American International University- Bangladesh

Faculty of CSE

INTRODUCTION TO DATABASE

Section: F (Summer 2017-2018)

Teacher: JUENA AHMED NOSHIN

Project Name: Superstore Management System

NAME: ISLAM, JAHIDUL

ID: 17-35223-2

Content

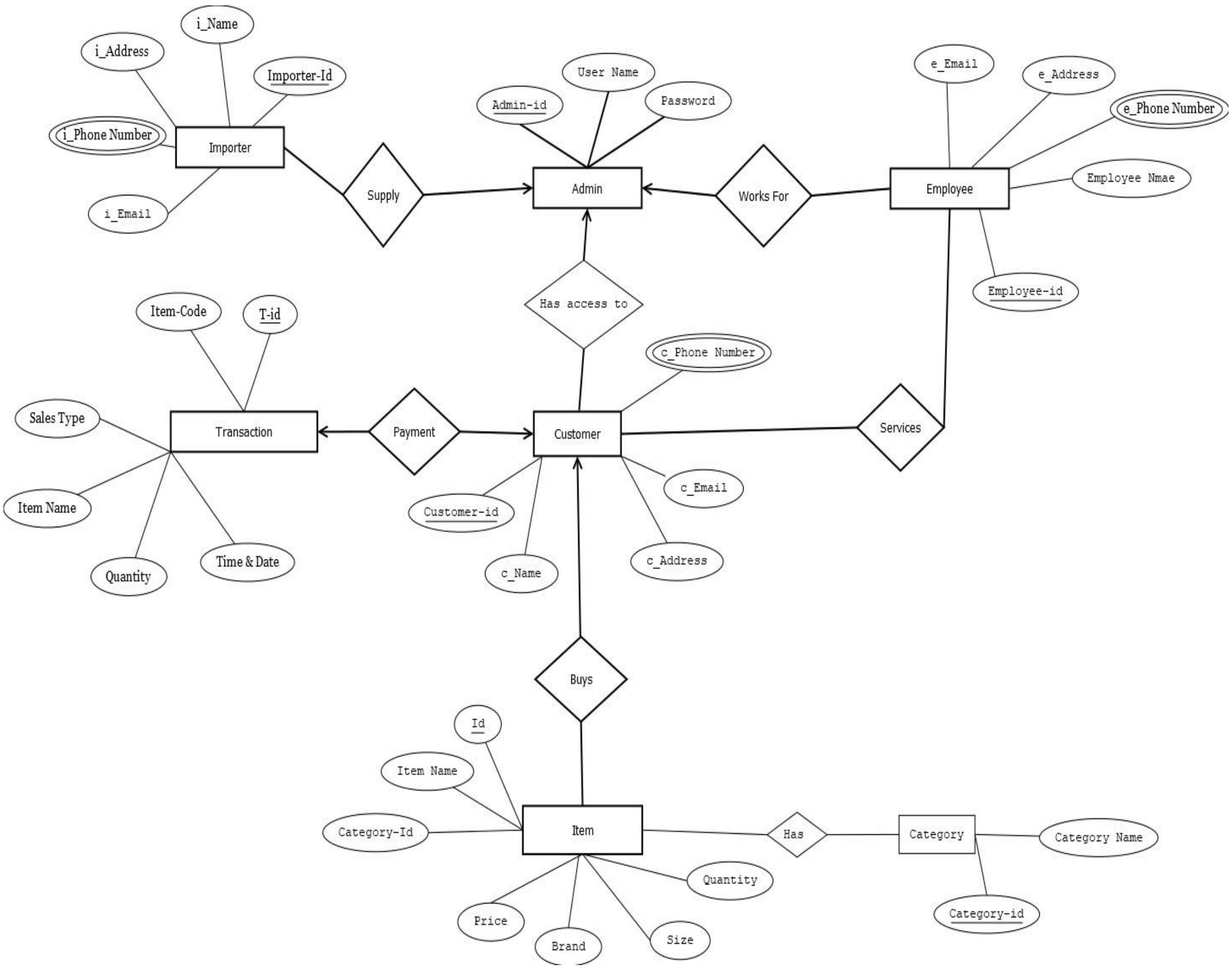
Introduction

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.

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.

ER 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

. Admin_id , User Name, Password, Importer_id, i_Name, i_Address,i_Phone Number,i_Email

NF :

Admin_id , User Name, Password

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

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

. Admin_id , User Name, Password,Employee_id, Employee Name, e_Address,e_Phone Number,e_Email

NF :

Admin_id , User Name, Password

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

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

. Employee_id,Employee Name, e_Address,e_Phone Number,e_Email,Customer-id , c_Name, c_Address, c_Phone Number,c_Email

NF :

Employee_id, Employee Name, e_Address,e_Phone Number,e_Email

Customer-id , c_Name, c_Address, c_Phone Number,c_Email

3 NF:

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

Employee_id, Employee Name, e_Address,e_Phone Number,e_Email

Customer-id , c_Name, c_Address, c_Phone Number,c_Email

Table creation :

Employee_id, Employee Name, e_Address,e_Phone Number,e_Email

Customer-id , c_Name, c_Address, c_Phone Number,c_Email

Employee_id, Customer-id

Has access to

Holds (Admin_id , User Name, Password, Customer-id , c_Name, c_Address, c_Phone Number,c_Email

)

1NF:

Phone Number is a multivalued attribute

. Admin_id , User Name, Password, Customer-id , c_Name, c_Address, c_Phone Number,c_Email

NF :

Admin_id , User Name, Password

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

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

, Customer-id , c_Name, c_Address, c_Phone Number,c_Email,T-id,Item-Code,Sales Type,Item Name,Quantity,Time & Date)

NF :

Customer-id , c_Name, c_Address, c_Phone Number,c_Email

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

3 NF:

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

Customer-id , c_Name, c_Address, c_Phone Number,c_Email

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

Table creation :

Customer-id , c_Name, c_Address, c_Phone Number,c_Email

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

Buys

Buys (Customer-id , c_Name, c_Address, c_Phone Number,c_Email, Id, Item Name,CategoryId,Price,Brand,Size,Quantity)

