AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Course: Introduction to Database

Project: Food Delivery & Management System

Table of Contents

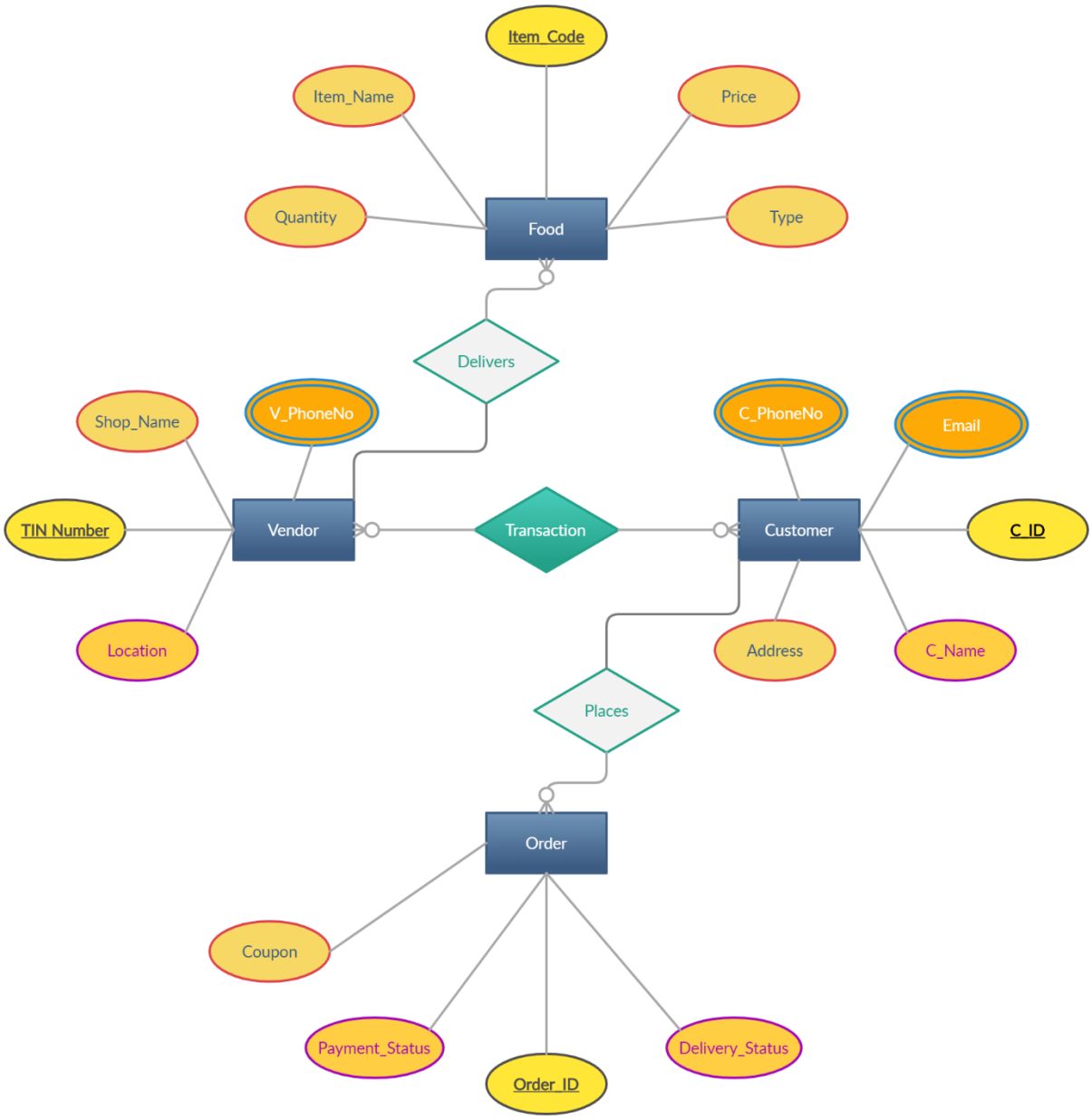
Introduction

Food delivery management system is a technique of ordering foods in online and applicable in any food delivery industry. The main advantage of this system is that it greatly simplifies the ordering process for both the customer and restaurant. This application helps the restaurant to do all functionalities more accurately and faster way. Food delivery management system reduces manual works and improves efficiency of restaurant. This software helps food orders to maintain day to day records in system.

Scenario Description

In a food delivery management system, the customer places their order staying at home or anywhere and delivery man deliver the food into the customer address. After receiving the order customer pays bill. Both the customer and the delivery man is identified by their Id. The system holds the basic information for both the Customer and Vendor; like name, phone number, address etc. So the relationship between the customer and the vendor is transaction. It also holds special attributes for each entity. When a customer replaces the order it holds key information of his or her in "Customer" entity. The system collects information like customer id, name, email, address, phone number. The vendor may also have the information is TIN number, Shop name, Location, phone number. When the customer replaces the order it holds the information about "Order" where coupon , payment , order id , delivery status are the attributes. Payment attribute holds cash on delivery and online payment. Then delivery status holds departure time and arrival time as attributes .As order entity has unique key calls order Id which describes that ever customer has unique id. Here customer and order has many to many relationship or called "places" Vendor delivers food where food system holds the information of item code, item name, type, quantity and price. Here vendor and food has one to many relationship called "Delivers". There are some special attributes are called derived attributes like Phone number, email address. In our food delivery management system, ER-Diagram we have showed 2 relationships which are one to many and many to many, then we've also showed 4 entities which are Vendors, Customer, Order and Food.

ER-Diagram


Item_Code
Item_Name
Price
Quantity
Food
Type
Delivers
Shop_Name
V_PhoneNo
C_PhoneNo
Email
TIN Number
Vendor
Transaction
Customer
C_ID
Location
Address
C_Name
Places
Order
Coupon
Payment_Status
Order_ID
Delivery_Status

Normalization

Delivers:

UNF:

delivers(TIN_Number, Shop_Name, V_PhoneNo,
Location, Item_Code, Item_Name, Price, Quantity, Type)

1NF:

Here, V_Phone is multi valued attribute.

1. TIN_Number, Shop_Name, V_PhoneNo, Location, Item_Code, Item_Name, Price, Quantity, Type

2NF:

1. TIN_Number, Shop_Name, V_PhoneNo, Location

2. Item_Code, Item_Name, Price, Quantity, Type

3NF:

1. TIN_Number, Shop_Name, V_PhoneNo, Location

2. Item_Code, Type

3. Item_Name, Price, Quantity

Table Creation:

1. TIN_Number, Shop_Name, V_PhoneNo, Location

2. Item_Code, Type, **Price_ID, TIN_Number**

3. Price_ID, Item_Name, Price, Quantity

Places:

UNF:

places(C_ID, C_Name, Email, C_PhoneNo,
Address, Order_ID, Delivery_Status, Payment_Status, Coupon)

1NF:

Here, Email and C_PhoneNo are multi valued attribute.

1. C_ID, C_Name, Email, C_PhoneNo,
Address, Order_ID, Delivery_Status, Payment_Status, Coupon

2NF:

1. C_ID, C_Name, Email, C_PhoneNo, Address

2. Order_ID, Delivery_Status, Payment_Status, Coupon

3NF:

1. C_ID, C_Name, Email, C_PhoneNo, Address

2. Order_ID, Coupon

3. Delivery_Status, Payment_Status

Table Creation:

1. C_ID, C_Name, Email, C_PhoneNo, Address

2. Order_ID, Coupon, **C_ID**

3. Delivery_ID, Delivery_Status, Payment_Status

Transaction:

UNF:

transaction(TIN_Number, Shop_Name, V_PhoneNo, Location, C_ID, C_Name,
Email, C_PhoneNo, Address)

1NF:

Here, V_PhoneNo, C_PhoneNo, Email are multi valued attribute.

