

**American International University- Bangladesh  
Faculty of CSE**

**Advance Database Management System**

Section: B (Fall 2022-2023)

Teacher: JUENA AHMED NOSHIN

Project Name:Superstore Management System

**Group Members NAME, ID and Contribution:**

|  |  |  |
| --- | --- | --- |
| Name | Id | Contribution |
| Md. Didarul Alam Alif | 20-44289-3 | ER Diagram, Normalization, Table Creation, Data Insertion, Schema Diagram ,Query Writing |
| Shaown Barua | 19-40674-1 | Introduction, Project Proposal , Scenario Description , Conclusion |
| Rakirul Haque | 19-39406-1 | ER Diagram,interfaces, Table Creation, Data Insertion |
| Shahriar, B.M. Mahir | 19-40538-1 | Cover Page, Class Diagram, Use Case Diagram, Activity Diagram |

Table of Contents

[Introduction 3](#_Toc118763912)

[Project Proposal 4](#_Toc118763913)

[Class Diagram 5](#_Toc118763914)

[Use case diagram 6](#_Toc118763915)

[Activity Diagram 7](#_Toc118763916)

[Interface 8](#_Toc118763917)

[Scenario Description 12](#_Toc118763918)

[ER Diagram 13](#_Toc118763919)

[Normalization 14](#_Toc118763920)

[Schema Diagram: 21](#_Toc118763921)

[Table Creation: 22](#_Toc118763922)

[Data Insertion: 29](#_Toc118763923)

[Query Writing: 35](#_Toc118763924)

[**Conclusion:** 41](#_Toc118763925)

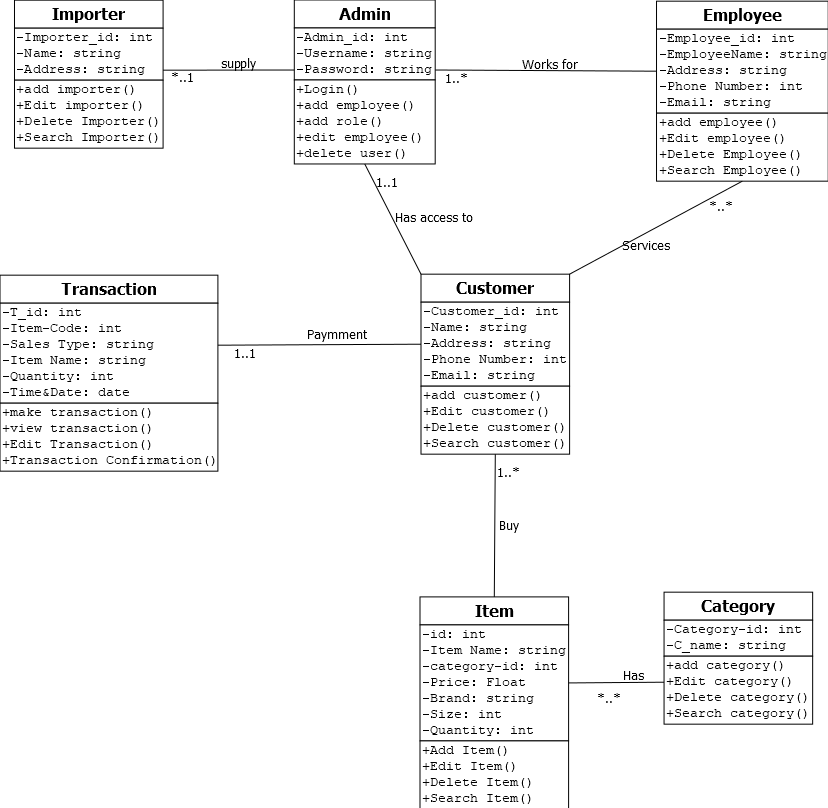
# Introduction

The purpose of database design for Super Store Management System is to assist in the buying of products, billing, and making purchase transactions quick and accurate. This management system supports in the storage and security of sales data. It allows employees and company owners to effortlessly monitor and modify business transactions. It could save time by eliminating the need to write down all of the pertinent facts and data about the business transaction and the customer. This system processes and saves all of the store's data and information. All information on customers, employees, items, sales, importers, and transactions will be stored in the system's database. This management system enables the owner to keep track of the items or sales that the consumer purchases. The user or authorized individual can only log in as a staff administrator on the Cashier system.

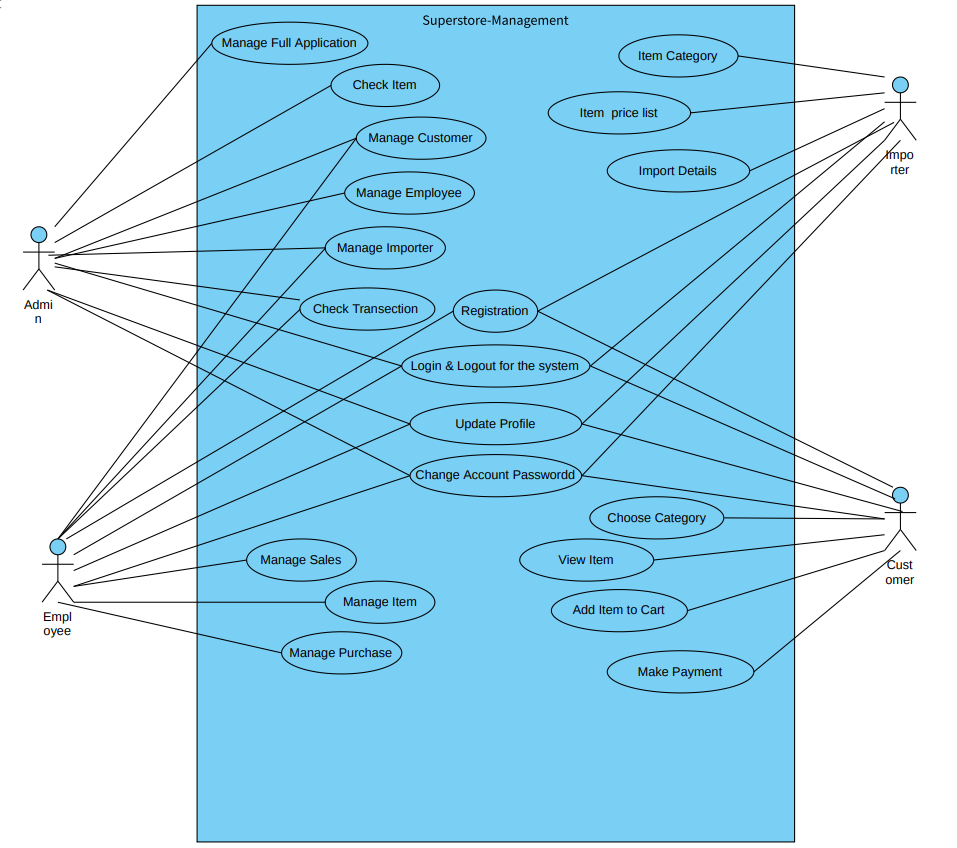
# Project Proposal

A super store is a large store that stocks many varieties of good in different departments. This is a retail offering a large number of consumer goods belonging to different product catagories. As we know a superstore deal with lots of importers, customers and stuffs. This superstore management system has realized the transmission and control of large goods, so as to facilitate the management and decision of sales, and reduce a big burden for superstore and superstore managers. It also can help to improve the work efficiency of superstore. Its requirements is to provide the basic information maintenance function of employees, customers and products so that managers can through the function to add, delete, and modify the basic information of employees and the employees can through it to add, modify and delete the basic information of customers and goods. A super store allowing the customers to buy his requirements under one roof. This project 'super store management system ' which is designed to handles all the transaction done in the shop. This system is developed to computerize the store works. When a bundle of products are purchased from the importers, all the details are noted down along with the importers name, id,phone no and email. The quantity purchased is entered in the stock. Initially the name of the product is saved in the database.as like customers are buying their requirement products and all the details are noted along with customers name,id,address, phone no, email and also noted down the payment transection .