1NF:

Phone Number is a multivalued attribute

, Customer-id , c_Name, c_Address, c_Phone Number,c_Email, Id, Item Name,CategoryId,Price,Brand,Size,Quantity

NF :

Customer-id , c_Name, c_Address, c_Phone Number,c_Email

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

3 NF:

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

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

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

Table creation :

Customer-id , c_Name, c_Address, c_Phone Number,c_Email

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

Has

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

1NF:

Phone Number is a multivalued attribute

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

NF :

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

Category_id,Category Name

3 NF:

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

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

Category_id,Category Name

Table creation :

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

Category_id,Category Name

Id ,Category_id

Schema Diagram:

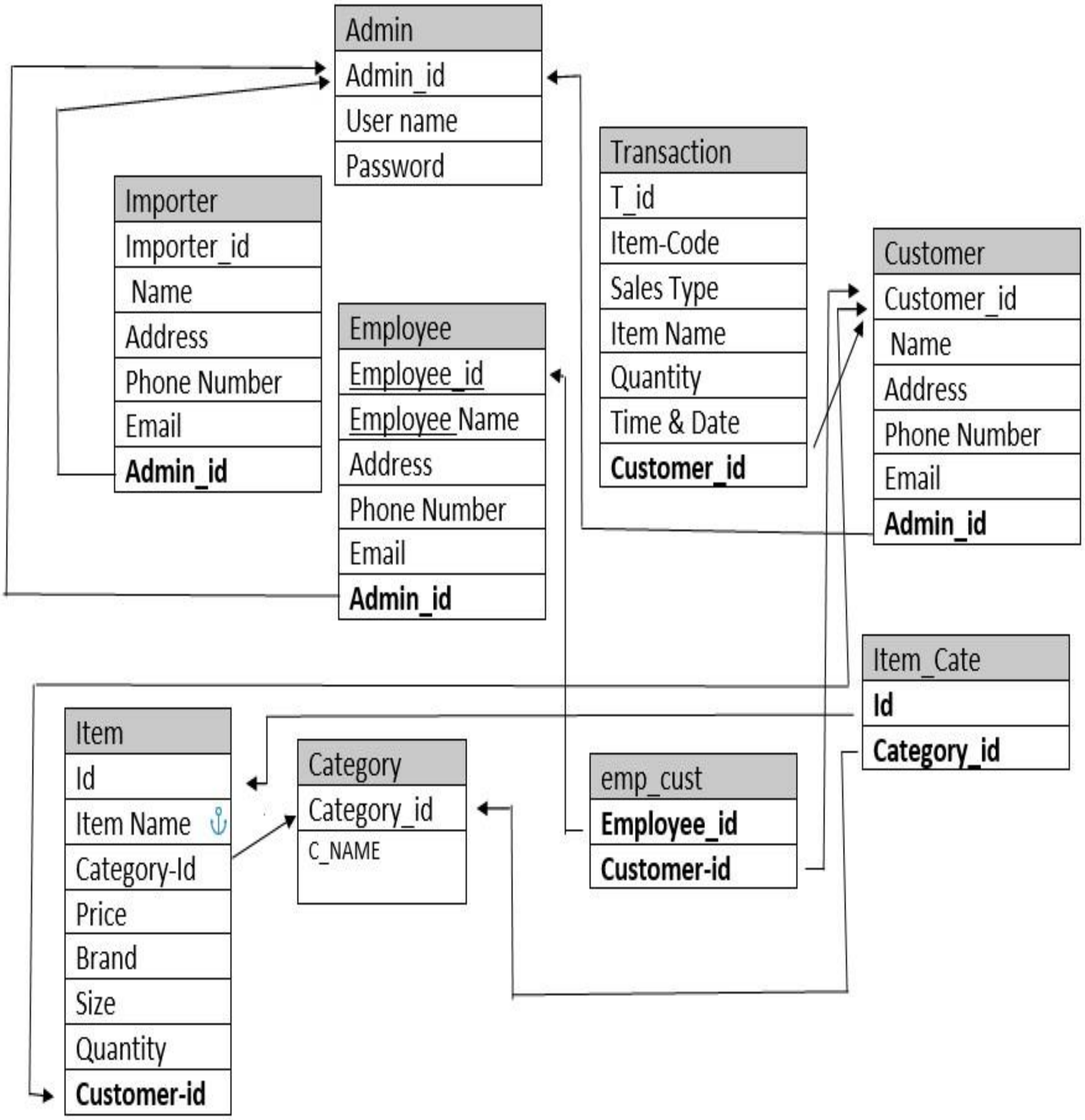


Table Creation:

-After Normalization create tables

Temporary tables

Admin_id , User Name, Password

Importer_id, i_Name, i_Address,i_Phone Number,i_Email, Admin_id

Admin_id , User Name, Password

Employee_id,Employee Name, e_Address,e_Phone Number,e_Email, Admin_id

Employee_id,Employee Name,e_ Address,e_Phone Number,e_Email,

Customer-id ,c_ Name, c_Address, c_Phone Number,c_Email

Employee_id, Customer-id

Admin_id , User Name, Password

Customer-id , c_Name, c_Address, c_Phone Number,c_Email, Admin_id

Customer-id , c_Name,c_ Address, c_Phone Number,c_Email

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

Customer-id , c_Name,c_ Address, c_Phone Number,c_Email

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

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

Category_id, Category _Name

Id ,Category_id

Final tables

Importer_id, i_Name, i_Address,i_Phone Number,i_Email, Admin_id

Employee_id,Employee Name, e_Address,e_Phone Number,e_Email, Admin_id

Employee_id, Customer-id

Admin_id , User Name, Password

Customer-id , c_Name, c_Address, c_Phone Number,c_Email, Admin_id

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

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

Category_id, Category_Name

Id ,Category_id

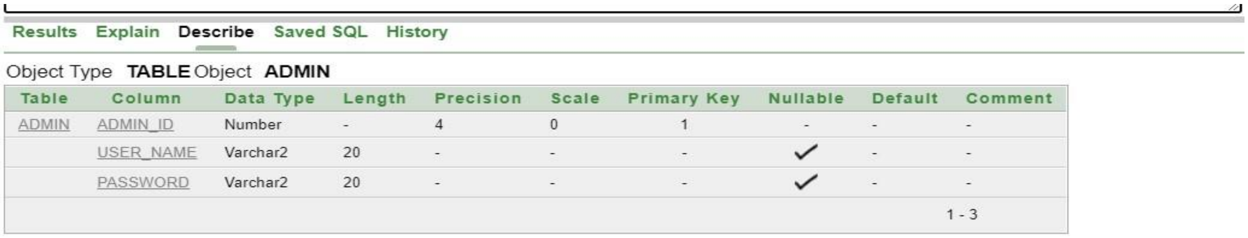
-Include constraints when creating Tables(Table's name in Dark red color)