1. TIN_Number, Shop_Name, V_PhoneNo, Location, C_ID, C_Name,
Email, C_PhoneNo, Address

2NF:

1. TIN_Number, Shop_Name, V_PhoneNo, Location

2. C_ID, C_Name, Email, C_PhoneNo, Address

3NF:

There is no transitive dependency. Relation already in 3NF

1. TIN_Number, Shop_Name, V_PhoneNo, Location

2. C_ID, C_Name, Email, C_PhoneNo, Address

Table Creation:

1. TIN_Number, Shop_Name, V_PhoneNo, Location

2. C_ID, C_Name, Email, C_PhoneNo, Address, **TIN_Number**

Temporary Table:

1. ~~. TIN_Number, Shop_Name, V_PhoneNo, Location~~

2. Item_Code, Type, **Price_ID, TIN_Number**

3. Price_ID, Item_Name, Price, Quantity

4. ~~C_ID, C_Name, Email, C_PhoneNo, Address~~

5. Order_ID, Coupon, **C_ID**

6. Delivery_ID, Delivery_Status, Payment_Status

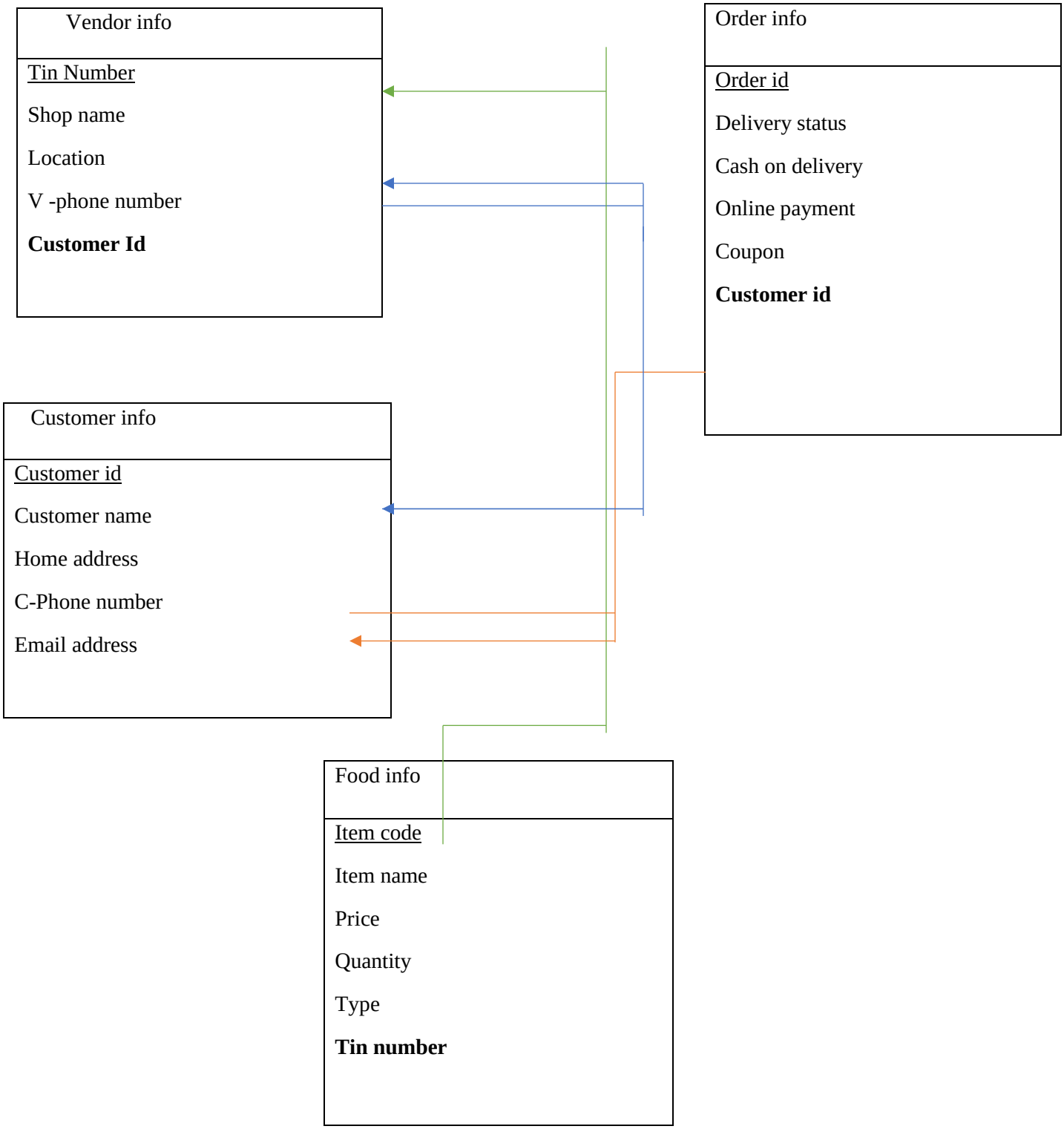
7. TIN_Number, Shop_Name, V_PhoneNo, Location

8. C_ID, C_Name, Email, C_PhoneNo, Address, **TIN_Number**

Final Table:

1. TIN_Number, Shop_Name, V_PhoneNo, Location- tin_info
2. C_ID, C_Name, Email, C_PhoneNo, Address, **TIN_Number**- customer_info
3. Item_Code, Type, **Price_ID, TIN_Number**- item
4. Order_ID, Coupon, **C_ID- order_info**
5. Delivery_ID, Delivery_Status, Payment_Status- delivery_info
6. Price_ID, Item_Name, Price, Quantity- item_info

Schema Diagram:

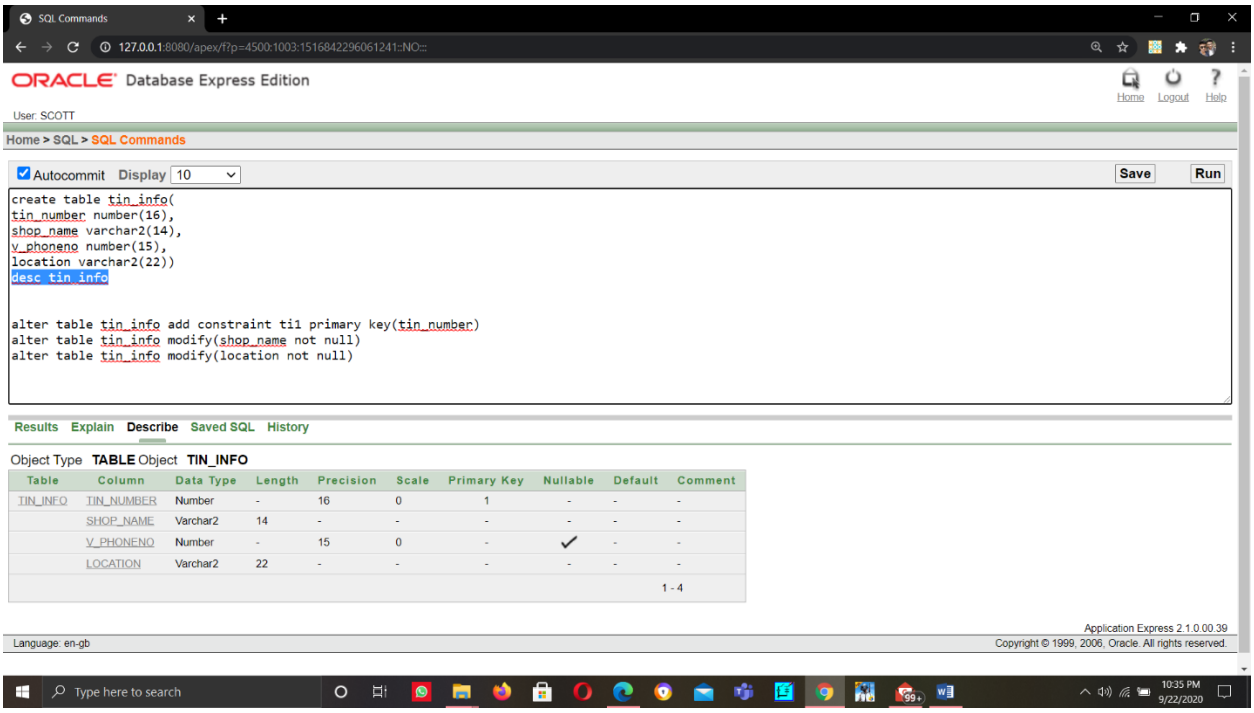


*Arrows point from foreign key to corresponding primary key.