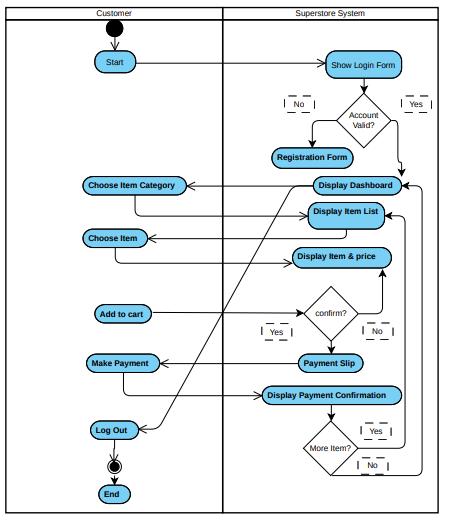
Class Diagram

****

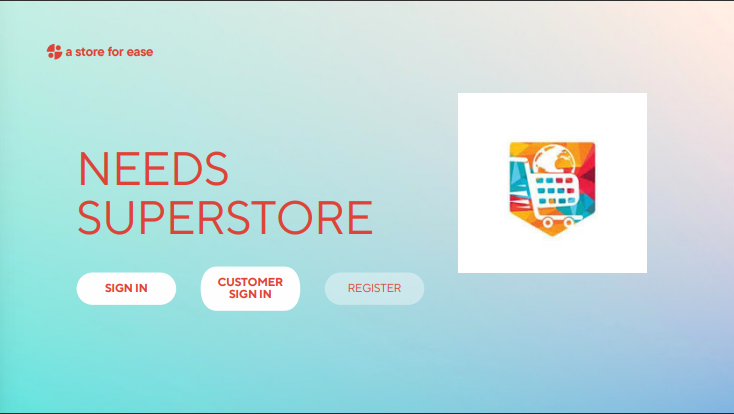
# Use case diagram

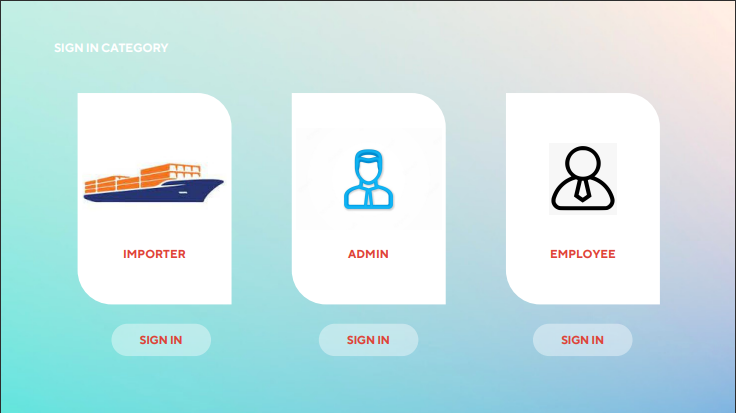


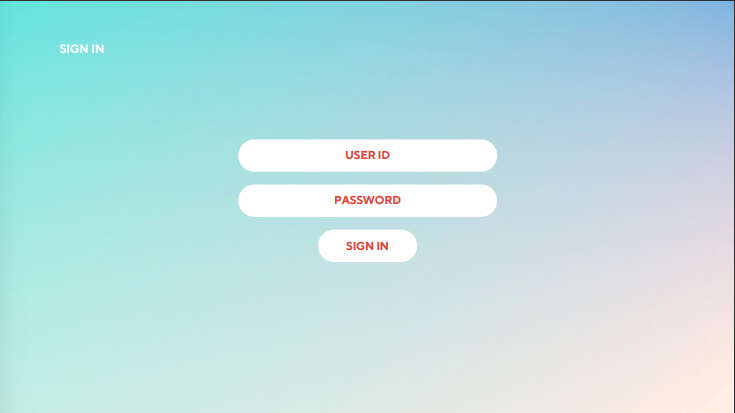
# Activity Diagram

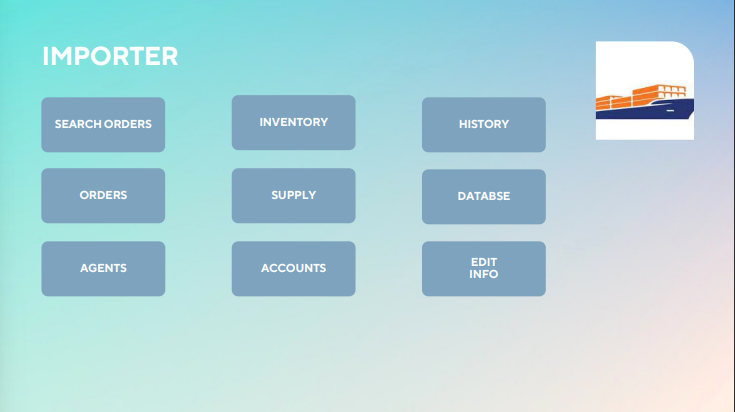


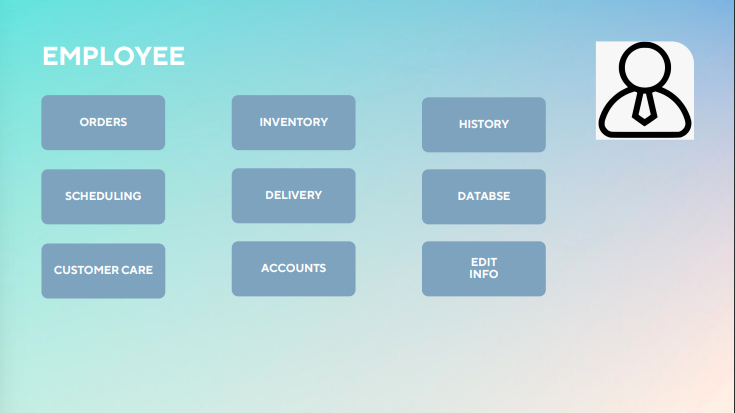
# Interface



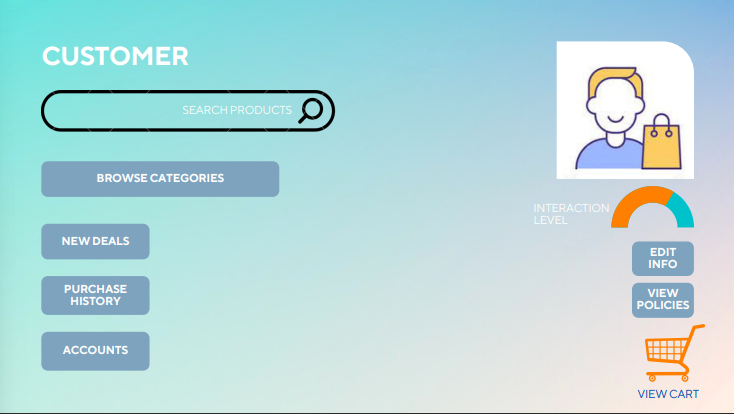


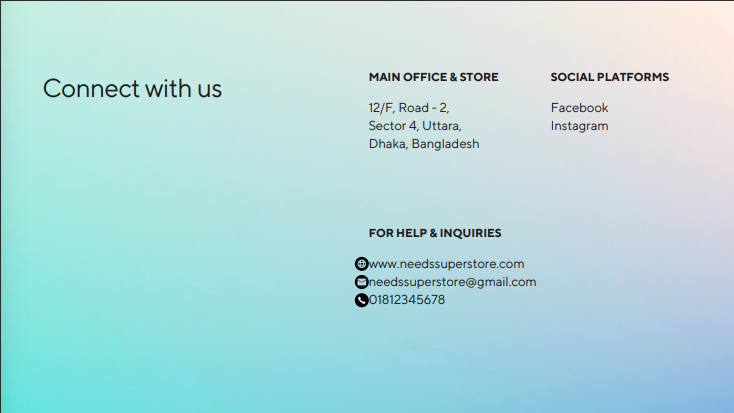












# Scenario Description

In a superstore management, a store is conducted by its admin that including(admin id, name and password). A store has many importers. They supply products for the store and they are identified by importer id, i\_name,i\_address,i\_phone number and i\_ email. A store can get products from many importers and one importer maybe supply to exactly one store. A store has multiple employees. But an employee could be work on exactly one store. An employee is identified by employee name, employee id, e\_ address, e\_ phone number and e\_email. A store has many customers. One customer can visit one store at a time. Customer is identified by c\_ name,customer\_ id,c\_ phone number, c\_email and c\_address. A customer can get service from many employees and an employee can give service to many customers.  An item is identified by  id,item name, category\_id, price, brand, size and quantity. An item has many category and a category is associated with several item .A category can be identified by category name,category-id.A customer has to pay bills for the products. The transaction is identified by t\_id, item\_code, item\_ name, quantity, sales type, time & date. A customer can make a transaction at a time and a transaction can be for one customer.

# Diagram Description automatically generatedER Diagram

Normalization**(Normalize up to 3rd Normal Form):**

**Supply**

**UNF :**

Supply ( Admin\_id , User Name, Password, Importer\_id, i\_Name, i\_Address,i\_Phone Number,i\_Email)

**1NF:**

Phone Number is a multivalued attribute

1 . Admin\_id , User Name, Password, Importer\_id, i\_Name, i\_Address,i\_Phone Number,i\_Email

**2 NF :**

1. Admin\_id , User Name, Password

2. Importer\_id, Name, Address,Phone Number,Email

**3 NF:**

There is no transitive dependency . Relation already in 3NF.

1 . Admin\_id , User Name, Password

2. Importer\_id, i\_Name, i\_Address,i\_Phone Number,i\_Email

**Table creation :**

1 . Admin\_id , User Name, Password

2. Importer\_id, i\_Name, i\_Address,i\_Phone Number,i\_Email, **Admin\_id**

**Works For**

**UNF :**

Works For ( Admin\_id , User Name, Password,Employee\_id,Employee Name, e\_Address,e\_Phone Number,e\_Email)

**1NF:**

Phone Number is a multivalued attribute

1 . Admin\_id , User Name, Password,Employee\_id, Employee Name, e\_Address,e\_Phone Number,e\_Email

**2 NF :**

1. Admin\_id , User Name, Password

2. Employee\_id, Employee Name, e\_Address,e\_Phone Number,e\_Email

**3 NF:**

There is no transitive dependency . Relation already in 3NF.

1 . Admin\_id , User Name, Password

2. Employee\_id, Employee Name, e\_Address,e\_Phone Number,e\_Email

**Table creation :**

1 . Admin\_id , User Name, Password

2. Employee\_id, Employee Name, e\_Address,e\_Phone Number,e\_Email, **Admin\_id**

**Services**

**UNF :**

Services ( Employee\_id,Employee Name, e\_Address,e\_Phone Number,e\_Email,Customer-id , c\_Name, c\_Address, c\_Phone Number,c\_Email)

**1NF:**

Phone Number is a multivalued attribute

1 . Employee\_id,Employee Name, e\_Address,e\_Phone Number,e\_Email,Customer-id , c\_Name, c\_Address, c\_Phone Number,c\_Email

**2 NF :**

1. Employee\_id, Employee Name, e\_Address,e\_Phone Number,e\_Email

2. Customer-id , c\_Name, c\_Address, c\_Phone Number,c\_Email

**3 NF:**

There is no transitive dependency . Relation already in 3NF.