Admin

Create table Admin(

admin_id number(4), user_name varchar2(20),password varchar2(20),CONSTRAINT pk1 PRIMARY KEY (admin_id));

Describe Admin;



Object Type TABLE Object ADMIN

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ADMIN	ADMIN_ID	Number	-	4	0	1	-	-	-
	USER_NAME	Varchar2	20	-	-	-	✓	-	-
	PASSWORD	Varchar2	20	-	-	-	✓	-	-

1 - 3

Importer

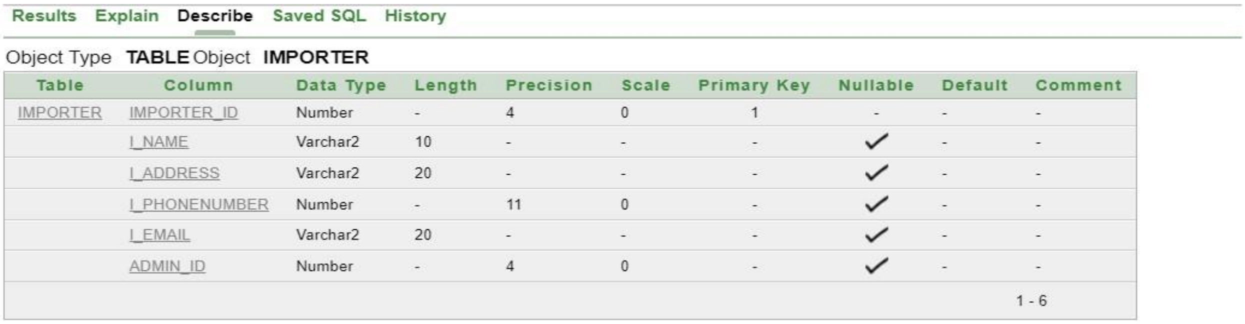
Create table Importer(

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;



Object Type TABLE Object IMPORTER

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
IMPORTER	IMPORTER_ID	Number	-	4	0	1	-	-	-
	I_NAME	Varchar2	10	-	-	-	✓	-	-
	I_ADDRESS	Varchar2	20	-	-	-	✓	-	-
	I_PHONENUMBER	Number	-	11	0	-	✓	-	-
	I_EMAIL	Varchar2	20	-	-	-	✓	-	-
	ADMIN_ID	Number	-	4	0	-	✓	-	-

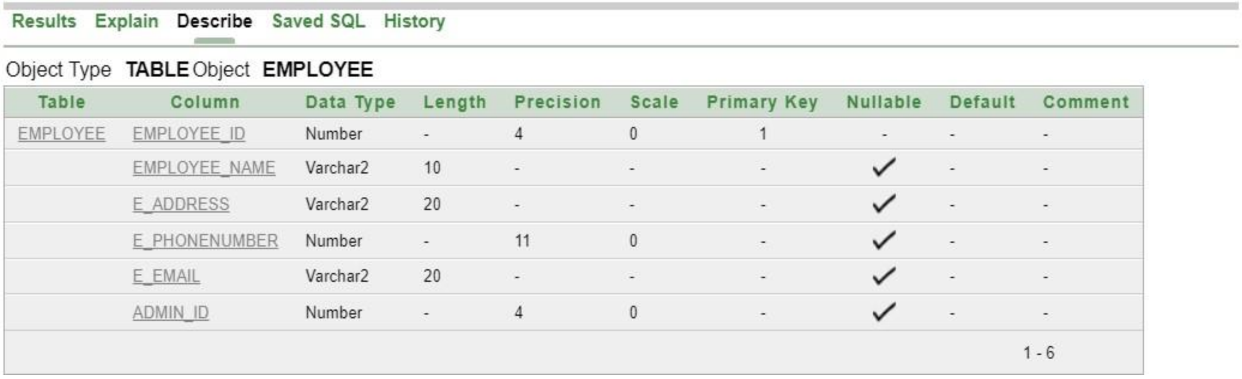
1 - 6

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;



Object Type TABLE Object EMPLOYEE

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EMPLOYEE	EMPLOYEE_ID	Number	-	4	0	1	-	-	-
	EMPLOYEE_NAME	Varchar2	10	-	-	-	✓	-	-
	E_ADDRESS	Varchar2	20	-	-	-	✓	-	-
	E_PHONENUMBER	Number	-	11	0	-	✓	-	-
	E_EMAIL	Varchar2	20	-	-	-	✓	-	-
	ADMIN_ID	Number	-	4	0	-	✓	-	-

1 - 6

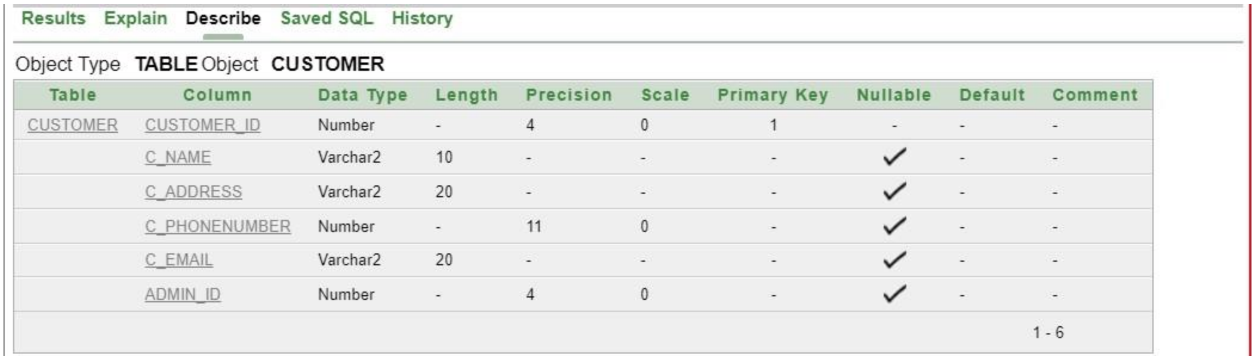
Customer

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;