Table Creation

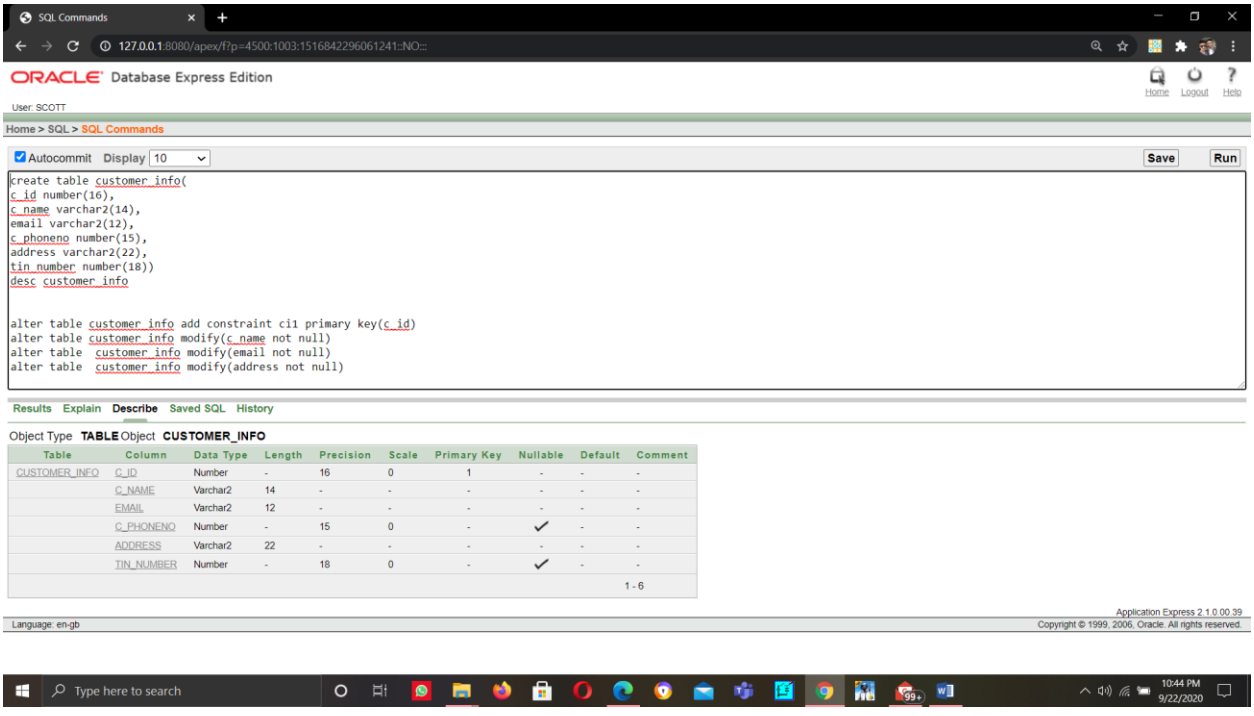
```
create table tin_info(
tin_number number (16),
shop_name varchar2(14),
v_phoneno number(15),
location varchar2(22))
desc tin_info

alter table tin_info add constraint ti1 primary key(tin_number)
alter table tin_info modify ( shop_name not null)
alter table tin_info modify (location not null)
```



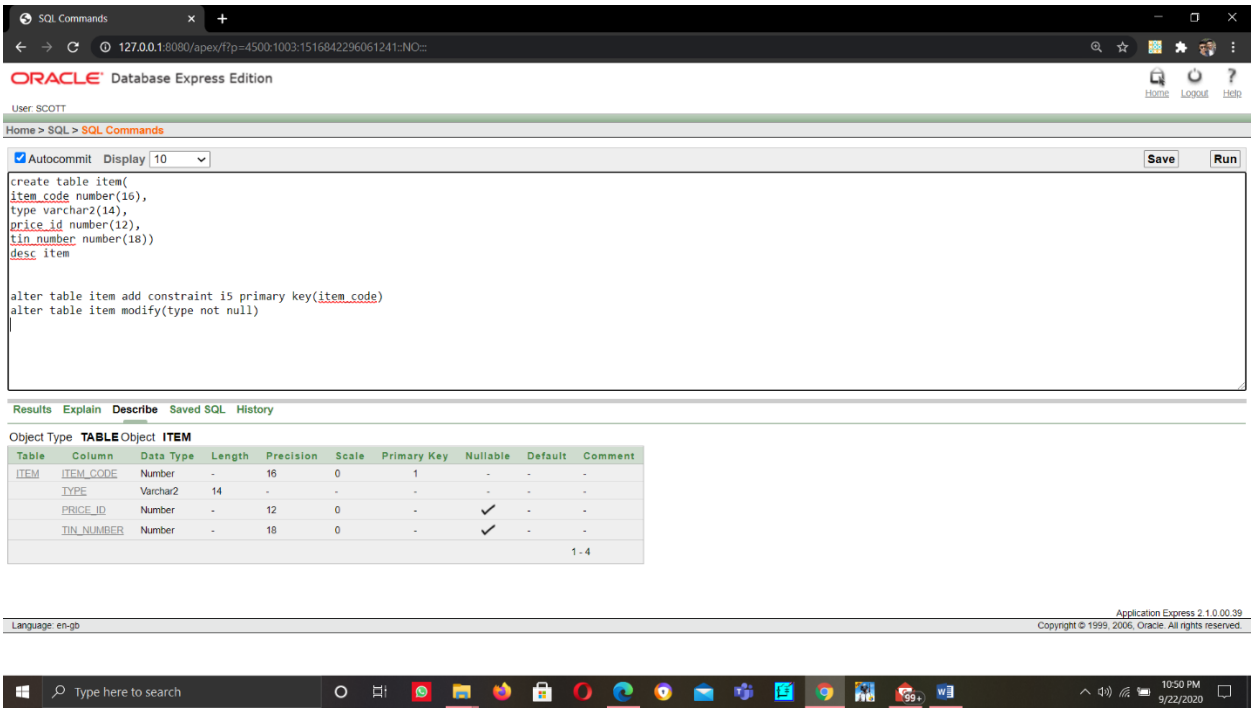
```
create table customer_info(
c_id number (16),
c_name varchar2(14),
email varchar2(12),
c_phoneno number(15),
address varchar2(22)
tin_number number(18))
desc customer_info


alter table customer_info add constraint ci1 primary key(c_id)
alter table customer_info modify ( c_name not null)
alter table customer_info modify (email not null)
alter table customer_info modify (address not null)
```

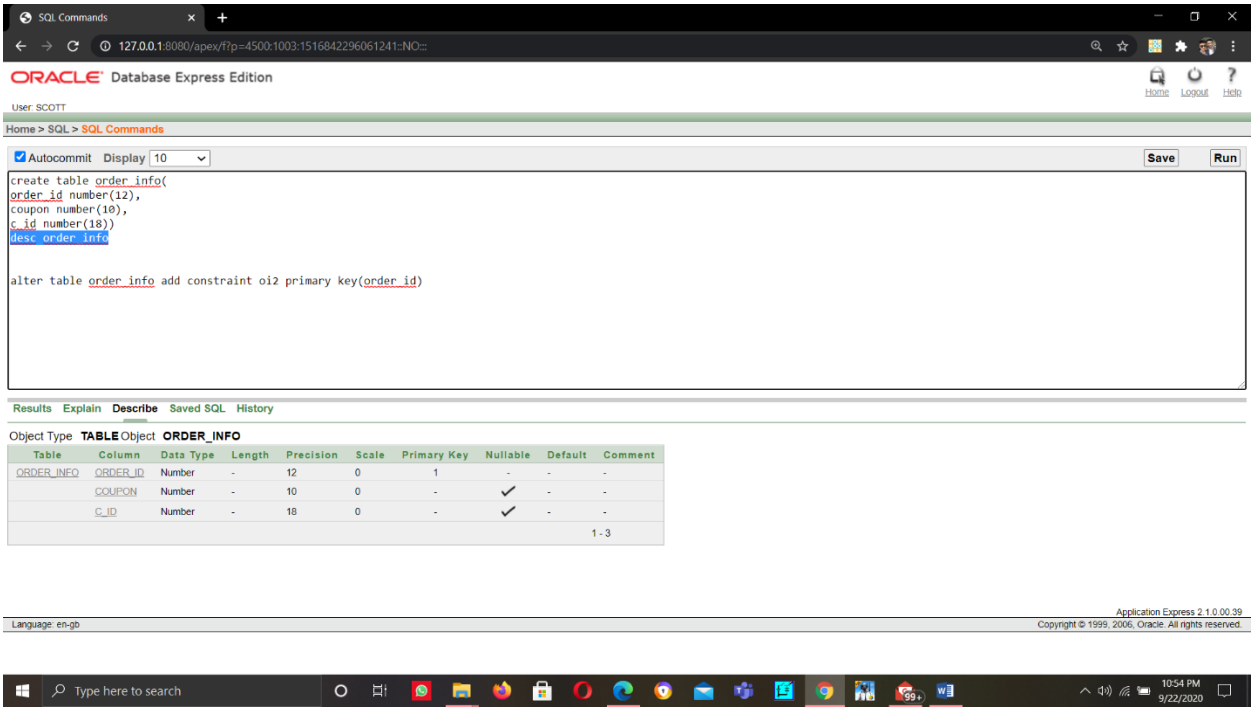


```
create table item(
item_code number(16),
type varchar2(14),
price_id number(12),
tin_number number(18))
desc item
```

```
alter table item add constraint is primary key(item_code)
alter table item modify ( type not null)
```

