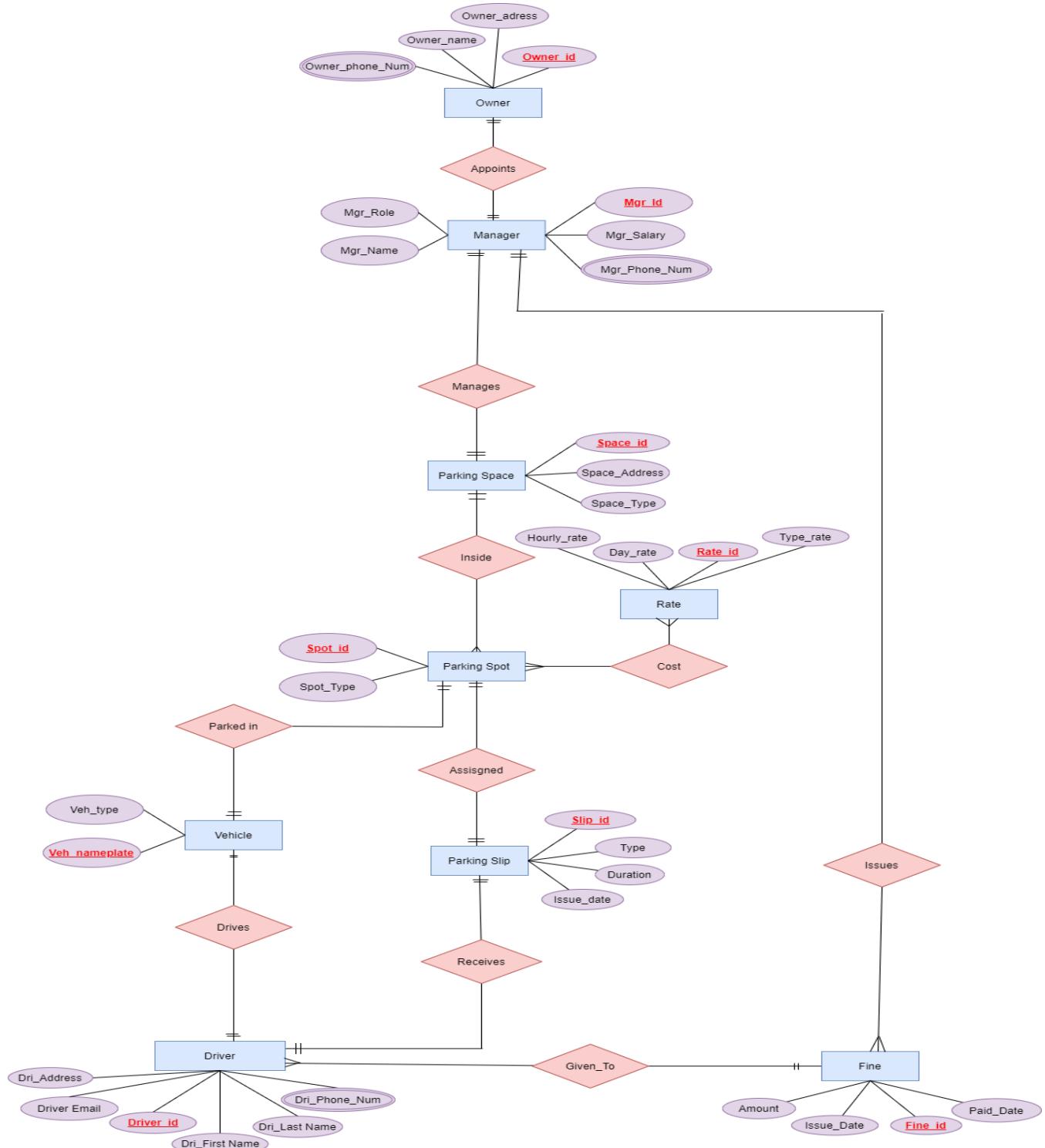
We cordially welcome you to utilize our parking management system from this location. So, at first, there is a parking owner who has their own owner id, and address, not only that he also has his name, and can have several phone numbers. As a result, one owner may appoint one manager to oversee everything. The manager also has a job to perform; he has his name and has a manager's unique ID, not only that he also has a role to do, but along with work he also gets a salary, not only The manager also has various phone numbers.

In the parking management system Managers also manages parking spaces, and each manager can manage one parking space at a time. Parking spaces have a space type as far as requirements. it has a unique space id, and, finally, an address. There is all so parking spots inside the parking space and one parking space can have many parking spots. The parking spot itself has a unique spot id and address. Not only Parking spots can also be rented at various rates such as hourly rates and daily rates, The rates have types and a unique rate id.

Vehicles are parked inside parking spaces, and each parking space may accommodate one car. Other entities are also available for vehicles, such as vehicle type and nameplate. Therefore, Vehicles are driven by drivers, and one driver can drive many vehicles. Each driver has their own email address, not only that they have their first and last name as well. Nevertheless, they also have a unique driver ID, driver address, and phone number. Nonetheles, each parking spot is assigned a parking slip, which the driver receives. Therefore, one parking slip can

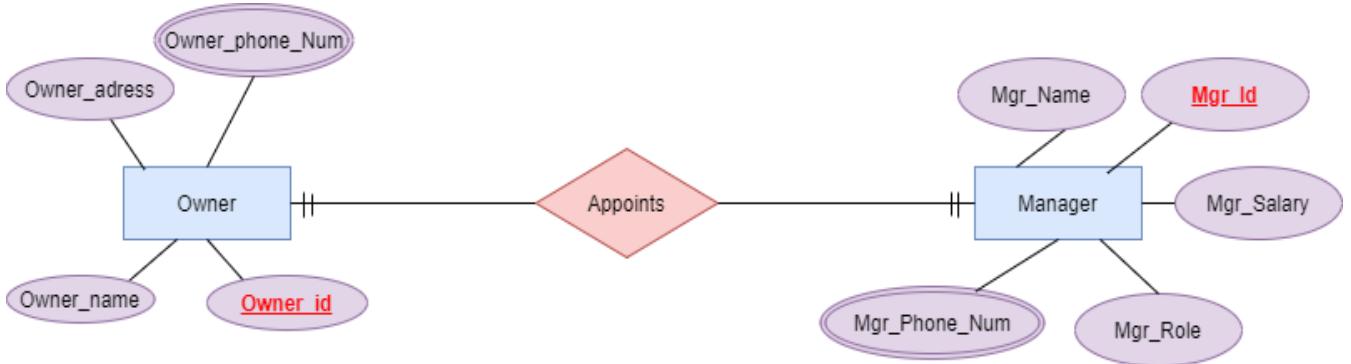
be received by one driver. Lastly, the parking slip has a duration for the drivers, not only that it has its parking type and issue date and lastly it has its own unique id. Last but not least, there are Drivers who get fines, and many drivers may get one or none. Fines have an amount to be paid, it also has an issuing date and a payment rate, and lastly, it has a unique fine id. Therefore, Managers impose penalties, and one manager can issue many fines.

ER Diagram



Normalization

Owner1.... Appoints1....Manager



UNF:

Appoints (Owner_Phone Num, Owner_Address, Owner_ID, Owner_Name, Mgr_name, Mgr_role, Mgr_ID, Salary, Mgr_phone Num)

1NF:

1st: Owner_phone_Num, Mgr_phone Num both are Multivalue Attribute

2nd: Owner_Phone Num, Owner_Address, Owner_ID, Owner_Name, Mgr_name, Mgr_role, Mgr_ID, Salary, Mgr_phone Num

2NF:

1st: Owner_Phone Num, Owner Name, Owner_Address, Owner_ID, Mgr_ID

2nd: Mgr_ID, Mgr_Name, Mgr_Role, Salary, Mgr_phone Num

3NF:

1st: NO TRANSITIVE ATTRIBUTE

2nd : Owner_Phone Num, Owner Name, Owner_Address, Owner_ID, Mgr_ID

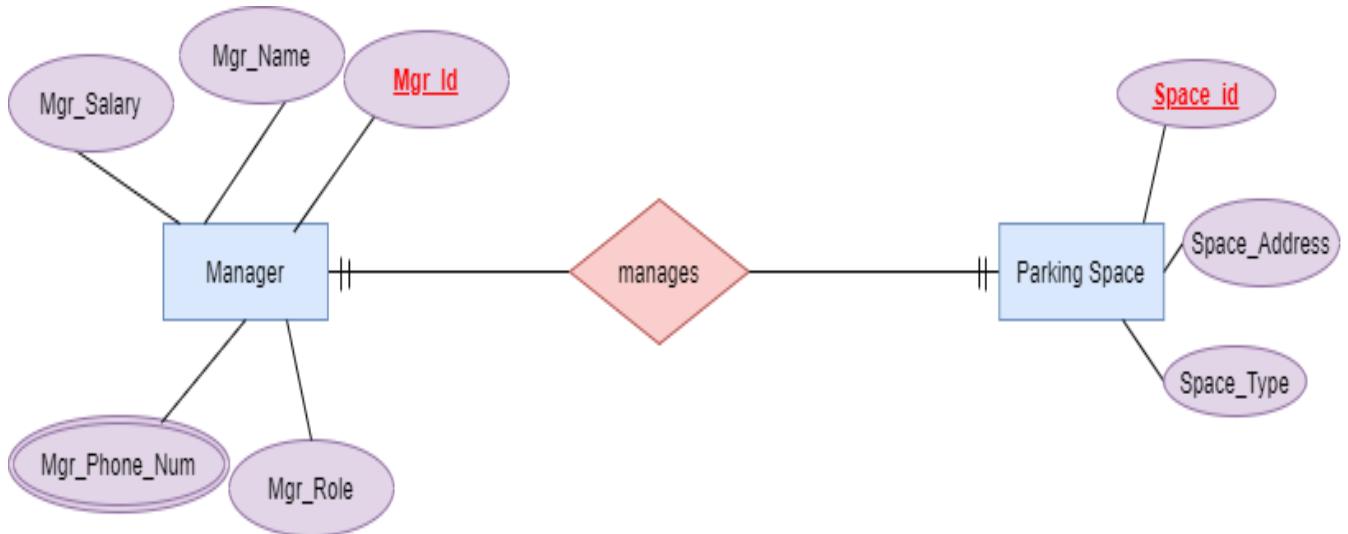
3rd : Mgr_ID, Mgr_Name, Mgr_Role, Salary, Mgr_phone Num

Final Table:

1st: Owner_Phone Num, Owner Name, Owner_Address, Owner_ID, Mgr_ID

2nd : Mgr_ID, Mgr_Name, Mgr_Role, Salary, Mgr_phone Num

Manager1.... Manages1..... Parking Space



UNF:

Manages(Mgr_ID, Mgr_Name, Mgr_Role, Mgr_Salary, Mgr_Phone_Num, Space_ID, Space_Address, Space_Type)

1NF:

1st: Mgr_Phone_Num is a multivalued attribute

2nd: Mgr_ID, Mgr_Name, Mgr_Role, Mgr_Salary, Mgr_Phone_Num, Space_ID, Space_Address, Space_Type

2NF:

1st: Mgr_ID, Mgr_Name, Mgr_Role, Mgr_Salary, Mgr_Phone_Num, Space_ID

2nd: Space_ID, Space_Address, Space_Type

3NF:

1st: Mgr_ID, Mgr_Name, Mgr_Role, Mgr_Salary, Mgr_Phone_Num, Space_ID

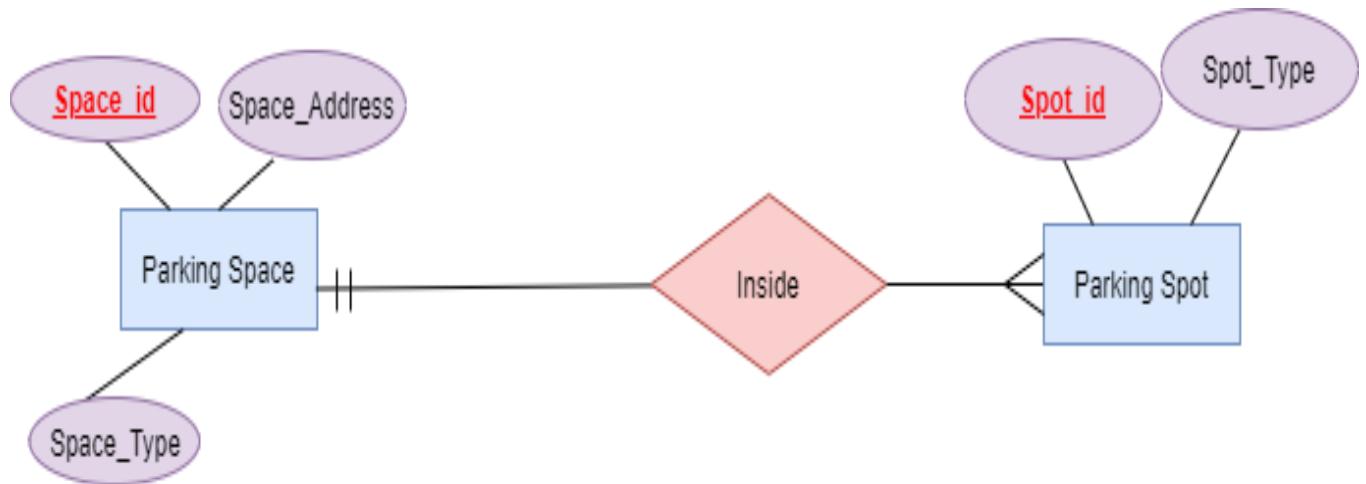
2nd: Space_ID, Space_Address, Space_Type

Final Table:

1st: Mgr_ID, Mgr_Name, Mgr_Role, Mgr_Salary, Mgr_Phone_Num, Space_ID

2nd: Space_ID, Space_Address, Space_Type

Parking Space....1....Inside* Parking Spot



UNF:

Inside (Space_ID, Space_Address, Space_Type, Spot_ID, Spot_Type)

1NF:

1st: No Multivalued Attribute

2nd : Space_ID, Space_Address, Space_Type, Spot_ID, Spot_Type

2NF:

1st: Space_ID, Space_Address, Space_Type

2nd: Spot_ID, Spot_Type, Space_ID

3NF:

1st: Space_ID, Space_Address, Space_Type

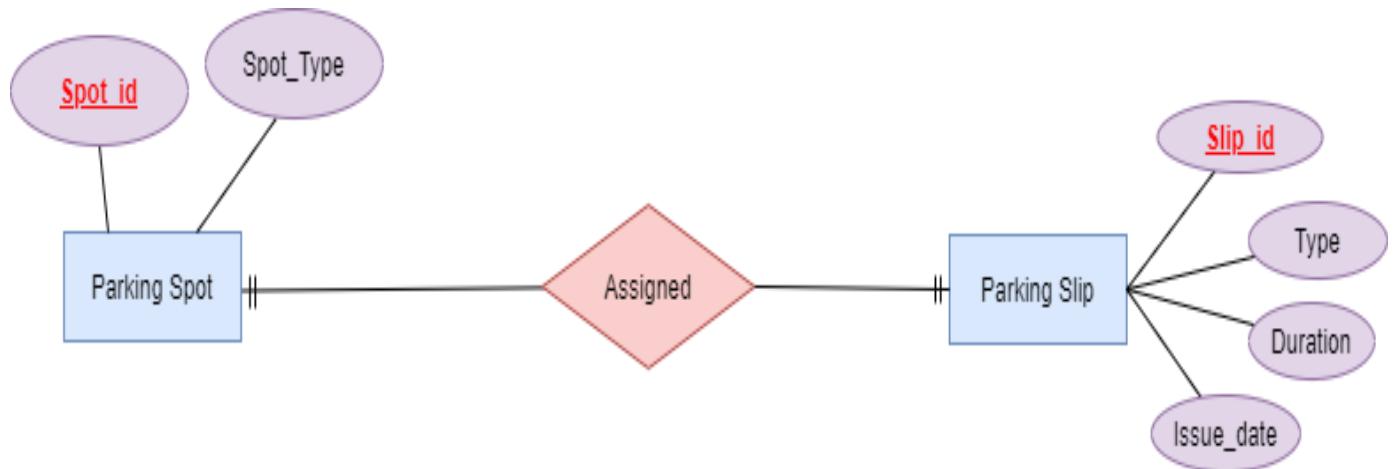
2nd: Spot_ID, Spot_Type, Space_ID

Final Table:

1st: Space_ID, Space_Address, Space_Type

2nd: Spot_ID, Spot_Type, Space_ID

Parking Spot....1.... Assigned1..... Parking Slip



UNF:

Assigned (Spot_ID, Spot_Type, Slip_ID, Slip_Type, Duration, Issue_Date)

1NF:

1st: No Multivalue Attribute

2nd: Spot_ID, Spot_Type, Slip_ID, Slip_Type, Duration, Issue_Date

2NF:

1st: Spot_ID, Spot_Type, Slip_ID

2nd: Slip_ID, Slip_Type, Duration, Issue_Date

3NF:

1st: No Transitive Attribute

2nd: Spot_ID, Spot_Type, Slip_ID

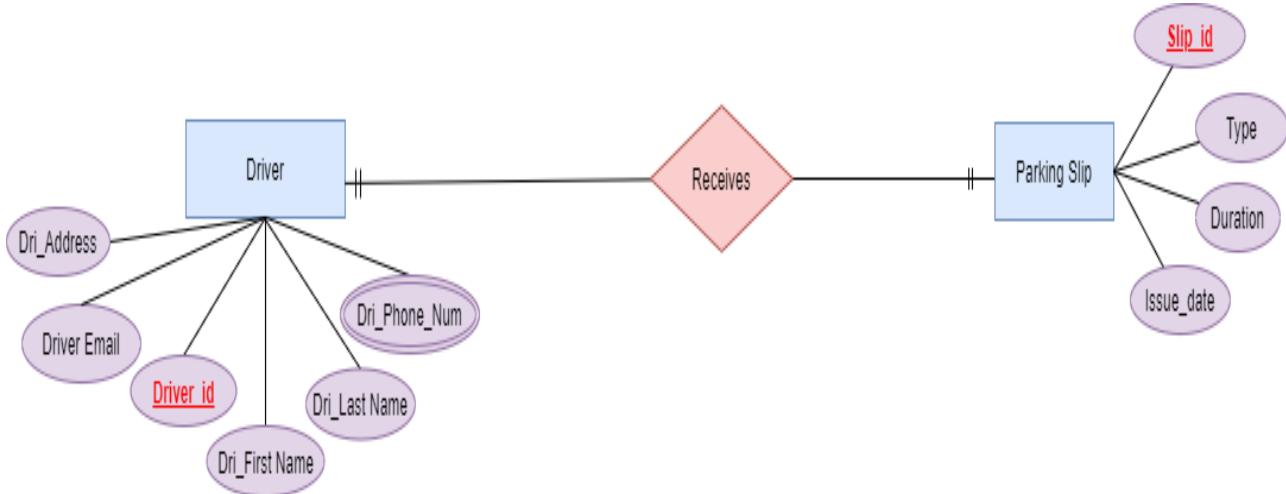
3rd: Slip_ID, Slip_Type, Duration, Issue_Date

Final Table:

1st: Spot_ID, Spot_Type, Slip_ID

2nd: Slip_ID, Slip_Type, Duration, Issue_Date

Parking Slip1.... Receive1..... Driver



UNF:

Receives (Slip_ID, Slip_Type, Duration, Issue_Date, Dri_First Name, Dri_Last Name, Dri_Phone Num, Dri_ID, Dri_Address)

1NF:

1st: Dri_Phone Num is a Multivalued Attribute

2nd: Slip_ID, Slip_Type, Duration, Issue_Date, Dri_First Name, Dri_Last Name, Dri_Phone Num, Dri_ID, Dri_Address

2NF:

1st: Slip_ID, Slip_Type, Duration, Issue_Date, Dri_ID,

2nd: Dri_ID, Dri_First Name, Dri_Last Name, Dri_Phone Num, Dri_Address

3NF:

1st: NO Transitive Attribute

2nd : Slip_ID, Slip_Type, Duration, Issue_Date, Dri_ID

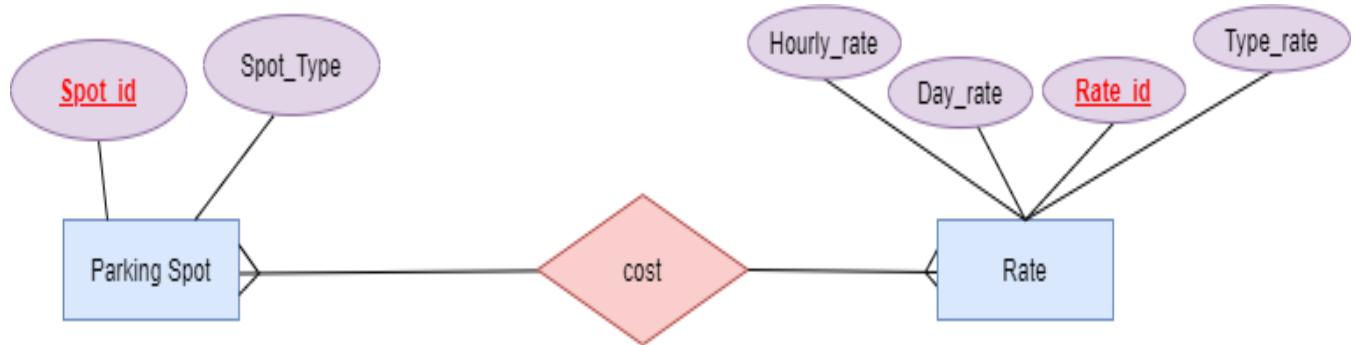
3rd : Dri_ID, Dri_First Name, Dri_Last Name, Dri_Phone Num, Dri_Address

Final Table:

1st: Slip_ID, Slip_Type, Duration, Issue_Date, Dri_ID

2nd: Dri_ID, Dri_First Name, Dri_Last Name, Dri_Phone Num, Dri_Address

Parking Spot....* Cost* Rate



UNF:

Cost (Spot_ID, Spot_Type, Rate_type, Hourly_Rate, Day_Rate, Rate_ID)

1NF:

1st: No Multivalued Attribute

2nd: Spot_ID, Spot_Type, Rate_type, Hourly_Rate, Day_Rate, Rate_ID

2NF:

1st: Spot_ID, Spot_Type

2nd: Rate_type, Hourly_Rate, Day_Rate, Rate_ID

3rd: Spot_ID, Rate_ID

3NF:

