Final Project

Introduction to Database

Section: J

Group 02

Project Title: Nursery Management System

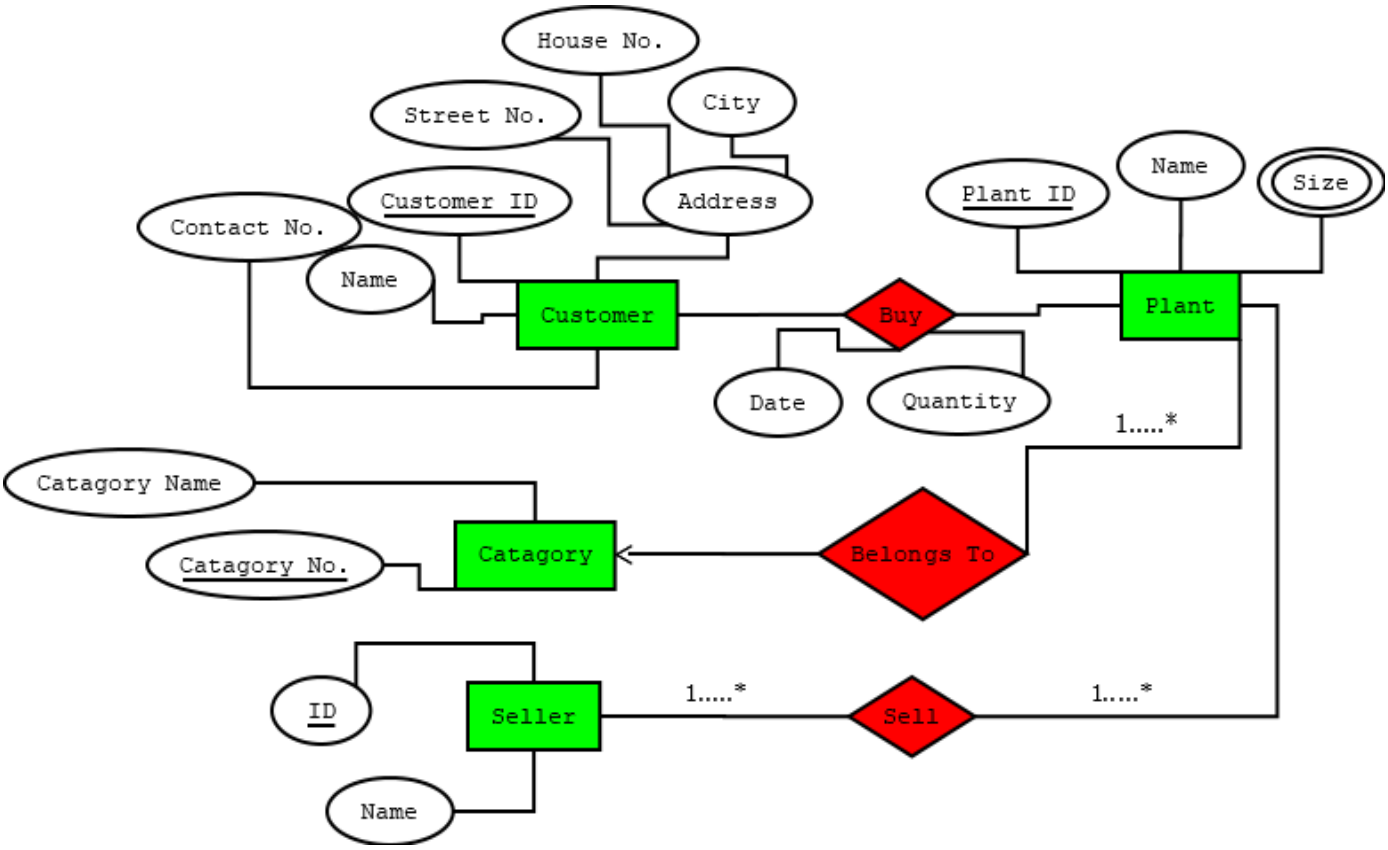
Group Members:

SL. NO.	NAME	ID
01	Sahosh, Zerin Hasan	20-43744-2
02	Hoque, Nayeema	20-43737-2
03	Reaj, Md. Asgor Hossain	20-43999-2
04	Abdullah, Yamlick	20-42687-1
05	Sarkar, Joy	20-43728-2

ER Diagram Paragraph

In a nursery management system, many customers may buy many plants. One plant may be bought by exactly one customer. A customer is identified by a customer number. The system also stores customer name, contact number and address. A customer address is composed of house number, street name and city. A plant is identified by plant ID. Plant name and size of a plant are also stored. There may be multiple size of a plant. While buying, the date of the buying and quantity number of the plant is stored. A plant belongs to exactly one category and for a category there must be at least one plant. Each category has a name and the unique property of each category is a category number. In this nursery many workers work called seller. A plant is sold by at least one seller. A seller may sell many plants but the system stores information of those sellers of who has sold at least one plant have in the nursery. To identify a seller, the system stores seller ID along with seller name.

ER Diagram


House No.
Street No.
City
Customer ID
Address
Contact No.
Name
Customer
Buy
Plant ID
Name
Size
Plant
Date
Quantity
1.....*
Catagory Name
Catagory No.
Catagory
Belongs To
ID
Seller
1.....*
Sell
1.....*
Name

Normalization

Buy: -

UNF:

Customer_ID, Name, Contact_No., Address, Street_No., House_No, City, Date, Quantity, Plant_ID, Name, Size

Size is a multivalued attribute.

1NF:

Customer_ID, Size, Plant_ID, Name, Contact_No., Address, Street_No., House_No, City, Date, Quantity, Name

2NF:

1. Customer_ID, Name, Contact_No., Street_No., House_No, City, Date, Quantity
2. Plant_ID, Name, Size

3NF:

1. Customer_ID, Name, Contact_No., Date, Quantity
2. Address_id, Street_No., House_No, City
3. Plant_ID, Name, Size

Total table from buy:

1. Customer_ID, Name, Contact_No., Date, Quantity, Address_id
2. Address_id, Street_No., House_No, City.
3. Plant_ID, Name, Customer_ID
4. Plant_ID, Size. (Composite key)

Belongs_To: -

UNF:

Category_ID, Category_Name, Plant_ID, Name, Size

Size is a multivalued attribute.

1NF: Category_ID, Size, Plant_ID, Category_Name, Name

2NF:

1. **Category_ID, Category_Name**
2. **Plant_ID, Name, Size**

3NF:

1. **Category_ID, Category_Name**
2. **Plant_ID, Name, Size**

Total table from Belongs_To

5. **Category_ID, Category_Name**
6. **Plant_ID, Name, Category_ID**
7. **Plant_ID, Size. (Composite key)**

Sell: -

UNF:

ID, Name, Plant_ID, Name, Size

Size is a multivalued attribute.

1NF:

ID, Size, Plant_ID, Name, Name,

2NF:

1. **ID, Name**
2. **Plant_ID, Name, Size**

3NF:

1. **ID, Name**
2. **Plant_ID, Name, Size**

Total table from sell:

8. **ID, Name**
9. **Plant_ID, Name, ID**
10. **Plant_ID, Size. (Composite key)**

Total Table

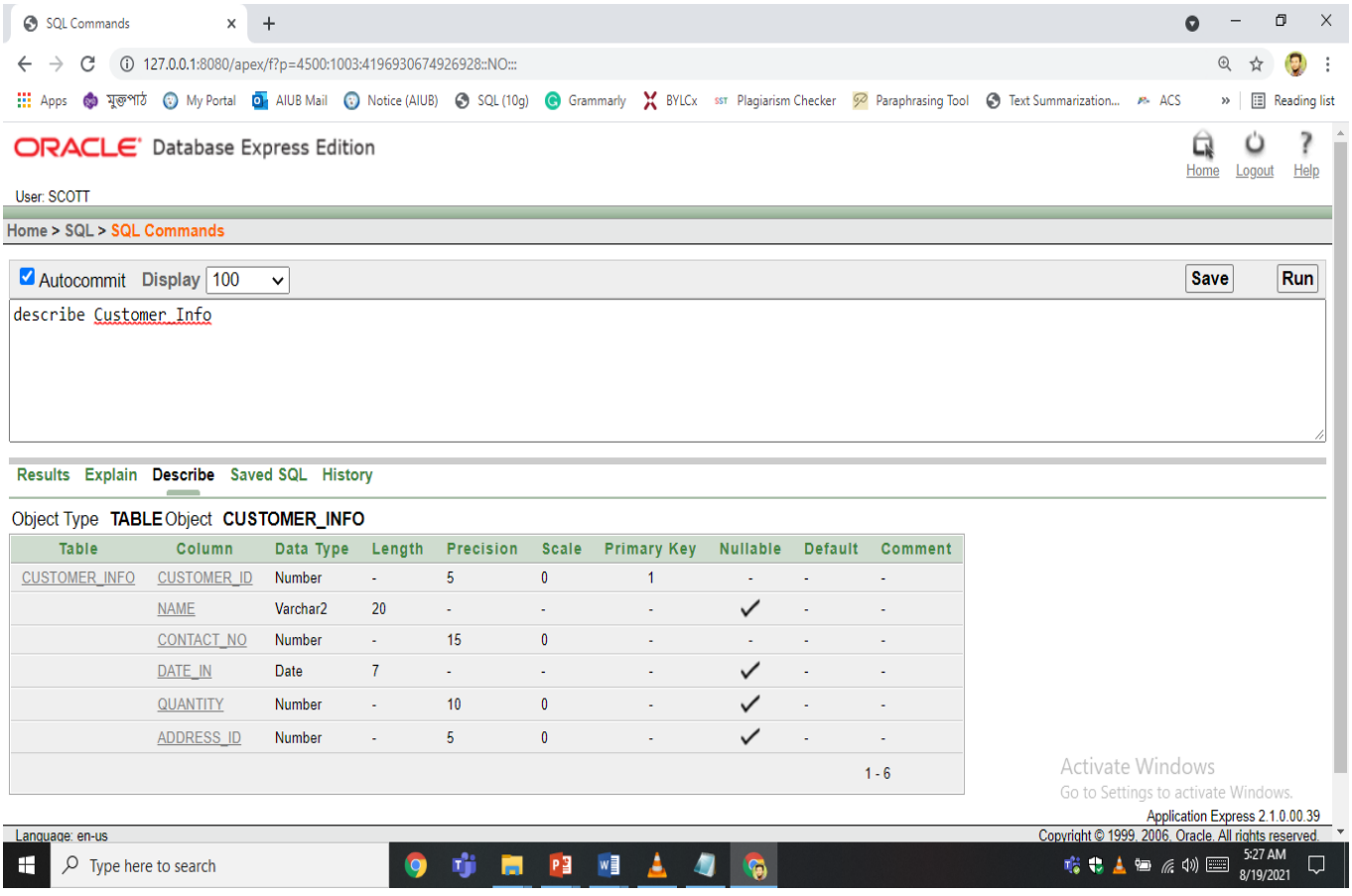
1. **Customer_ID, Name, Contact_No., Date, Quantity, Address_id**
2. **Address_id, Street_No., House_No, City.**
3. **Plant_ID, Name, Customer_ID, ID, Category_ID**
4. **Plant_ID, Size.**
5. **Category_ID, Category_Name**
6. ID, Name