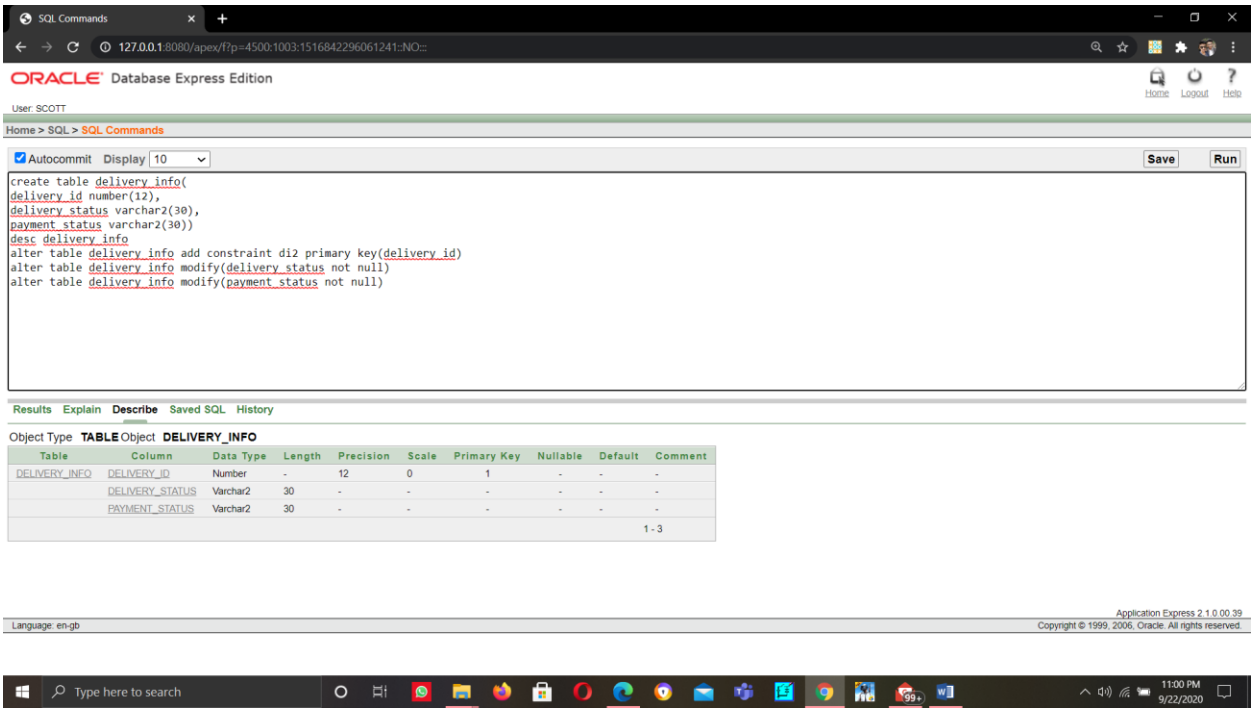


```
create table order_info(
order_id number(12),
coupon number(10),
c_id number(18)
desc order_info
alter table order_info add constraint oi2 primary key(order_id)
```



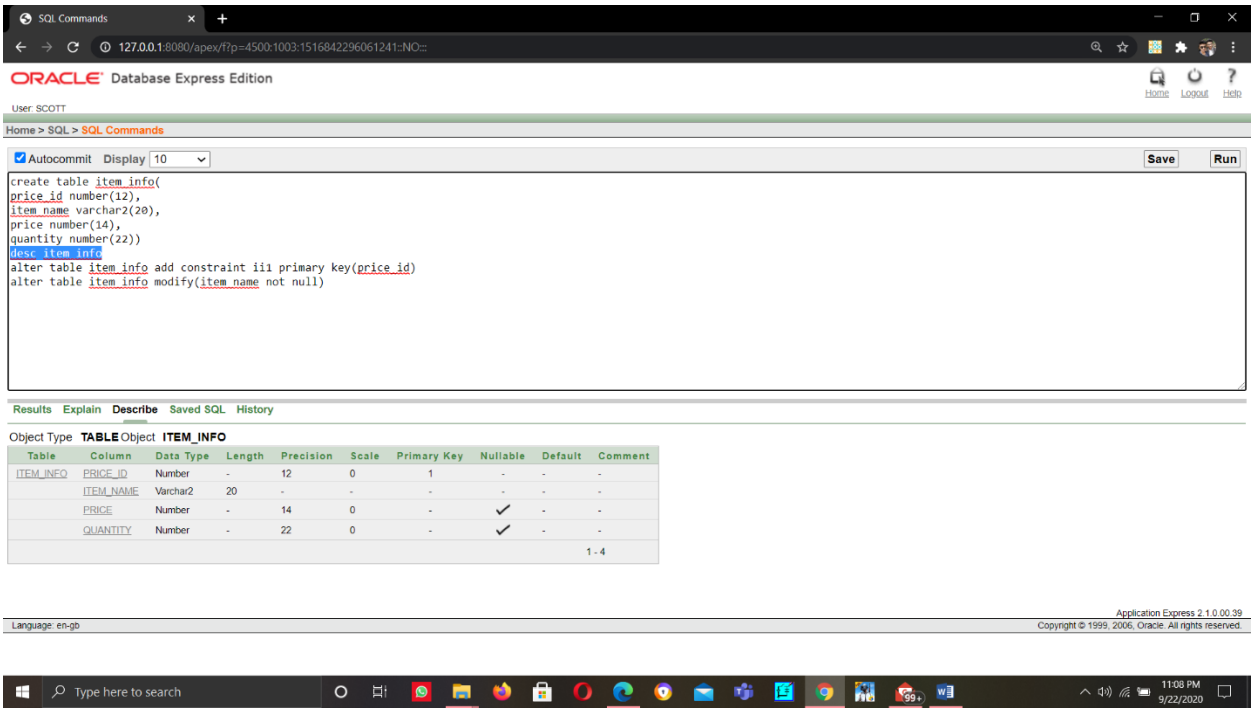
```
create table delivery_info(
delivery_id number(12),
delivery_status varchar2(30),
payment_status varchar2(30),
desc delivery_info

alter table delivery_info add constraint di2 primary key(delivery_id)
alter table delivery_info modify (delivery_status not null)
alter table delivery_info modify(payment_status not null)
```



```
create table item_info(
price_id number(12),
item_name varchar2(20),
price number(14),
quantity number(22))
desc item_info

alter table item_info add constraint ii2 primary key(price_id)
alter table item_info modify (item_name not null)
```

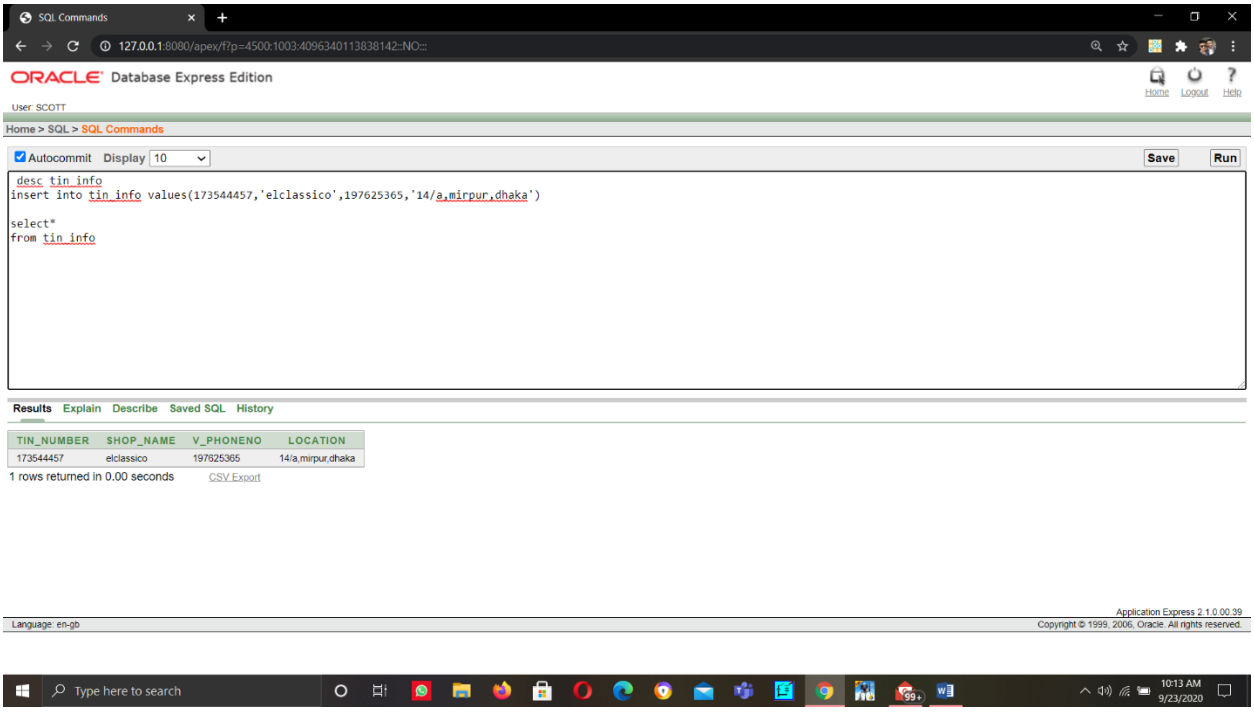


Data Insertion and Display Table

desc tin_info

insert into tin_info values (173544457, ‘elclassico’,197625365, ‘14/a,mirpur,dhaka’)

select* from tin_info



Desc customer_info

insert into customer_info values(615141, ‘shawon’, ‘sha21@gmail’,19854423, ‘12/a mirpur,dhaka’, 173544457)

insert into customer_info values (135641, ‘nisat’,’nis41@gmail’, 17544423,’10/c gulsan, dhaka’, 173544457)