1st: No Transitive Value

2nd: Spot_ID, Spot_Type

3rd: Rate_type, Hourly_Rate, Day_Rate, Rate_ID

4th: Spot_ID, Rate_ID

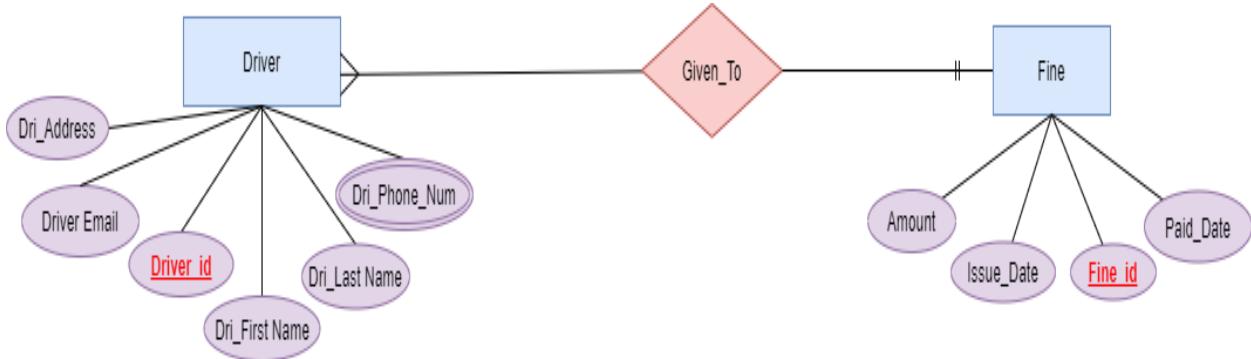
Final Table:

1st: Spot_ID, Spot_Type

2nd: Rate_type, Hourly_Rate, Day_Rate, Rate_ID

3rd: Spot_ID, Rate_ID

Driver....* Given To1..... Fine



UNF:

Given_To (Dri_First Name, Dri_Last Name, Dri_Phone Num, Dri_ID,
Dri_Address, Fine_ID, Amount, Issue_Date, Paid_Date)

1NF:

1st: Dri_Phone Num is a Multivalued Attribute

2nd: Dri_First Name, Dri_Last Name, Dri_Phone Num, Dri_ID, Dri_Address,
Fine_ID, Amount, Issue_Date, Paid_Date

2NF:

1st: Dri_First Name, Dri_Last Name, Dri_Phone Num, Dri_ID, Dri_Address,
Fine_ID

2nd: Fine_ID, Amount, Issue_Date, Paid_Date

3NF:

1st: NO Transitive Attribute

2nd: Dri_First Name, Dri_Last Name, Dri_Phone Num, Dri_ID, Dri_Address,
Fine_ID

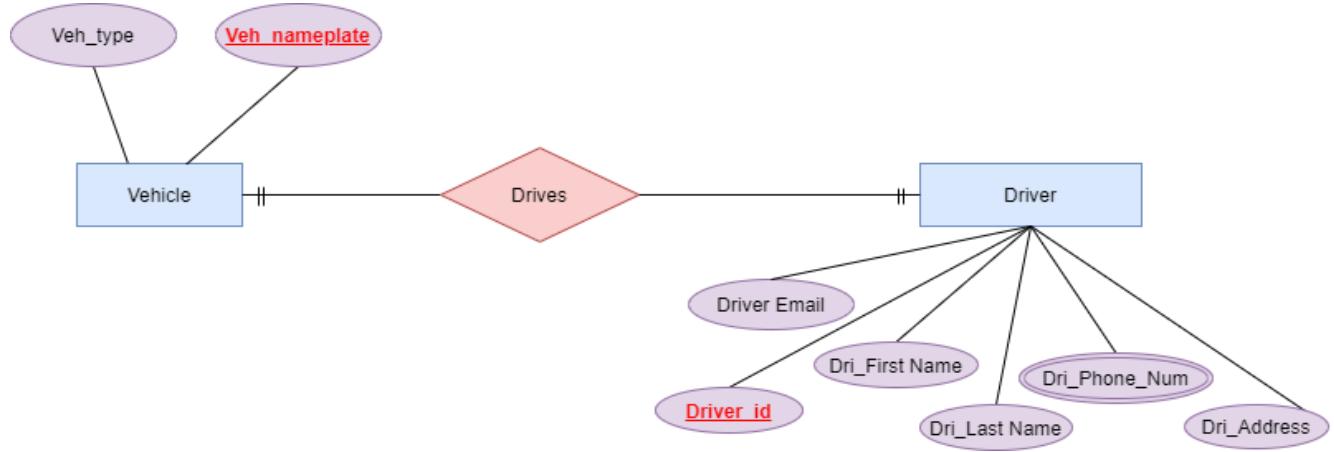
3rd: Fine_ID, Amount, Issue_Date, Paid_Date

Final Table:

1st: Dri_First Name, Dri_Last Name, Dri_Phone Num, Dri_ID, Dri_Address,
Fine_ID

2nd: Fine_ID, Amount, Issue_Date, Paid_Date

Driver....1.... Drives1..... Vehicle



UNF:

Drives (Dri_First_Name, Dri_Last_Name, Dri_Phone_Num, Dri_ID, Dri_Address, Veh_Nameplate, Veh_Type)

1NF:

1st: Dri_Phone_Num is a Multivalued Attribute

2nd: Dri_First_Name, Dri_Last_Name, Dri_Phone_Num, Dri_ID, Dri_Address, Veh_Nameplate, Veh_Type

2NF:

1st: Dri_First_Name, Dri_Last_Name, Dri_Phone_Num, Dri_ID, Dri_Address, Veh_Nameplate

2nd: Veh_Nameplate, Veh_Type

3NF:

1st: NO transitive Attribute

2nd: Dri_First_Name, Dri_Last_Name, Dri_Phone_Num, Dri_ID, Dri_Address, Veh_Nameplate

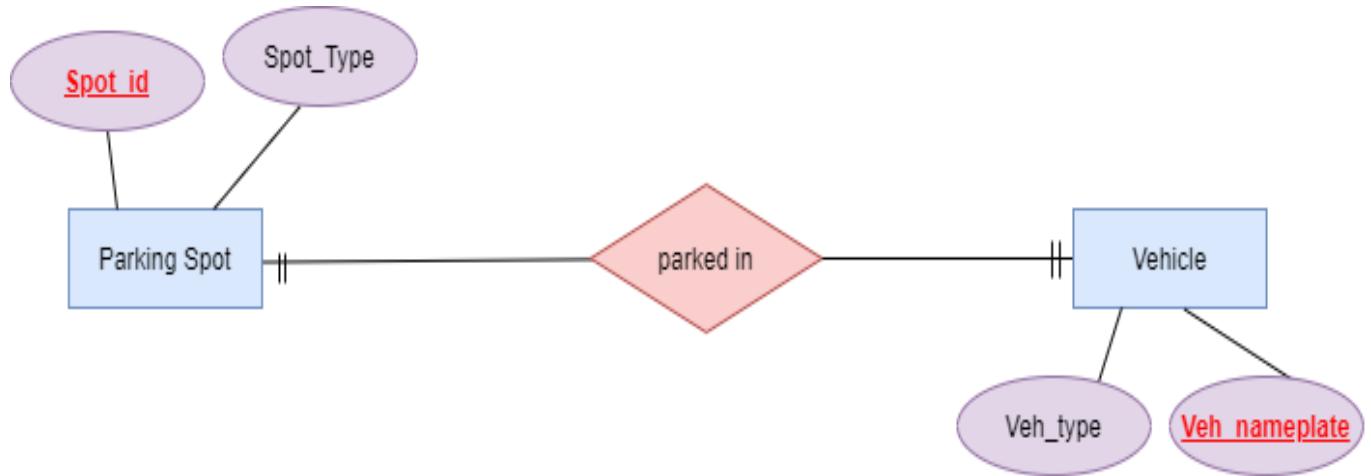
3rd: Veh_Nameplate, Veh_Type

Final Table:

1st: Dri_First_Name, Dri_Last_Name, Dri_Phone_Num, Dri_ID, Dri_Address, Veh_Nameplate

2nd: Veh_Nameplate, Veh_Type

Parking Spot1.... Parked in1 Vehicle



UNF:

Parked in (Spot_ID, Spot_Type, Veh_Nameplate, Veh_Type)

1NF:

1st: No Multivalue attribute

2nd: Spot_ID, Spot_Type, Veh_Nameplate, Veh_Type

2NF:

1st: Spot_ID, Spot_Type

2nd: Veh_Nameplate, Veh_Type, Spot_ID

3NF:

1st: No transitive attribute

2nd: Spot_ID, Spot_Type

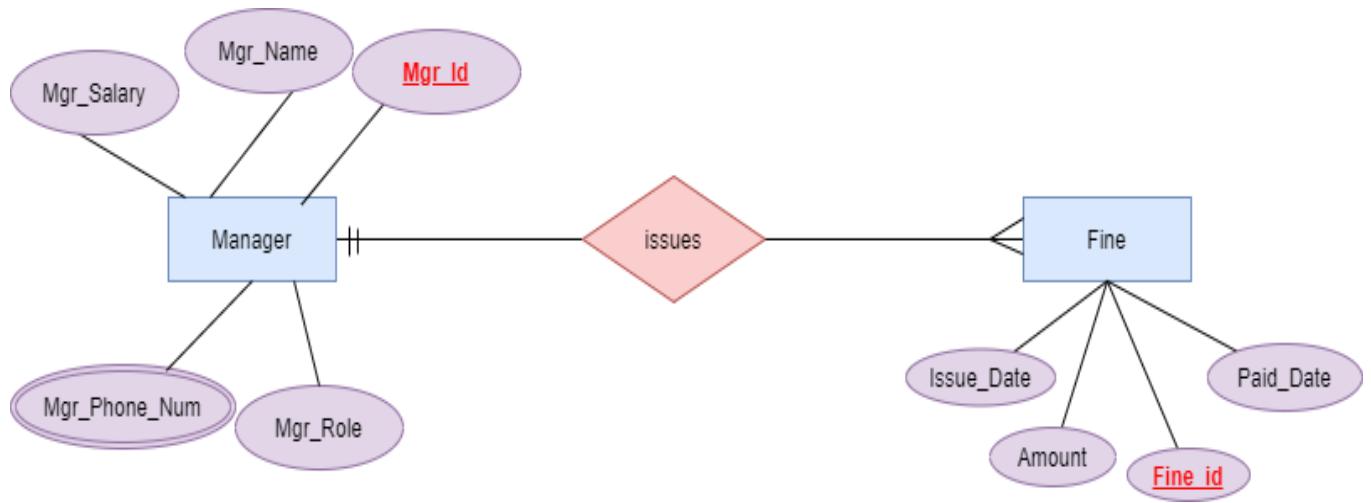
3rd: Veh_Nameplate, Veh_Type, Spot_ID

Final Table:

1st: Spot_ID, Spot_Type

2nd: Veh_Nameplate, Veh_Type, Spot_ID

Manager1.... Issues* Fine



UNF:

Issues (Mgr_ID, Mgr_Name, Mgr_Role, Mgr_Salary, Mgr_Phone_Num, Fine_ID, Amount, Issue_Date, Paid_Date)

1NF:

1st: Mgr_Phone_Num is a multivalue attribute

2nd: Mgr_ID, Mgr_Name, Mgr_Role, Mgr_Salary, Mgr_Phone_Num, Fine_ID, Amount, Issue_Date, Paid_Date

2NF:

1st: Mgr_ID, Mgr_Name, Mgr_Role, Mgr_Salary, Mgr_Phone_Num

2nd: Fine_ID, Amount, Issue_Date, Paid_Date, Mgr_ID

3NF:

1st: No Transitive attribute

2nd: Mgr_ID, Mgr_Name, Mgr_Role, Mgr_Salary, Mgr_Phone_Num

3rd: Fine_ID, Amount, Issue_Date, Paid_Date, Mgr_ID

Final Table:

1st: Mgr_ID, Mgr_Name, Mgr_Role, Mgr_Salary, Mgr_Phone_Num

2nd: Fine_ID, Amount, Issue_Date, Paid_Date, Mgr_ID

TOTAL TABLE:

1. Owner_Phone Num, Owner Name, Owner_Address, Owner_ID, Mgr_ID
2. Mgr_ID, Mgr_Name, Mgr_Role, Salary, Mgr_phone Num
3. Mgr_ID, Mgr_Name, Mgr_Role, Mgr_Salary, Mgr_Phone_Num, Space_ID
4. Space_ID, Space_Address, Space_Type
5. ~~Space_ID, Space_Address, Space_Type~~
6. Spot_ID, Spot_Type, Space_ID
7. Spot_ID, Spot_Type, Slip_ID
8. Slip_ID, Slip_Type, Duration, Issue_Date
9. Slip_ID, Slip_Type, Duration, Issue_Date, Dri_ID
10. Dri_ID, Dri_First Name, Dri_Last Name, Dri_Phone Num, Dri_Address
11. Spot_ID, Spot_Type
12. Rate_type, Hourly_Rate, Day_Rate, Rate_ID
13. Spot_ID, Rate_ID
14. Dri_First Name, Dri_Last Name, Dri_Phone Num, Dri_ID, Dri_Address, Fine_ID
15. Fine_ID, Amount, Issue_Date, Paid_Date
16. Dri_First Name, Dri_Last Name, Dri_Phone Num, Dri_ID, Dri_Address, Veh_Nameplate
17. : Veh_Nameplate, Veh_Type
18. ~~Spot_ID, Spot_Type~~
19. Veh_Nameplate, Veh_Type, Spot_ID
20. ~~Mgr_ID, Mgr_Name, Mgr_Role, Mgr_Salary, Mgr_Phone_Num~~
21. Fine_ID, Amount, Issue_Date, Paid_Date, Mgr_ID

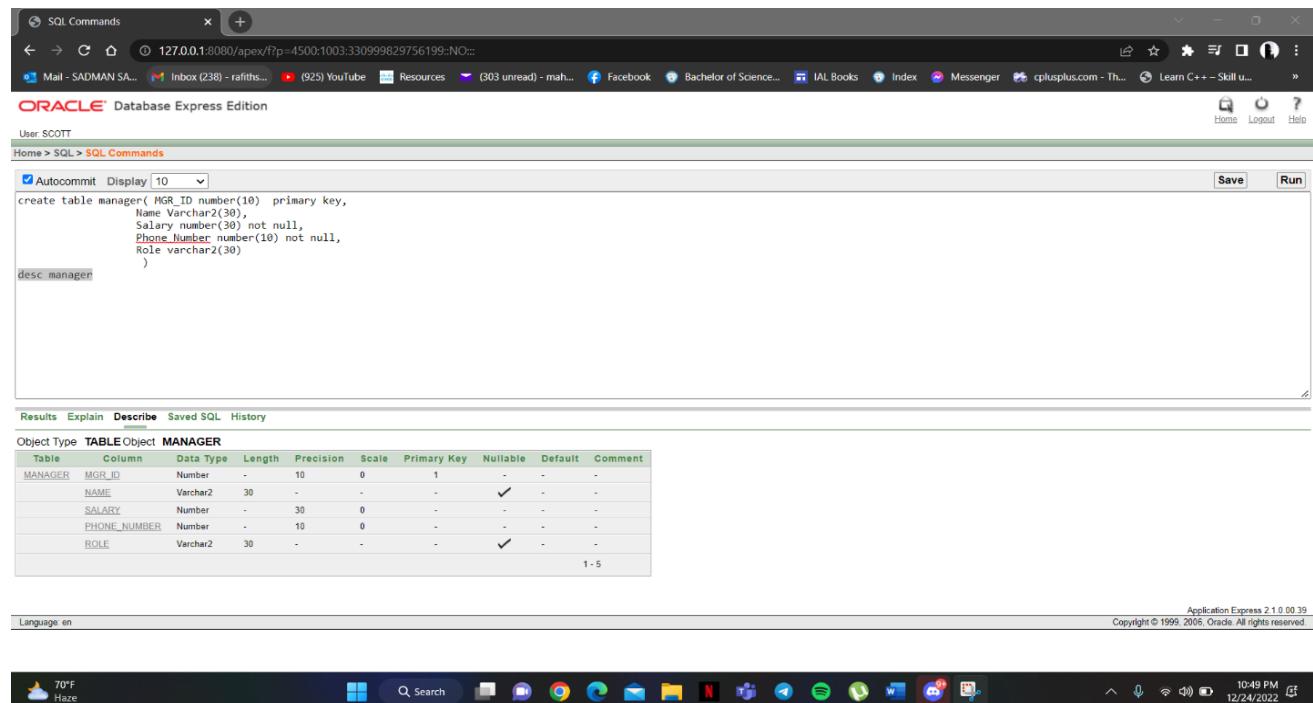
FINAL TABLE:

1. Owner_Phone Num, Owner Name, Owner_Address, Owner_ID, Mgr_ID
2. Mgr_ID, Mgr_Name, Mgr_Role, Salary, Mgr_phone Num
3. Mgr_ID, Mgr_Name, Mgr_Role, Mgr_Salary, Mgr_Phone_Num, Space_ID
4. Space_ID, Space_Address, Space_Type
5. Spot_ID, Spot_Type, Space_ID
6. Spot_ID, Spot_Type, Slip_ID
7. Slip_ID, Slip_Type, Duration, Issue_Date
8. Slip_ID, Slip_Type, Duration, Issue_Date, Dri_ID
9. Dri_ID, Dri_First Name, Dri_Last Name, Dri_Phone Num, Dri_Address
10. Spot_ID, Spot_Type
11. Rate_type, Hourly_Rate, Day_Rate, Rate_ID
12. Spot_ID, Rate_ID
13. Dri_First Name, Dri_Last Name, Dri_Phone Num, Dri_ID, Dri_Address, Fine_ID
14. Fine_ID, Amount, Issue_Date, Paid_Date
15. Dri_First Name, Dri_Last Name, Dri_Phone Num, Dri_ID, Dri_Address, Veh_Nameplate
16. : Veh_Nameplate, Veh_Type
17. Veh_Nameplate, Veh_Type, Spot_ID
18. Fine_ID, Amount, Issue_Date, Paid_Date, Mgr_ID

Table Creation

MANAGER TABLE:

```
create table Manager( MGR_ID number(10) primary key,  
                      Name Varchar2(30),  
                      Salary number(30) not null,  
                      Phone_Number number(10) not null,  
                      Role varchar2(30)  
)  
  
desc manager
```



The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following SQL code is entered:

```
create table Manager( MGR_ID number(10) primary key,  
                      Name Varchar2(30),  
                      Salary number(30) not null,  
                      Phone_Number number(10) not null,  
                      Role varchar2(30)  
)  
  
desc manager
```

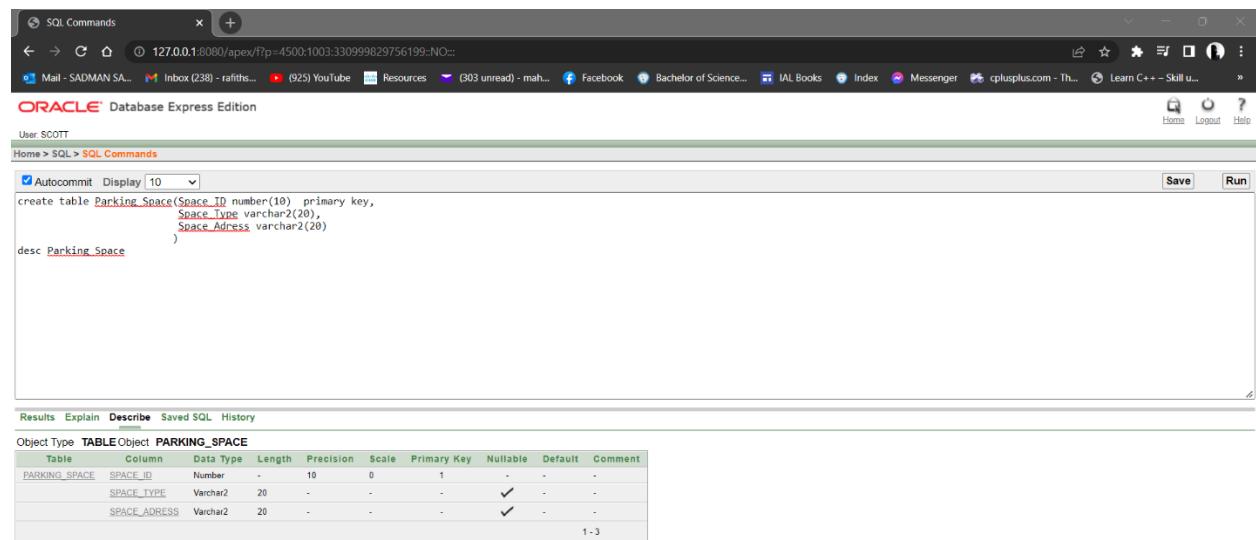
Below the code, the results of the `desc manager` command are displayed, showing the table structure:

Object Type	TABLE	Object	MANAGER						
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MANAGER	MGR_ID	Number	-	10	0	1	-	-	-
	NAME	Varchar2	30	-	-	-	✓	-	-
	SALARY	Number	-	30	0	-	-	-	-
	PHONE_NUMBER	Number	-	10	0	-	-	-	-
	ROLE	Varchar2	30	-	-	-	✓	-	-

At the bottom of the interface, the status bar shows "Language: en" and "Application Express 2.1.0.0.39 Copyright © 1999-2005, Oracle. All rights reserved." The system tray at the bottom of the screen shows various icons for system notifications.