1. Employee\_id, Employee Name, e\_Address,e\_Phone Number,e\_Email

2. Customer-id , c\_Name, c\_Address, c\_Phone Number,c\_Email

**Table creation :**

1. Employee\_id, Employee Name, e\_Address,e\_Phone Number,e\_Email

2. Customer-id , c\_Name, c\_Address, c\_Phone Number,c\_Email

3. **Employee\_id, Customer-id**

**Has access to**

**UNF :**

Holds ( Admin\_id , User Name, Password, Customer-id , c\_Name, c\_Address, c\_Phone Number,c\_Email

)

**1NF:**

Phone Number is a multivalued attribute

1 . Admin\_id , User Name, Password, Customer-id , c\_Name, c\_Address, c\_Phone Number,c\_Email

**2 NF :**

1. Admin\_id , User Name, Password

2. Customer-id , c\_Name, c\_Address, c\_Phone Number,c\_Email

**3 NF:**

There is no transitive dependency . Relation already in 3NF.

1 . Admin\_id , User Name, Password

2. Customer-id , c\_Name, c\_Address, c\_Phone Number,c\_Email

**Table creation :**

1 . Admin\_id , User Name, Password

2. Customer-id , c\_Name, c\_Address, c\_Phone Number,c\_Email, **Admin\_id**

**Payment**

**UNF :**

Payment (Customer-id , c\_Name, c\_Address, c\_Phone Number,c\_Email, T-id,Item-Code,Sales Type,Item Name,Quantity,Time & Date)

**1NF:**

Phone Number is a multivalued attribute

1 , Customer-id , c\_Name, c\_Address, c\_Phone Number,c\_Email,T-id,Item-Code,Sales Type,Item Name,Quantity,Time & Date)

**2 NF :**

1. Customer-id , c\_Name, c\_Address, c\_Phone Number,c\_Email

2. T-id,Item-Code,Sales Type,Item Name,Quantity,Time & Date

**3 NF:**

There is no transitive dependency . Relation already in 3NF.

1. Customer-id , c\_Name, c\_Address, c\_Phone Number,c\_Email

2. T-id,Item-Code,Sales Type,Item Name,Quantity,Time & Date

**Table creation :**

1. Customer-id , c\_Name, c\_Address, c\_Phone Number,c\_Email

2. T-id,Item-Code,Sales Type,Item Name,Quantity,Time & Date, **Customer-id**

**Buys**

**UNF :**

Buys (Customer-id , c\_Name, c\_Address, c\_Phone Number,c\_Email, Id, Item Name,Category-Id,Price,Brand,Size,Quantity)

**1NF:**

Phone Number is a multivalued attribute

1 , Customer-id , c\_Name, c\_Address, c\_Phone Number,c\_Email, Id, Item Name,Category-Id,Price,Brand,Size,Quantity

**2 NF :**

1. Customer-id , c\_Name, c\_Address, c\_Phone Number,c\_Email

2. Id, Item Name,Category-Id,Price,Brand,Size,Quantity

**3 NF:**

There is no transitive dependency . Relation already in 3NF.

1. Customer-id , Name, Address, Phone Number,Email

2. Id, Name,Category-Id,Price,Brand,Size,Quantity

**Table creation :**

1. Customer-id , c\_Name, c\_Address, c\_Phone Number,c\_Email

2. Id, Item Name,Category-Id,Price,Brand,Size,Quantity,**Customer-id**

**Has**

**UNF :**

Has (Id, Item Name,Category-Id,Price,Brand,Size,Quantity,Category\_id,Category Name)

**1NF:**

Phone Number is a multivalued attribute

1 . Id, Item Name,Category-Id,Price,Brand,Size,Quantity,Category\_id,Category Name

**2 NF :**

1. Id, Item Name,Category-Id,Price,Brand,Size,Quantity

2. Category\_id,Category Name

**3 NF:**

There is no transitive dependency . Relation already in 3NF.

1. Id, Item Name,Category-Id,Price,Brand,Size,Quantity

2. Category\_id,Category Name

**Table creation :**

1. Id, Item Name,Category-Id,Price,Brand,Size,Quantity

2. Category\_id,Category Name

3. **Id ,Category\_id**

# Schema Diagram:

# Table Creation:

**-After Normalization create tables**

**Temporary tables**

1. ~~Admin\_id , User Name, Password~~
2. Importer\_id, i\_Name, i\_Address,i\_Phone Number,i\_Email, **Admin\_id**
3. ~~Admin\_id , User Name, Password~~
4. Employee\_id,Employee Name, e\_Address,e\_Phone Number,e\_Email, **Admin\_id**
5. ~~Employee\_id,Employee Name,e\_ Address,e\_Phone Number,e\_Email,~~
6. ~~Customer-id ,c\_ Name, c\_Address, c\_Phone Number,c\_Email~~
7. **Employee\_id, Customer-id**
8. Admin\_id , User Name, Password
9. Customer-id , c\_Name, c\_Address, c\_Phone Number,c\_Email, **Admin\_id**
10. ~~Customer-id , c\_Name,c\_ Address, c\_Phone Number,c\_Email~~
11. T-id,Item-Code,Sales Type,Item Name,Quantity,Time & Date, **Customer-id**
12. ~~Customer-id , c\_Name,c\_ Address, c\_Phone Number,c\_Email~~
13. Id, Item Name,Category-Id,Price,Brand,Size,Quantity, **Customer-id**
14. ~~Id,Item Name,Category-Id,Price,Brand,Size,Quantity~~
15. Category\_id, Category \_Name
16. **Id ,Category\_id**

**Final tables**

1. Importer\_id, i\_Name, i\_Address,i\_Phone Number,i\_Email, **Admin\_id**
2. Employee\_id,Employee Name, e\_Address,e\_Phone Number,e\_Email, **Admin\_id**
3. **Employee\_id, Customer-id**
4. Admin\_id , User Name, Password
5. Customer-id , c\_Name, c\_Address, c\_Phone Number,c\_Email, **Admin\_id**
6. T-id,Item-Code,Sales Type,Item Name,Quantity,Time & Date, **Customer-id**
7. Id, Item Name,Category-Id,Price,Brand,Size,Quantity, **Customer-id**
8. Category\_id, Category\_Name
9. **Id ,Category\_id**

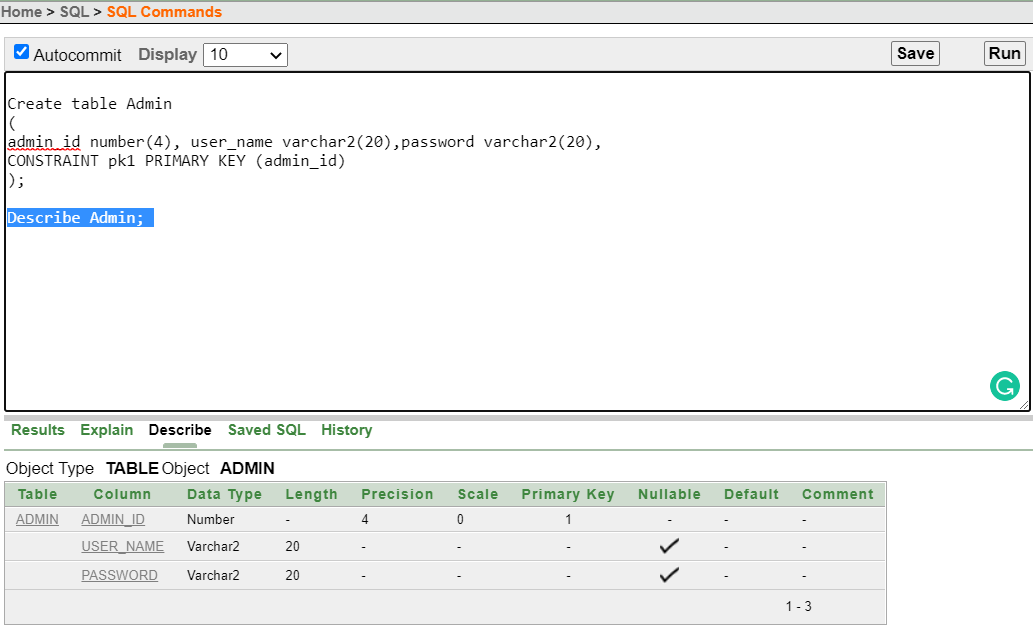
**-Include constraints when creating Tables(Table’s name in Dark red color)**

**1. Admin**

Create table Admin(

admin\_id number(4), user\_name varchar2(20),password varchar2(20),CONSTRAINT pk1 PRIMARY KEY (admin\_id));

Describe Admin;



**2. Importer**

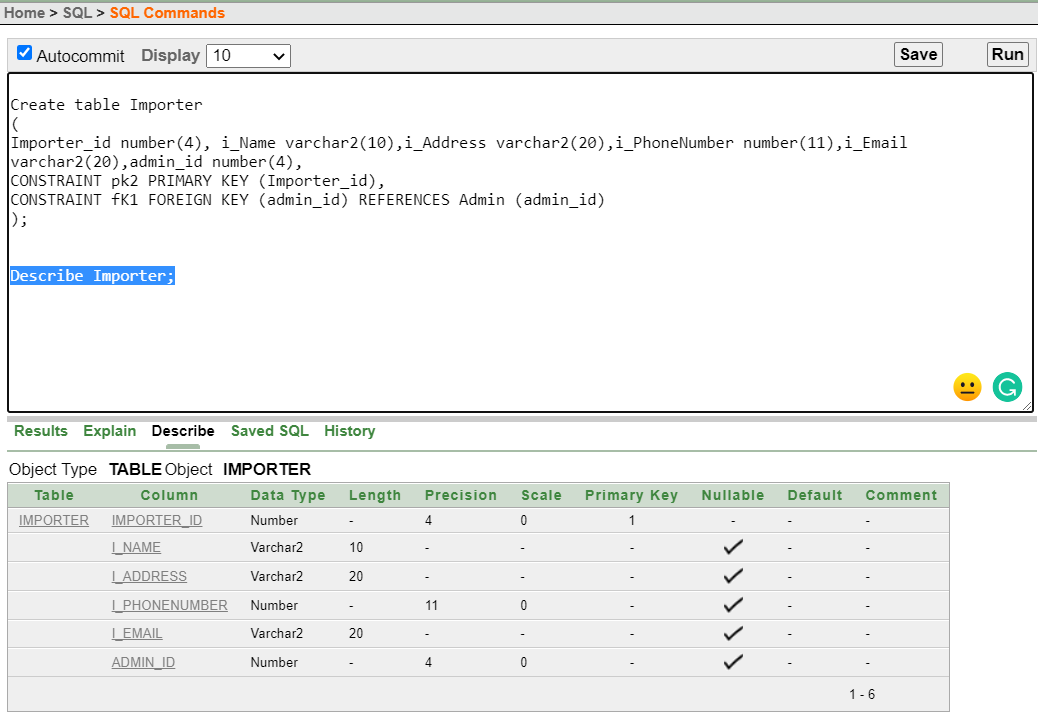
Create table Importer(

Importer\_id number(4), i\_Name varchar2(10),i\_Address varchar2(20),i\_PhoneNumber number(11),i\_Email varchar2(20),admin\_id number(4),