Select* from customer_info

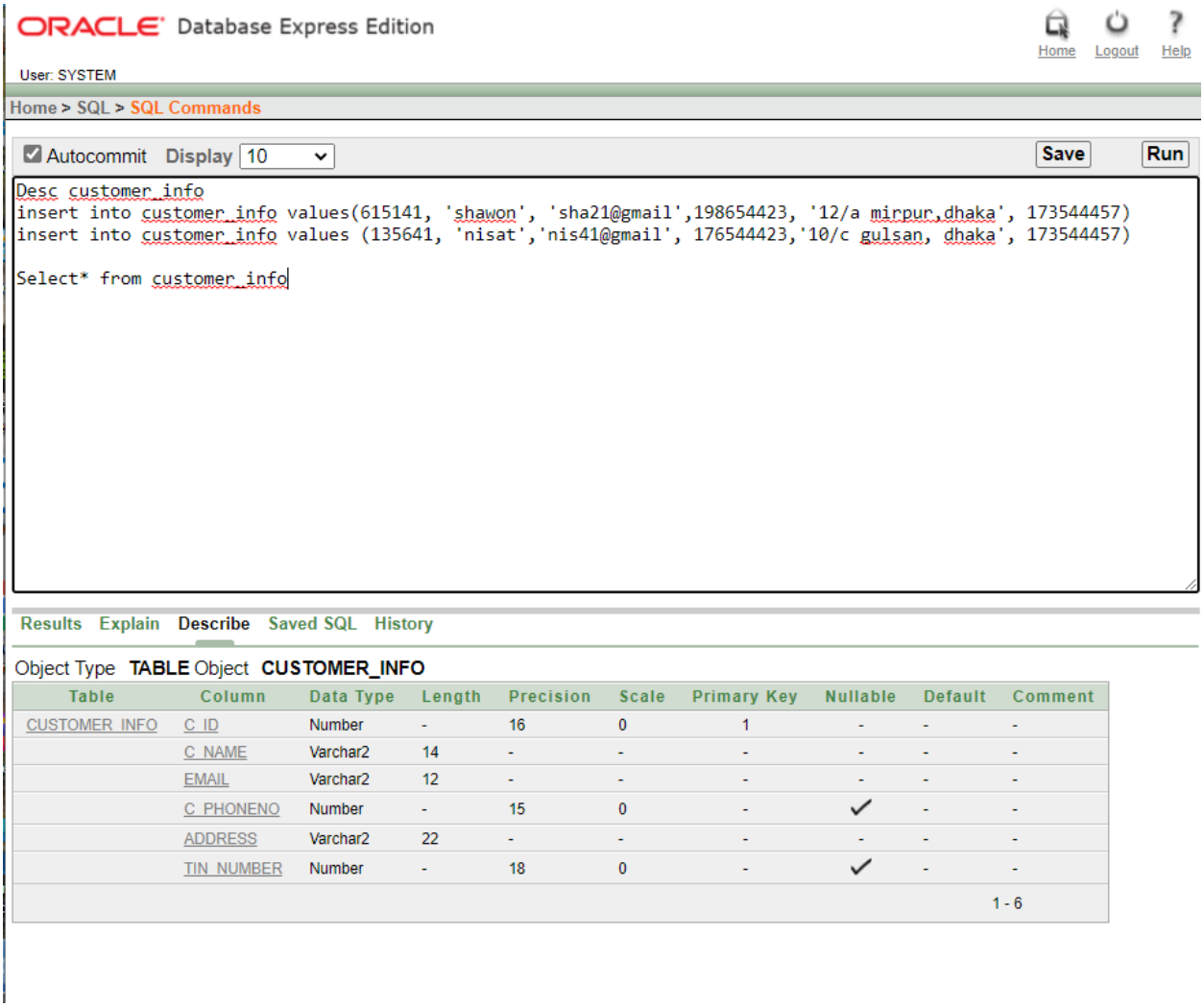


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER_INFO	C_ID	Number	-	16	0	1	-	-	-
	C_NAME	Varchar2	14	-	-	-	-	-	-
	EMAIL	Varchar2	12	-	-	-	-	-	-
	C_PHONENO	Number	-	15	0	-	✓	-	-
	ADDRESS	Varchar2	22	-	-	-	-	-	-
	TIN_NUMBER	Number	-	18	0	-	✓	-	-

1 - 6

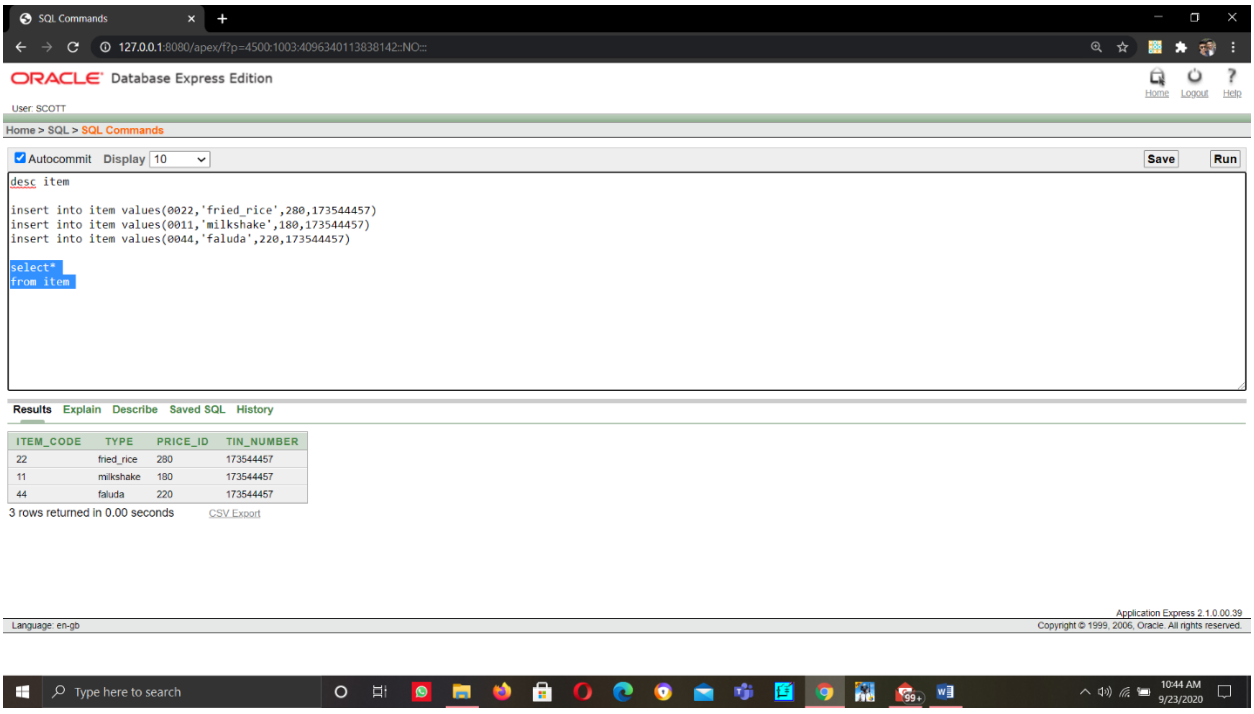
desc item

insert into item values (0022, ‘fried_rice’,280,173544457)

insert into item values(0011, ‘milkshake’, 180,173544457)

insert into item values(0044, ‘faluda’, 220, 173544457)