PARKING SPACE TABLE:

```
create table Parking_Space(Space_ID number(10) primary key,  
                           Space_Type varchar2(20),  
                           Space_Adress varchar2(20)  
)  
  
desc Parking_Space
```



The screenshot shows the Oracle Database Express Edition SQL Commands window. The SQL editor contains the following code:

```
create table Parking_Space(Space_ID number(10) primary key,  
                           Space_Type varchar2(20),  
                           Space_Adress varchar2(20)  
)  
  
desc Parking_Space
```

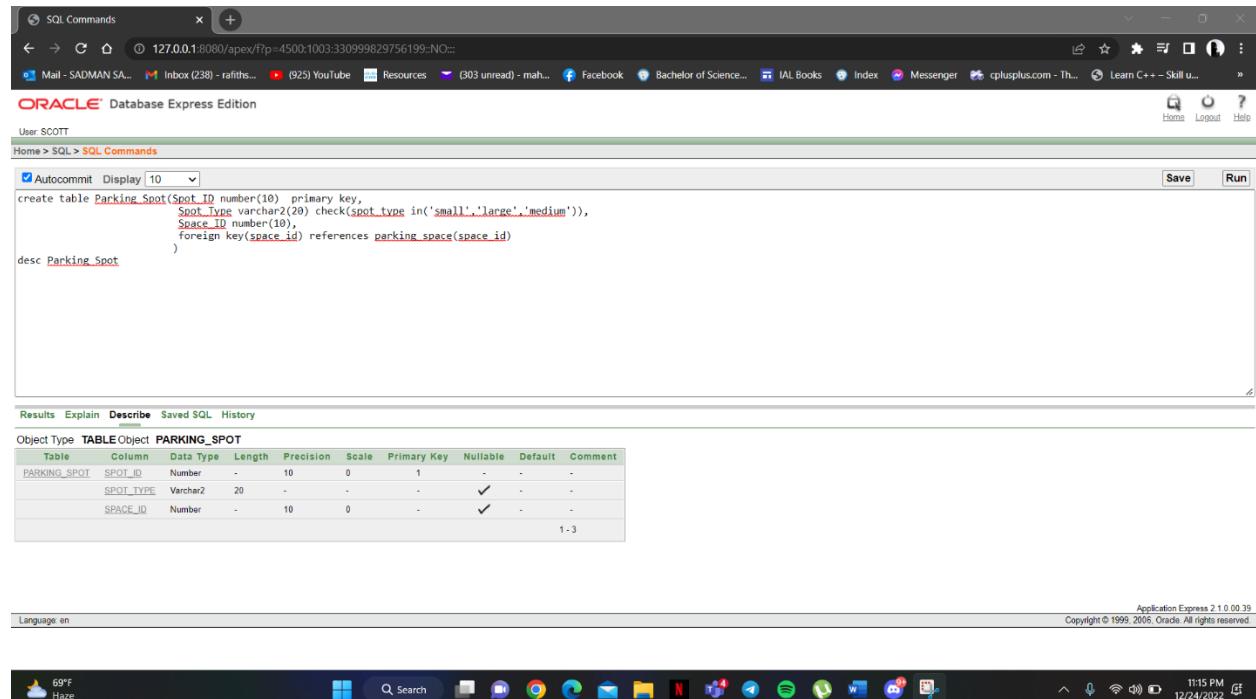
Below the code, the table structure is displayed:

Object Type	TABLE	Object Name	PARKING_SPACE							
PARKING_SPACE	SPACE_ID	Number	-	10	0	-	1	-	-	-
	SPACE_TYPE	Varchar2	20	-	-	-	-	✓	-	-
	SPACE_ADDRESS	Varchar2	20	-	-	-	-	✓	-	-



PARKING SPOT TABLE:

```
create table Parking_Spot(Spot_ID number(10) primary key,
                           Spot_Type varchar2(20) check(spot_type
                           in('small','large','medium')),
                           Space_ID number(10),
                           foreign key(space_id) references parking_space(space_id)
                           )
desc Parking_Spot
```

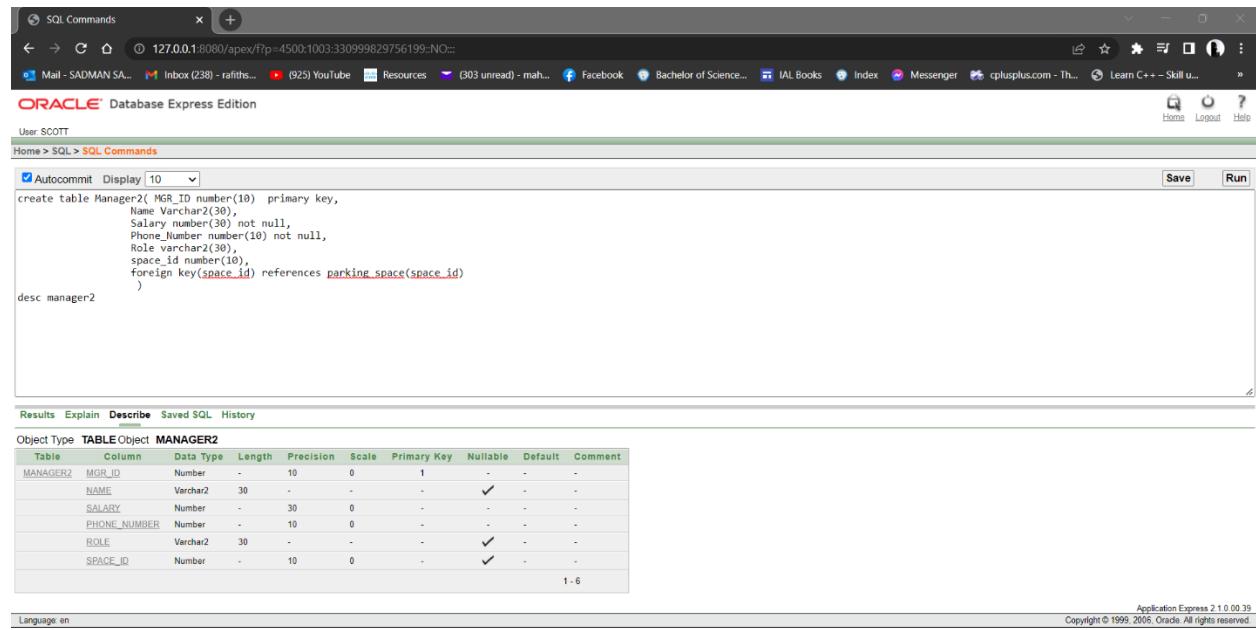


The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL command window displays the creation of the `PARKING_SPOT` table and its description. The table structure includes columns `Spot_ID` (number(10), primary key), `Spot_Type` (varchar2(20) with a check constraint for values 'small', 'large', 'medium'), and `Space_ID` (number(10)). A foreign key constraint `Space_ID` references the `parking_space` table. The command `desc Parking_Spot` is also shown. Below the command window, the object list shows the `PARKING_SPOT` table with its columns: `SPOT_ID` (Number, Primary Key), `SPOT_TYPE` (Varchar2), and `SPACE_ID` (Number). The interface also includes a toolbar, a menu bar, and a status bar at the bottom.

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PARKING_SPOT	SPOT_ID	Number	-	10	0	1	-	-	-
	SPOT_TYPE	Varchar2	20	-	-	-	✓	-	-
	SPACE_ID	Number	-	10	0	-	✓	-	-

MANAGER2 TABLE:

```
create table Manager2( MGR_ID number(10) primary key,
                      Name Varchar2(30),
                      Salary number(30) not null,
                      Phone_Number number(10) not null,
                      Role varchar2(30),
                      space_id number(10),
                      foreign key(space_id) references parking_space(space_id)
)
desc manager2
```



The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window displays the SQL code for creating the Manager2 table and its description. The table has columns for MGR_ID (primary key, number(10)), Name (Varchar2(30)), Salary (number(30) not null), Phone_Number (number(10) not null), Role (varchar2(30)), and space_id (number(10)). A foreign key constraint links space_id to the space_id column in the parking_space table. The 'Display' dropdown is set to 10. Below the code, the 'desc manager2' command is shown. The results tab is selected, displaying the table structure with columns: Table, Column, Data Type, Length, Precision, Scale, Primary Key, Nullable, Default, and Comment. The table has 6 columns: MGR_ID, NAME, SALARY, PHONE_NUMBER, ROLE, and SPACE_ID. The Primary Key is MGR_ID. Nullable is checked for all columns except Primary Key. The 'Object Type' is TABLE and the 'Object Name' is MANAGER2. The bottom status bar shows the date and time as 12:14 AM 12/25/2022.

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MANAGER2	MGR_ID	Number	-	10	0	1	-	-	-
	NAME	Varchar2	30	-	-	-	✓	-	-
	SALARY	Number	-	30	0	-	-	-	-
	PHONE_NUMBER	Number	-	10	0	-	-	-	-
	ROLE	Varchar2	30	-	-	-	✓	-	-
	SPACE_ID	Number	-	10	0	-	✓	-	-

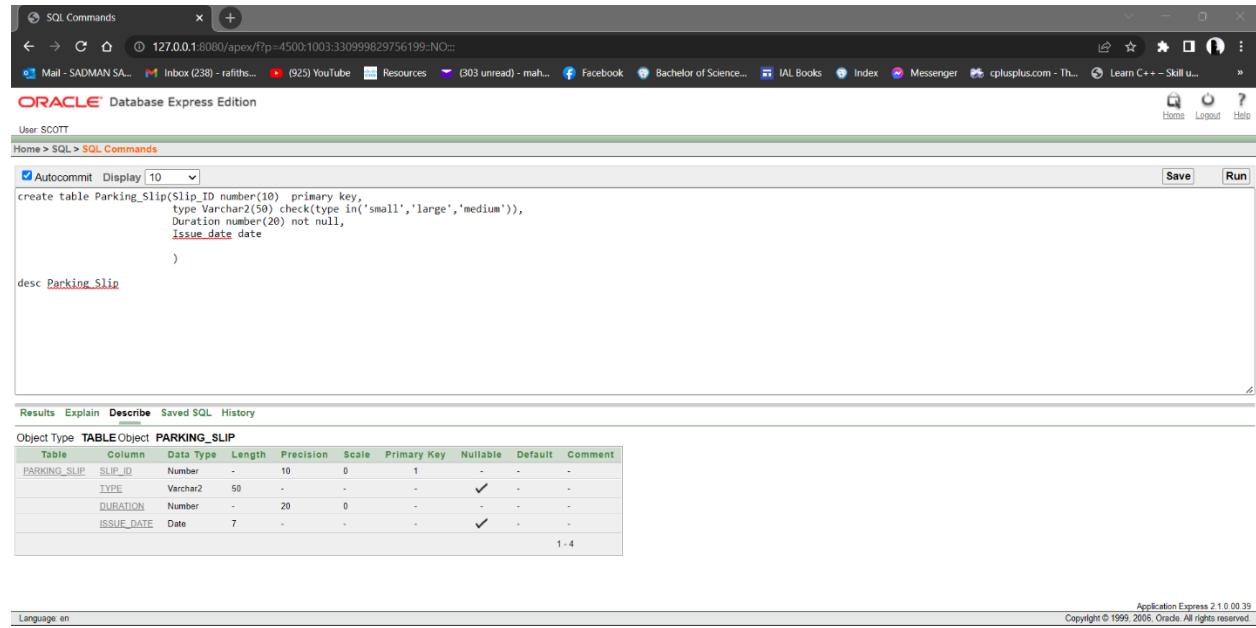


PARKING SLIP TABLE:

```
create table Parking_Slip(Slip_ID number(10) primary key,
                           type Varchar2(50) check(type in('small','large','medium')),
                           Duration number(20) not null,
                           Issue_date date
)

```

```
desc Parking_Slip
```



The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following SQL code is entered:

```
create table Parking_Slip(Slip_ID number(10) primary key,
                           type Varchar2(50) check(type in('small','large','medium')),
                           Duration number(20) not null,
                           Issue_date date
)
desc Parking_Slip
```

Below the code, the results of the `desc` command are displayed:

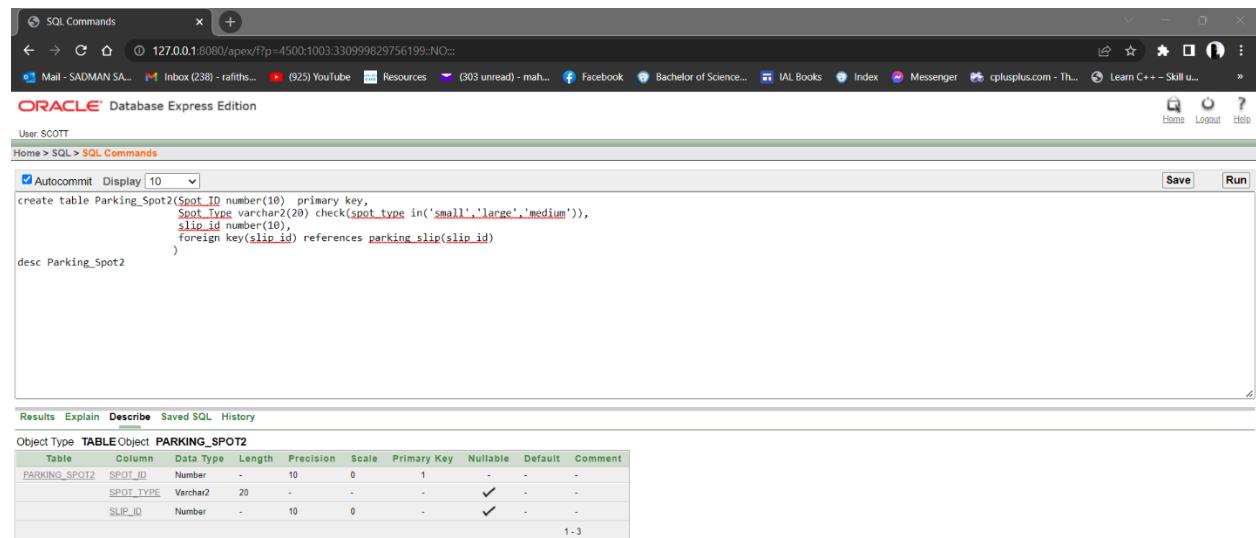
Object Type	TABLE	Object	PARKING_SLIP						
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PARKING_SLIP	SLIP_ID	Number	-	10	0	1	-	-	-
	TYPE	Varchar2	50	-	-	-	✓	-	-
	DURATION	Number	-	20	0	-	-	-	-
	ISSUE_DATE	Date	7	-	-	-	✓	-	-

At the bottom of the interface, the status bar shows "Language: en" and "Application Express 2.1.0.00.39 Copyright © 1995, 2006, Oracle. All rights reserved."

PARKING SPOT 2 TABLE:

```
create table Parking_Spot2(Spot_ID number(10) primary key,  
                           Spot_Type varchar2(20) check(spot_type  
                           in('small','large','medium')),  
                           slip_id number(10),  
                           foreign key(slip_id) references parking_slip(slip_id)  
)
```

desc Parking_Spot2



The screenshot shows the Oracle Database Express Edition SQL Commands window. The SQL code entered is:

```
create table Parking_Spot2(Spot_ID number(10) primary key,  
                           Spot_Type varchar2(20) check(spot_type in('small','large','medium')),  
                           slip_id number(10),  
                           foreign key(slip_id) references parking_slip(slip_id)  
)  
  
desc Parking_Spot2
```

Below the code, the results of the **desc** command are displayed in a table:

Object Type	TABLE Object	PARKING_SPOT2							
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PARKING_SPOT2	SPOT_ID	Number	-	10	0	1	-	-	-
	SPOT_TYPE	Varchar2	20	-	-	-	✓	-	-
	SLIP_ID	Number	-	10	0	-	✓	-	-



DRIVER TABLE:

```
create table Driver(Dri_ID number(10) primary key,  
                    dri_First_name Varchar2(50),  
                    dri_Last_name Varchar2(50),  
                    dri_Phone_Num number(11) unique,  
                    dri_Email varchar2(50) unique,  
                    dri_address varchar2(50)  
)
```

desc Driver

SQL Commands x +

127.0.0.1:8080/apex/?p=4500:1003:330999829756199:No::

Mail - SADMAN SA... Inbox (238 - rafiths... YouTube Resources (303 unread) - mah... Facebook Bachelor of Science... IAL Books Index Messenger cplusplus.com - Th... Learn C++ – Skill u... Home Logout Help

ORACLE Database Express Edition

User: SCOTT

Home > SQL > SQL Commands

Autocommit

```
create table Driver(Dri_ID number(10) primary key,
dri_First_name Varchar2(50),
dri_Last_name Varchar2(50),
dri_Phone_Num number(11) unique,
dri_Email varchar2(50) unique,
dri_address varchar2(50)
)
desc Driver
```

Results Explain Describe Saved SQL History

Object Type: TABLE Object: DRIVER

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DRIVER	DRI_ID	Number	-	10	0	1	-	-	-
DRIVER	DRI_FIRST_NAME	Varchar2	50	-	-	-	✓	-	-
DRIVER	DRI_LAST_NAME	Varchar2	50	-	-	-	✓	-	-
DRIVER	DRI_PHONE_NUM	Number	-	11	0	-	✓	-	-
DRIVER	DRI_EMAIL	Varchar2	50	-	-	-	✓	-	-
DRIVER	DRI_ADDRESS	Varchar2	50	-	-	-	✓	-	-

1 - 6



PARKING SLIP 2 TABLE:

```
create table Parking_Slip2(Slip_ID number(10) primary key,
                           type Varchar2(50) check(type in('small','large','medium')),
                           Duration number(20) not null,
                           Issue_date date,
                           dri_id number(10),
                           foreign key(dri_id) references driver(dri_id)
                           )
```

```
desc Parking_Slip2
```

The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window displays the creation of the PARKING_SLIP2 table with the specified columns and constraints. Below the SQL window, the 'Results' tab is selected, showing the table description with columns: SLIP_ID, TYPE, DURATION, ISSUE_DATE, and DRI_ID. The table has 5 rows and 5 columns. The bottom status bar indicates the application version and copyright information.

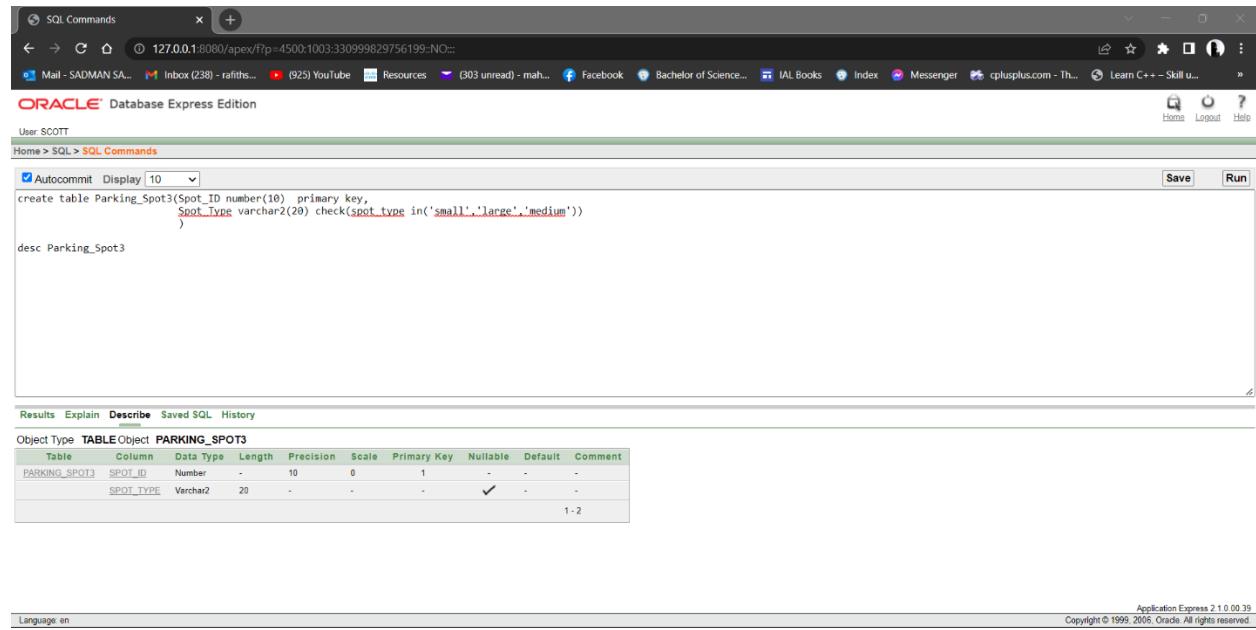
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PARKING_SLIP2	SLIP_ID	Number	-	10	0	1	-	-	-
	TYPE	Varchar2	50	-	-	-	✓	-	-
	DURATION	Number	-	20	0	-	-	-	-
	ISSUE_DATE	Date	7	-	-	-	✓	-	-
	DRI_ID	Number	-	10	0	-	✓	-	-



PARKING SPOT 3 TABLE:

```
create table Parking_Spot3(Spot_ID number(10) primary key,  
                           Spot_Type varchar2(20) check(spot_type  
                           in('small','large','medium'))  
                           )
```

```
desc Parking_Spot3
```



The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL command window contains the following code:

```
create table Parking_Spot3(Spot_ID number(10) primary key,  
                           Spot_Type varchar2(20) check(spot_type in('small','large','medium'))  
                           )  
  
desc Parking_Spot3
```

Below the code, the results of the `desc` command are shown:

Object Type	TABLE	Object	PARKING_SPOT3						
PARKING_SPOT3	SPOT_ID	Number	-	10	0	1	-	-	-
	SPOT_TYPE	VARCHAR2	20	-	-	-	✓	-	-

At the bottom of the interface, the status bar shows "Language: en" and "Application Express 2.1.0.00.39 Copyright © 1995, 2005, Oracle. All rights reserved."



RATE TABLE:

```
create table Rate( Rate_ID number(10) primary key,  
    rate_type Varchar2(30),  
    rate_hourly_rate number(30) not null,  
    rate_day_rate number(10)  
)
```

```
desc rate
```

The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window displays the creation of the RATE table and its description. The table structure is as follows:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
RATE	RATE_ID	Number	-	10	0	1	-	-	-
RATE_TYPE	Varchar2	30	-	-	-	-	✓	-	-
RATE_HOURLY_RATE	Number	-	30	0	-	-	-	-	-
RATE_DAY_RATE	Number	-	10	0	-	-	✓	-	-

At the bottom, the status bar shows "Language: en" and "Application Express 2.1.0.0.39 Copyright © 1995, 2005, Oracle. All rights reserved."

SPOT RATE:

```
create table spot_rate( Rate_ID number(10),
                      foreign key(rate_id) references rate(rate_id),
                      spot_id number(10),
                      foreign key(spot_id) references parking_spot(spot_id)
)
```

```

create table spot_rate( Rate_ID number(10),
    foreign key(rate_id) references rate(rate_id),
    spot_id number(10),
    foreign key(spot_id) references parking_spot(spot_id)
)
desc spot_rate

```

Results Explain Describe Saved SQL History

Object Type TABLE Object SPOT_RATE

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
SPOT_RATE	RATE_ID	Number	-	10	0	-	✓	-	-
	SPOT_ID	Number	-	10	0	-	✓	-	-

1 - 2



FINE TABLE:

```
create table Fine(Fine_ID number(10) primary key,  
                  Amount number(30) not null,  
                  Issue_date date,  
                  Paid_date date  
)
```

```
desc Fine
```

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following SQL code is entered:

```
create table Fine(Fine_ID number(10) primary key,  
                  Amount number(30) not null,  
                  Issue_date date,  
                  Paid_date date  
)  
  
desc Fine
```

Below the code, the results of the `desc Fine` command are displayed, showing the table structure:

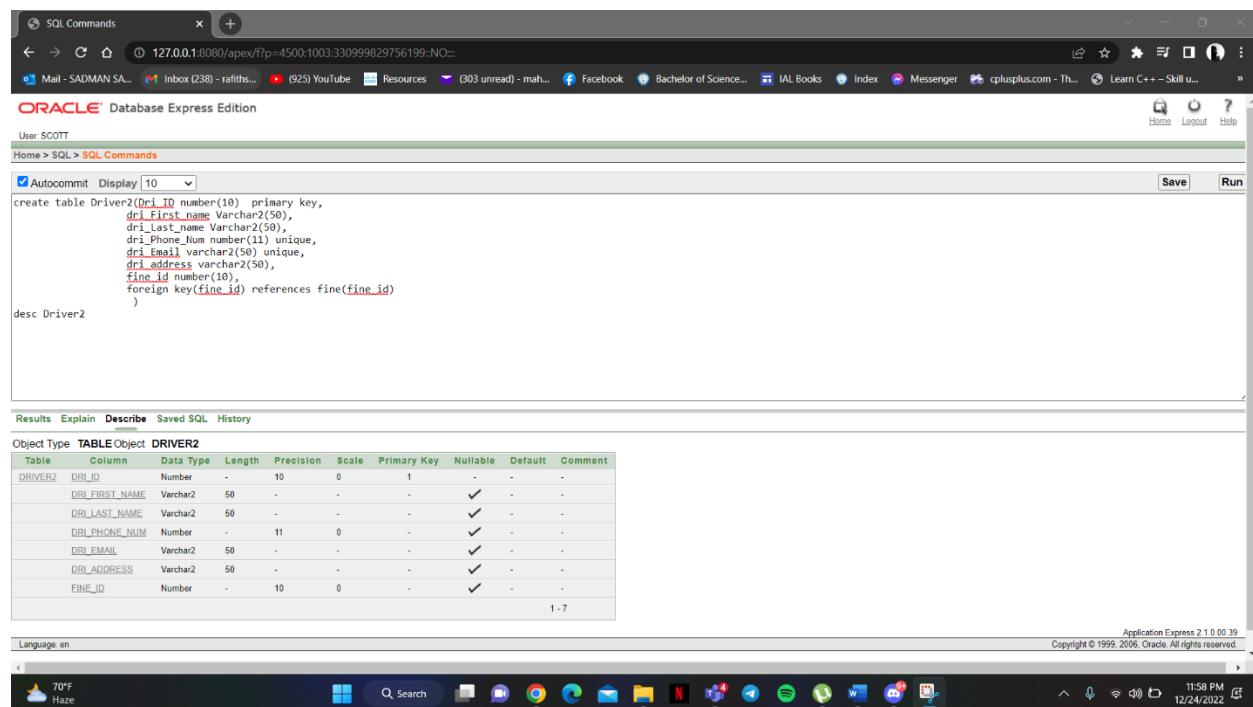
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
FINE	FINE_ID	Number	-	10	0	1	-	-	-
	AMOUNT	Number	-	30	0	-	-	-	-
	ISSUE_DATE	Date	7	-	-	-	✓	-	-
	Paid_DATE	Date	7	-	-	-	✓	-	-

At the bottom of the interface, the status bar shows "Language: en" and "Application Express 21.0.0.30". The system tray at the bottom of the screen shows the date and time as "11:54 PM 12/24/2022".