Object Type TABLE Object CUSTOMER

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER	CUSTOMER_ID	Number	-	4	0	1	-	-	-
	C_NAME	Varchar2	10	-	-	-	✓	-	-
	C_ADDRESS	Varchar2	20	-	-	-	✓	-	-
	C_PHONENUMBER	Number	-	11	0	-	✓	-	-
	C_EMAIL	Varchar2	20	-	-	-	✓	-	-
	ADMIN_ID	Number	-	4	0	-	✓	-	-

1 - 6

Transaction

Create table Transaction(

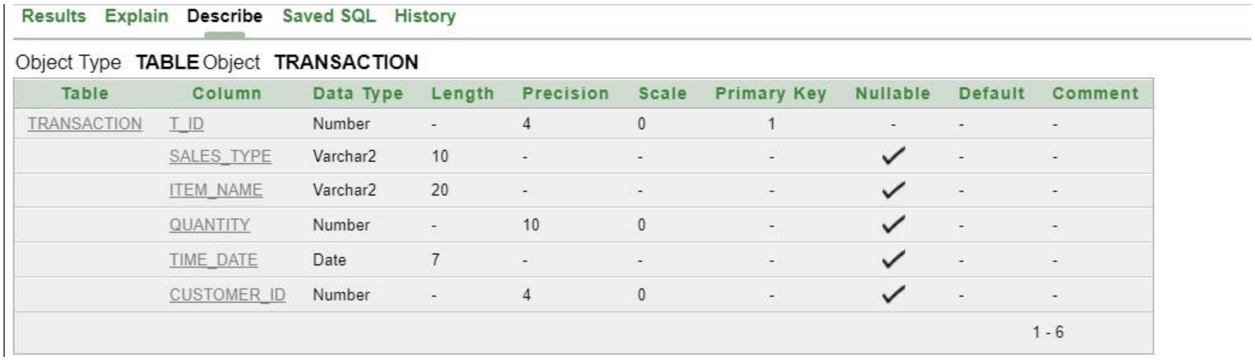
T_id number(4), Sales_Type varchar2(10),Item_name varchar2(20),Quantity number(10),Time_Date Date,Customer_id number(4), CONSTRAINT pk5 PRIMARY KEY (T_id),

CONSTRAINT fK4 FOREIGN KEY (Customer_id) REFERENCES

Customer(Customer_id)

);

Describe Transaction;



Object Type TABLE Object TRANSACTION

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TRANSACTION	T_ID	Number	-	4	0	1	-	-	-
	SALES_TYPE	Varchar2	10	-	-	-	✓	-	-
	ITEM_NAME	Varchar2	20	-	-	-	✓	-	-
	QUANTITY	Number	-	10	0	-	✓	-	-
	TIME_DATE	Date	7	-	-	-	✓	-	-
	CUSTOMER_ID	Number	-	4	0	-	✓	-	-

1 - 6

Item

Create table Item(

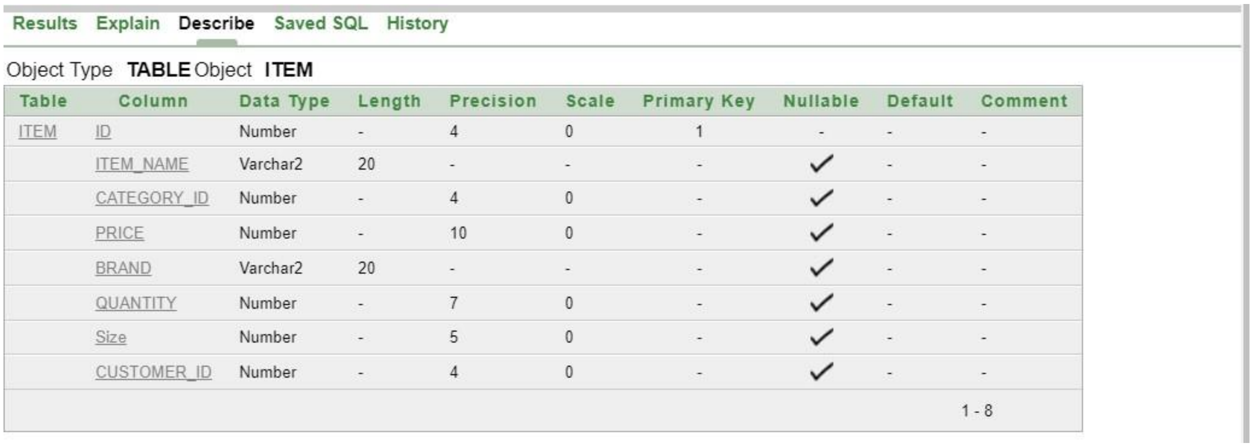
Id number(4),Item_name varchar2(20),Category_id number(4),Price number(10),Brand varchar2(20),Quantity number(7),"Size" number(5),Customer_id number(4),

CONSTRAINT pk6 PRIMARY KEY (Id),

CONSTRAINT fK5 FOREIGN KEY (Customer_id) REFERENCES Customer

(Customer_id));

Describe Item;



Object Type TABLE Object ITEM

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ITEM	ID	Number	-	4	0	1	-	-	-
	ITEM_NAME	Varchar2	20	-	-	-	✓	-	-
	CATEGORY_ID	Number	-	4	0	-	✓	-	-
	PRICE	Number	-	10	0	-	✓	-	-
	BRAND	Varchar2	20	-	-	-	✓	-	-
	QUANTITY	Number	-	7	0	-	✓	-	-
	Size	Number	-	5	0	-	✓	-	-
	CUSTOMER_ID	Number	-	4	0	-	✓	-	-

