Project Name: Ecommerce Management System

Course Name: Advanced Database Management System

Sec: A

Serial No	Name	ID	Contribution
01	SHAHRIYAR SHAZIB	18-36610-1	Scenario,Class Diagram,Er-Diagram,TableCreation,Data Insertion,Queries(single row,group,sub,join,) PLSQL(function, record, cursor)
02	MD. TANIM IFTEKHER JALAL	18-37973-2	Use Case,ActivityDiagram, interface design
03	RAHMAN, MD. SHAKIBUR	18-36598-1	Normalization,Schema Diagram PL/SQL(packege ,triggure and procedure)
04	SM BAKIBILLAH LEMON	18-38101-2	TableCreation,query (view)
05	MD. ABUL BASAR PIAS	18-38128-2	Introduction,Project Proposal,Query,(Synonyms), Conclusion

Content Table:

2.Introduction

E-commerce is fast gaining ground as an accepted and used business paradigm. More and more business houses are implementing web sites providing functionality for performing commercial transactions over the web. It is reasonable to say that the process of shopping on the web is becoming commonplace.

The objective of this project is to show the Database System of an ecommerce store where products like daily goods can be bought from the comfort of home through the Internet.

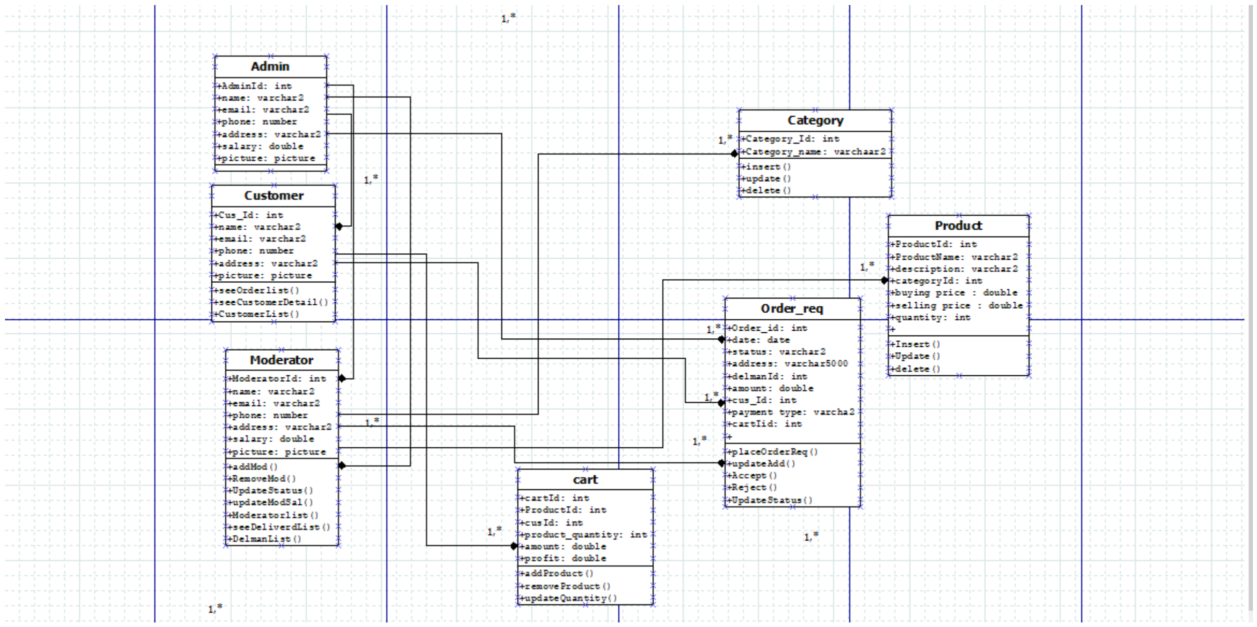
An online store is a virtual store on the Internet where customers can browse the catalog and select products of interest. The selected items may be collected in a shopping cart. At checkout time, the items in the shopping cart will be presented as an order. At that time, more information will be needed to complete the transaction. Usually, the customer will be asked to fill or select a billing address, a shipping address, a shipping option, and payment information such as a credit card number.

3.Project Proposal

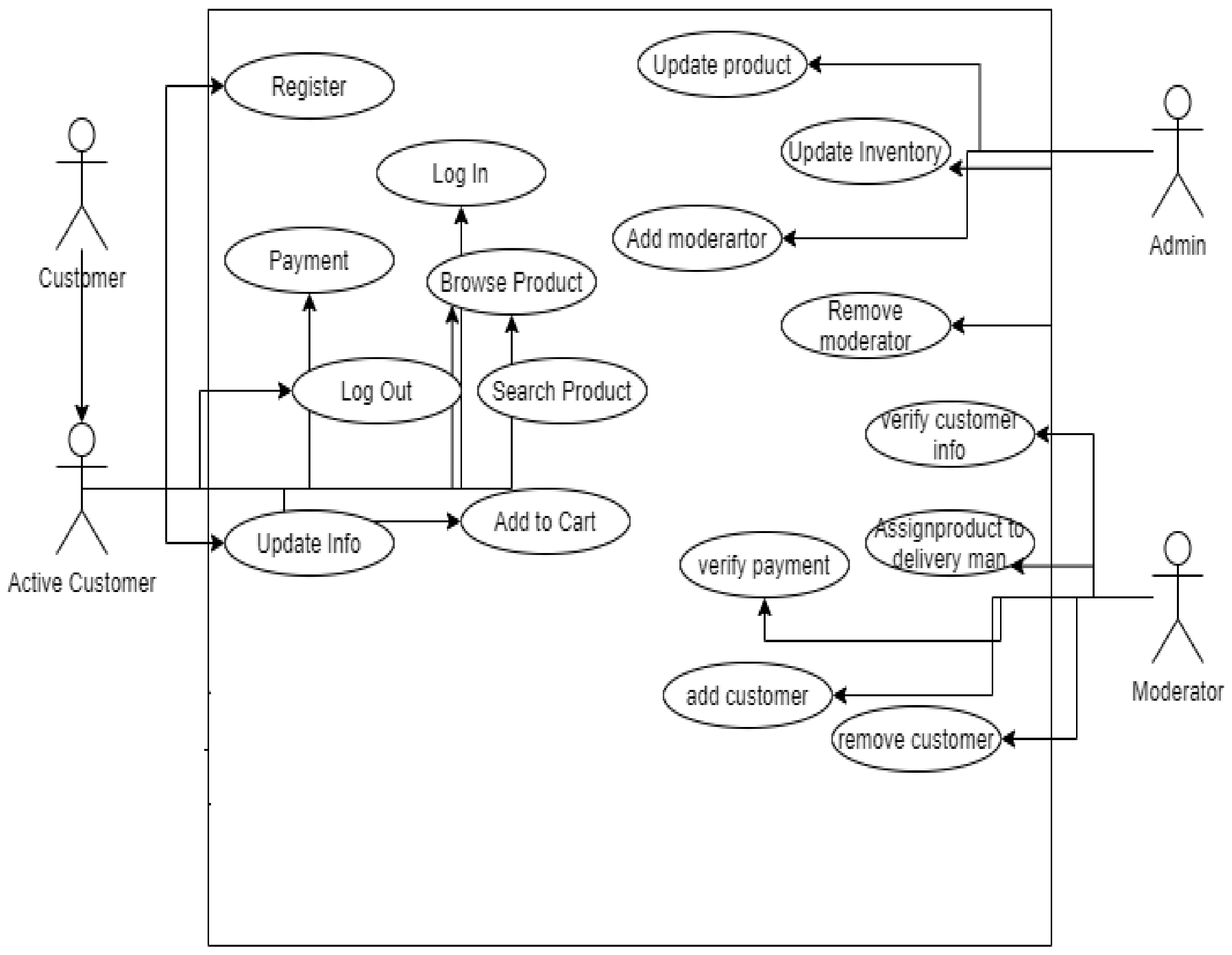
We are going to create an online E-Commerce website where customers can buy any product through our website. There will be 5 users: 1. Admin, 2. User, 3. Customers Moderator