CONSTRAINT pk2 PRIMARY KEY (Importer\_id),

CONSTRAINT fK1 FOREIGN KEY (admin\_id) REFERENCES Admin (admin\_id));

Describe Importer;



**3. Employee**

Create table Employee(

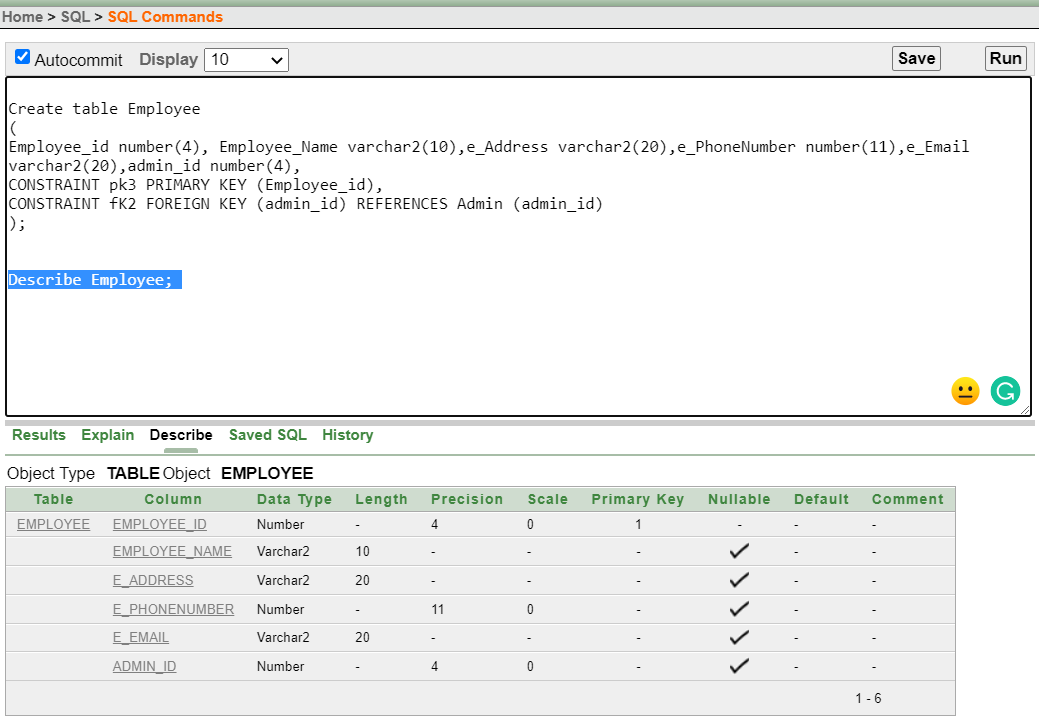
Employee\_id number(4), Employee\_Name varchar2(10),e\_Address varchar2(20),

e\_PhoneNumber number(11),e\_Email varchar2(20),admin\_id number(4),

CONSTRAINT pk3 PRIMARY KEY (Employee\_id),

CONSTRAINT fK2 FOREIGN KEY (admin\_id) REFERENCES Admin (admin\_id));

Describe Employee;



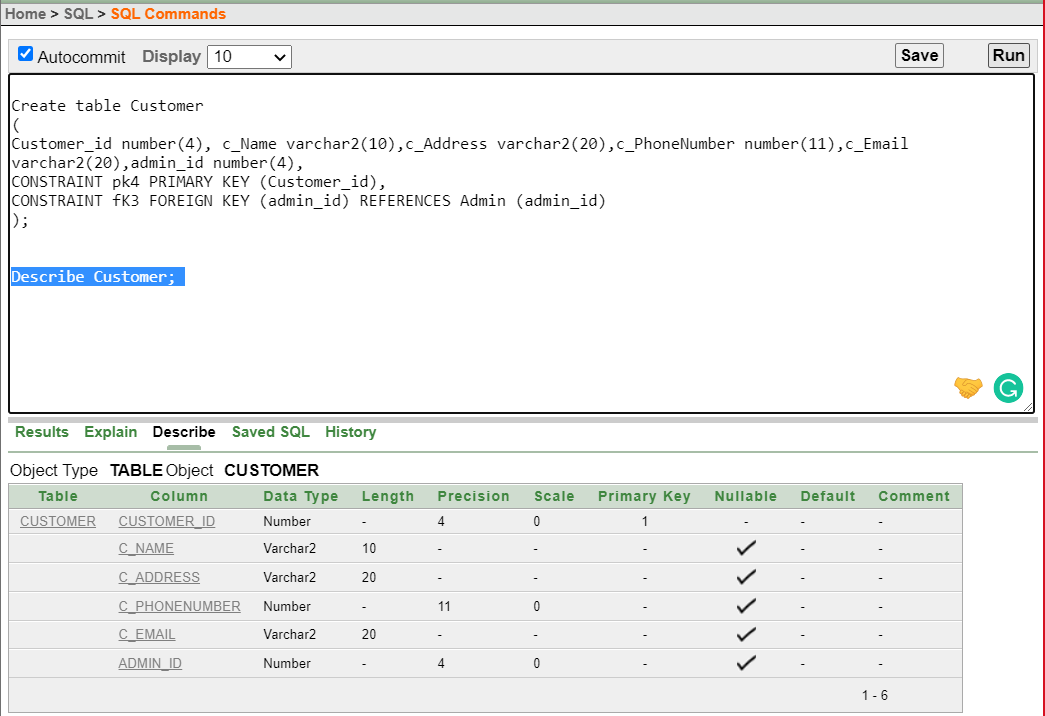
**4. Customer**

Create table Customer(

Customer\_id number (4), c\_Name varchar2 (10), c\_Address varchar2 (20), c\_PhoneNumber number (11),c\_Email varchar2(20), admin\_id number (4), CONSTRAINT pk4 PRIMARY KEY (Customer\_id),

CONSTRAINT FK3 FOREIGN KEY (admin\_id) REFERENCES Admin (admin\_id));

Describe Customer;



**5. Transaction**

Create table Transaction(

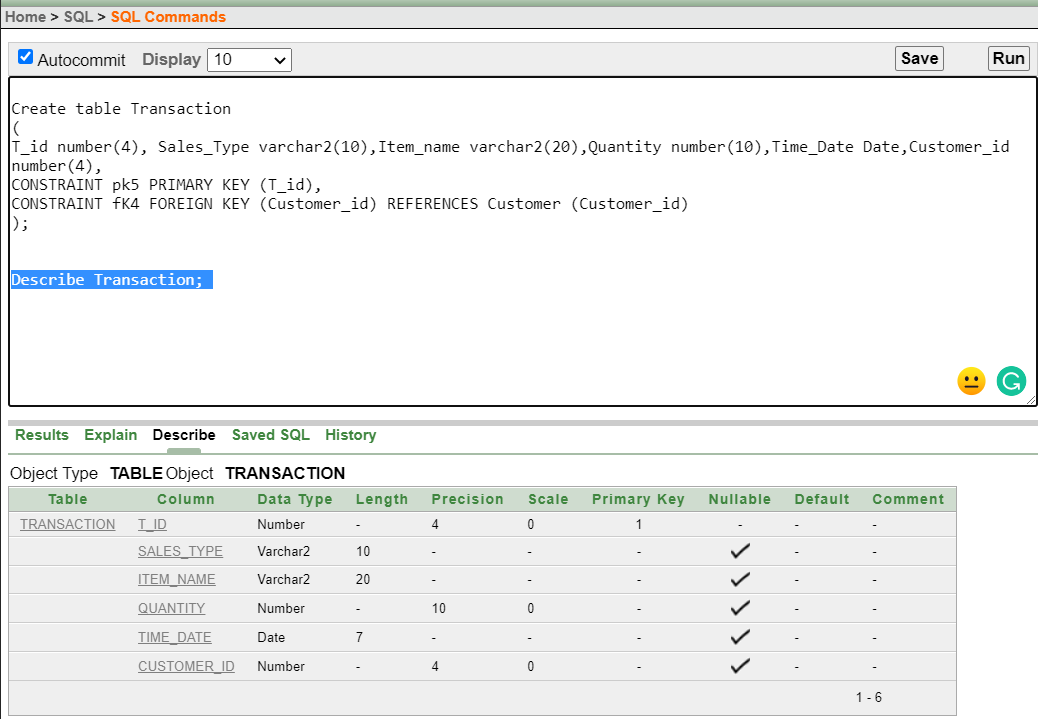
T\_id number(4), Sales\_Type varchar2(10),Item\_name varchar2(20),Quantity number(10),Time\_Date Date,Customer\_id number(4),