Table Creation

01.Customer_Info:

```
CREATE TABLE Customer_Info
  ( Customer_ID NUMBER(5) CONSTRAINT PK_CUSTOMER PRIMARY KEY,
   Name VARCHAR2(20),
   Contact_No NUMBER(15) NOT NULL,
   Date_IN DATE,
   Quantity NUMBER(10),
   Address_id NUMBER(5) CONSTRAINT Address_FK REFERENCES
Customer_Address_Details);
```

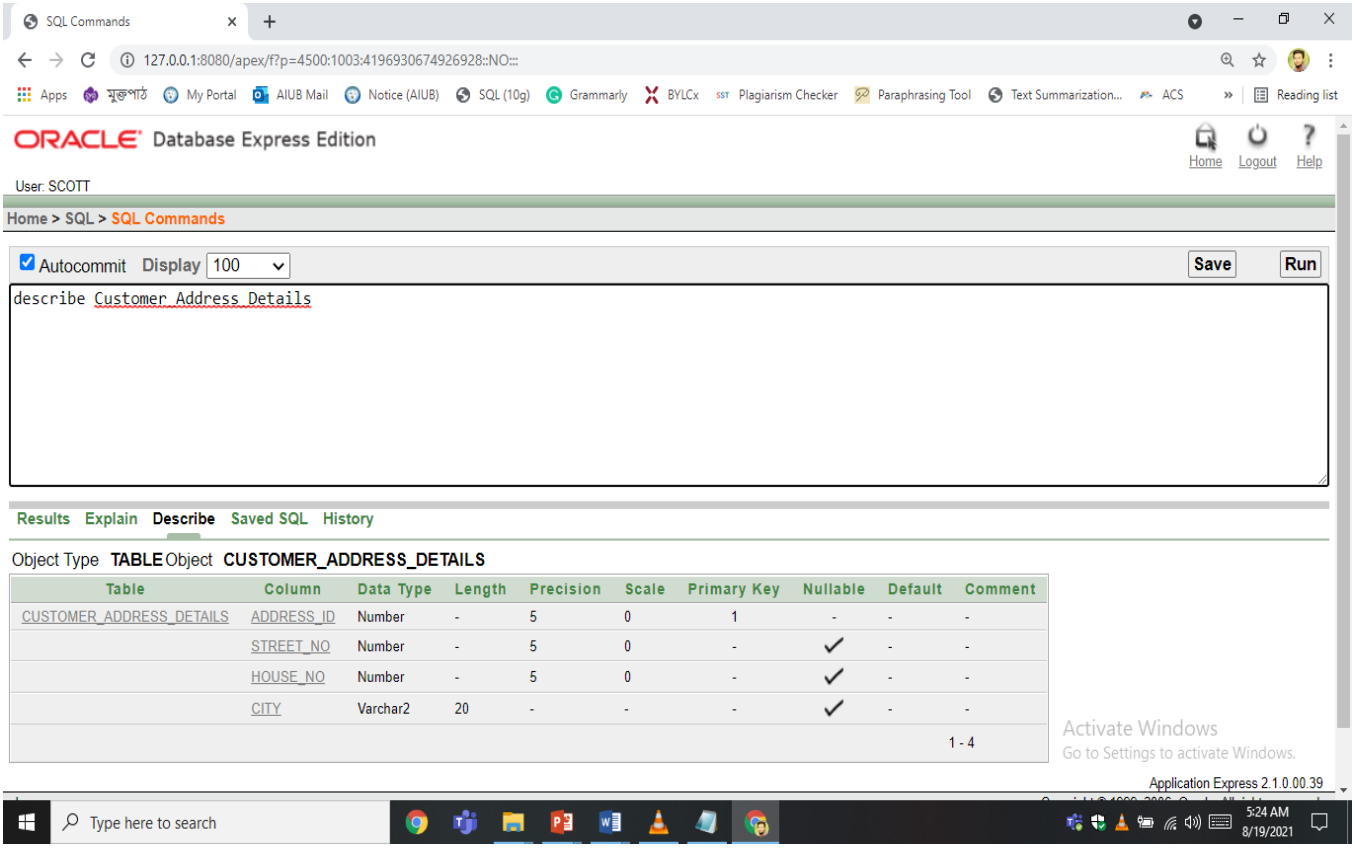
Screenshot:



02. Customer_Address_Details:

```
CREATE TABLE Customer_Address_Details
  ( Address_id NUMBER(5) CONSTRAINT PK_Address_ID PRIMARY KEY,
    Street_No NUMBER(5),
    House_No NUMBER(5),
    City VARCHAR2(20));
```

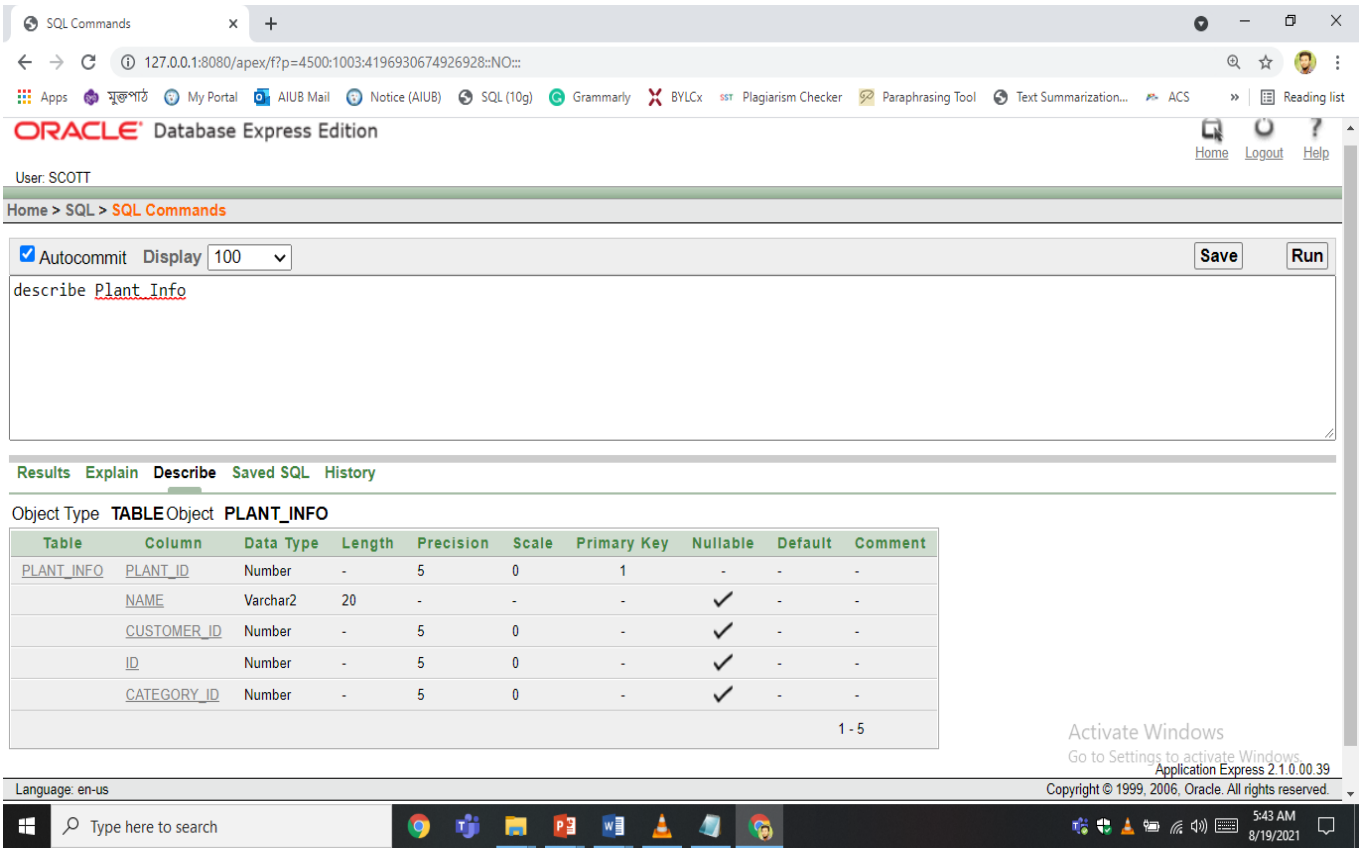
Screenshot:



03. Plant_Info:

```
CREATE TABLE Plant_Info
  ( Plant_ID NUMBER(5) CONSTRAINT PK_Plant_ID PRIMARY KEY,
    Name VARCHAR2(20),
    Customer_ID NUMBER(5) CONSTRAINT Customer_FK REFERENCES Customer_Info,
    ID NUMBER(5) CONSTRAINT ID_FK REFERENCES Seller_Info,
    Category_ID NUMBER(5) CONSTRAINT Category_ID_FK REFERENCES
Category_Details);
```

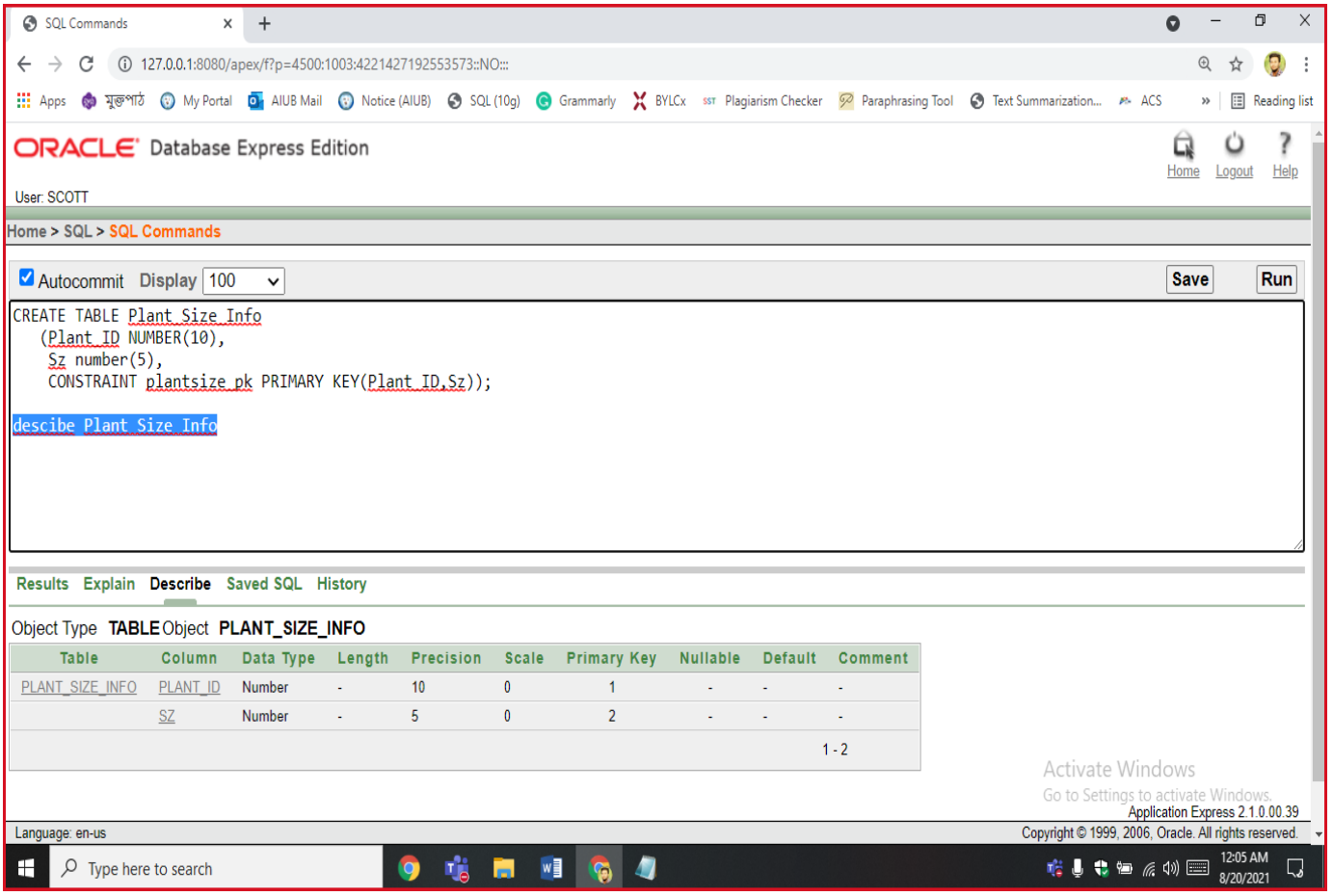
Screenshot:



4. Plant_Size_Info:

```
CREATE TABLE Plant_Size_Info
  (Plant_ID NUMBER(10),
   Sz NUMBER(5),
   CONSTRAINT PLANTSIZE_PK PRIMARY KEY(Plant_ID,Sz));
```

Screenshot:



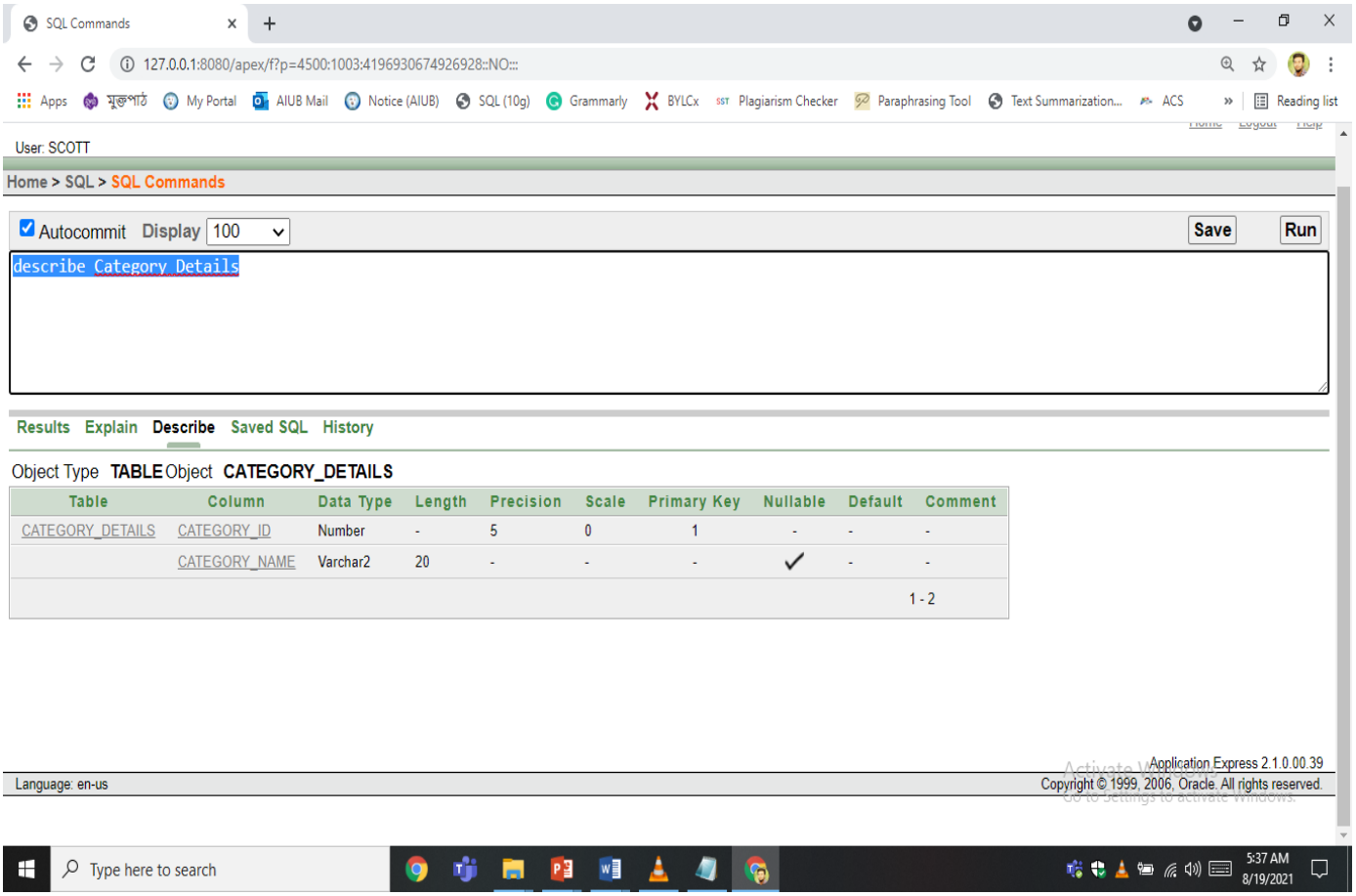
05. Category_Details:

CREATE TABLE Category_Details

(Category_ID NUMBER(5) CONSTRAINT PK_Category_ID PRIMARY KEY,

Category_Name VARCHAR2(20));

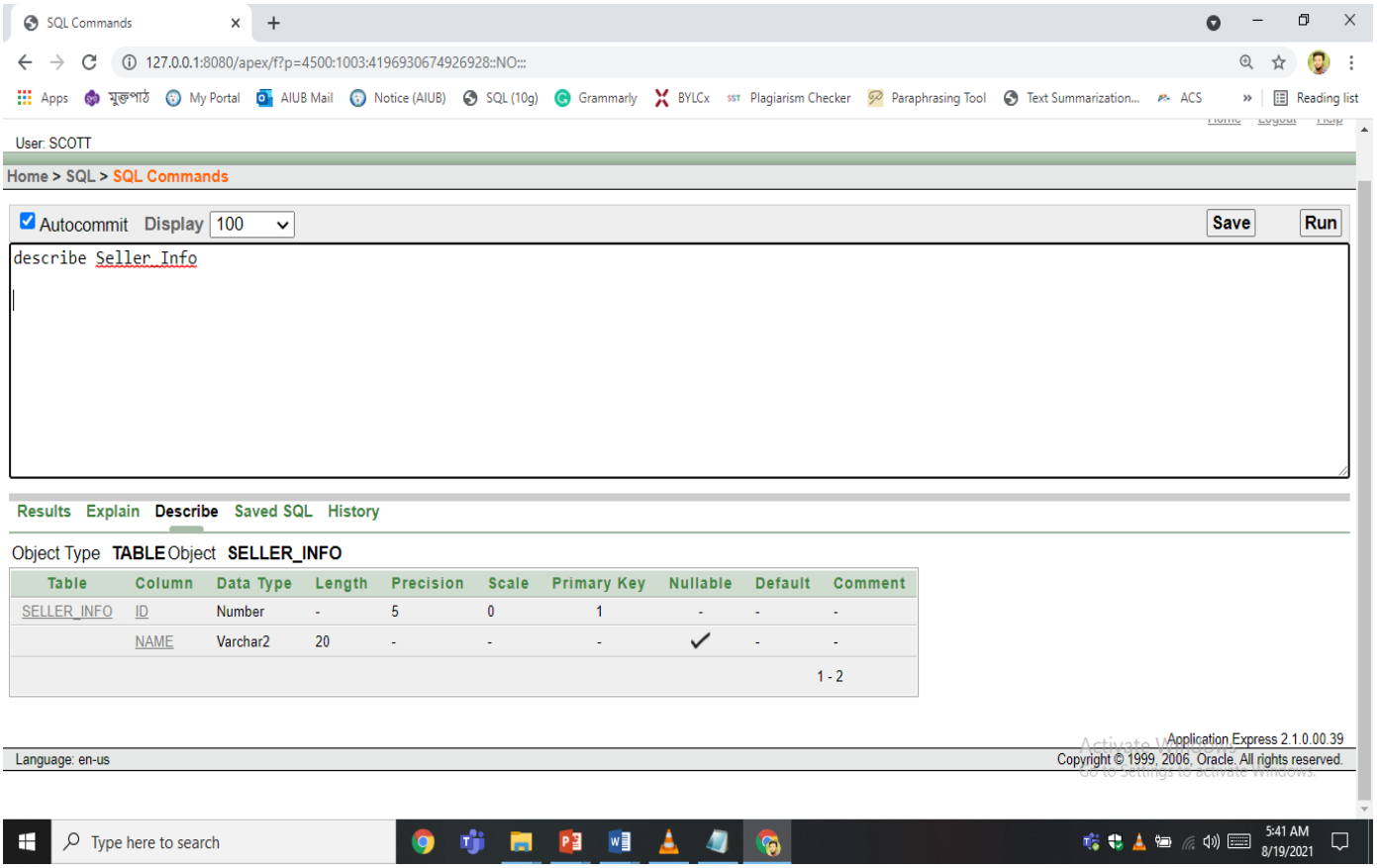
Screenshot:



06. Seller_Info:

```
CREATE TABLE Seller_Info
  (ID NUMBER(5) CONSTRAINT PK_ID PRIMARY KEY,
   Name VARCHAR2(20));
```

Screenshot:



Data Insertion

01. Customer_Info:

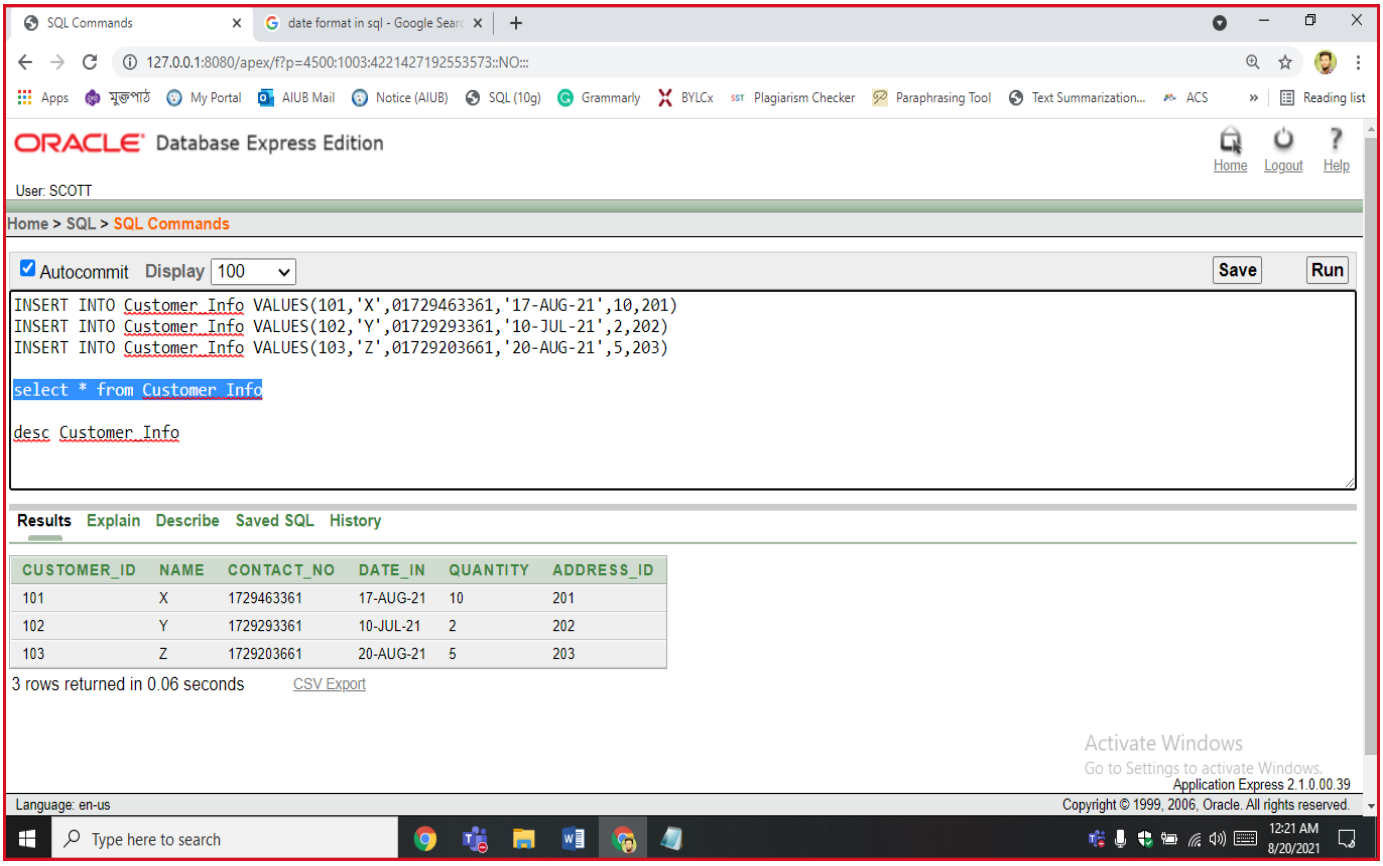
INSERT INTO Customer_Info VALUES(101,'X',01729463361,'17-AUG-21',10,201);

INSERT INTO Customer_Info VALUES(102,'Y',01729293361,'10-JUL-21',2,202);

INSERT INTO Customer_Info VALUES(103,'Z',01729203661,'20-AUG-21',5,203);

SELECT * FROM Customer_Info;

Screenshot:



02. Customer_Address_Details:

INSERT INTO Customer_Address_Details VALUES(201,04,21,'BANGLADESH');

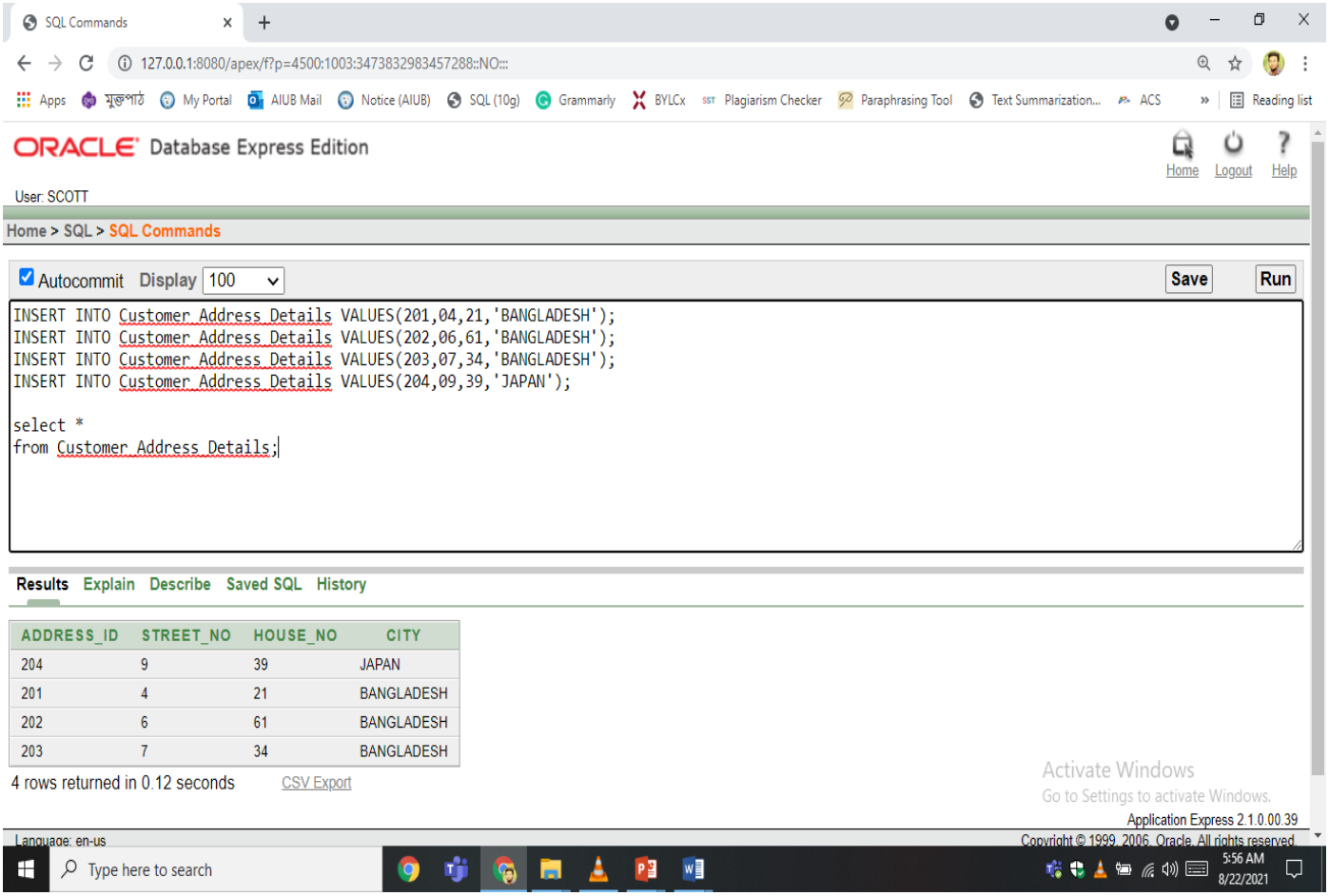
INSERT INTO Customer_Address_Details VALUES(202,06,61,'BANGLADESH');

INSERT INTO Customer_Address_Details VALUES(203,07,34,'BANGLADESH');

INSERT INTO Customer_Address_Details VALUES(204,09,39,'JAPAN');

SELECT * FROM Customer_Address_Details;

Screenshot:



03. Plant_Info:

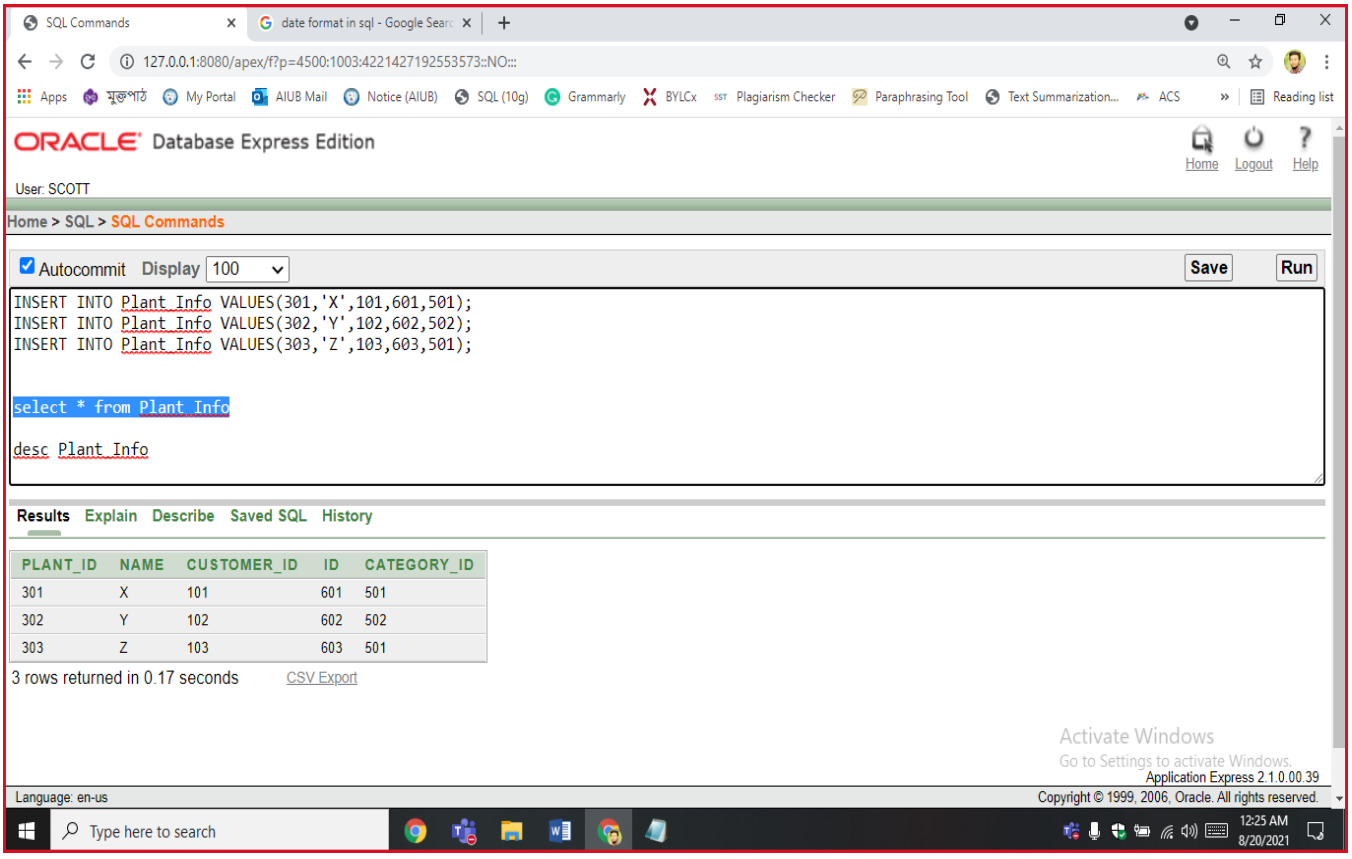
INSERT INTO Plant_Info VALUES(301,'X',101,601,501);

