VIT®

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

ECOMMERCE PORTAL
DATABASE MANAGEMENT SYSTEMS
ITE1003
NOVEMBER 2019
FACULTY :- BIMAL KUMAR RAY

**NIRAJ VANDAKARI : 18BIT0001**
**AMAN CHAWLA:  18BIT0055**
**BL RAMAKRISHNA : 18BIT0175**
**ADARSH VATSA : 18BIT0295**

# ABSTRACT OF THE PROJECT:-

The objective of this project is to develop a general purpose ecommerce store where product like clothes 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 credit card number. An e-mail notification is sent to the customer as soon as the order is placed.

Electronic Commerce is process of doing business through computer networks. A person sitting on his chair in front of a computer can access all the facilities of the Internet to buy or sell the products.

Unlike traditional commerce that is carried out physically with effort of a person to go & get products, ecommerce has made it easier for human to reduce physical work and to save time. E-Commerce which was started in early 1990's has taken a great leap in the world of computers, but the fact that has hindered the growth of e- commerce is security. Security is the challenge facing e-commerce today & there is still a lot of advancement made in the field of security.

The main advantage of e-commerce over traditional commerce is the user can browse online shops, compare prices and order merchandise sitting at home on their PC.

For increasing the use of e-commerce in developing countries the B2B e-commerce is implemented for improving access to global markets for firms in developing countries. For a developing country advancement in the field of e-commerce is essential. The research strategy shows the importance of the e-commerce in developing countries for business applications.

Electronic commerce or ecommerce is a term for any type of business, or commercial transaction, that involves the transfer of information across the Internet. It covers a range of different types of businesses, from consumer based retail sites, through auction or music sites, to business exchanges trading goods and services between corporations. It is currently one of the most important aspects of the Internet to emerge

## : DATA REQUIREMENTS

Let us consider data requirements of our Ecommerce website.

Website must contain customer database where each customer must have a unique **customer id and email**.

It should contain username, email and password for each customer. Personal information such as Name, Phone no, and address. A customer can have multiple phone no and address so that must be taken into consideration.

A customer can choose certain **category** of products so it must have its own database which will contain category id and category name.

Each category can have multiple **brands** associated with them so a database is required to store unique brand id and brand name.

It is not always necessary for customer to choose from brand he/she can directly choose a **product**.The product database must contain a unique product id along with product name,price and modes.

Information about **supplier** of each product is crucial for any business.Suppliers can be identified by unique supplier_id .

Also contact number of company is needed along with its name where a company can have first and last name.

Each customer has his/her unique **cart** which is recognized by its unique cart id.The main feature of cart is to display total cost of all the products purchased by the customer.

The customer can make transaction by making **payment**.

A unique payment id is required for each transaction.Also there can different modes of payment.

To make payment it is necessary to have **order** first.There must be unique id for each order.Also each order must contain information about its expected arrival time along with day of shipment of product.

## FUNCTIONAL REQUIREMENTS:-

- The selected items should be collected in a shopping cart.
- At checkout time, the items in the shopping cart will be presented as an order.
- At checkout 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 credit card number
- An e-mail notification is sent to the customer as soon as the order is placed.
- Any member can register and view available products.
- Only registered member can purchase multiple products regardless of quantity.
- There are three roles available: Visitor, User and Admin.
  - Visitor can view available products.
  - User can view and purchase products.
  - An Admin has some extra privilege including all privilege of visitor and user.
- Admin can add products, edit product information and add/remove product. Admin can add user, edit user information and can remove user. Admin can ship order to user based on order placed by sending confirmation mail.
- Once users order item they are able to see ordered products and grand total.

## FOR DATA RETREIVAL.

: View Users :: The admin will have a list view of all the users registered in the system. Admin can view all the details of each user in the list except password.

: View Order :: Administrator can view the Orders which is generated by the users. He can verify the details of the purchase.

## FOR DATA REMOVAL

: Delete &Block Users :: Administrator has a right to delete or block a user. The default status of a new user registered is set as blocked. The admin must accept the new user by unblocking him.

:Delete Products :: Administrator can delete the products based on the stock of that particular product.

## FOR DATA MODIFICATION

: Add to cart :: The user can add the desired product into his cart by clicking add to cart option on the product. He can view his cart by clicking on the cart button. All products added by cart can be viewed in the cart. User can remove an item from the cart by clicking remove.

: Add Users :: Admin has privileges to add a user directly by providing the details

This database consists of customer:-

- customer and Admin information is added to database with Unique ID based on their roles.
- Complete products information is stored in this table.
- Customer ordered products, status and delivery information is stored in this table.

## ***Non-Functional / Operational Requirements :-***

### ***Security:-***

- Pages of the website must be access in the way they were intended to be accessed.
- Included files shall not be accessed outside of their parent file. Administrator can only perform administrative task on pages they are privileged to access.
- Customers will not be allowed to access the administrator pages.

### ***Efficiency and Maintainability:-***

- Page loads should be returned and formatted in a timely fashion depending on the request being made.
- Administrators will have the ability to edit the aspects of the order forms, product descriptions, prices and website directly

**Entities,** *attributes* and primary key used**:**

1. **CUSTOMER**

   *ATTRIBUTES:*

- CUSTOMER_ID
- EMAIL-ID
- USERNAME
- NAME { F_NAME, L_NAME} – Composite Attribute
- PASSWORD
- Address – Multi valued Attribute
- PHONE_NO

2. **CATEGORY**

   *ATTRIBUTES:*

- CATEGORY_ID
- CATEGORY_NAME

**3. BRAND**

*ATTRIBUTES:*

- BRAND_ID
- BRAND_NAME

**4. PRODUCTS**

*ATTRIBUTES:*

- PRODUCT_ID
- PRODUCT_NAME
- PRICE
- MODES

**5. SUPPLIER**

*ATTRIBUTES:*

- SUPPLIER_ID
- COMPANY_CONTACT
- NAME { C_FNAME, C_LNAME} – Composite Attribute

*6.* **ORDERS**

(WEAK ENTITY)

*ATTRIBUTES:*

- ORDER_ID (DISCRIMINANT)

EXPECTED_DATE

- SHIPPED_DATE

*7.* **PAYMENT**

(WEAK ENTITY)

*ATTRIBUTES:*

- PAYMENT_ID (DISCRIMINANT)
- TYPE
-

**8. CART**

(WEAK ENTITY)