CONSTRAINT pk5 PRIMARY KEY (T\_id),

CONSTRAINT fK4 FOREIGN KEY (Customer\_id) REFERENCES Customer(Customer\_id)

);

Describe Transaction;



**6. Item**

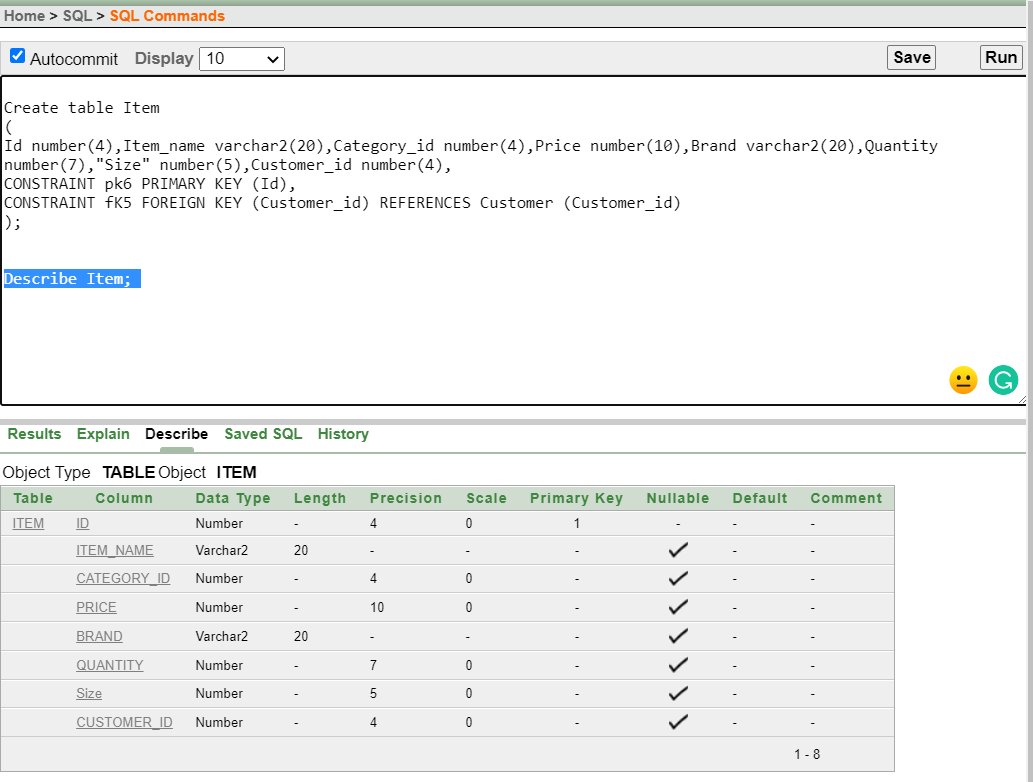
Create table Item(

Id number(4),Item\_name varchar2(20),Category\_id number(4),Price number(10),Brand varchar2(20),Quantity number(7),"Size" number(5),Customer\_id number(4),

CONSTRAINT pk6 PRIMARY KEY (Id),

CONSTRAINT fK5 FOREIGN KEY (Customer\_id) REFERENCES Customer (Customer\_id));

Describe Item;

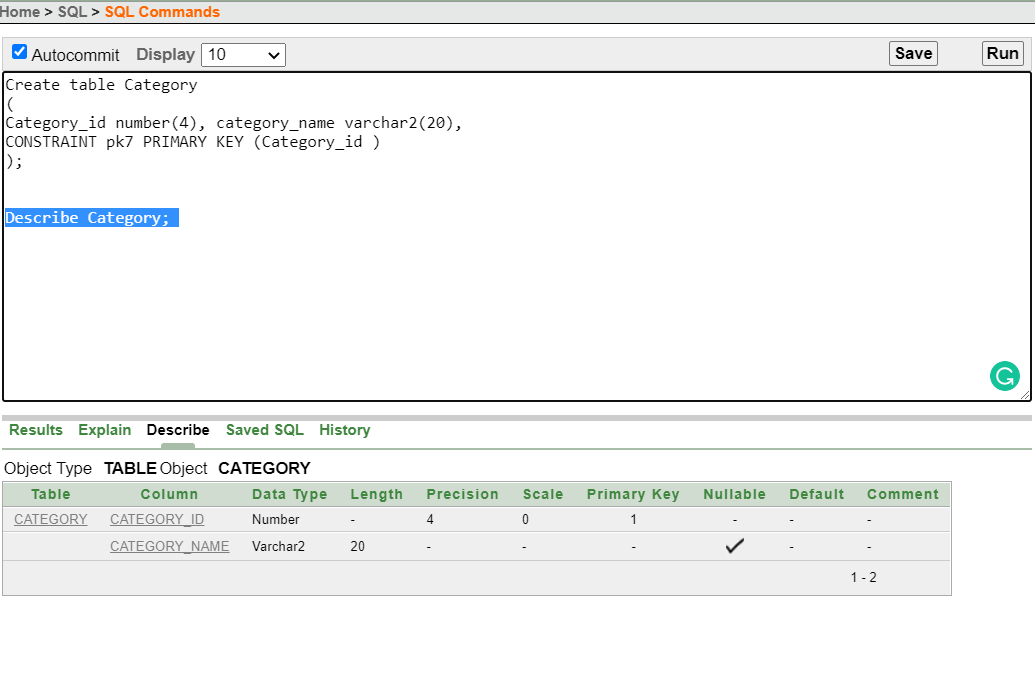


**7. Category**

Create table Category

(Category\_id number (4), category\_name varchar2(20), CONSTRAINT pk7 PRIMARY KEY (Category\_id) );

Describe Category;



**8. Item\_Cate**

Create table Item\_Cate

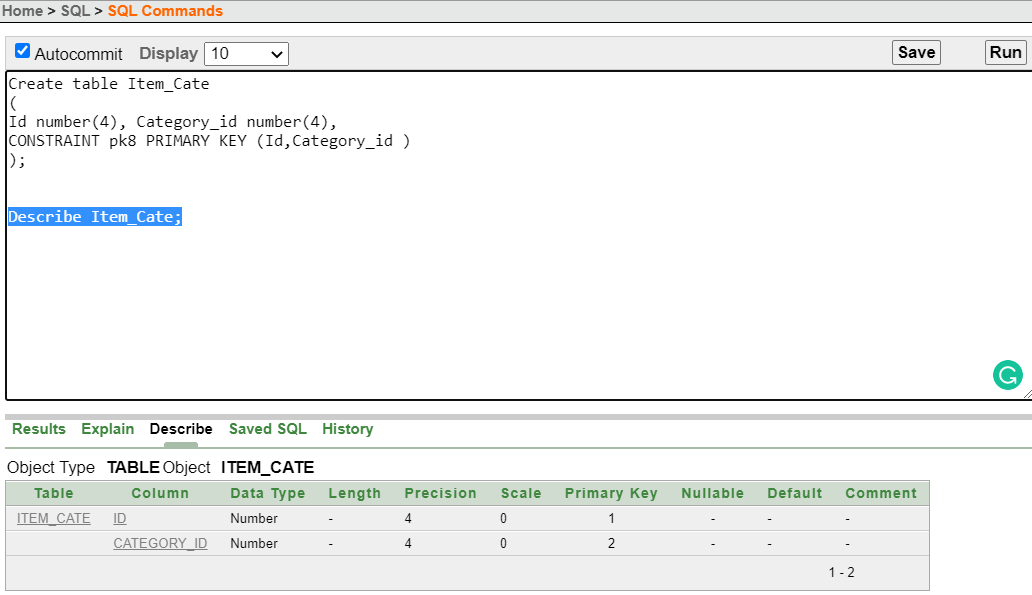
(

Id number(4), Category\_id number(4),

CONSTRAINT pk8 PRIMARY KEY (Id,Category\_id )

);

Describe Item\_Cate;



**9. emp\_cust**

Create table emp\_cust

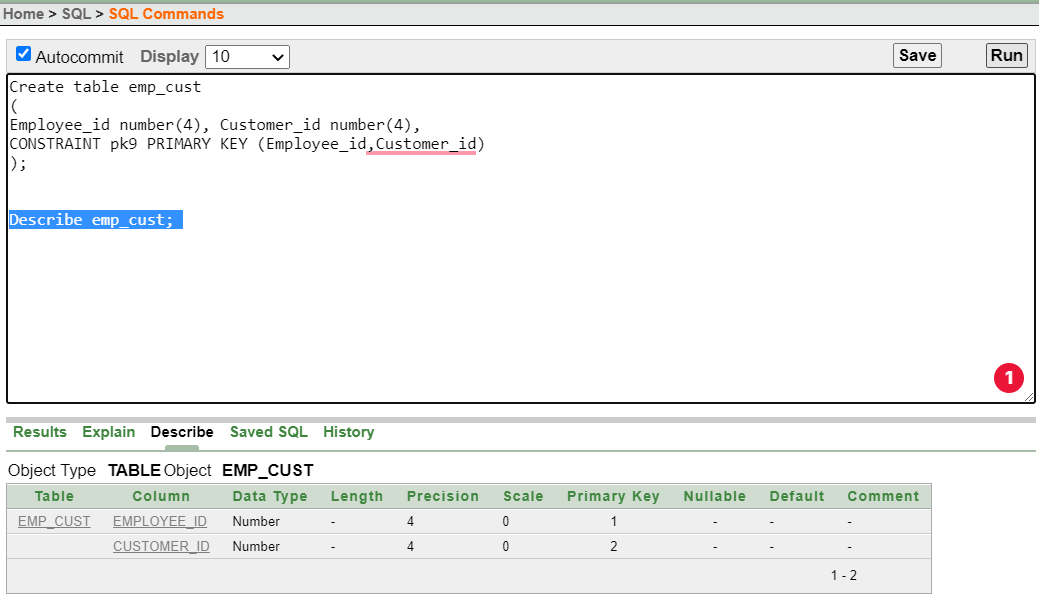
(

Employee\_id number(4), Customer\_id number(4),

CONSTRAINT pk9 PRIMARY KEY (Employee\_id,Customer\_id)