DRIVER2_TABLE:

```
create table Driver2(Dri_ID number(10) primary key,  
                     dri_First_name Varchar2(50),  
                     dri_Last_name Varchar2(50),  
                     dri_Phone_Num number(11) unique,  
                     dri_Email varchar2(50) unique,  
                     dri_address varchar2(50),  
                     fine_id number(10),  
                     foreign key(fine_id) references fine(fine_id)  
                )
```

```
desc Driver2
```



The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window displays the creation of the Driver2 table and its description. The table structure is as follows:

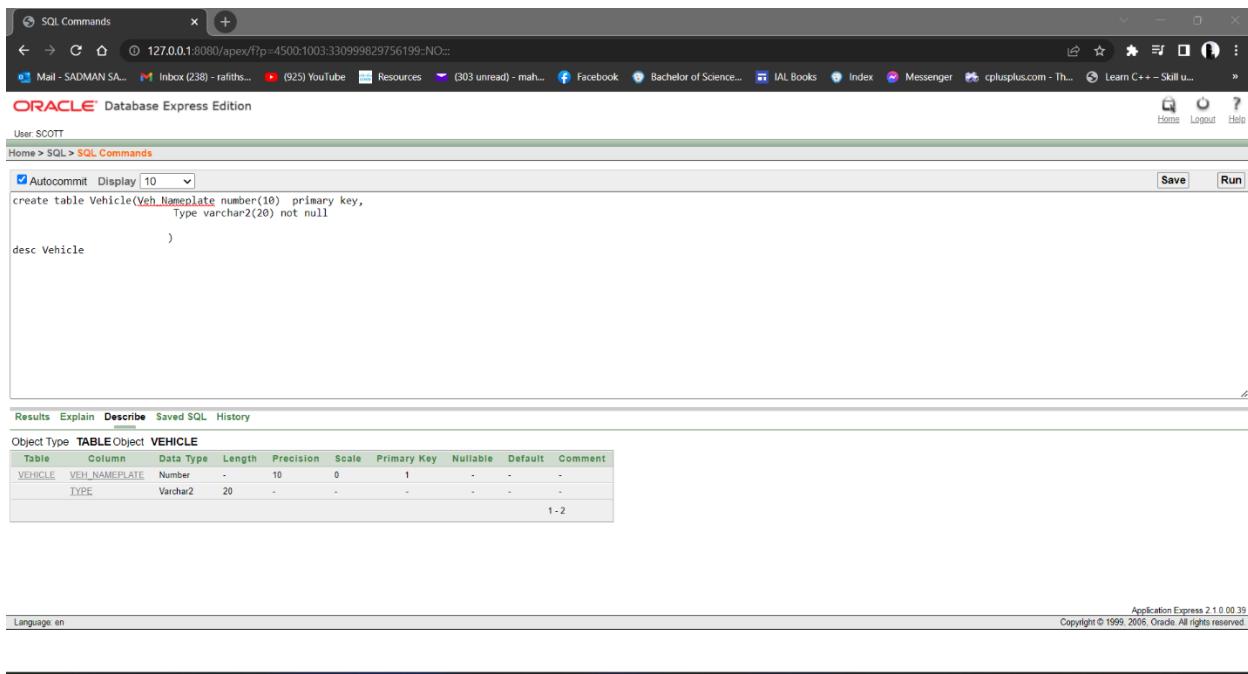
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DRIVER2	DRI_ID	Number	10	0	1	✓	-	-	
DRIVER2	DRI_FIRST_NAME	Varchar2	50	-	-	✓	-	-	
DRIVER2	DRI_LAST_NAME	Varchar2	50	-	-	✓	-	-	
DRIVER2	DRI_PHONE_NUM	Number	-	11	0	✓	-	-	
DRIVER2	DRI_EMAIL	Varchar2	50	-	-	✓	-	-	
DRIVER2	DRI_ADDRESS	Varchar2	50	-	-	✓	-	-	
DRIVER2	FINE_ID	Number	-	10	0	✓	-	-	

The results tab shows the table creation command and the description output. The status bar at the bottom right indicates the application version (Application Express 2.1.0.00.39) and copyright information (Copyright © 1999, 2006, Oracle. All rights reserved).

VEHICLE TABLE:

```
create table Vehicle(Veh_Nameplate number(10) primary key,  
                     Type varchar2(20) not null  
                     )
```

desc Vehicle

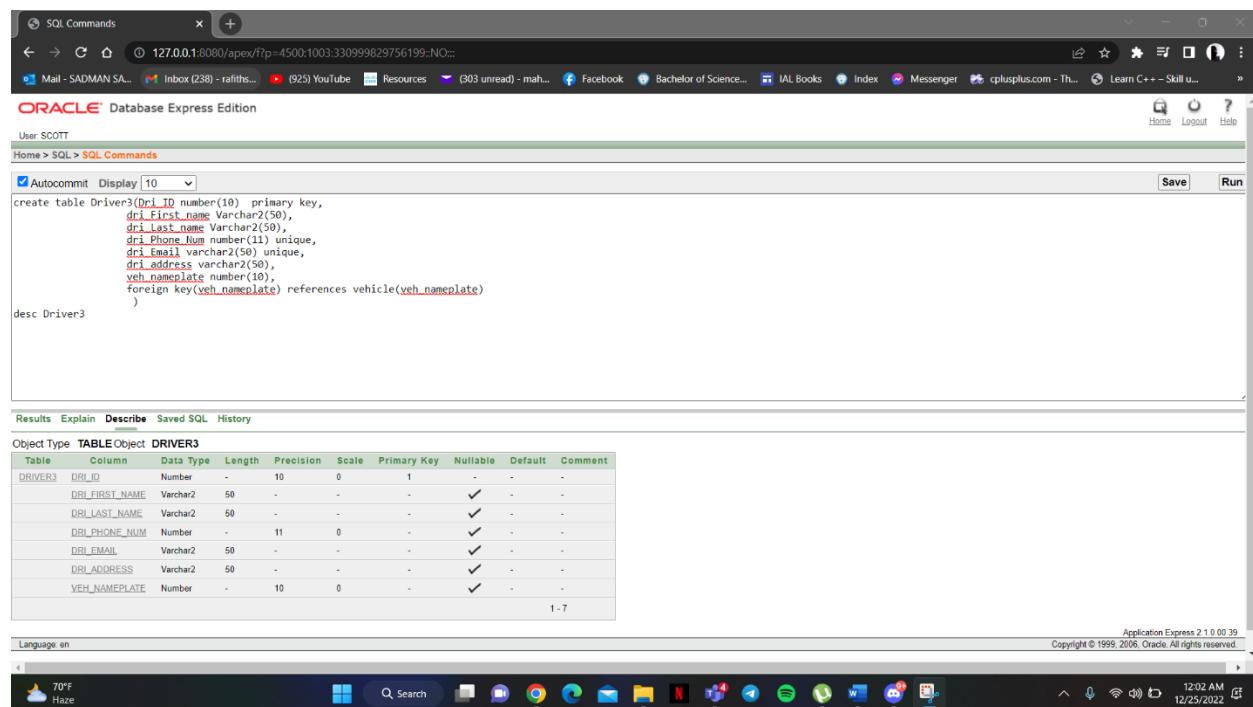


DRIVER3_TABLE:

```
create table Driver3(Dri_ID number(10) primary key,
                     dri_First_name Varchar2(50),
                     dri_Last_name Varchar2(50),
                     dri_Phone_Num number(11) unique,
                     dri_Email varchar2(50) unique,
                     dri_address varchar2(50),
                     veh_nameplate number(10),
                     foreign key(veh_nameplate) references vehicle(veh_nameplate)
)

```

```
desc Driver3
```



The screenshot shows a browser window for Oracle Database Express Edition. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:330999829756199:NO:::1. The page displays SQL commands for creating the Driver3 table and its description. The table structure is as follows:

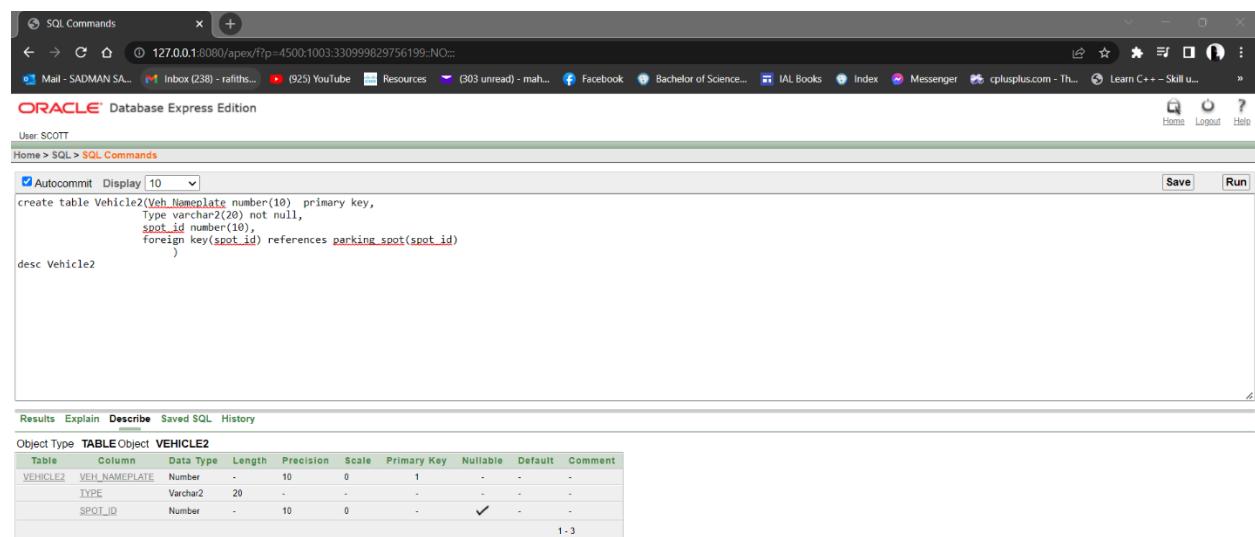
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DRIVER3	DRI_ID	Number	10	0	1	✓	-	-	
DRIVER3	DRI_FIRST_NAME	Varchar2	50	-	-	✓	-	-	
DRIVER3	DRI_LAST_NAME	Varchar2	50	-	-	✓	-	-	
DRIVER3	DRI_PHONE_NUM	Number	-	11	0	✓	-	-	
DRIVER3	DRI_EMAIL	Varchar2	50	-	-	✓	-	-	
DRIVER3	DRI_ADDRESS	Varchar2	50	-	-	✓	-	-	
DRIVER3	VEH_NAMEPLATE	Number	-	10	0	✓	-	-	

At the bottom of the page, there is a note: "Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved." The browser status bar shows the date and time: 12/25/2022 12:02 AM.

VEHICLE2 TABLE:

```
create table Vehicle2(Veh_Nameplate number(10) primary key,  
                      Type varchar2(20) not null,  
                      spot_id number(10),  
                      foreign key(spot_id) references parking_spot(spot_id)  
                      )
```

```
desc Vehicle2
```



The screenshot shows the Oracle Database Express Edition SQL Commands window. The SQL code entered is:

```
create table Vehicle2(Veh_Nameplate number(10) primary key,  
                      Type varchar2(20) not null,  
                      spot_id number(10),  
                      foreign key(spot_id) references parking_spot(spot_id)  
                      )  
  
desc Vehicle2
```

Below the code, the results of the `desc Vehicle2` command are displayed in a table:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
VEHICLE2	VEH_NAMEPLATE	Number	-	10	0	1	-	-	-
	TYPE	Varchar2	20	-	-	-	-	-	-
	SPOT_ID	Number	-	10	0	-	✓	-	-

At the bottom of the window, the status bar shows "Language: en" and "Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved."



FINE2 TABLE:

```
create table Fine2(Fine_ID number(10) primary key,  
                    Amount number(30) not null,  
                    Issue_date date,  
                    Paid_date date,  
                    MGR_ID number(10),  
                    foreign key(mgr_id) references manager(mgr_id)  
                    )
```

```
desc Fine2
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL command window contains the following code:

```
create table Fine2(Fine_ID number(10) primary key,  
                    Amount number(30) not null,  
                    Issue_date date,  
                    Paid_date date,  
                    MGR_ID number(10),  
                    foreign key(mgr_id) references manager(mgr_id)  
                    )  
desc Fine2
```

Below the code, the 'Results' tab is selected, showing the description of the FINE2 table:

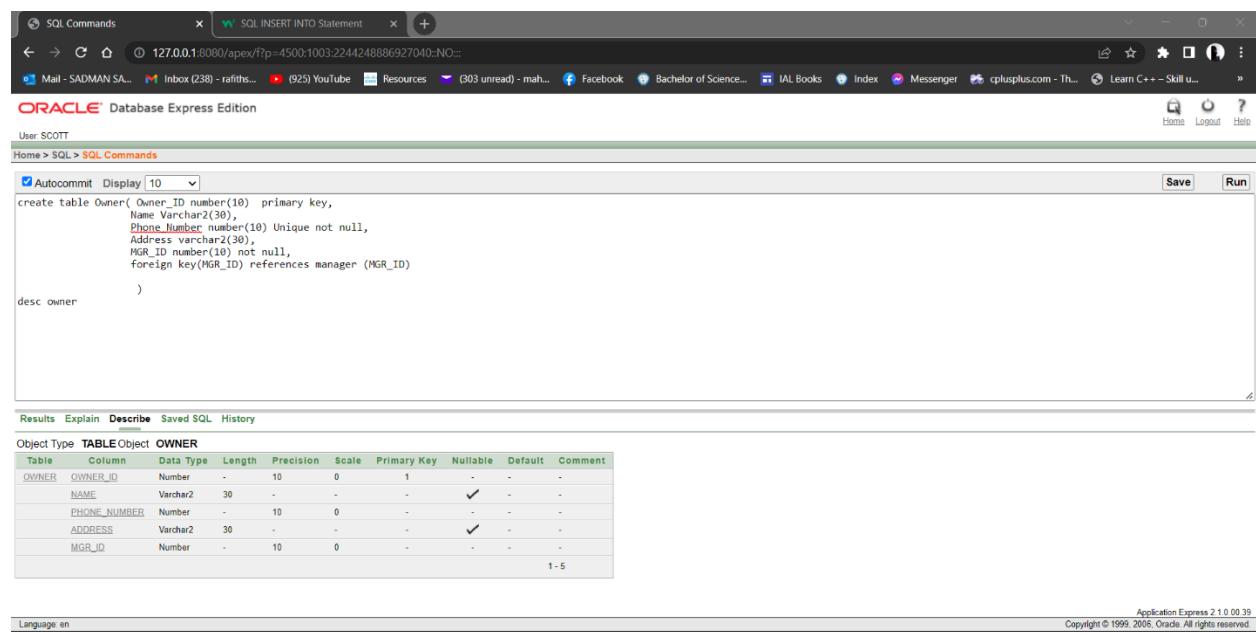
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
FINE2	FINE_ID	Number	-	10	0	1	-	-	-
	AMOUNT	Number	-	30	0	-	-	-	-
	ISSUE_DATE	Date	7	-	-	-	✓	-	-
	PAYOUT_DATE	Date	7	-	-	-	✓	-	-
	MGR_ID	Number	-	10	0	-	✓	-	-

At the bottom of the interface, the status bar shows 'Language: en' and 'Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.'



OWNER TABLE:

```
create table Owner( Owner_ID number(10) primary key,  
                    Name Varchar2(30),  
                    Phone_Number number(10) Unique not null,  
                    MGR_ID number(10) not null,  
                    foreign key(MGR_ID) references manager (MGR_ID)  
)  
desc owner
```



The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window contains the SQL code for creating the OWNER table and its description. The table has columns for Owner_ID (primary key, number(10)), Name (Varchar2(30)), Phone_Number (number(10), Unique, not null), and MGR_ID (number(10), not null, foreign key referencing the MGR_ID column of the manager table). The desc owner command is also present. Below the SQL window, the Object Type TABLE Object OWNER is displayed, showing the table structure with columns: OWNER_ID, NAME, PHONE_NUMBER, ADDRESS, and MGR_ID. The table has 5 rows. At the bottom of the interface, the Application Express 2.1.0.00.39 version and Oracle copyright information are visible.

```
create table Owner( Owner_ID number(10) primary key,  
                    Name Varchar2(30),  
                    Phone_Number number(10) Unique not null,  
                    MGR_ID number(10) not null,  
                    foreign key(MGR_ID) references manager (MGR_ID)  
)  
desc owner
```

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
OWNER	OWNER_ID	Number	-	10	0	1	✓	-	-
	NAME	Varchar2	30	-	-	-	✓	-	-
	PHONE_NUMBER	Number	-	10	0	-	-	-	-
	ADDRESS	Varchar2	30	-	-	-	✓	-	-
	MGR_ID	Number	-	10	0	-	-	-	-



Data Insertion

MANAGER_TABLE:

```
create sequence mgr_squ
```

```
increment by 1
```

```
start with 10
```

```
maxvalue 100
```

```
insert into manager(mgr_id, name, phone_number, salary, role) values  
(mgr_squ.nextval, 'Diana Lorentz', 5904235567, 42000, 'genral manager')
```

```
insert into manager(mgr_id, name, phone_number, salary, role) values  
(mgr_squ.nextval, 'John Chen', 515124426, 82000, 'accounts manager')
```

```
insert into manager(mgr_id, name, phone_number, salary, role) values  
(mgr_squ.nextval, 'Ismael Sciarra', 5151244369, 77000, 'staff manager')
```

```
insert into manager(mgr_id, name, phone_number, salary, role) values  
(mgr_squ.nextval, 'Jose Manuel Urman', 5151244469, 78000, 'parking lot  
manager')
```

```
insert into manager(mgr_id, name, phone_number, salary, role) values  
(mgr_squ.nextval, 'hero alom', 5904235567, 42000, 'genral manager')
```

```
insert into manager(mgr_id, name, phone_number, salary, role) values  
(mgr_squ.nextval, 'anonto jalil', 5904235567, 42000, 'genral manager')
```

```
insert into manager(mgr_id, name, phone_number, salary, role) values  
(mgr_squ.nextval, 'pori moni', 5904235567, 42000, 'genral manager')
```

```
select * from manager
```

```
insert into manager(mgr_id, name, phone_number, salary, role) values (mgr_squ.nextval, 'Diana Lorentz', 5904235567, 42000, 'genral manager')  
insert into manager(mgr_id, name, phone_number, salary, role) values (mgr_squ.nextval, 'John Chen', 515124426, 82000, 'accounts manager')  
insert into manager(mgr_id, name, phone_number, salary, role) values (mgr_squ.nextval, 'Ismael Sciarra', 5151244369, 77000, 'staff manager')  
insert into manager(mgr_id, name, phone_number, salary, role) values (mgr_squ.nextval, 'Jose Manuel Urman', 5151244469, 78000, 'parking lot manager')  
insert into manager(mgr_id, name, phone_number, salary, role) values (mgr_squ.nextval, 'hero alom', 5904235567, 42000, 'genral manager')  
insert into manager(mgr_id, name, phone_number, salary, role) values (mgr_squ.nextval, 'anonto jalil', 5904235567, 42000, 'genral manager')  
insert into manager(mgr_id, name, phone_number, salary, role) values (mgr_squ.nextval, 'pori moni', 5904235567, 42000, 'genral manager')  
select * from manager
```

Results Explain Describe Saved SQL History				
MGR_ID	NAME	SALARY	PHONE_NUMBER	ROLE
11	John Chen	82000	515124426	accounts manager
12	Ismael Sciarra	77000	5151244369	staff manager
13	Jose Manuel Urman	78000	5151244469	parking lot manager
30	hero alom	42000	5904235567	genral manager
31	anonto jalil	42000	5904235567	genral manager
32	pori moni	42000	5904235567	genral manager
10	Diana Lorentz	42000	5904235567	genral manager

7 rows returned in 0.00 seconds CSV Export

Application Express 21.1.0.39
Copyright © 1999, 2006, Oracle. All rights reserved.



PARKING SPACE TABLE:

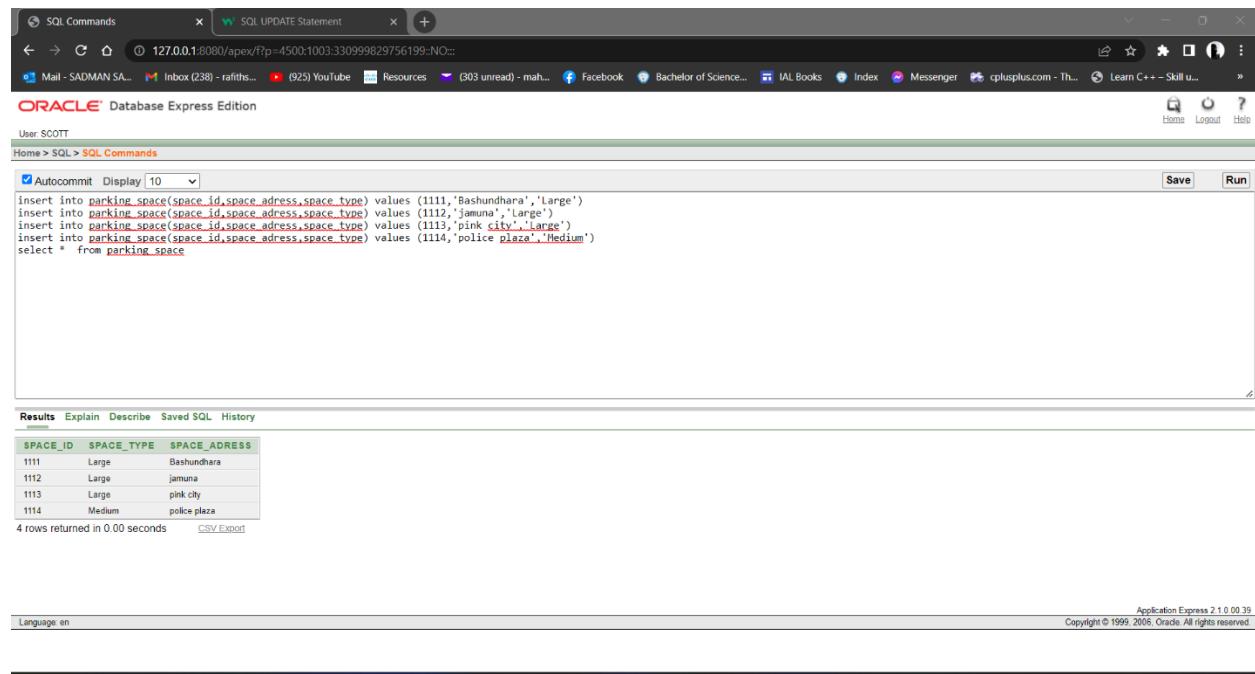
```
insert into parking_space(space_id,space_adress,space_type) values  
(1111,'Bashundhara','Large')
```