select* from item

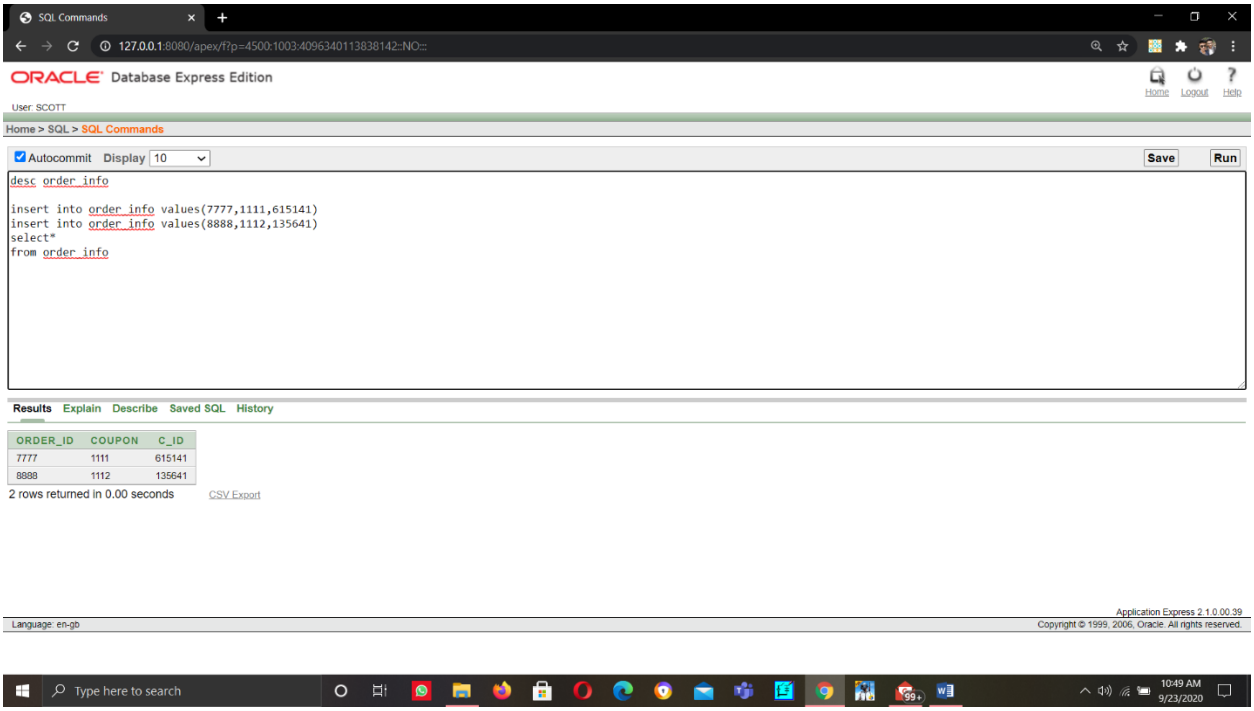


desc order_info

insert into order_info values (7777,1111,615141)

insert into order_info values(8888,1112,135641)

select* from order_info

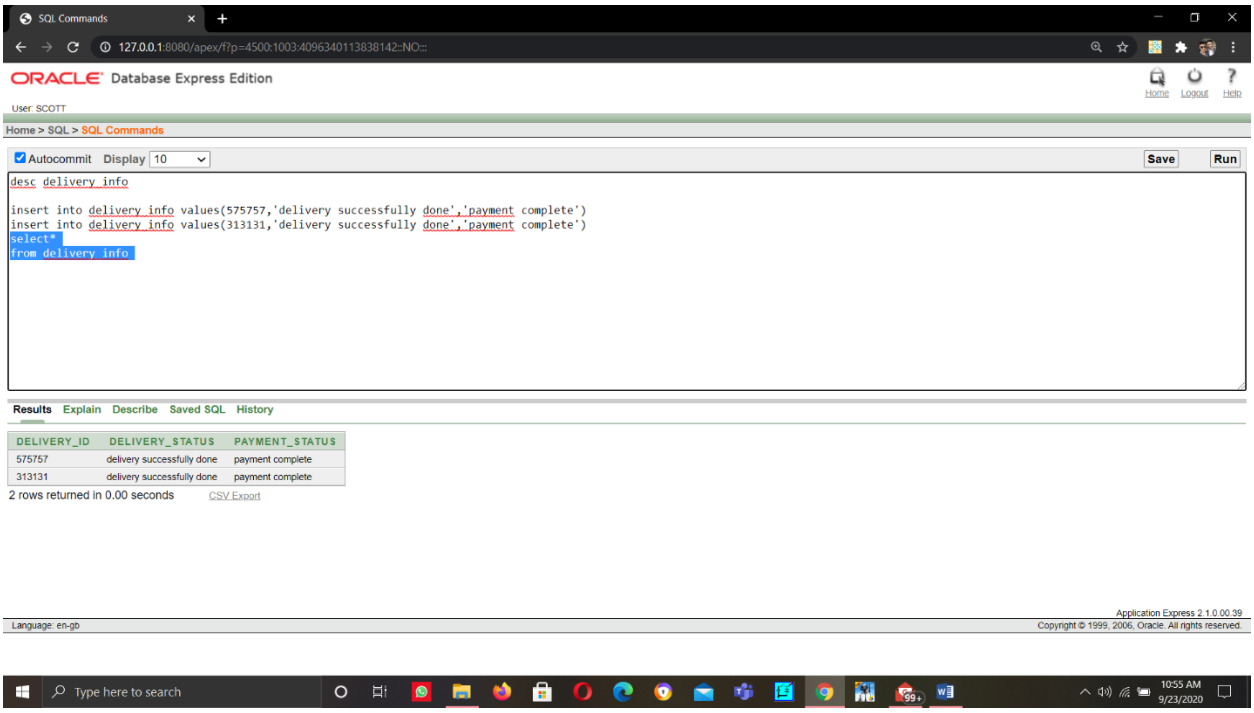


desc delivery_info

insert into delivery_info values (575757, ‘delivery successfully done’,’payment complete’)

insert into delivery_info values (313131, ‘delivery successfully done’,’payment complete’)

Select* from delivery_info



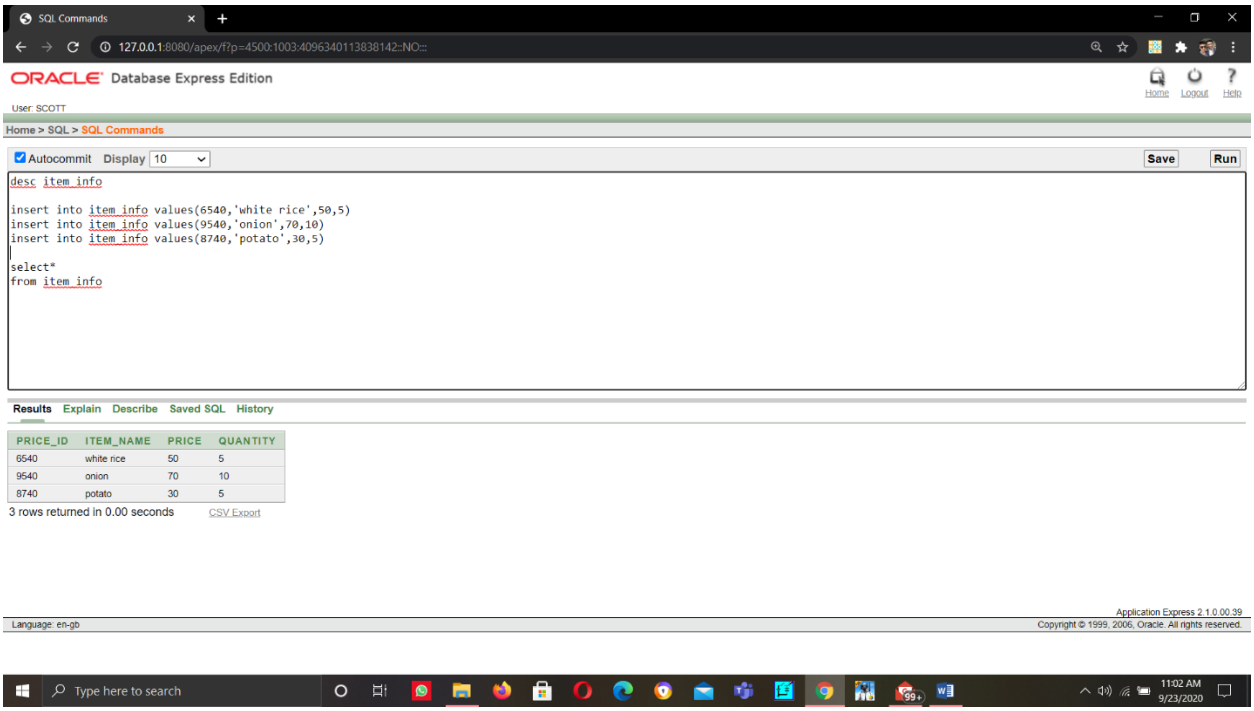
desc item_info

insert into item_info values (6540,’white rice’, 50,5)

insert into item_info values (9540, ‘onion’, 70,5)

insert into item_info values (8740, ‘potato’, 30, 5)