);

Describe emp\_cust;



**-Include the queries required to create sequence**

1.CREATE SEQUENCE Admin\_adminid

INCREMENT BY 1

START WITH 111;

2.CREATE SEQUENCE Importer\_Importerid

INCREMENT BY 1

START WITH 222;

3.CREATE SEQUENCE Employee\_empid

INCREMENT BY 1

START WITH 2234;

4.CREATE SEQUENCE Customer\_Customerid

INCREMENT BY 1

START WITH 333;

5.CREATE SEQUENCE Transaction\_Tid

INCREMENT BY 1

START WITH 444;

6.CREATE SEQUENCE Item\_Id

INCREMENT BY 1

START WITH 6666;

7.CREATE SEQUENCE Category\_Categoryid

INCREMENT BY 1

START WITH 777;

8. CREATE SEQUENCE Item\_CateId

INCREMENT BY 1

START WITH 6686;

9.CREATE SEQUENCE emp\_custid

INCREMENT BY 1

START WITH 2234;

# Data Insertion:

**1.Admin:**

INSERT INTO Admin (admin\_id, user\_name, password) VALUES (Admin\_adminid.NEXTVAL, 'a', 'abcd1234');

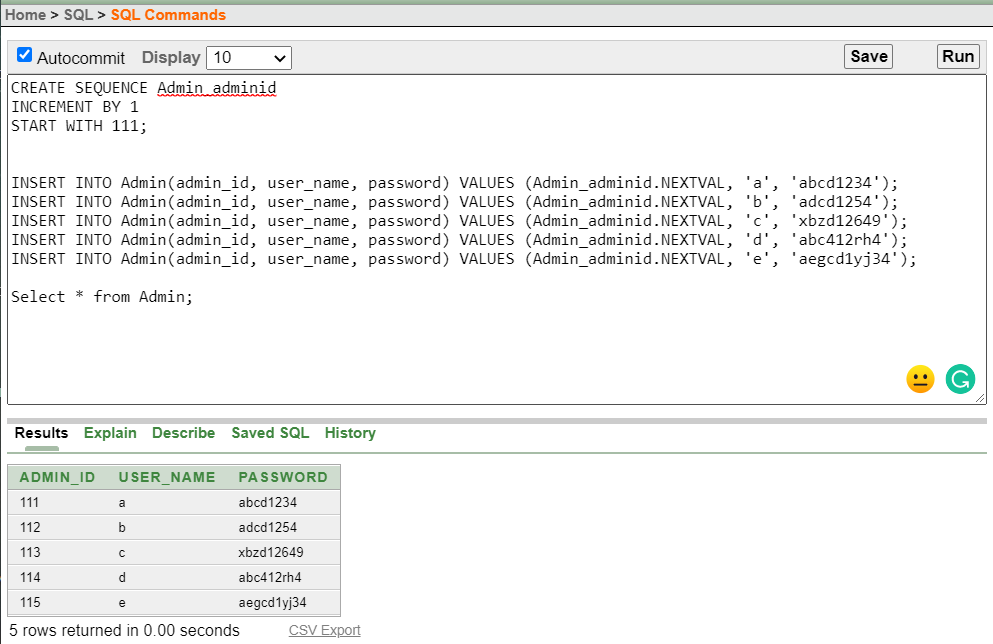
INSERT INTO Admin (admin\_id, user\_name, password) VALUES (Admin\_adminid.NEXTVAL, 'b','adcd1254');

INSERT INTO Admin (admin\_id, user\_name, password) VALUES (Admin\_adminid.NEXTVAL, 'c', 'xbzd12649');

INSERT INTO Admin (admin\_id, user\_name, password) VALUES (Admin\_adminid.NEXTVAL, 'd', 'abc412rh4');

INSERT INTO Admin (admin\_id, user\_name, password) VALUES (Admin\_adminid.NEXTVAL, 'e', 'aegcdlyj34');

Select \* from Admin;



**2.Importer:**

INSERT INTO Importer(Importer\_id, i\_name, i\_address,i\_PhoneNumber,i\_Email,admin\_id) VALUES (Importer\_Importerid.NEXTVAL, 'x', 'Beijing-China','+8616521689','x1@gmail.com','111');

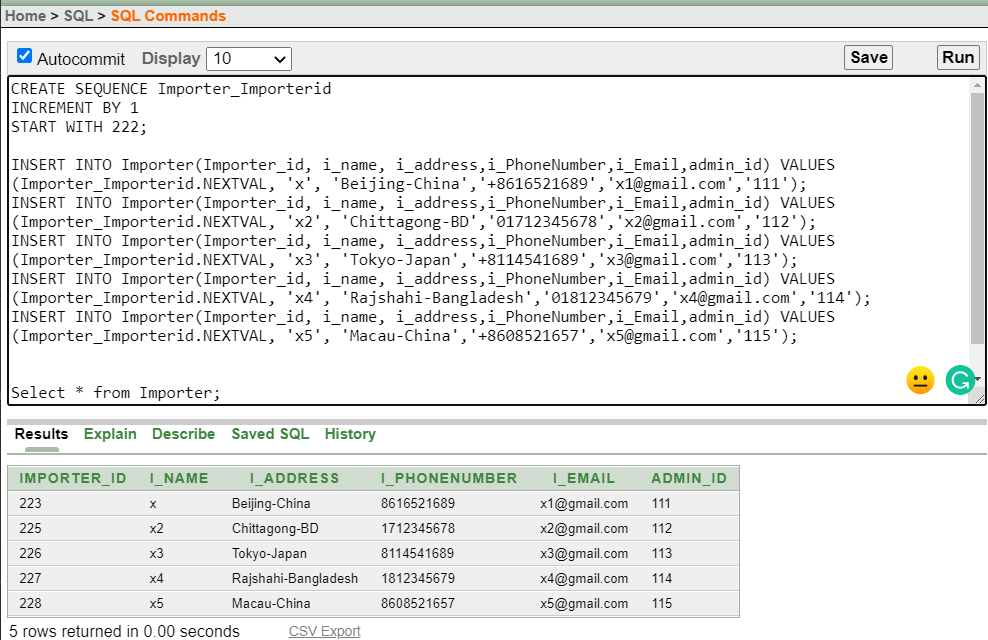
INSERT INTO Importer(Importer\_id, i\_name, i\_address,i\_PhoneNumber,i\_Email,admin\_id) VALUES (Importer\_Importerid.NEXTVAL, 'x2', 'Chittagong-BD','01712345678','x2@gmail.com','112');

INSERT INTO Importer(Importer\_id, i\_name, i\_address,i\_PhoneNumber,i\_Email,admin\_id) VALUES (Importer\_Importerid.NEXTVAL, 'x3', 'Tokyo-Japan','+8114541689','x3@gmail.com','113');

INSERT INTO Importer(Importer\_id, i\_name, i\_address,i\_PhoneNumber,i\_Email,admin\_id) VALUES (Importer\_Importerid.NEXTVAL, 'x4', 'Rajshahi-Bangladesh','01812345679','x4@gmail.com','114');

INSERT INTO Importer(Importer\_id, i\_name, i\_address,i\_PhoneNumber,i\_Email,admin\_id) VALUES (Importer\_Importerid.NEXTVAL, 'x5', 'Macau-China','+8608521657','x5@gmail.com','115');

Select \* from Importer;



**3.Employee:**

INSERT INTO

Employee(Employee\_id,Employee\_Name,e\_Address,e\_PhoneNumber,e\_Email,admin\_id) VALUES

(Employee\_empid.NEXTVAL, 'd1' ,'Dhaka-BD','01988256435','d1@gmail.com','111')

INSERT INTO Employee(Employee\_id,Employee\_Name,e\_Address,e\_PhoneNumber,e\_Email,admin\_id) VALUES

(Employee\_empid.NEXTVAL, 'd2','Kushtia-BD','01988253576','d2@gmail.com','112')

INSERT INTO Employee(Employee\_id,Employee\_Name,e\_Address,e\_PhoneNumber,e\_Email,admin\_id) VALUES

(Employee\_empid.NEXTVAL, 'd3' ,'Pabna-BD','01988256777','d3@gmail.com','113')

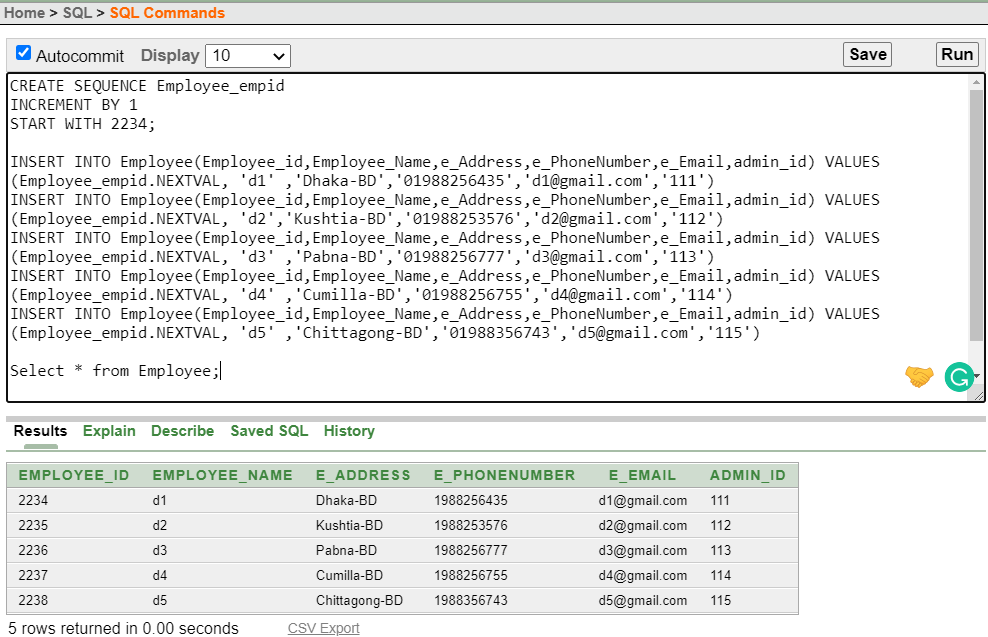
INSERT INTO Employee(Employee\_id,Employee\_Name,e\_Address,e\_PhoneNumber,e\_Email,admin\_id) VALUES