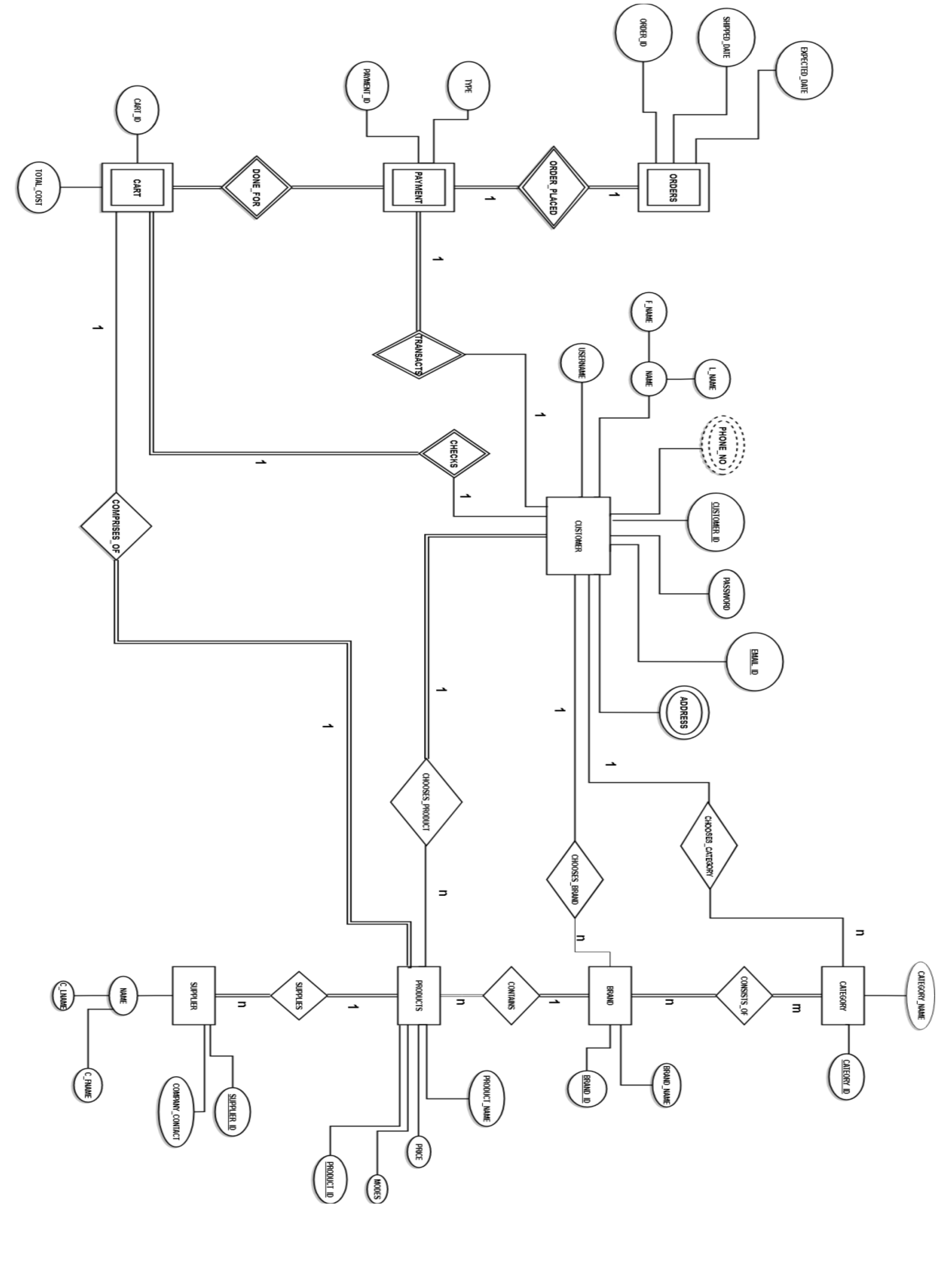
*ATTRIBUTES:*

- CART_ID (DISCRIMINANT)
- TOTAL_COST

## RELATIONSHIP TYPE:

- **1-1**
- CUSTOMER & PAYMENT
- ORDERS & PAYMENT
- PAYMENT & CART
- CART & PRODUCTS
- CUSTOMER & CART
- **1:N or N:1**
  - CUSTOMER & CATEGORY
  - CUSTOMER & BRAND
  - CUSTOMER & PRODUCT
  - BRAND & PRODUCTS
  - PRODUCTS & SUPPLIER
- **M:N**
- CATEGORY & BRAND

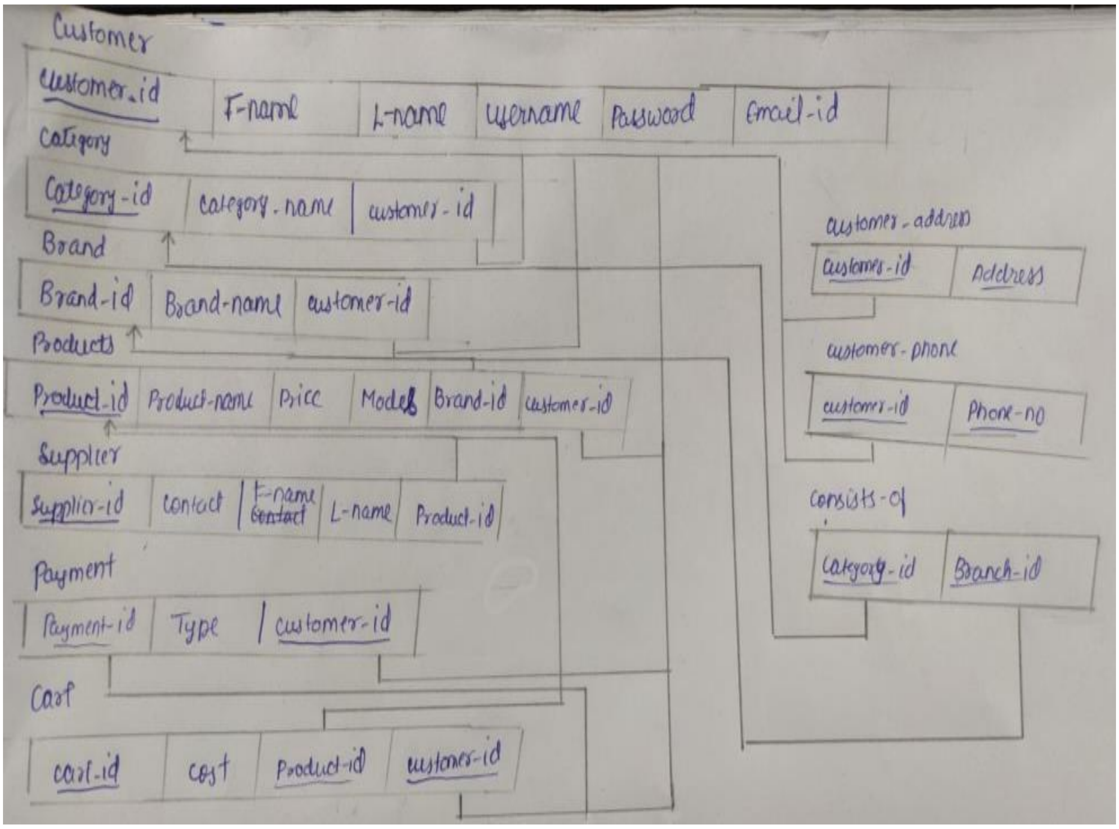
# Entity Relationship Diagram.



# REVIEW 2:-

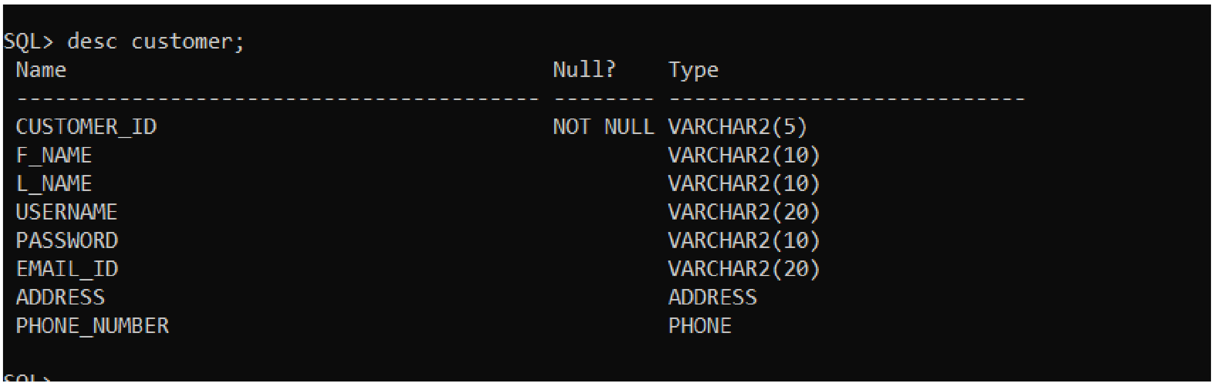
# DATA INPUTS AND TABLE CREATION

## RELATIONSHIP SCHEMA



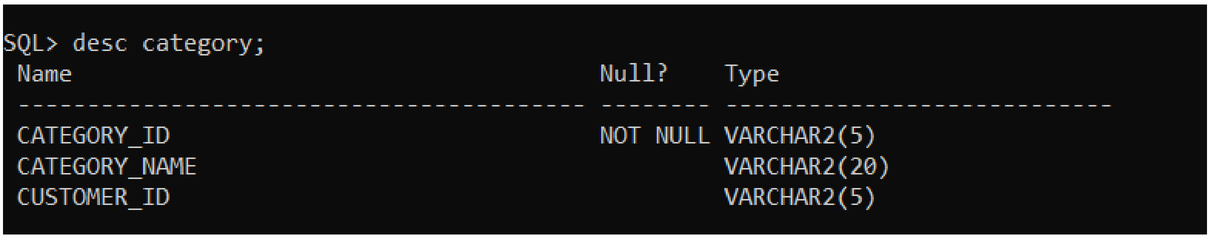
## TABLE CREATIONS

1.customer table



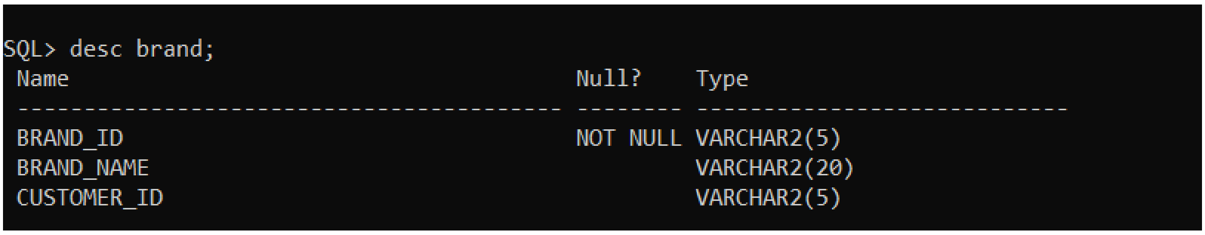
```
SQL> desc customer;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CUSTOMER_ID                               NOT NULL VARCHAR2(5)
 F_NAME                                             VARCHAR2(10)
 L_NAME                                             VARCHAR2(10)
 USERNAME                                           VARCHAR2(20)
 PASSWORD                                           VARCHAR2(10)
 EMAIL_ID                                           VARCHAR2(20)
 ADDRESS                                            ADDRESS
 PHONE_NUMBER                                       PHONE

SQL>
```

