

Introduction to Database

Project Name: Apartment Management System

Section: K

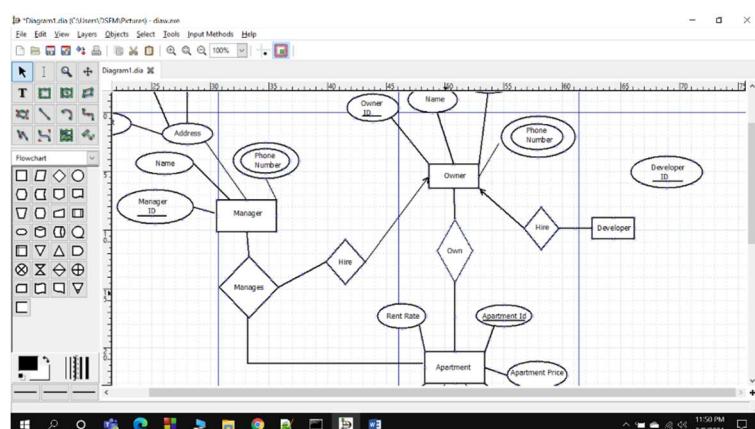
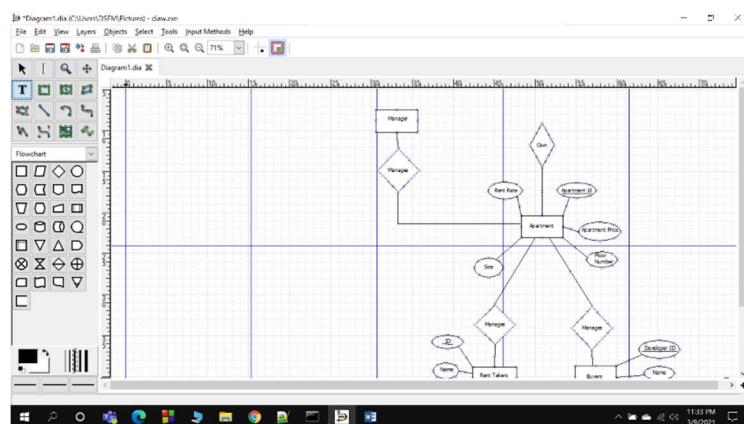
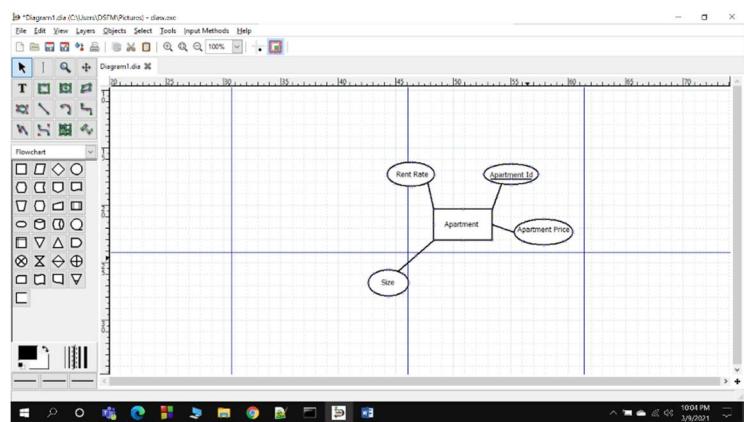
Group Name: Default

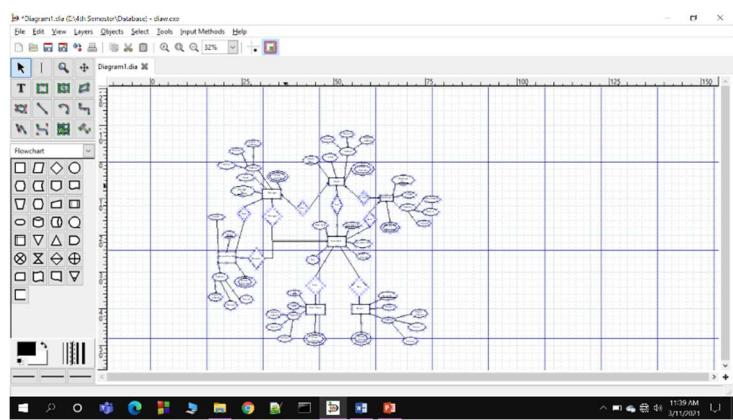
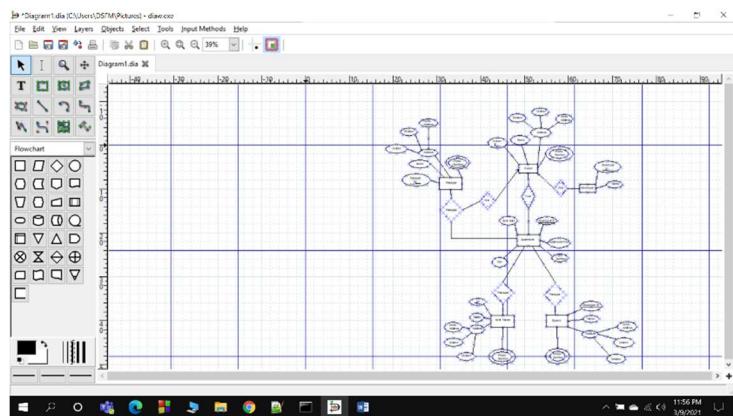
Team Members:

- Sajid Ibna Mahbub (20-42109-1)
- Kazi Maksuda Rana Smrity (20-42351-1)

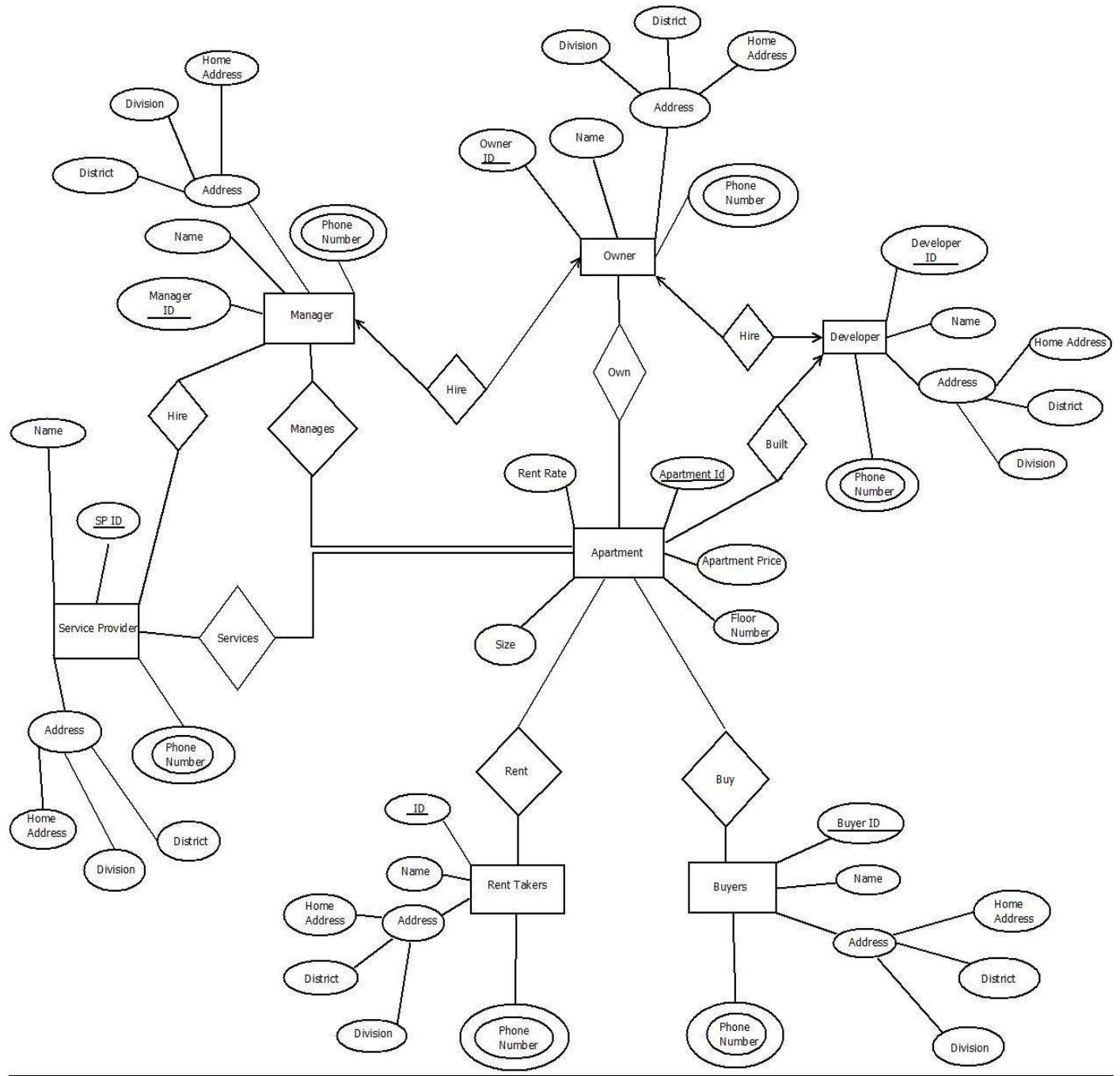
Case Study: In the apartment management system, an owner may recruit a developer for making an apartment. One owner can recruit one developer at a time. On the other hand, a single developer can build many apartments. An owner is identified by a unique owner id. This system also stores owners name, address and phone number. Developer is identified by developer id. Also developer's name, address and phone number is stored. Similarly, apartment price and size is also stored. Owner may sell or rent several apartments for his various purposes. Owner also can recruit only one manager who manages various apartments. Manager is identified by manager id, name, address and phone number which is also stored in the system. Manager contacts various service providers to help maintain the apartment activities. Service provider is identified by his id and also name, phone number and job type. Apartment is rented to renters. In this case. Renter has id, name, address and phone number. Buyers buy apartments. Buyer also had id, name address and phone number stored

Screenshots:





ER Diagram:



Normalization

Owner Hires Developer :

UNF:

1st : Owner_ID, Owner_Name, Owner_District, Owner_Division,
Owner_HomeAddress, Owner_PhoneNumber, Developer_ID, Developer_Name,
Developer_District, Developer_Division, Developer_HomeAddress,
Developer_PhoneNumber

1NF:

1st : Owner_ID, Owner_Name, Owner_District, Owner_Division,
Owner_HomeAddress, Owner_PhoneNumber, Developer_ID, Developer_Name,
Developer_Address, Developer_PhoneNumber

2NF :

1st: Owner_ID, Owner_Name, Owner_District, Owner_Division,
Owner_HomeAddress, Owner_PhoneNumber

2nd : Developer_ID, Developer_Name, Developer_Address,
Developer_PhoneNumber

3NF :

1st : Owner_ID, Owner_Name, Owner_PhoneNumber

2nd : Owner_ID, Owner_District, Owner_Division, Owner_HomeAddress

3rd : Developer_ID, Developer_Name, Developer_Address,
Developer_PhoneNumber

4th : Owner_ID, Developer_ID

Owner Hires Manager :

UNF:

1st : Owner_ID, Owner_Name, Owner_District, Owner_Division,
Owner_HomeAddress, Owner_PhoneNumber, Manager_ID, Manager_Name,
Manager_District, Manager_Division, Manager_HomeAddress,
Manager_PhoneNumber

1NF :

1st : Owner_ID, Owner_Name, Owner_District, Owner_Division,
Owner_HomeAddress, Owner_PhoneNumber, Manager_ID, Manager_Name,
Manager_District, Manager_Division, Manager_HomeAddress,
Manager_PhoneNumber

2NF :

1st : Owner_ID, Owner_Name, Owner_District, Owner_Division,
Owner_HomeAddress, Owner_PhoneNumber

2nd : Manager_ID, Manager_Name, Manager_District, Manager_Division,
Manager_HomeAddress, Manager_PhoneNumber

3NF :

1st : Owner_ID, Owner_Name, Owner_PhoneNumber

2nd : Owner_ID, Owner_District, Owner_Division, Owner_HomeAddress

3rd : Manager_ID, Manager_Name, Manager_PhoneNumber

4th : Manager_ID, Manager_District, Manager_Division, Manager_HomeAddress

5th : Owner_ID, Manager_ID

Owner own Apartment :

UNF :

1st : Owner_ID, Owner_Name, Owner_District, Owner_Division,
Owner_HomeAddress, Owner_PhoneNumber, Apartment_ID, Apartment_Size,
Floor_No, Price, Rent_Rate

1NF :

1st : Owner_ID, Owner_Name, Owner_District, Owner_Division,
Owner_HomeAddress, Owner_PhoneNumber, Apartment_ID, Apartment_Size,
Floor_No, Price, Rent_Rate

2NF :

1st : Owner_ID, Owner_Name, Owner_District, Owner_Division,
Owner_HomeAddress, Owner_PhoneNumber

2nd : Apartment_ID, Apartment_Size, Floor_No, Price, Rent_Rate

3NF :

1st : Owner_ID, Owner_Name, Owner_PhoneNumber

2nd : Owner_ID, Owner_District, Owner_Division, Owner_HomeAddress

3rd : Apartment_ID, Apartment_Size, Floor_No, Price, Rent_Rate

4th : Owner_ID, Apartment_ID

Developer build Apartment :

UNF :

1st : Developer_ID, Developer_Name, Developer_District, Developer_Address, Developer_PhoneNumber, Apartment_ID, Apartment_Size, Floor_No, Price, Rent_Rate

1NF :

1st : Developer_ID, Developer_Name, Developer_District, Developer_Address, Developer_PhoneNumber, Apartment_ID, Apartment_Size, Floor_No, Price, Rent_Rate

2NF :

1st : Developer_ID, Developer_Name, Developer_District, Developer_Address, Developer_PhoneNumber

2nd : Apartment_ID, Apartment_Size, Floor_No, Price, Rent_Rate

3NF :

1st : Developer_ID, Developer_Name, Developer_District, Developer_Address, Developer_PhoneNumber

2nd : Apartment_ID, Apartment_Size, Floor_No, Price, Rent_Rate

3rd : Apartment_ID, Developer_ID

Manager Hire Service Provider :

UNF :

1st : Manager_ID, Manager_Name, Manager_District, Manager_Division, Manager_HomeAddress, Manager_PhoneNumber, SP_ID, SP_Name, SP_District, SP_Division, SP_HomeAddress, SP_PhoneNumber

1NF :

1st : Manager_ID, Manager_Name, Manager_District, Manager_Division, Manager_HomeAddress, Manager_PhoneNumber, SP_ID, SP_Name, SP_District, SP_Division, SP_HomeAddress, SP_PhoneNumber

2NF :

1st : Manager_ID, Manager_Name, Manager_District, Manager_Division, Manager_HomeAddress, Manager_PhoneNumber

2nd : SP_ID, SP_Name, SP_District, SP_Division, SP_HomeAddress, SP_PhoneNumber\

3NF :

1st : Manager_ID, Manager_Name, Manager_District, Manager_Division, Manager_HomeAddress, Manager_PhoneNumber

2nd : Manager_ID, Manager_District, Manager_Division, Manager_HomeAddress

3rd : SP_ID, SP_Name, SP_PhoneNumber

4th : SP_ID, SP_District, SP_Division, SP_HomeAddress

5th : SP_ID, Manager_ID

Manager Manages Apartment :

UNF :

1st : Manager_ID, Manager_Name, Manager_District, Manager_Division, Manager_HomeAddress, Manager_PhoneNumber, SP_ID, SP_Name, SP_District, SP_Division, SP_HomeAddress, SP_PhoneNumber

1NF :

1st : Manager_ID, Manager_Name, Manager_District, Manager_Division, Manager_HomeAddress, Manager_PhoneNumber, Apartment_ID, Apartment_Size, Floor_No, Price, Rent_Rate

2NF :

1st : Manager_ID, Manager_Name, Manager_District, Manager_Division, Manager_HomeAddress, Manager_PhoneNumber

2nd : Apartment_ID, Apartment_Size, Floor_No, Price, Rent_Rate

3NF :

1st : Manager_ID, Manager_Name, Manager_District, Manager_Division, Manager_HomeAddress, Manager_PhoneNumber

2nd : Manager_ID, Manager_District, Manager_Division, Manager_HomeAddress

3rd : Apartment_ID, Apartment_Size, Floor_No, Price, Rent_Rate

4th : Manager_ID, Apartment_ID

Service Provider Serve Services in

Apartments :

UNF :

1st : SP_ID, SP_Name, SP_District, SP_Division, SP_HomeAddress, SP_PhoneNumber, Apartment_ID, Apartment_Size, Floor_No, Price, Rent_Rate

1NF :

1st : SP_ID, SP_Name, SP_District, SP_Division, SP_HomeAddress, SP_PhoneNumber, Apartment_ID, Apartment_Size, Floor_No, Price, Rent_Rate

2NF :

1st : SP_ID, SP_Name, SP_District, SP_Division, SP_HomeAddress, SP_PhoneNumber

2nd : Apartment_ID, Apartment_Size, Floor_No, Price, Rent_Rate

3NF :

1st : SP_ID, SP_Name, SP_PhoneNumber

2nd : SP_ID, SP_Name, SP_District, SP_Division, SP_HomeAddress

3rd : Apartment_ID, Apartment_Size, Floor_No, Price, Rent_Rate

4th : SP_ID, Apartment_ID

Renter Rent Apartment :

UNF :

1st : Renter_ID, Renter_Name, Renter_District, Renter_Division, Renter_HomeAddress, Renter_PhoneNumber, Apartment_ID, Apartment_Size, Floor_No, Price, Rent_Rate

1NF :

1st : Renter_ID, Renter_Name, Renter_District, Renter_Division, Renter_HomeAddress, Renter_PhoneNumber, Apartment_ID, Apartment_Size, Floor_No, Price, Rent_Rate

2NF :

1st : Renter_ID, Renter_Name, Renter_District, Renter_Division, Renter_HomeAddress, Renter_PhoneNumber

2nd : Apartment_ID, Apartment_Size, Floor_No, Price, Rent_Rate

3NF :

1st : Renter_ID, Renter_Name, Renter_District, Renter_Division,
Renter_HomeAddress, Renter_PhoneNumber

2nd : Renter_ID, Renter_District, Renter_Division, Renter_HomeAddress

3rd : Apartment_ID, Apartment_Size, Floor_No, Price, Rent_Rate

4th : Renter_ID, Apartment_ID

Buyer buy Apartment :

UNF :

1st : Buyer_ID, Buyer_Name, Buyer_District, Buyer_Division,
Buyer_HomeAddress, Buyer_PhoneNumber, Apartment_ID, Apartment_Size,
Floor_No, Price, Rent_Rate

1NF :

1st : Buyer_ID, Buyer_Name, Buyer_District, Buyer_Division,
Buyer_HomeAddress, Buyer_PhoneNumber, Apartment_ID, Apartment_Size,
Floor_No, Price, Rent_Rate

2NF :

1st : Buyer_ID, Buyer_Name, Buyer_District, Buyer_Division,
Buyer_HomeAddress, Buyer_PhoneNumber

2nd : Apartment_ID, Apartment_Size, Floor_No, Price, Rent_Rate

3NF :

1st : Buyer_ID, Buyer_Name, Buyer_PhoneNumber

2nd : Buyer_ID, Buyer_District, Buyer_Division, Buyer_HomeAddress

3rd : Apartment_ID, Apartment_Size, Floor_No, Price, Rent_Rate

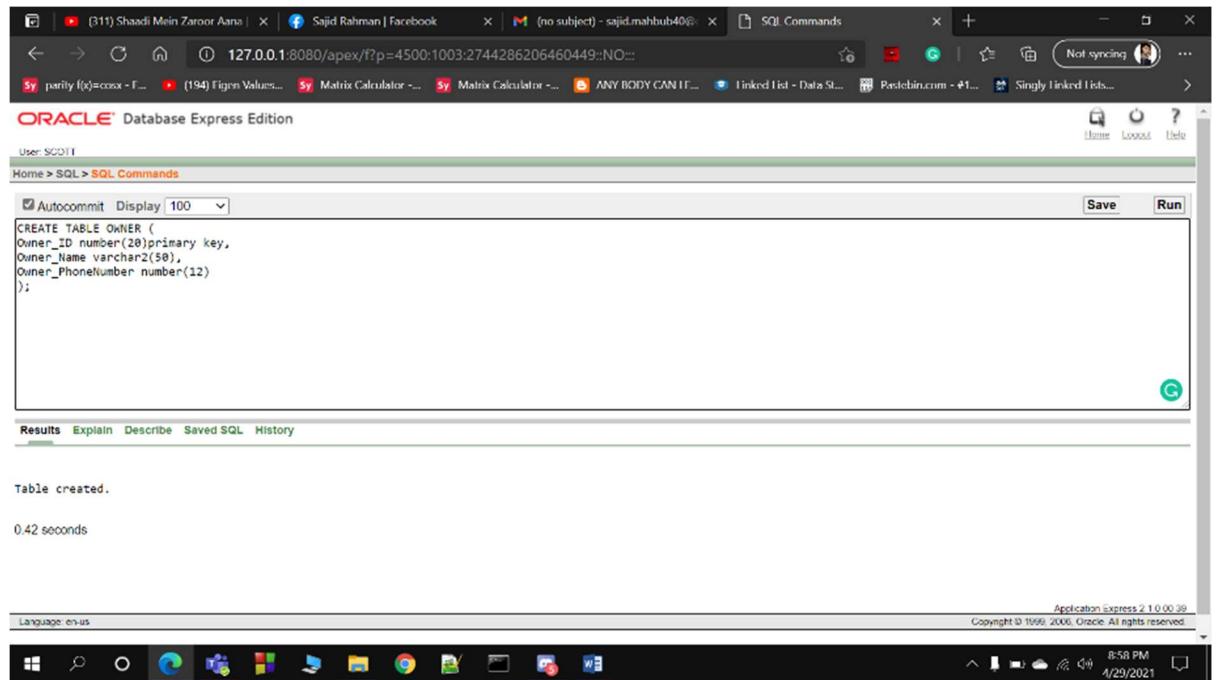
4th : Buyer_ID, Apartment_ID

Table Creation

```

CREATE TABLE OWNER (
    Owner_ID number(20)primary key,
    Owner_Name varchar2(50),
    Owner_PhoneNumber number(12)
);

```



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

```

CREATE TABLE OWNER (
    Owner_ID number(20)primary key,
    Owner_Name varchar2(50),
    Owner_PhoneNumber number(12)
);

```

The results pane shows the message "Table created." and a duration of "0.42 seconds". The bottom status bar indicates the application version is 2.1.0.00.99 and the copyright is from 1996-2006 Oracle.

```

CREATE TABLE Owner_Address (
    Owner_ID number(6) primary key,
    Owner_District varchar2(50),
    Owner_Division varchar2(50),
    Owner_HomeAddress varchar2(50)
);

```

The screenshot shows a Windows desktop environment with multiple browser tabs open. In the foreground, the Oracle Application Express interface is displayed. The user is connected as SCOTT. The SQL Commands page shows the following SQL code:

```
CREATE TABLE OWNER (
  Owner_ID number(20) primary key,
  Owner_Name varchar2(50),
  Owner_PhoneNumber number(12)
);
```

The results pane displays the error message: ORA-00955: name is already used by an existing object. The status bar at the bottom right indicates the application version is 2.1.0.0.39 and the copyright is from 1999-2000, Oracle. The taskbar shows various application icons.

```
CREATE TABLE Developer (
  Developer_ID number(20) primary key,
  Developer_Name varchar2(50),
  Developer_Address varchar2(50),
  Developer_PhoneNumber number(12)
);
```

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

```
CREATE TABLE Developer (
Developer_ID number(20) primary key,
Developer_Name varchar2(50),
Developer_Address varchar2(50),
Developer_PhoneNumber number(12)
);
```

The results show:

Table created.

0.00 seconds

At the bottom, the status bar indicates Application Express 2.1.0.0.39 and Copyright © 1999, 2008, Oracle. All rights reserved.

```
CREATE TABLE Owner_Developer(
Owner_ID number(20),
Developer_ID number(20)
);
```

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

```
CREATE TABLE Owner_Developer(
Owner_ID number(20),
Developer_ID number(20)
);
```

The results show:

Table created.

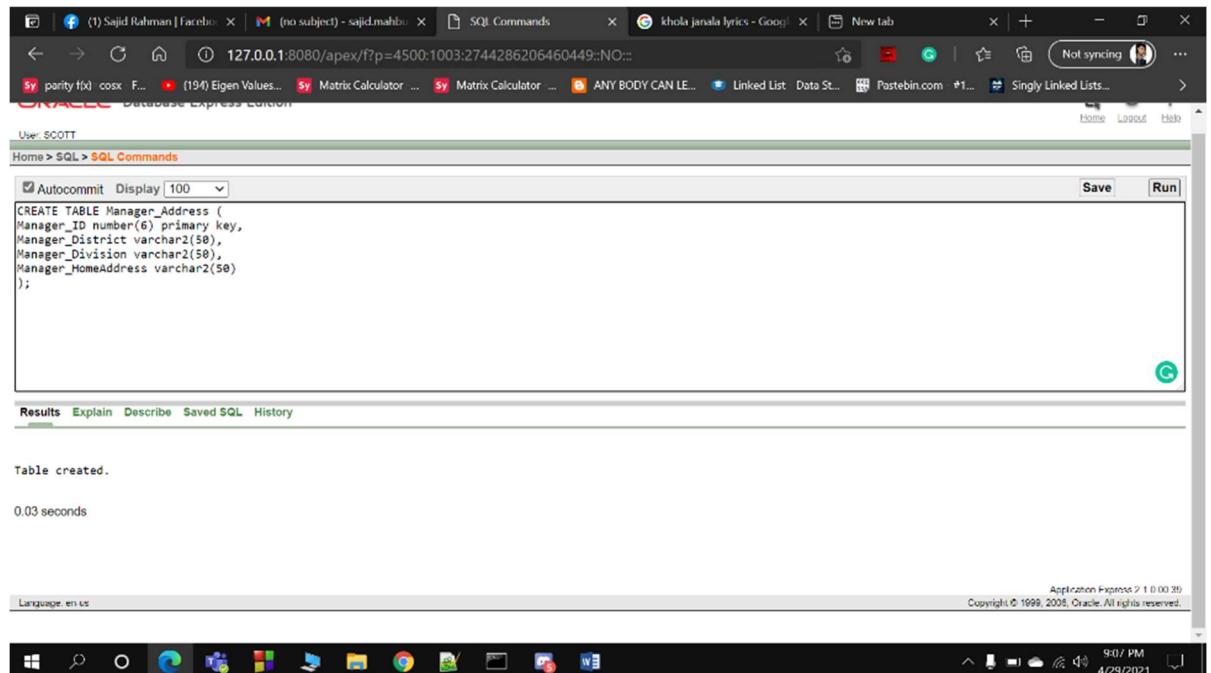
0.08 seconds

At the bottom, the status bar indicates Application Express 2.1.0.0.39 and Copyright © 1999, 2008, Oracle. All rights reserved.

```

CREATE TABLE Manager_Address (
    Manager_ID number(6) primary key,
    Manager_District varchar2(50),
    Manager_Division varchar2(50),
    Manager_HomeAddress varchar2(50)
);

```



The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window is open, displaying the SQL code for creating the Manager_Address table. The table has four columns: Manager_ID (primary key), Manager_District, Manager_Division, and Manager_HomeAddress, all of type varchar2(50). The code is executed successfully, resulting in the message "Table created." The window also shows other tabs like Results, Explain, Describe, Saved SQL, and History.

```

CREATE TABLE Manager_Address (
    Manager_ID number(6) primary key,
    Manager_District varchar2(50),
    Manager_Division varchar2(50),
    Manager_HomeAddress varchar2(50)
);

Table created.

0.03 seconds

```

```

CREATE TABLE Manager (
    Manager_ID number(20) primary key,
    Manager_Name varchar2(50),
    Manager_PhoneNumber number(12)
);

```

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

```
CREATE TABLE Manager (
    Manager_ID number(28) primary key,
    Manager_Name varchar2(58),
    Manager_PhoneNumber number(12)
);
```

The results pane shows the message "Table created." and a duration of "0.03 seconds". The status bar at the bottom right indicates "Application Express 2.1.0.0.39" and "Copyright © 1999, 2006, Oracle. All rights reserved".

```
CREATE TABLE Owner_Manager(
    Owner_ID number(20),
    Manager_ID number(20)
);
```

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

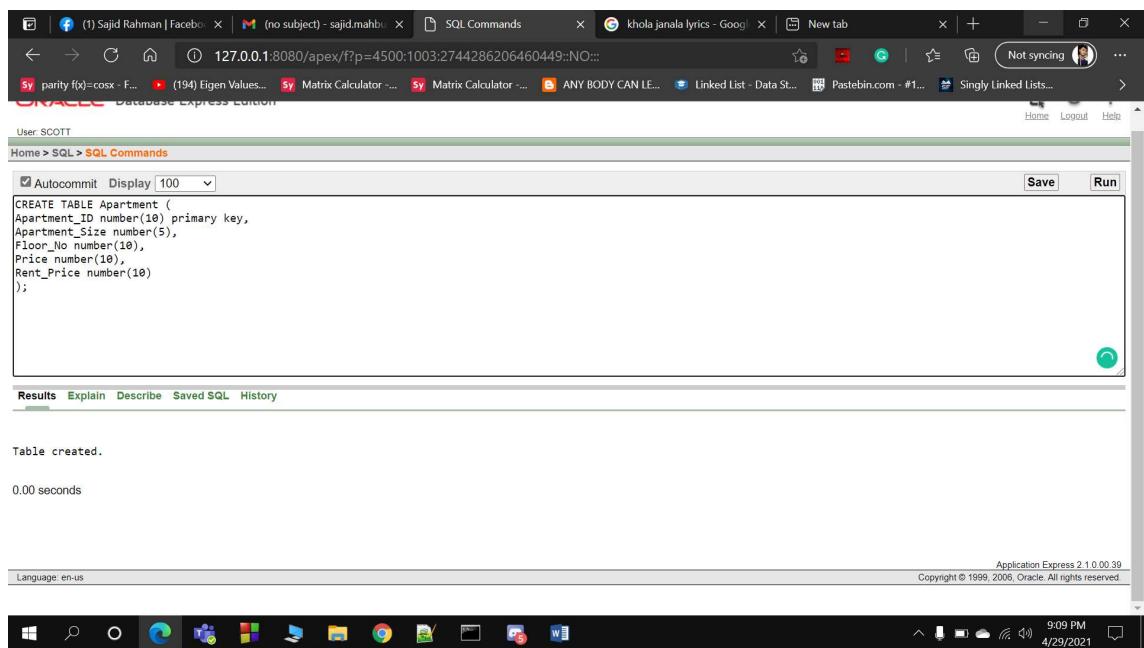
```
CREATE TABLE Owner_Manager(
    Owner_ID number(28),
    Manager_ID number(28)
);
```

The results pane shows the message "Table created." and a duration of "0.01 seconds". The status bar at the bottom right indicates "Application Express 2.1.0.0.39" and "Copyright © 1999, 2006, Oracle. All rights reserved".

```

CREATE TABLE Apartment (
    Apartment_ID number(10) primary key,
    Apartment_Size number(5),
    Floor_No number(10),
    Price number(10),
    Rent_Price number(10)
);

```



The screenshot shows a browser window with multiple tabs open. The active tab is titled "SQL Commands" and contains the SQL code for creating the "Apartment" table. The code is identical to the one provided above. Below the code, the message "Table created." is displayed, along with a timestamp of "0.00 seconds". At the bottom of the page, there is footer information: "Language: en-us", "Application Express 2.1.0.00.39", and "Copyright © 1999, 2006, Oracle. All rights reserved". The browser's address bar shows the URL "127.0.0.1:8080/apex/f?p=4500:1003:2744286206460449::NO::". The operating system taskbar at the bottom shows various application icons.

```

CREATE TABLE Apartment (
    Apartment_ID number(10) primary key,
    Apartment_Size number(5),
    Floor_No number(10),
    Price number(10),
    Rent_Price number(10)
);

Table created.

0.00 seconds

Language: en-us Application Express 2.1.0.00.39
Copyright © 1999, 2006, Oracle. All rights reserved

```

```

CREATE TABLE Developer_Apartment(
    Developer_ID number(20),
    Apartment_ID number(20)
);

```

User SCOTT

Home > SQL > SQL Commands

Autocommit Display 100 Save Run

```
CREATE TABLE Developer_Apartment(
Developer_ID number(20),
Apartment_ID number(20)
);
```

Results Explain Describe Saved SQL History

Table created.

0.00 seconds

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved

CREATE TABLE ServiceProvider (

SP_ID number(20)primary key,

SP_Name varchar2(50),

SP_PhoneNumber number(12)

User SCOTT

Home > SQL > SQL Commands

Autocommit Display 100 Save Run

```
CREATE TABLE ServiceProvider (
SP_ID number(20)primary key,
SP_Name varchar2(50),
SP_PhoneNumber number(12)
);
```

Results Explain Describe Saved SQL History

Table created.

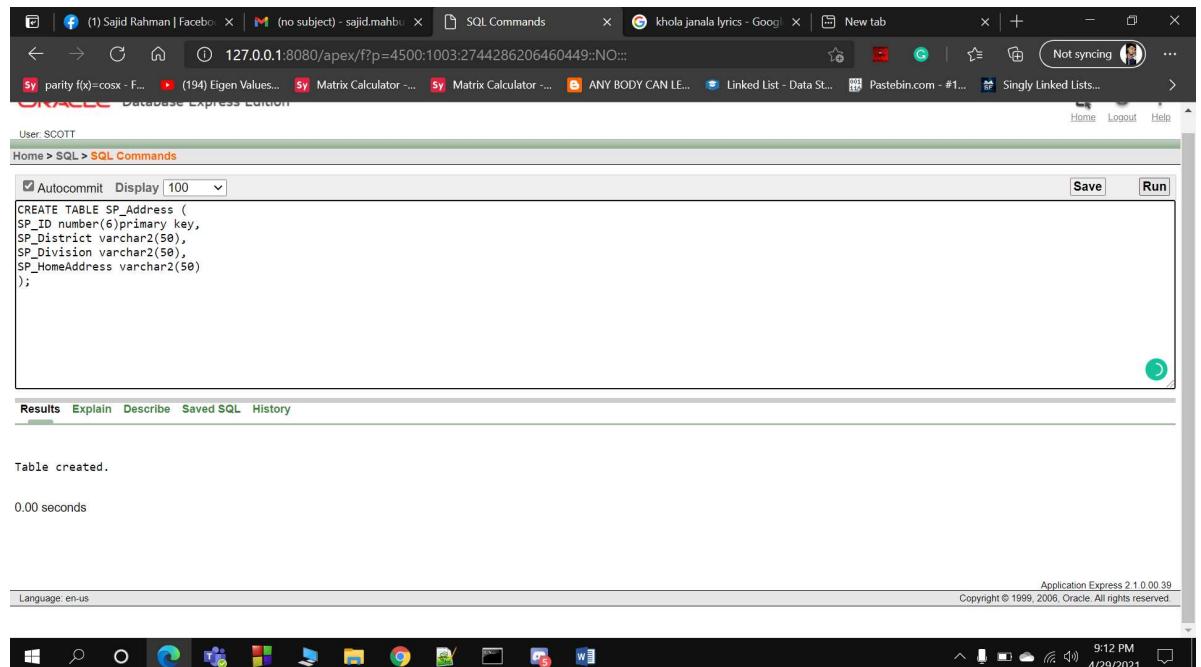
0.01 seconds

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved

```

CREATE TABLE SP_Address (
    SP_ID number(6) primary key,
    SP_District varchar2(50),
    SP_Division varchar2(50),
    SP_HomeAddress varchar2(50)
);

```



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

```

CREATE TABLE SP_Address (
    SP_ID number(6) primary key,
    SP_District varchar2(50),
    SP_Division varchar2(50),
    SP_HomeAddress varchar2(50)
);

```

The results pane below the editor shows the output:

```

Table created.
0.00 seconds

```

The status bar at the bottom right indicates the application version and copyright information.

```

CREATE TABLE Owner_SP(
    Owner_ID number(20),
    SP_ID number(20)
);

```

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

```
CREATE TABLE Manager_SP(
Manager_ID number(20) primary key,
SP_ID number(20)
);

CREATE TABLE SP_Apartment(
SP_ID number(20) primary key,
Apartment_ID number(20)
);
```

The results pane shows the output of the commands:

Table created.
0.05 seconds

Language: en-us Application Express 2.1.0.00.39
Copyright © 1999, 2006, Oracle. All rights reserved.

```
CREATE TABLE Manager_SP(
Manager_ID number(20) primary key,
SP_ID number(20)
);

CREATE TABLE SP_Apartment(
SP_ID number(20) primary key,
Apartment_ID number(20)
);
```

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

```
CREATE TABLE SP_Apartment(
SP_ID number(20)primary key,
Apartment_ID number(20)
);
```

The results pane shows the output:

```
Table created.
```

Execution details:

- Time: 0.42 seconds
- Language: en-us
- Application Express 2.1.0.0.39
- Copyright © 1999, 2006, Oracle. All rights reserved.

```
CREATE TABLE Renter (
Renter_ID number(20)primary key,
Renter_Name varchar2(50),
Renter_PhoneNumber number(12)
);
```

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

```
CREATE TABLE Renter (
Renter_ID number(20)primary key,
Renter_Name varchar2(50),
Renter_PhoneNumber number(12)
);
```

The results pane shows the output:

```
Table created.
```

Execution details:

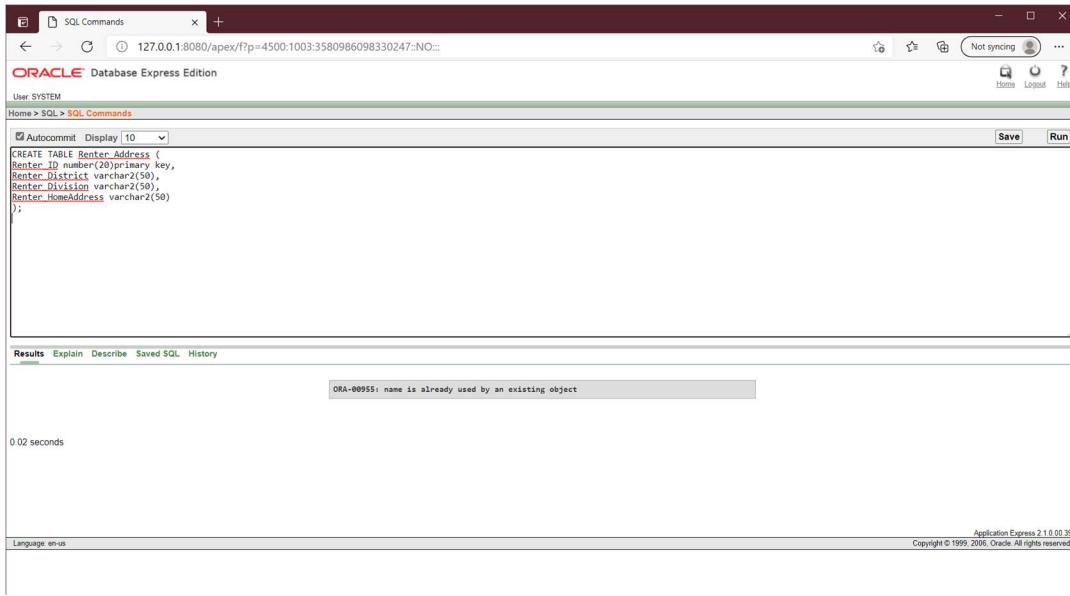
- Time: 0.48 seconds
- Language: en-us
- Application Express 2.1.0.0.39
- Copyright © 1999, 2006, Oracle. All rights reserved.

```
CREATE TABLE Renter_Address (
```

```

Renter_ID number(20)primary key,
Renter_District varchar2(50),
Renter_Division varchar2(50),
Renter_HomeAddress varchar2(50)
);

```



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

```

CREATE TABLE Renter_Address (
Renter_ID number(20) primary key,
Renter_District varchar2(50),
Renter_Division varchar2(50),
Renter_HomeAddress varchar2(50)
);

```

The results pane shows the error message: "ORA-00955: name is already used by an existing object". The application version is Application Express 2.1.0.00.39.

```

CREATE TABLE Renter_Apartment(
Renter_ID number(20)primary key,
Apartment_ID number(20)
);

```

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

```
CREATE TABLE Renter_Apartment(
Renter_ID number(20)primary key,
Apartment_ID number(20)
);
```

The results pane shows the output:

```
Table created.
```

At the bottom, it says "0.23 seconds".

```
CREATE TABLE Buyer (
Buyer_ID number(10)primary key,
Buyer_Name varchar2(50),
Buyer_PhoneNumber number(12)
);
```

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

```
CREATE TABLE Buyer (
Buyer_ID number(10)primary key,
Buyer_Name varchar2(50),
Buyer_PhoneNumber number(12)
);
```

The results pane shows the output:

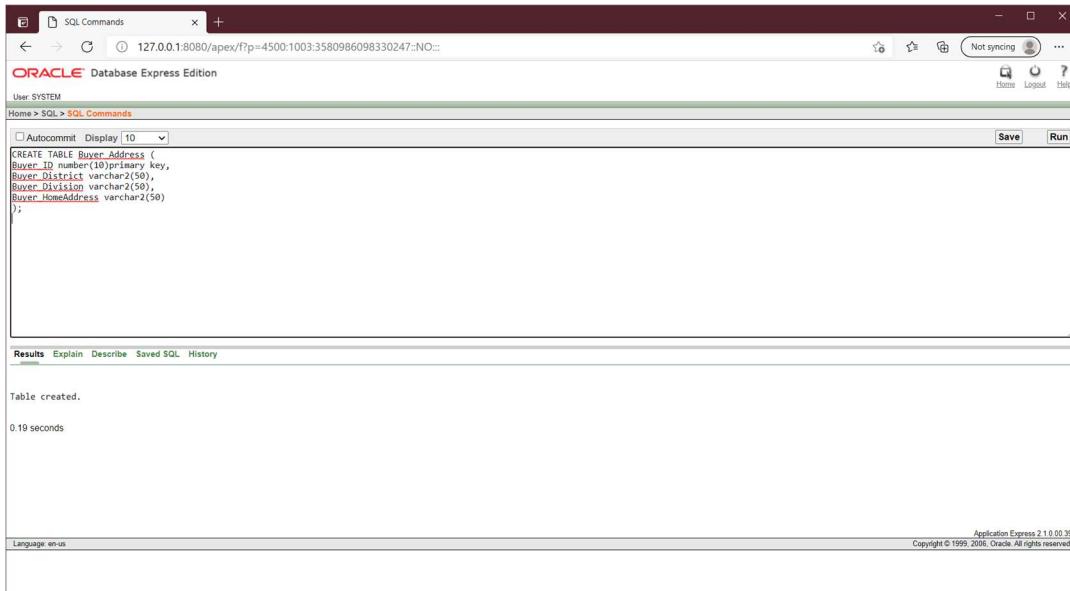
```
Table created.
```

At the bottom, it says "0.25 seconds".

```

CREATE TABLE Buyer_Address (
    Buyer_ID number(10)primary key,
    Buyer_District varchar2(50),
    Buyer_Division varchar2(50),
    Buyer_HomeAddress varchar2(50)
);

```



The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL command for creating the Buyer_Address table is entered in the main pane:

```

CREATE TABLE Buyer_Address (
    Buyer_ID number(10)primary key,
    Buyer_District varchar2(50),
    Buyer_Division varchar2(50),
    Buyer_HomeAddress varchar2(50)
);

```

The interface includes a toolbar at the top, a navigation bar with 'User SYSTEM' and 'Home > SQL > SQL Commands', and a bottom status bar with 'Language: en-us' and 'Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved.'

```

CREATE TABLE Buyer_Apartment(
    Buyer_ID number(10)primary key,
    Apartment_ID number(10)
);

```

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

```
CREATE TABLE Owner_Apartment(
    Owner_ID number(10) primary key,
    Apartment_ID number(10)
);
```

The results pane shows the output:

```
Table created.  
0.01 seconds
```

At the bottom, it says "Language: en-us" and "Application Express 2.1.0.0.39 Copyright © 1999, 2005, Oracle. All rights reserved."

```
CREATE TABLE Owner_Apartment(
    Owner_ID number(10) primary key,
    Apartment_ID number(10)
);
```

```
CREATE TABLE Manager_Apartment(
    Manager_ID number(20),
    Apartment_ID number(20)
);
```

```
CREATE TABLE Rentgrade(
    Grade number(10) primary key,
    Lorent number(10),
    Hirrent number(10)
);
```

Table Insertion

1. Owner Table

- Insert INTO OWNER VALUES (01, 'Ifaj Hossan', 01859159874);
- Insert INTO OWNER VALUES (02, 'Shahidul Ifte', 01855564367);
- Insert INTO OWNER VALUES (03, 'Mustasir rahman', 01855764366);
- Insert INTO OWNER VALUES (04, 'Mugdho Rahim', 01856764367);
- Insert INTO OWNER VALUES (05, 'Nazim Buiya', 0185566758);

```
Insert INTO OWNER_Address VALUES (01, 'Dhaka', 'Dhaka', '67/5, Uttara');  
Insert INTO OWNER_Address VALUES (02, 'Dhaka', 'Dhaka', '76/3, Mirpur');  
Insert INTO OWNER_Address VALUES (03, 'Dhaka', 'Dhaka', '56, North Kafrul');  
Insert INTO OWNER_Address VALUES (04, 'Dhaka', 'Dhaka', '44/3, Tejgaon');  
Insert INTO OWNER_Address VALUES (05, 'Dhaka', 'Dhaka', '455, South Kafrul');
```

2. Developer Table:

```
Insert INTO DEVELOPER VALUES (3110, 'Ahim','Mirpur', 01859159844);  
Insert INTO DEVELOPER VALUES (3111, 'Rahim','Mirpur', 01859159899);  
Insert INTO DEVELOPER VALUES (3112, 'Zahib','Mirpur', 0185915988);  
Insert INTO DEVELOPER VALUES (3113, 'Karim','Mirpur', 0185915989);  
Insert INTO DEVELOPER VALUES (3114, 'Malik','Mirpur', 0185915990);
```

3. Manager Table:

```
Insert INTO MANAGER VALUES (2111, 'IQZI', 01759159874);  
Insert INTO MANAGER VALUES (2112, 'Kazi', 01759159877);  
Insert INTO MANAGER VALUES (2113, 'Atiq', 01759159878);  
Insert INTO MANAGER VALUES (2114, 'Fuad', 01759159887);  
Insert INTO MANAGER VALUES (2115, 'Pulok', 01659159887);
```

```
Insert INTO MANAGER_Address VALUES (2111, 'Narangang', 'Dhaka', '12, Fatulla');

Insert INTO MANAGER_Address VALUES (2112, 'Narangang', 'Dhaka', '34, Cokhbazar');

Insert INTO MANAGER_Address VALUES (2113, 'Narangang', 'Dhaka', '65, Biruia');

Insert INTO MANAGER_Address VALUES (2114, 'Narangang', 'Dhaka', '88, Shiroi');

Insert INTO MANAGER_Address VALUES (2115, 'Narangang', 'Dhaka', '91, Altiarpur');
```

4. Apartment Table:

```
Insert INTO Apartment VALUES (7856, 1204, 5, 7800000, 20000)

Insert INTO Apartment VALUES (7857, 1205, 5, 8000000, 30000)

Insert INTO Apartment VALUES (7858, 1206, 5, 8500000, 30000);

Insert INTO Apartment VALUES (7859, 1207, 6, 6000000, 60000);

Insert INTO Apartment VALUES (7860, 1208, 6, 6100000, 69000);
```

5. Service Provider Table:

```
Insert INTO ServiceProvider VALUES (6540, 'Sean', 01631215211);

Insert INTO ServiceProvider VALUES (6541, 'Momo', 01631215434);

Insert INTO ServiceProvider VALUES (6542, 'Shams', 01631215667);

Insert INTO ServiceProvider VALUES (6543, 'Jubin', 01631215669);
```

Insert INTO ServiceProvider VALUES (6544, 'Jahin', 01731515669);

Insert INTO SP_Address VALUES (6540, 'Cumilla', 'Chitagong', 'Islampur');

Insert INTO SP_Address VALUES (6541, 'Cumilla', 'Chitagong', 'Kandirpar');

Insert INTO SP_Address VALUES (6542, 'Cumilla', 'Chitagong', 'Kandirpar');

Insert INTO SP_Address VALUES (6543, 'Cumilla', 'Chitagong', 'Police line');

Insert INTO SP_Address VALUES (6544, 'Cumilla', 'Chitagong', 'Bagichar');

6. Renter Table:

Insert INTO RENTER VALUES (1150, 'Imam', 01859159774);

Insert INTO RENTER VALUES (1151, 'Sagor Mohir', 01759159774);

Insert INTO RENTER VALUES (1152, 'Ovy Mohammad', 01659159777);

Insert INTO RENTER VALUES (1153, 'Aloy Rahman', 01959159745);

Insert INTO RENTER VALUES (1154, 'Protik Parlay', 0177915974);

Insert INTO Renter_Address VALUES (1150, 'Pabna', 'Rajshahi', '23, Tikkachor');

Insert INTO Renter_Address VALUES (1151, 'Dhaka', 'Dhaka', '55, Gulshan');

Insert INTO Renter_Address VALUES (1152, 'Dhaka', 'Dhaka', '57, Banani');

Insert INTO Renter_Address VALUES (1153, 'Dhaka', 'Dhaka', '78, New Market');

Insert INTO Renter_Address VALUES (1154, 'Dhaka', 'Dhaka', '88, Science Club');

7. Buyer Table :

Insert INTO Buyer VALUES (8220, 'Maruf Rahman', 01312425175);

Insert INTO Buyer VALUES (8221, 'Prinoy Khan', 01312425187);

Insert INTO Buyer VALUES (8222, 'Muntasir Mim', 01942425187);

Insert INTO Buyer VALUES (8223, 'Nadim Bhuiya', 01762425187);

Insert INTO Buyer VALUES (8224, 'Dr. Abdus Salam', 01762425877);

Insert INTO Buyer_Address VALUES (8220, 'Chadpur', 'Chitagong', 'Bahrirpara');

Insert INTO Buyer_Address VALUES (8221, 'Dhaka', 'Dhaka', 'Cantonment');

Insert INTO Buyer_Address VALUES (8222, 'Dhaka', 'Dhaka', 'Adamjee Quater');

Insert INTO Buyer_Address VALUES (8223, 'Dhaka', 'Dhaka', 'Shaheenbag');

Insert INTO Buyer_Address VALUES (8224, 'Dhaka', 'Dhaka', 'Nikunjo');

8. Owner_Developer Table:

Insert INTO Owner_Developer VALUES (01, 3110);

```
Insert INTO Owner_Developer VALUES (02, 3111);
```

```
Insert INTO Owner_Developer VALUES (03, 3112);
```

```
Insert INTO Owner_Developer VALUES (04, 3113);
```

```
Insert INTO Owner_Developer VALUES (05, 3114);
```

9. Owner_Manager Table:

```
Insert INTO Owner_Manager VALUES (01, 2111);
```

```
Insert INTO Owner_Manager VALUES (02, 2112);
```

```
Insert INTO Owner_Manager VALUES (03, 2113);
```

```
Insert INTO Owner_Manager VALUES (04, 2114);
```

```
Insert INTO Owner_Manager VALUES (05, 2115);
```

10. Manager_Apartment Table:

```
Insert INTO Manager_Apartment VALUES ( 2111,7856);
```

```
Insert INTO Manager_Apartment VALUES ( 2112,7857);
```

```
Insert INTO Manager_Apartment VALUES ( 2113,7860);
```

11. Manager_SP Table:

```
Insert INTO Manager_SP VALUES (2111, 6540);
```

```
Insert INTO Manager_SP VALUES (2112, 6541);
```

12.SP_Apartment Table:

```
Insert INTO SP_Apartment VALUES (6540,7856);  
Insert INTO SP_Apartment VALUES (6541,7857);
```

13.Developer_Apartment Table:

```
Insert INTO Developer_Apartment VALUES (3110,7856);  
Insert INTO Developer_Apartment VALUES (3111,7857);  
Insert INTO Developer_Apartment VALUES (3112, 7858);  
Insert INTO Developer_Apartment VALUES (3113, 7859);  
Insert INTO Developer_Apartment VALUES (3114, 7860);
```

14.Renter_Apartment Table:

```
Insert INTO Renter_Apartment VALUES (1150,7856);  
Insert INTO Renter_Apartment VALUES (1151,7858);
```

15.Buyer_Apartment Table:

```
Insert INTO Buyer_Apartment VALUES (8220,7857);  
Insert INTO Buyer_Apartment VALUES (8221,7859);
```

16. Owner_Apartment Table:

```
Insert INTO Owner_Apartment VALUES (01,7856);  
Insert INTO Owner_Apartment VALUES (02,7857);  
Insert INTO Owner_Apartment VALUES (03, 7858);  
Insert INTO Owner_Apartment VALUES (04, 7859);
```

```
Insert INTO Owner_Apartment VALUES (05, 7860);
```

16. Rentgrade Table:

```
Insert INTO Rentgrade VALUES (1,10000,20000);
```

```
Insert INTO Rentgrade VALUES (2,20001,30000);
```

```
Insert INTO Rentgrade VALUES (3,30001,40000);
```

```
Insert INTO Rentgrade VALUES (4,40001,50000);
```

```
Insert INTO Rentgrade VALUES (5,50001,60000);
```

```
Insert INTO Rentgrade VALUES (6,60001,70000);
```

```
Insert INTO Rentgrade VALUES (7,70001,80000);
```

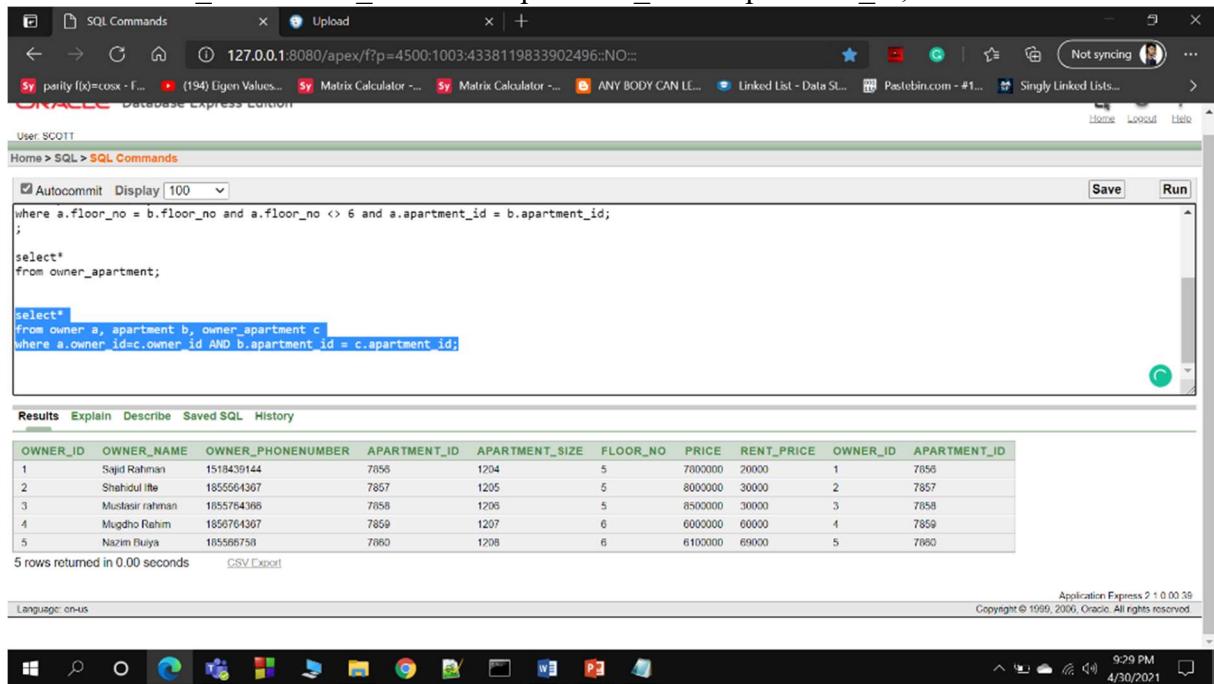
Task

Joining

- 1) Display all the information of the owners and their apartments

Ans: select*

```
from owner a, apartment b, owner_apartment c  
where a.owner_id=c.owner_id AND b.apartment_id = c.apartment_id;
```



The screenshot shows the Oracle Database Express Edition interface. A SQL command window is open with the following SQL code:

```
where a.floor_no = b.floor_no and a.floor_no <> 6 and a.apartment_id = b.apartment_id;  
;  
select*  
from owner_apartment;  
  
select*  
from owner a, apartment b, owner_apartment c  
where a.owner_id=c.owner_id AND b.apartment_id = c.apartment_id;
```

The results section displays a table with the following data:

OWNER_ID	OWNER_NAME	OWNER_PHONENUMBER	APARTMENT_ID	APARTMENT_SIZE	FLOOR_NO	PRICE	RENT_PRICE	OWNER_ID	APARTMENT_ID
1	Sajid Rahman	1518439144	7856	1204	5	7800000	20000	1	7856
2	Shehidul Ifte	1855564367	7857	1205	5	8000000	30000	2	7857
3	Mustasir rahman	1855764366	7858	1206	5	8500000	30000	3	7858
4	Mugdho Rahim	1856764367	7859	1207	6	6000000	60000	4	7859
5	Nazim Buuya	185686758	7860	1208	6	6100000	69000	5	7860

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

Application Express 21.0.0.39
Copyright © 1999, 2000, Oracle. All rights reserved.

Language: en-US 9:29 PM 4/30/2021

- 2) Display managerid, apartmentid and rent price of the Managers and their apartments

Ans: select e.manager_id, b.apartment_id

```
from apartment b, manager e, manager_apartment c
```

```
where e.manager_id = c.manager_id AND b.apartment_id = c.apartment_id;
```

The screenshot shows the Oracle Database Express Edition Application Express interface. The browser tab is titled "127.0.0.1:8080/apex/f?p=4500:1003:1893545540886945::NO::". The main content area displays a SQL command:

```
select e.manager_id, b.apartment_id
from apartment b, manager e, manager_apartment c
where e.manager_id = c.manager_id AND b.apartment_id = c.apartment_Id;
```

The results section shows a table with three rows:

MANAGER_ID	APARTMENT_ID
2111	7856
2112	7857
2113	7860

Below the table, it says "3 rows returned in 0.00 seconds" and there is a "CSV Export" link.

At the bottom of the page, the footer includes "Application Express 2.1 0:00:39", "Copyright © 1999, 2006, Oracle. All rights reserved.", and a system status bar showing "Language: en-us", "12:28 PM 5/1/2021", and various icons.

- 3) Show all the owners name, phone number, rentgrade.

Ans :

```
select a.owner_name, a.owner_phonenumber, grade
from owner a, apartment b, owner_apartment c, rentgrade d
where a.owner_id=c.owner_id AND b.apartment_id = c.apartment_id AND b.rent_price
between d.lorent AND d.hirent;
```

```

User SCOTT
Home > SQL > SQL Commands
Autocommit Display | 100 | Save | Run
;
select*
from apartment;

select c.manager_id,b.apartment_id, b.rent_price
from owner a, apartment b, owner_manager c, owner_apartment d, manager e
where a.owner_id=c.owner_id AND b.apartment_id = d.apartment_id AND a.owner_id = d.owner_id AND e.manager_id=c.manager_id;

select a.owner_name, a.owner_phonenumber, grade
from owner a, apartment b, owner_apartment c, rentgrade d
where a.owner_id=c.owner_id AND b.apartment_id = c.apartment_id AND b.rent_price between d.lorent AND d.hirent;

```

Results

OWNER_NAME	OWNER_PHONENUMBER	GRADE
Nazim Buya	185566758	6
Mugdho Rahim	1856764367	5
Shandul Ifte	1855564367	2
Mustasir rahman	1855764366	2
Sajid Rahman	1518439144	1

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

Application Express 2.1 0:00:39
Copyright © 1999, 2006, Oracle. All rights reserved.

- 4) Show all the apartment_id of the apartments that are on the 5th floor

Ans:

```

select a.apartment_id
from apartment a, apartment b
where a.floor_no = b.floor_no and a.floor_no <> 6 and a.apartment_id = b.apartment_id

```

```

User SCOTT
Home > SQL > SQL Commands
Autocommit Display | 100 | Save | Run
;
from owner a, apartment b, owner_manager c, owner_apartment d, manager e
where a.owner_id=c.owner_id AND b.apartment_id = d.apartment_id AND a.owner_id = d.owner_id AND e.manager_id=c.manager_id;

select a.owner_name, a.owner_phonenumber, grade
from owner a, apartment b, owner_apartment c, rentgrade d
where a.owner_id=c.owner_id AND b.apartment_id = c.apartment_id AND b.rent_price between d.lorent AND d.hirent;

select a.apartment_id
from apartment a, apartment b
where a.floor_no = b.floor_no and a.floor_no <> 6 and a.apartment_id = b.apartment_id;

```

Results

APARTMENT_ID
7856
7857
7858

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

Application Express 2.1 0:00:39
Copyright © 1999, 2006, Oracle. All rights reserved.

- 5) Display the apartments without any manager

Ans:

```

select a.manager_name, b.apartment_id
from manager a, apartment b, manager_apartment c
where a.manager_id(+) = c.manager_id AND b.apartment_id = c.apartment_id(+);

```

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

```

select a.manager_name, b.apartment_id
from manager a, apartment b, manager_apartment c
where a.manager_id(+) = c.manager_id AND b.apartment_id = c.apartment_id(+);

```

The results window displays the following data:

MANAGER_NAME	APARTMENT_ID
IQZI	7856
Kazi	7857
Atiq	7860
-	7858
-	7859

5 rows returned in 0.00 seconds

CSV Export

Application Express 2.1 0.00.39
Copyright © 1999, 2006, Oracle. All rights reserved.

Subquery

- 6) Display the apartment information that has the highest price tag

Ans : select*

from apartment

where price = (select max(price) from apartment);

```

User SCOTT
Home > SQL > SQL Commands
 Autocommit Display | 100 | + Save Run
select*
from apartment;

select*
from apartment
where price = (select max(price) from apartment );

```

Results Explain Describe Saved SQL History

APARTMENT_ID	APARTMENT_SIZE	FLOOR_NO	PRICE	RENT_PRICE
7858	1206	5	850000	30000

1 rows returned in 0.05 seconds [CSV Export](#)

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

- 7) Display the third maximum price of an apartment

Ans:

```

select max(price)
from apartment
where price < (select max(price) from apartment where price <(select max(price) from
apartment) );

```

```

User SCOTT
Home > SQL > SQL Commands
 Autocommit Display | 100 | + Save Run
select*
from apartment;

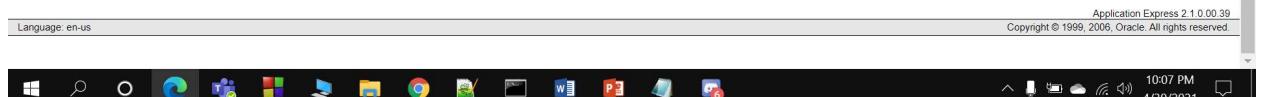
select max(price)
from apartment
where price < (select max(price) from apartment where price <(select max(price) from apartment) );

```

Results Explain Describe Saved SQL History

MAX(PRICE)
780000

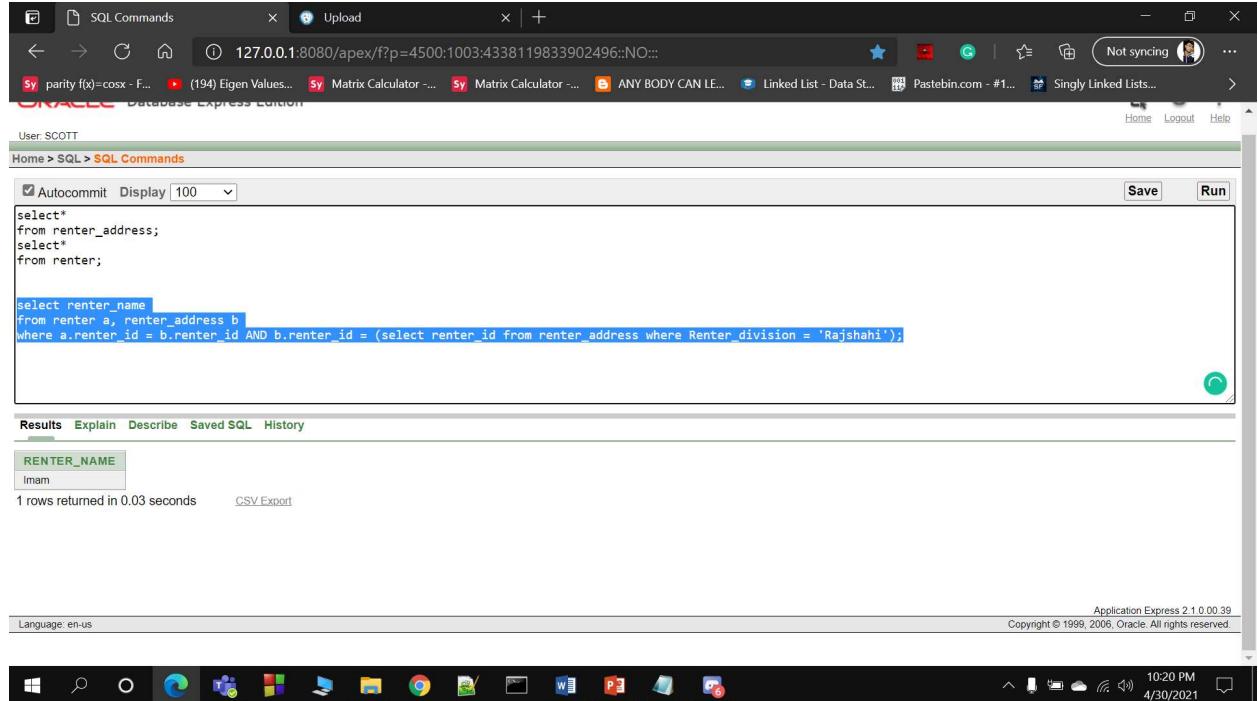
1 rows returned in 0.00 seconds [CSV Export](#)



- 8) Display the renters name whose home division is Rajshahi

Ans :

```
select renter_name
from renter a, renter_address b
where a.renter_id = b.renter_id AND b.renter_id = (select renter_id from renter_address where
Renter_division = 'Rajshahi');
```



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

```
select*
from renter_address;
select*
from renter;

select renter_name
from renter a, renter_address b
where a.renter_id = b.renter_id AND b.renter_id = (select renter_id from renter_address where Renter_division = 'Rajshahi');
```

The results pane shows a single row with the column header "RENTER_NAME" and the value "Imam". Below the results, it says "1 rows returned in 0.03 seconds".

At the bottom of the interface, there is a Windows taskbar with various icons and a system tray showing the date and time.

- 9) Display the name of the renter who rented apartment number 7856

Ans :

```
select renter_name
from renter
where renter_id = (select renter_id from renter_apartment where apartment_id = 7856);
```

SQL Commands Index

127.0.0.1:8080/apex/f?p=4500:1003:4338119833902496::NO::

User SCOTT

Home > SQL > SQL Commands

Autocommit Display | 100 Save Run

```
select*
from renter_address;
select*
from renter_apartment;

select renter_name
from renter
where renter_id = (select renter_id from renter_apartment where apartment_id = 7856);
```

Results Explain Describe Saved SQL History

RENTER_NAME
Imam

1 rows returned in 0.00 seconds CSV Export

Application Express 2.1.0.0:39
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

10:32 PM 4/30/2021

View

- 1) CREATE A VIEW NAMED APART_VU THAT CONTAINS THE BUYER NAME of the apartments that are priced higher than average apartment price

Ans:

```
create OR REPLACE view apart_vu (Name)
AS SELECT A.BUYER_NAME
FROM BUYER A, BUYER_APARTMENT B, APARTMENT C
WHERE A.BUYER_ID = B.BUYER_ID AND C.PRICE > (SELECT AVG(PRICE) FROM APARTMENT)
GROUP BY A.BUYER_NAME;
```

The screenshot shows the Oracle Database Express Edition interface. The title bar says "SQL Commands". The main area contains the following SQL code:

```
select*
from buyer;
select*
from buyer_apartment;
SELECT*
FROM APART_VU;

create OR REPLACE view apart_vu (Name)
AS SELECT A.BUYER_NAME
FROM BUYER A, BUYER_APARTMENT B, APARTMENT C
WHERE A.BUYER_ID = B.BUYER_ID AND C.PRICE > (SELECT AVG(PRICE) FROM APARTMENT)
GROUP BY A.BUYER_NAME;
```

Below the code, there are tabs for "Results", "Explain", "Describe", "Saved SQL", and "History". The "Results" tab is selected, showing a table with one row:

NAME
Maruf Rahman
Pinoy Khan

At the bottom, it says "2 rows returned in 0.00 seconds" and "CSV Export".

At the very bottom of the window, it says "Language: en-us" and "Application Express 2.1.0.0.0.39".

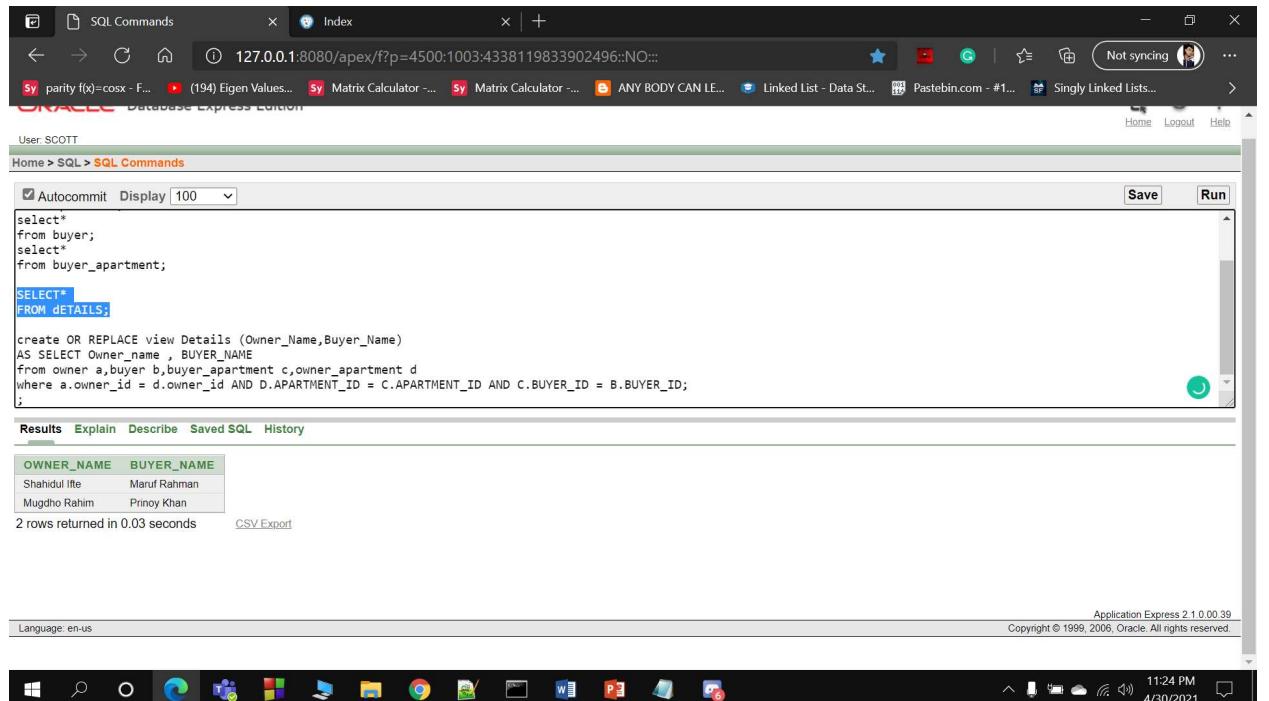
2)

Create a view Details which contains the owner name and buyer name of an apartment

Ans:

```
create OR REPLACE view Details (Owner_Name,Buyer_Name)
AS SELECT Owner_name , BUYER_NAME
from owner a,buyer b,buyer_apartment c,owner_apartment d
```

```
where a.owner_id = d.owner_id AND D.APARTMENT_ID = C.APARTMENT_ID AND C.BUYER_ID =  
B.BUYER_ID;
```



The screenshot shows the Oracle Application Express SQL Commands interface. The user is SCOTT. The SQL code entered is:

```
select*  
from buyer;  
select*  
from buyer_apartment;  
  
SELECT*  
FROM DETAILS;  
  
create OR REPLACE view Details (Owner_Name,Buyer_Name)  
AS SELECT Owner_name , BUYER_NAME  
from owner a,buyer b,buyer_apartment c,owner_apartment d  
where a.owner_id = d.owner_id AND D.APARTMENT_ID = C.APARTMENT_ID AND C.BUYER_ID = B.BUYER_ID;  
;
```

The results section displays a table with two rows:

OWNER_NAME	BUYER_NAME
Shahidul Ifte	Maruf Rahman
Mugdha Rahim	Prinoy Khan

2 rows returned in 0.03 seconds [CSV Export](#)

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

Constraint

- 1) alter table owner modify owner_id not null;