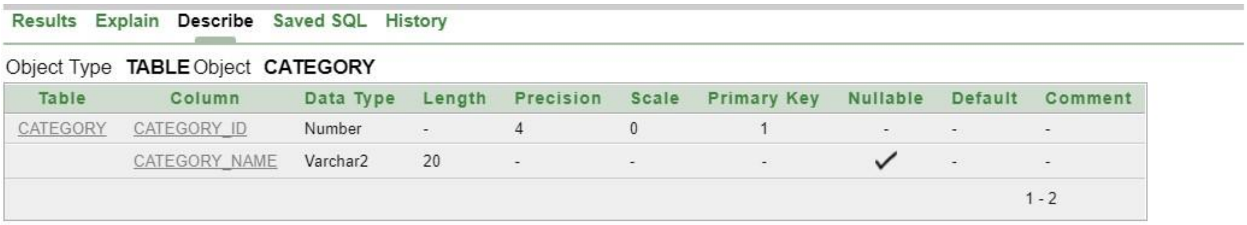
1 - 8

Category

Create table Category

(Category_id number (4), category_name varchar2(20), CONSTRAINT pk7 PRIMARY KEY (Category_id));

Describe Category;



Object Type **TABLE** Object **CATEGORY**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CATEGORY	CATEGORY_ID	Number	-	4	0	1	-	-	-
	CATEGORY_NAME	Varchar2	20	-	-	-	✓	-	-

1 - 2

Item_Cate

Create table Item_Cate

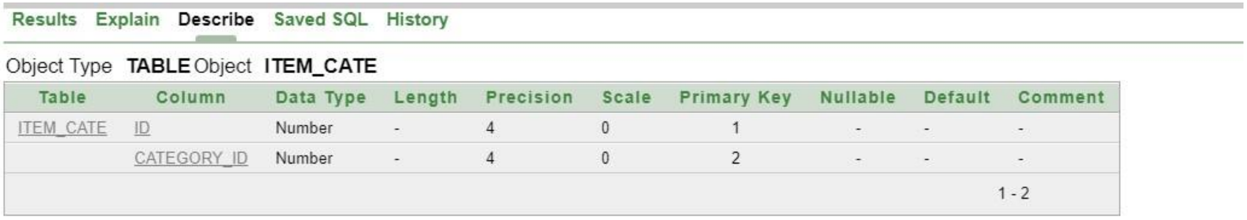
(

Id number(4), Category_id number(4),

CONSTRAINT pk8 PRIMARY KEY (Id,Category_id)

);

Describe Item_Cate;



Object Type **TABLE** Object **ITEM_CATE**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ITEM_CATE	ID	Number	-	4	0	1	-	-	-
	CATEGORY_ID	Number	-	4	0	2	-	-	-

1 - 2

emp_cust

Create table emp_cust

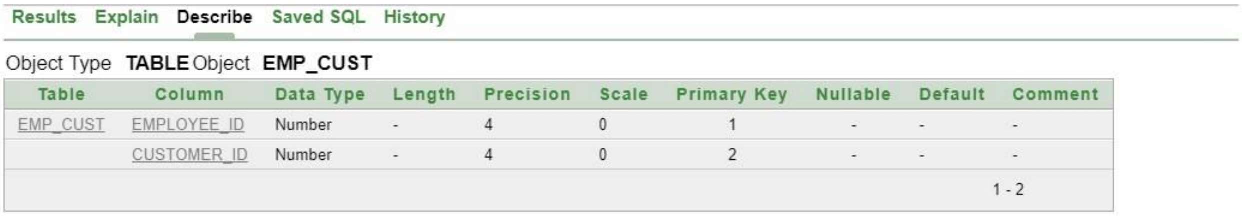
(

Employee_id number(4), Customer_id number(4),

CONSTRAINT pk9 PRIMARY KEY (Employee_id,Customer_id)

);

Describe emp_cust;



Object Type **TABLE** Object **EMP_CUST**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EMP_CUST	EMPLOYEE_ID	Number	-	4	0	1	-	-	-
	CUSTOMER_ID	Number	-	4	0	2	-	-	-

1 - 2

-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;
```

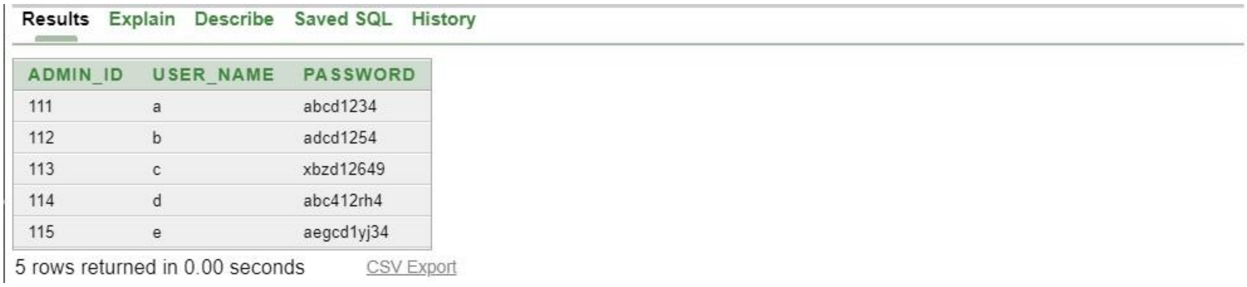
```
9.CREATE SEQUENCE emp_custid
INCREMENT BY 1
START WITH 2234;
```

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;
```