Select* from item_info

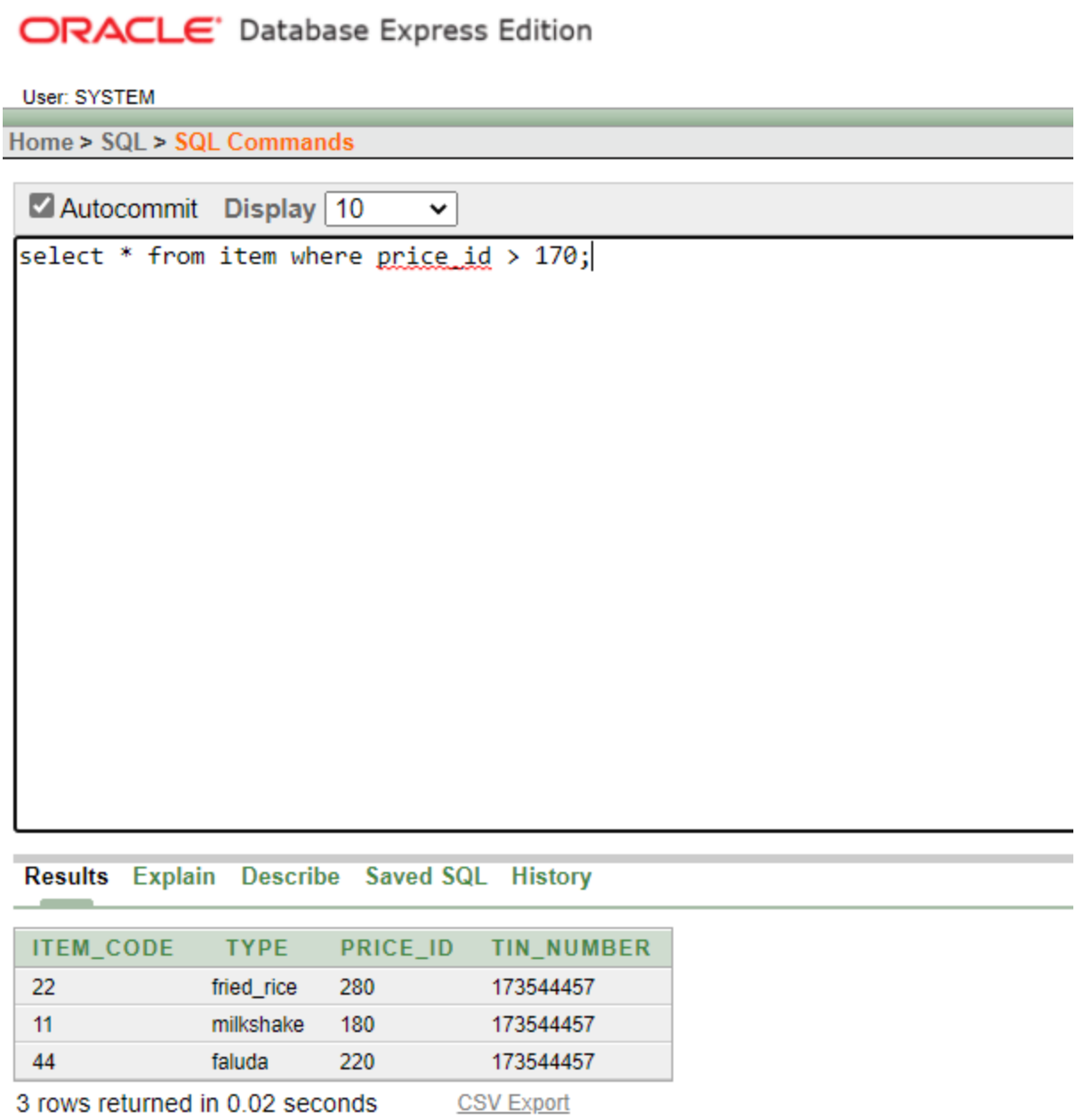


Query Writing

Subquery-

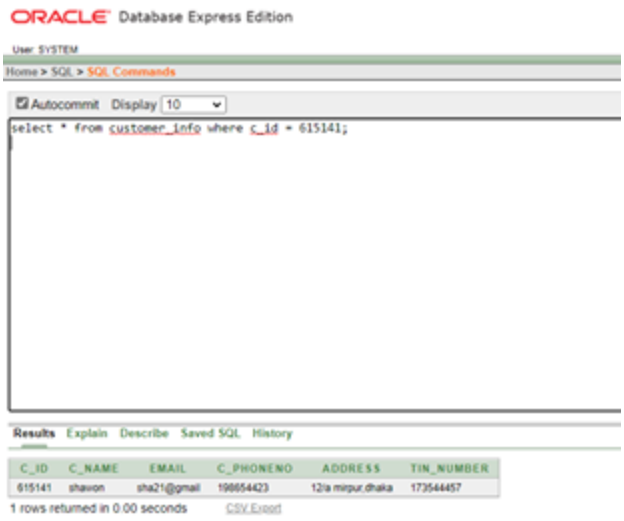
1. Find the item which are expensive than Price_id= 170

Ans: Select * from item where price_id>170;



2.Find the customer whose id is 615141

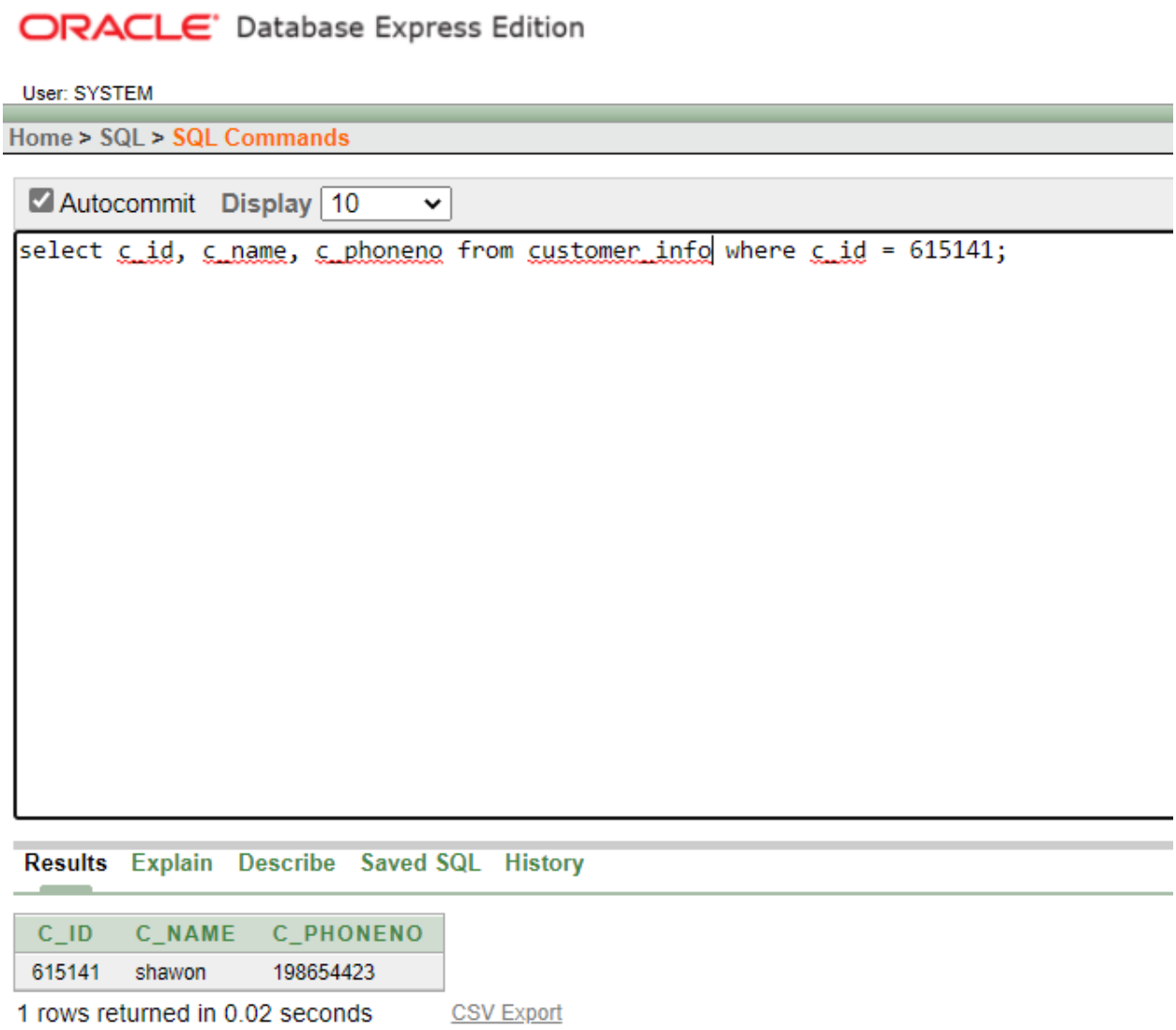
Ans: Select * from customer_info where c_id=615141;