(Employee\_empid.NEXTVAL, 'd4' ,'Cumilla-BD','01988256755','d4@gmail.com','114')

INSERT INTO Employee(Employee\_id,Employee\_Name,e\_Address,e\_PhoneNumber,e\_Email,admin\_id) VALUES

(Employee\_empid.NEXTVAL, 'd5' ,'Chittagong-BD','01988356743','d5@gmail.com','115')

Select \* from Employee;



**4.Customer:**

INSERT INTO Customer(Customer\_id, c\_name, c\_address,c\_PhoneNumber,c\_Email,admin\_id) VALUES (Customer\_Customerid.NEXTVAL, 'c1', 'Dhaka-Bd','01758792437','c1@gmail.com','111');

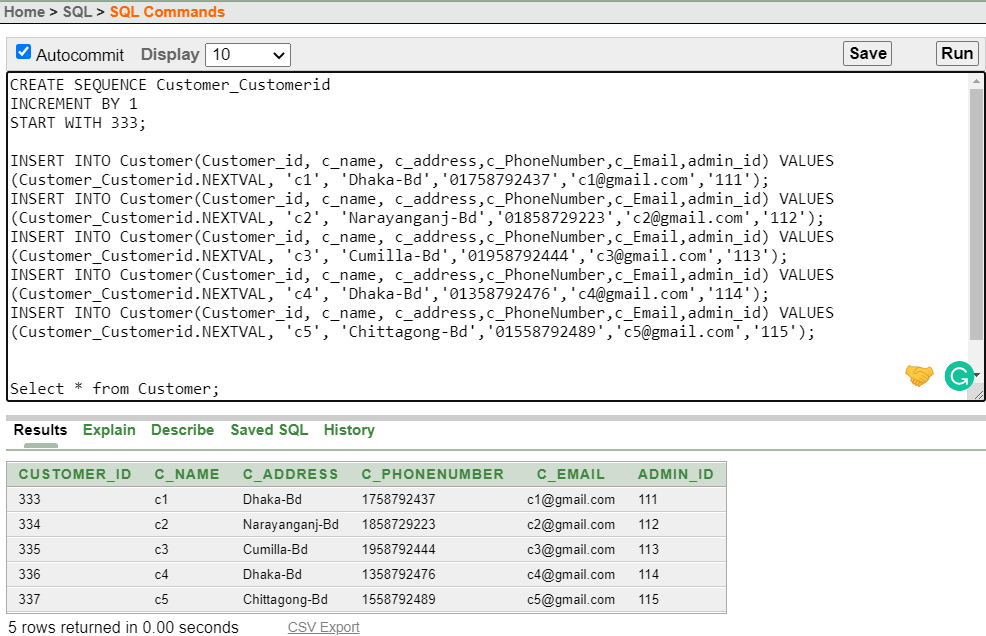
INSERT INTO Customer(Customer\_id, c\_name, c\_address,c\_PhoneNumber,c\_Email,admin\_id) VALUES (Customer\_Customerid.NEXTVAL, 'c2', 'Narayanganj-Bd','01858729223','c2@gmail.com','112');

INSERT INTO Customer(Customer\_id, c\_name, c\_address,c\_PhoneNumber,c\_Email,admin\_id) VALUES (Customer\_Customerid.NEXTVAL, 'c3', 'Cumilla-Bd','01958792444','c3@gmail.com','113');

INSERT INTO Customer(Customer\_id, c\_name, c\_address,c\_PhoneNumber,c\_Email,admin\_id) VALUES (Customer\_Customerid.NEXTVAL, 'c4', 'Dhaka-Bd','01358792476','c4@gmail.com','114');

INSERT INTO Customer(Customer\_id, c\_name, c\_address,c\_PhoneNumber,c\_Email,admin\_id) VALUES (Customer\_Customerid.NEXTVAL, 'c5', 'Chittagong-Bd','01558792489','c5@gmail.com','115');

Select \* from Customer;



**5.Transaction:**

INSERT INTO Transaction(T\_id, Sales\_Type, Item\_name,Quantity,Time\_Date,Customer\_id) VALUES (Transaction\_Tid.NEXTVAL, 'Online', 'T\_Shirt','3','19-oct-2022','333');

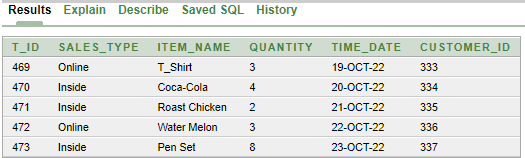
INSERT INTO Transaction(T\_id, Sales\_Type, Item\_name,Quantity,Time\_Date,Customer\_id) VALUES (Transaction\_Tid.NEXTVAL, 'Inside ', 'Coca-Cola', '4','20-oct-2022', '334') ;

INSERT INTO Transaction(T\_id, Sales\_Type, Item\_name,Quantity,Time\_Date,Customer\_id) VALUES (Transaction\_Tid.NEXTVAL, 'Inside ', 'Roast Chicken','2','21-oct-2022','335');

INSERT INTO Transaction(T\_id, Sales\_Type, Item\_name,Quantity,Time\_Date,Customer\_id) VALUES (Transaction\_Tid.NEXTVAL, 'Online', 'Water Melon','3','22-oct-2022','336');

INSERT INTO Transaction(T\_id, Sales\_Type, Item\_name,Quantity,Time\_Date,Customer\_id) VALUES (Transaction\_Tid.NEXTVAL, 'Inside ', 'Pen Set','8', '23-oct-2022','337');

Select \* from Transaction;



**6.Item:**

INSERT INTO Item(Id ,Item\_name,Category\_id,Price,Brand,Quantity,"Size",Customer\_id ) VALUES(Item\_Id.NEXTVAL, 'Pepsi' ,'777','20','Pepsico','1','1','333');

INSERT INTO Item(Id ,Item\_name,Category\_id,Price,Brand,Quantity,"Size",Customer\_id ) VALUES(Item\_Id.NEXTVAL,'Apple','778','195','None','1','1','334');

INSERT INTO Item(Id ,Item\_name,Category\_id,Price,Brand,Quantity,"Size",Customer\_id ) VALUES(Item\_Id.NEXTVAL, 'Egg' ,'779','9','Kazi','1','1','335');

INSERT INTO Item(Id ,Item\_name,Category\_id,Price,Brand,Quantity,"Size",Customer\_id ) VALUES(Item\_Id.NEXTVAL, 'Rice' ,'780','75','Pran','1','1','336');

INSERT INTO Item(Id ,Item\_name,Category\_id,Price,Brand,Quantity,"Size",Customer\_id ) VALUES(Item\_Id.NEXTVAL, 'Orange' ,'781','238','NONE','1','1','337');

Select \* from Item;



**7.Category:**

INSERT INTO Category(Category\_id, Category\_name) VALUES (Category\_Categoryid.NEXTVAL, 'cate1');

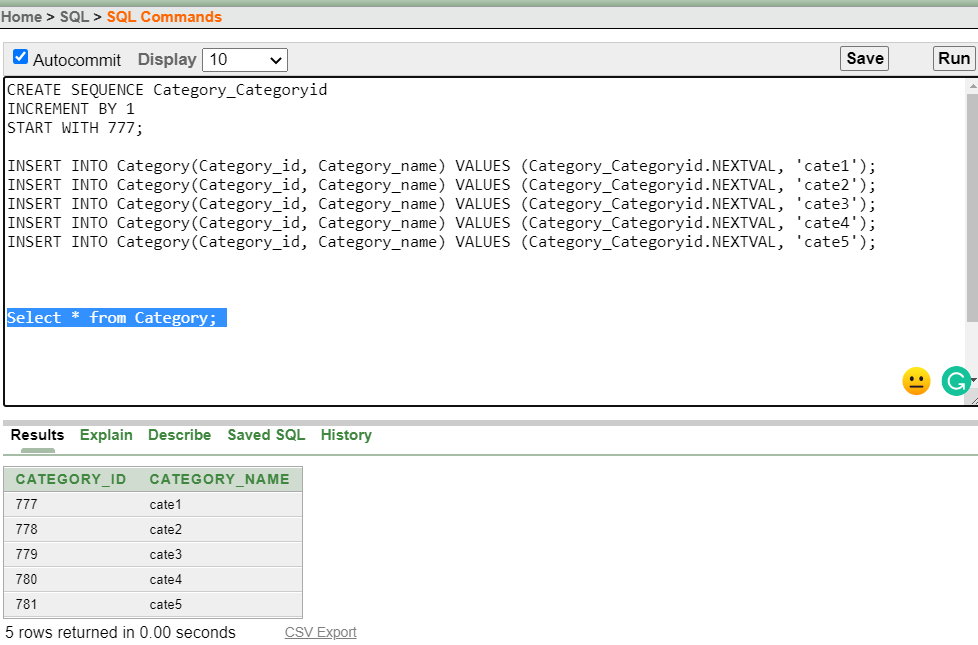
INSERT INTO Category(Category\_id, Category\_name) VALUES (Category\_Categoryid.NEXTVAL, 'cate2');