INSERT INTO Plant_Info VALUES(302,'Y',102,602,502);

INSERT INTO Plant_Info VALUES(303,'Z',103,603,501);

SELECT * FROM Plant_Info;

Screenshot:



04. Plant_Size_Info:

INSERT INTO Plant_Size_Info VALUES(301,12);

INSERT INTO Plant_Size_Info VALUES(301,05);

INSERT INTO Plant_Size_Info VALUES(302,10);

INSERT INTO Plant_Size_Info VALUES(302,12);

INSERT INTO Plant_Size_Info VALUES(401,12);

INSERT INTO Plant_Size_Info VALUES(401,05);

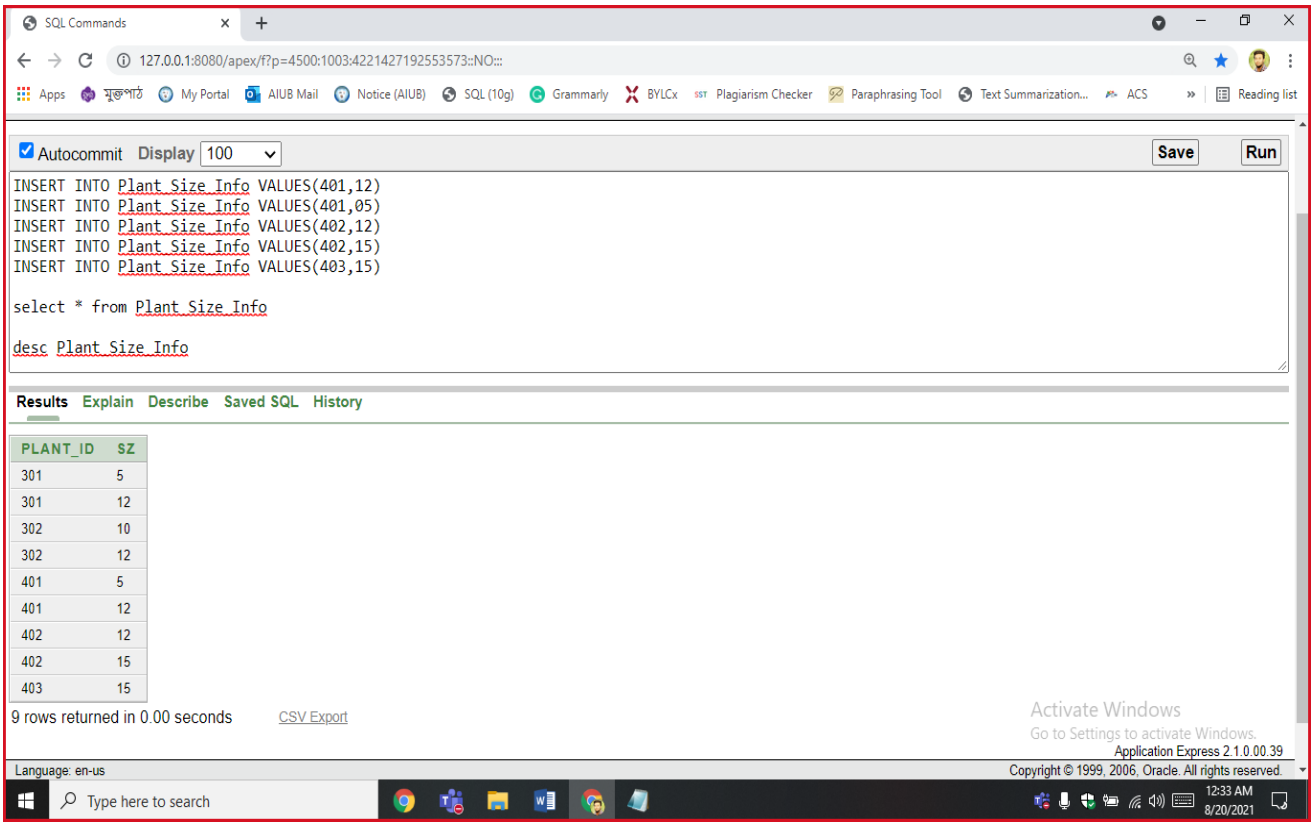
INSERT INTO Plant_Size_Info VALUES(402,12);

INSERT INTO Plant_Size_Info VALUES(402,15);

INSERT INTO Plant_Size_Info VALUES(403,15);

SELECT * FROM Plant_Size_Info;

Screenshot:



05. Category_Details:

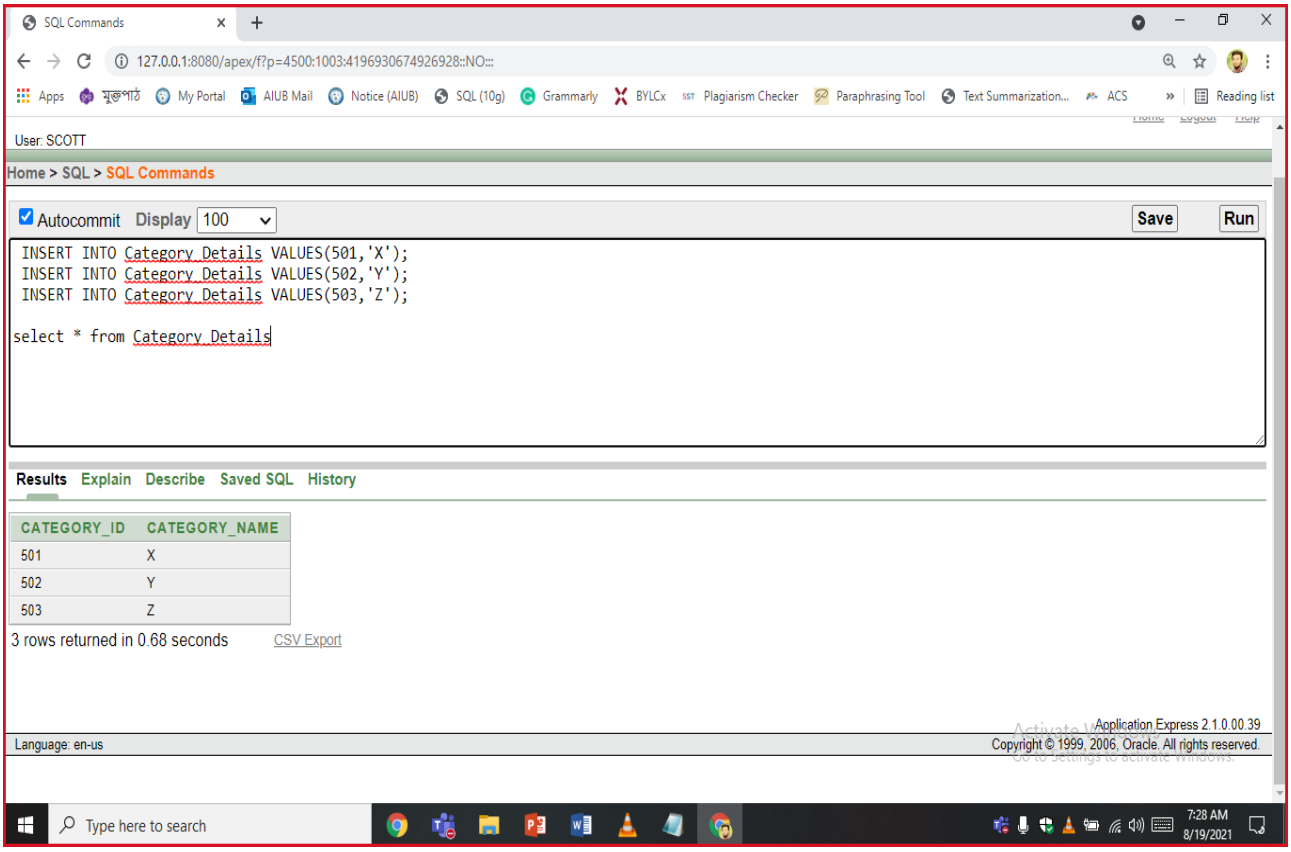
INSERT INTO Category_Details VALUES(501,'X');

INSERT INTO Category_Details VALUES(502,'Y');

INSERT INTO Category_Details VALUES(503,'Z');

SELECT * FROM Category_Details;

Screenshot:



06. Seller_Info

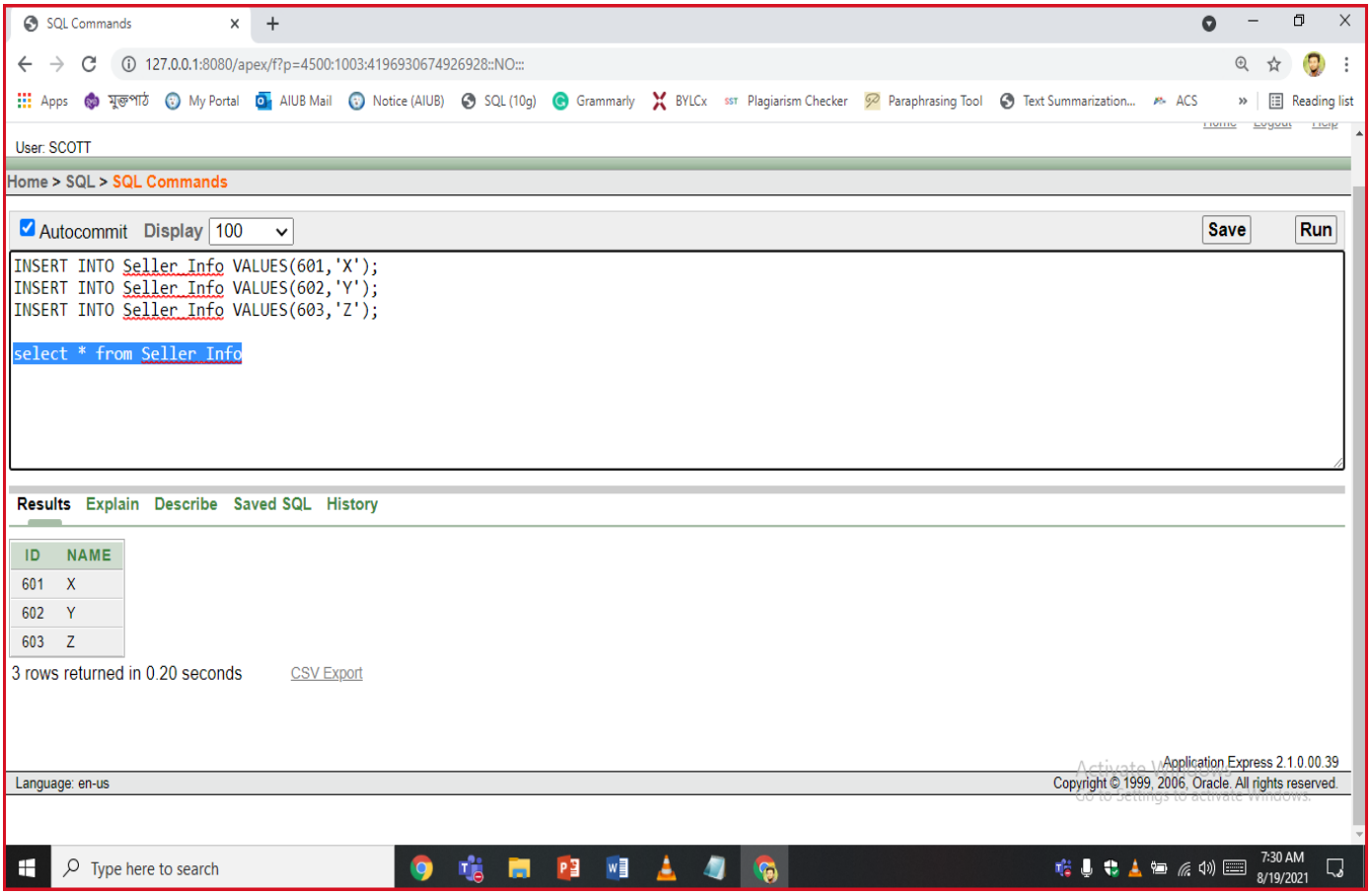
INSERT INTO Seller_Info VALUES(601,'X');

INSERT INTO Seller_Info VALUES(602,'Y');

INSERT INTO Seller_Info VALUES(603,'Z');

SELECT * FROM Seller_Info;

Screenshot:



Joining

Equijoin:

Question:

Display plant id, plant name, customer id and plant size.