Joining-

1.display the c_id,name,phoneno of the customer whose id is 615141

Ans: Select c_id, c_name, c_phoneno from customer_info where c_id=615141;



2.display all the infromation of item_info

Ans: Select * from item_info;

View-

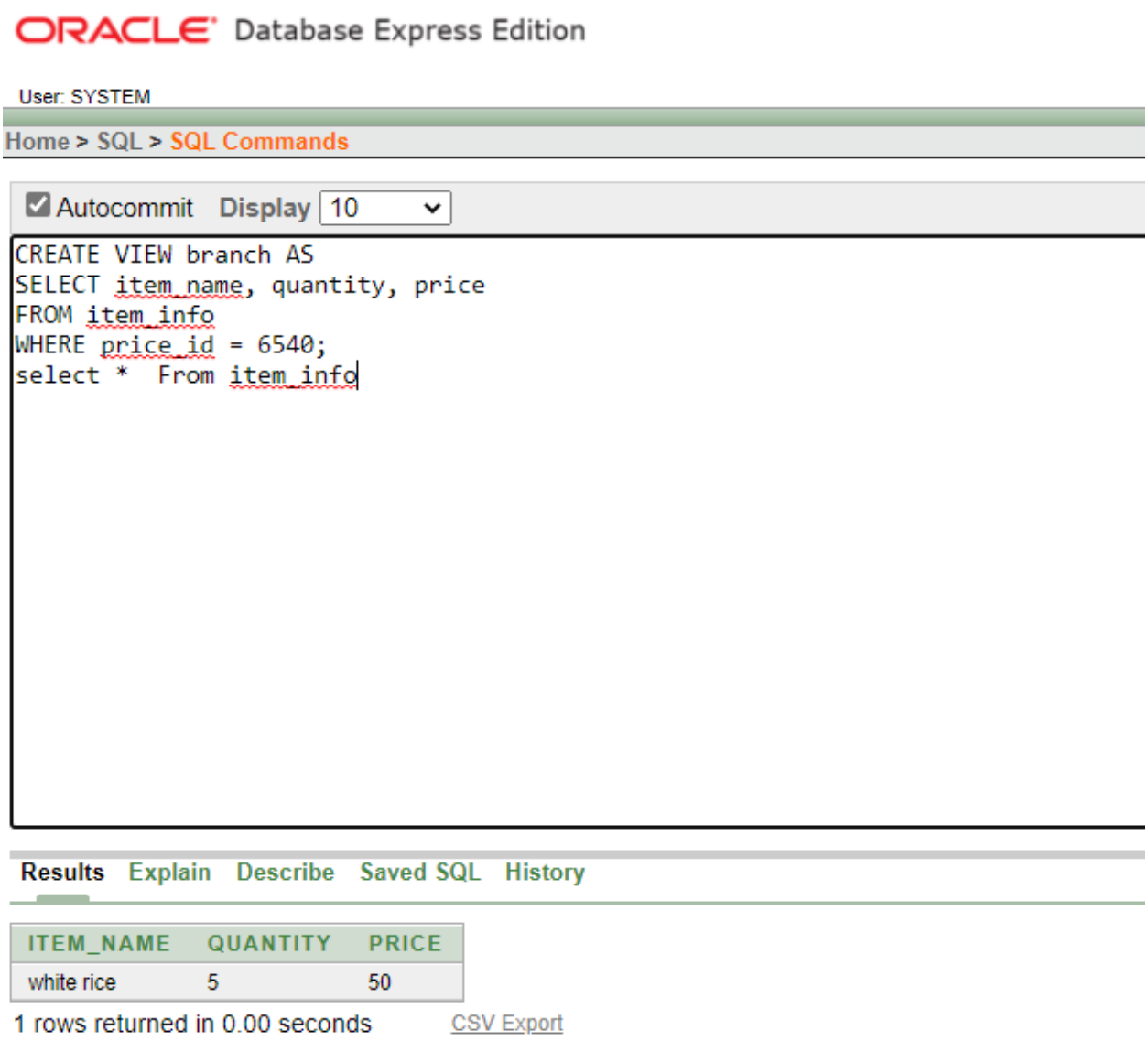
1.create a view named branch which will display name,quantity,price and price_id = 65402.

Ans: CREATE VIEW branch AS

SELECT item_name, quantity, price

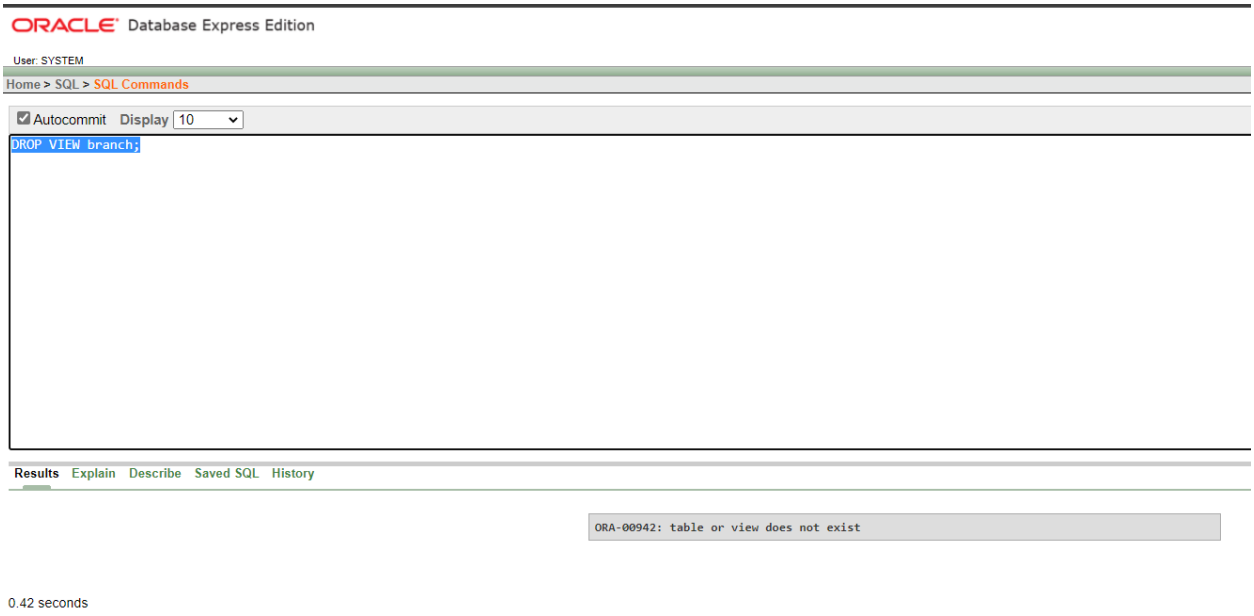
FROM item_info

WHERE price_id=65402;



2. Delete the view ‘branch’.

Ans: DROP VIEW branch



Relational Algebra

1. show item cheaper than 180 taka.

Ans: $\Pi_{TYPE}\ (\sigma_{PRICE\_ID\ <\ 180}\ (item))$

2. show item names expensive than 180 taka

Ans: $\Pi_{TYPE}\ (\sigma_{PRICE\_ID\ >\ 180}\ (item))$

3. Find the item name of the item whose QUANTITY is 10.

Ans: $\Pi_{ITEM\_NAME}(\sigma_{QUANTITY=\ \text{“10”}}\ (item\_info))$

4. show item names expensive than 110 taka

Ans: $\prod_{TYPE} (\sigma_{PRICE\_ID > 110} (item))$

5. Find the price of the item whose QUANTITY is 10.

Ans: $\prod_{Price} (\sigma_{QUANTITY= \text{“10”}} (item\_info))$

Conclusion

Our project is based on Online food delivery management system. In our project we have tried our best to create a model that will allow our system to store data and all the other information that is related to online food delivery management system. It is well tested and we assure our customer that it will be free from all sorts of anomalies. The main advantage of this system is that it greatly simplifies the ordering process for both the customer and restaurant. Besides they can get different types of specific information about the food which they want. And because of online delivery they can get their food at home. And there have also includes cash on delivery or online payment. Which is easier for any new users.