Category:-



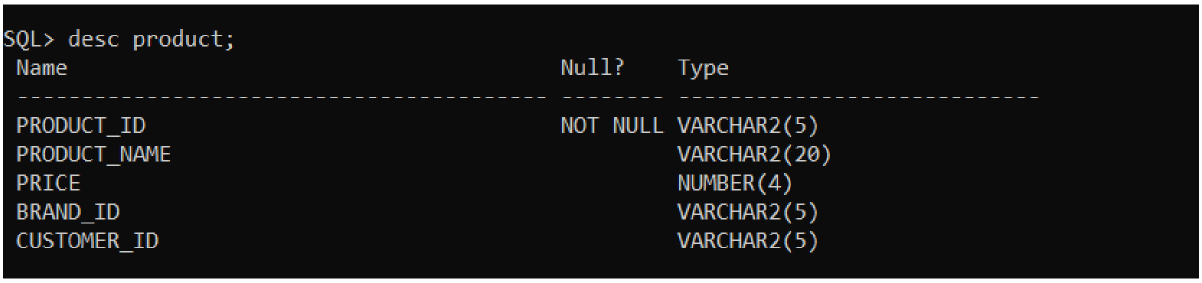
```
SQL> desc category;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CATEGORY_ID                               NOT NULL VARCHAR2(5)
 CATEGORY_NAME                                      VARCHAR2(20)
 CUSTOMER_ID                                        VARCHAR2(5)
```

Brand:-



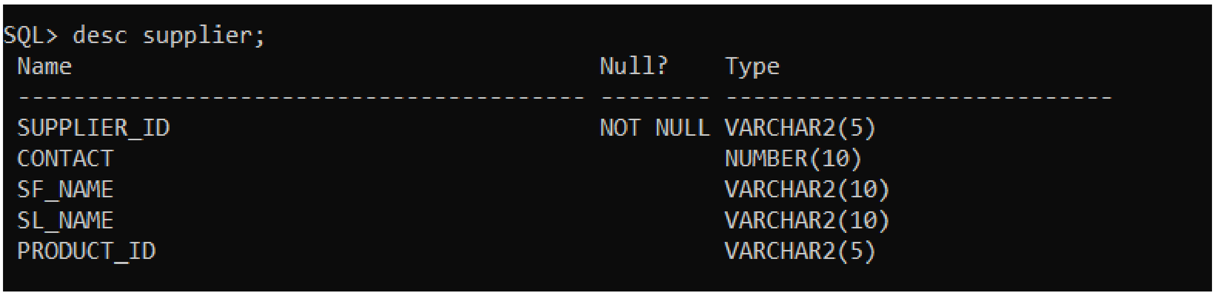
```
SQL> desc brand;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BRAND_ID                                  NOT NULL VARCHAR2(5)
 BRAND_NAME                                         VARCHAR2(20)
 CUSTOMER_ID                                        VARCHAR2(5)
```

Product:-



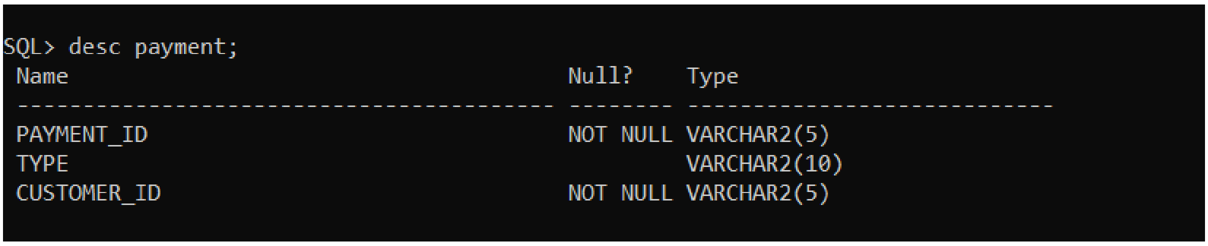
```
SQL> desc product;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PRODUCT_ID                                NOT NULL VARCHAR2(5)
 PRODUCT_NAME                                       VARCHAR2(20)
 PRICE                                              NUMBER(4)
 BRAND_ID                                           VARCHAR2(5)
 CUSTOMER_ID                                        VARCHAR2(5)
```

Supplier:-



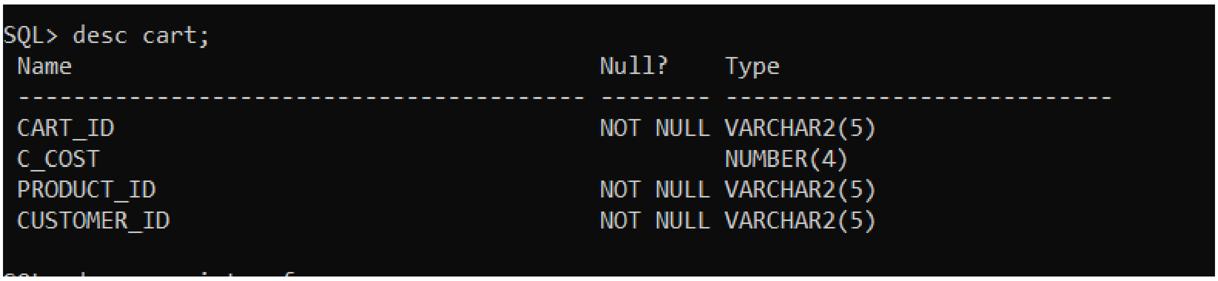
```
SQL> desc supplier;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SUPPLIER_ID                               NOT NULL VARCHAR2(5)
 CONTACT                                            NUMBER(10)
 SF_NAME                                            VARCHAR2(10)
 SL_NAME                                            VARCHAR2(10)
 PRODUCT_ID                                         VARCHAR2(5)
```

Payment:-



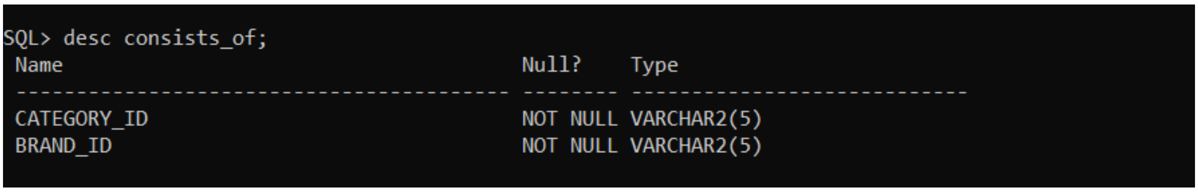
```
SQL> desc payment;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PAYMENT_ID                                NOT NULL VARCHAR2(5)
 TYPE                                               VARCHAR2(10)
 CUSTOMER_ID                               NOT NULL VARCHAR2(5)
```

Cart:-



```
SQL> desc cart;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CART_ID                                   NOT NULL VARCHAR2(5)
 C_COST                                             NUMBER(4)
 PRODUCT_ID                                NOT NULL VARCHAR2(5)
 CUSTOMER_ID                               NOT NULL VARCHAR2(5)
```