4. Diagram

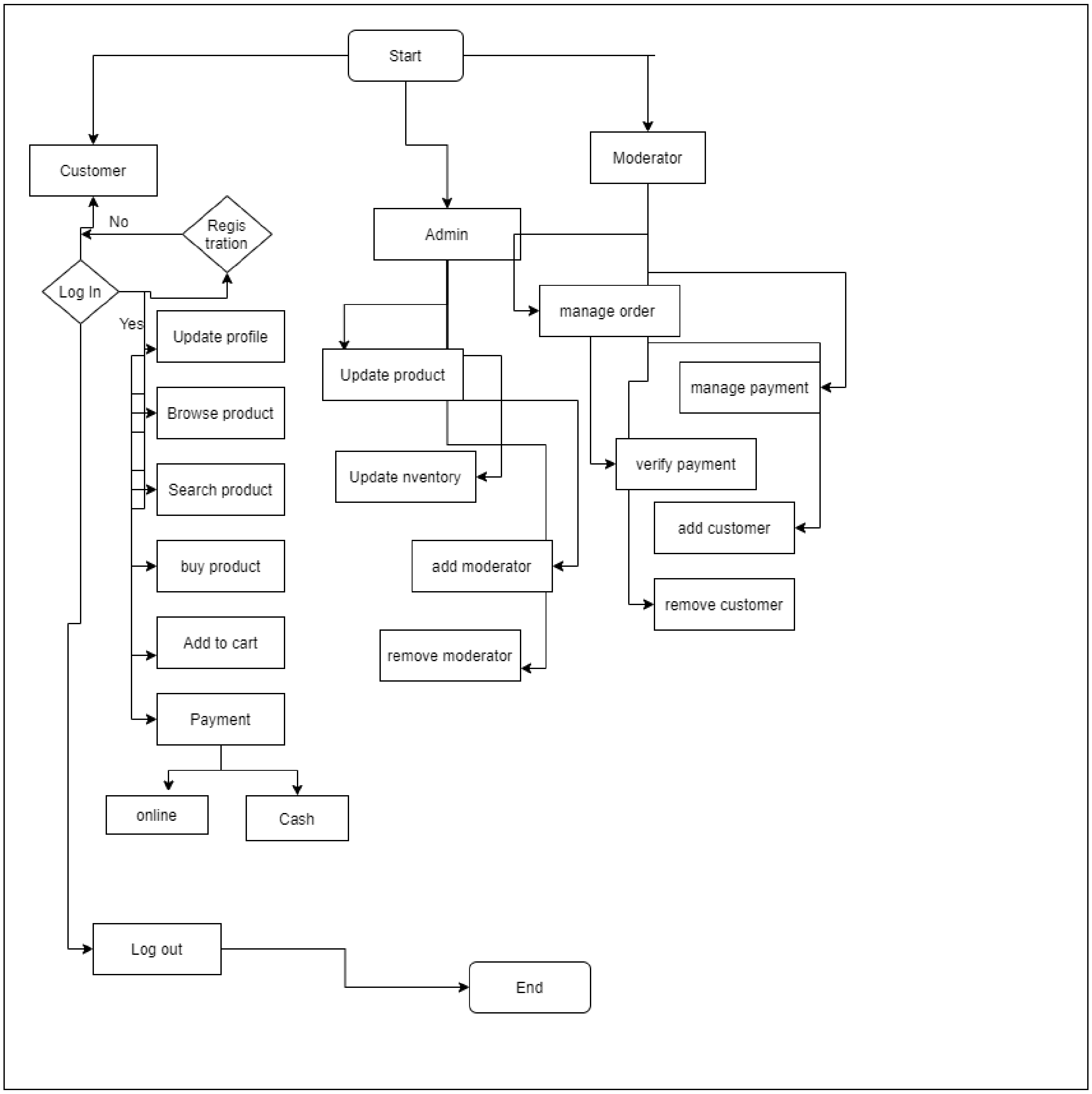
4.1. Class Diagram



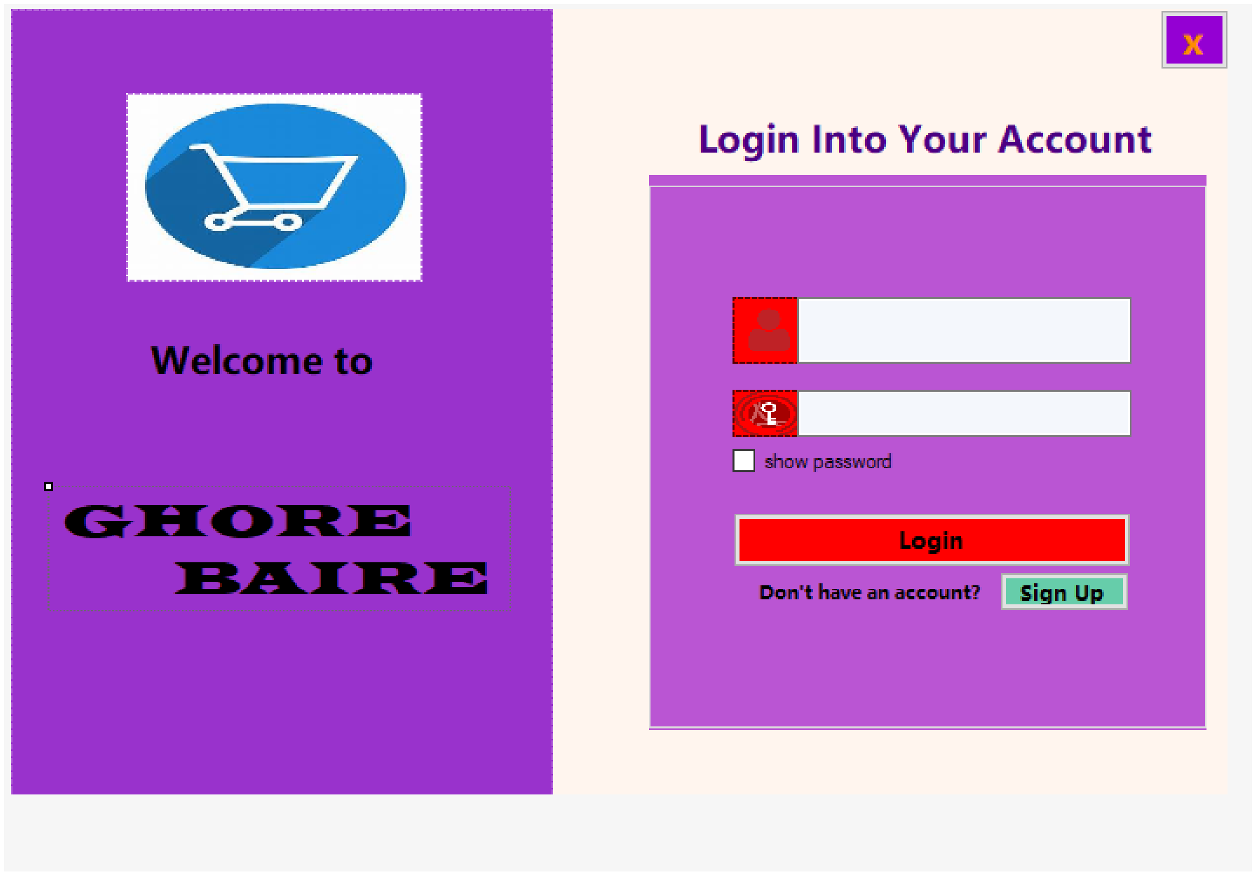
4.2. Use Case Diagram

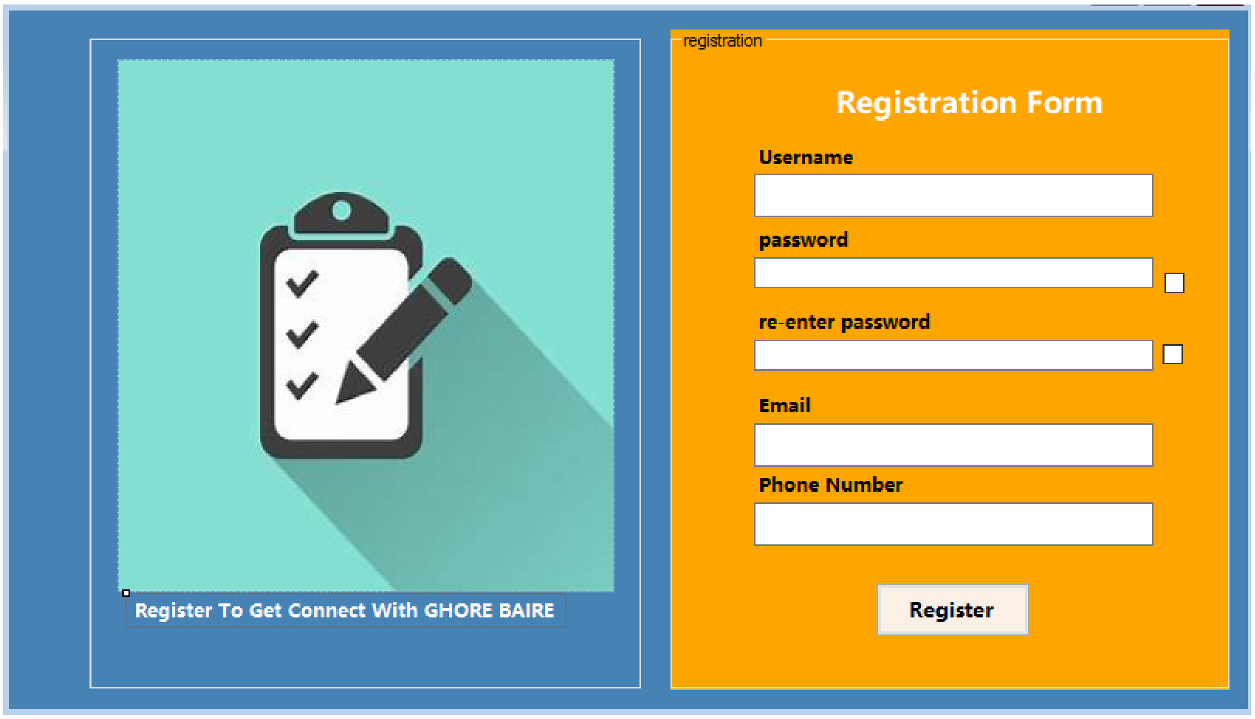


4.3. Activity Diagram



5.User Interface





registration

Registration Form

Username

password

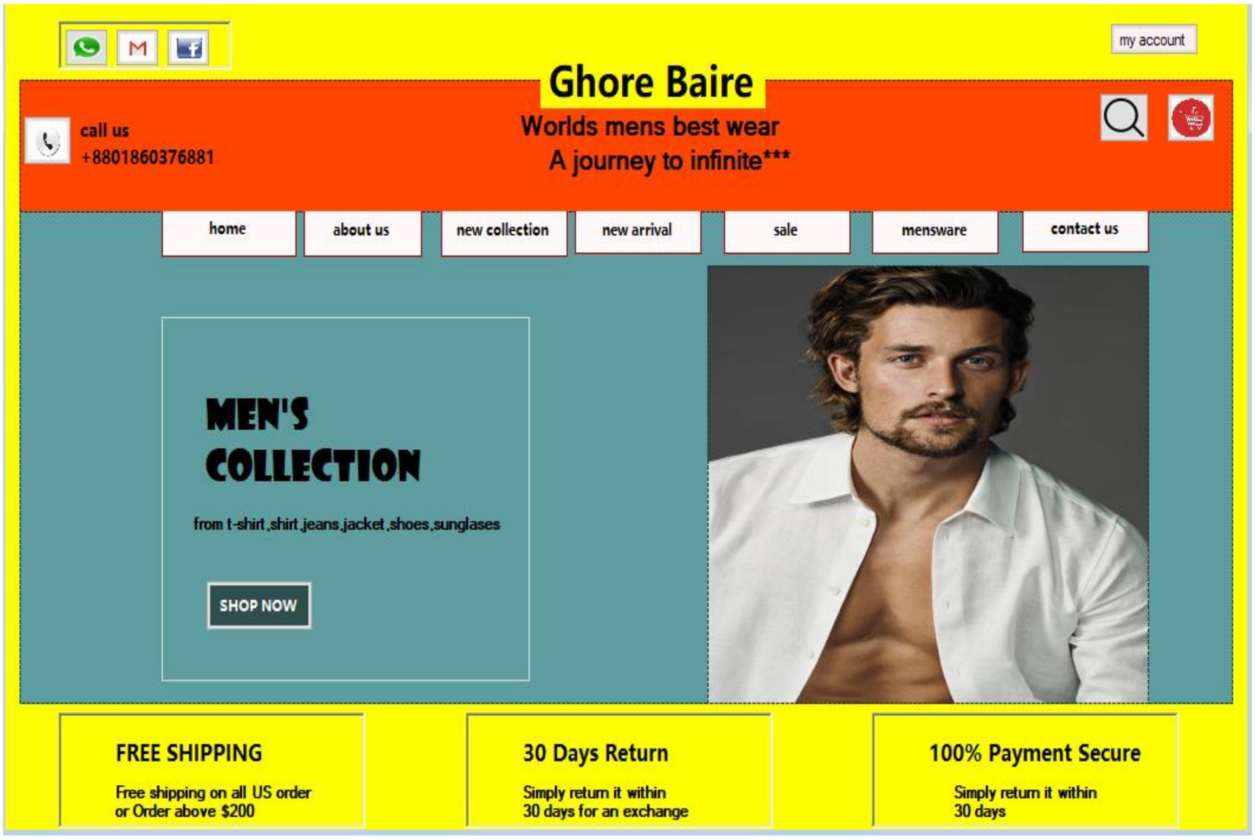
re-enter password

Email

Phone Number

Register

Register To Get Connect With GHORE BAIRE



my account

Ghore Baire

Worlds mens best wear

A journey to infinite***

call us
+8801860376881

home | about us | new collection | new arrival | sale | mensware | contact us

MEN'S COLLECTION

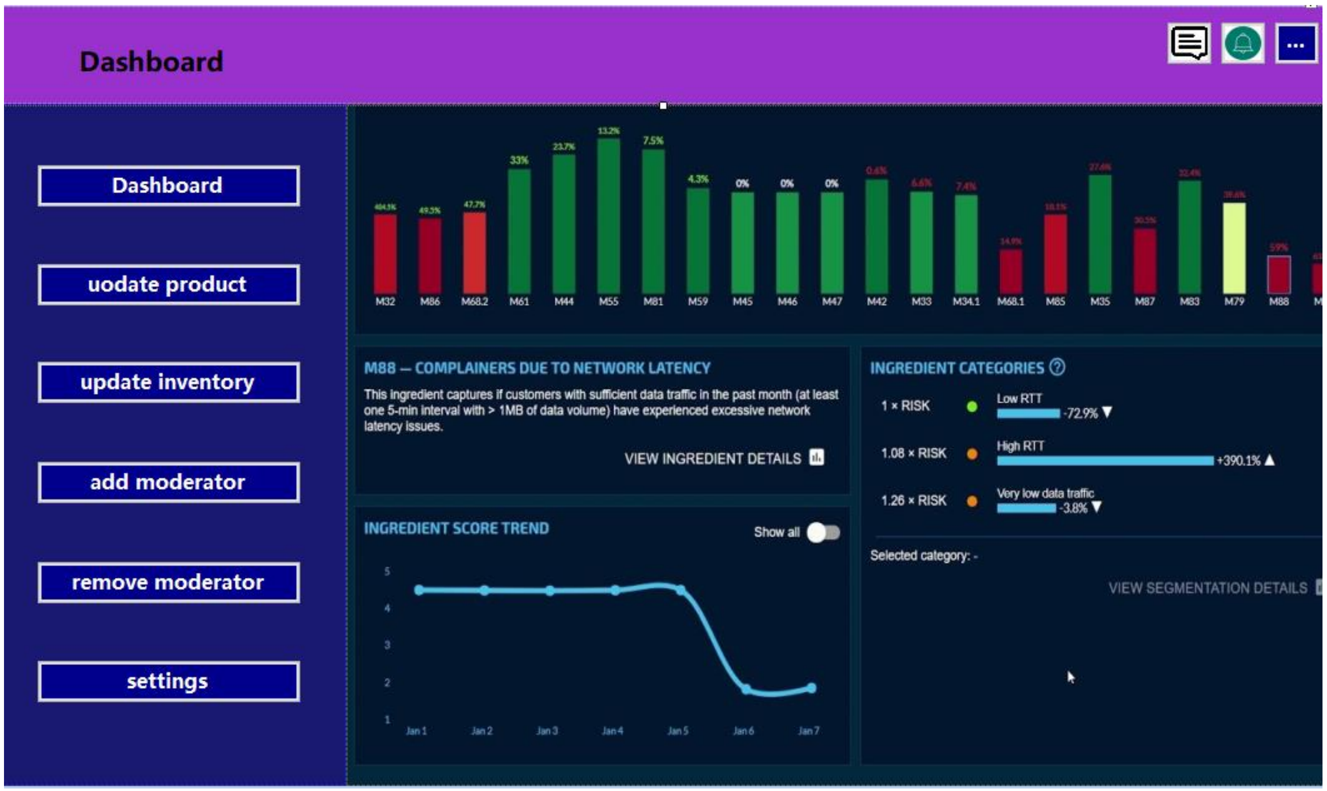
from t-shirt,shirt,jeans,jacket,shoes,sunglases

SHOP NOW

FREE SHIPPING
Free shipping on all US order
or Order above $200

30 Days Return
Simply return it within
30 days for an exchange

100% Payment Secure
Simply return it within
30 days



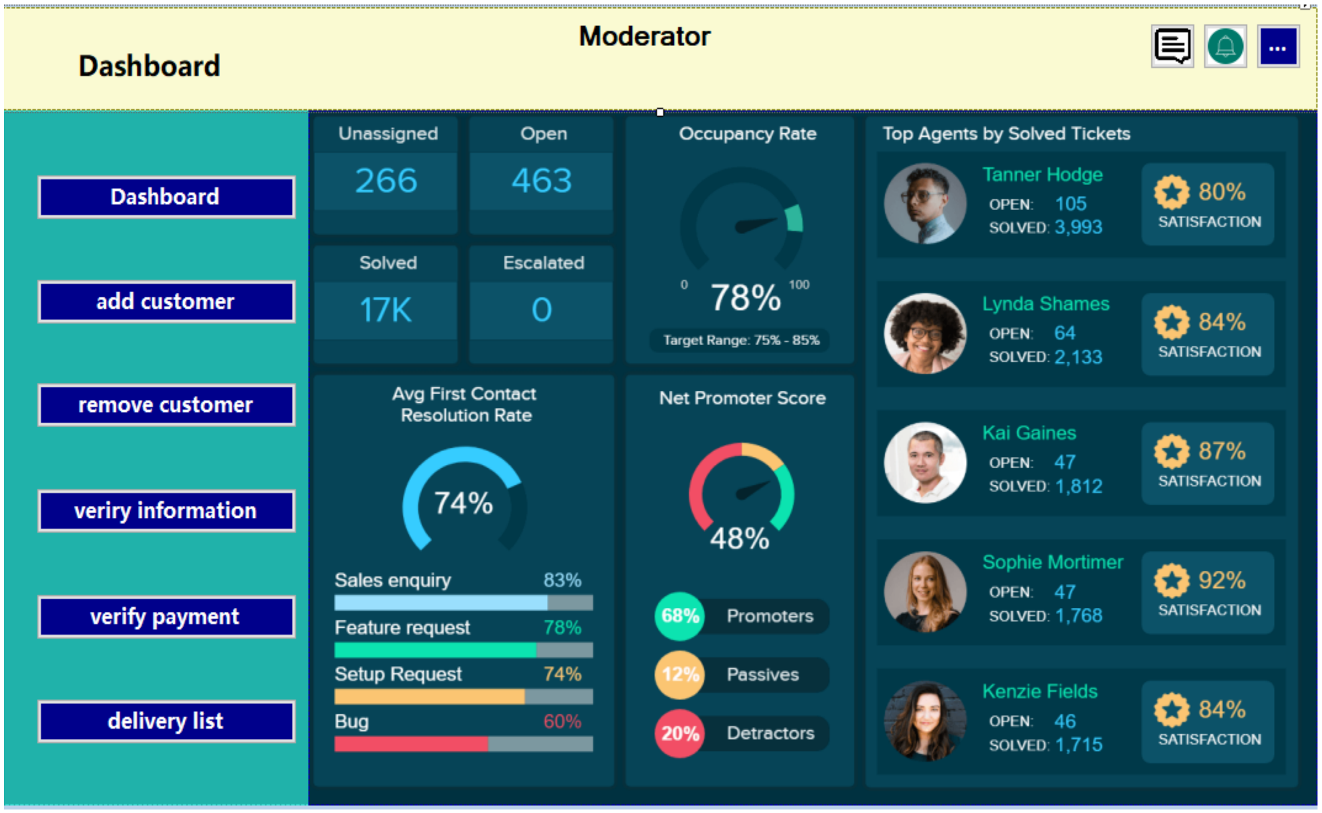
M88 — COMPLAINERS DUE TO NETWORK LATENCY

This ingredient captures if customers with sufficient data traffic in the past month (at least one 5-min interval with > 1MB of data volume) have experienced excessive network latency issues.

INGREDIENT CATEGORIES

1 × RISK	●	Low RTT -72.9% ▼
1.08 × RISK	●	High RTT +390.1% ▲
1.26 × RISK	●	Very low data traffic -3.8% ▼

Selected category: -



Unassigned	Open	Solved	Escalated
266	463	17K	0

Occupancy Rate: 78%

Avg First Contact Resolution Rate: 74%

Sales enquiry	83%
Feature request	78%
Setup Request	74%
Bug	60%

Net Promoter Score: 48%

68%	Promoters
12%	Passives
20%	Detractors

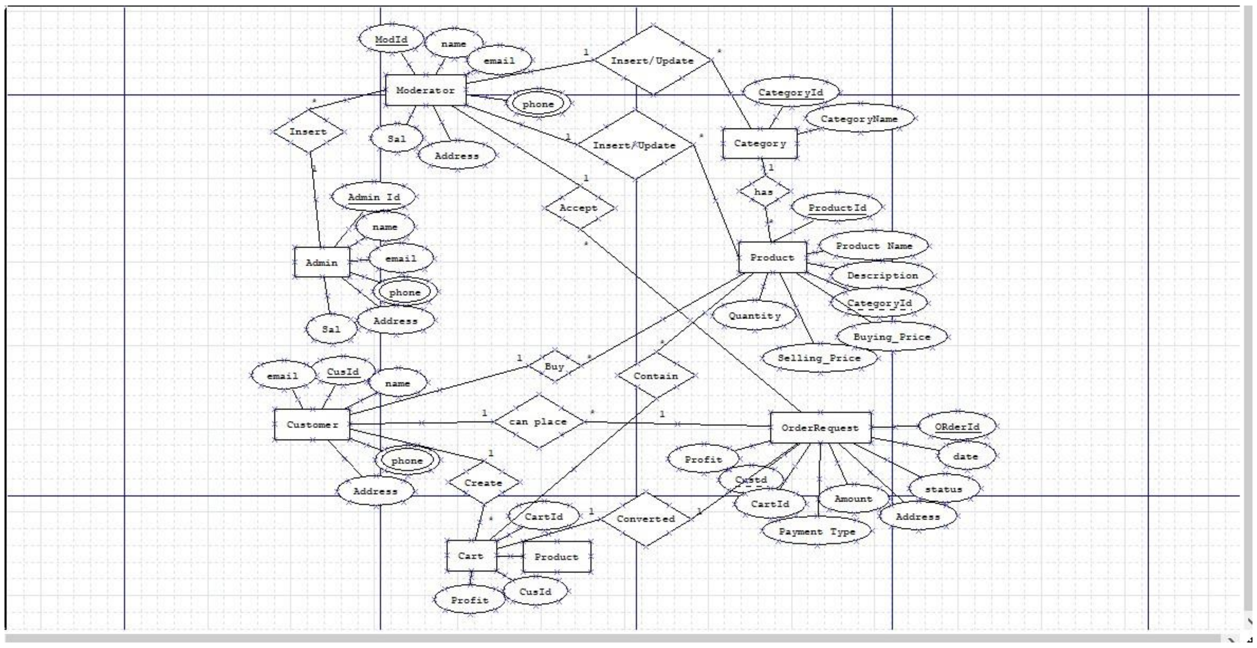
Top Agents by Solved Tickets

	OPEN	SOLVED	SATISFACTION
Tanner Hodge	105	3,993	80%
Lynda Shames	64	2,133	84%
Kai Gaines	47	1,812	87%
Sophie Mortimer	47	1,768	92%
Kenzie Fields	46	1,715	84%

6. Scenario Description:

In an e-commerce web site customer may buy or add products in their cart. Each customer has unique customer Id and other data’s like name, email, phone, address, picture, total purchase amount also stored in the system. A customer can buy multiple product but a product can only be sold to a customer. a customer can add product in their cart before confirm order. cart has unique id and with other attributes like product id, customer id, product quantity, total amount, and profit will be stored. After confirm order the order request will be stored in the order request table a customer can place multiple order. And the order table will have a unique id and date, status, address, amount, customer id, payment type, and dolmen id .for every order delivery man will be assigned by the admin or moderator the admin and moderator will have a unique id, name, email, phone, Sal, address, picture which will assign by the moderator or admin. Admin can also add moderator or deliveryman whereas a moderator can appoint a user by the permission of admin and add product and also update the product and as well as category each product will have a category but a category can have multiple products. Every product contains a unique with the name a short description, buying price, selling price, quantity. Like product category has also a category id with category name. everyone in the system is a user and the user has a unique id along with password, type, status admin can add a user and remove the user

7.ER Diagram



8. Normalization

Insert/Update(<u>ModId</u>, name,email,phone,address,sal, <u>CategoryId</u>,categoryName)
1NF:
Phone is multivalued attribute.
<u>ModId</u>, name,email,phone,address,sal,<u>CategoryId</u>,categoryName
2NF:

1. <u>ModId</u>, name,email,phone,address,sal
2. <u>CategoryId</u>,categoryName

3NF:
There is no transitive dependency.

1. <u>ModId</u>, name,email,phone,address,sal
2. <u>CategoryId</u>,categoryName

Create Table:

1. <u>ModId</u>,name,email,phone,address,sal
2. <u>CategoryId</u>,categoryName

Insert/Update(<u>ModId</u>, name,email,phone,address,sal,

ProductID,ProductName,Description,**CategoryId**,Buying_price,Selling_price,Quantity)
1NF:
Phone is multivalued attribute.
ModId,name,email,phone,address,sal,ProductID,ProductName,Description,**CategoryId**,Buying_price,Selling_price,Quantity
2NF:

1. ModId,name,email,phone,address,sal
2. ProductID,ProductName,Description,**CategoryId**,Buying_price,Selling_price,Quantity

3NF:
There is no transitive dependency.

1. ModId,name,email,phone,address,sal
2. ProductID,ProductName,Description,**CategoryId**,Buying_price,Selling_price,Quantity

Create Table:

1. ModId,name,email,phone,address,sal
2. ProductID,ProductName,Description,**CategoryId**,Buying_price,Selling_price,Quantity

has(CategoryId,CategoryName,ProductId,ProductName,Description,**CategoryId**,Buying_price, Selling_price,Quantity)
1NF:
NO multivalued attribute.
CategoryId,CategoryName,ProductId,ProductName,Description,**CategoryId**,Buying_price,Selling_price,Quantity
2NF:

1. CategoryId,CategoryName
2. ProductId,ProductName,Description,**CategoryId**,Buying_price,Selling_price,Quantity

3NF:
There is no transitive dependency.

1. CategoryId,CategoryName
2. ProductId,ProductName,Description,**CategoryId**,Buying_price,Selling_price,Quantity

Create Table;

1. CategoryId,CategoryName
2. ProductId,ProductName,Description,**CategoryId**,Buying_price,Selling_price,Quantity

Buy(CusId,name,email,phone,address , ProductId,ProductName,Description,**CategoryId**,Buying_price,Selling_price,Quantity)
1NF:
PHONE IS MULTIVALUED ATTRIBUTE.
CusId,name,email,phone,address , ProductId,ProductName,Description,**CategoryId**,Buying_price,Selling_price,Quantity
2NF:

1. CusId,name,email,phone,address
2. ProductId,ProductName,Description,**CategoryId**,Buying_price,Selling_price,Quantity

3NF:

There is no transitive dependency.

1. CusId,name,email,phone,address
2. ProductId,ProductName,Description,**CategoryId**,Buying_price,Selling_price,Quantity

CREATE TABLE:

1. CusId,name,email,phone,address
2. ProductId,ProductName,Description,**CategoryId**,Buying_price,Selling_price,Quantity

Canplace (CusId,name,email,phone,address ,
orderId,date,status,address,delmanID,amount,_profit,**CusId** payment_type,**CartId**)

1NF:

Phone is multivalued attribute.

CusId,name,email,phone,address ,orderId,date,status,address,delmanID,amount, profit, **CusId** payment_type,**CartId**

2NF:

1. CusId,name,email,phone,address
2. orderId,date,status,address,delmanID, profit,amount, **CusId** payment_type,**CartId**

3NF:

There is no transitive dependency.

1. CusId,name,email,phone,address
2. orderId,date,status,address,delmanID,amount, profit, **CusId** payment_type,**CartId**

Create Table:

1. CusId,name,email,phone,address
2. orderId,date,status,address,delmanID,amount, profit, **CusId** payment_type,**CartId**

Insert(Admin_Id,name,email,phone,address,sal, ModId, name,email,phone,address,sal)

1NF:

Phone is multivalued attribute.

Admin_Id,name,email,phone,address,sal, ModId, name,email,phone,address,sal

2NF:

1. Admin_Id,name,email,phone,address,sal
2. ModId, name,email,phone,address,sal

3NF:

There is no transitive dependency.

1. Admin_Id,name,email,phone,address,sal
2. ModId, name,email,phone,address,sal

Create Table:

1. Admin_Id,name,email,phone,address,sal
2. ModId, name,email,phone,address,sal

Contain(Cart_id,**ProductId**,**CusId**,profit,
ProductId,ProductName,Description,**CategoryId**,Buying_price,Selling_price,Quantity)

1NF:

No multivalued attribute.