```
insert into parking_space(space_id,space_adress,space_type) values  
(1112,'jamuna','Large')
```

```
insert into parking_space(space_id,space_adress,space_type) values  
(1113,'pink city','Large')
```

```
insert into parking_space(space_id,space_adress,space_type) values  
(1114,'police plaza','Medium')
```

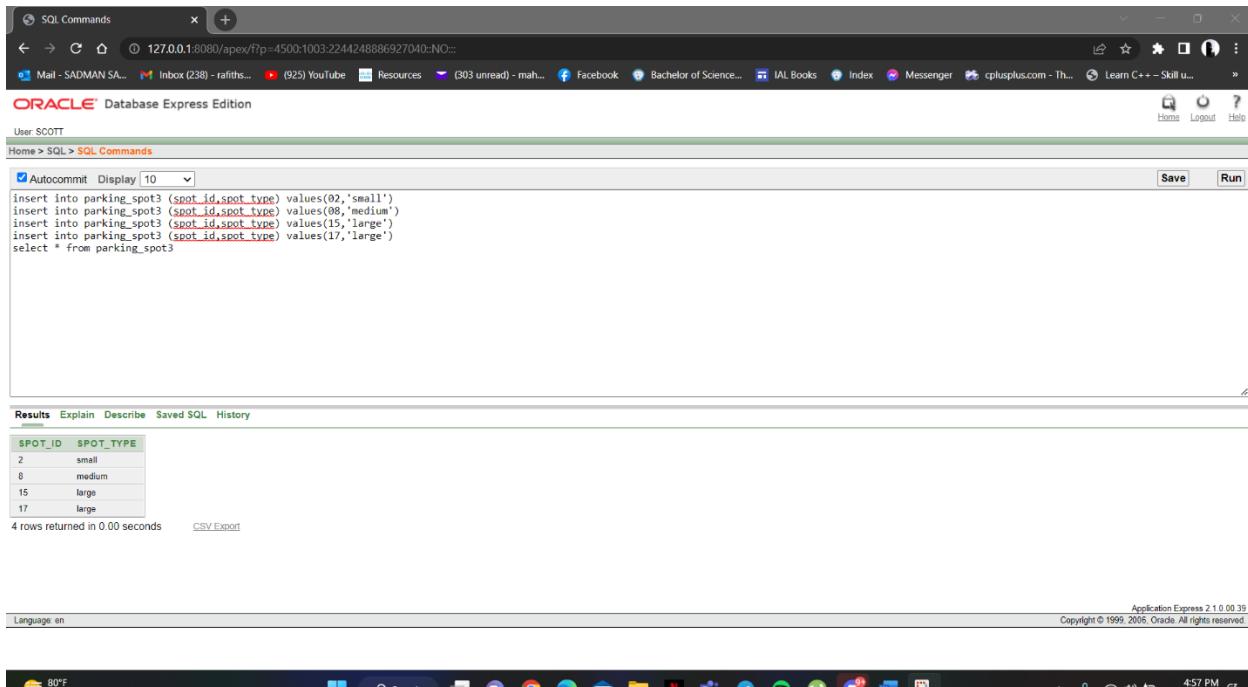
```
select * from parking_space
```



```
SQL Commands SQL UPDATE Statement +  
127.0.0.1:8080/apex/f?p=4500:1003:330999829756199:NO:  
User SCOTT  
Home > SQL > SQL Commands  
Autocommit Display 10 Save Run  
insert into parking_space(space_id,space_adress,space_type) values (1111,'Bashundhara','Large')  
insert into parking_space(space_id,space_adress,space_type) values (1112,'jamuna','Large')  
insert into parking_space(space_id,space_adress,space_type) values (1113,'pink city','Large')  
insert into parking_space(space_id,space_adress,space_type) values (1114,'police plaza','Medium')  
select * from parking_space  
  
Results Explain Describe Saved SQL History  
SPACE_ID SPACE_TYPE SPACE_ADDRESS  
1111 Large Bashundhara  
1112 Large jamuna  
1113 Large pink city  
1114 Medium police plaza  
4 rows returned in 0.00 seconds CSV Export  
Language: en Application Express 21.0.0.39  
Copyright © 1999, 2005, Oracle. All rights reserved.
```

PARKING SPOT TABLE:

```
insert into parking_spot (spot_id,spot_type,space_id) values(02,'small',1111)
insert into parking_spot (spot_id,spot_type,space_id) values(08,'medium',1111)
insert into parking_spot (spot_id,spot_type,space_id) values(15,'large',1111)
insert into parking_spot (spot_id,spot_type,space_id) values(17,'large',1111)
select * from parking_spot
```



The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window displays the following SQL code:

```
insert into parking_spot3 (spot_id,spot_type) values(02,'small')
insert into parking_spot3 (spot_id,spot_type) values(08,'medium')
insert into parking_spot3 (spot_id,spot_type) values(15,'large')
insert into parking_spot3 (spot_id,spot_type) values(17,'large')
select * from parking_spot3
```

The results window shows the following table output:

SPOT_ID	SPOT_TYPE
2	small
8	medium
15	large
17	large

4 rows returned in 0.00 seconds [CSV Export](#)

Language: en Application Express 21.0.0.39
Copyright © 1995, 2005, Oracle. All rights reserved.

MANAGER 2 TABLE:

```
insert into manager2(mgr_id, name, phone_number, salary, role, space_id) values
(10, 'Diana Lorentz', 5904235567, 42000, 'genral manager', 1111)

insert into manager2(mgr_id, name, phone_number, salary, role, space_id) values
(11, 'John Chen', 515124426, 82000, 'acounts manager', 1111)

insert into manager2(mgr_id, name, phone_number, salary, role, space_id) values
(12, 'Ismael Sciarra', 5151244369, 77000, 'staff manager', 1111)

insert into manager2(mgr_id, name, phone_number, salary, role, space_id) values
(13, 'Jose Manuel Urman', 5151244469, 78000, 'parking lot manager', 1111)

insert into manager2(mgr_id, name, phone_number, salary, role, space_id) values
(30, 'hero alom', 5904235567, 42000, 'genral manager', 1112)

insert into manager2(mgr_id, name, phone_number, salary, role, space_id) values
(31, 'anonto jalil', 5904235567, 42000, 'genral manager', 1113)

insert into manager2(mgr_id, name, phone_number, salary, role, space_id) values
(32, 'pori moni', 5904235567, 42000, 'genral manager', 1114)
```

```
select * from manager2
```

```
insert into manager2(mgr_id, name, phone_number, salary, role, space_id) values (10, 'Diana Lorentz', 5904235567, 42000, 'genral manager', 1111)
insert into manager2(mgr_id, name, phone_number, salary, role, space_id) values (11, 'John Chen', 515124426, 82000, 'acounts manager', 1111)
insert into manager2(mgr_id, name, phone_number, salary, role, space_id) values (12, 'Ismael Sciarra', 5151244369, 77000, 'staff manager', 1111)
insert into manager2(mgr_id, name, phone_number, salary, role, space_id) values (13, 'Jose Manuel Urman', 5151244469, 78000, 'parking lot manager', 1111)
insert into manager2(mgr_id, name, phone_number, salary, role, space_id) values (30, 'hero alom', 5904235567, 42000, 'genral manager', 1112)
insert into manager2(mgr_id, name, phone_number, salary, role, space_id) values (31, 'anonto jalil', 5904235567, 42000, 'genral manager', 1113)
insert into manager2(mgr_id, name, phone_number, salary, role, space_id) values (32, 'pori moni', 5904235567, 42000, 'genral manager', 1114)
```

```
select * from manager2
```

Results Explain Describe Saved SQL History					
MGR_ID	NAME	SALARY	PHONE_NUMBER	ROLE	SPACE_ID
12	Ismael Sciarra	77000	5151244369	staff manager	1111
13	Jose Manuel Urman	78000	5151244469	parking lot manager	1111
30	hero alom	42000	5904235567	genral manager	1112
31	anonto jalil	42000	5904235567	genral manager	1113
32	pori moni	42000	5904235567	genral manager	1114
10	Diana Lorentz	42000	5904235567	genral manager	1111
11	John Chen	82000	515124426	acounts manager	1111

7 rows returned in 0.00 seconds

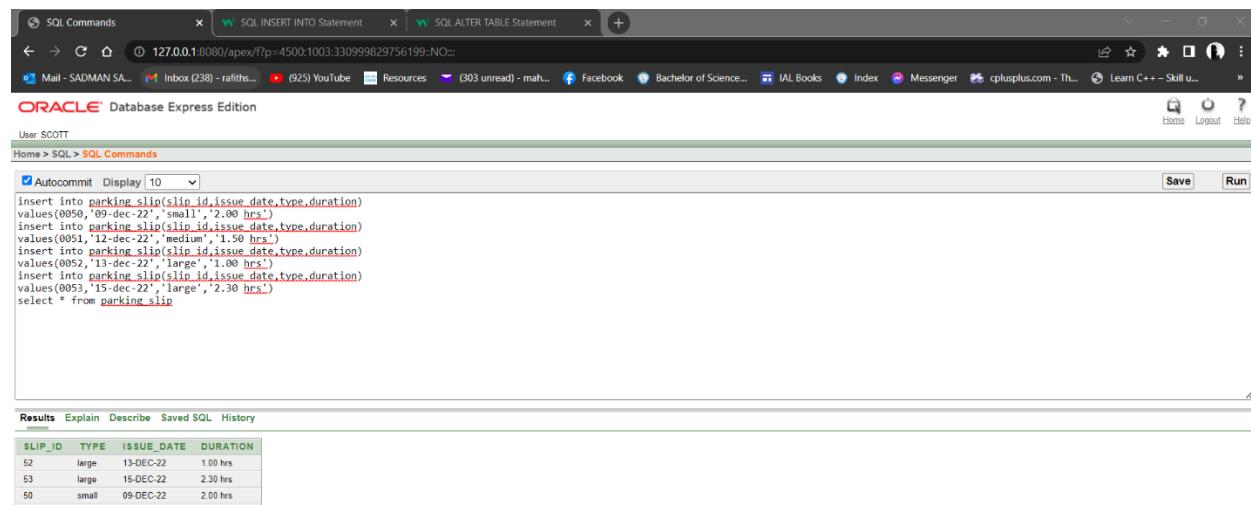
[CSV Export](#)

Application Express 2.1.0.00.39
Copyright © 1999, 2006, Oracle. All rights reserved.



PARKING SLIP TABLE:

```
insert into parking_slip(slip_id,issue_date,type,duration)
values(0050,'09-dec-22','small','2.00 hrs')
insert into parking_slip(slip_id,issue_date,type,duration)
values(0051,'12-dec-22','medium','1.50 hrs')
insert into parking_slip(slip_id,issue_date,type,duration)
values(0052,'13-dec-22','large','1.00 hrs')
insert into parking_slip(slip_id,issue_date,type,duration)
values(0053,'15-dec-22','large','2.30 hrs')
select * from parking_slip
```



The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window contains the provided SQL script. The Results tab displays the output of the 'select * from parking_slip' command, showing the following data:

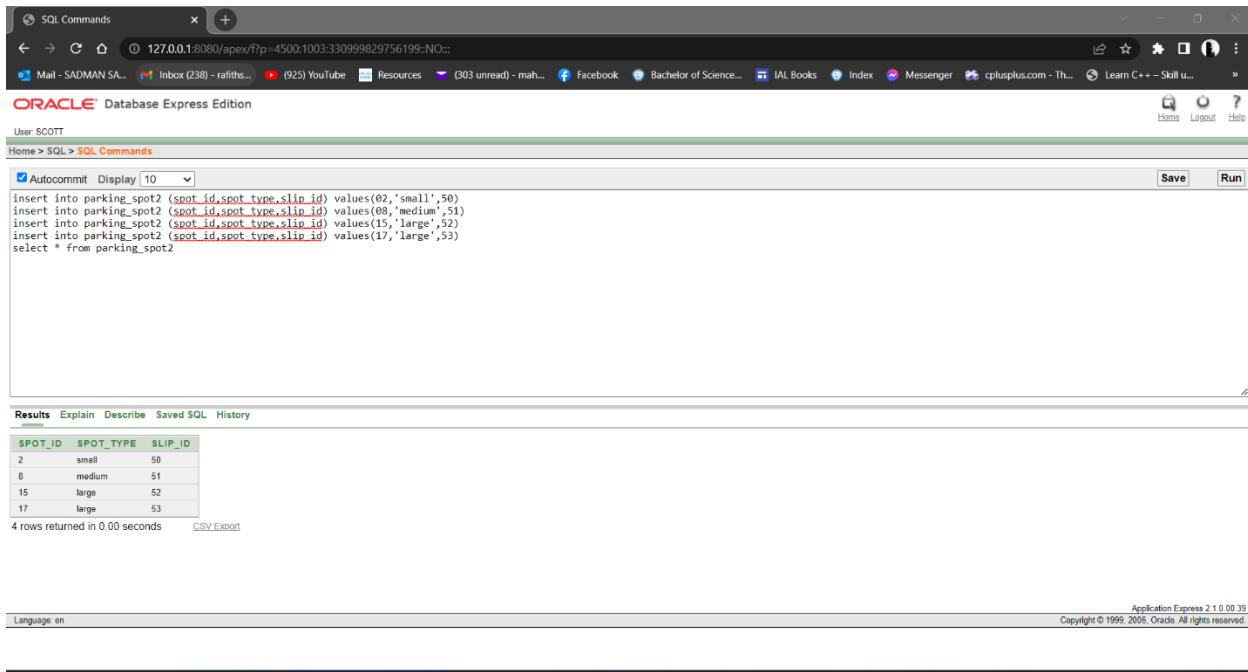
SLIP_ID	TYPE	ISSUE_DATE	DURATION
52	large	13-DEC-22	1.00 hrs
53	large	15-DEC-22	2.30 hrs
50	small	09-DEC-22	2.00 hrs
51	medium	12-DEC-22	1.50 hrs

4 rows returned in 0.00 seconds



PARKING SPOT 2 TABLE:

```
insert into parking_spot2 (spot_id,spot_type,slip_id) values(02,'small',50)
insert into parking_spot2 (spot_id,spot_type,slip_id) values(08,'medium',51)
insert into parking_spot2 (spot_id,spot_type,slip_id) values(15,'large',52)
insert into parking_spot2 (spot_id,spot_type,slip_id) values(17,'large',53)
select * from parking_spot2
```



The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window displays the following SQL code:

```
insert into parking_spot2 (spot_id,spot_type,slip_id) values(02,'small',50)
insert into parking_spot2 (spot_id,spot_type,slip_id) values(08,'medium',51)
insert into parking_spot2 (spot_id,spot_type,slip_id) values(15,'large',52)
insert into parking_spot2 (spot_id,spot_type,slip_id) values(17,'large',53)
select * from parking_spot2
```

The results pane shows the data inserted into the table:

SPOT_ID	SPOT_TYPE	SLIP_ID
2	small	50
8	medium	51
15	large	52
17	large	53

4 rows returned in 0.00 seconds

At the bottom, the status bar shows: Language: en Application Express 2.1.0.0.39 Copyright © 1995, 2005, Oracle. All rights reserved.

DRIVER TABLE:

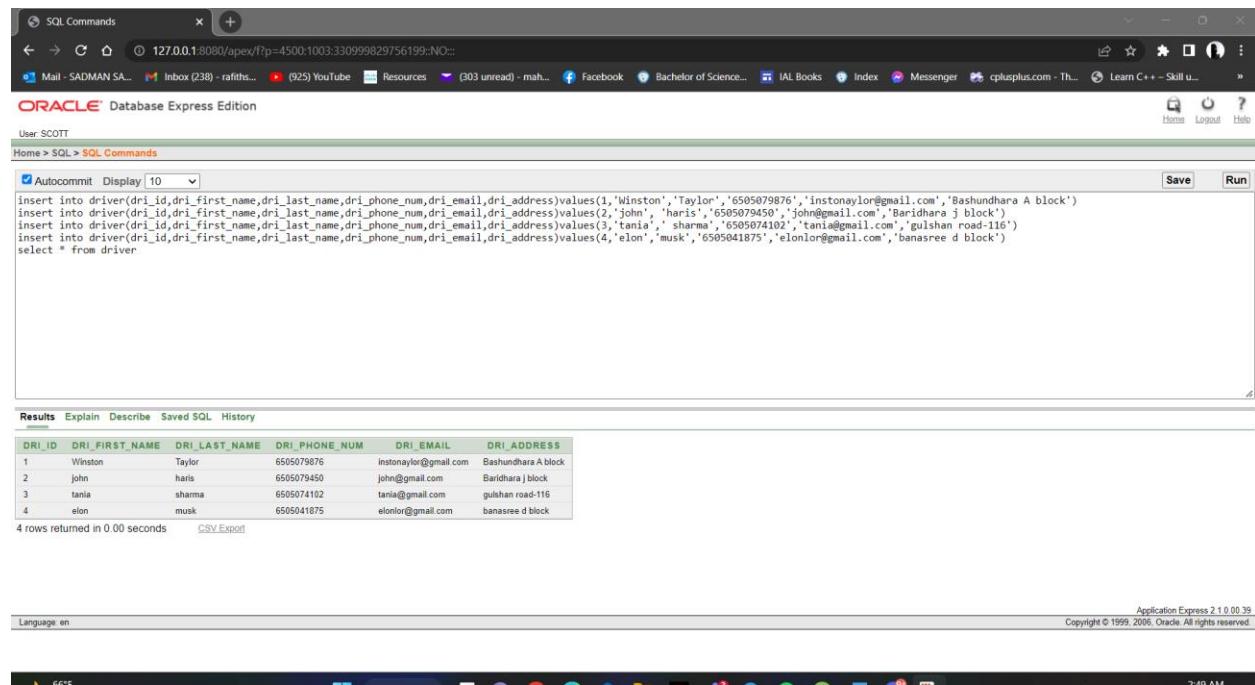
```
insert into
driver(dri_id,dri_first_name,dri_last_name,dri_phone_num,dri_email,dri_address)values(1,'Winston','Taylor','6505079876','instonaylor@gmail.com','Bashundhara A block')

insert into
driver(dri_id,dri_first_name,dri_last_name,dri_phone_num,dri_email,dri_address)values(2,'john', 'haris','6505079450','john@gmail.com','Baridhara j block')

insert into
driver(dri_id,dri_first_name,dri_last_name,dri_phone_num,dri_email,dri_address)values(3,'tania', ' sharma','6505074102','tania@gmail.com','gulshan road-116')

insert into
driver(dri_id,dri_first_name,dri_last_name,dri_phone_num,dri_email,dri_address)values(4,'elon','musk','6505041875','elonlor@gmail.com','banasree d block')

select * from driver
```



The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window displays the following SQL code and its execution results:

```
insert into driver(dri_id,dri_first_name,dri_last_name,dri_phone_num,dri_email,dri_address)values(1,'Winston','Taylor','6505079876','instonaylor@gmail.com','Bashundhara A block')
insert into driver(dri_id,dri_first_name,dri_last_name,dri_phone_num,dri_email,dri_address)values(2,'john', 'haris','6505079450','john@gmail.com','Baridhara j block')
insert into driver(dri_id,dri_first_name,dri_last_name,dri_phone_num,dri_email,dri_address)values(3,'tania', ' sharma','6505074102','tania@gmail.com','gulshan road-116')
insert into driver(dri_id,dri_first_name,dri_last_name,dri_phone_num,dri_email,dri_address)values(4,'elon','musk','6505041875','elonlor@gmail.com','banasree d block')
select * from driver
```

The results section shows the following data:

DRI_ID	DRI_FIRST_NAME	DRI_LAST_NAME	DRI_PHONE_NUM	DRI_EMAIL	DRI_ADDRESS
1	Winston	Taylor	6505079876	instonaylor@gmail.com	Bashundhara A block
2	john	haris	6505079450	john@gmail.com	Baridhara j block
3	tania	sharma	6505074102	tania@gmail.com	gulshan road-116
4	elon	musk	6505041875	elonlor@gmail.com	banasree d block

4 rows returned in 0.00 seconds [CSV Export](#)

Application Express 2.1 0.00.39
Copyright © 1995, 2006, Oracle. All rights reserved.

PARKING SLIP2 TABLE:

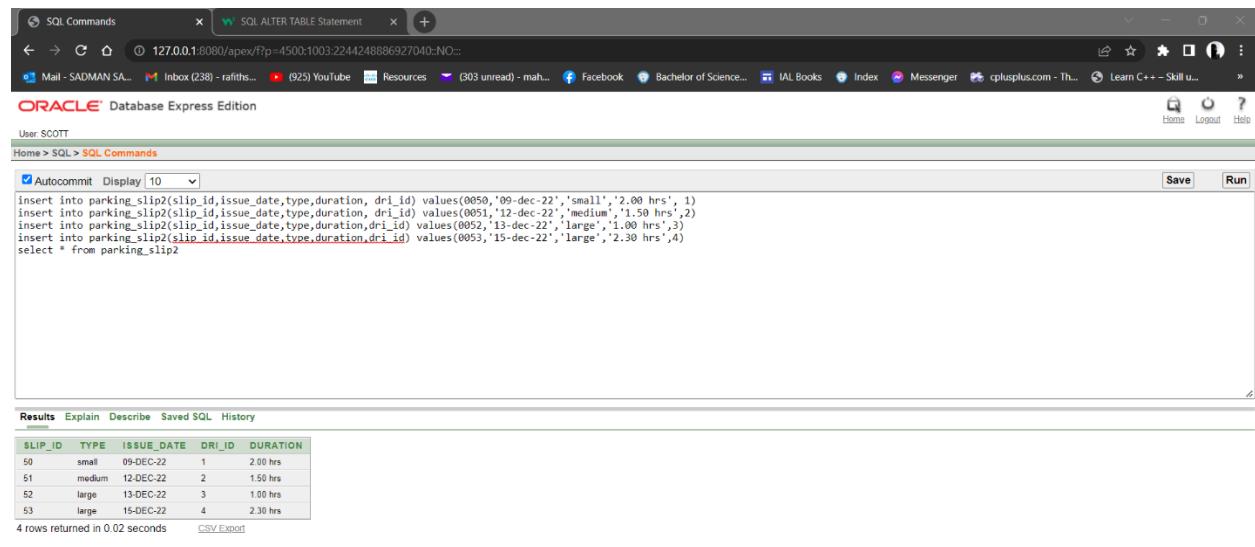
```
insert into parking_slip2(slip_id,issue_date,type,duration, dri_id)
values(0050,'09-dec-22','small','2.00 hrs', 1)
```

```
insert into parking_slip2(slip_id,issue_date,type,duration, dri_id)
values(0051,'12-dec-22','medium','1.50 hrs',2)
```

```
insert into parking_slip2(slip_id,issue_date,type,duration,dri_id)
values(0052,'13-dec-22','large','1.00 hrs',3)
```

```
insert into parking_slip2(slip_id,issue_date,type,duration,dri_id)
values(0053,'15-dec-22','large','2.30 hrs',4)
```

```
select * from parking_slip2
```



The screenshot shows the Oracle Database Express Edition SQL Commands window. The SQL editor contains the following code:

```
insert into parking_slip2(slip_id,issue_date,type,duration, dri_id)
values(0050,'09-dec-22','small','2.00 hrs', 1)
insert into parking_slip2(slip_id,issue_date,type,duration, dri_id)
values(0051,'12-dec-22','medium','1.50 hrs',2)
insert into parking_slip2(slip_id,issue_date,type,duration,dri_id)
values(0052,'13-dec-22','large','1.00 hrs',3)
insert into parking_slip2(slip_id,issue_date,type,duration,dri_id)
values(0053,'15-dec-22','large','2.30 hrs',4)
select * from parking_slip2
```

The results section displays the following table:

SLIP_ID	TYPE	ISSUE_DATE	DRI_ID	DURATION
50	small	09-DEC-22	1	2.00 hrs
51	medium	12-DEC-22	2	1.50 hrs
52	large	13-DEC-22	3	1.00 hrs
53	large	15-DEC-22	4	2.30 hrs

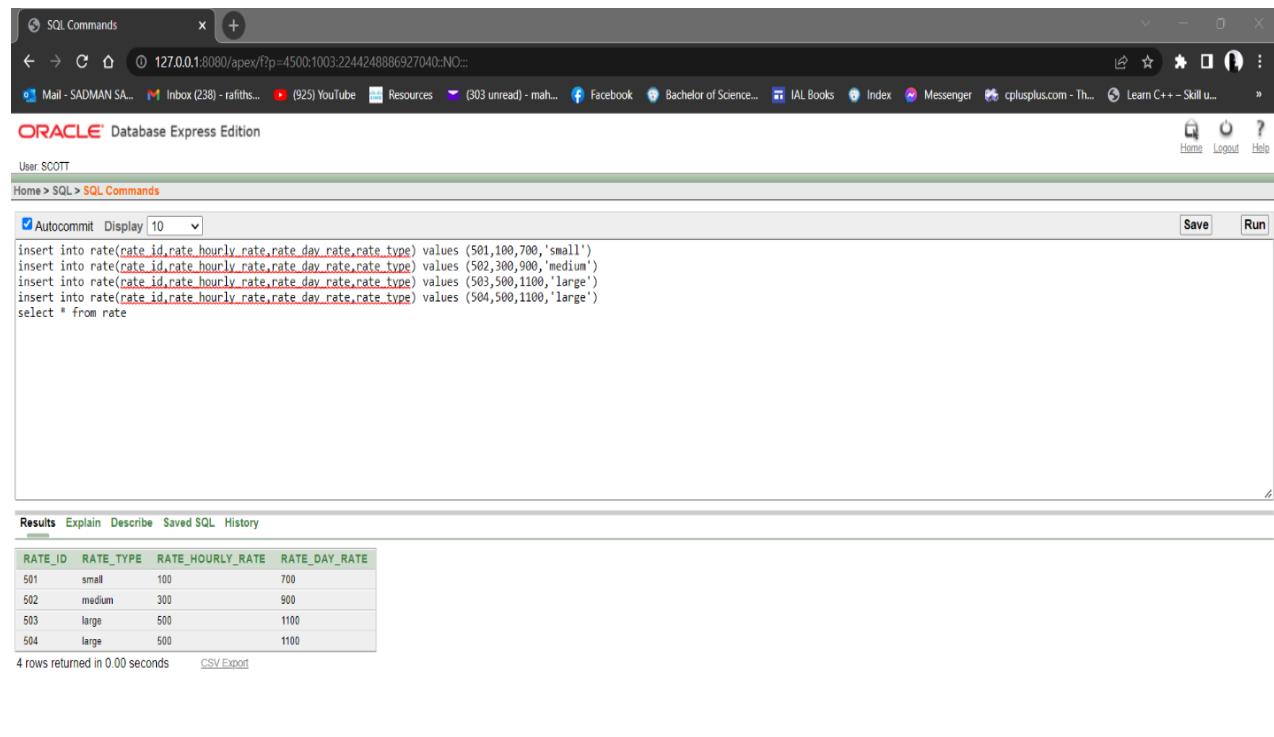
4 rows returned in 0.02 seconds [CSV Export](#)

Language: en Application Express 21.0.0.39
Copyright © 1999, 2005, Oracle. All rights reserved.



RATE_TABLE:

```
insert into rate(rate_id,rate_hourly_rate,rate_day_rate,rate_type) values  
(501,100,700,'small')  
  
insert into rate(rate_id,rate_hourly_rate,rate_day_rate,rate_type) values  
(502,300,900,'medium')  
  
insert into rate(rate_id,rate_hourly_rate,rate_day_rate,rate_type) values  
(503,500,1100,'large')  
  
insert into rate(rate_id,rate_hourly_rate,rate_day_rate,rate_type) values  
(504,500,1100,'large')  
  
select * from rate
```



The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code is entered in the command window, and the results are displayed in a table below. The table has columns: RATE_ID, RATE_TYPE, RATE_HOURLY_RATE, and RATE_DAY_RATE. The data is as follows:

RATE_ID	RATE_TYPE	RATE_HOURLY_RATE	RATE_DAY_RATE
501	small	100	700
502	medium	300	900
503	large	500	1100
504	large	500	1100

4 rows returned in 0.00 seconds [CSV Export](#)



SPOT RATE TABLE:

```
insert into spot_rate(rate_id,spot_id) values (501,2)
insert into spot_rate(rate_id,spot_id) values (502,8)
insert into spot_rate(rate_id,spot_id) values (503,15)
insert into spot_rate(rate_id,spot_id) values (504,17)
select * from spot_rate
```

```
insert into spot_rate(rate_id,spot_id) values (501,2)
insert into spot_rate(rate_id,spot_id) values (502,8)
insert into spot_rate(rate_id,spot_id) values (503,15)
insert into spot_rate(rate_id,spot_id) values (504,17)
select * from spot_rate
```

Results Explain Describe Saved SQL History

RATE_ID	SPOT_ID
501	2
502	8
503	15
504	17

4 rows returned in 0.00 seconds [CSV Export](#)

Application Express 2.1.0.00.39
Copyright © 1999, 2006, Oracle. All rights reserved.



FINE TABLE:

```
create sequence fine_squ
  increment by 5
  start with 100
  maxvalue 500
```

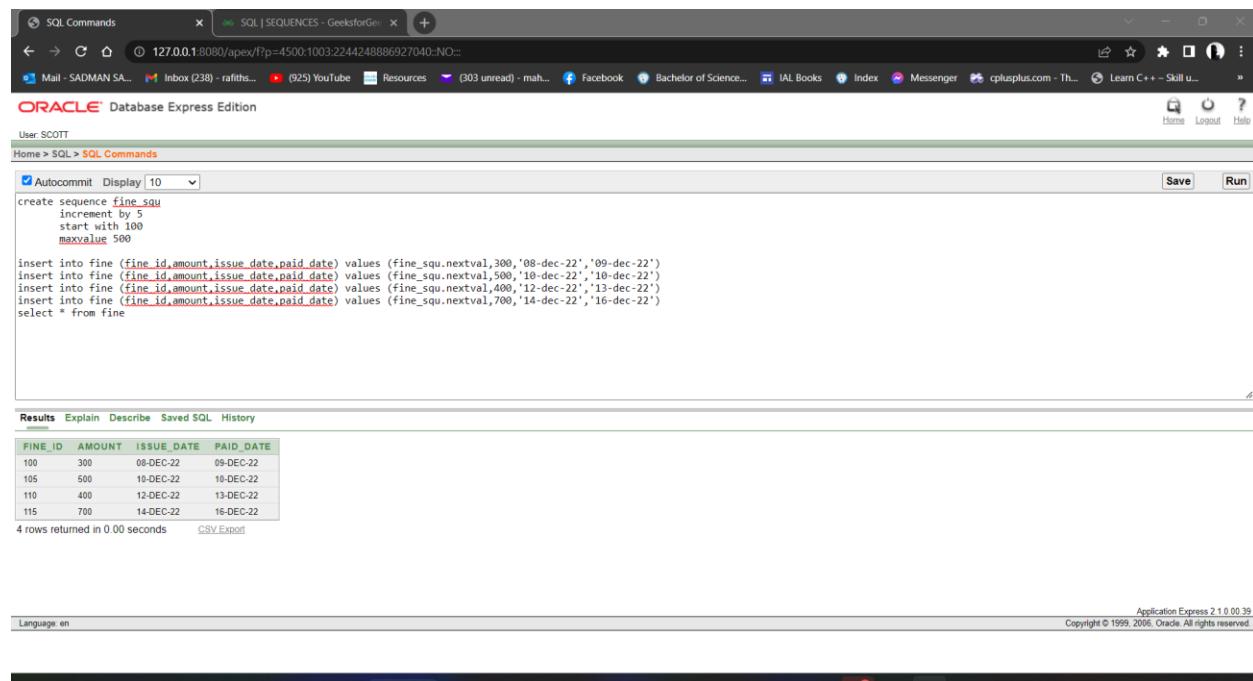
```
insert into fine (fine_id,amount,issue_date,paid_date) values
(fine_squ.nextval,300,'08-dec-22','09-dec-22')
```

```
insert into fine (fine_id,amount,issue_date,paid_date) values
(fine_squ.nextval,500,'10-dec-22','10-dec-22')
```

```
insert into fine (fine_id,amount,issue_date,paid_date) values
(fine_squ.nextval,400,'12-dec-22','13-dec-22')
```

```
insert into fine (fine_id,amount,issue_date,paid_date) values
(fine_squ.nextval,700,'14-dec-22','16-dec-22')
```

```
select * from fine
```



The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window displays the following SQL code:

```
create sequence fine_squ
  increment by 5
  start with 100
  maxvalue 500

insert into fine (fine_id,amount,issue_date,paid_date) values (fine_squ.nextval,300,'08-dec-22','09-dec-22')
insert into fine (fine_id,amount,issue_date,paid_date) values (fine_squ.nextval,500,'10-dec-22','10-dec-22')
insert into fine (fine_id,amount,issue_date,paid_date) values (fine_squ.nextval,400,'12-dec-22','13-dec-22')
insert into fine (fine_id,amount,issue_date,paid_date) values (fine_squ.nextval,700,'14-dec-22','16-dec-22')
select * from fine
```

The Results tab shows the output of the query:

FINE_ID	AMOUNT	ISSUE_DATE	PAID_DATE
100	300	08-DEC-22	09-DEC-22
105	500	10-DEC-22	10-DEC-22
110	400	12-DEC-22	13-DEC-22
115	700	14-DEC-22	16-DEC-22

4 rows returned in 0.00 seconds [CSV Export](#)

At the bottom, the status bar shows: Application Express 2.1 0.00.39, Copyright © 1995, 2006, Oracle. All rights reserved.

DRIVER2_TABLE:

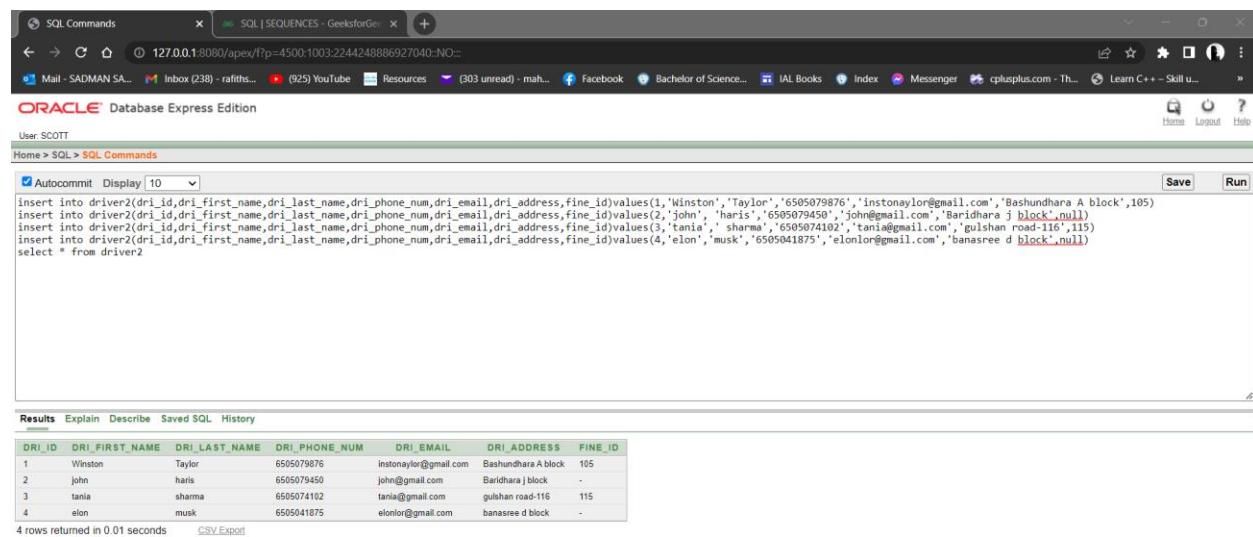
```
insert into
driver2(dri_id,dri_first_name,dri_last_name,dri_phone_num,dri_email,dri_address,fine_id)values(1,'Winston','Taylor','6505079876','instonaylor@gmail.com','
Bashundhara A block',105)

insert into
driver2(dri_id,dri_first_name,dri_last_name,dri_phone_num,dri_email,dri_address,fine_id)values(2,'john', 'haris','6505079450','john@gmail.com','Baridhara j
block',null)

insert into
driver2(dri_id,dri_first_name,dri_last_name,dri_phone_num,dri_email,dri_address,fine_id)values(3,'tania', ' sharma','6505074102','tania@gmail.com','gulshan
road-116',115)

insert into
driver2(dri_id,dri_first_name,dri_last_name,dri_phone_num,dri_email,dri_address,fine_id)values(4,'elon', 'musk','6505041875','elonlor@gmail.com','banasree
d block',null)

select * from driver2
```



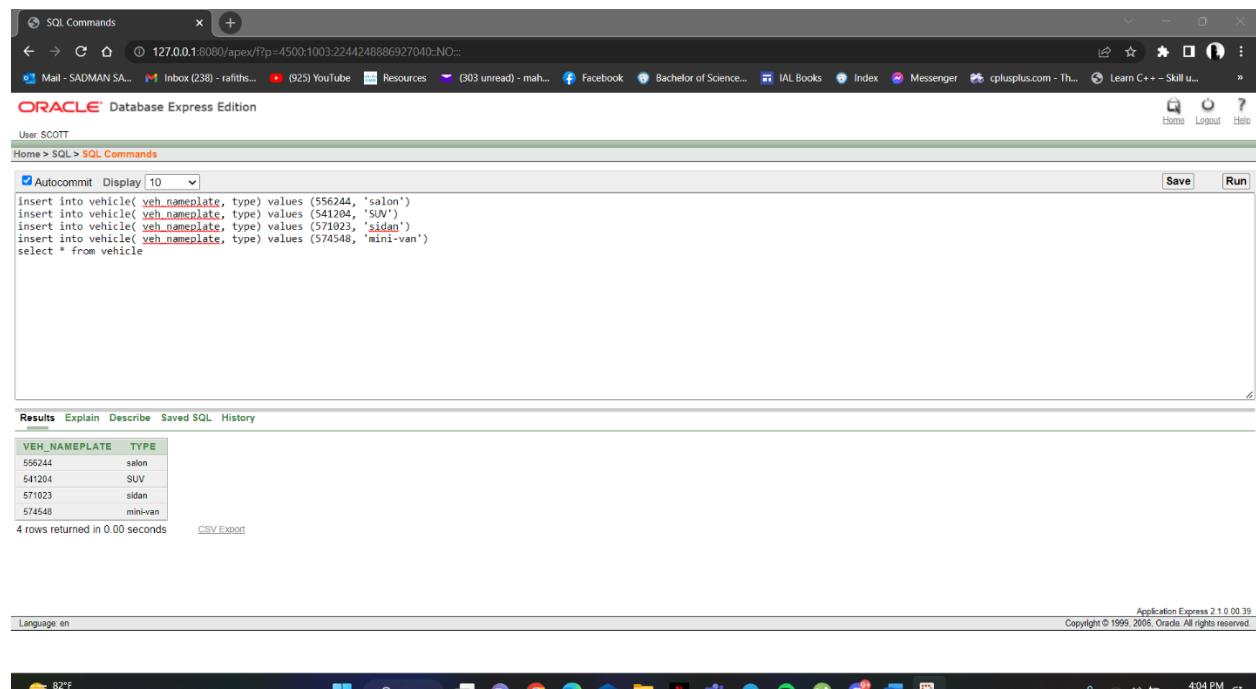
The screenshot shows the Oracle Database Express Edition SQL Commands window. The SQL code is pasted into the command line, and the results are displayed in a table below. The table contains four rows of data, each representing a driver with their details: dri_id, dri_first_name, dri_last_name, dri_phone_num, dri_email, dri_address, and fine_id.

DRI_ID	DRI_FIRST_NAME	DRI_LAST_NAME	DRI_PHONE_NUM	DRI_EMAIL	DRI_ADDRESS	FINE_ID
1	Winston	Taylor	6505079876	instonaylor@gmail.com	Bashundhara A block	105
2	john	haris	6505079450	john@gmail.com	Baridhara j block	-
3	tania	sharma	6505074102	tania@gmail.com	gulshan road-116	115
4	elon	musk	6505041875	elonlor@gmail.com	banasree d block	-

4 rows returned in 0.01 seconds [CSV Export](#)

VEHICLE TABLE:

```
insert into vehicle( veh_nameplate, type) values (556244, 'salon')
insert into vehicle( veh_nameplate, type) values (541204, 'SUV')
insert into vehicle( veh_nameplate, type) values (571023, 'sidan')
insert into vehicle( veh_nameplate, type) values (574548, 'mini-van')
select * from vehicle
```



The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window displays the following SQL code:

```
insert into vehicle( veh_nameplate, type) values (556244, 'salon')
insert into vehicle( veh_nameplate, type) values (541204, 'SUV')
insert into vehicle( veh_nameplate, type) values (571023, 'sidan')
insert into vehicle( veh_nameplate, type) values (574548, 'mini-van')
select * from vehicle
```

The Results tab shows the output of the query:

VEH_NAMEPLATE	TYPE
556244	salon
541204	SUV
571023	sidan
574548	mini-van

4 rows returned in 0.00 seconds

CSV Export

Application Express 2.1.0.00.39
Copyright © 1995, 2005, Oracle. All rights reserved.

DRIVER3_TABLE:

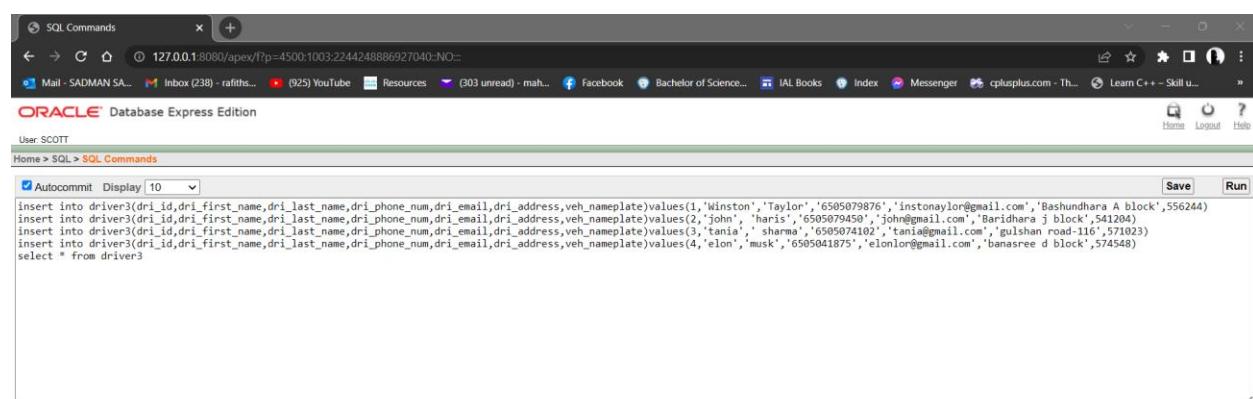
```
insert into
driver3(dri_id,dri_first_name,dri_last_name,dri_phone_num,dri_email,dri_address,veh_nameplate)values(1,'Winston','Taylor','6505079876','instonaylor@gmail.com','Bashundhara A block',556244)

insert into
driver3(dri_id,dri_first_name,dri_last_name,dri_phone_num,dri_email,dri_address,veh_nameplate)values(2,'john',
'haris','6505079450','john@gmail.com','Baridhara j block',541204)

insert into
driver3(dri_id,dri_first_name,dri_last_name,dri_phone_num,dri_email,dri_address,veh_nameplate)values(3,'tania',
'sharma','6505074102','tania@gmail.com','gulshan road-116',571023)

insert into
driver3(dri_id,dri_first_name,dri_last_name,dri_phone_num,dri_email,dri_address,veh_nameplate)values(4,'elon','musk','6505041875','elonlor@gmail.com','banasree d block',574548)

select * from driver3
```



The screenshot shows the Oracle Database Express Edition SQL Commands window. The SQL code entered is:

```
insert into driver3(dri_id,dri_first_name,dri_last_name,dri_phone_num,dri_email,dri_address,veh_nameplate)values(1,'Winston','Taylor','6505079876','instonaylor@gmail.com','Bashundhara A block',556244)
insert into driver3(dri_id,dri_first_name,dri_last_name,dri_phone_num,dri_email,dri_address,veh_nameplate)values(2,'john','haris','6505079450','john@gmail.com','Baridhara j block',541204)
insert into driver3(dri_id,dri_first_name,dri_last_name,dri_phone_num,dri_email,dri_address,veh_nameplate)values(3,'tania','sharma','6505074102','tania@gmail.com','gulshan road-116',571023)
insert into driver3(dri_id,dri_first_name,dri_last_name,dri_phone_num,dri_email,dri_address,veh_nameplate)values(4,'elon','musk','6505041875','elonlor@gmail.com','banasree d block',574548)
select * from driver3
```

The results section shows a table with 4 rows:

DRI_ID	DRI_FIRST_NAME	DRI_LAST_NAME	DRI_PHONE_NUM	DRI_EMAIL	DRI_ADDRESS	VEH_NAMEPLATE
1	Winston	Taylor	6505079876	instonaylor@gmail.com	Bashundhara A block	556244
2	john	haris	6505079450	john@gmail.com	Baridhara j block	541204
4	elon	musk	6505041875	elonlor@gmail.com	banasree d block	574548
3	tania	sharma	6505074102	tania@gmail.com	gulshan road-116	571023

4 rows returned in 0.00 seconds [CSV Export](#)

VEHICLE TABLE:

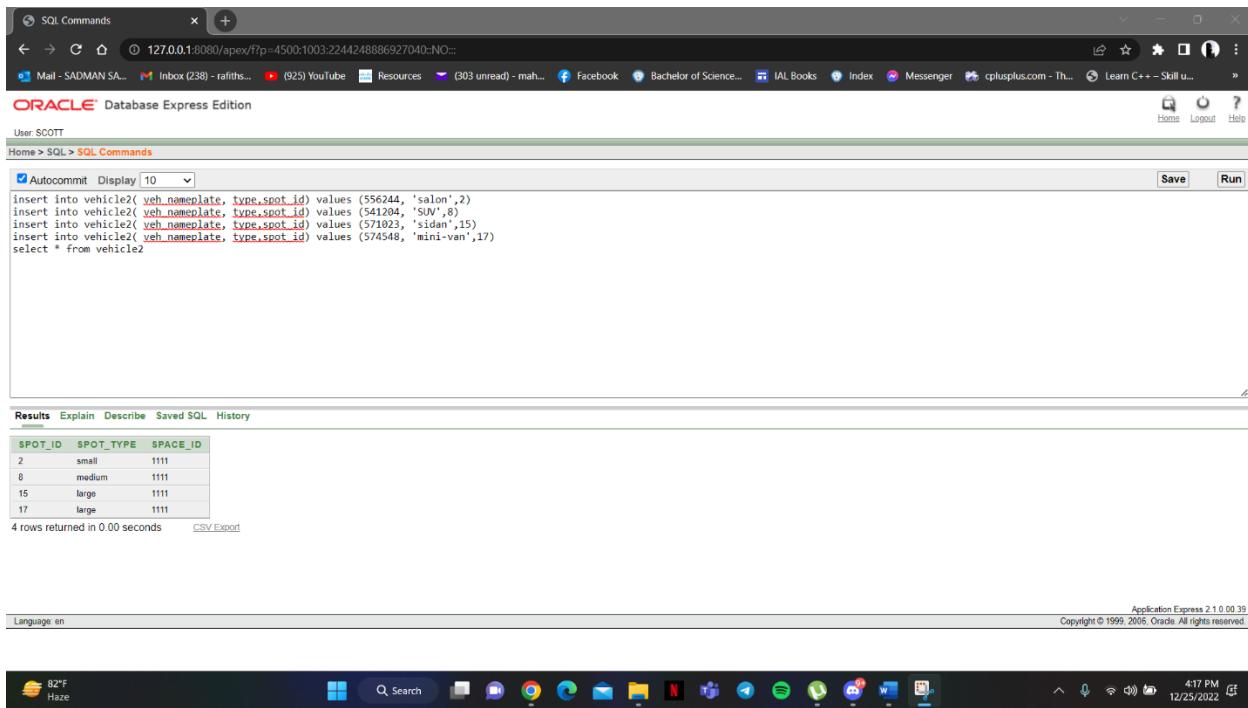
```
insert into vehicle2( veh_nameplate, type,spot_id) values (556244, 'salon',2)
```

```
insert into vehicle2( veh_nameplate, type,spot_id) values (541204, 'SUV',8)
```

```
insert into vehicle2( veh_nameplate, type,spot_id) values (571023, 'sidan',15)
```

```
insert into vehicle2( veh_nameplate, type,spot_id) values (574548, 'mini-van',17)
```

```
select * from vehicle2
```



The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL command window contains the following SQL statements:

```
insert into vehicle2( veh_nameplate, type,spot_id) values (556244, 'salon',2)
insert into vehicle2( veh_nameplate, type,spot_id) values (541204, 'SUV',8)
insert into vehicle2( veh_nameplate, type,spot_id) values (571023, 'sidan',15)
insert into vehicle2( veh_nameplate, type,spot_id) values (574548, 'mini-van',17)
select * from vehicle2
```

The results tab shows a table with the following data:

SPOT_ID	SPOT_TYPE	SPACE_ID
2	small	1111
8	medium	1111
15	large	1111
17	large	1111

4 rows returned in 0.00 seconds

CSV Export

Application Express 2.1.0.0.39
Copyright © 1999, 2006, Oracle. All rights reserved.

FINE 2 TABLE:

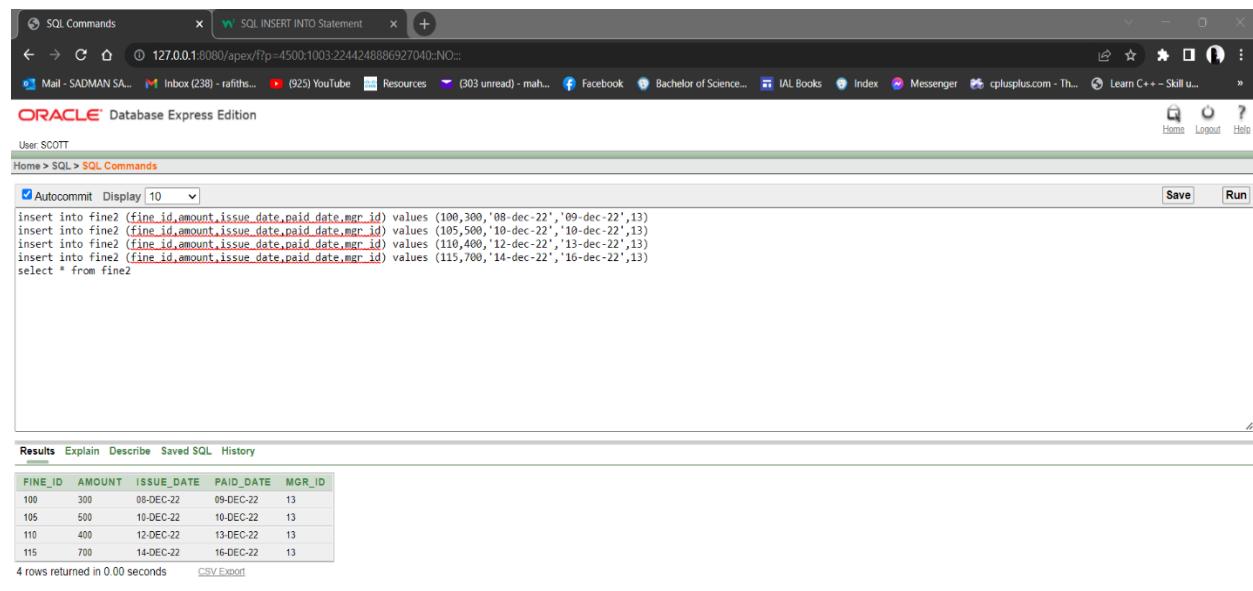
```
insert into fine2 (fine_id,amount,issue_date,paid_date,mgr_id) values  
(100,300,'08-dec-22','09-dec-22',13)
```

```
insert into fine2 (fine_id,amount,issue_date,paid_date,mgr_id) values  
(105,500,'10-dec-22','10-dec-22',13)
```

```
insert into fine2 (fine_id,amount,issue_date,paid_date,mgr_id) values  
(110,400,'12-dec-22','13-dec-22',13)
```

```
insert into fine2 (fine_id,amount,issue_date,paid_date,mgr_id) values  
(115,700,'14-dec-22','16-dec-22',13)
```

```
select * from fine2
```



The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window is open, displaying the following SQL statements:

```
insert into fine2 (fine_id,amount,issue_date,paid_date,mgr_id) values (100,300,'08-dec-22','09-dec-22',13)  
insert into fine2 (fine_id,amount,issue_date,paid_date,mgr_id) values (105,500,'10-dec-22','10-dec-22',13)  
insert into fine2 (fine_id,amount,issue_date,paid_date,mgr_id) values (110,400,'12-dec-22','13-dec-22',13)  
insert into fine2 (fine_id,amount,issue_date,paid_date,mgr_id) values (115,700,'14-dec-22','16-dec-22',13)  
select * from fine2
```

The Results tab is selected, showing the following data:

FINE_ID	AMOUNT	ISSUE_DATE	PAID_DATE	MGR_ID
100	300	08-DEC-22	09-DEC-22	13
105	500	10-DEC-22	10-DEC-22	13
110	400	12-DEC-22	13-DEC-22	13
115	700	14-DEC-22	16-DEC-22	13

4 rows returned in 0.00 seconds [CSV Export](#)

OWNER TABLE:

```
create sequence owner_seq
  increment by 1
  start with 600
  maxvalue 900

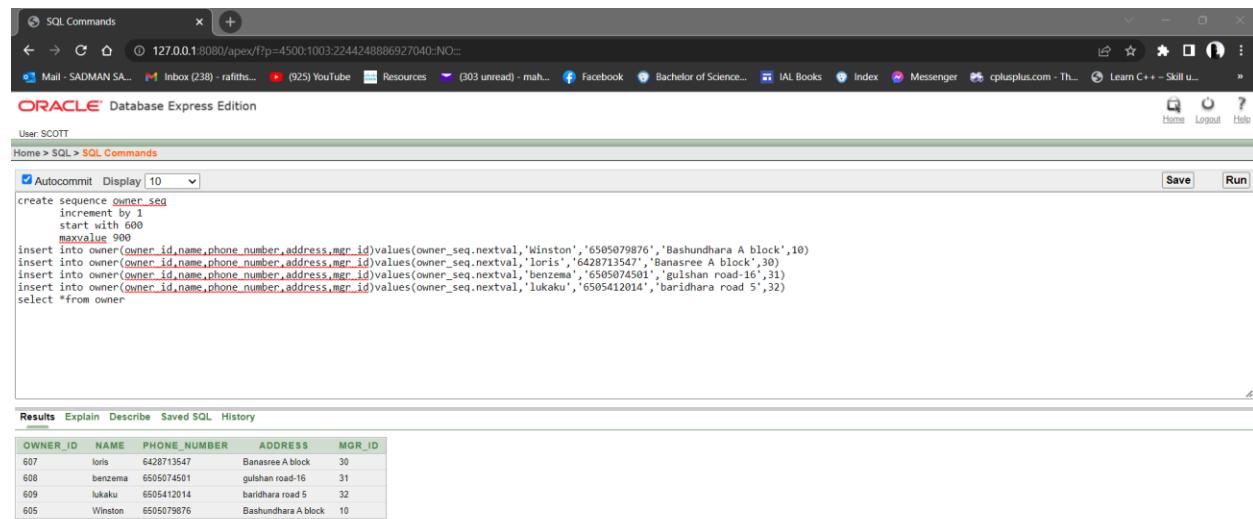
insert into
owner(owner_id,name,phone_number,address,mgr_id)values(owner_seq.nextval,'Winston','6505079876','Bashundhara A block',10)

insert into
owner(owner_id,name,phone_number,address,mgr_id)values(owner_seq.nextval,'loris','6428713547','Banasree A block',30)

insert into
owner(owner_id,name,phone_number,address,mgr_id)values(owner_seq.nextval,'benzema','6505074501','gulshan road-16',31)

insert into
owner(owner_id,name,phone_number,address,mgr_id)values(owner_seq.nextval,'lukaku','6505412014','baridhara road 5',32)

select *from owner
```



The screenshot shows the Oracle Database Express Edition SQL Commands window. The SQL code is pasted into the command line, and the results are displayed in a table below. The table contains four rows of data, each representing an owner with their ID, name, phone number, address, and manager ID.

OWNER_ID	NAME	PHONE_NUMBER	ADDRESS	MGR_ID
607	loris	6428713547	Banasree A block	30
608	benzema	6505074501	gulshan road-16	31
609	lukaku	6505412014	baridhara road 5	32
605	Winston	6505079876	Bashundhara A block	10

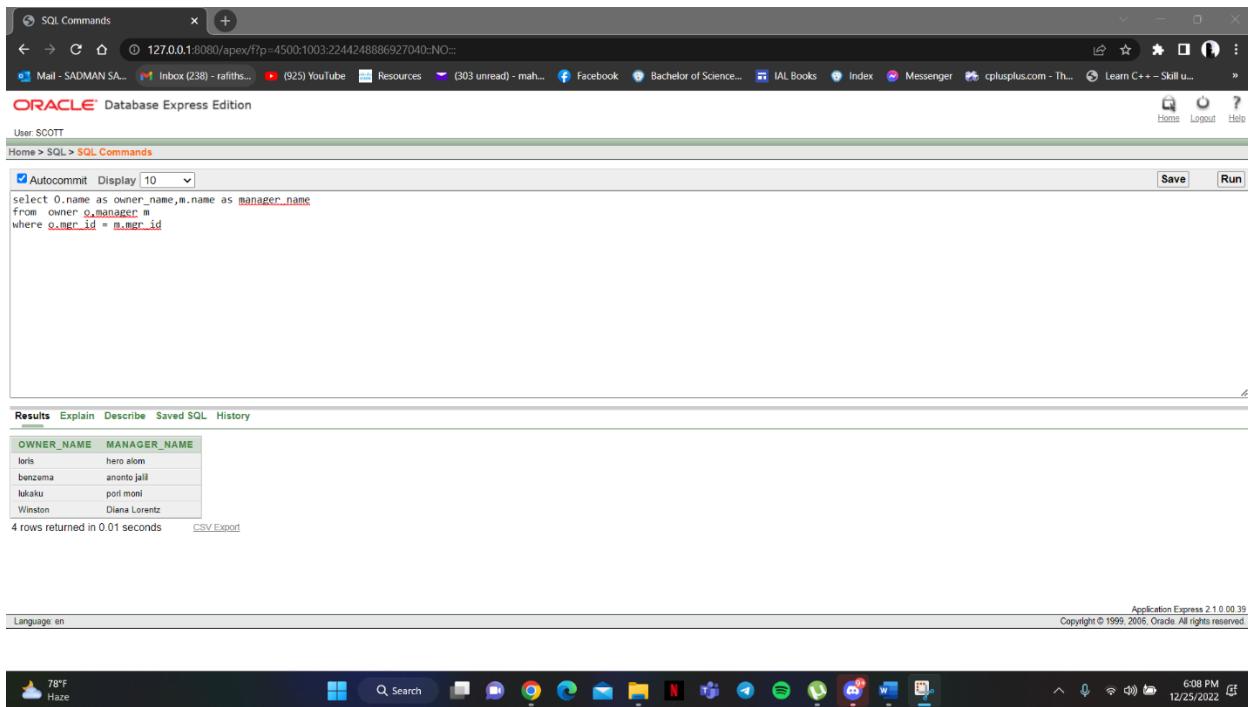


Query

1. Equijoin:

Question: Display the owner name and their manager's name.

```
select O.name as owner_name,m.name as manager_name
from owner o,manager m
where o.mgr_id = m.mgr_id
```



The screenshot shows a Windows desktop with a taskbar at the bottom. The taskbar icons include a weather app (78°F Haze), the Start button, a search bar, File Explorer, a browser, File Explorer, Netflix, Task View, a message center, and system icons for battery, signal, and date/time (6:08 PM, 12/25/2022).

The main window is titled "SQL Commands" and is connected to "127.0.0.1:8080/apex/f?p=4500:1003:2244248886927040:NO::". The Oracle Database Express Edition interface is visible, showing the SQL command:

```
select O.name as owner_name,m.name as manager_name
from owner o,manager m
where o.mgr_id = m.mgr_id
```

The results section displays the following data:

OWNER_NAME	MANAGER_NAME
Ioris	hero alien
benzema	amonto jall
lukaku	porfi more
Winston	Diana Lorentz

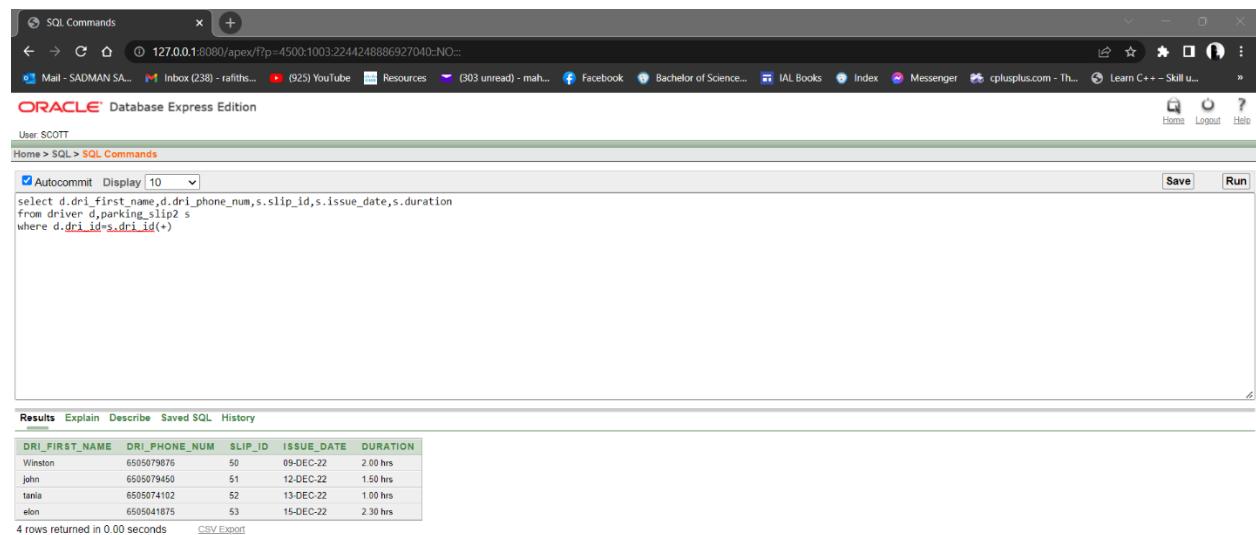
4 rows returned in 0.01 seconds [CSV Export](#)

At the bottom of the window, it says "Language: en" and "Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved."

2. OUTER-JOIN:

Question: Display the first name, and phone number of all drivers who have parked any car.

```
select
d.dri_first_name,d.dri_phone_num,s.slip_id,s.issue_date,s.duration
from driver d,parking_slip2 s
where d.dri_id=s.dri_id(+)
```



The screenshot shows the Oracle Database Express Edition interface. The SQL command is entered in the command window:

```
select d.dri_first_name,d.dri_phone_num,s.slip_id,s.issue_date,s.duration
from driver d,parking_slip2 s
where d.dri_id=s.dri_id(+)
```

The results window displays the following data:

DRI_FIRST_NAME	DRI_PHONE_NUM	SLIP_ID	ISSUE_DATE	DURATION
Winston	6505079876	50	09-DEC-22	2.00 hrs
john	6505079450	51	12-DEC-22	1.50 hrs
tanila	6505074102	52	13-DEC-22	1.00 hrs
elon	6505041875	53	15-DEC-22	2.30 hrs

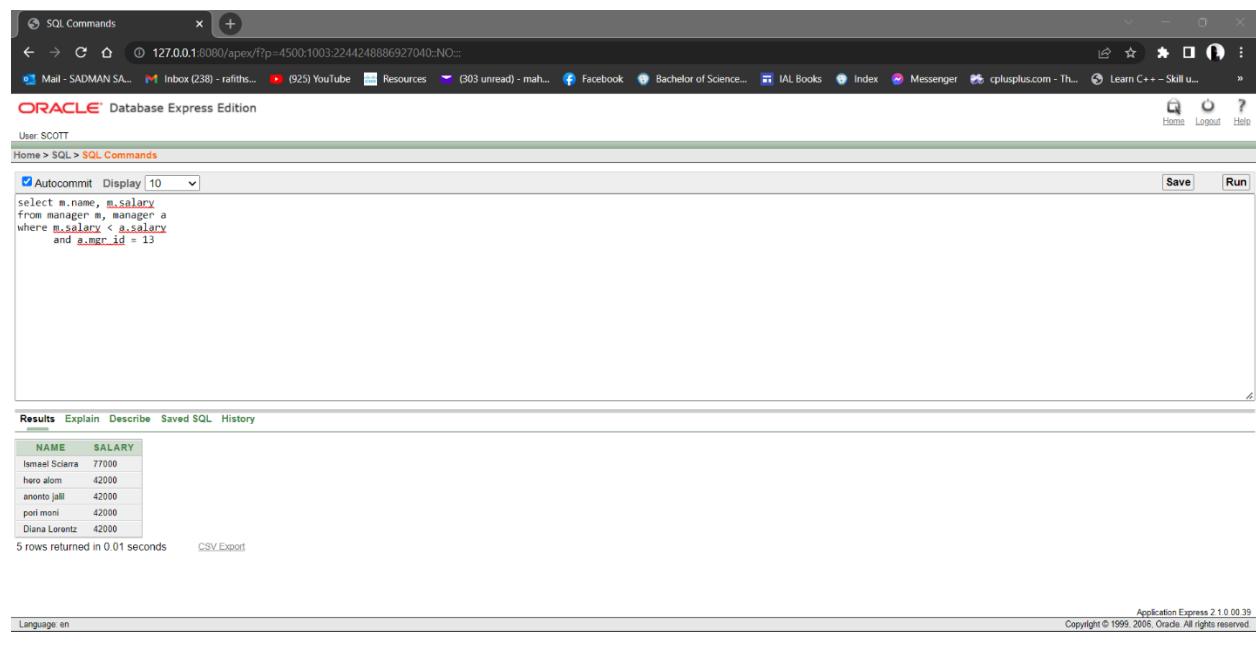
4 rows returned in 0.00 seconds [CSV Export](#)



3. SELF-JOIN:

Question: Display the name and salary of those managers who earn less than the manager earn whose id number is 13.

```
select m.name, m.salary
from manager m, manager a
where m.salary < a.salary
and a.mgr_id = 13
```



The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window displays the following SQL query:

```
select m.name, m.salary
from manager m, manager a
where m.salary < a.salary
and a.mgr_id = 13
```

The results window shows the following data:

NAME	SALARY
Ismael Scialfa	77000
hero alom	42000
anonto jalli	42000
pori moni	42000
Diana Lorentz	42000

5 rows returned in 0.01 seconds

CSV Export

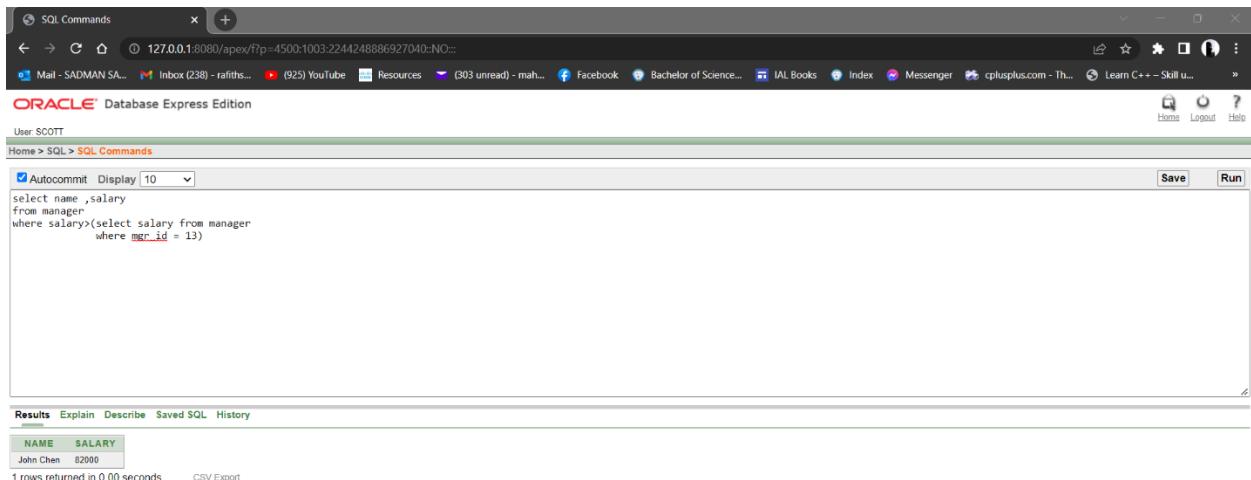
Application Express 2.1.0.00.39
Copyright © 1999, 2006, Oracle. All rights reserved.



SUB-QUERRY (1)

Question: Display the name and salary for all managers who earn more than manager id 13.

```
select name ,salary
from manager
where salary>(select salary from manager
                 where mgr_id = 13)'
```



The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window displays the following SQL query:

```
select name ,salary
from manager
where salary>(select salary from manager
                 where mgr_id = 13)
```

The results pane shows a single row of data:

NAME	SALARY
John Chen	82000

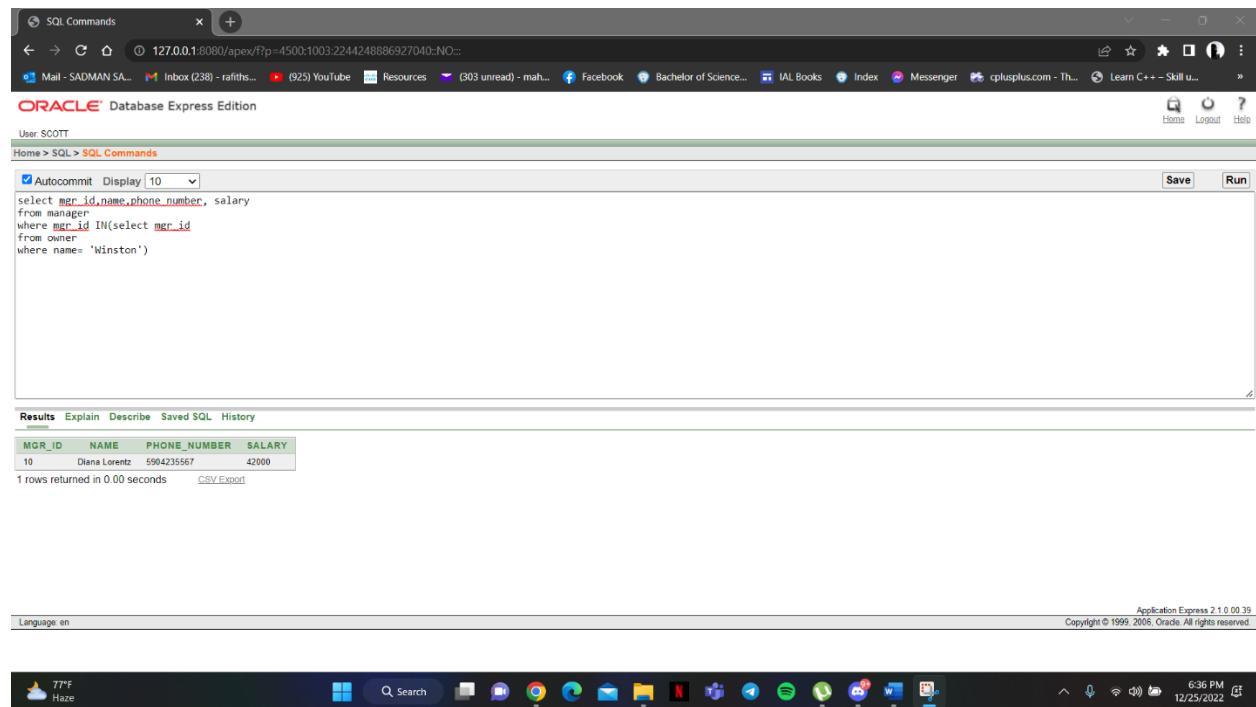
1 rows returned in 0.00 seconds



SUB-QUERRY (2)

Question: Display the id, name, and phone number of owner whose manager name is Winston.

```
select mgr_id, name, phone_number, salary
from manager
where mgr_id IN(select mgr_id
from owner
where name= 'Winston')
```



The screenshot shows a browser window for Oracle Database Express Edition. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:2244248886927040:NO:::1. The page title is "SQL Commands". The SQL command entered is:

```
select mgr_id, name, phone_number, salary
from manager
where mgr_id IN(select mgr_id
from owner
where name= 'Winston')
```

The results table shows one row:

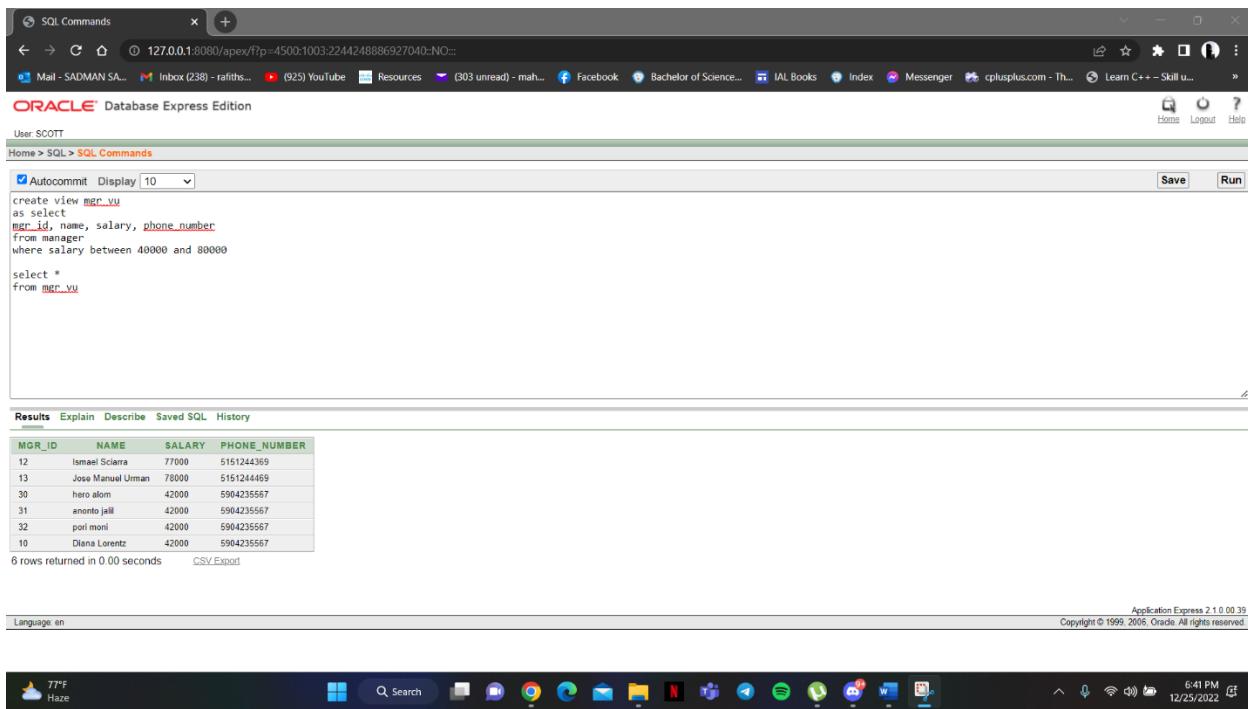
MGR_ID	NAME	PHONE_NUMBER	SALARY
10	Diana Lorentz	5904235567	42000

At the bottom, it says "1 rows returned in 0.00 seconds" and "CSV Export". The status bar at the bottom right shows "Application Express 2.1.0.0.39" and "Copyright © 1995, 2005, Oracle. All rights reserved".

SIMPLE VIEW

Question: create a view name mgr_vu and display id, name, salary, and phone number where the manager salary is between 40000 to 80000.

```
create view mgr_vu
as select
mgr_id, name, salary, phone_number
from manager
where salary between 40000 and 80000
select *
from mgr_vu
```



The screenshot shows a SQL Commands window in Oracle Database Express Edition. The SQL code entered is:

```
create view mgr_vu
as select
mgr_id, name, salary, phone_number
from manager
where salary between 40000 and 80000
select *
from mgr_vu
```

The results section displays a table with 6 rows, showing manager details:

MGR_ID	NAME	SALARY	PHONE_NUMBER
12	Ismael Sciarra	77000	5151244369
13	José Manuel Urman	78000	5151244469
30	hero alom	42000	5904235567
31	antonio jall	42000	5904235567
32	port moni	42000	5904235567
10	Diana Lorentz	42000	5904235567

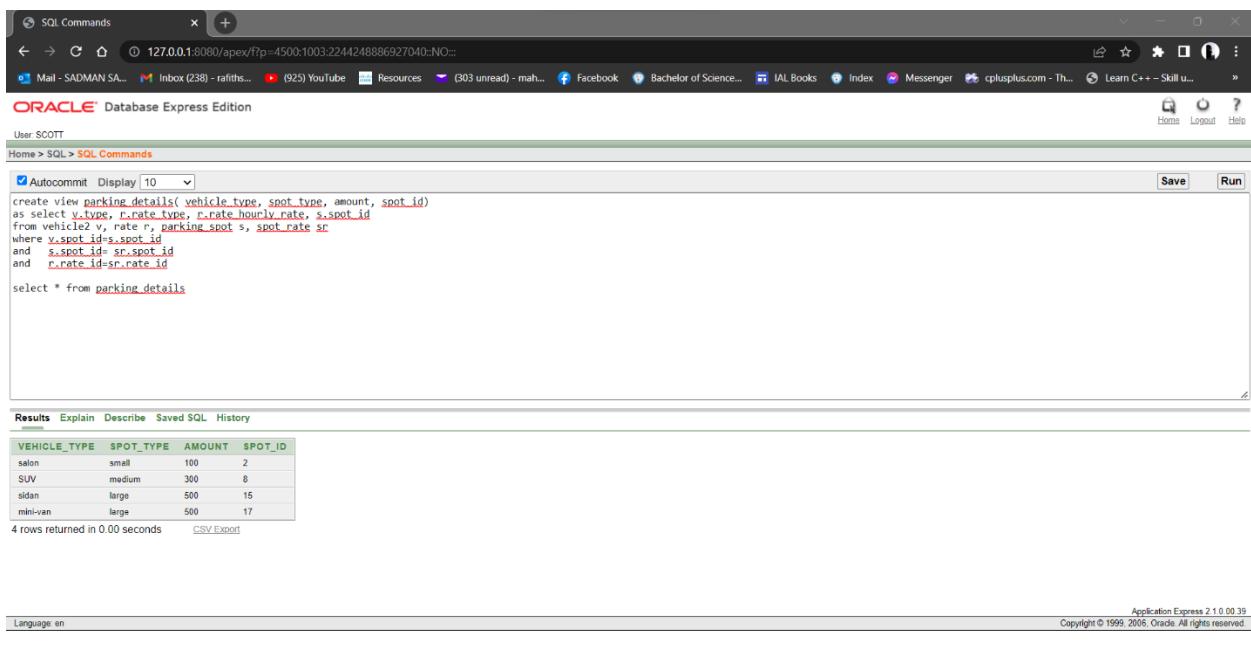
6 rows returned in 0.00 seconds [CSV Export](#)

Application Express 2.1.0.0.39
Copyright © 1996, 2006, Oracle. All rights reserved.

COMPLEX VIEW

```
create view parking_details( vehicle_type, spot_type, amount, spot_id)
as select v.type, r.rate_type, r.rate_hourly_rate, s.spot_id
from vehicle2 v, rate r, parking_spots s, spot_rate sr
where v.spot_id=s.spot_id
and s.spot_id= sr.spot_id
and r.rate_id=sr.rate_id

select * from parking_details
```



The screenshot shows the Oracle Database Express Edition SQL Commands window. The SQL code is as follows:

```
create view parking_details( vehicle_type, spot_type, amount, spot_id)
as select v.type, r.rate_type, r.rate_hourly_rate, s.spot_id
from vehicle2 v, rate r, parking_spots s, spot_rate sr
where v.spot_id=s.spot_id
and s.spot_id= sr.spot_id
and r.rate_id=sr.rate_id

select * from parking_details
```

The results table shows the following data:

VEHICLE_TYPE	SPOT_TYPE	AMOUNT	SPOT_ID
salon	small	100	2
SUV	medium	300	8
sedan	large	500	15
mini-van	large	500	17

4 rows returned in 0.00 seconds [CSV Export](#)

Application Express 2.1.0.00.39
Copyright © 1995, 2005, Oracle. All rights reserved.

Constrain

1. Primary Key

```
create table Manager( MGR_ID number(10) primary key,  
                      Name Varchar2(30),  
                      Salary number(30) not null,  
                      Phone_Number number(10) not null,  
                      Role varchar2(30)  
)
```

```
desc manager
```

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following SQL code is entered and executed:

```
create table manager( MGR_ID number(10) primary key,  
                      Name Varchar2(30),  
                      Salary number(30) not null,  
                      Phone_Number number(10) not null,  
                      Role varchar2(30)  
)  
desc manager
```

The results window displays the description of the Manager table:

Object Type	Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MANAGER		MGR_ID	Number	-	10	0	-	-	-	-
		NAME	Varchar2	30	-	-	-	✓	-	-
		SALARY	Number	-	30	0	-	-	-	-
		PHONE_NUMBER	Number	-	10	0	-	-	-	-
		ROLE	Varchar2	30	-	-	-	✓	-	-

At the bottom of the interface, the system tray shows the date and time as 12/24/2022 10:49 PM.

2. Foreign Key:

```
create table spot_rate( Rate_ID number(10),
                        foreign key(rate_id) references rate(rate_id),
                        spot_id number(10),
                        foreign key(spot_id) references parking_spot(spot_id)
)
```

Desc spot_rate

The screenshot shows the Oracle Application Express interface. At the top, there is a SQL editor window containing the SQL code for creating the SPOT_RATE table. Below the code, the 'Describe' tab is selected, showing the table structure with two columns: RATE_ID and SPOT_ID. The bottom of the window displays the results of the DESCRIBE command, showing the table name and the two columns. The status bar at the bottom right indicates the application version (2.1.0.00.39) and copyright information (Copyright © 1999, 2005, Oracle. All rights reserved.).

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
SPOT_RATE	RATE_ID	Number	-	10	0	-	✓	-	-
SPOT_RATE	SPOT_ID	Number	-	10	0	-	✓	-	-

3. Not Null

```
create table Manager( MGR_ID number(10) primary key,  
                      Name Varchar2(30),  
                      Salary number(30) not null,  
                      Phone_Number number(10) not null,  
                      Role varchar2(30)  
)
```

```
desc manager
```

The screenshot shows the Oracle Database Express Edition SQL Commands window. The SQL command to create the Manager table is entered in the text area. The 'Display' dropdown is set to 10. The 'Run' button is visible. Below the text area, the 'Results' tab is selected, showing the table structure for 'MANAGER'. The table has five columns: MGR_ID, NAME, SALARY, PHONE_NUMBER, and ROLE. MGR_ID is the primary key. The 'Run' button is also visible at the bottom of the results table.

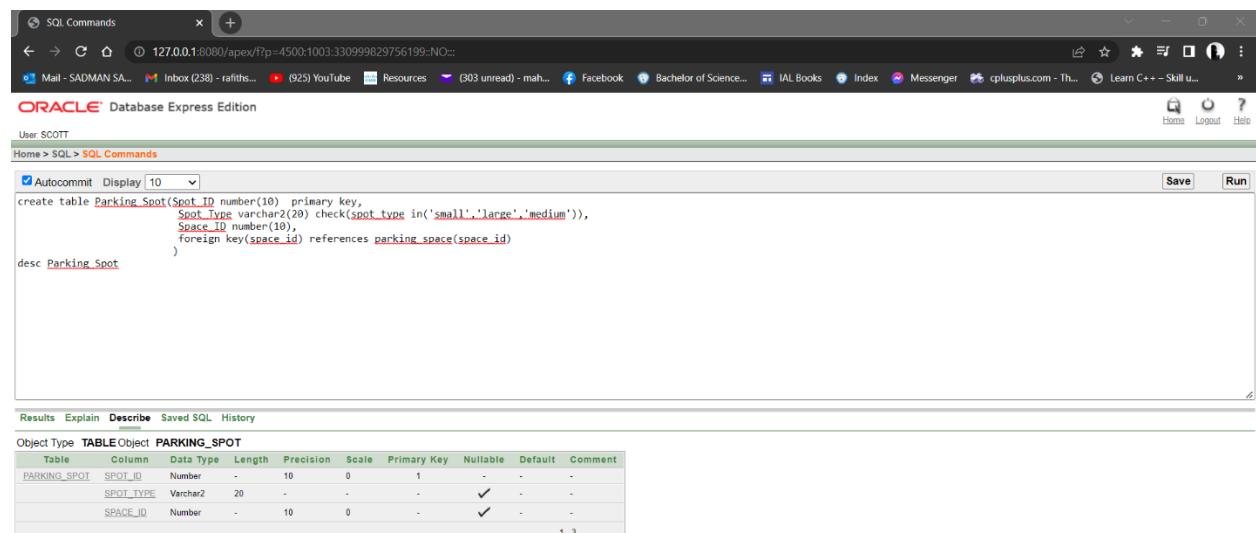
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MANAGER	MGR_ID	Number	-	10	0	1	-	-	-
	NAME	Varchar2	30	-	-	-	✓	-	-
	SALARY	Number	-	30	0	-	-	-	-
	PHONE_NUMBER	Number	-	10	0	-	-	-	-
	ROLE	Varchar2	30	-	-	-	✓	-	-

Language: en Application Express 2.1.0.00.39
Copyright © 1999, 2005, Oracle. All rights reserved.

4. Check:

```
create table Parking_Spot(Spot_ID number(10) primary key,  
                           Spot_Type varchar2(20) check(spot_type  
                           in('small','large','medium')),  
                           Space_ID number(10),  
                           foreign key(space_id) references parking_space(space_id)  
)
```

```
desc Parking_Spot
```



The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL command window contains the creation of the `Parking_Spot` table and its description. The table structure is as follows:

```
create table Parking_Spot(Spot_ID number(10) primary key,  
                           Spot_Type varchar2(20) check(spot_type in('small','large','medium')),  
                           Space_ID number(10),  
                           foreign key(space_id) references parking_space(space_id)  
)
```

The `desc Parking_Spot` command is also present. Below the command window, the object type is shown as `TABLE` and the name is `PARKING_SPOT`. A detailed table structure is displayed:

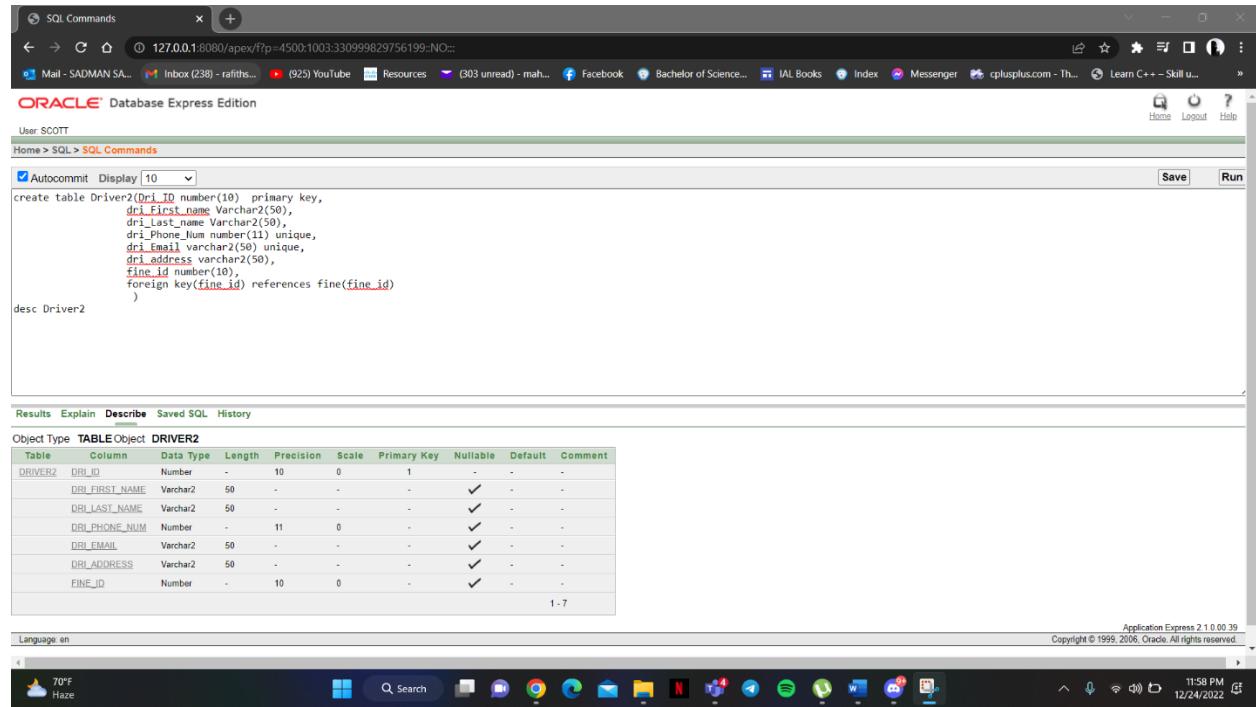
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PARKING_SPOT	SPOT_ID	Number	-	10	0	1	-	-	-
	SPOT_TYPE	Varchar2	20	-	-	-	✓	-	-
	SPACE_ID	Number	-	10	0	-	✓	-	-



5. Unique

```
create table Driver2(Dri_ID number(10) primary key,  
                     dri_First_name Varchar2(50),  
                     dri_Last_name Varchar2(50),  
                     dri_Phone_Num number(11) unique,  
                     dri_Email varchar2(50) unique,  
                     dri_address varchar2(50),  
                     fine_id number(10),  
                     foreign key(fine_id) references fine(fine_id)  
)
```

```
desc Driver2
```



The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window displays the creation of the Driver2 table with unique constraints on the phone number and email fields. The table structure is then described, showing the columns and their properties. The bottom status bar indicates the application version is 21.0.0.39 and the copyright is from 1999-2009 Oracle.

```
create table Driver2(Dri_ID number(10) primary key,  
                     dri_First_name Varchar2(50),  
                     dri_Last_name Varchar2(50),  
                     dri_Phone_Num number(11) unique,  
                     dri_Email varchar2(50) unique,  
                     dri_address varchar2(50),  
                     fine_id number(10),  
                     foreign key(fine_id) references fine(fine_id)  
)  
  
desc Driver2
```

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DRIVER2	DRI_ID	Number	-	10	0	1	-	-	-
DRIVER2	DRI_FIRST_NAME	Varchar2	50	-	-	-	✓	-	-
DRIVER2	DRI_LAST_NAME	Varchar2	50	-	-	-	✓	-	-
DRIVER2	DRI_PHONE_NUM	Number	-	11	0	-	✓	-	-
DRIVER2	DRI_EMAIL	Varchar2	50	-	-	-	✓	-	-
DRIVER2	DRI_ADDRESS	Varchar2	50	-	-	-	✓	-	-
DRIVER2	FINE_ID	Number	-	10	0	-	✓	-	-