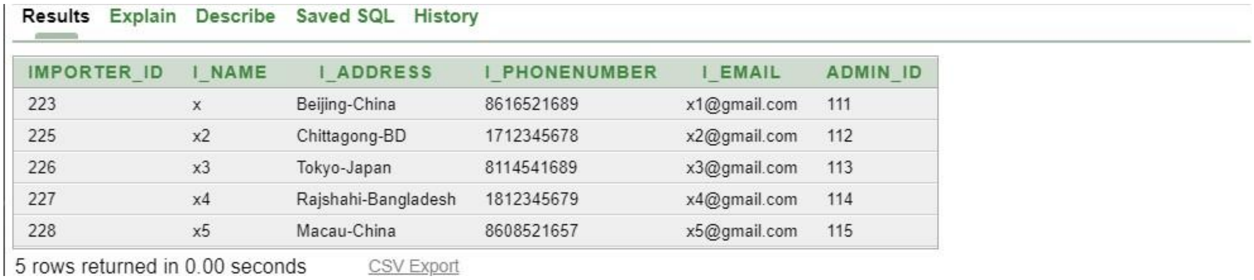
ADMIN_ID	USER_NAME	PASSWORD
111	a	abcd1234
112	b	adcd1254
113	c	xbzd12649
114	d	abc412rh4
115	e	aegcd1yj34

5 rows returned in 0.00 seconds

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', 'MacauChina','+8608521657','x5@gmail.com','115');
```

```
Select * from Importer;
```



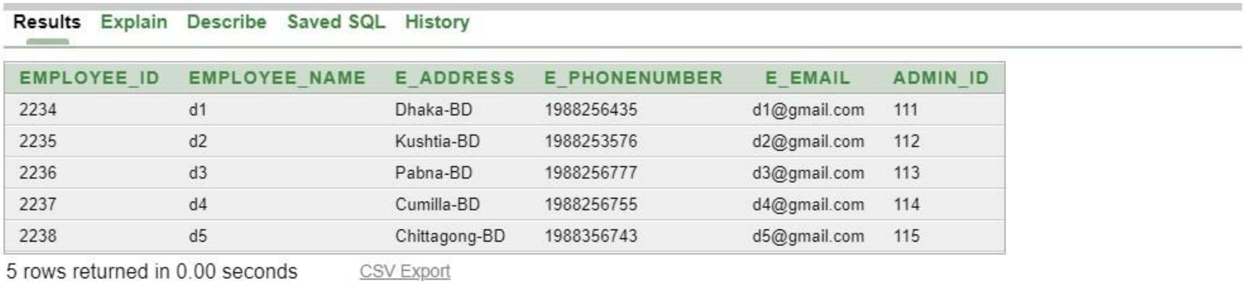
IMPORTER_ID	I_NAME	I_ADDRESS	I_PHONENUMBER	I_EMAIL	ADMIN_ID
223	x	Beijing-China	8616521689	x1@gmail.com	111
225	x2	Chittagong-BD	1712345678	x2@gmail.com	112
226	x3	Tokyo-Japan	8114541689	x3@gmail.com	113
227	x4	Rajshahi-Bangladesh	1812345679	x4@gmail.com	114
228	x5	Macau-China	8608521657	x5@gmail.com	115

5 rows returned in 0.00 seconds

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;
```



EMPLOYEE_ID	EMPLOYEE_NAME	E_ADDRESS	E_PHONENUMBER	E_EMAIL	ADMIN_ID
2234	d1	Dhaka-BD	1988256435	d1@gmail.com	111
2235	d2	Kushtia-BD	1988253576	d2@gmail.com	112
2236	d3	Pabna-BD	1988256777	d3@gmail.com	113
2237	d4	Cumilla-BD	1988256755	d4@gmail.com	114
2238	d5	Chittagong-BD	1988356743	d5@gmail.com	115

5 rows returned in 0.00 seconds

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-

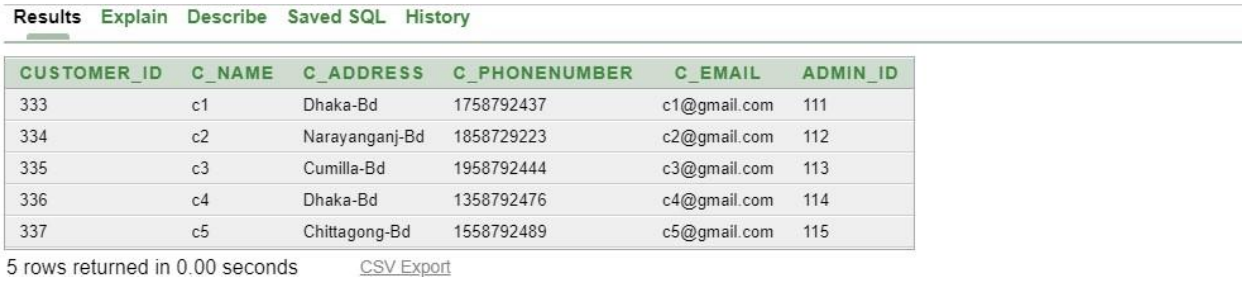
Bd','01358792476','c4@gmail.com','114');

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;



CUSTOMER_ID	C_NAME	C_ADDRESS	C_PHONENUMBER	C_EMAIL	ADMIN_ID
333	c1	Dhaka-Bd	1758792437	c1@gmail.com	111
334	c2	Narayanganj-Bd	1858729223	c2@gmail.com	112
335	c3	Cumilla-Bd	1958792444	c3@gmail.com	113
336	c4	Dhaka-Bd	1358792476	c4@gmail.com	114
337	c5	Chittagong-Bd	1558792489	c5@gmail.com	115

5 rows returned in 0.00 seconds CSV Export

5.Transaction:

INSERT INTO Transaction(T_id, Sales_Type, Item_name,Quantity,Time_Date,Customer_id) VALUES (Transaction_Tid.NEXTVAL, 'Online', 'T_Shirt','3','20-FEB-2021','333');