1. Cart_id,**ProductId**,**CusId**,profit, ProductId,ProductName,Description,**CategoryId**,Buying_price,Selling_price,Quantity

2NF:

1. Cart_id,**ProductId**,**CusId,**profit

2. ProductId,ProductName,Description,**CategoryId**,Buying_price,Selling_price,Quantity

3NF:
There is no transitive dependency.
1. Cart_id,**ProductId**,**CusId,**profit
2. ProductId,ProductName,Description,**CategoryId**,Buying_price,Selling_price,Quantity

Create Table:
1. Cart_id,**ProductId**,**CusId,**profit
2. ProductId,ProductName,Description,**CategoryId**,Buying_price,Selling_price,Quantity

Converted(Cart_id,**ProductId**,**CusId,**profit,
orderId,date,status,address,delmanID,amount, profit, **CusId** payment_type,**CartId**)

1NF:
NO multivalued attribute.
1. Cart_id,**ProductId**,**CusId,**profit,
orderId,date,status,address,delmanID,amount, profit, profit,**CusId** payment_type,**CartId**

2NF:
1. Cart_id,**ProductId**,**CusId,**profit
2. orderId,date,status,address,delmanID,amount, profit, profit,**CusId** payment_type,**CartId**

3NF:
There is no transitive dependency.
1. Cart_id,**ProductId**,**CusId,**profit
2. orderId,date,status,address,delmanID,amount, profit, profit, profit, **CusId** payment_type, **CartId**

Create table:
1. Cart_id,**ProductId**,**CusId,**profit
2. orderId,date,status,address,delmanID,amount, profit, profit, **CusId** payment_type,**CartId**

Create (CusId,name,email,phone,address, Cart_id,**ProductId**,**CusId,**profit)
1NF:
Phone is multivalued attribute
CusId,name,email,phone,address, Cart_id,**ProductId**,**CusId,**profit
2NF:
1. CusId,name,email,phone,address
2. Cart_id,**ProductId**,**CusId,**profit

3NF:
There is no transitive dependency
1. CusId,name,email,phone,address
2. Cart_id,**ProductId**,**CusId,**profit

CREATE TABLE:
1. CusId,name,email,phone,address
2. Cart_id,**ProductId**,**CusId,**profit

Accept (ModId, name,email,phone,address,sal,_
orderId,date,status,address,delmanID,amount, profit, **CusId** payment_type,**CartId)**
1NF:
Phone is multivalued attribute
ModId, name,email,phone,address,sal, orderId,date,status,address,delmanID,amount, profit, **Cus Id** payment_type,**CartId**
2NF:

1. ModId, name,email,phone,address,sal
2. orderId,date,status,address,delmanID,amount, profit, **CusId** payment_type,**CartId**

3NF:
There is no transitive dependency

1. ModId, name,email,phone,address,sal
2. orderId,date,status,address,delmanID,amount, profit, **CusId** payment_type,**CartId**

CREATE TABLE:

1. ModId, name,email,phone,address,sal
2. orderId,date,status,address,delmanID,amount, profit, **CusId** payment_type,**CartId**

TEMPORARY TABLES:

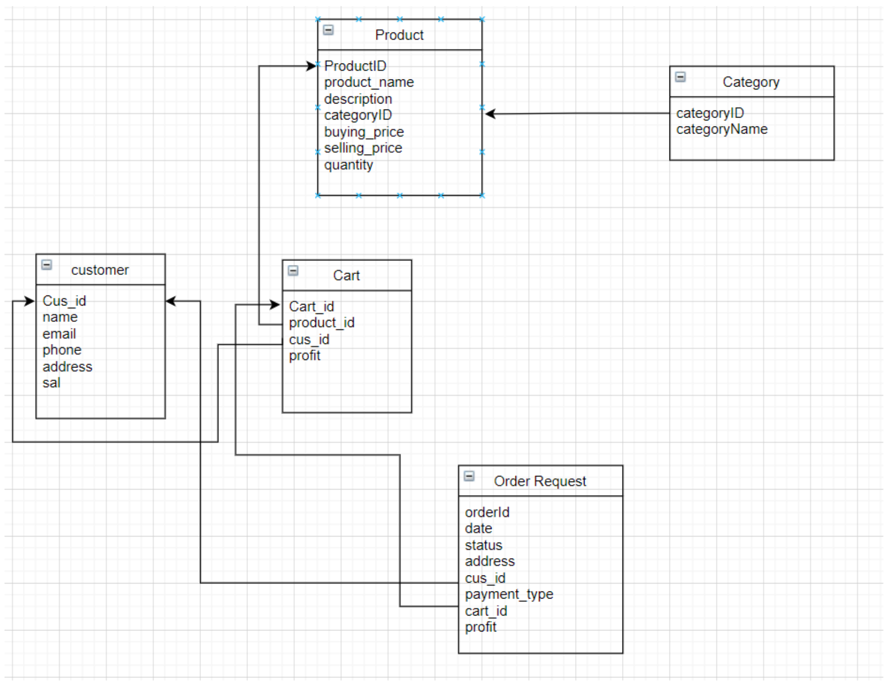
1. ModId,name,email,phone,address,sal
2. CategoryId,categoryName
3. ~~ModId,name,email,phone,address,sal~~
4. ProductID,ProductName,Description,**CategoryId**,Buying_price,Selling_price,Quantity
5. ~~CategoryId,CategoryName~~
6. ~~ProductId,ProductName,Description,**CategoryId**,Buying_price,Selling_price,Quantity~~
7. CusId,name,email,phone,address
8. ~~ProductId,ProductName,Description,**CategoryId**,Buying_price,Selling_price,Quantity~~
9. ~~CusId,name,email,phone,address~~
10. orderId,date,status,address,delmanID,amount, profit, **CusId** payment_type,**CartId**
11. Admin_Id,name,email,phone,address,sal
12. ~~ModId, name,email,phone,address,sal~~
13. Cart_id,**ProductId**,**CusId,**profit
14. ~~ProductId,ProductName,Description,**CategoryId**,Buying_price,Selling_price,Quantity~~
15. ~~Cart_id,**ProductId**,**CusId,**profit~~
16. ~~orderId,date,status,address,delmanID,amount, profit,~~
~~profit, **CusId** payment_type,**CartId**~~
17. ~~CusId,name,email,phone,address~~
18. ~~Cart_id,**ProductId**,**CusId,**profit~~
19. ~~ModId, name,email,phone,address,sal~~
20. ~~orderId,date,status,address,delmanID,amount, profit, **CusId** payment_type,**CartId**~~

FINAL TABLES:

1. ModId,name,email,phone,address,sal
2. CategoryId,categoryName
3. ProductID,ProductName,Description,**CategoryId**,Buying_price,Selling_price,Quantity
4. CusId,name,email,phone,address

5. orderId,date,status,address,delmanID,amount, profit, **CusId** payment_type,**CartId**
6. Admin_Id,name,email,phone,address,sal
7. Cart_id,**ProductId**,**CusId,**profit

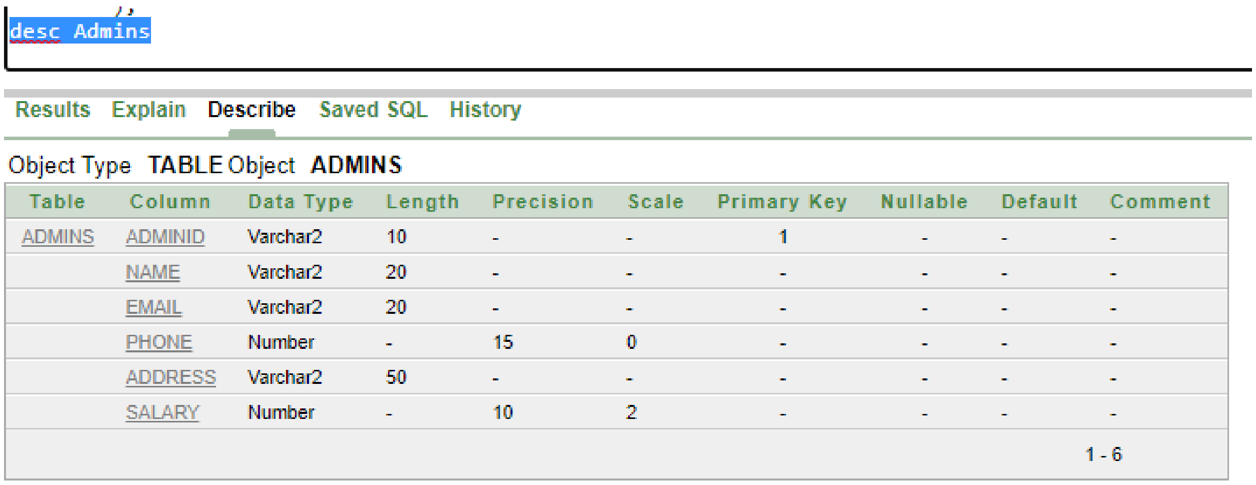
9. Schema Diagram



10.Table Creation:

Table Admin:

```
CREATE TABLE Admins

    (

    adminId varchar2(10) NOT NULL,

    name varchar2(20) NOT NULL,

    email varchar2(20) NOT NULL,
    phone NUMBER(15) NOT NULL,

    address varchar2(50) NOT NULL,

    salary NUMBER(10,2) NOT NULL,

    PRIMARY KEY (AdminId)

    );
```



```
desc Admins
```

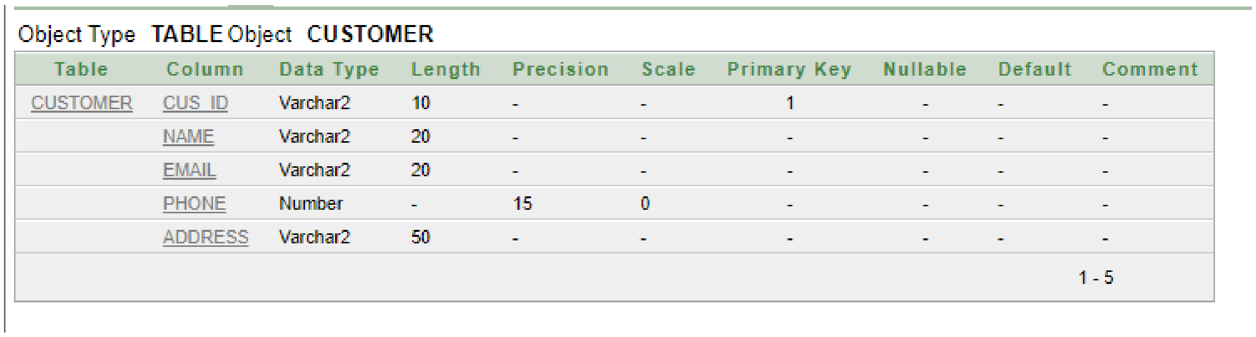
Object Type TABLE Object ADMINS

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ADMINS	ADMINID	Varchar2	10	-	-	1	-	-	-
	NAME	Varchar2	20	-	-	-	-	-	-
	EMAIL	Varchar2	20	-	-	-	-	-	-
	PHONE	Number	-	15	0	-	-	-	-
	ADDRESS	Varchar2	50	-	-	-	-	-	-
	SALARY	Number	-	10	2	-	-	-	-

1 - 6

Table Customer:

```
CREATE TABLE customer

        (

        Cus_Id varchar2 (10) NOT NULL,

        name varchar2(20) NOT NULL,

        email varchar2(20) NOT NULL,

        phone NUMBER(15) NOT NULL,

        address varchar2(50) NOT NULL,

        PRIMARY KEY (Cus_Id)

        );
```

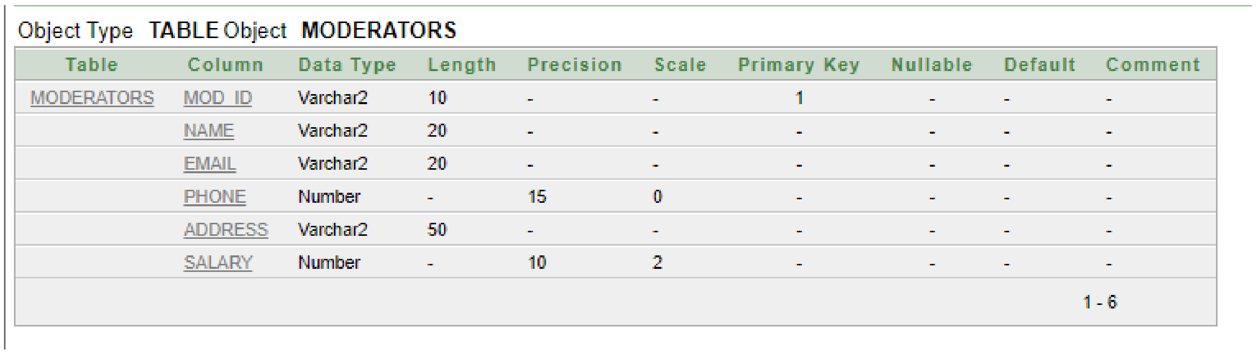


Object Type **TABLE** Object **CUSTOMER**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER	CUS_ID	Varchar2	10	-	-	1	-	-	-
	NAME	Varchar2	20	-	-	-	-	-	-
	EMAIL	Varchar2	20	-	-	-	-	-	-
	PHONE	Number	-	15	0	-	-	-	-
	ADDRESS	Varchar2	50	-	-	-	-	-	-
								1 - 5	

Table Moderator:

```
CREATE TABLE Moderators

        (

        Mod_Id varchar2(10) NOT NULL,

        name varchar2(20) NOT NULL,

        email varchar2(20) NOT NULL,

        phone NUMBER(15) NOT NULL,

        address varchar2(50) NOT NULL,

        salary NUMBER(10,2) NOT NULL,

        PRIMARY KEY (Mod_Id )

        );
```

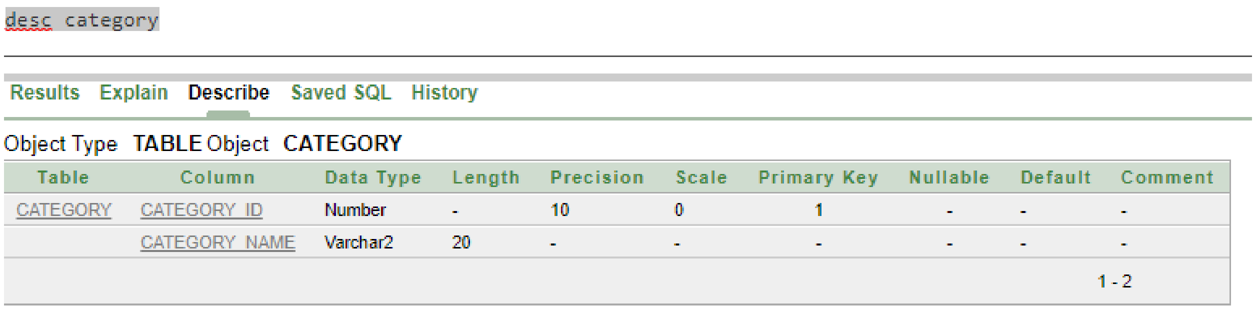


Object Type **TABLE** Object **MODERATORS**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MODERATORS	MOD_ID	Varchar2	10	-	-	1	-	-	-
	NAME	Varchar2	20	-	-	-	-	-	-
	EMAIL	Varchar2	20	-	-	-	-	-	-
	PHONE	Number	-	15	0	-	-	-	-
	ADDRESS	Varchar2	50	-	-	-	-	-	-
	SALARY	Number	-	10	2	-	-	-	-
								1 - 6	

Table Category:

```
CREATE TABLE category
        (
        Category_Id number(10) NOT NULL,
        Category_name varchar2(20) NOT NULL,
        PRIMARY KEY (Category_Id )
        );
```



```
desc category
```

Object Type TABLE Object CATEGORY

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CATEGORY	CATEGORY_ID	Number	-	10	0	1	-	-	-
	CATEGORY_NAME	Varchar2	20	-	-	-	-	-	-

1 - 2

Table Product:

```
CREATE TABLE product
        (
        Product_Id number(10) NOT NULL,
        product_name varchar2(20) NOT NULL,
        description varchar2(200) NOT NULL,
        Category_Id  number(10) NOT NULL,
```

buying_price number(10,2) NOT NULL,

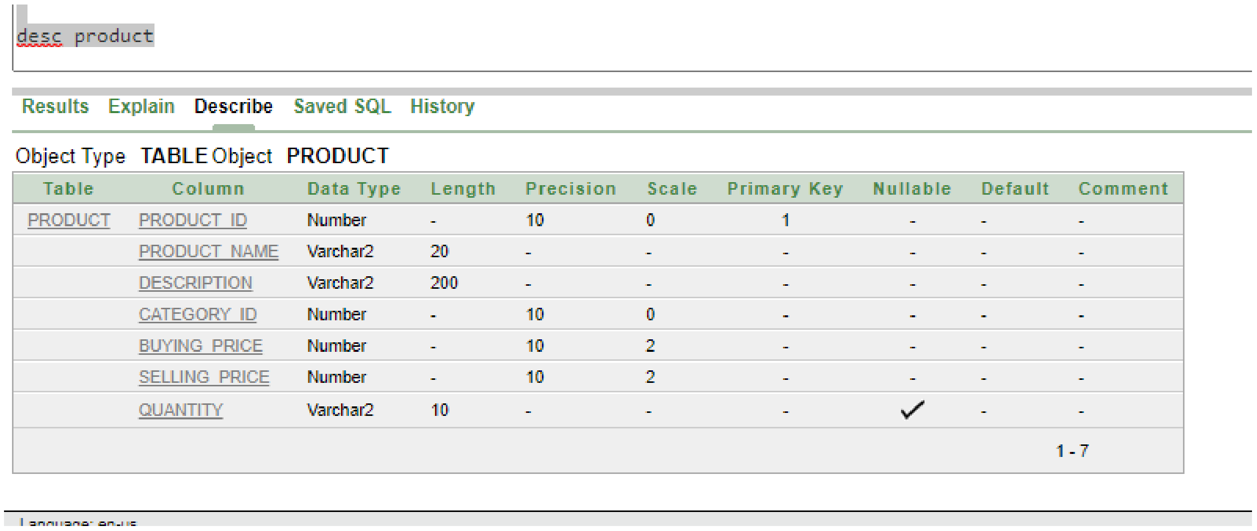
selling_price number(10,2) NOT NULL,

quantity varchar2(10),

PRIMARY KEY (Product_Id),

FOREIGN KEY (Category_Id) REFERENCES Category(Category_Id)

);



```
desc product
```

Object Type TABLE Object PRODUCT

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PRODUCT	PRODUCT_ID	Number	-	10	0	1	-	-	-
	PRODUCT_NAME	Varchar2	20	-	-	-	-	-	-
	DESCRIPTION	Varchar2	200	-	-	-	-	-	-
	CATEGORY_ID	Number	-	10	0	-	-	-	-
	BUYING_PRICE	Number	-	10	2	-	-	-	-
	SELLING_PRICE	Number	-	10	2	-	-	-	-
	QUANTITY	Varchar2	10	-	-	-	✓	-	-

1 - 7

Language: en-us

Table Cart:

CREATE TABLE cart

(

cartId number(10) NOT NULL,

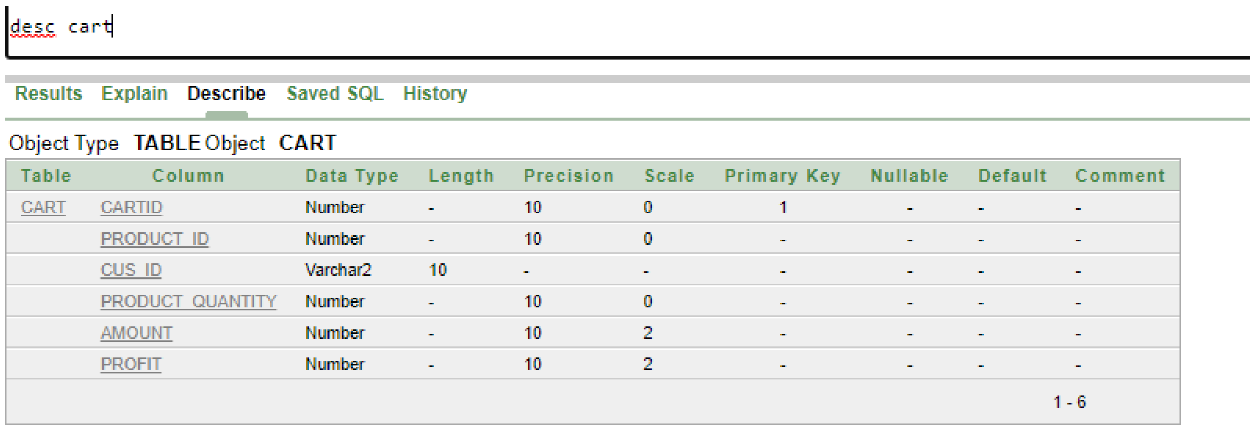
Product_Id number(10) NOT NULL,

Cus_Id varchar2(10) NOT NULL,

product_quantity number(10) NOT NULL,

amount number(10,2) NOT NULL,

```
profit number(10,2) NOT NULL,

PRIMARY KEY (cartId ),

FOREIGN KEY (Cus_Id ) REFERENCES customer(Cus_Id ),

FOREIGN KEY (Product_Id ) REFERENCES product(product_Id )

);
```



```
desc cart
```

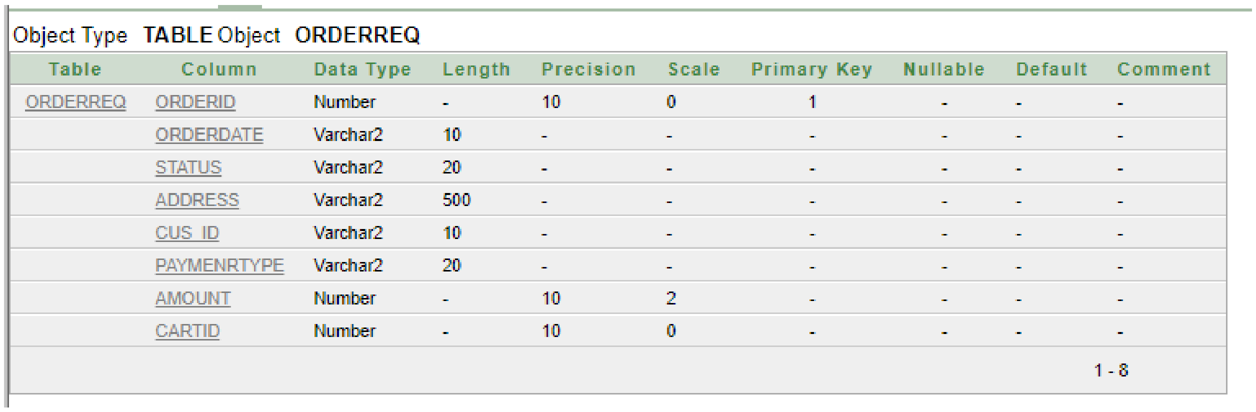
Object Type TABLE Object CART

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CART	CARTID	Number	-	10	0	1	-	-	-
	PRODUCT_ID	Number	-	10	0	-	-	-	-
	CUS_ID	Varchar2	10	-	-	-	-	-	-
	PRODUCT_QUANTITY	Number	-	10	0	-	-	-	-
	AMOUNT	Number	-	10	2	-	-	-	-
	PROFIT	Number	-	10	2	-	-	-	-

1 - 6

Table Order Request:

```
CREATE TABLE OrderReq

(

orderId number(10) NOT NULL,

orderDate varchar2(10) NOT NULL,

Status varchar2(20) NOT NULL,

Address varchar2(500) NOT NULL,
Cus_Id varchar2(10) NOT NULL,

PaymenrType varchar2(20) NOT NULL,

amount number(10,2) NOT NULL,

cartId number(10) NOT NULL,

PRIMARY KEY (OrderId),

FOREIGN KEY (Cus_Id ) REFERENCES customer(Cus_Id ),

FOREIGN KEY (cartId) REFERENCES cart(cartId ));
```



Object Type TABLE Object ORDERREQ

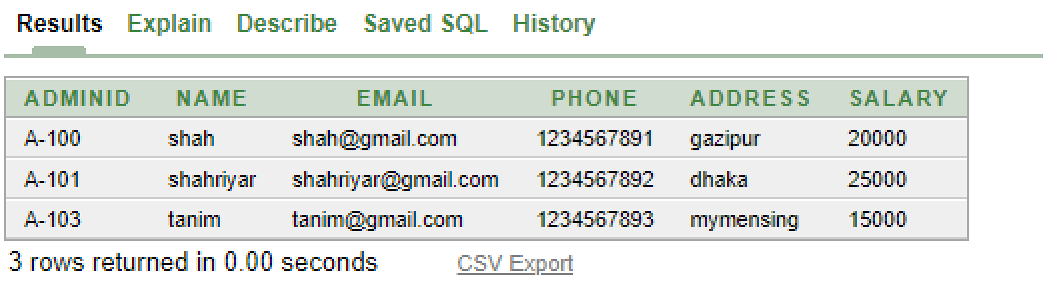
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ORDERREQ	ORDERID	Number	-	10	0	1	-	-	-
	ORDERDATE	Varchar2	10	-	-	-	-	-	-
	STATUS	Varchar2	20	-	-	-	-	-	-
	ADDRESS	Varchar2	500	-	-	-	-	-	-
	CUS_ID	Varchar2	10	-	-	-	-	-	-
	PAYMENRTYPE	Varchar2	20	-	-	-	-	-	-
	AMOUNT	Number	-	10	2	-	-	-	-
	CARTID	Number	-	10	0	-	-	-	-

1 - 8

11.Data Insertion :

Data Insertion:

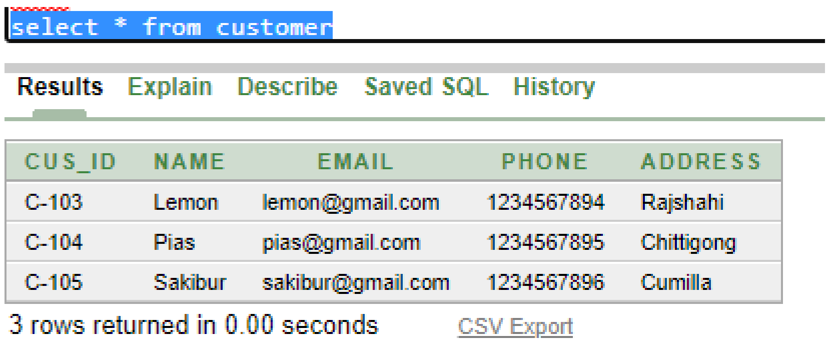
Admin Table:


Results Explain Describe Saved SQL History

ADMINID	NAME	EMAIL	PHONE	ADDRESS	SALARY
A-100	shah	shah@gmail.com	1234567891	gazipur	20000
A-101	shahriyar	shahriyar@gmail.com	1234567892	dhaka	25000
A-103	tanim	tanim@gmail.com	1234567893	mymensing	15000

3 rows returned in 0.00 seconds CSV Export

Customer Table:

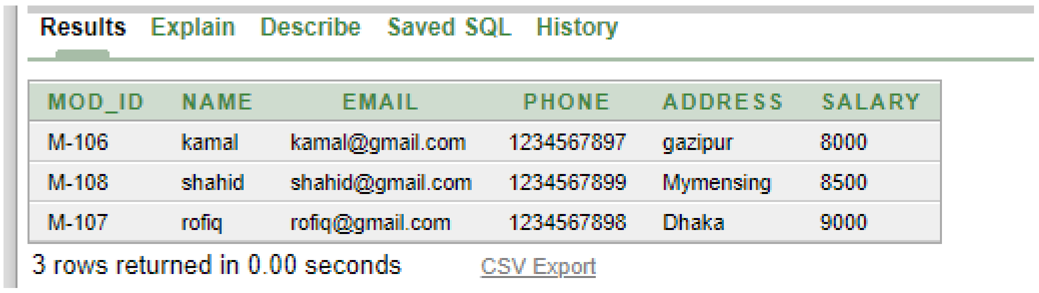

```
select * from customer
```

Results Explain Describe Saved SQL History

CUS_ID	NAME	EMAIL	PHONE	ADDRESS
C-103	Lemon	lemon@gmail.com	1234567894	Rajshahi
C-104	Pias	pias@gmail.com	1234567895	Chittigong
C-105	Sakibur	sakibur@gmail.com	1234567896	Cumilla

3 rows returned in 0.00 seconds CSV Export

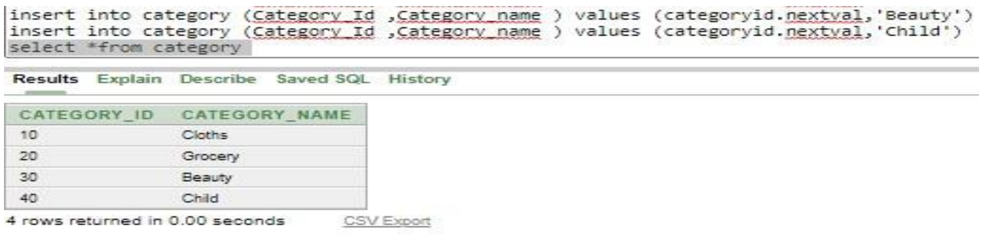
Moderator Table:



MOD_ID	NAME	EMAIL	PHONE	ADDRESS	SALARY
M-106	kamal	kamal@gmail.com	1234567897	gazipur	8000
M-108	shahid	shahid@gmail.com	1234567899	Mymensing	8500
M-107	rofiq	rofiq@gmail.com	1234567898	Dhaka	9000

3 rows returned in 0.00 seconds CSV Export

Category Table:

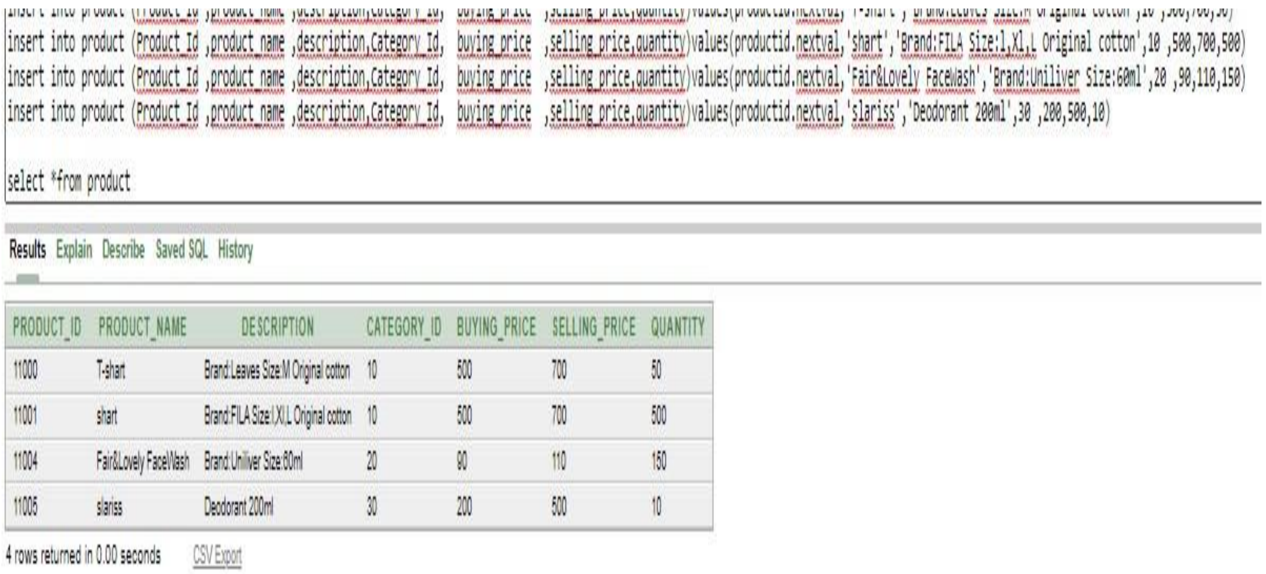


```
insert into category (Category_Id ,Category_name ) values (categoryid.nextval,'Beauty')
insert into category (Category_Id ,Category_name ) values (categoryid.nextval,'Child')
select *from category
```

CATEGORY_ID	CATEGORY_NAME
10	Cloths
20	Grocery
30	Beauty
40	Child

4 rows returned in 0.00 seconds CSV Export

Product Table:

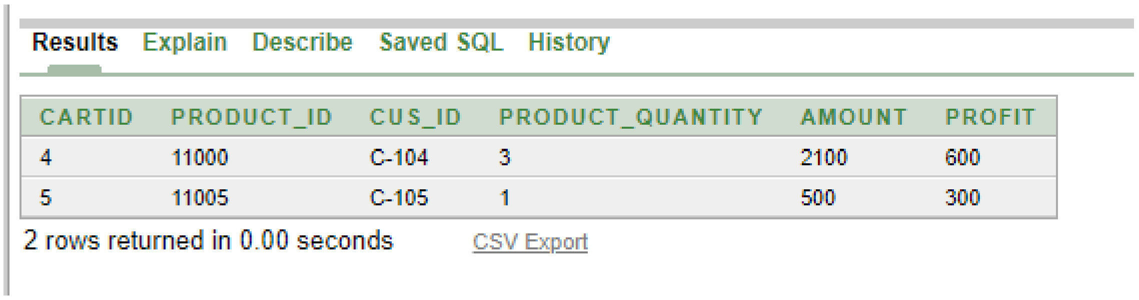


```
[illegible]
insert into product (Product_Id ,product_name ,description,Category_Id,  buying_price  ,selling_price,quantity)values(productid.nextval,'shart','Brand:FILA Size:l,Xl,L Original cotton',10 ,500,700,500)
insert into product (Product_Id ,product_name ,description,Category_Id,  buying_price  ,selling_price,quantity)values(productid.nextval,'Fair&Lovely FaceWash','Brand:Uniliver Size:60ml',20 ,90,110,150)
insert into product (Product_Id ,product_name ,description,Category_Id,  buying_price  ,selling_price,quantity)values(productid.nextval,'slariss','Deodorant 200ml',30 ,200,500,10)

select *from product
```

PRODUCT_ID	PRODUCT_NAME	DESCRIPTION	CATEGORY_ID	BUYING_PRICE	SELLING_PRICE	QUANTITY
11000	T-shart	Brand:Leaves Size:M Original cotton	10	500	700	50
11001	shart	Brand:FILA Size:l,Xl,L Original cotton	10	500	700	500
11004	Fair&Lovely FaceWash	Brand:Uniliver Size:60ml	20	90	110	150
11005	slariss	Deodorant 200ml	30	200	500	10

4 rows returned in 0.00 seconds CSV Export

Cart Table:

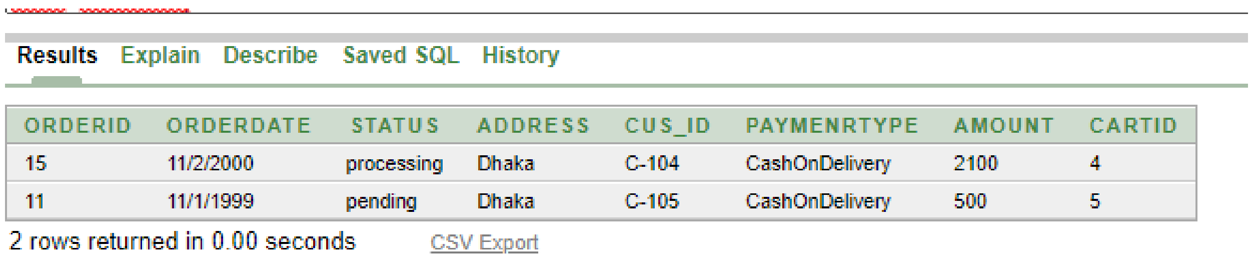


CARTID	PRODUCT_ID	CUS_ID	PRODUCT_QUANTITY	AMOUNT	PROFIT
4	11000	C-104	3	2100	600
5	11005	C-105	1	500	300

2 rows returned in 0.00 seconds

CSV Export

Order Req Table:



ORDERID	ORDERDATE	STATUS	ADDRESS	CUS_ID	PAYMENRTYPE	AMOUNT	CARTID
15	11/2/2000	processing	Dhaka	C-104	CashOnDelivery	2100	4
11	11/1/1999	pending	Dhaka	C-105	CashOnDelivery	500	5

2 rows returned in 0.00 seconds

CSV Export

Logs:

```
insert into
OrderReq(ORDERID,ORDERDATE,STATUS,ADDRESS,CUS_ID,PAYMENRTYPE,
AMOUNT,CARTID) values(15,'11/2/2000','processing','Dhaka','C-
104','CashOnDelivery',2100,4)

insert into
OrderReq(ORDERID,ORDERDATE,STATUS,ADDRESS,CUS_ID,PAYMENRTYPE,
AMOUNT,CARTID) values(11,'11/1/1999','pending','Dhaka','C-
105','CashOnDelivery',500,5)

select * from product;



insert into cart (cartId,Product_Id ,Cus_Id,product_quantity ,amount, profit )
values(cartid.nextval,11000,'C-104',3,(3* 700 ),3* 200 )
```

```
insert into cart (cartId,Product_Id ,Cus_Id,product_quantity ,amount, profit )
values(cartid.nextval,11003,'C-105',1,(1* 500),1* 300)

desc category;

insert into product (product_id,product_name,description,category_id,
buying_price,selling_price,quantity)values(productid.nextval,'T-shirt','Brand:Leaves
size:M Original cotton',10,500,700,50)

insert into product (product_id,product_name,description,category_id,
buying_price,selling_price,quantity)values(productid.nextval,'shirt','Brand:FLA size:XL
Original cotton',10,500,700,500)

insert into product (product_id,product_name,description,category_id,
buying_price,selling_price,quantity)values(productid.nextval,'Fair&lovely
Facewash','Brand:Uniliver size:60ml Original cotton',20,90,110,150)

insert into product (product_id,product_name,description,category_id,
buying_price,selling_price,quantity)values(productid.nextval,'slariss','Dedorant
200ml',30,200,500,10)


CREATE SEQUENCE cartid

START WITH 4

INCREMENT BY 1;


insert into category (category_id,category_name) values(categoryid.nextval,'clothes')

insert into category (category_id,category_name) values(categoryid.nextval,'Grocery')

insert into category (category_id,category_name) values(categoryid.nextval,'Beauty')

insert into category (category_id,category_name) values(categoryid.nextval,'Child')
```

insert into customer(Cus_Id ,name,email,phone,address)values('C-103','Lemon','lemon@gmail.com',01234567894,'Rajshahi')

insert into customer(Cus_Id ,name,email,phone,address)values('C-104','Pias','pias@gmail.com',01234567895,'Chittigong')

insert into customer(Cus_Id ,name,email,phone,address)values('C-105','Sakibur','sakibur@gmail.com',01234567896,'Cumilla')

insert into Moderators(Mod_Id ,name,email,phone,address,salary)values('M-106','kamal','kamal@gmail.com',01234567897,'gazipur',8000.00)

insert into Moderators(Mod_Id ,name,email,phone,address,salary)values('M-108','shahid','shahid@gmail.com',01234567899,'Mymensing',8500.00)

insert into Moderators(Mod_Id ,name,email,phone,address,salary)values('M-107','rofiq','rofiq@gmail.com',01234567898,'Dhaka',9000.00)

insert into admins(adminid,name,email,phone,address,salary)values('A-100','shah','shah@gmail.com',1234567891,'gazipur',20000)