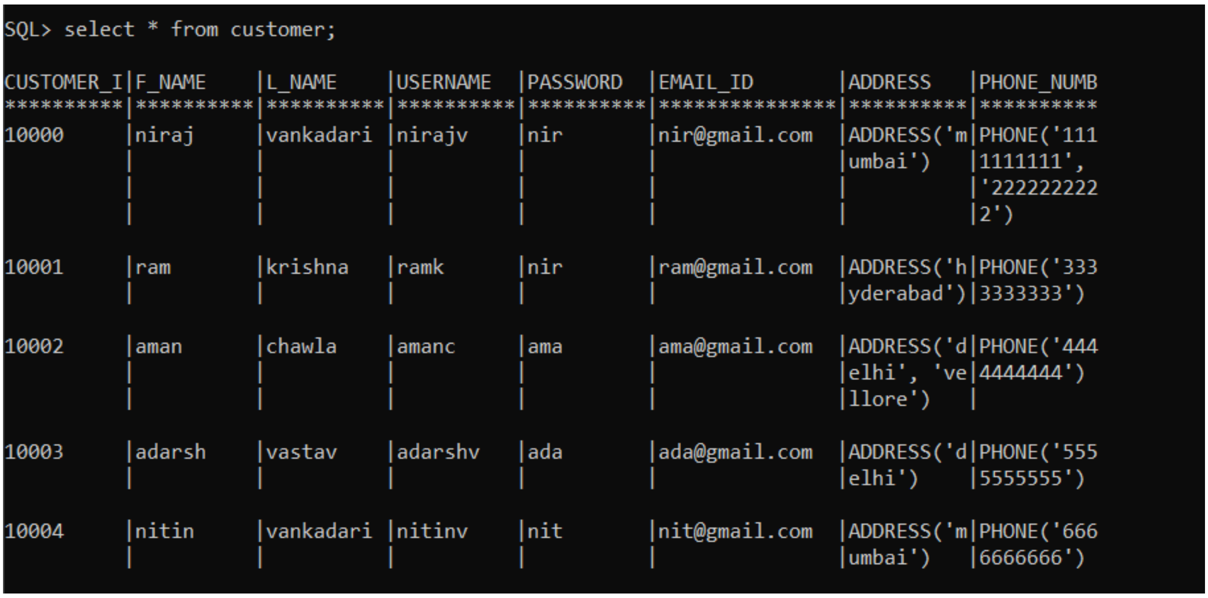
# Consists_of:-



```
SQL> desc consists_of;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CATEGORY_ID                               NOT NULL VARCHAR2(5)
 BRAND_ID                                  NOT NULL VARCHAR2(5)
```

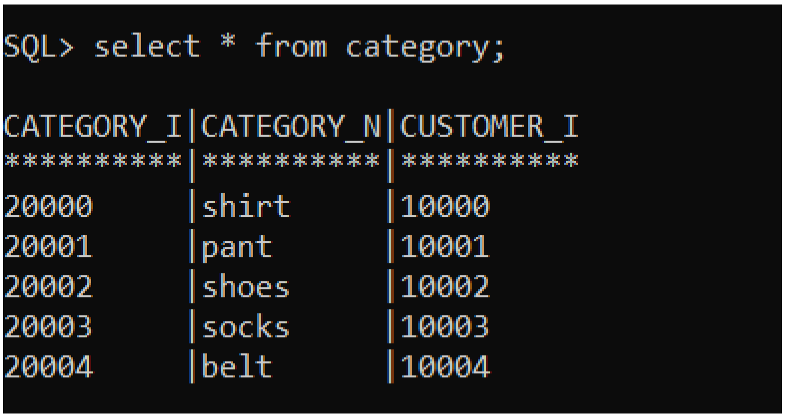
## TABLES

Customer table data:-



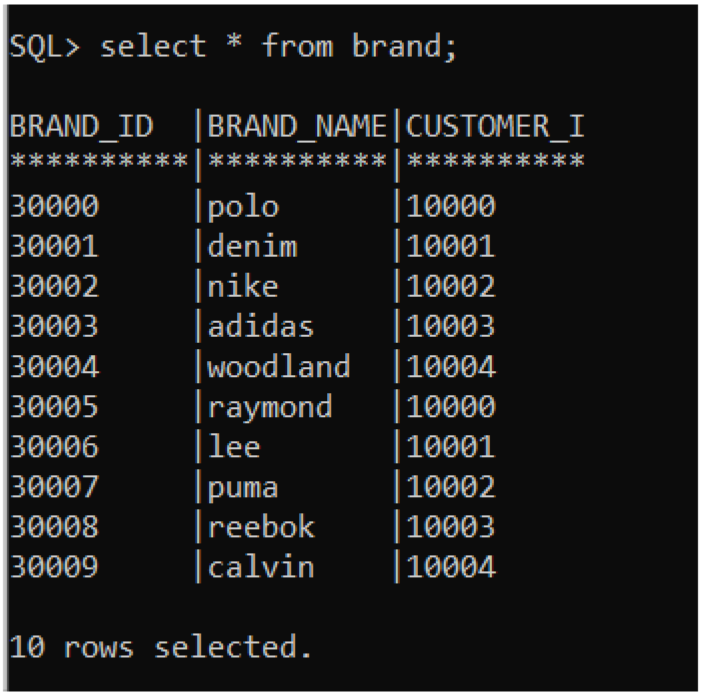
```
SQL> select * from customer;

CUSTOMER_I|F_NAME    |L_NAME    |USERNAME  |PASSWORD  |EMAIL_ID       |ADDRESS   |PHONE_NUMB
**********|**********|**********|**********|**********|***************|**********|**********
10000     |niraj     |vankadari |nirajv    |nir       |nir@gmail.com  |ADDRESS('m|PHONE('111
          |          |          |          |          |               |umbai')   |1111111',
          |          |          |          |          |               |          |'222222222
          |          |          |          |          |               |          |2')

10001     |ram       |krishna   |ramk      |nir       |ram@gmail.com  |ADDRESS('h|PHONE('333
          |          |          |          |          |               |yderabad')|3333333')

10002     |aman      |chawla    |amanc     |ama       |ama@gmail.com  |ADDRESS('d|PHONE('444
          |          |          |          |          |               |elhi', 've|4444444')
          |          |          |          |          |               |llore')   |

10003     |adarsh    |vastav    |adarshv   |ada       |ada@gmail.com  |ADDRESS('d|PHONE('555
          |          |          |          |          |               |elhi')    |5555555')

10004     |nitin     |vankadari |nitinv    |nit       |nit@gmail.com  |ADDRESS('m|PHONE('666
          |          |          |          |          |               |umbai')   |6666666')
```

Category table data:-



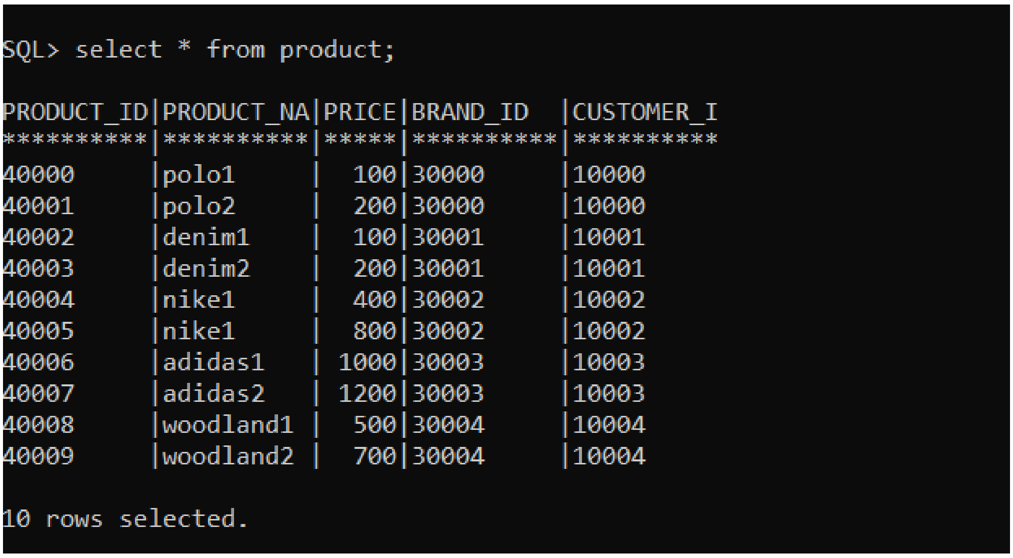
```
SQL> select * from category;

CATEGORY_I|CATEGORY_N|CUSTOMER_I
**********|**********|**********
20000     |shirt     |10000
20001     |pant      |10001
20002     |shoes     |10002
20003     |socks     |10003
20004     |belt      |10004
```

Brand table data:--



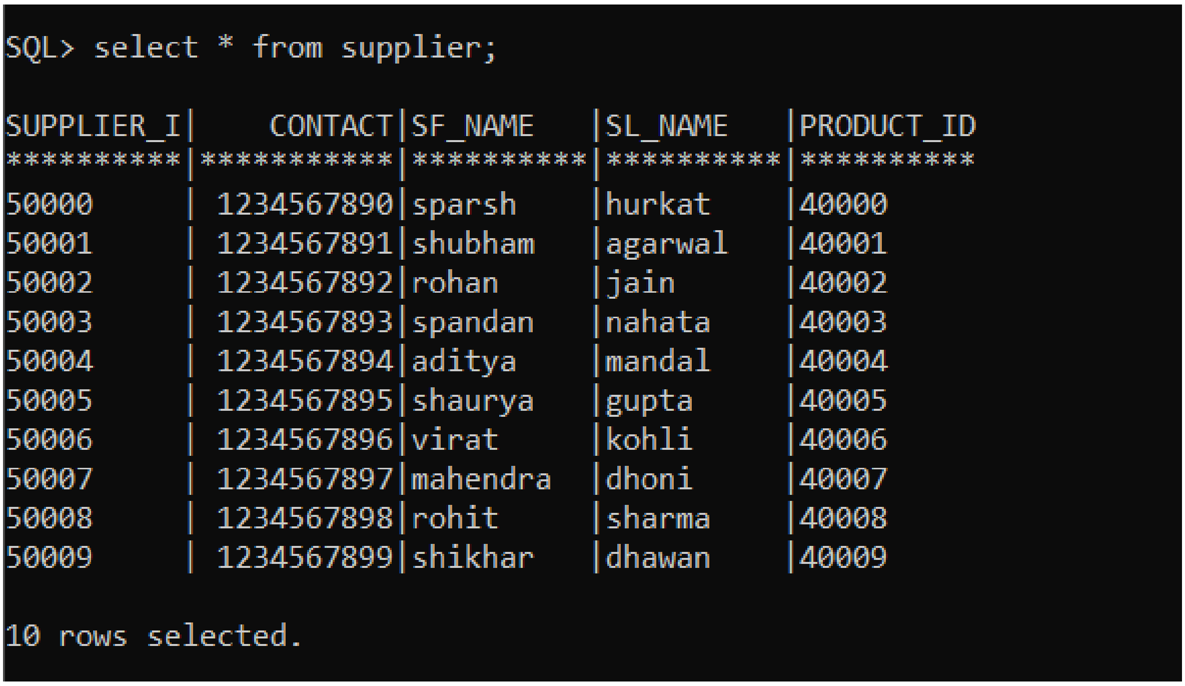
```
SQL> select * from brand;

BRAND_ID  |BRAND_NAME|CUSTOMER_I
**********|**********|**********
30000     |polo      |10000
30001     |denim     |10001
30002     |nike      |10002
30003     |adidas    |10003
30004     |woodland  |10004
30005     |raymond   |10000
30006     |lee       |10001
30007     |puma      |10002
30008     |reebok    |10003
30009     |calvin    |10004

10 rows selected.
```

Product table data:-



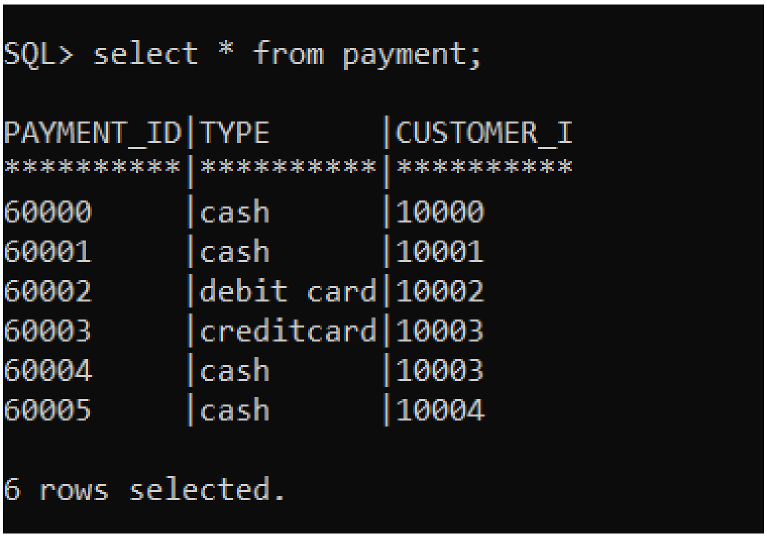
```
SQL> select * from product;

PRODUCT_ID|PRODUCT_NA|PRICE|BRAND_ID  |CUSTOMER_I
**********|**********|*****|**********|**********
40000     |polo1     |  100|30000     |10000
40001     |polo2     |  200|30000     |10000
40002     |denim1    |  100|30001     |10001
40003     |denim2    |  200|30001     |10001
40004     |nike1     |  400|30002     |10002
40005     |nike1     |  800|30002     |10002
40006     |adidas1   | 1000|30003     |10003
40007     |adidas2   | 1200|30003     |10003
40008     |woodland1 |  500|30004     |10004
40009     |woodland2 |  700|30004     |10004

10 rows selected.
```

Supplier table date:-



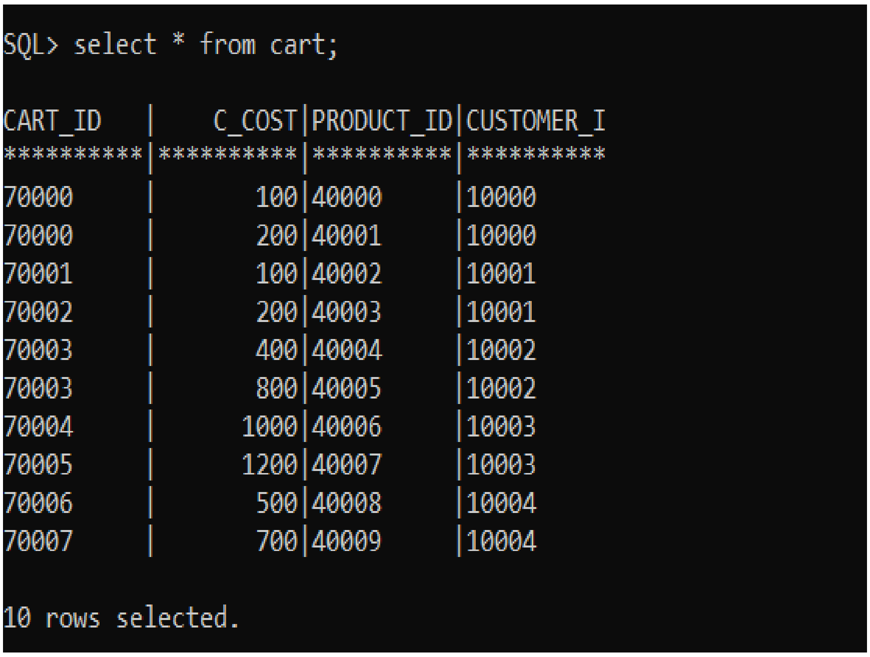
```
SQL> select * from supplier;

SUPPLIER_I|    CONTACT|SF_NAME   |SL_NAME   |PRODUCT_ID
**********|***********|**********|**********|**********
50000     | 1234567890|sparsh    |hurkat    |40000
50001     | 1234567891|shubham   |agarwal   |40001
50002     | 1234567892|rohan     |jain      |40002
50003     | 1234567893|spandan   |nahata    |40003
50004     | 1234567894|aditya    |mandal    |40004
50005     | 1234567895|shaurya   |gupta     |40005
50006     | 1234567896|virat     |kohli     |40006
50007     | 1234567897|mahendra  |dhoni     |40007
50008     | 1234567898|rohit     |sharma    |40008
50009     | 1234567899|shikhar   |dhawan    |40009

10 rows selected.
```

Payment table data:-



```
SQL> select * from payment;

PAYMENT_ID|TYPE      |CUSTOMER_I
**********|**********|**********
60000     |cash      |10000
60001     |cash      |10001
60002     |debit card|10002
60003     |creditcard|10003
60004     |cash      |10003
60005     |cash      |10004

6 rows selected.
```

Cart table data:-



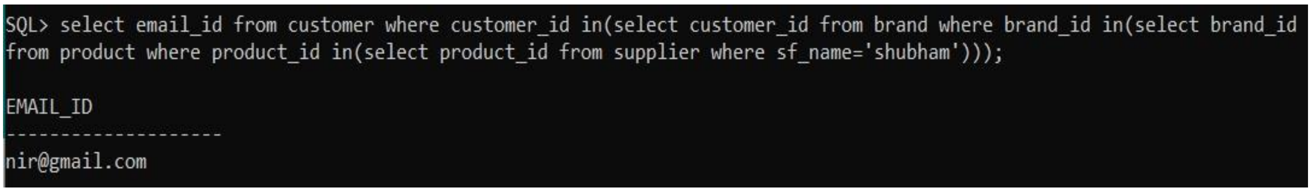
```
SQL> select * from cart;

CART_ID   |    C_COST|PRODUCT_ID|CUSTOMER_I
**********|**********|**********|**********
70000     |       100|40000     |10000
70000     |       200|40001     |10000
70001     |       100|40002     |10001
70002     |       200|40003     |10001
70003     |       400|40004     |10002
70003     |       800|40005     |10002
70004     |      1000|40006     |10003
70005     |      1200|40007     |10003
70006     |       500|40008     |10004
70007     |       700|40009     |10004

10 rows selected.
```

# REVIEW 3

## DATABASE OPERATIONS

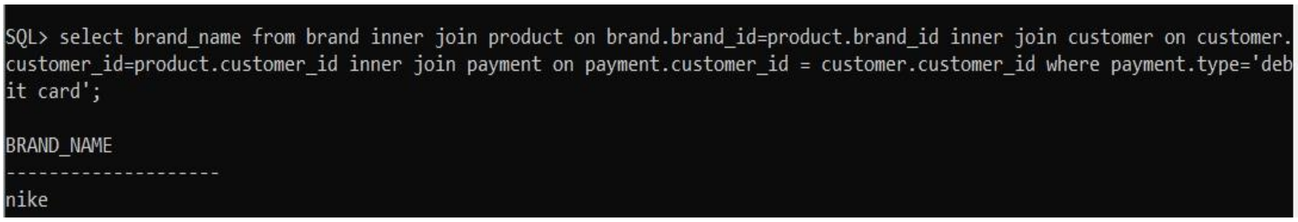
### DATABASE RETRIVAL:-

Use of nested query to retrieve email of customer who has purchased product supplied by supplier named ‘subham’



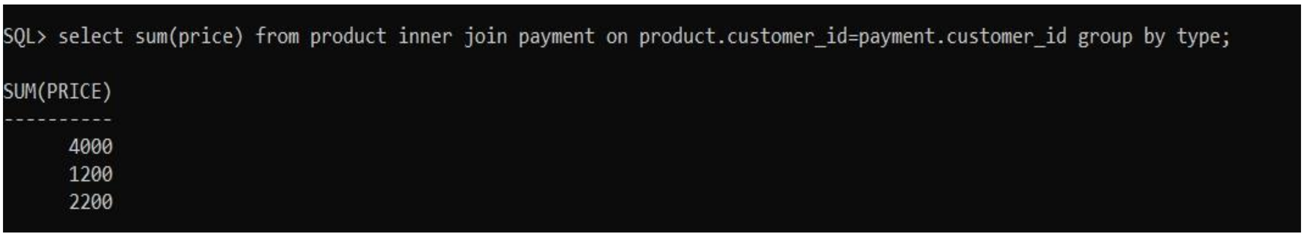
```
SQL> select email_id from customer where customer_id in(select customer_id from brand where brand_id in(select brand_id
from product where product_id in(select product_id from supplier where sf_name='shubham')));

EMAIL_ID
--------------------
nir@gmail.com
```

Join query to retrieve product that has been purchased using debit card



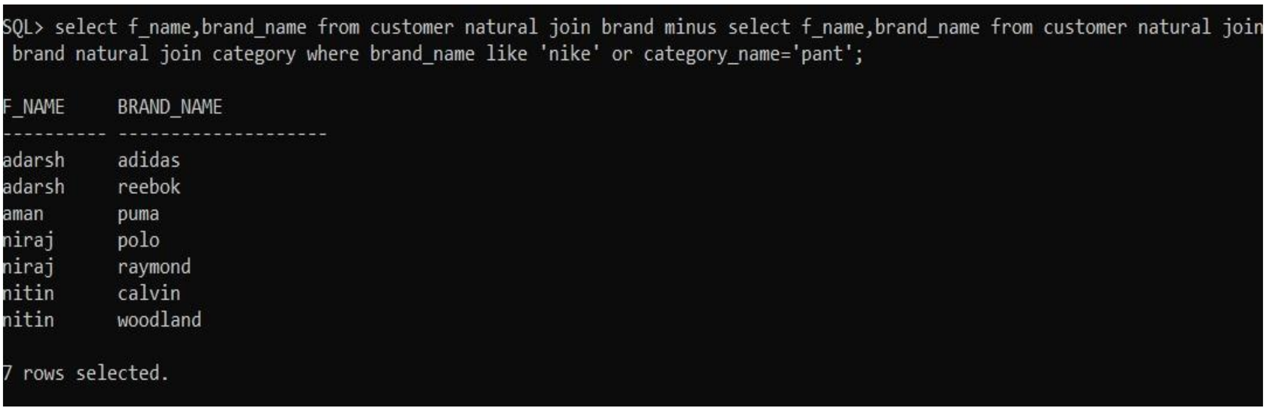
```
SQL> select brand_name from brand inner join product on brand.brand_id=product.brand_id inner join customer on customer.
customer_id=product.customer_id inner join payment on payment.customer_id = customer.customer_id where payment.type='deb
it card';

BRAND_NAME
--------------------
nike
```

Sum of price of each object sorted according product’s category.



```
SQL> select sum(price) from product inner join payment on product.customer_id=payment.customer_id group by type;

SUM(PRICE)
----------
      4000
      1200
      2200
```

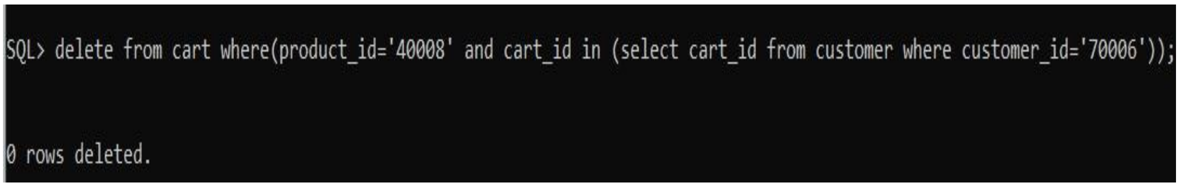
First name of customers who haven’t purchased nike or a pant.



```
SQL> select f_name,brand_name from customer natural join brand minus select f_name,brand_name from customer natural join
 brand natural join category where brand_name like 'nike' or category_name='pant';

F_NAME     BRAND_NAME
---------- --------------------
adarsh     adidas
adarsh     reebok
aman       puma
niraj      polo
niraj      raymond
nitin      calvin
nitin      woodland

7 rows selected.
```

## DATABASE DELETIONS

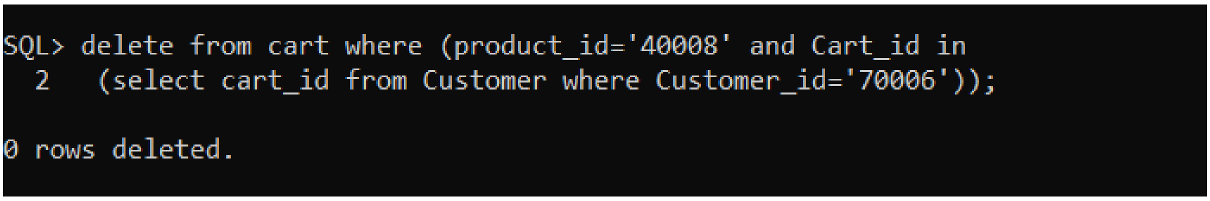
### If Customer wants to delete an item from the cart



```
SQL> delete from cart where(product_id='40008' and cart_id in (select cart_id from customer where customer_id='70006'));


0 rows deleted.
```

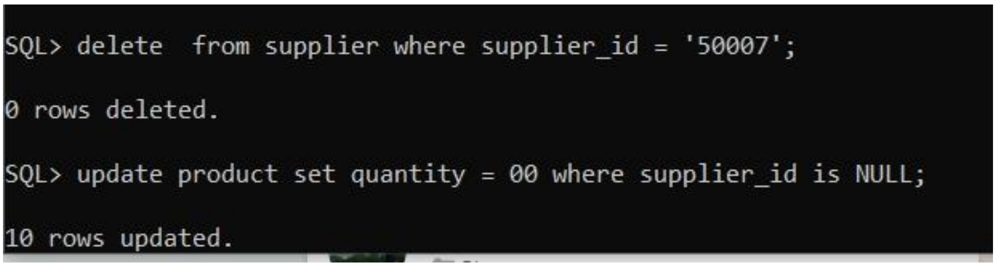
## DELETE QUERY

**customer wants to delete an item from  the cart**



```
SQL> delete from cart where (product_id='40008' and Cart_id in
  2   (select cart_id from Customer where Customer_id='70006'));

0 rows deleted.
```

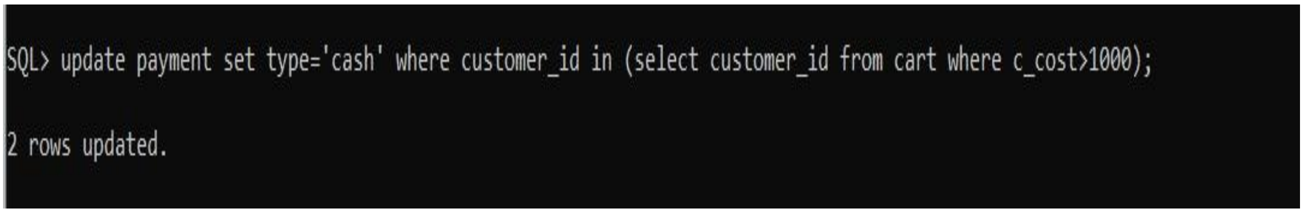
If supplier stops selling his product



```
SQL> delete  from supplier where supplier_id = '50007';

0 rows deleted.

SQL> update product set quantity = 00 where supplier_id is NULL;

10 rows updated.
```

## Modification

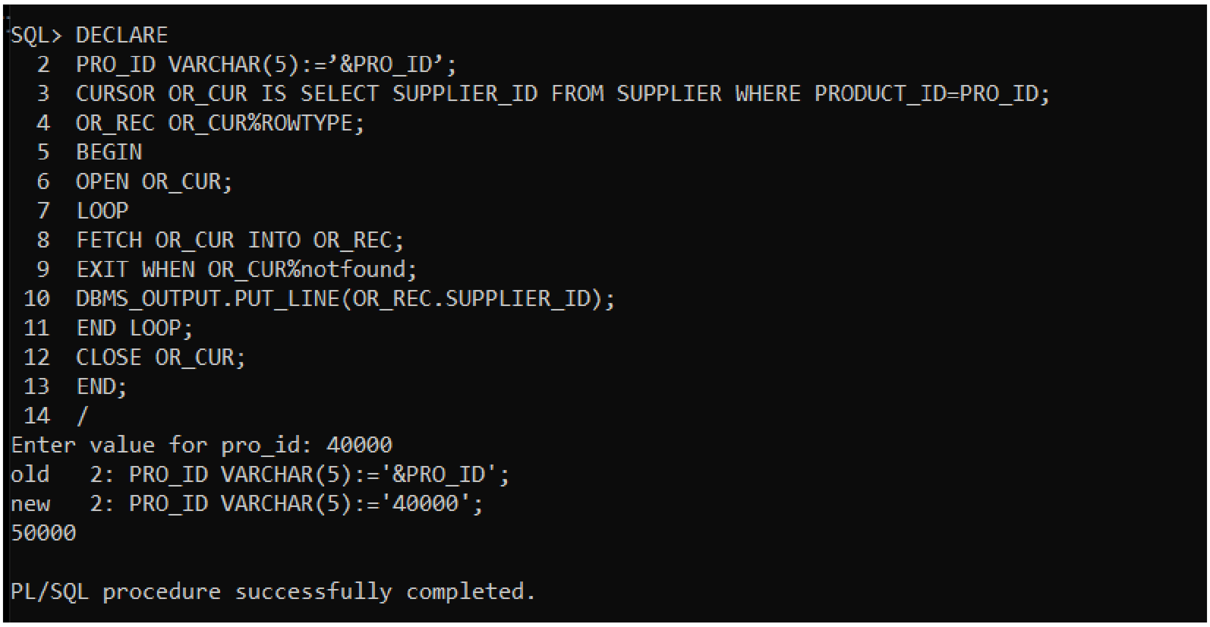
Update any payment method to cash if cost>1000



```
SQL> update payment set type='cash' where customer_id in (select customer_id from cart where c_cost>1000);

2 rows updated.
```

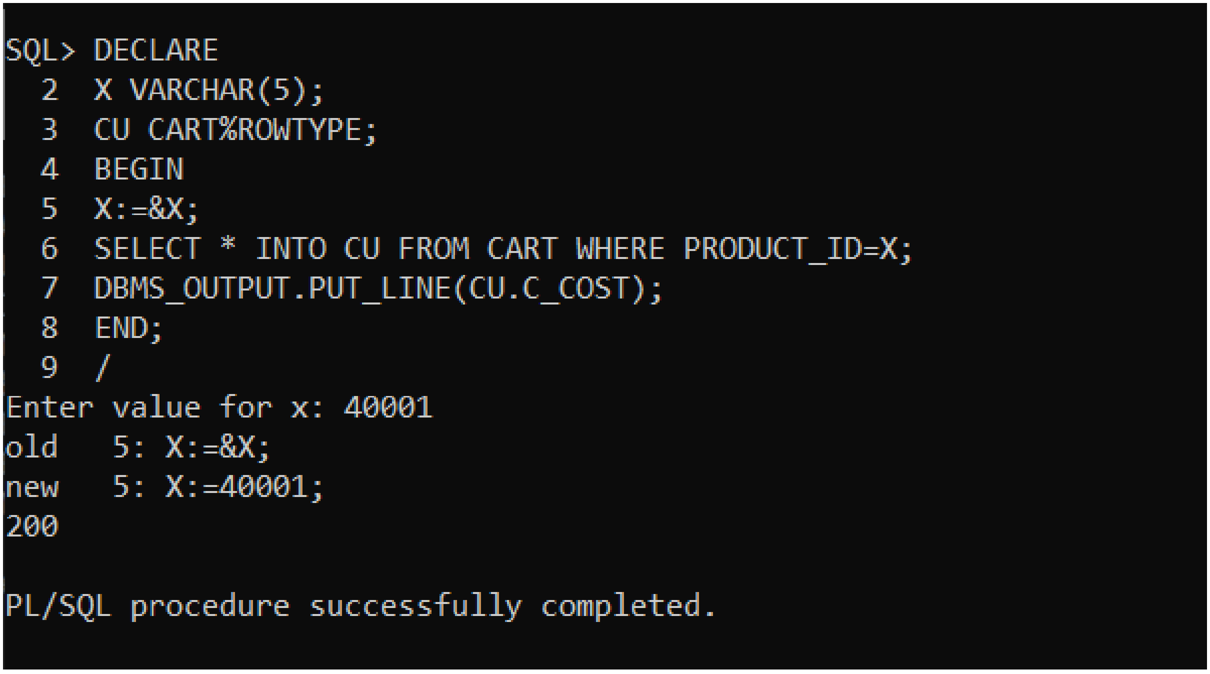
# PL/SQL Procedure and functions

## Finding supplier id who is producing particular product with the help of product id



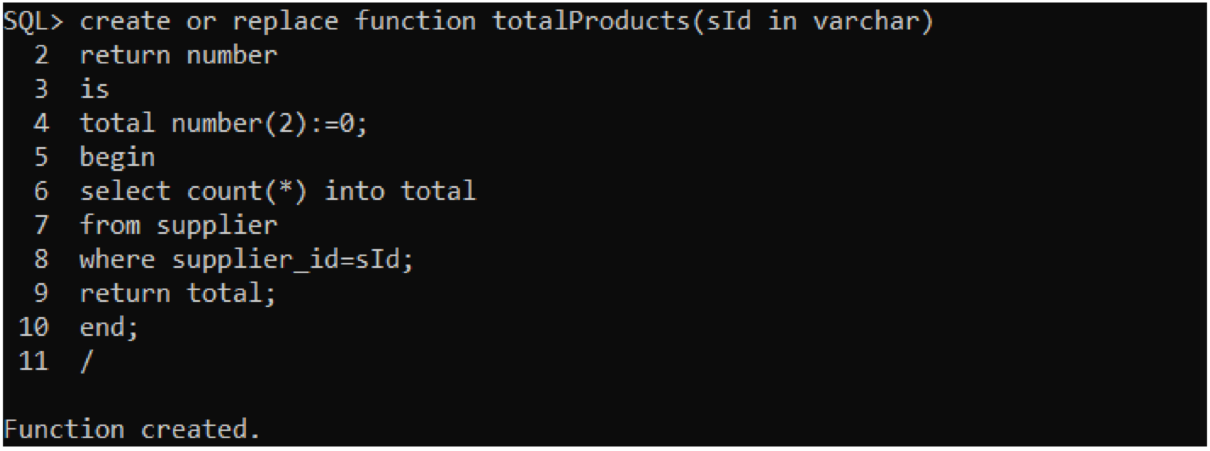
```
SQL> DECLARE
  2  PRO_ID VARCHAR(5):=’&PRO_ID’;
  3  CURSOR OR_CUR IS SELECT SUPPLIER_ID FROM SUPPLIER WHERE PRODUCT_ID=PRO_ID;
  4  OR_REC OR_CUR%ROWTYPE;
  5  BEGIN
  6  OPEN OR_CUR;
  7  LOOP
  8  FETCH OR_CUR INTO OR_REC;
  9  EXIT WHEN OR_CUR%notfound;
 10  DBMS_OUTPUT.PUT_LINE(OR_REC.SUPPLIER_ID);
 11  END LOOP;
 12  CLOSE OR_CUR;
 13  END;
 14  /
Enter value for pro_id: 40000
old   2: PRO_ID VARCHAR(5):='&PRO_ID';
new   2: PRO_ID VARCHAR(5):='40000';
50000

PL/SQL procedure successfully completed.
```