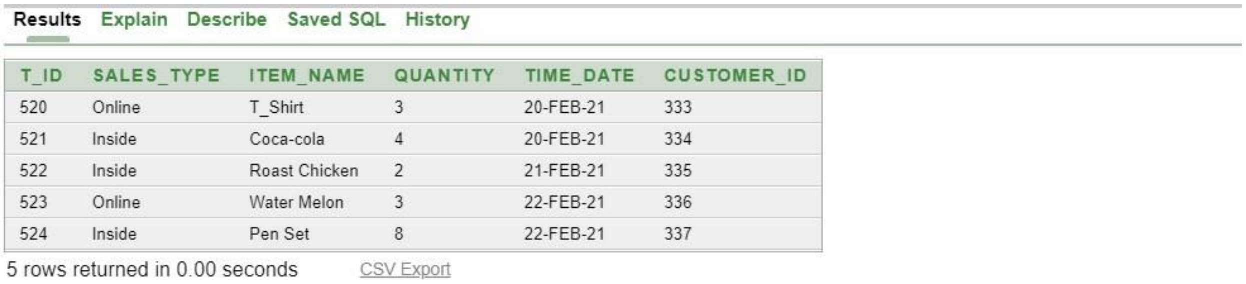
INSERT INTO Transaction(T_id, Sales_Type, Item_name,Quantity,Time_Date,Customer_id) VALUES (Transaction_Tid.NEXTVAL, 'Inside ', 'Coca-cola','4','20-FEB-2021','334');

INSERT INTO Transaction(T_id, Sales_Type, Item_name,Quantity,Time_Date,Customer_id) VALUES (Transaction_Tid.NEXTVAL, 'Inside ', 'Roast Chicken','2','21-FEB-2021','335');

INSERT INTO Transaction(T_id, Sales_Type, Item_name,Quantity,Time_Date,Customer_id)

VALUES (Transaction_Tid.NEXTVAL, 'Online', 'Water Melon','3','22-FEB-2021','336'); INSERT INTO Transaction(T_id, Sales_Type, Item_name,Quantity,Time_Date,Customer_id)

VALUES (Transaction_Tid.NEXTVAL, 'Inside ', 'Pen Set','8','22-FEB-2021','337'); Select * from Transaction;



T_ID	SALES_TYPE	ITEM_NAME	QUANTITY	TIME_DATE	CUSTOMER_ID
520	Online	T_Shirt	3	20-FEB-21	333
521	Inside	Coca-cola	4	20-FEB-21	334
522	Inside	Roast Chicken	2	21-FEB-21	335
523	Online	Water Melon	3	22-FEB-21	336
524	Inside	Pen Set	8	22-FEB-21	337

5 rows returned in 0.00 seconds CSV Export

6.Item:

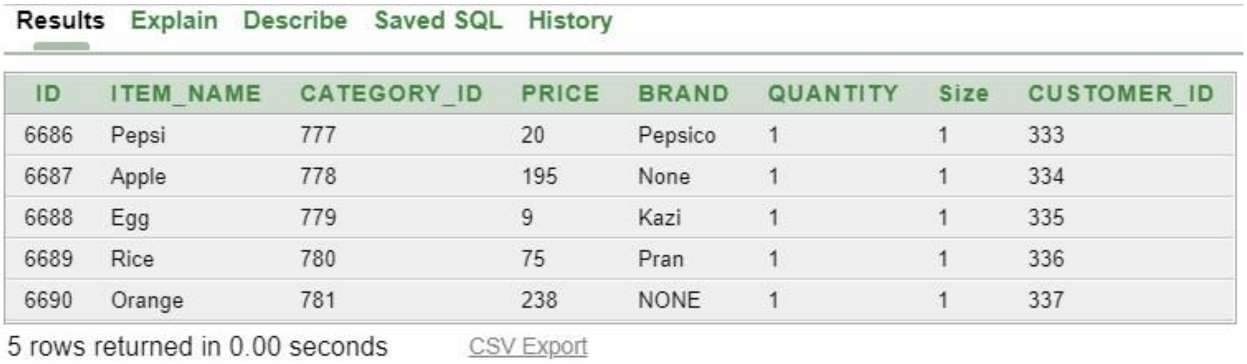
INSERT INTO Item(Id ,Item_name,Category_id,Price,Brand,Quantity,"Size",Customer_id) VALUES(Item_Id.NEXTVAL, 'Pepsi' ,'777','20','Pepsico','1','1','333');

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;



ID	ITEM_NAME	CATEGORY_ID	PRICE	BRAND	QUANTITY	Size	CUSTOMER_ID
6686	Pepsi	777	20	Pepsico	1	1	333
6687	Apple	778	195	None	1	1	334
6688	Egg	779	9	Kazi	1	1	335
6689	Rice	780	75	Pran	1	1	336
6690	Orange	781	238	NONE	1	1	337

5 rows returned in 0.00 seconds CSV Export

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');

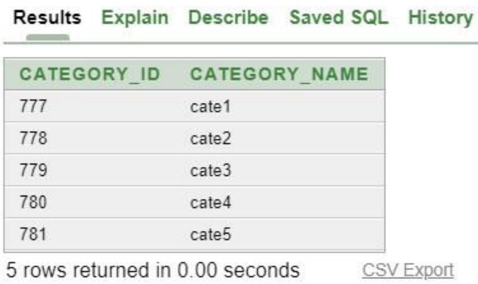
INSERT INTO Category(Category_id, Category_name) VALUES (Category_Categoryid.NEXTVAL, 'cate3');

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;



CATEGORY_ID	CATEGORY_NAME
777	cate1
778	cate2
779	cate3
780	cate4
781	cate5

5 rows returned in 0.00 seconds CSV Export

8.Item_Cate:

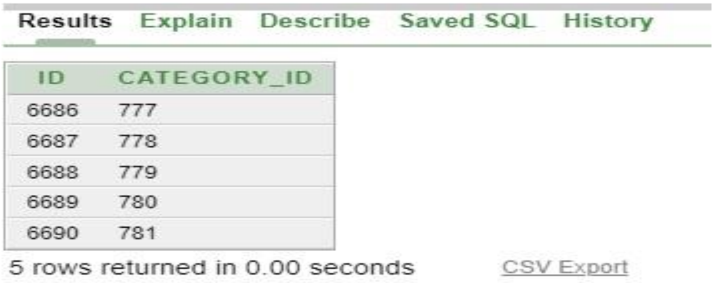
INSERT INTO Item_Cate(Id,Category_id) VALUES(Item_CateId.NEXTVAL,'777');

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;



ID	CATEGORY_ID
6686	777
6687	778
6688	779
6689	780
6690	781

5 rows returned in 0.00 seconds CSV Export

9.emp_cust:

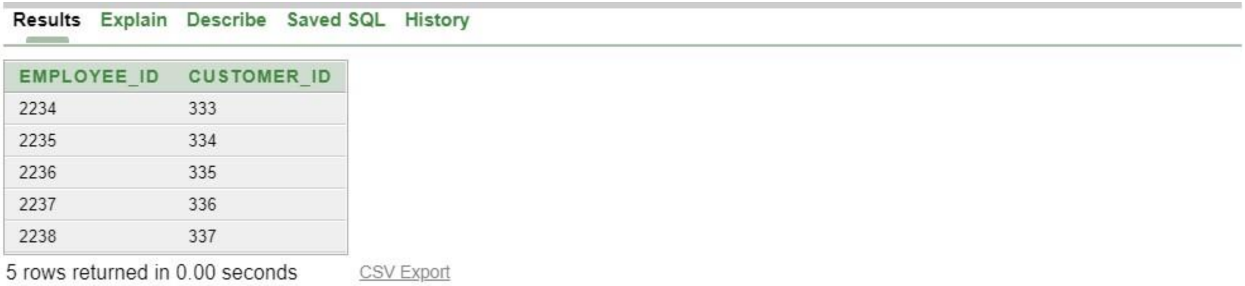
INSERT INTO emp_cust(Employee_Id,Customer_id) VALUES(emp_custid.NEXTVAL,'333')

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;



EMPLOYEE_ID	CUSTOMER_ID
2234	333
2235	334
2236	335
2237	336
2238	337

5 rows returned in 0.00 seconds

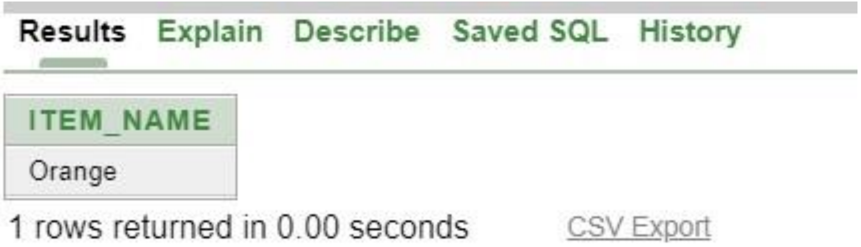
Query Writing:

-2 subquery

Q1:Display the Item_name which is more costly than Item Apple.

Ans:

Select * from Item; select Item_name from Item where Price> (select Price from Item where Item_name ='Apple')



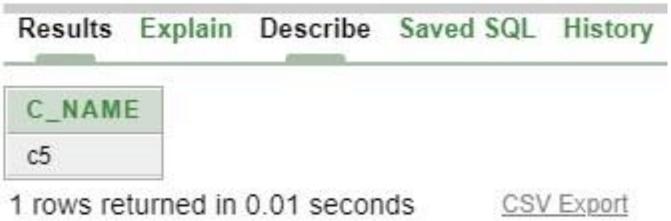
ITEM_NAME
Orange

1 rows returned in 0.00 seconds

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');

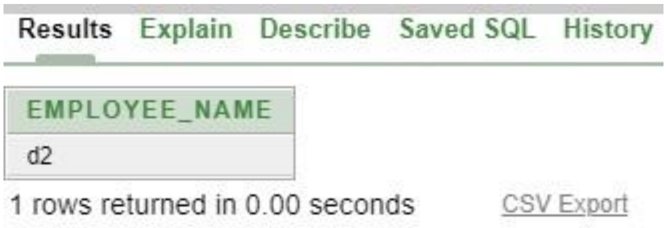


C_NAME
c5

1 rows returned in 0.01 seconds

-2 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';

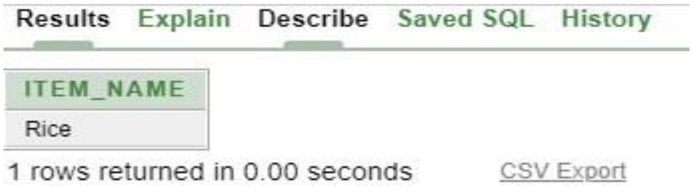


EMPLOYEE_NAME
d2

1 rows returned in 0.00 seconds

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';



ITEM_NAME
Rice

1 rows returned in 0.00 seconds

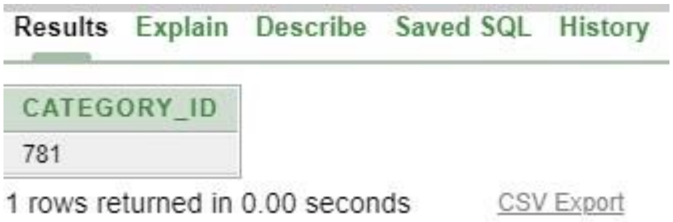
-2 View

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;
```



CATEGORY_ID
781

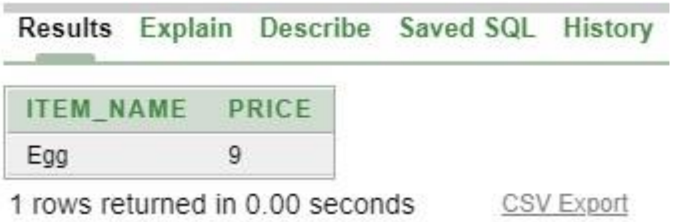
1 rows returned in 0.00 seconds

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;



ITEM_NAME	PRICE
Egg	9

1 rows returned in 0.00 seconds

Relational Algebra

Select only those rows of data where c_address is Dhaka-Bd from the Customer Table.

Ans: σc_address = ”dhaka_bd” (Customer)

Find the e_phonenumber of the Employee whose employee_id is 2236.

Ans: ∏ e_phonenumber (σ employee_id =2236(Employee))

Find the t_id, sales_type and item_name where quantity is 3.

Ans: ∏t_id, sales_type and item name (σquantity = 3 (Transaction))

Find the id, item_name, price, brand and quantity where size is 1.

Ans: ∏ id, item_name, price, brand and quantity (σsize = 1 (Item))

Find the category_name where category_id is 780.

Ans: ∏ category_name (σcategory_id = 780 (Category))

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