INSERT INTO Category(Category\_id, Category\_name) VALUES (Category\_Categoryid.NEXTVAL, 'cate3');

INSERT INTO Category(Category\_id, Category\_name) VALUES (Category\_Categoryid.NEXTVAL, 'cate4');

INSERT INTO Category(Category\_id, Category\_name) VALUES (Category\_Categoryid.NEXTVAL, 'cate5');

Select \* from Category;



**8.Item\_Cate:**

INSERT INTO Item\_Cate(Id,Category\_id) VALUES(Item\_CateId.NEXTVAL,'777');

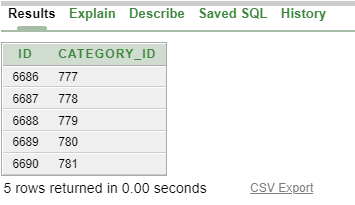
INSERT INTO Item\_Cate(Id,Category\_id) VALUES(Item\_CateId.NEXTVAL,'778');

INSERT INTO Item\_Cate(Id,Category\_id) VALUES(Item\_CateId.NEXTVAL,'779');

INSERT INTO Item\_Cate(Id,Category\_id) VALUES(Item\_CateId.NEXTVAL,'780');

INSERT INTO Item\_Cate(Id,Category\_id) VALUES(Item\_CateId.NEXTVAL,'781');

Select\* From Item\_Cate;



**9.emp\_cust:**

INSERT INTO emp\_cust(Employee\_Id,Customer\_id) VALUES(emp\_custid.NEXTVAL,'333')

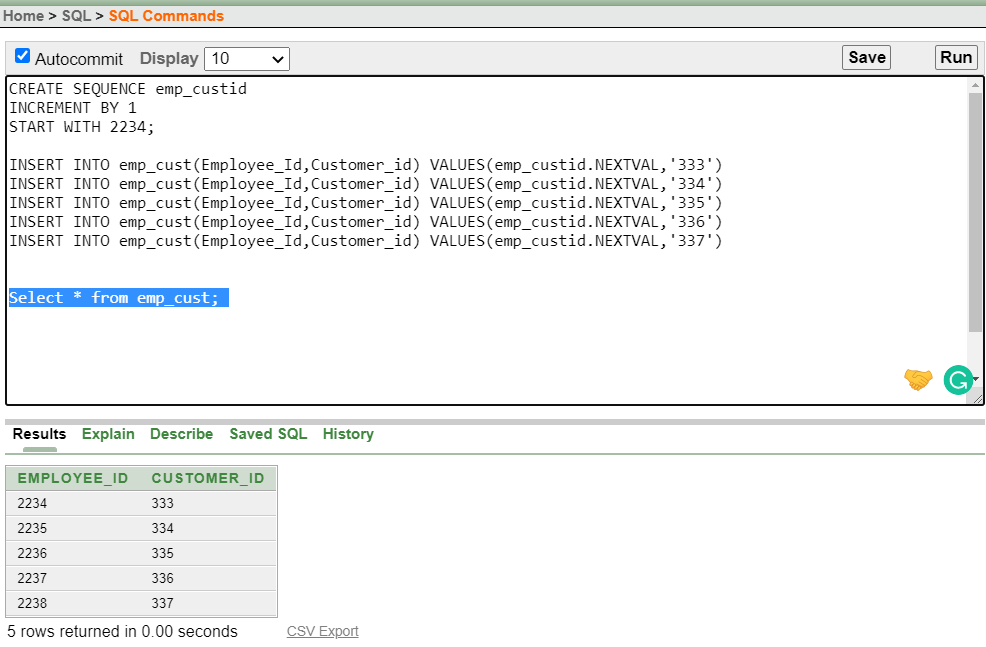
INSERT INTO emp\_cust(Employee\_Id,Customer\_id) VALUES(emp\_custid.NEXTVAL,'334')

INSERT INTO emp\_cust(Employee\_Id,Customer\_id) VALUES(emp\_custid.NEXTVAL,'335')

INSERT INTO emp\_cust(Employee\_Id,Customer\_id) VALUES(emp\_custid.NEXTVAL,'336')

INSERT INTO emp\_cust(Employee\_Id,Customer\_id) VALUES(emp\_custid.NEXTVAL,'337')

Select \* from emp\_cust;



# Query Writing:

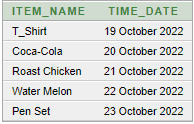
**-3 single row function**

Q1:Use TO\_CHAR function with dates to show item enlist date

Ans: SELECT Item\_name,

TO\_CHAR(Time\_Date, 'fmDD Month YYYY') Time\_Date

FROM Transaction;



Q2:Use CONCAT function to concatenate two string values.

Ans: SELECT CONCAT (Item\_name, Brand)

FROM Item

WHERE rownum < 6;

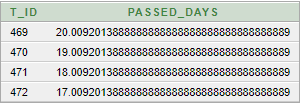


Q3: Calculate how many days have passed since one product purchase using the date arithmetic function.

Ans: SELECT T\_id, (sysdate - Time\_Date) Passed\_days

FROM Transaction

WHERE rownum < 5;



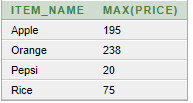
**-3 group function**Q1:Display the Item\_name which is more costly than egg and group by its name

Ans: SELECT Item\_name, max(Price)

FROM Item

GROUP BY Item\_name

HAVING max(Price)>9;



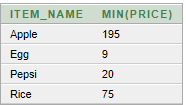
Q2:Display the Item\_name which is more costlier than egg and group by its name

Ans: SELECT Item\_name, min(Price)

FROM Item

GROUP BY Item\_name

HAVING min(Price)<238;



Q3:Display the oldest and latest date of transaction

Ans: SELECT MIN (Time\_Date) oldest, MAX (Time\_Date) latest

FROM Transaction;



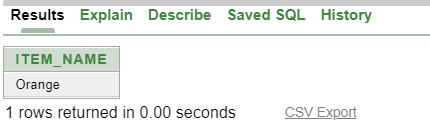
**-3 subquery**

Q1:Display the Item\_name which is more less costlier than orange.

Ans:

Select \* from Item;

select Item\_name from Item where Price> (select Price from Item where Item\_name ='Apple')

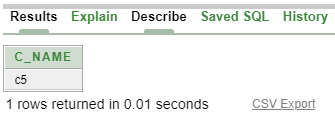


Q2: Display the c\_name who use email address "c5@gmail.com"

Ans:

Select \* from Customer;

select c\_name from Customer where c\_Email = (select c\_Email from Customer where c\_Email ='c5@gmail.com');



Q3: Display which items are bought earlier than pen set

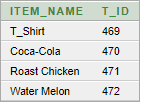
Ans: SELECT Item\_name,T\_Id

FROM Transaction

WHERE Time\_Date< (SELECT Time\_Date

FROM Transaction

WHERE T\_id=473);



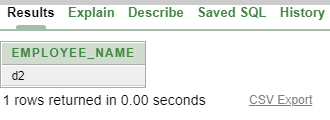
**-3 joining**

Q1: Display the name of the employee who give services to customer id 334.

Ans: SELECT Employee.Employee\_Name

from Employee ,emp\_cust

where Employee.Employee\_id=emp\_cust.Employee\_id and emp\_cust.Customer\_id='334';



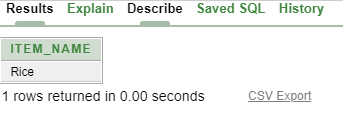
Q2: Display the name of the Item which one belongs to category id "780"

Ans:

SELECT Item.Item\_name

from Item,Item\_Cate

where Item.Category\_id=Item\_Cate.Category\_id and Item\_Cate.Category\_id='780';



Q3: Display the name of the Item which bought by the particular customer

Ans: SELECT Customer.c\_name from Customer,Transaction where Customer.Customer\_id=Transaction .Customer\_id and Transaction .Customer\_id =336;



**-3 View**

**Q1:** Create a view called CategoryView based on the Category\_id from the Category table.

**Ans:**

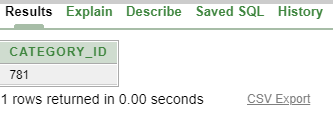
CREATE VIEW CategoryView

AS SELECT Category\_id

FROM Category

WHERE Category\_name = 'cate5';

Select \* from CategoryView;



**Q2:** Create a view called ItemView based on the Item\_name and Price from the Item table.

**Ans:**

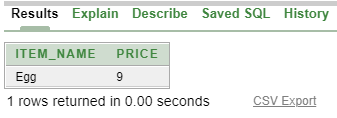
CREATE VIEW ItemView

AS SELECT Item\_name,Price

FROM Item

WHERE Id = 6688;

Select \* from ItemView;



**Q3:** Create a view called ItemInfo that contains item name,price and brand

**Ans:** CREATE VIEW ItemInfo

AS SELECT Item\_name , Price,Brand

FROM Item

where Id=6686;

**-3 synonym**

**Q1:** Create a synonym for view called ItemInfo

**Ans:** CREATE SYNONYM ItemI

For ItemInfo;

**Q2:** Create a synonym for view called ItemView

**Ans:** CREATE SYNONYM ItemV

For ItemView;

**Q3:** Create a synonym for view called CategoryView

**Ans:** CREATE SYNONYM CateV

For CategoryView;

# **Conclusion:**

This is a primary scenario of a superstore management system. The main lacking of this application is we cannot add all the sectors of a superstore. In reality the superstore management system is too big and the stored data is huge. But it might be helpful for them who want to build a superstore. Owner can implement the management of a superstore by taking the idea