## Finding the cost of a particular product name

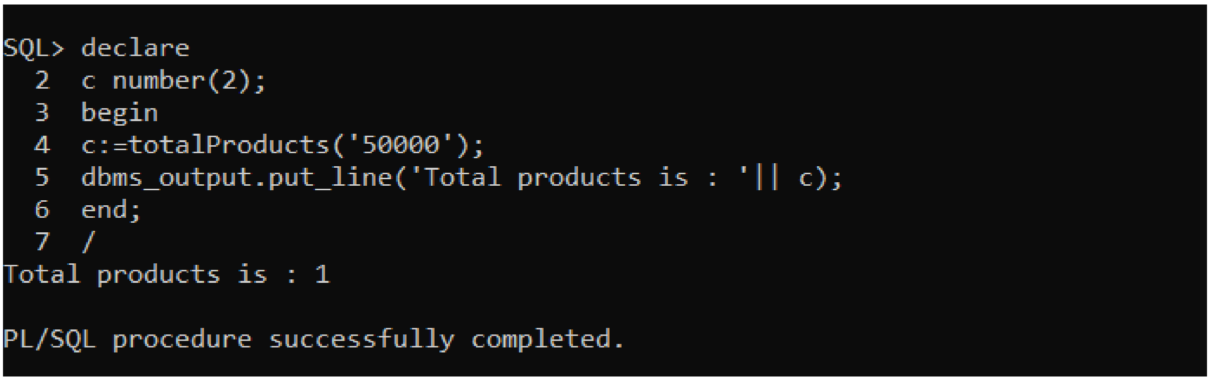


```
SQL> DECLARE
  2  X VARCHAR(5);
  3  CU CART%ROWTYPE;
  4  BEGIN
  5  X:=&X;
  6  SELECT * INTO CU FROM CART WHERE PRODUCT_ID=X;
  7  DBMS_OUTPUT.PUT_LINE(CU.C_COST);
  8  END;
  9  /
Enter value for x: 40001
old   5: X:=&X;
new   5: X:=40001;
200

PL/SQL procedure successfully completed.
```

Function to find total number of products a supplier sells



```
SQL> create or replace function totalProducts(sId in varchar)
return number
is
total number(2):=0;
begin
select count(*) into total
from supplier
where supplier_id=sId;
return total;
end;
/

Function created.
```

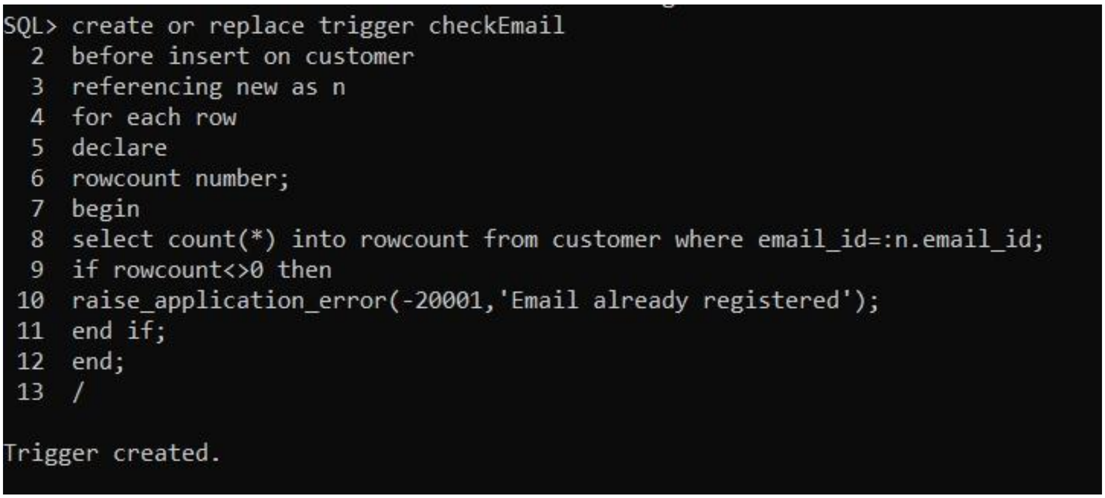


```
SQL> declare
c number(2);
begin
c:=totalProducts('50000');
dbms_output.put_line('Total products is : '|| c);
end;
/
Total products is : 1

PL/SQL procedure successfully completed.
```

# TRIGGERS

## Scenario:

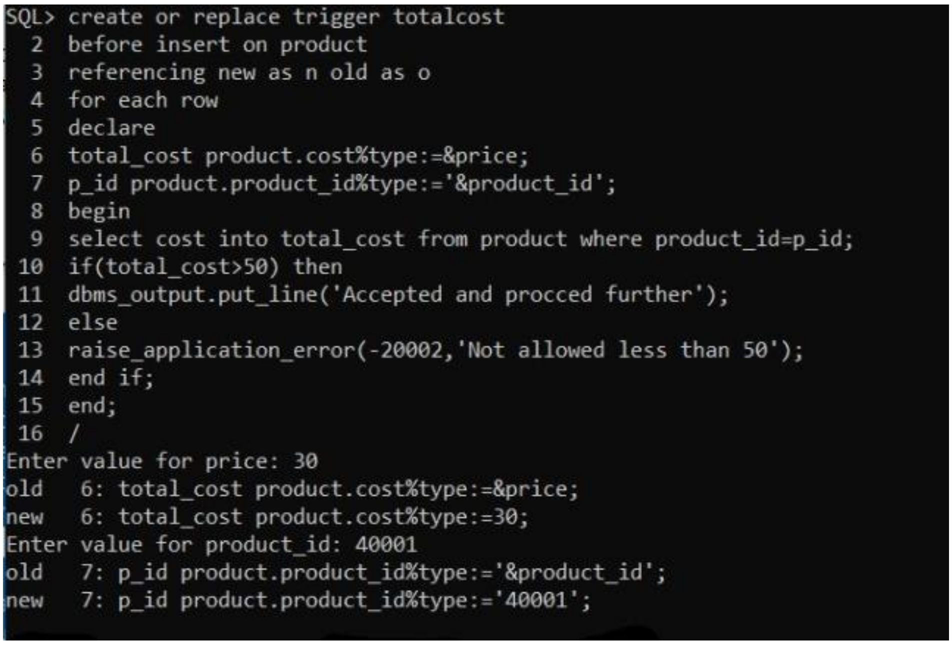
Two customers cannot have the same email id registered



```
SQL> create or replace trigger checkEmail
  2  before insert on customer
  3  referencing new as n
  4  for each row
  5  declare
  6  rowcount number;
  7  begin
  8  select count(*) into rowcount from customer where email_id=:n.email_id;
  9  if rowcount<>0 then
 10  raise_application_error(-20001,'Email already registered');
 11  end if;
 12  end;
 13  /

Trigger created.
```

Scenario:

Minimum cost of product must be greater than 50



```
SQL> create or replace trigger totalcost
  2  before insert on product
  3  referencing new as n old as o
  4  for each row
  5  declare
  6  total_cost product.cost%type:=&price;
  7  p_id product.product_id%type:='&product_id';
  8  begin
  9  select cost into total_cost from product where product_id=p_id;
 10  if(total_cost>50) then
 11  dbms_output.put_line('Accepted and procced further');
 12  else
 13  raise_application_error(-20002,'Not allowed less than 50');
 14  end if;
 15  end;
 16  /
Enter value for price: 30
old   6: total_cost product.cost%type:=&price;
new   6: total_cost product.cost%type:=30;
Enter value for product_id: 40001
old   7: p_id product.product_id%type:='&product_id';
new   7: p_id product.product_id%type:='40001';
```