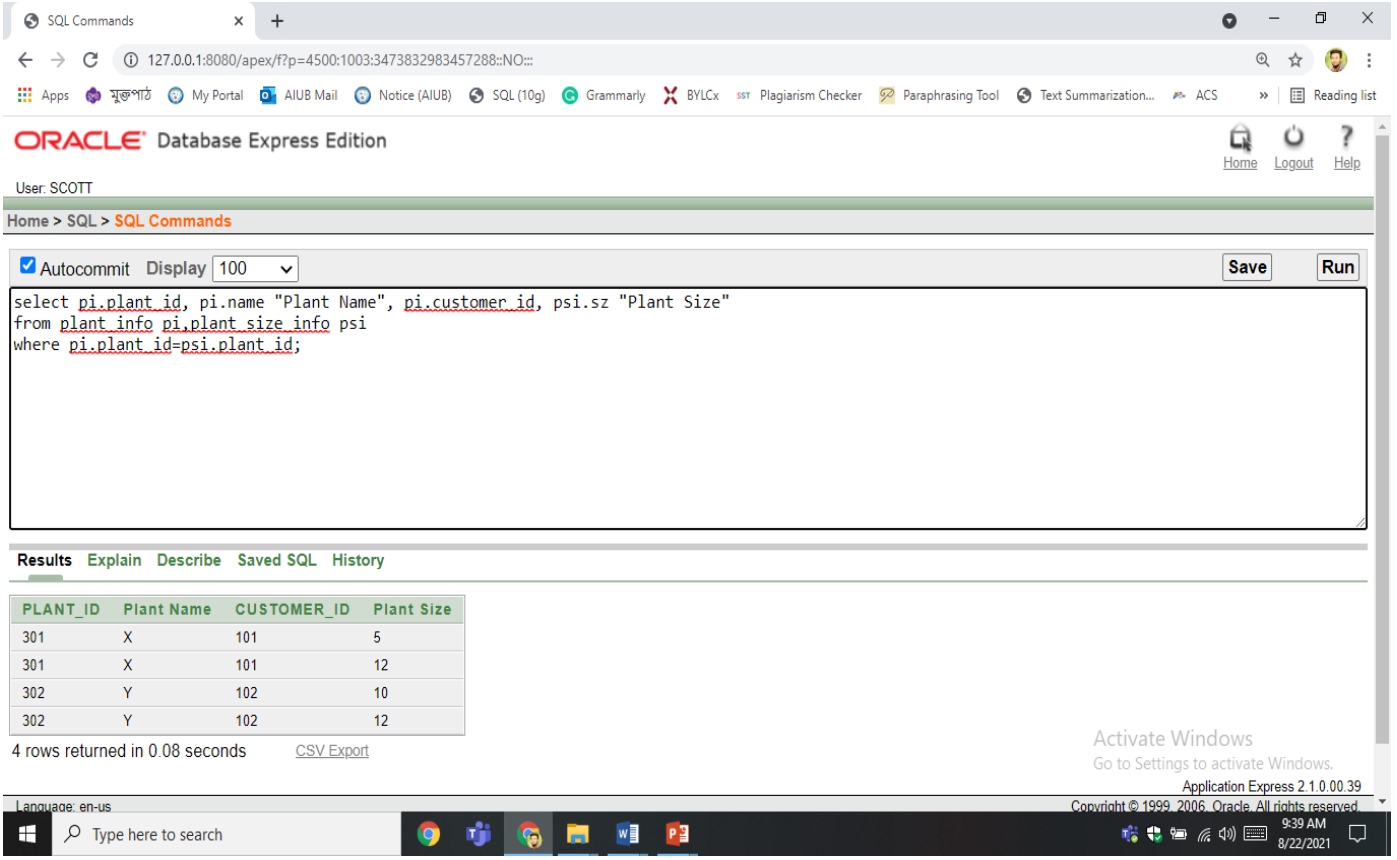
Query:

select pi.plant_id, pi.name "Plant Name", pi.customer_id, psi.sz "Plant Size"

from plant_info pi, plant_size_info psi

where pi.plant_id=psi.plant_id;

Screenshot:



Outer join:

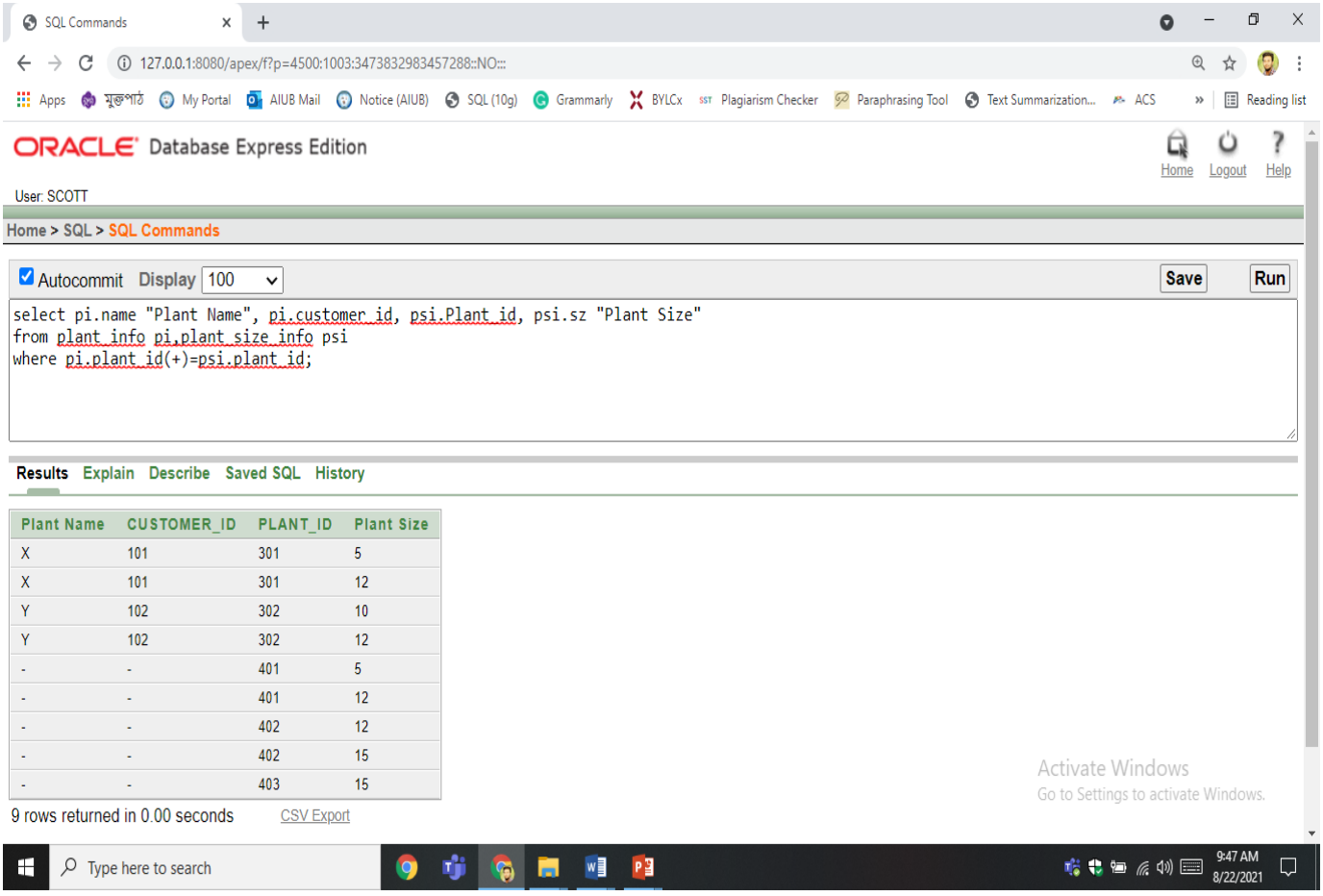
Question:

Display plant name, Customer id, all plant id and all plant size

Query:

```
select pi.name "Plant Name", pi.customer_id, psi.Plant_id, psi.sz "Plant Size"
from plant_info pi,plant_size_info psi
where pi.plant_id(+)=psi.plant_id;
```

Screenshot:



Self-join:

Question:

Display customer id and customer address using self-join

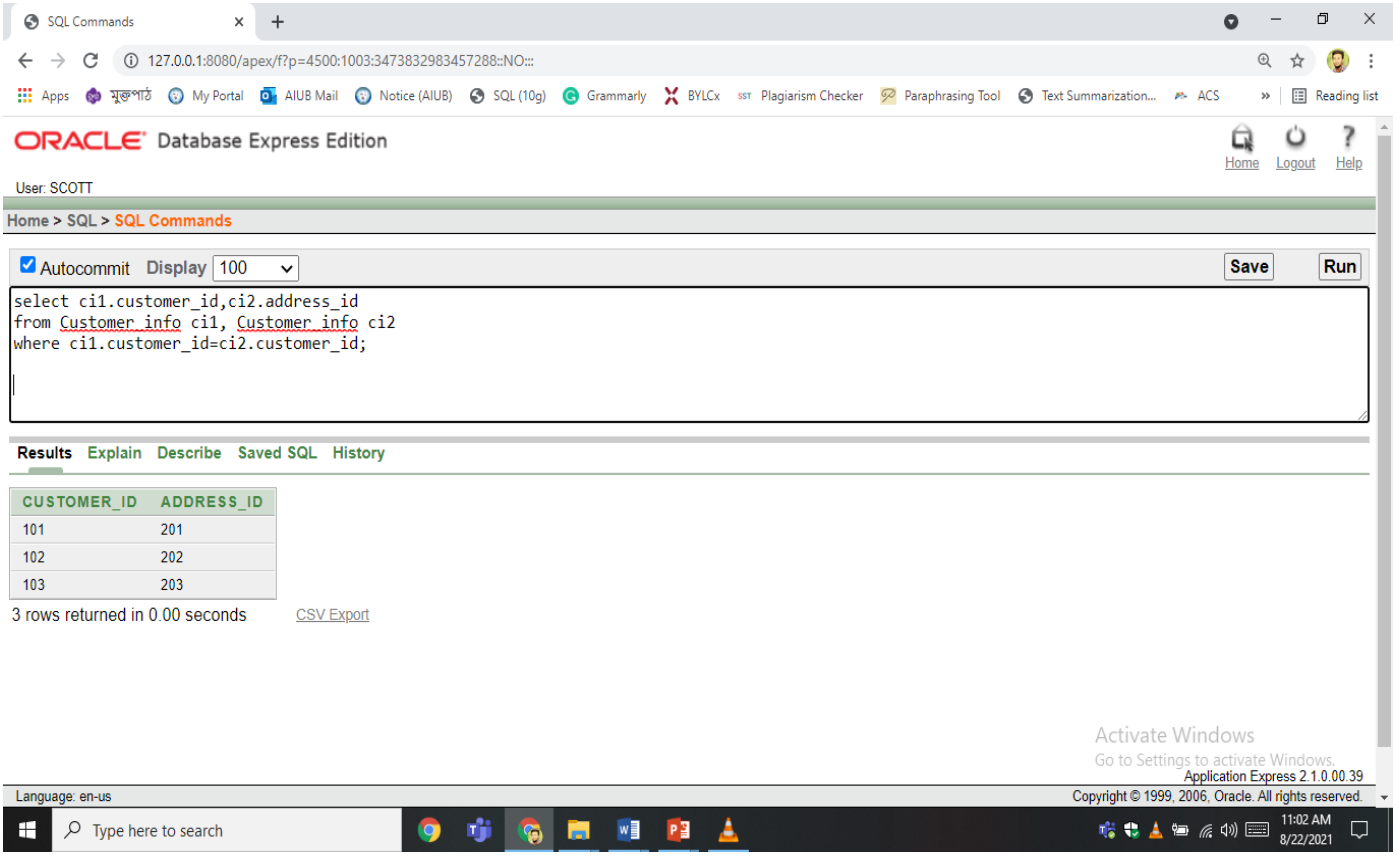
Query:

select ci1.customer_id, ci2.address_id

from Customer_info ci1, Customer_info ci2

where ci1.customer_id=ci2.customer_id;

Screenshot:



Subquery

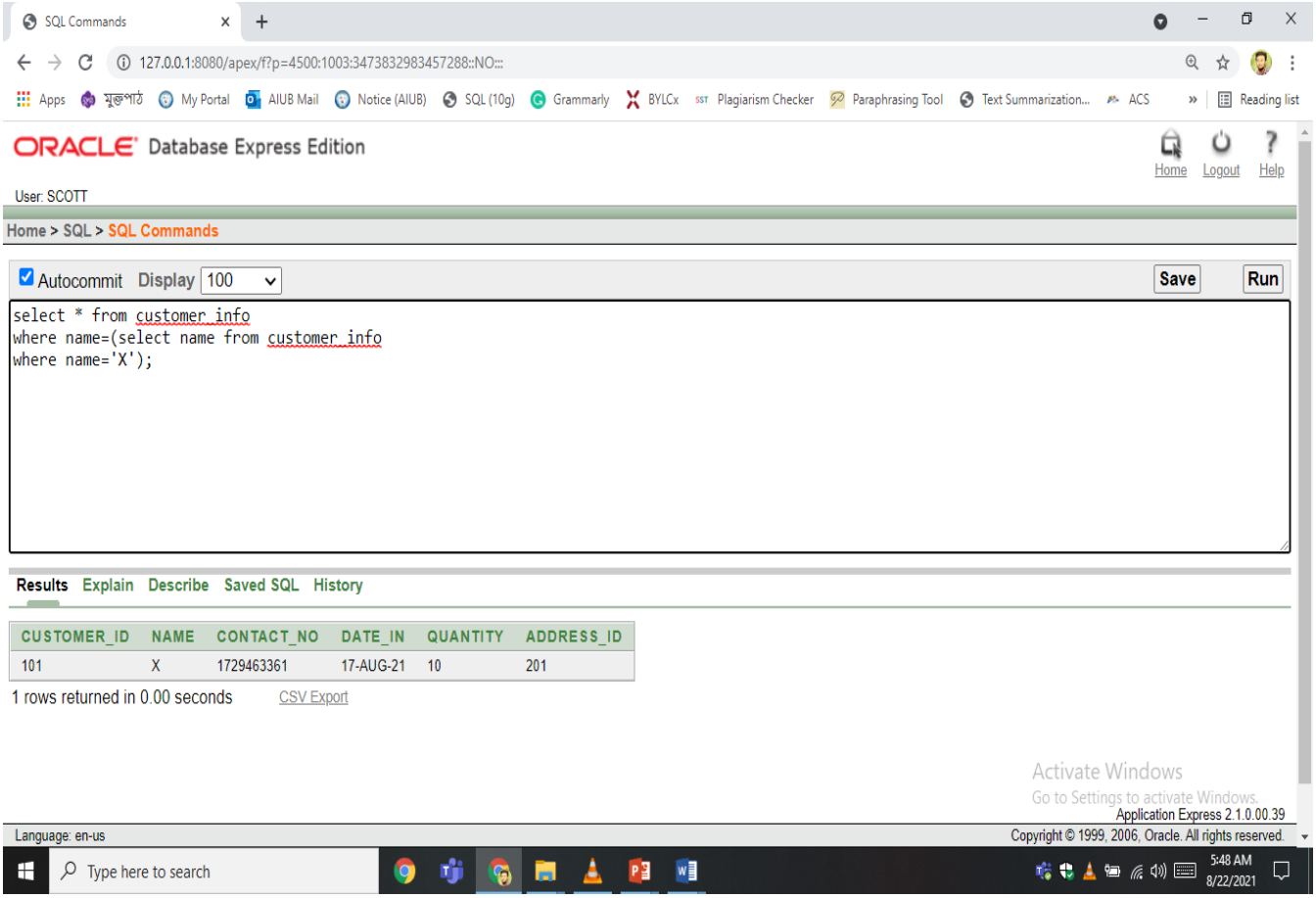
01. Question:

Display all information of X.

Query:

```
select * from customer_info
where name=(select name from customer_info
where name='X');
```

Screenshot:



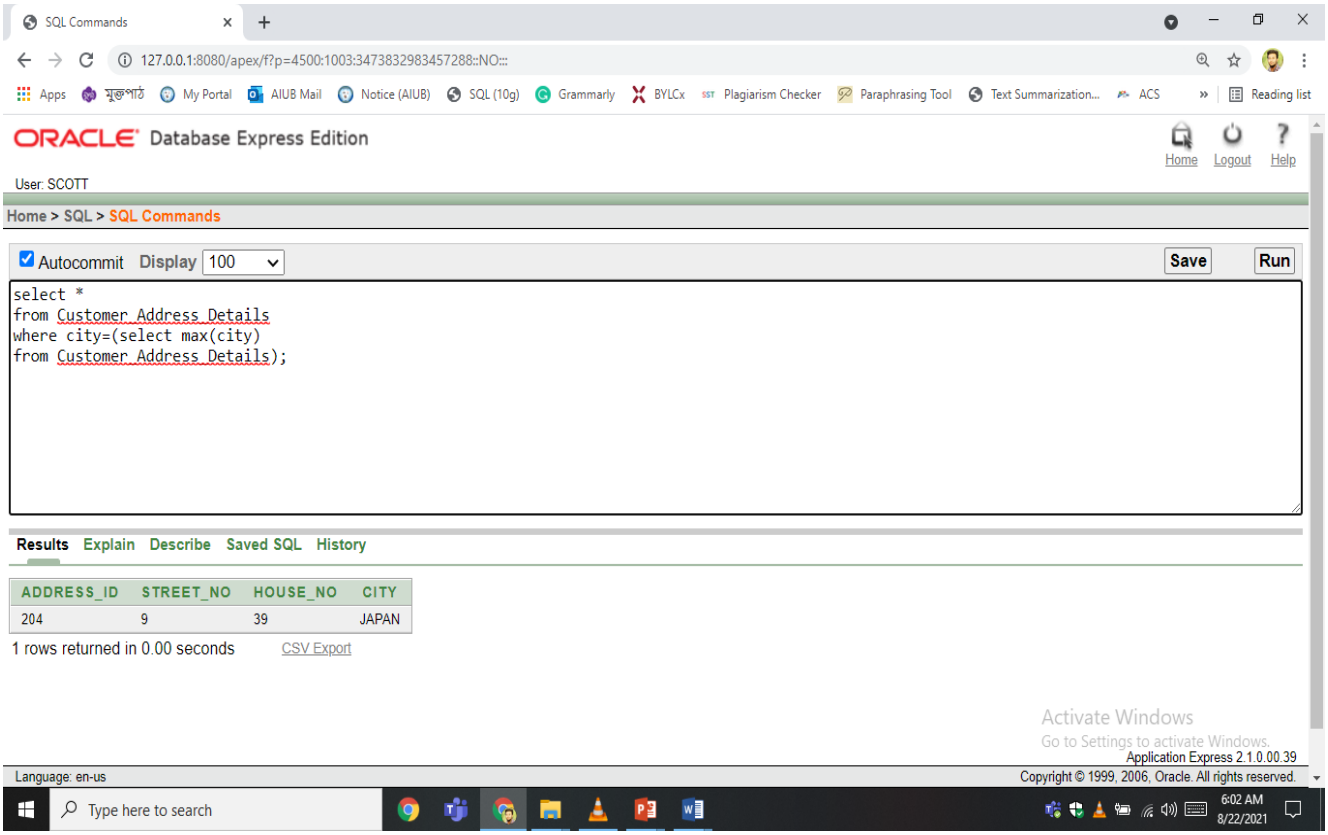
02. Question:

Display Maximum home area.