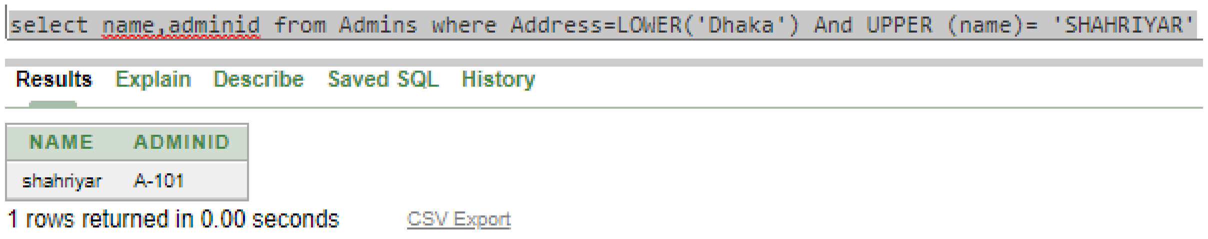
insert into admins(adminid,name,email,phone,address,salary)values('A-101','shahriyar','shahriyar@gmail.com',1234567892,'dhaka',25000)

insert into admins(adminid,name,email,phone,address,salary)values('A-103','tanim','tanim@gmail.com',1234567893,'mymensing',15000)

12.Query Writing

12.1. Single Row Function

- · Display the name of Admin who lives in Dhaka and name is SHAHRIYAR
  Ans: select name,adminid from Admins where Address=LOWER('Dhaka') And UPPER (name)= 'SHAHRIYAR'



```
select name,adminid from Admins where Address=LOWER('Dhaka') And UPPER (name)= 'SHAHRIYAR'
```

Results Explain Describe Saved SQL History

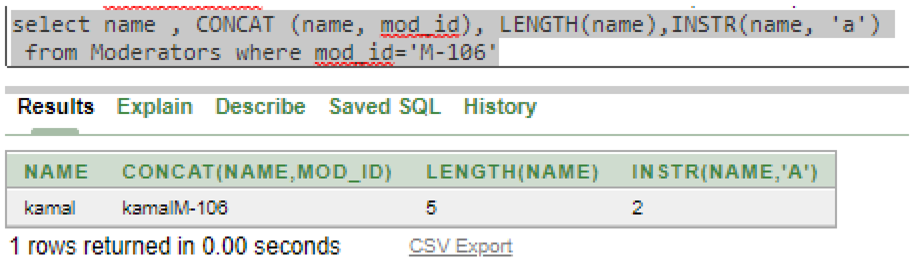
NAME	ADMINID
shahriyar	A-101

1 rows returned in 0.00 seconds CSV Export

- · Display the name of the moderator whose Mod_id 106 and connate name and id also show how many a in their name?

  Ans: select name , CONCAT (name, mod_id), LENGTH(name),INSTR(name, 'a')

  from Moderators where mod_id='M-106'



```
select name , CONCAT (name, mod_id), LENGTH(name),INSTR(name, 'a')
 from Moderators where mod_id='M-106'
```

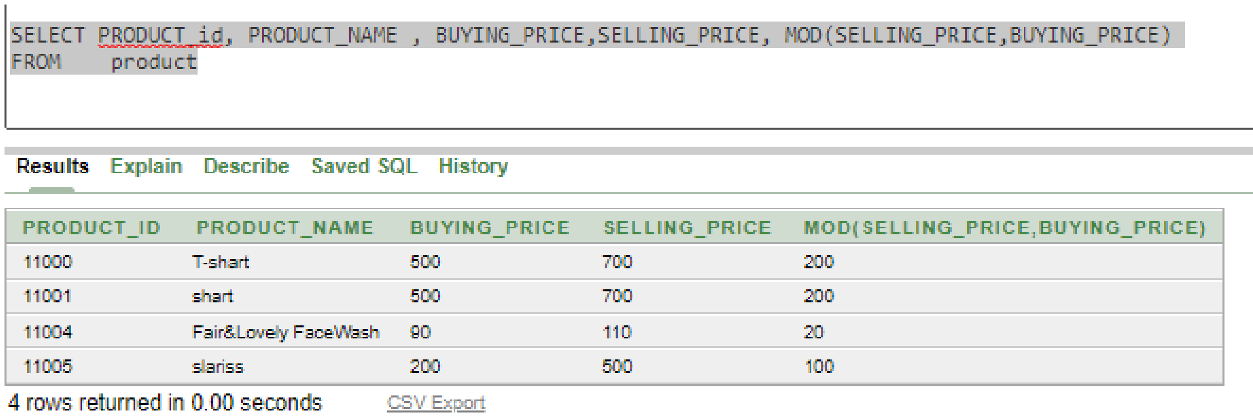
Results Explain Describe Saved SQL History

NAME	CONCAT(NAME,MOD_ID)	LENGTH(NAME)	INSTR(NAME,'A')
kamal	kamalM-106	5	2

1 rows returned in 0.00 seconds CSV Export

- Calculate the remainder of the ratio of selling price to buying price for all product
  .

Ans:



```
SELECT PRODUCT_id, PRODUCT_NAME , BUYING_PRICE,SELLING_PRICE, MOD(SELLING_PRICE,BUYING_PRICE)
FROM    product
```

Results Explain Describe Saved SQL History

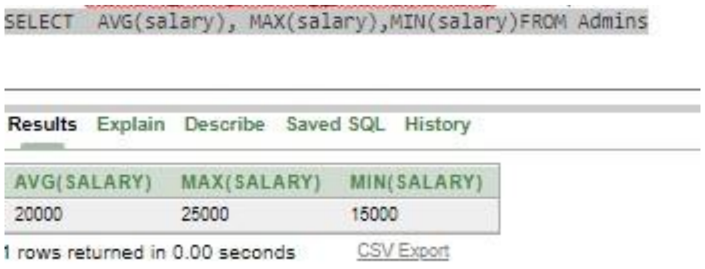
PRODUCT_ID	PRODUCT_NAME	BUYING_PRICE	SELLING_PRICE	MOD(SELLING_PRICE,BUYING_PRICE)
11000	T-shart	500	700	200
11001	shart	500	700	200
11004	Fair&Lovely FaceWash	90	110	20
11005	slariss	200	500	100

4 rows returned in 0.00 seconds   CSV Export

12.2. Group Function

- · Display average salary, minimum salary, maximum salary from admin

Ans: SELECT AVG(salary), MAX(salary),MIN(salary)FROM Admins



```
SELECT  AVG(salary), MAX(salary),MIN(salary)FROM Admins
```

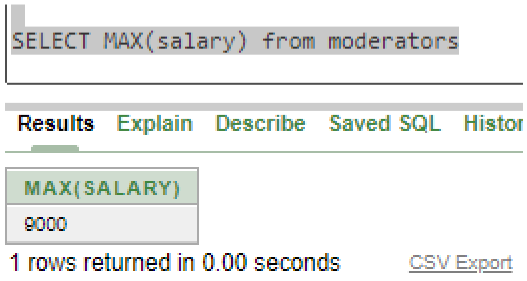
Results Explain Describe Saved SQL History

AVG(SALARY)	MAX(SALARY)	MIN(SALARY)
20000	25000	15000

1 rows returned in 0.00 seconds   CSV Export

- Display the Maximum , salary from moderators

Ans: SELECT MAX(salary) from moderators



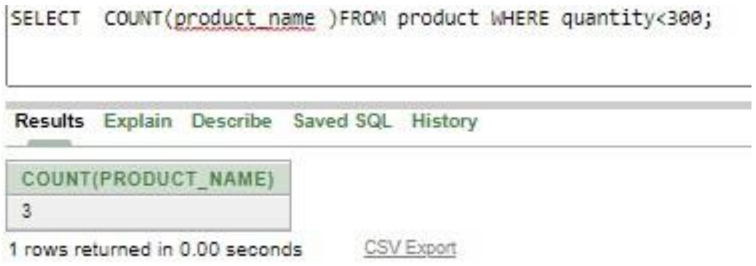
```
SELECT MAX(salary) from moderators
```

MAX(SALARY)
9000

1 rows returned in 0.00 seconds

- Display total number of the product those have quantity less than 300

  Ans:SELECT COUNT(product_name)FROM product WHERE quantity<300;



```
SELECT  COUNT(product_name )FROM product WHERE quantity<300;
```

COUNT(PRODUCT_NAME)
3

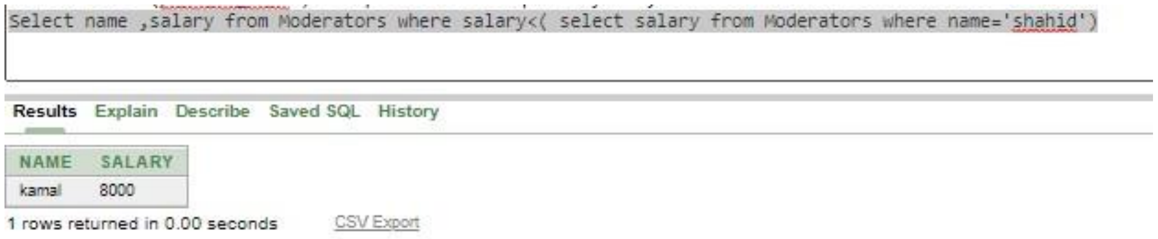
1 rows returned in 0.00 seconds

12.3.Subquery

- Display the Moderator names salary that earn a salary that is lower than the salary of all Shahid

  Ans:

  Select name ,salary from Moderators where salary<(select salary from Moderators where name='shahid')



```
Select name ,salary from Moderators where salary<( select salary from Moderators where name='shahid')
```

NAME	SALARY
kamal	8000

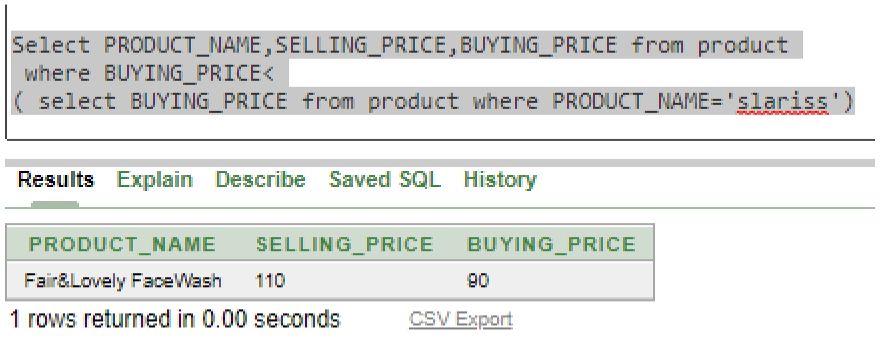
1 rows returned in 0.00 seconds

- Display the product names and selling price of which buying price is higher than slariss

  Ans:

  Select PRODUCT_NAME,SELLING_PRICE,BUYING_PRICE from product where BUYING_PRICE<

(select BUYING_PRICE from product where PRODUCT_NAME='slariss')



```
Select PRODUCT_NAME,SELLING_PRICE,BUYING_PRICE from product
 where BUYING_PRICE<
( select BUYING_PRICE from product where PRODUCT_NAME='slariss')
```

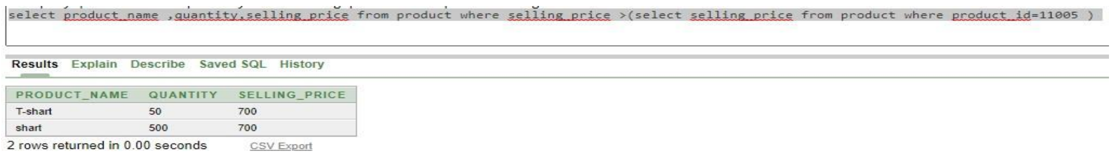
PRODUCT_NAME	SELLING_PRICE	BUYING_PRICE
Fair&Lovely FaceWash	110	90

1 rows returned in 0.00 seconds

- Display product name quantity and selling price which price is higher than 11005

Ans:

select product_name ,quantity,selling_price from product where selling_price >(select selling_price from product where product_id=11005)



```
select product_name ,quantity,selling_price from product where selling_price >(select selling_price from product where product_id=11005 )
```

PRODUCT_NAME	QUANTITY	SELLING_PRICE
T-shart	50	700
shart	500	700

2 rows returned in 0.00 seconds

12.4.Joining

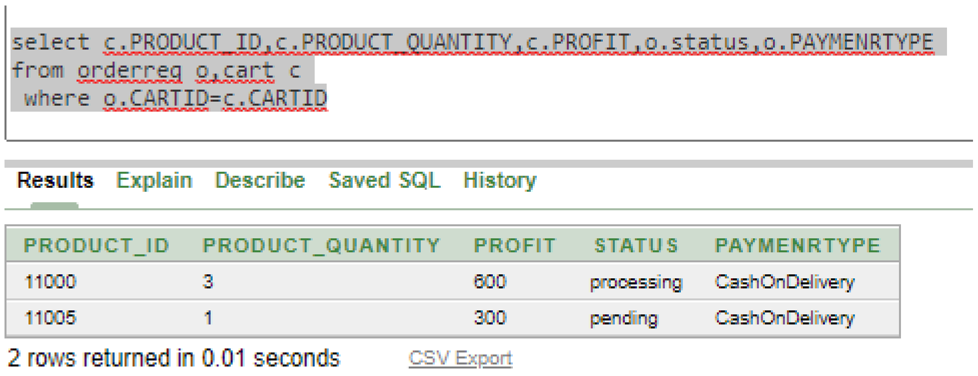
- Display the product id ,quantity,profit ,order status paymen ttype status which order id is 15

Ans:

select c.PRODUCT_ID,c.PRODUCT_QUANTITY,c.PROFIT,o.status,o.PAYMENRTYPE

from orderreq o,cart c

where o.CARTID=c.CARTID



```
select c.PRODUCT_ID,c.PRODUCT_QUANTITY,c.PROFIT,o.status,o.PAYMENRTYPE
from orderreq o,cart c
 where o.CARTID=c.CARTID
```

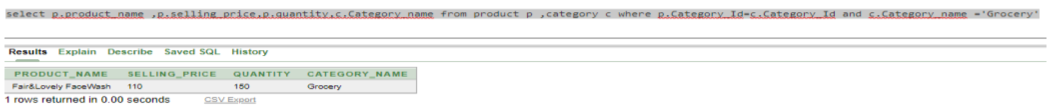
PRODUCT_ID	PRODUCT_QUANTITY	PROFIT	STATUS	PAYMENRTYPE
11000	3	600	processing	CashOnDelivery
11005	1	300	pending	CashOnDelivery

2 rows returned in 0.01 seconds

- Display the product name, price, quantity, and category name under the Grocery category

  Ans:

  select p.product_name ,p.selling_price,p.quantity,c.Category_name from product p ,category c where p.Category_Id=c.Category_Id and c.Category_name ='Grocery'



```
select p.product_name ,p.selling_price,p.quantity,c.Category_name from product p ,category c where p.Category_Id=c.Category_Id and c.Category_name ='Grocery'
```

PRODUCT_NAME	SELLING_PRICE	QUANTITY	CATEGORY_NAME
Fair&Lovely FaceWash	110	150	Grocery

1 rows returned in 0.00 seconds

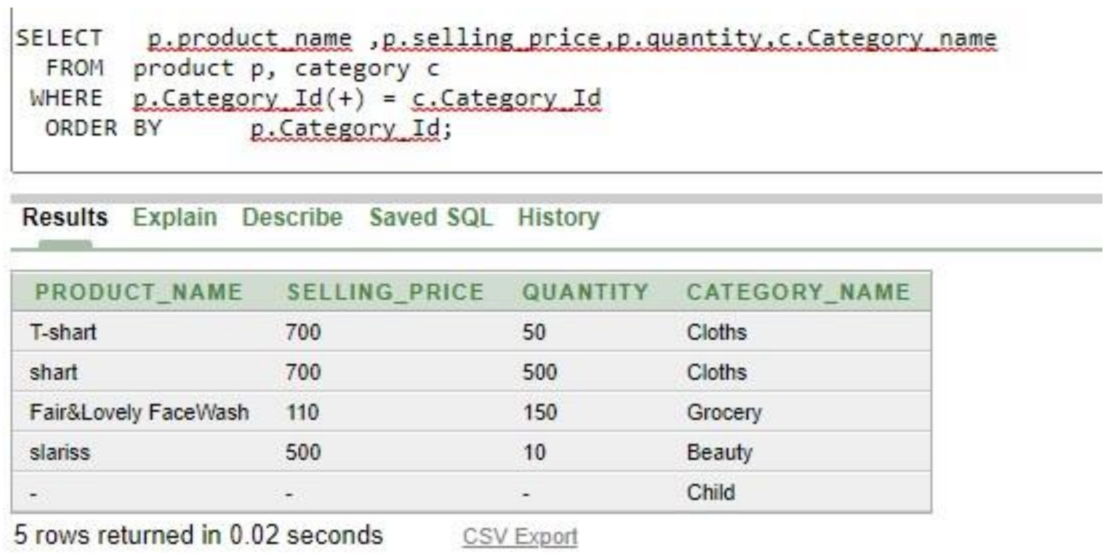
- Display each product with category name also display all category name who has no product

  Ans: SELECT p.product_name ,p.selling_price,p.quantity,c.Category_name

  FROM product p, category c

  WHERE p.Category_Id(+) = c.Category_Id

  ORDER BY p.Category_Id;



```
SELECT   p.product_name ,p.selling_price,p.quantity,c.Category_name
  FROM  product p, category c
 WHERE  p.Category_Id(+) = c.Category_Id
  ORDER BY      p.Category_Id;
```

Results Explain Describe Saved SQL History

PRODUCT_NAME	SELLING_PRICE	QUANTITY	CATEGORY_NAME
T-shart	700	50	Cloths
shart	700	500	Cloths
Fair&Lovely FaceWash	110	150	Grocery
slariss	500	10	Beauty
-	-	-	Child

5 rows returned in 0.02 seconds CSV Export

12.5. View

1. CREATE VIEW Customer_view AS
   SELECT name, email, address FROM Customer ;
   GRANT SELECT ON Customer_view TO Cus_Id;

2. CREATE VIEW payment_view AS
   SELECT cartId, Cus_Id, date, payment_type,amount FROM Order_req;
   GRANT SELECT, UPDATE ON payment_view TO ModeratorId;

3. CREATE VIEW product_view AS
SELECT ProductName, ProductId, description, selling_price   FROM Product;
GRANT SELECT, UPDATE,DELETE ON product_view TO AdminId ;

12.6. Synonym

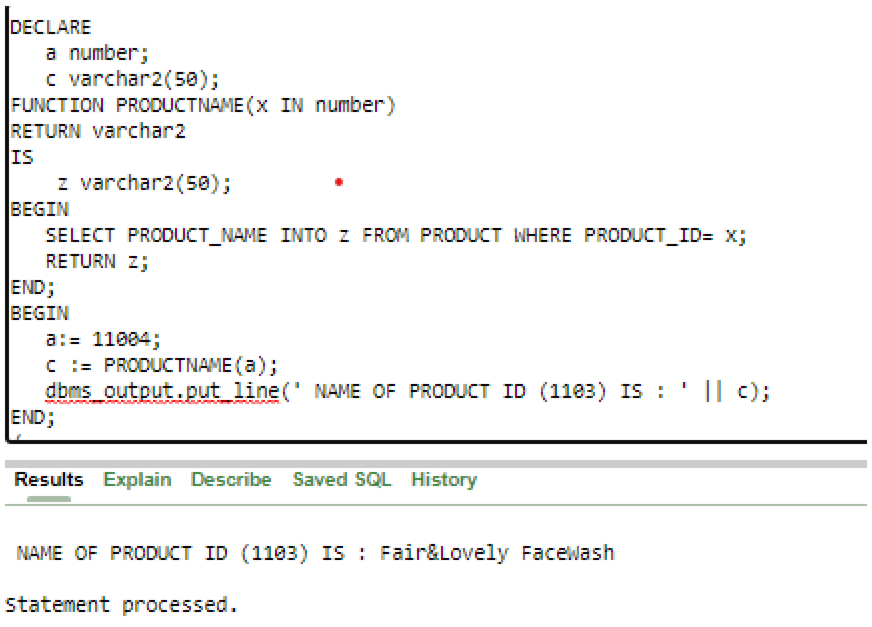
1. CREATE SYNONYM cus
   FOR customer;
2. CREATE SYNONYM mod
   FOR Moderator;
3. CREATE SYNONYM d_man
   FOR DelivaryMan;

13.PL/SQL

13.1 Function

- Display the product name of product id 11004

```
DECLARE
   a number;
   c varchar2(50);
FUNCTION PRODUCTNAME(x IN number)
RETURN varchar2
IS
    z varchar2(50);
BEGIN
   SELECT PRODUCT_NAME INTO z FROM PRODUCT WHERE PRODUCT_ID= x;
   RETURN z;
END;
BEGIN
   a:= 11004;
   c := PRODUCTNAME(a);
   dbms_output.put_line(' NAME OF PRODUCT ID (1103) IS : ' || c);
END;
```



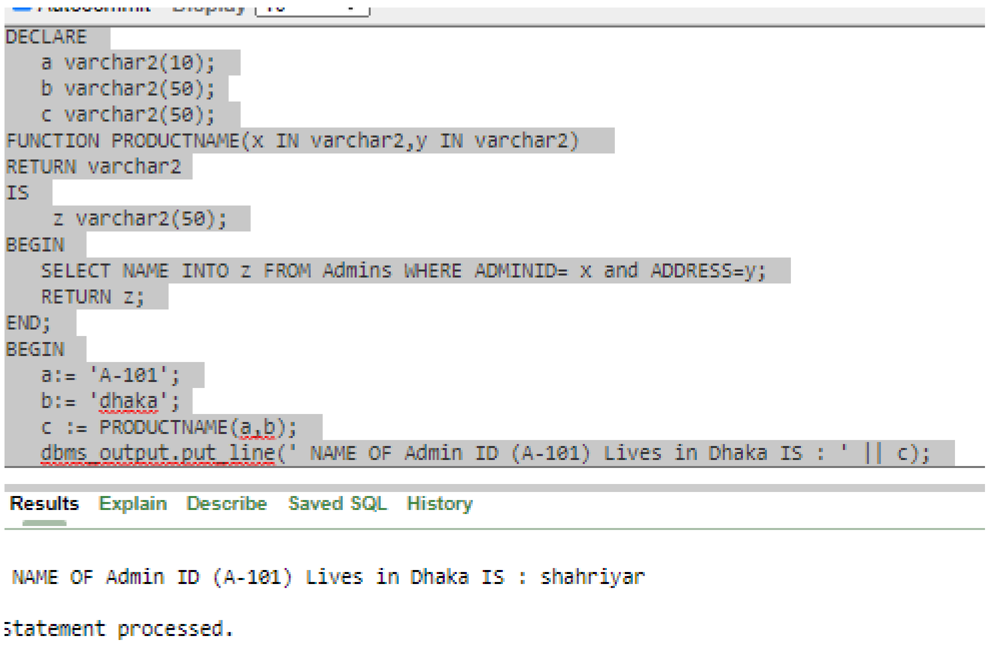
```
DECLARE
   a number;
   c varchar2(50);
FUNCTION PRODUCTNAME(x IN number)
RETURN varchar2
IS
    z varchar2(50);
BEGIN
   SELECT PRODUCT_NAME INTO z FROM PRODUCT WHERE PRODUCT_ID= x;
   RETURN z;
END;
BEGIN
   a:= 11004;
   c := PRODUCTNAME(a);
   dbms_output.put_line(' NAME OF PRODUCT ID (1103) IS : ' || c);
END;
```

```
 NAME OF PRODUCT ID (1103) IS : Fair&Lovely FaceWash

Statement processed.
```

- Display the name of Admin who lives in Dhaka and admin id is A-101

```
DECLARE

  a varchar2(10);

  b varchar2(50);

  c varchar2(50);

FUNCTION PRODUCTNAME(x IN varchar2,y IN varchar2)

RETURN varchar2

IS

   z varchar2(50);

BEGIN

  SELECT NAME INTO z FROM Admins WHERE ADMINID= x and ADDRESS=y;

  RETURN z;

END;

BEGIN

  a:= 'A-101';

  b:= 'dhaka';
```

```
    c := PRODUCTNAME(a,b);

    dbms_output.put_line(' NAME OF Admin ID (A-101) Lives in Dhaka IS : ' || c);

END;
```



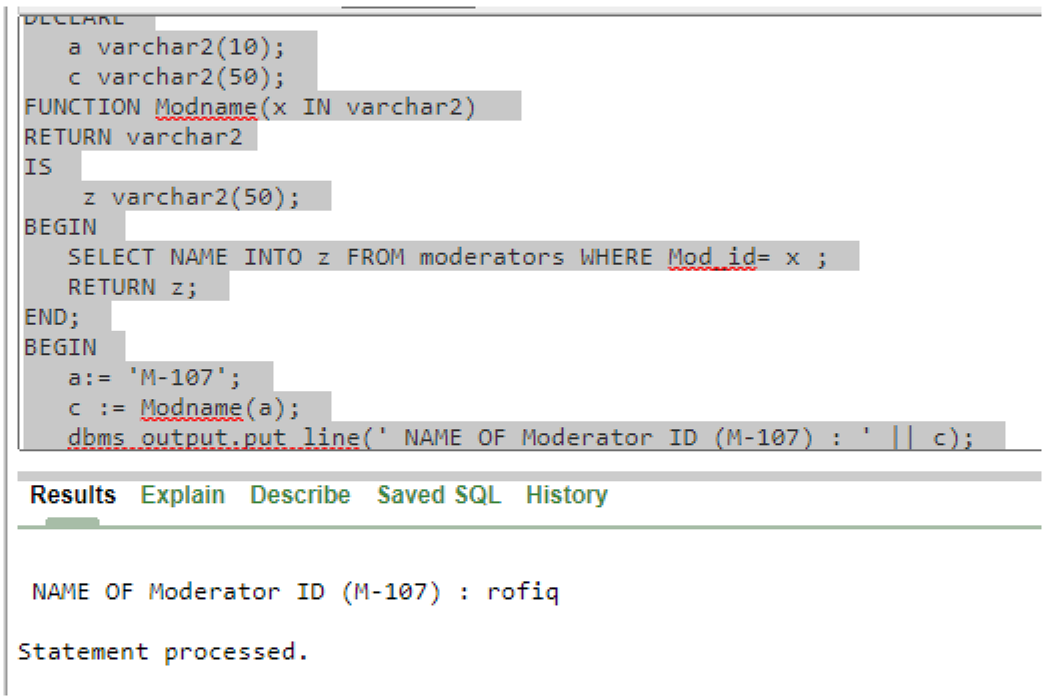
```
DECLARE
   a varchar2(10);
   b varchar2(50);
   c varchar2(50);
FUNCTION PRODUCTNAME(x IN varchar2,y IN varchar2)
RETURN varchar2
IS
    z varchar2(50);
BEGIN
   SELECT NAME INTO z FROM Admins WHERE ADMINID= x and ADDRESS=y;
   RETURN z;
END;
BEGIN
   a:= 'A-101';
   b:= 'dhaka';
   c := PRODUCTNAME(a,b);
   dbms_output.put_line(' NAME OF Admin ID (A-101) Lives in Dhaka IS : ' || c);
```

```
 NAME OF Admin ID (A-101) Lives in Dhaka IS : shahriyar

Statement processed.
```

- Display the name of the moderator whose Mod_id M-106

```
DECLARE

    a varchar2(10);

    c varchar2(50);

FUNCTION Modname(x IN varchar2)

RETURN varchar2

IS

     z varchar2(50);

BEGIN

    SELECT NAME INTO z FROM moderators WHERE Mod_id= x ;

    RETURN z;

END;

BEGIN

    a:= 'M-107';

    c := Modname(a);
```

```
  dbms_output.put_line(' NAME OF Moderator ID (M-107) : ' || c);
END;
```



```
   a varchar2(10);
   c varchar2(50);
FUNCTION Modname(x IN varchar2)
RETURN varchar2
IS
    z varchar2(50);
BEGIN
   SELECT NAME INTO z FROM moderators WHERE Mod_id= x ;
   RETURN z;
END;
BEGIN
   a:= 'M-107';
   c := Modname(a);
   dbms_output.put_line(' NAME OF Moderator ID (M-107) : ' || c);
```

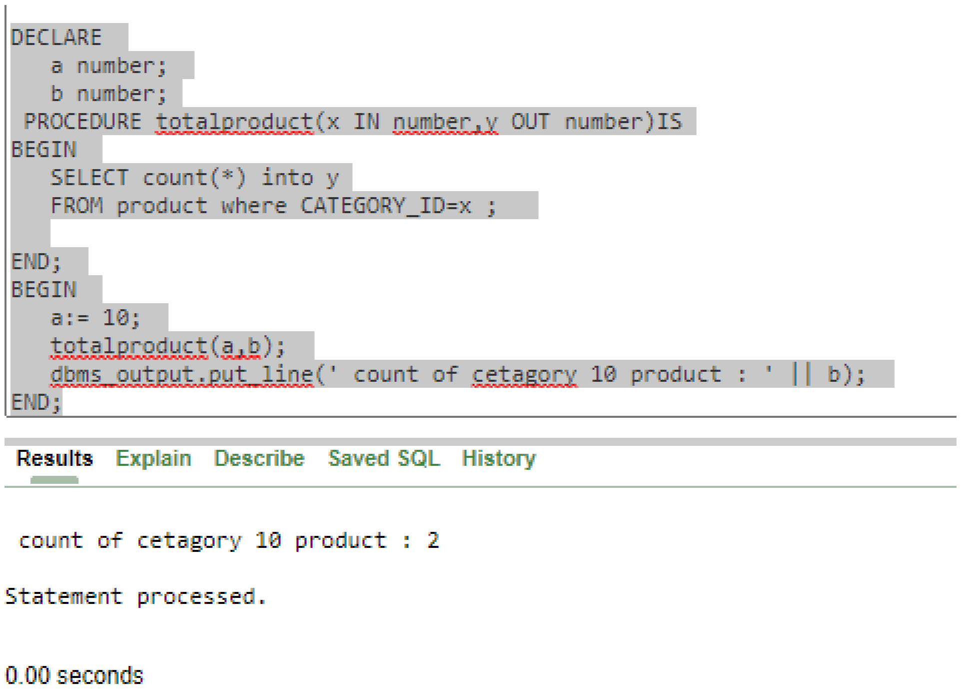
```
 NAME OF Moderator ID (M-107) : rofiq

Statement processed.
```

13.2 Procedure

- display the product count of category 10

```
DECLARE
  a number;
  b number;
 PROCEDURE totalproduct(x IN number,y OUT number)IS
BEGIN
  SELECT count(*) into y
  FROM product where CATEGORY_ID=x ;


END;
BEGIN
  a:= 10;
  totalproduct(a,b);
  dbms_output.put_line(' count of cetagory 10 product : ' || b);
```

END;



```
DECLARE
   a number;
   b number;
 PROCEDURE totalproduct(x IN number,y OUT number)IS
BEGIN
   SELECT count(*) into y
   FROM product where CATEGORY_ID=x ;

END;
BEGIN
   a:= 10;
   totalproduct(a,b);
   dbms_output.put_line(' count of cetagory 10 product : ' || b);
END;
```

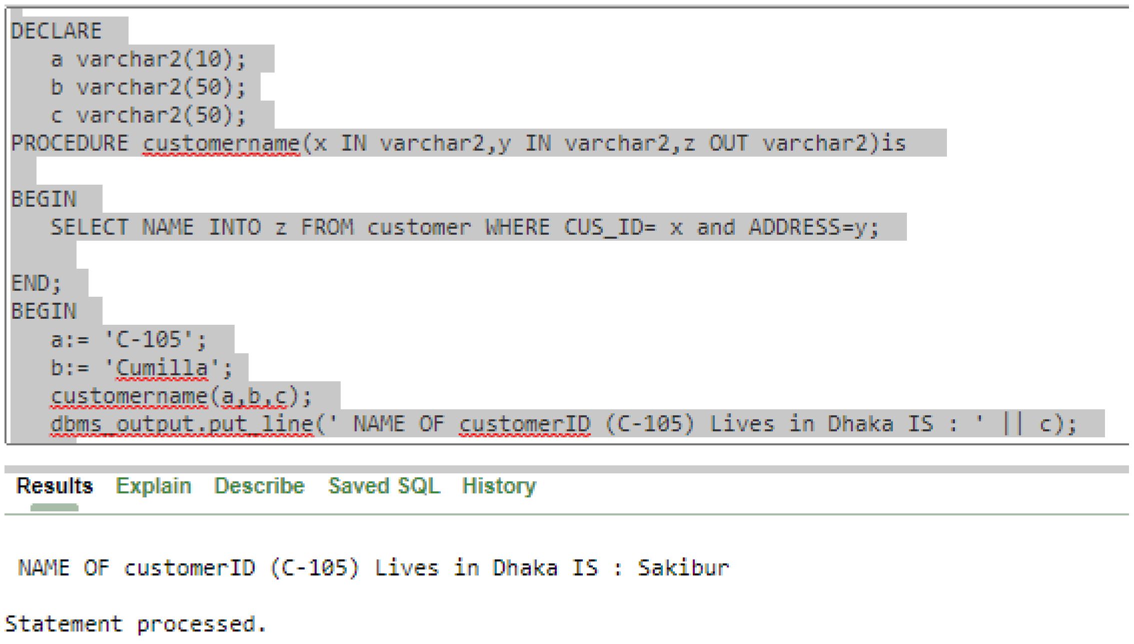
```
 count of cetagory 10 product : 2

Statement processed.

0.00 seconds
```

- Display the name of customer who lives in cumilla and customerid id is C-105

```
DECLARE

  a varchar2(10);

  b varchar2(50);

  c varchar2(50);

PROCEDURE customername(x IN varchar2,y IN varchar2,z OUT varchar2)is


BEGIN

  SELECT NAME INTO z FROM customer WHERE CUS_ID= x and ADDRESS=y;


END;

BEGIN

  a:= 'C-105';

  b:= 'Cumilla';

  customername(a,b,c);
```

dbms_output.put_line(' NAME OF customerID (C-105) Lives in Dhaka IS : ' || c);

END;

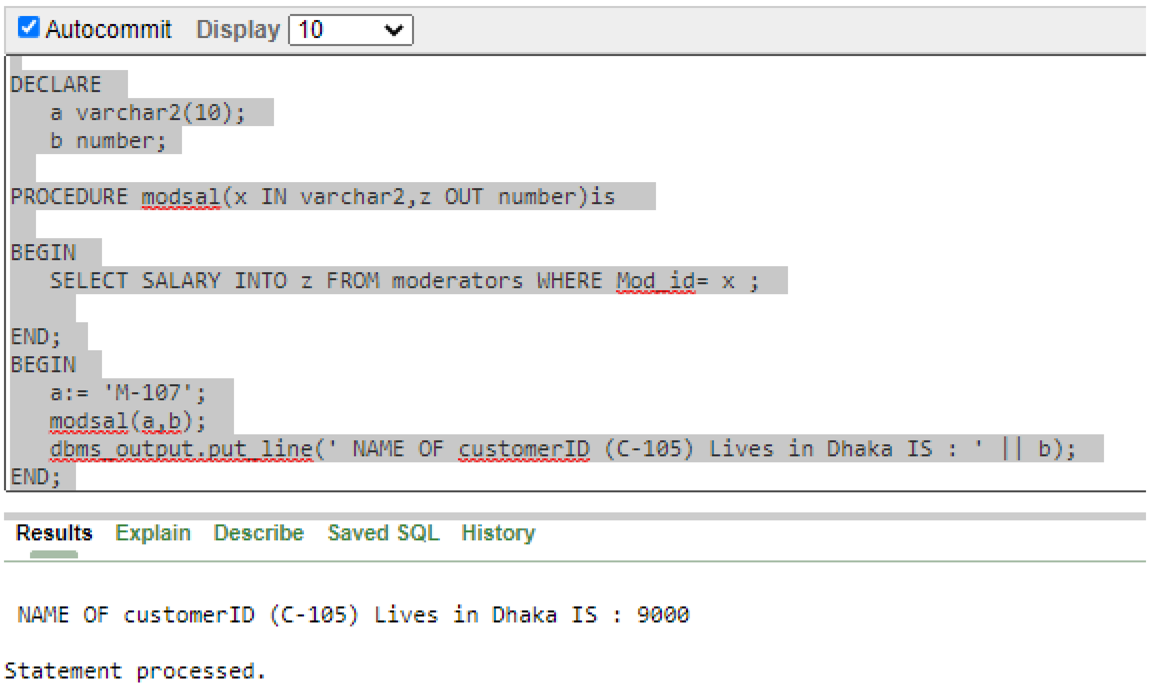


```
DECLARE
   a varchar2(10);
   b varchar2(50);
   c varchar2(50);
PROCEDURE customername(x IN varchar2,y IN varchar2,z OUT varchar2)is

BEGIN
   SELECT NAME INTO z FROM customer WHERE CUS_ID= x and ADDRESS=y;

END;
BEGIN
   a:= 'C-105';
   b:= 'Cumilla';
   customername(a,b,c);
   dbms_output.put_line(' NAME OF customerID (C-105) Lives in Dhaka IS : ' || c);
```

```
NAME OF customerID (C-105) Lives in Dhaka IS : Sakibur

Statement processed.
```

- Display the salary of the moderator whose Mod_id M-107

```
DECLARE
   a varchar2(10);
   b number;
PROCEDURE modsal(x IN varchar2,z OUT number)is
BEGIN
   SELECT SALARY INTO z FROM moderators WHERE Mod_id= x ;
END;
BEGIN
   a:= 'M-107';
   modsal(a,b);
   dbms_output.put_line(' NAME OF customerID (C-105) Lives in Dhaka IS : ' || b);
END;
```



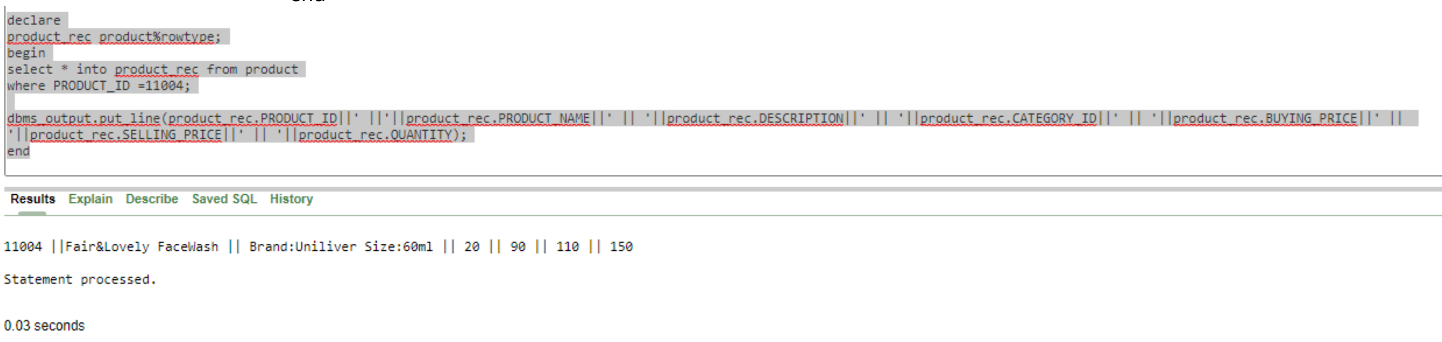
```
DECLARE
   a varchar2(10);
   b number;

PROCEDURE modsal(x IN varchar2,z OUT number)is

BEGIN
   SELECT SALARY INTO z FROM moderators WHERE Mod_id= x ;

END;
BEGIN
   a:= 'M-107';
   modsal(a,b);
   dbms_output.put_line(' NAME OF customerID (C-105) Lives in Dhaka IS : ' || b);
END;
```

```
 NAME OF customerID (C-105) Lives in Dhaka IS : 9000

Statement processed.
```

13.3 Record

- Print the product table where product id is 11004 (table based record)

```
declare

product_rec product%rowtype;

begin

select * into product_rec from product

where PRODUCT_ID =11004;

dbms_output.put_line(product_rec.PRODUCT_ID||'          ||'||product_rec.PRODUCT_NAME||'          ||
'||product_rec.DESCRIPTION||'  ||  '||product_rec.CATEGORY_ID||'  ||  '||product_rec.BUYING_PRICE||'  ||
'||product_rec.SELLING_PRICE||' || '||product_rec.QUANTITY);

end
```



```
declare
product_rec product%rowtype;
begin
select * into product_rec from product
where PRODUCT_ID =11004;

dbms_output.put_line(product_rec.PRODUCT_ID||' ||'||product_rec.PRODUCT_NAME||' || '||product_rec.DESCRIPTION||' || '||product_rec.CATEGORY_ID||' || '||product_rec.BUYING_PRICE||' ||
'||product_rec.SELLING_PRICE||' || '||product_rec.QUANTITY);
end
```

```
11004 ||Fair&Lovely FaceWash || Brand:Uniliver Size:60ml || 20 || 90 || 110 || 150

Statement processed.

0.03 seconds
```

- Create a record of an moderator whose name is rofiq (Corsure based hx)

declare

cursor c_mod is

select * from moderators where NAME='rofiq';

rec_mod moderators%rowtype;

begin

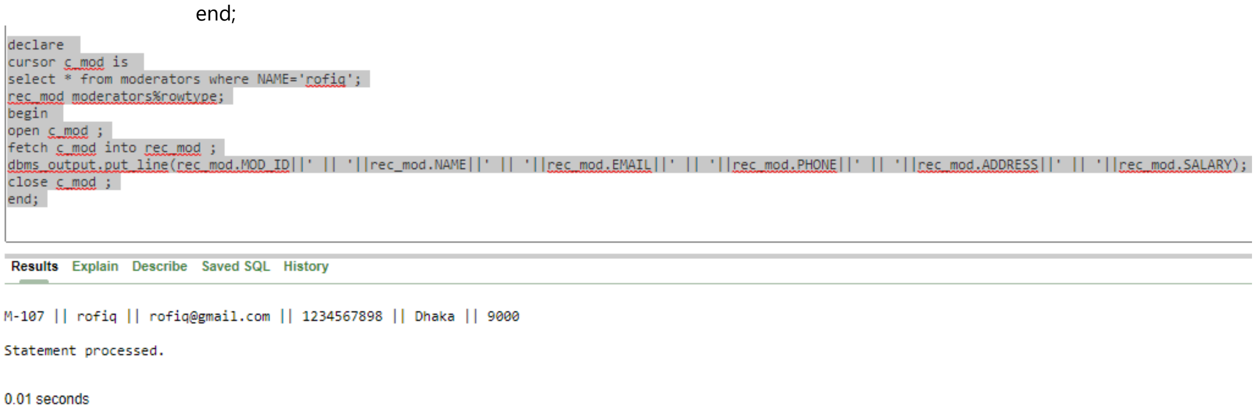
open c_mod ;

fetch c_mod into rec_mod ;

dbms_output.put_line(rec_mod.MOD_ID||' || '||rec_mod.NAME||' || '||rec_mod.EMAIL||' || '||rec_mod.PHONE||' || '||rec_mod.ADDRESS||' || '||rec_mod.SALARY);

close c_mod ;

end;



```
declare
cursor c_mod is
select * from moderators where NAME='rofiq';
rec_mod moderators%rowtype;
begin
open c_mod ;
fetch c_mod into rec_mod ;
dbms_output.put_line(rec_mod.MOD_ID||' || '||rec_mod.NAME||' || '||rec_mod.EMAIL||' || '||rec_mod.PHONE||' || '||rec_mod.ADDRESS||' || '||rec_mod.SALARY);
close c_mod ;
end;
```

```
M-107 || rofiq || rofiq@gmail.com || 1234567898 || Dhaka || 9000

Statement processed.

0.01 seconds
```

- Create a record of an moderator whose id is M-108(Corsure based record)

declare

cursor c_mod is

select * from moderators where MOD_ID='M-108';

rec_mod moderators%rowtype;

begin

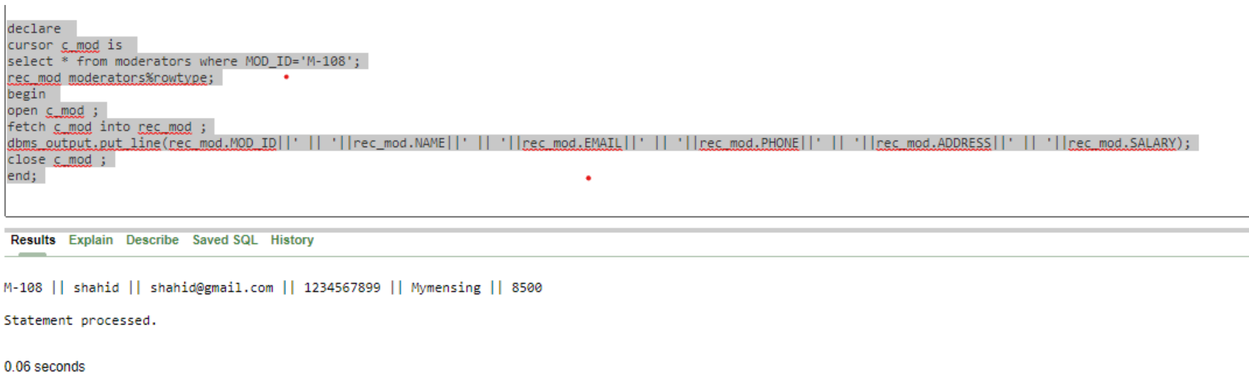
open c_mod ;

fetch c_mod into rec_mod ;

dbms_output.put_line(rec_mod.MOD_ID||' || '||rec_mod.NAME||' || '||rec_mod.EMAIL||' || '||rec_mod.PHONE||' || '||rec_mod.ADDRESS||' || '||rec_mod.SALARY);

close c_mod ;

end;



```
declare
cursor c_mod is
select * from moderators where MOD_ID='M-108';
rec_mod moderators%rowtype;
begin
open c_mod ;
fetch c_mod into rec_mod ;
dbms_output.put_line(rec_mod.MOD_ID||' || '||rec_mod.NAME||' || '||rec_mod.EMAIL||' || '||rec_mod.PHONE||' || '||rec_mod.ADDRESS||' || '||rec_mod.SALARY);
close c_mod ;
end;
```

```
M-108 || shahid || shahid@gmail.com || 1234567899 || Mymensing || 8500

Statement processed.

0.06 seconds
```

13.4 Cursor

- Update all moderators salary add 500 with the previous salary of moderator(implisit)

DECLARE

total_rows number(2);

BEGIN

UPDATE moderators

SET SALARY= SALARY+ 500;

IF sql%notfound THEN

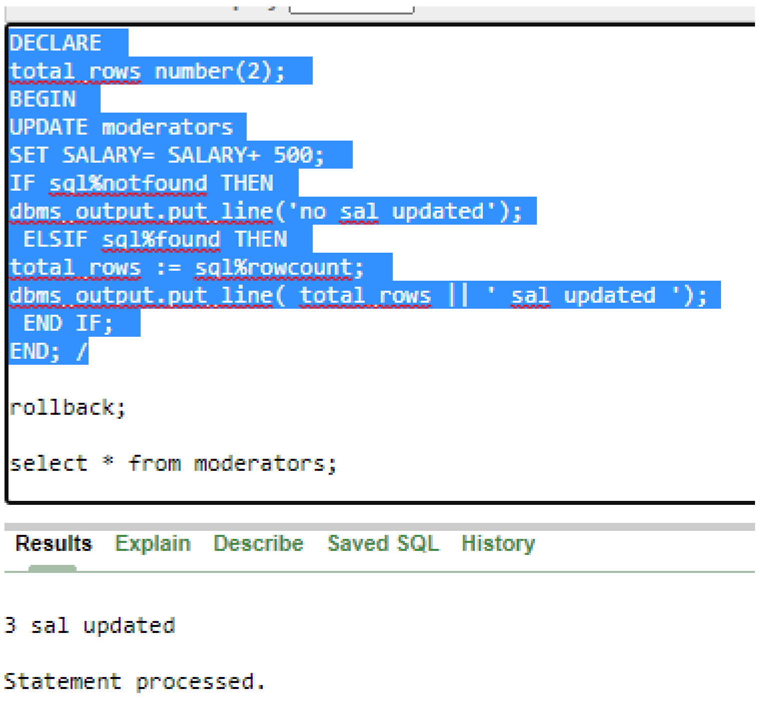
dbms_output.put_line('no sal updated');

ELSIF sql%found THEN

total_rows := sql%rowcount;

dbms_output.put_line(total_rows || ' sal updated ');

END IF; END;



```
DECLARE
total_rows number(2);
BEGIN
UPDATE moderators
SET SALARY= SALARY+ 500;
IF sql%notfound THEN
dbms_output.put_line('no sal updated');
 ELSIF sql%found THEN
total_rows := sql%rowcount;
dbms_output.put_line( total_rows || ' sal updated ');
 END IF;
END; /

rollback;

select * from moderators;
```

```
3 sal updated

Statement processed.
```

- create a carsure that print admin id and salary from admin table

DECLARE

id Admins.Adminid%type;

n Admins.NAME%type;

e Admins.EMAIL%type;

p Admins.PHONE%type;

add Admins.ADDRESS%type;

s Admins.salary%type;

cursor adminsal is

select * from Admins;

begin

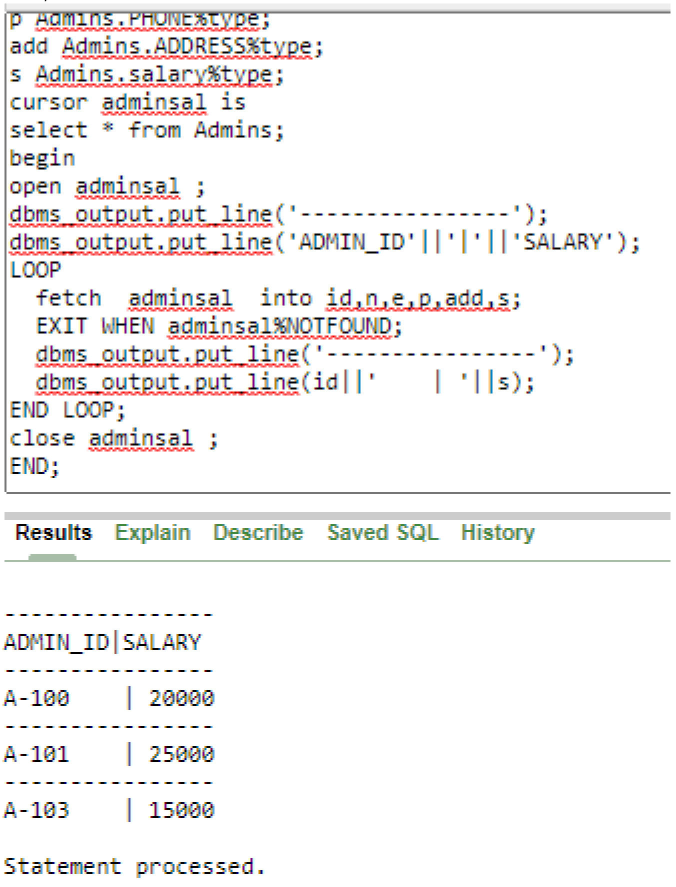
open adminsal ;

dbms_output.put_line('----------------');

dbms_output.put_line('ADMIN_ID'||'|'||'SALARY');

LOOP

```
  fetch  adminsal  into id,n,e,p,add,s;

  EXIT WHEN adminsal%NOTFOUND;

  dbms_output.put_line('----------------');

  dbms_output.put_line(id||'    | '||s);

END LOOP;

close adminsal ;

END;
```



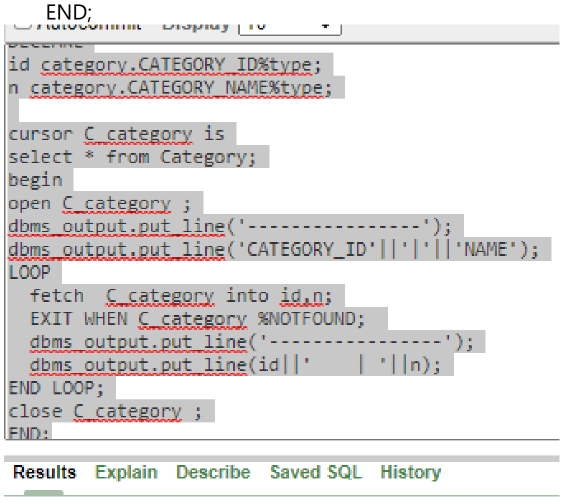
```
p Admins.PHONE%type;
add Admins.ADDRESS%type;
s Admins.salary%type;
cursor adminsal is
select * from Admins;
begin
open adminsal ;
dbms_output.put_line('----------------');
dbms_output.put_line('ADMIN_ID'||'|'||'SALARY');
LOOP
  fetch  adminsal  into id,n,e,p,add,s;
  EXIT WHEN adminsal%NOTFOUND;
  dbms_output.put_line('----------------');
  dbms_output.put_line(id||'    | '||s);
END LOOP;
close adminsal ;
END;
```

```
----------------
ADMIN_ID|SALARY
----------------
A-100    | 20000
----------------
A-101    | 25000
----------------
A-103    | 15000

Statement processed.
```

- create a carsure that print caategory id and name from category table

```
DECLARE

id category.CATEGORY_ID%type;

n category.CATEGORY_NAME%type;
```

```
cursor C_category is
select * from Category;
begin
open C_category ;
dbms_output.put_line('----------------');
dbms_output.put_line('CATEGORY_ID'||'|'||'NAME');
LOOP
  fetch  C_category into id,n;
  EXIT WHEN C_category %NOTFOUND;
  dbms_output.put_line('----------------');
  dbms_output.put_line(id||'    | '||n);
END LOOP;
close C_category ;
END;
```



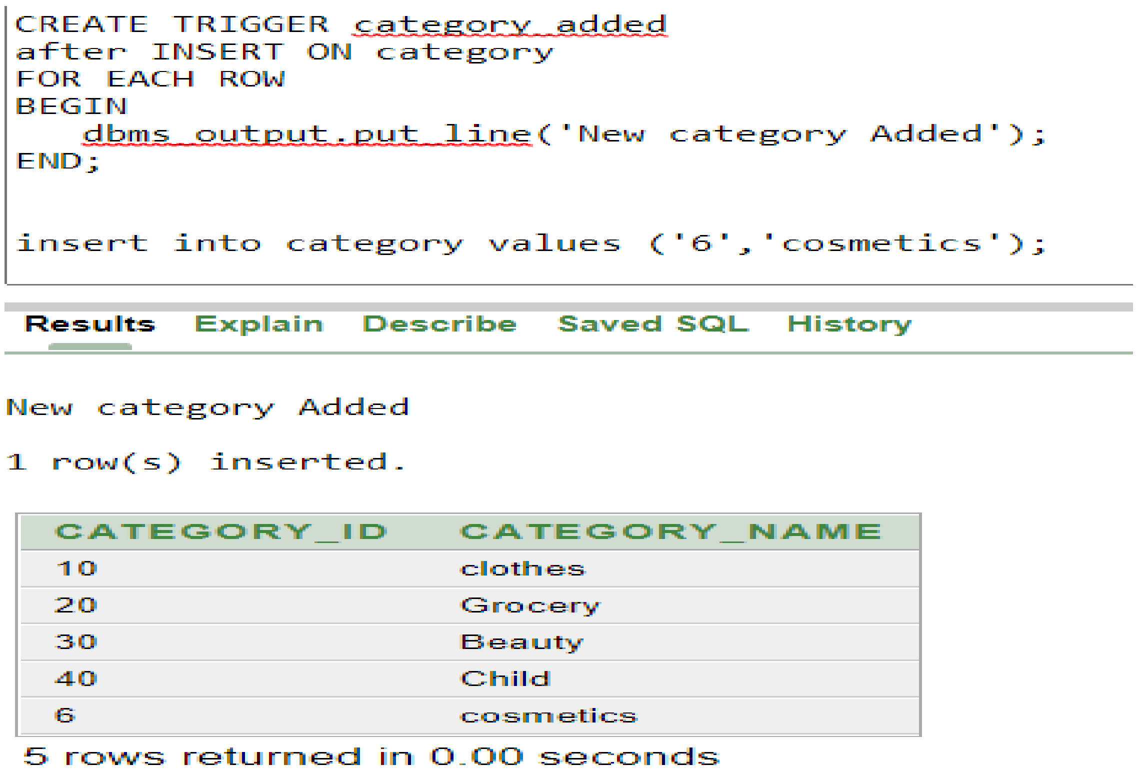
```
DECLARE
id category.CATEGORY_ID%type;
n category.CATEGORY_NAME%type;

cursor C_category is
select * from Category;
begin
open C_category ;
dbms_output.put_line('----------------');
dbms_output.put_line('CATEGORY_ID'||'|'||'NAME');
LOOP
  fetch  C_category into id,n;
  EXIT WHEN C_category %NOTFOUND;
  dbms_output.put_line('----------------');
  dbms_output.put_line(id||'    | '||n);
END LOOP;
close C_category ;
END;
```

```
----------------
CATEGORY_ID|NAME
----------------
10     | Cloths
----------------
20     | Grocery
----------------
30     | Beauty
----------------
40     | Child

Statement processed.
```

13.5 Trigger

1. Create a trigger in such a way that whenever a new row is inserted into the category table an output ‘New Category Added’ is generated.

```
CREATE TRIGGER category_added

after INSERT ON category

FOR EACH ROW

BEGIN

  dbms_output.put_line('New category Added');

END;



insert into category values ('6','cosmetics');
```



```
CREATE TRIGGER category_added
after INSERT ON category
FOR EACH ROW
BEGIN
   dbms_output.put_line('New category Added');
END;

insert into category values ('6','cosmetics');
```

```
New category Added

1 row(s) inserted.
```

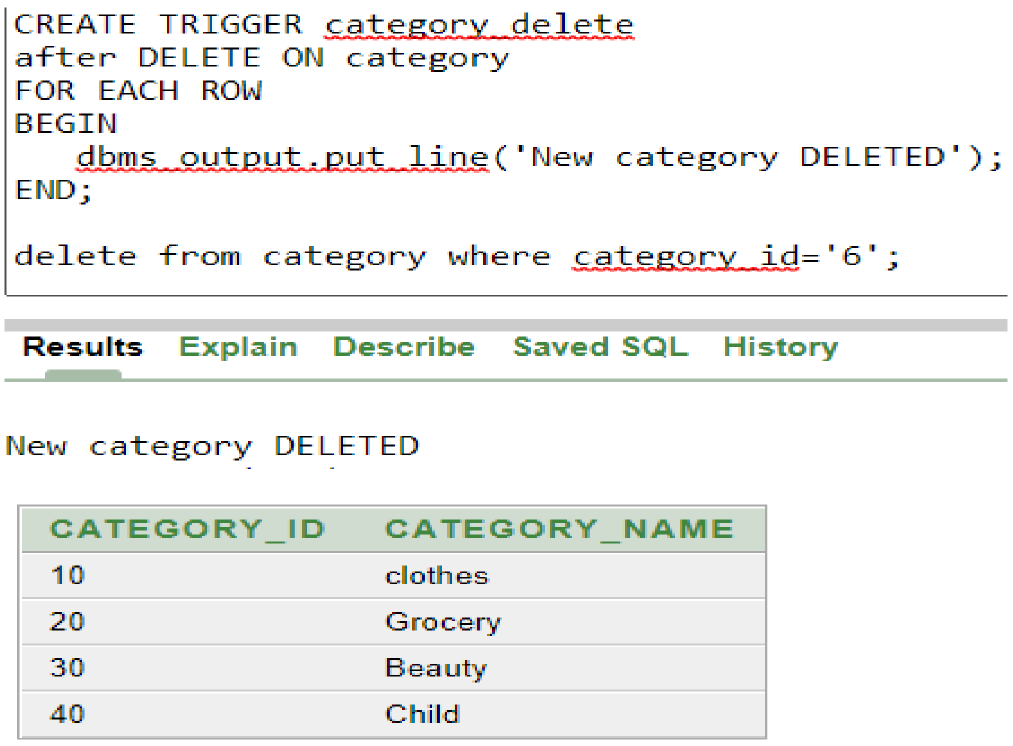
CATEGORY_ID	CATEGORY_NAME
10	clothes
20	Grocery
30	Beauty
40	Child
6	cosmetics

5 rows returned in 0.00 seconds

2. Create a trigger in such a way that whenever a row is deleted from the category table an output ‘A Category Deleted’ is generated.

```
CREATE TRIGGER category_delete
```

```
after DELETE ON category

FOR EACH ROW

BEGIN

   dbms_output.put_line('New category DELETED');

END;



delete from category where category_id='6';



select * from category;
```



```
CREATE TRIGGER category_delete
after DELETE ON category
FOR EACH ROW
BEGIN
   dbms_output.put_line('New category DELETED');
END;

delete from category where category_id='6';
```

```
New category DELETED
```

CATEGORY_ID	CATEGORY_NAME
10	clothes
20	Grocery
30	Beauty
40	Child

3. Create a ***trigger*** in such a way that whenever a row is deleted from the category table an output ‘Category Updated is generated

```
CREATE TRIGGER category_updated

after UPDATE ON category

FOR EACH ROW
```

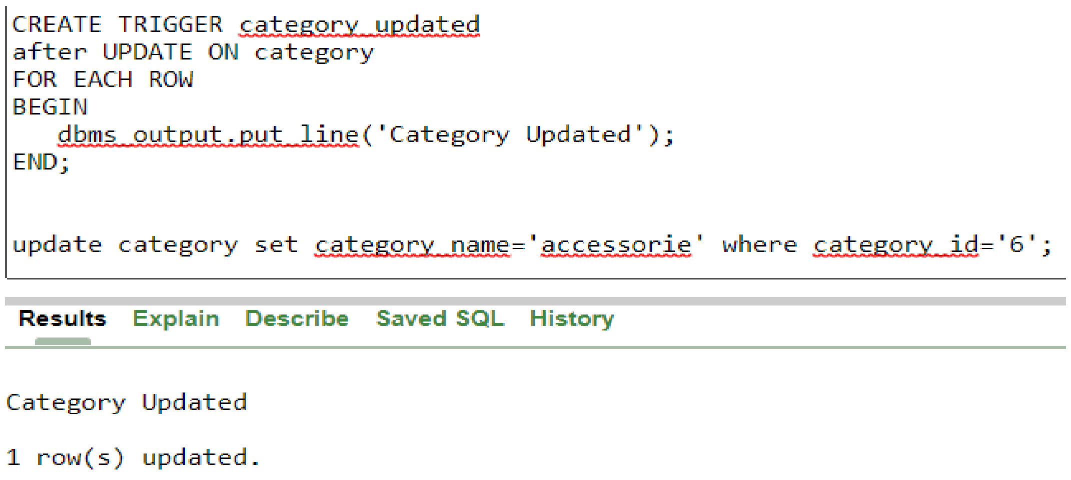
BEGIN

dbms_output.put_line('Category Updated');

END;

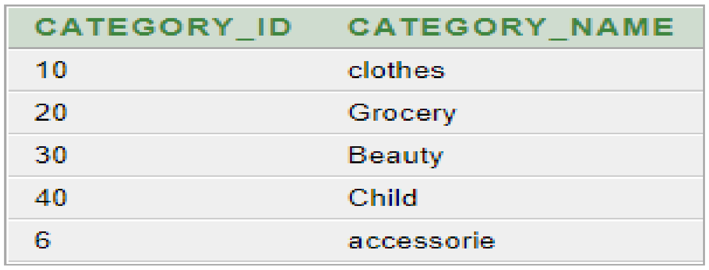
update category set category_name='accessorie' where category_id='6';

select * from category;



```
CREATE TRIGGER category_updated
after UPDATE ON category
FOR EACH ROW
BEGIN
   dbms_output.put_line('Category Updated');
END;


update category set category_name='accessorie' where category_id='6';
```

```
Category Updated

1 row(s) updated.
```



CATEGORY_ID	CATEGORY_NAME
10	clothes
20	Grocery
30	Beauty
40	Child
6	accessorie

13.6 Package

1. Create a package that contains a procedure which can display the product name of any product whose id is passed as its parameter.

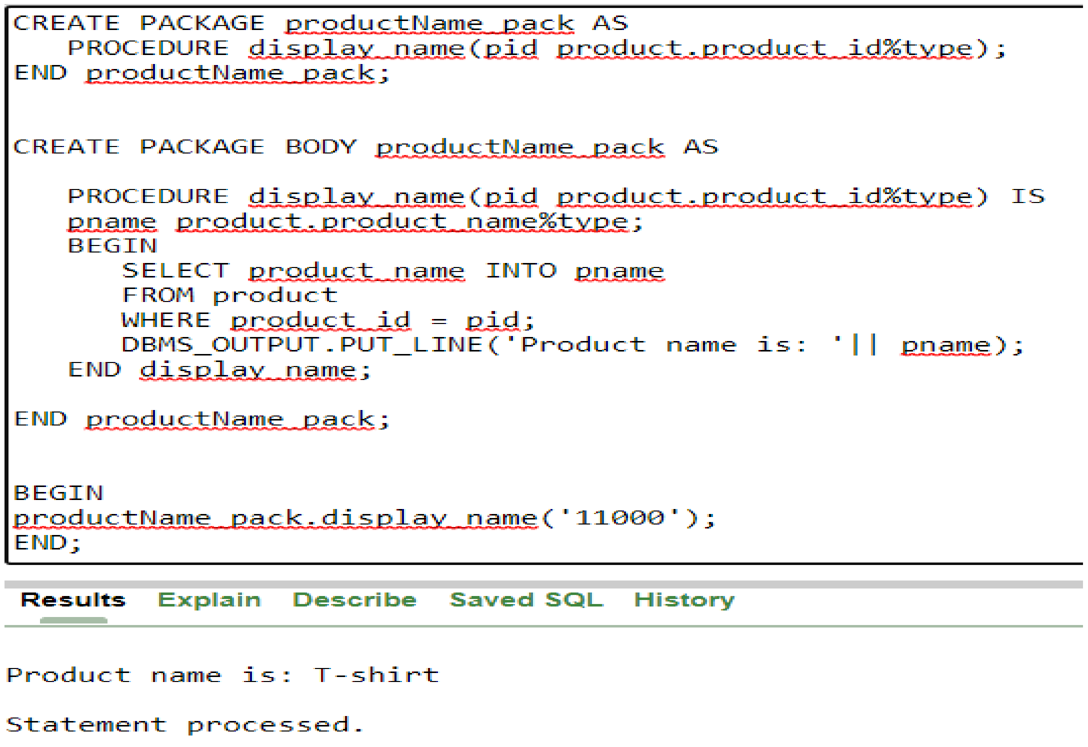
CREATE PACKAGE productName_pack AS

PROCEDURE display_name(pid product.product_id%type);

END productName_pack;

CREATE PACKAGE BODY productName_pack AS

```
  PROCEDURE display_name(pid product.product_id%type) IS

  pname product.product_name%type;

  BEGIN

    SELECT product_name INTO pname

    FROM product

    WHERE product_id = pid;

    DBMS_OUTPUT.PUT_LINE('Product name is: '|| pname);

  END display_name;


END productName_pack;



BEGIN

productName_pack.display_name('11000');

END;
```

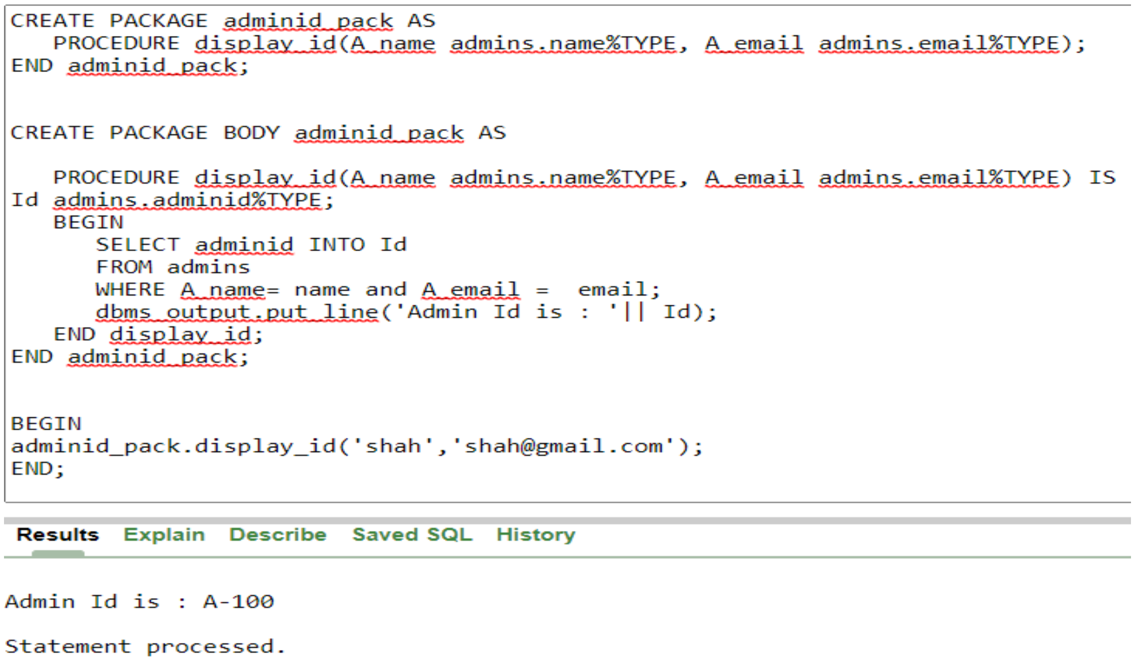


```
CREATE PACKAGE productName_pack AS
   PROCEDURE display_name(pid product.product_id%type);
END productName_pack;


CREATE PACKAGE BODY productName_pack AS

   PROCEDURE display_name(pid product.product_id%type) IS
   pname product.product_name%type;
   BEGIN
      SELECT product_name INTO pname
      FROM product
      WHERE product_id = pid;
      DBMS_OUTPUT.PUT_LINE('Product name is: '|| pname);
   END display_name;

END productName_pack;


BEGIN
productName_pack.display_name('11000');
END;
```

```
Product name is: T-shirt

Statement processed.
```

2. Create a package that contains a procedure which can display the adminid whose name and email is passed as its parameter.

```
CREATE PACKAGE adminid_pack AS

  PROCEDURE display_id(A_name admins.name%TYPE, A_email admins.email%TYPE);

END adminid_pack;




CREATE PACKAGE BODY adminid_pack AS



  PROCEDURE display_id(A_name admins.name%TYPE, A_email admins.email%TYPE) IS

Id admins.adminid%TYPE;

  BEGIN

    SELECT adminid INTO Id

    FROM admins

    WHERE A_name= name and A_email =  email;
    dbms_output.put_line('Admin Id is : '|| Id);

  END display_id;

END adminid_pack;



BEGIN

adminid_pack.display_id('shah','shah@gmail.com');

END;
```



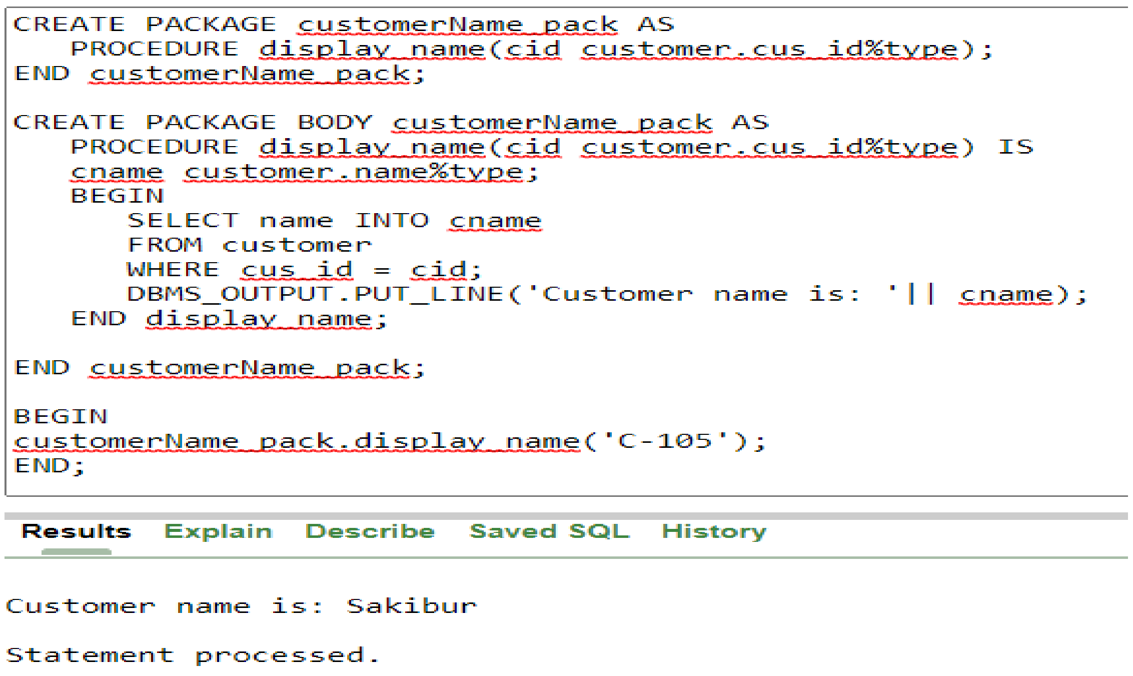
```
CREATE PACKAGE adminid_pack AS
   PROCEDURE display_id(A_name admins.name%TYPE, A_email admins.email%TYPE);
END adminid_pack;


CREATE PACKAGE BODY adminid_pack AS

   PROCEDURE display_id(A_name admins.name%TYPE, A_email admins.email%TYPE) IS
Id admins.adminid%TYPE;
   BEGIN
      SELECT adminid INTO Id
      FROM admins
      WHERE A_name= name and A_email =  email;
      dbms_output.put_line('Admin Id is : '|| Id);
   END display_id;
END adminid_pack;


BEGIN
adminid_pack.display_id('shah','shah@gmail.com');
END;
```

```
Admin Id is : A-100

Statement processed.
```

3. Create a package that contains a procedure which can display the customer name of any customer whose id is passed as its parameter.

```
CREATE PACKAGE customerName_pack AS

  PROCEDURE display_name(cid customer.cus_id%type);

END customerName_pack;



CREATE PACKAGE BODY customerName_pack AS

  PROCEDURE display_name(cid customer.cus_id%type) IS
```

```
      cname customer.name%type;

      BEGIN

        SELECT name INTO cname

        FROM customer

        WHERE cus_id = cid;

        DBMS_OUTPUT.PUT_LINE('Customer name is: '|| cname);

      END display_name;

END customerName_pack;

    BEGIN

    customerName_pack.display_name('C-105');

    END;
```



```
CREATE PACKAGE customerName_pack AS
   PROCEDURE display_name(cid customer.cus_id%type);
END customerName_pack;

CREATE PACKAGE BODY customerName_pack AS
   PROCEDURE display_name(cid customer.cus_id%type) IS
   cname customer.name%type;
   BEGIN
      SELECT name INTO cname
      FROM customer
      WHERE cus_id = cid;
      DBMS_OUTPUT.PUT_LINE('Customer name is: '|| cname);
   END display_name;

END customerName_pack;

BEGIN
customerName_pack.display_name('C-105');
END;
```

```
Customer name is: Sakibur

Statement processed.
```

14.Conclusion

The Internet has become a major resource in modern business, thus electronic shopping has gained significance not only from the entrepreneur’s but also from the customer’s point of view. For the entrepreneur, electronic shopping generates new business opportunities and for the customer, it makes comparative shopping possible.

As per a survey, most consumers of online stores are impulsive and usually decide to stay on a site within the first few seconds. “Website design is like a shop interior. If the shop looks poor or like hundreds of other shops the customer is most likely to skip to the other site. Hence we have designed the project to provide the user with easy navigation, retrieval of data, and necessary feedback as much as possible. In this project, the user is provided with an e-commerce website that can be used to buy books online. To implement this as a web application we used C# as the programming language and visual studio IDE.

A good shopping cart design must be accompanied by user-friendly shopping cart application logic. It should be convenient for the customer to view the contents of their cart and to be able to remove or add items to their cart. The shopping cart application described in this project provides many features that are designed to make the customer more comfortable.

This project helps in understanding the creation of an interactive web page and the technologies used to implement it. The design of the project which includes the Data Model and Process Model illustrates how the database is built with different tables, how the data is accessed and processed from the tables. The building of the project has given me precise knowledge about how ASP.NET is used to develop a website, how it connects to the database to access the data, and how the data and web pages are modified to provide the user with a shopping cart application