Query:

```
select *
from Customer_Address_Details
where city=(select max(city)
from Customer_Address_Details);
```

Screenshot:



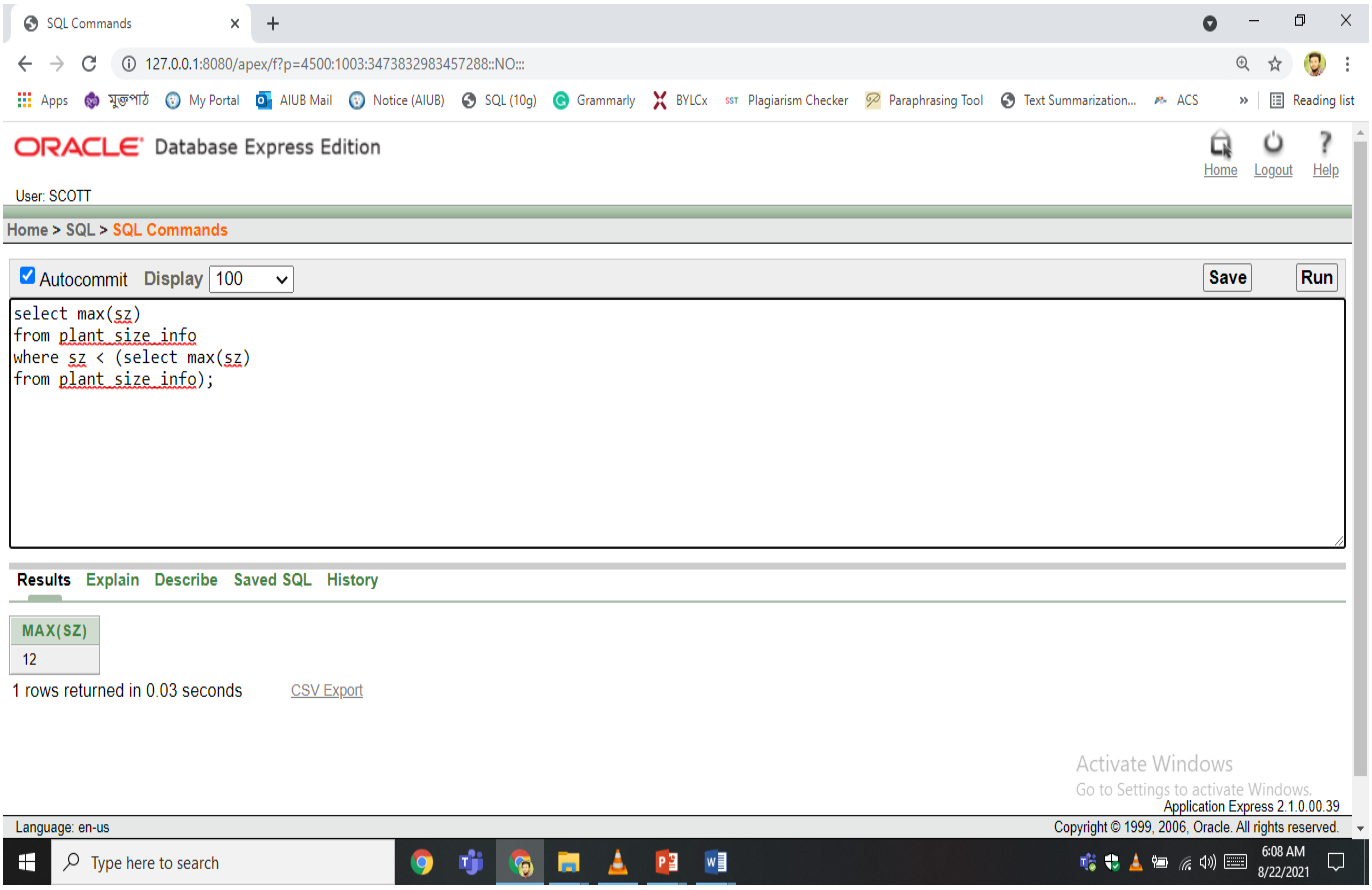
03. Question:

Display second maximum size of plant.

Query:

```
select max(sz)
from plant_size_info
where sz < (select max(sz)
from plant_size_info);
```

Screenshot:



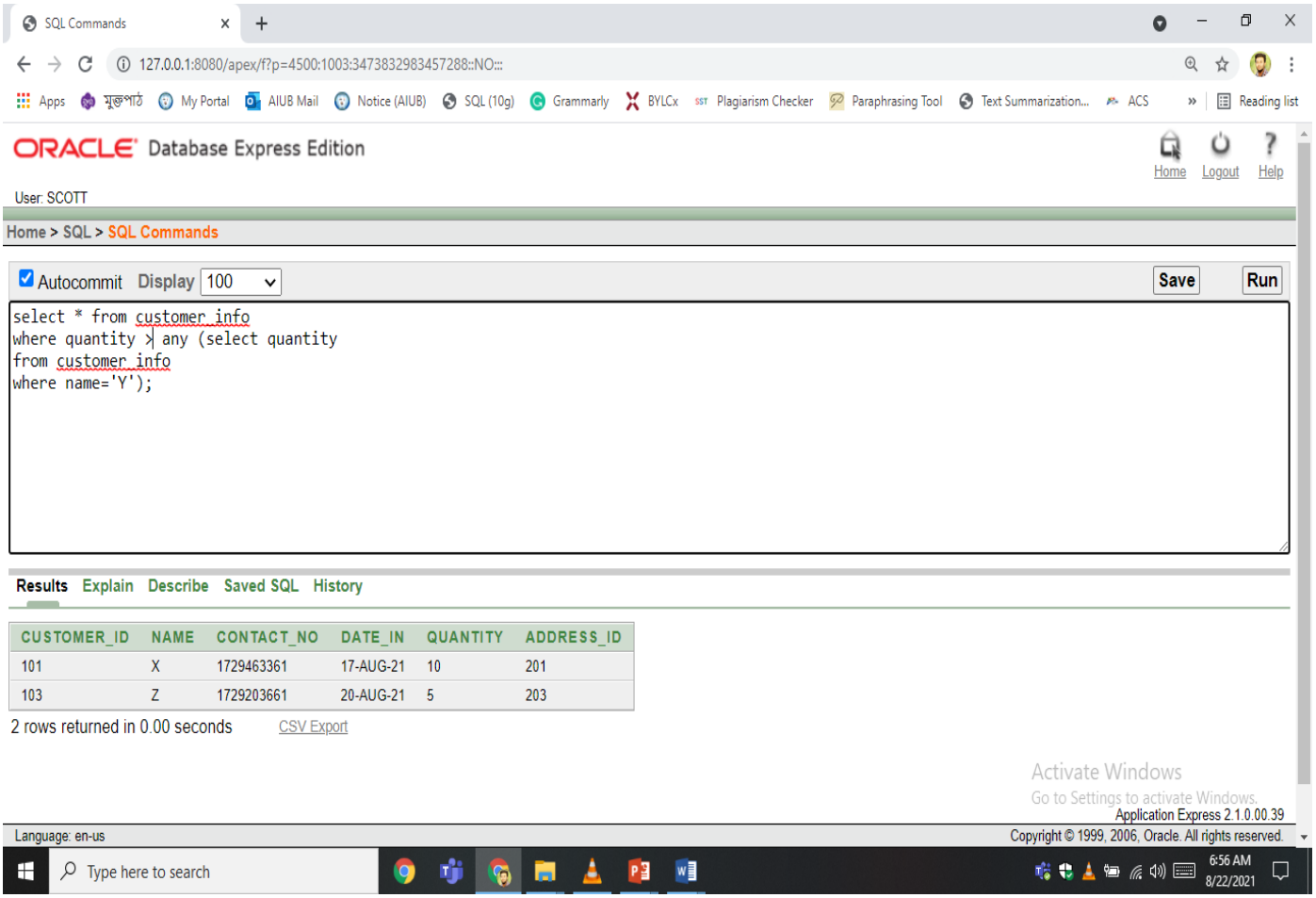
04. Question:

Display all information of those customer who bought more plant than Y.

Query:

```
select * from customer_info
where quantity > any (select quantity
from customer_info
where name='Y');
```

Screenshot:



View (Complex view)

Question:

Create a view of customer id, customr name and customer address from customer_info and customer_address_info table

Query:

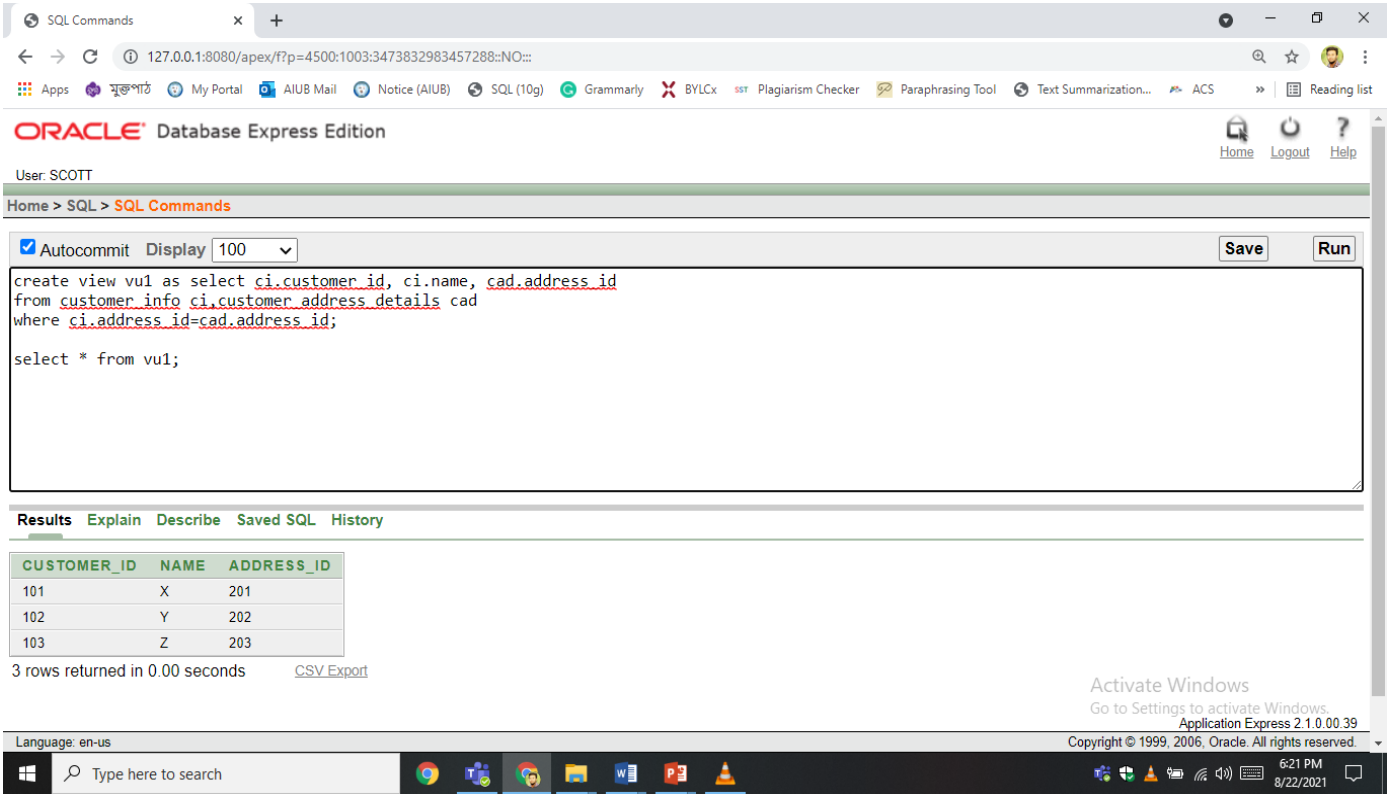
create view vu1 as select ci.customer_id, ci.name, cad.address_id

from customer_info ci,customer_address_details cad

where ci.address_id=cad.address_id;

select * from vu1;

Screenshot:



Add a constraint

Question:

Make street no unique.

Query:

alter table customer_address_details add constraint uq unique(street_no);

describe customer_address_details;

Screenshot: