DBMS PROJECT

E-commerce Management System

ABSTRACT

Electronic Commerce is a 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 the effort of a person to go & get products, ecommerce has made it easier for humans 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 e-commerce in developing countries for business applications.

In this modern era of online shopping no seller wants to be left behind, moreover due to its simplicity the shift from offline selling model to an online selling model is witnessing rampant growth.

Therefore, as an engineer our job is to ease the path of this transition for the seller. Amongst many things that an online site requires the most important is a database system. Hence in this project we are planning to design a database where sellers can sell their product online.

Also e-commerce will make it easy for customers to buy all the products that they need from a single website.

We will also provide them with all the basic access like order history,cart,payment and all the other various things expected out of an e-commerce website.

INTRODUCTION

Definition

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.

Working of E-COMMERCE

The consumer moves through the internet to the merchant's web site. From there, he decides that he wants to purchase something, so he is moved to the online transaction server, where all of the information he gives is encrypted. Once he has placed his order, the information moves through a private gateway to a Processing Network, where the issuing and acquiring banks complete or deny the transaction. This generally takes place in no more than 5-7seconds..

There are many different payment systems available to accommodate the varied processing needs of merchants, from those who have a few orders a day

to those who process thousands of transactions daily.With the addition of Secure Layer Technology,E-C0mmerce is also a very safe way to complete transactions

Functional Requirements

- A new user can register on the website.
- A customer can see details of the product present in the cart
- A customer can view his order history.
- Admin can start a sale with certain discount on every product.
- Customer can filter the product based on the product details.
- A customer can add or delete a product from the cart.
- A seller can unregister/ stop selling his product.
- A seller/ customer can update his details.
- Admin can view the products purchased on particular date.
- Admin can view number of products sold on a particular date.
- A customer can view the total price of product present in the cart unpurchased.
- Admin can view details of customer who have not purchased anything.
- Admin can view total profit earned from the website.

Tables And Attributes

1. Customer : name,c_pass,address,phone_no,pincode,cart_id.
2. Product: type,color,size,cost,seller_id,quantity,commission,size,age_group,gender.
3. Payment : payment_date,Total_amount,payment_id.
4. Cart: Cart_id
5. Cart_item : Date_added,Quantity_wished,purchased
6. Seller : name, address, s_pass, seller_id, phone_no

WORK BREAKDOWN STRUCTURE

CART AND CART_ITEM

CUSTOMER

PAYMENT AND SELLER PHONE NUM

PRODUCT AND SELLER

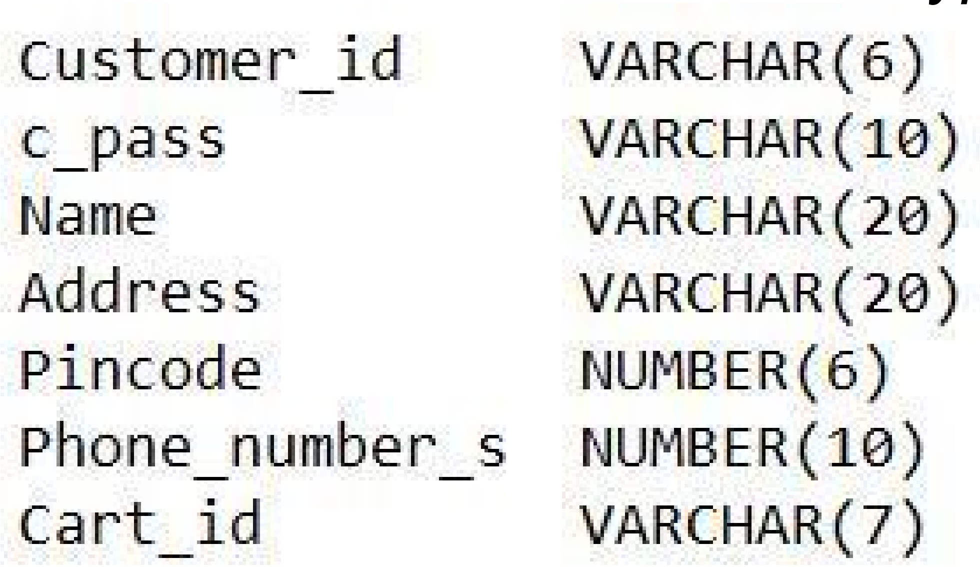
CART TABLE SCHEMA

ATTRIBUTES	***DATATYPE***
Cart_id	VARCHAR(7)

CART_ITEM SCHEMA

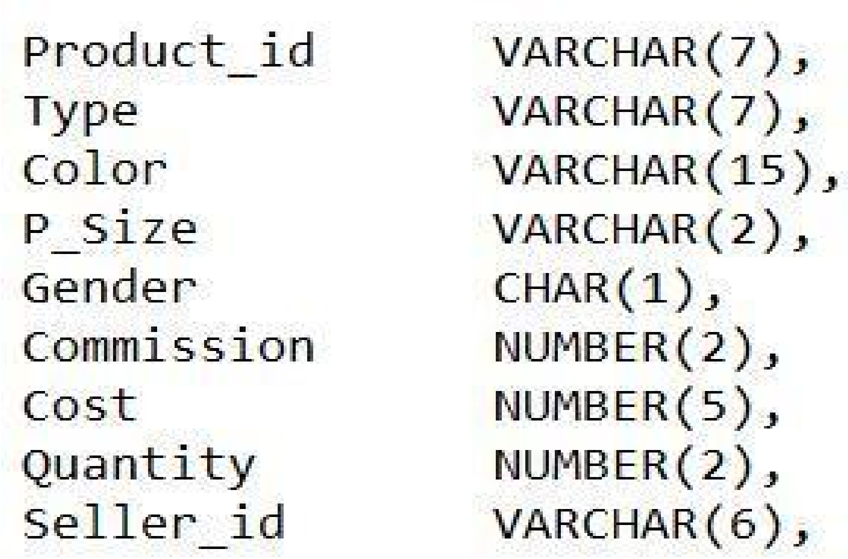
ATTRIBUTES	***DATATYPE***
Quantity_wished	NUMBER(1)
Date_Added	DATE
Cart_id	VARCHAR(7)
Product_id	VARCHAR(7)

Customer Table Schema



Attributes	*Datatypes*
Customer_id	VARCHAR(6)
c_pass	VARCHAR(10)
Name	VARCHAR(20)
Address	VARCHAR(20)
Pincode	NUMBER(6)
Phone_number_s	NUMBER(10)
Cart_id	VARCHAR(7)

Product Table Schema

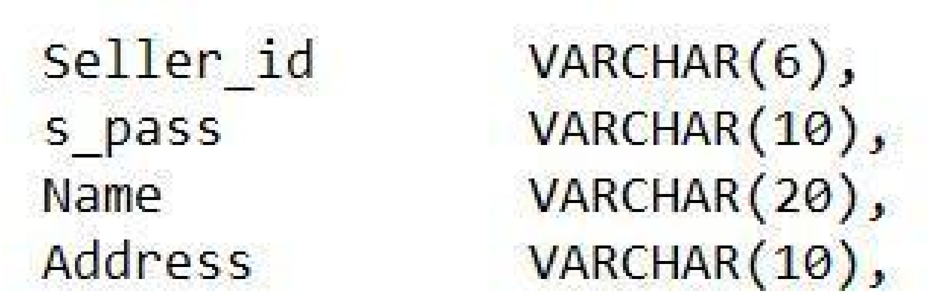


Attributes	*Datatype*
Product_id	VARCHAR(7),
Type	VARCHAR(7),
Color	VARCHAR(15),
P_Size	VARCHAR(2),
Gender	CHAR(1),
Commission	NUMBER(2),
Cost	NUMBER(5),
Quantity	NUMBER(2),
Seller_id	VARCHAR(6),

PAYMENT TABLE SCHEMA

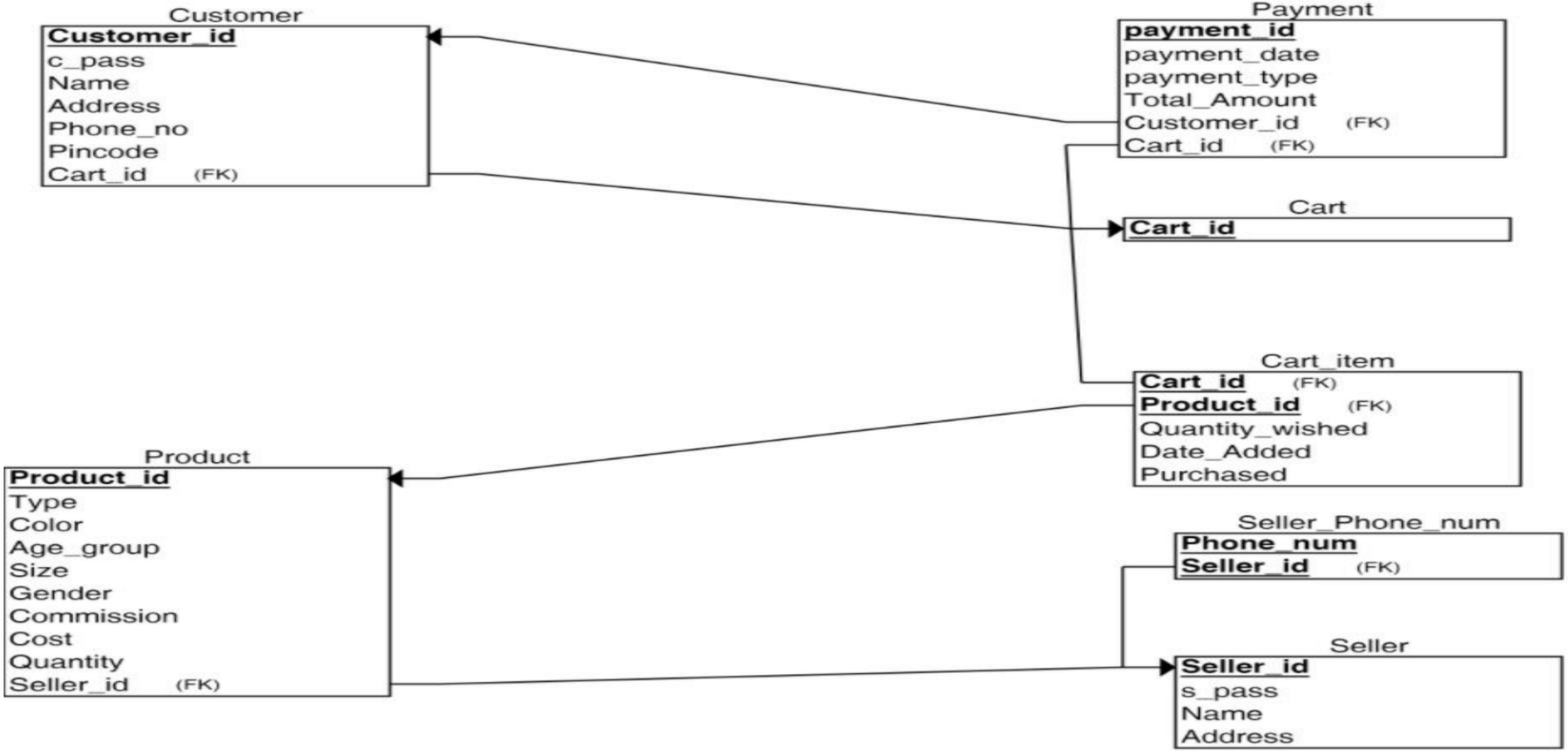
ATTRIBUTES	*DATATYPE*
payment_id	VARCHAR(7)
payment_date	DATE
Payment_type	VARCHAR(10)
Customer_id	VARCHAR(6)
Cart_id	VARCHAR(7)
total_amount	number(6)

Seller Table Schema



Attributes	*Datatypes*
Seller_id	VARCHAR(6),
s_pass	VARCHAR(10),
Name	VARCHAR(20),
Address	VARCHAR(10),

Relational Database Schema



Customer
Customer_id
c_pass
Name
Address
Phone_no
Pincode
Cart_id (FK)

Payment
payment_id
payment_date
payment_type
Total_Amount
Customer_id (FK)
Cart_id (FK)

Cart
Cart_id

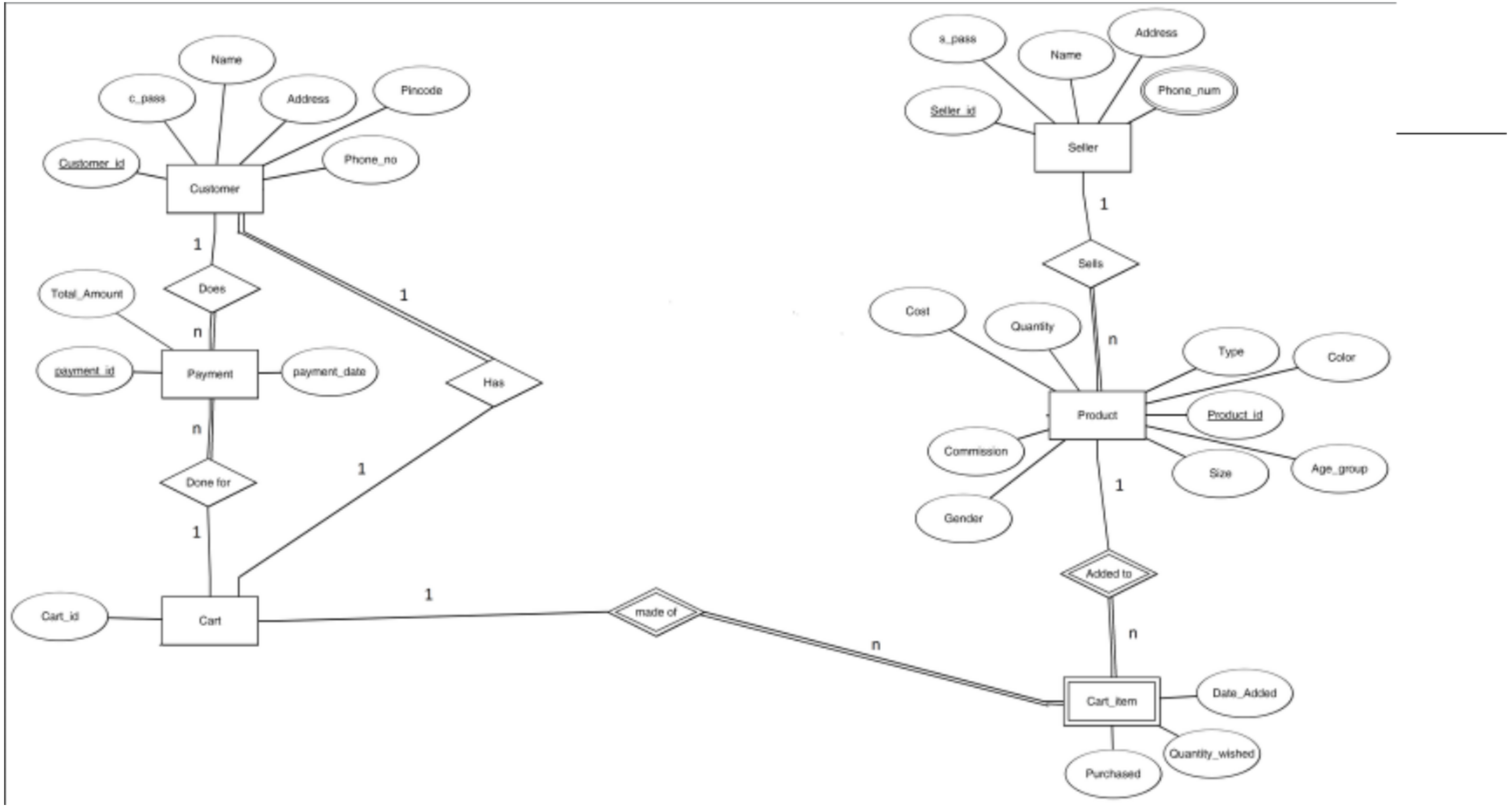
Cart_item
Cart_id (FK)
Product_id (FK)
Quantity_wished
Date_Added
Purchased

Product
Product_id
Type
Color
Age_group
Size
Gender
Commission
Cost
Quantity
Seller_id (FK)

Seller_Phone_num
Phone_num
Seller_id (FK)

Seller
Seller_id
s_pass
Name
Address

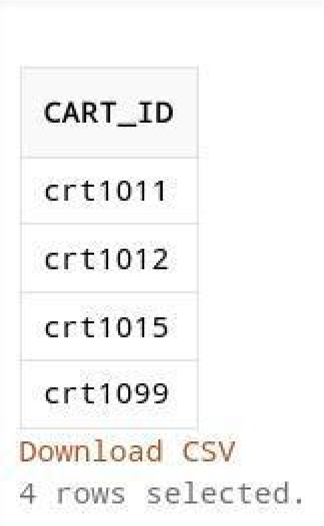
ER Diagram



Name
c_pass
Address
Pincode
Customer_id
Phone_no
Customer
1
Does
Total_Amount
n
payment_id
Payment
payment_date
n
Done for
1
Cart_id
Cart
1
Has
1
1
made of
n
s_pass
Name
Address
Seller_id
Phone_num
Seller
1
Sells
n
Cost
Quantity
Type
Color
Product
Product_id
Commission
Size
Age_group
Gender
1
Added to
n
Cart_item
Date_Added
Quantity_wished
Purchased

Tables And Table Values

1.Cart

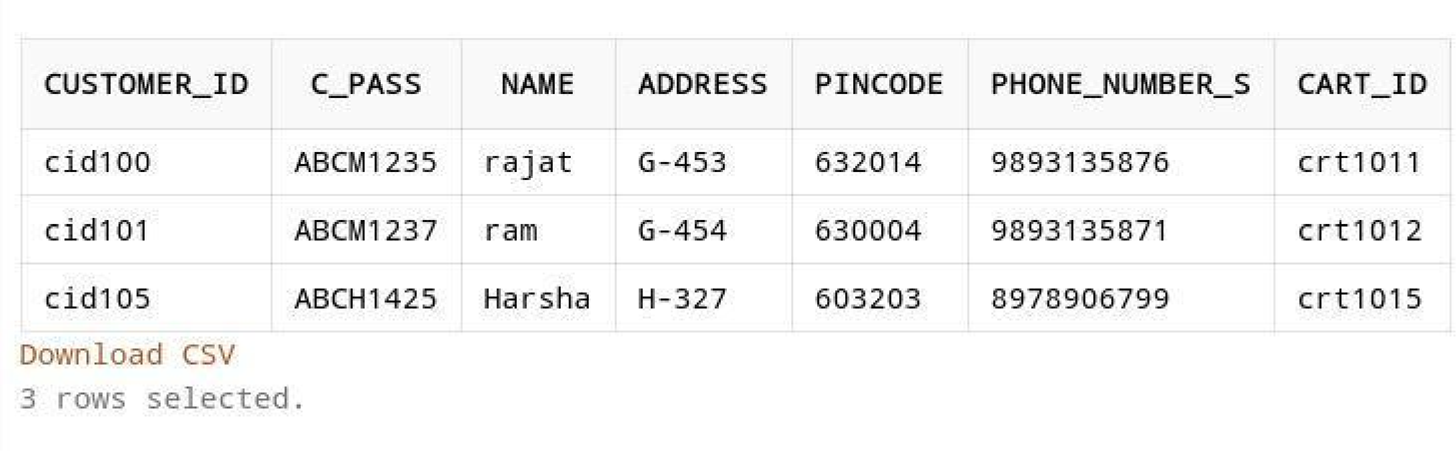


CART_ID
crt1011
crt1012
crt1015
crt1099

Download CSV

4 rows selected.

2.Customer

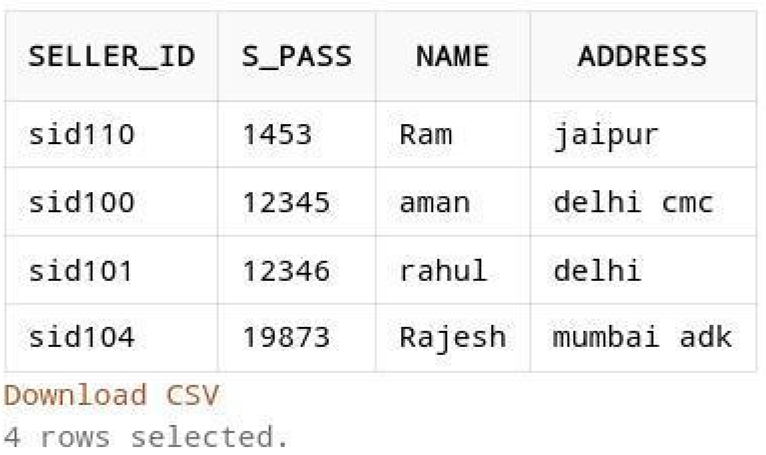


CUSTOMER_ID	C_PASS	NAME	ADDRESS	PINCODE	PHONE_NUMBER_S	CART_ID
cid100	ABCM1235	rajat	G-453	632014	9893135876	crt1011
cid101	ABCM1237	ram	G-454	630004	9893135871	crt1012
cid105	ABCH1425	Harsha	H-327	603203	8978906799	crt1015

Download CSV

3 rows selected.

3. Seller

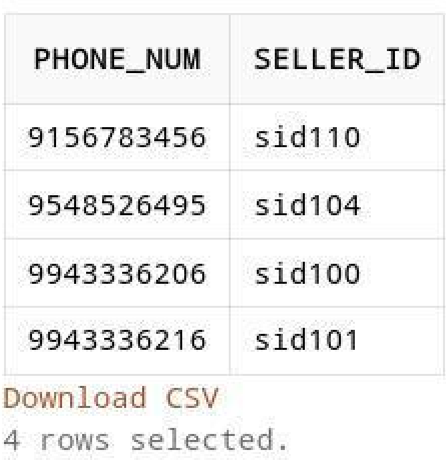


SELLER_ID	S_PASS	NAME	ADDRESS
sid110	1453	Ram	jaipur
sid100	12345	aman	delhi cmc
sid101	12346	rahul	delhi
sid104	19873	Rajesh	mumbai adk

Download CSV

```
4 rows selected.
```

4. Seller_Phone_num

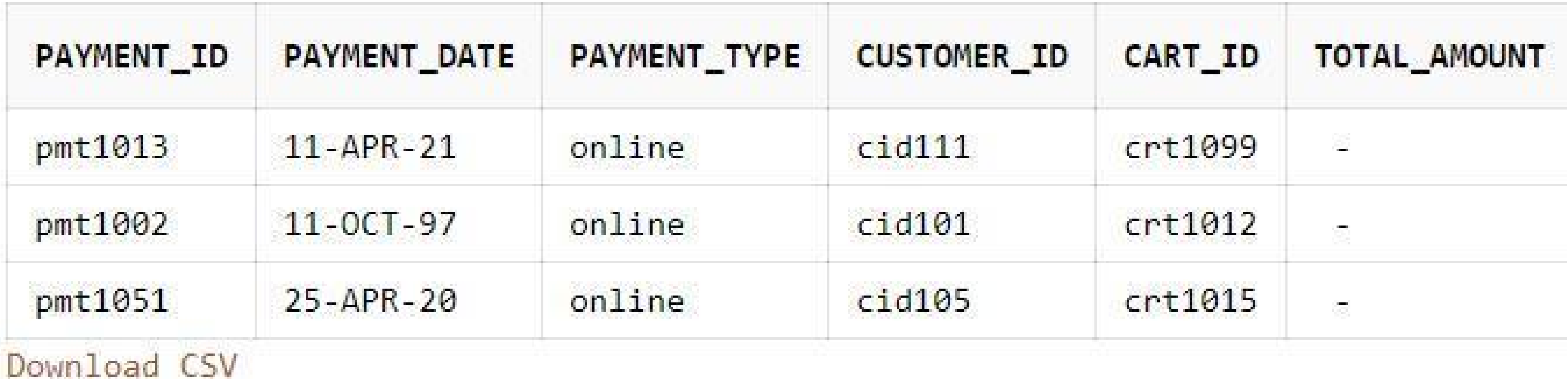


PHONE_NUM	SELLER_ID
9156783456	sid110
9548526495	sid104
9943336206	sid100
9943336216	sid101

Download CSV

4 rows selected.

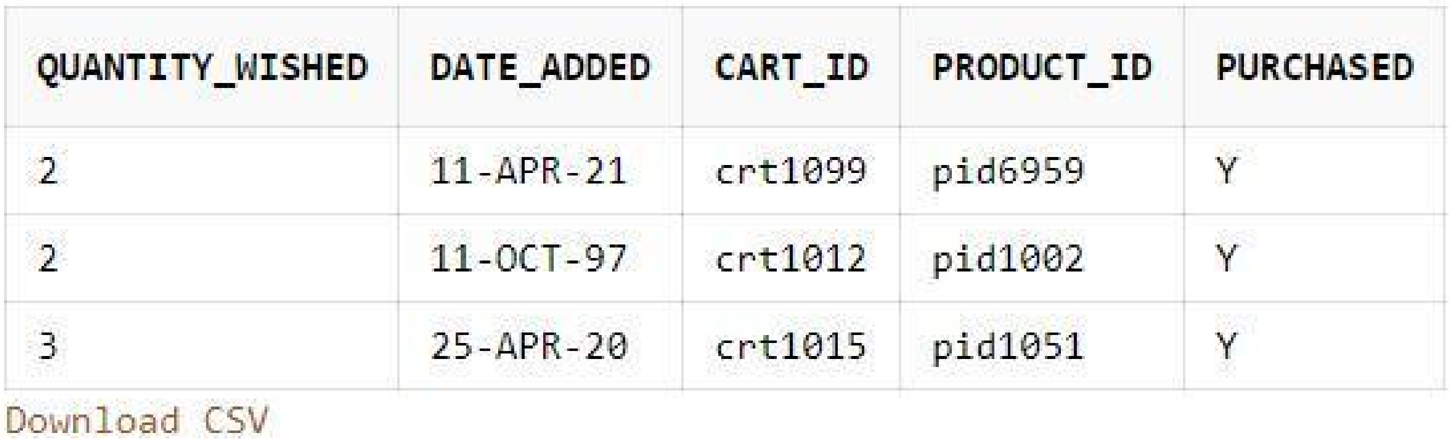
5 .PAYMENT TABLE VALUES



PAYMENT_ID	PAYMENT_DATE	PAYMENT_TYPE	CUSTOMER_ID	CART_ID	TOTAL_AMOUNT
pmt1013	11-APR-21	online	cid111	crt1099	-
pmt1002	11-OCT-97	online	cid101	crt1012	-
pmt1051	25-APR-20	online	cid105	crt1015	-

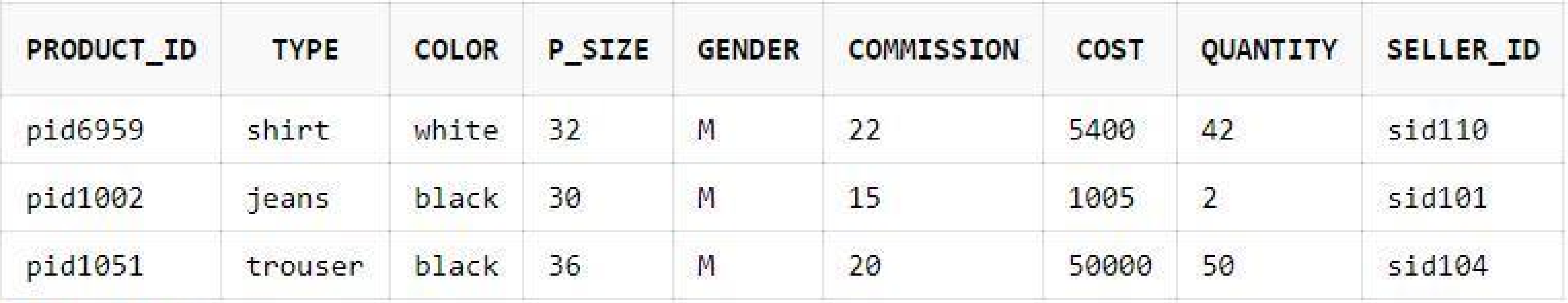
Download CSV

6.CART ITEMS TABLE VALUES



QUANTITY_WISHED	DATE_ADDED	CART_ID	PRODUCT_ID	PURCHASED
2	11-APR-21	crt1099	pid6959	Y
2	11-OCT-97	crt1012	pid1002	Y
3	25-APR-20	crt1015	pid1051	Y

7.PRODUCT TABLE VALUES



PRODUCT_ID	TYPE	COLOR	P_SIZE	GENDER	COMMISSION	COST	QUANTITY	SELLER_ID
pid6959	shirt	white	32	M	22	5400	42	sid110
pid1002	jeans	black	30	M	15	1005	2	sid101
pid1051	trouser	black	36	M	20	50000	50	sid104

Basic Queries

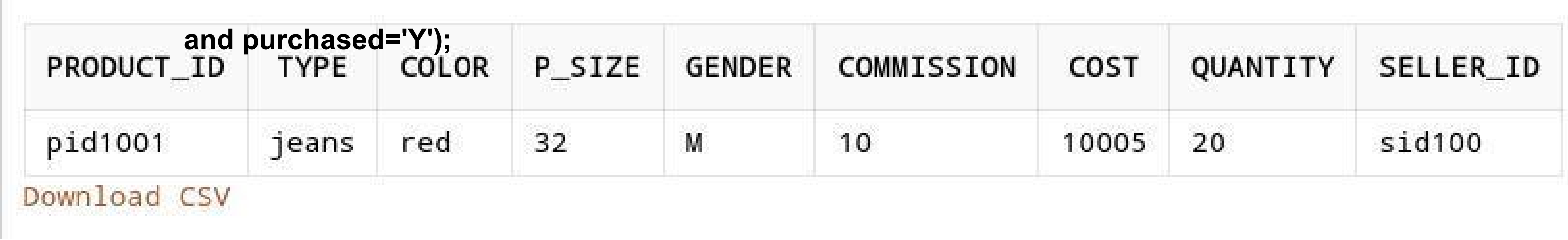
1.If the customer wants to see details of product purchased

code

```
select * from product where product_id in(
                    select product_id from Cart_item where (Cart_id in (
                    select Cart_id from Customer where Customer_id='cid100'
                    ))
```

output

```
                    and purchased='Y');
```



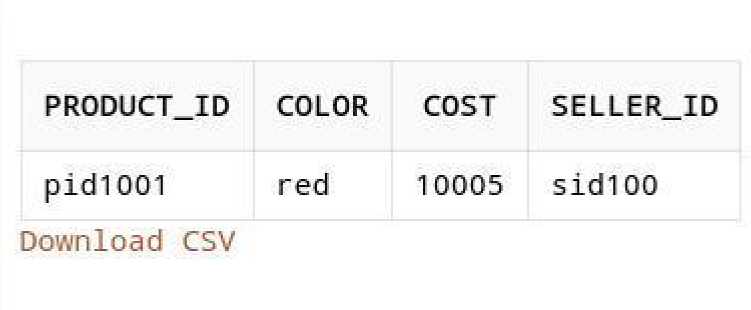
PRODUCT_ID	TYPE	COLOR	P_SIZE	GENDER	COMMISSION	COST	QUANTITY	SELLER_ID
pid1001	jeans	red	32	M	10	10005	20	sid100

3.Customer wants to see filtered product on basis of size,gender,type

Code:

```
select product_id, color, cost, seller_id from product where (type='jeans' and
p_size='32' and gender='F' and quantity>0)
```

Output:



PRODUCT_ID	COLOR	COST	SELLER_ID
pid1001	red	10005	sid100

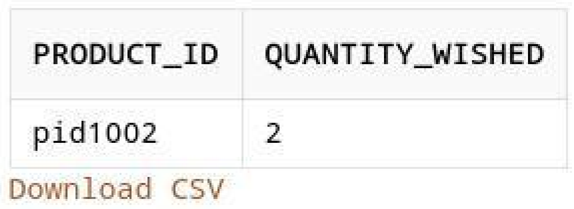
Download CSV

2.If a customer wants to see order history

Code

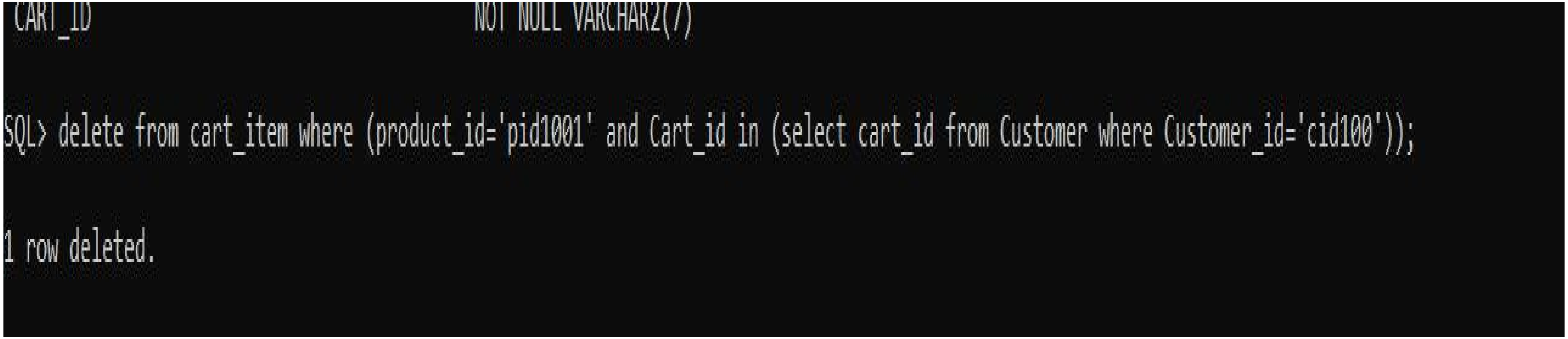
```
select product_id,Quantity_wished from Cart_item where (purchased='Y' and Cart_id in (select Cart_id from customer where Customer_id='cid101'));
```

Output



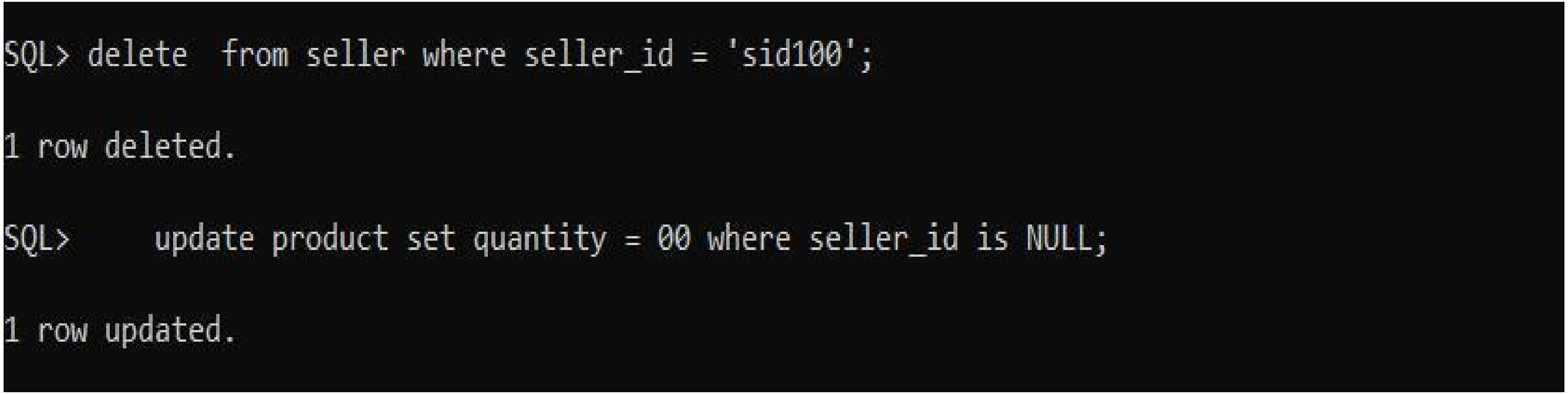
PRODUCT_ID	QUANTITY_WISHED
pid1002	2

If customer wants to modify the cart



```
 CART_ID                                    NOT NULL VARCHAR2(7)

SQL> delete from cart_item where (product_id='pid1001' and Cart_id in (select cart_id from Customer where Customer_id='cid100'));

1 row deleted.
```

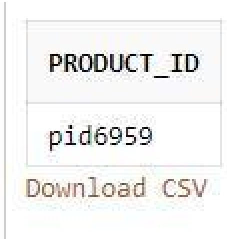
If a seller stops selling his product



```
SQL> delete  from seller where seller_id = 'sid100';

1 row deleted.

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

1 row updated.
```

If admin want to see what are the product purchased on the particular date

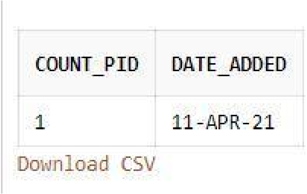
```
select product_id from cart_item where (purchased='Y' and
date_added='11-Apr-2021');
```



PRODUCT_ID
pid6959

How much product sold on the particular date

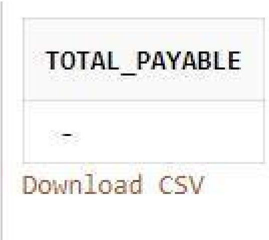
```
select count(product_id) count_pid,date_added from Cart_item
where purchased='Y' group by(date_added);
```



COUNT_PID	DATE_ADDED
1	11-APR-21

If a customer want to know the total price present in the cart

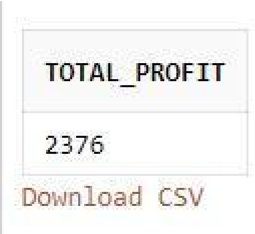
```
 select sum(quantity_wished * cost) total_payable from product p join cart_item
c on p.product_id=c.product_id where c.product_id in (select product_id from
cart_item where cart_id in(select Cart_id from customer where
customer_id='cid101') and purchased='Y');
```



TOTAL_PAYABLE
-

Find total profit of the website from sales.

```
select sum(quantity_wished * cost * commission/100) total_profit from product
p join cart_item c on p.product_id=c.product_id where purchased='Y';
```



TOTAL_PROFIT
2376

PL/SQL FUNCTIONS

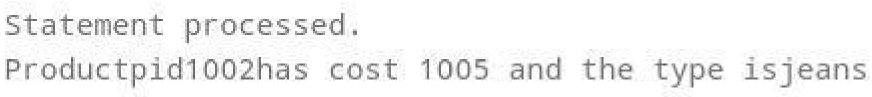
1.**Procedure which returns the type of product with the cost less than the given cost**

Code:

```
create or replace procedure cost_filter(c in number,t in varchar)
   is
    cs product.cost%type;
    ty product.type%type;
   id product.product_id%type;
   cursor cf is
   select product_id,cost,type from product where cost<c and type=t;
   begin
   open cf;
   loop
   fetch cf into id,cs,ty;
   exit when cf%notfound;
   dbms_output.put_line('Product' || id || 'has cost ' || cs || ' and the type is'
|| ty);
   end loop;
   close cf;
   exception
   when no_data_found then
   dbms_output.put_line('Sorry no such products
   exist'); end;
```

Output:

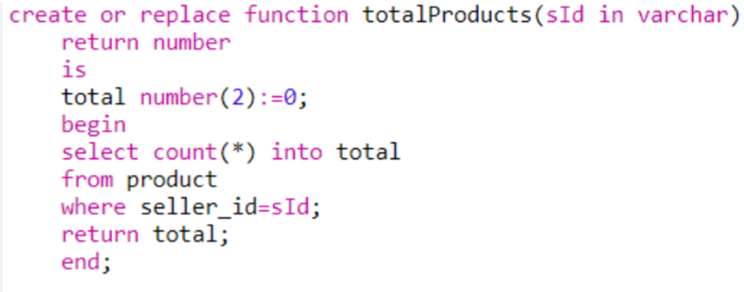
EXEC cost_filter(10000,'jeans');



```
Statement processed.
Productpid1002has cost 1005 and the type isjeans
```

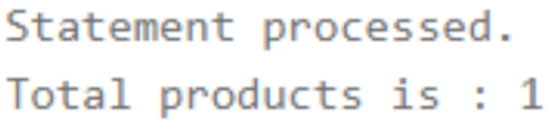
PL/SQL FUNCTIONS

2.Function which returns total number of products which a particular seller sells



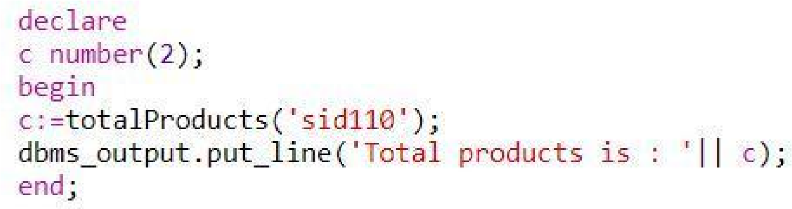
```
create or replace function totalProducts(sId in varchar)
    return number
    is
    total number(2):=0;
    begin
    select count(*) into total
    from product
    where seller_id=sId;
    return total;
    end;
```

OUTPUT



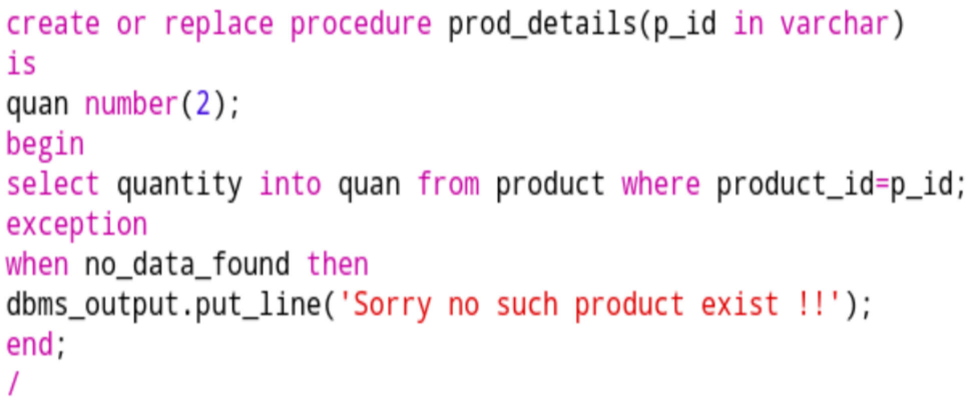
```
Statement processed.
Total products is : 1
```

Function Execution

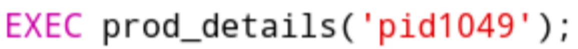


```
declare
c number(2);
begin
c:=totalProducts('sid110');
dbms_output.put_line('Total products is : '|| c);
end;
```

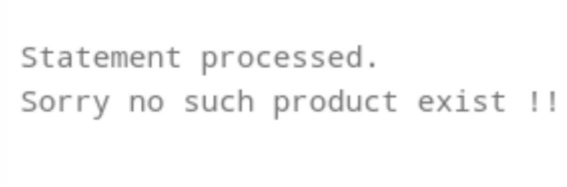
Procedure which returns the total quantity of product with the given ID



```
create or replace procedure prod_details(p_id in varchar)
is
quan number(2);
begin
select quantity into quan from product where product_id=p_id;
exception
when no_data_found then
dbms_output.put_line('Sorry no such product exist !!');
end;
/
```



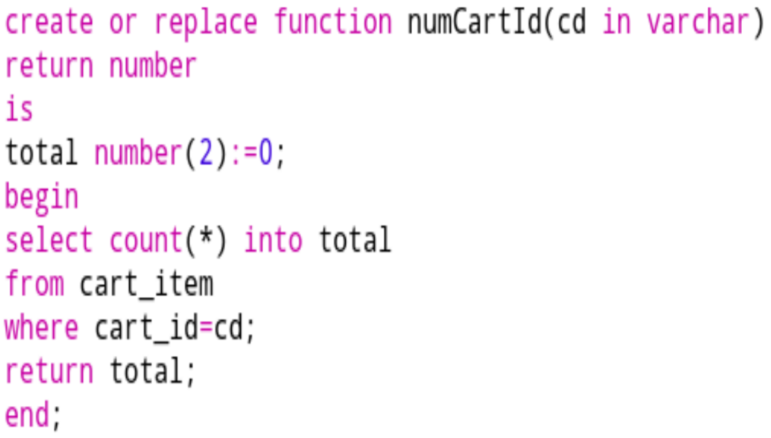
```
EXEC prod_details('pid1049');
```



```
Statement processed.
Sorry no such product exist !!
```

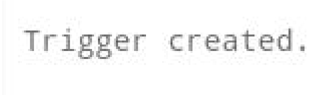
Trigger

1.**Trigger that will execute before inserting new customer to database and inserting a new cartId to the cart_items table**

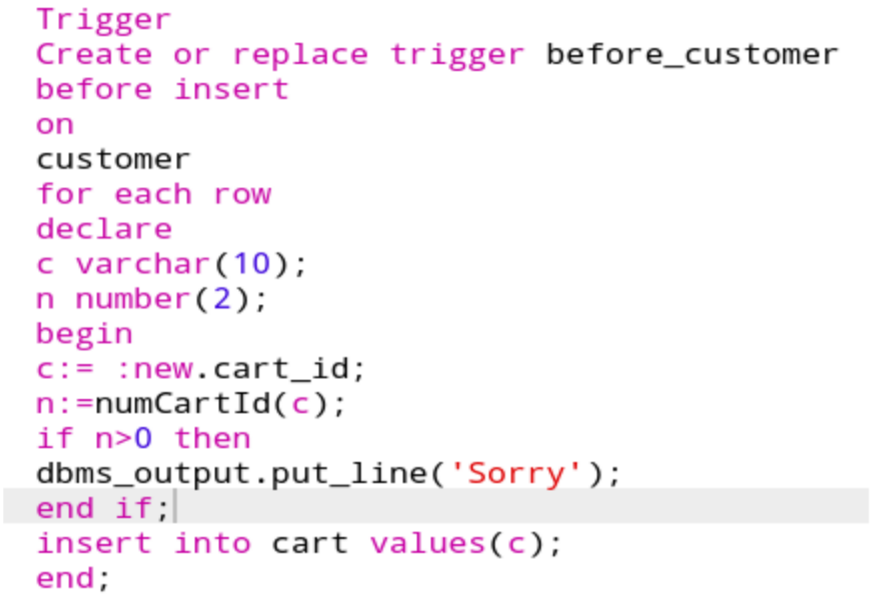


```
create or replace function numCartId(cd in varchar)
return number
is
total number(2):=0;
begin
select count(*) into total
from cart_item
where cart_id=cd;
return total;
end;
```

OUTPUT



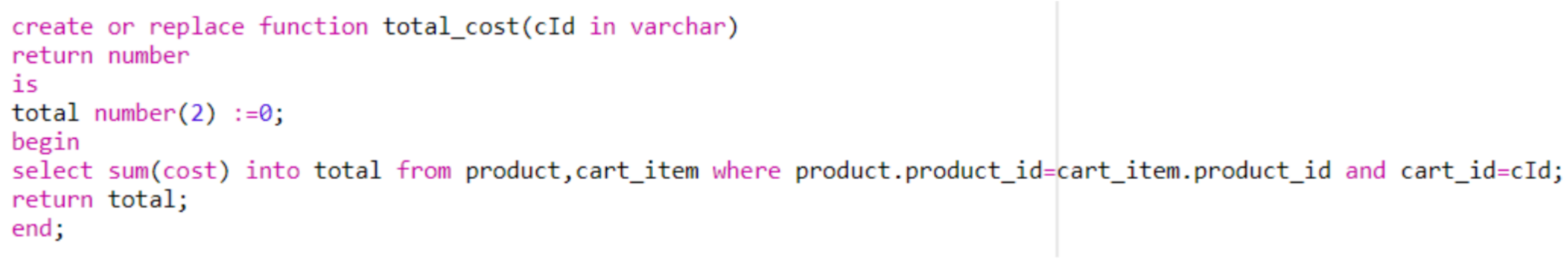
```
Trigger created.
```



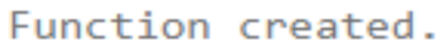
```
Trigger
Create or replace trigger before_customer
before insert
on
customer
for each row
declare
c varchar(10);
n number(2);
begin
c:= :new.cart_id;
n:=numCartId(c);
if n>0 then
dbms_output.put_line('Sorry');
end if;
insert into cart values(c);
end;
```

Trigger

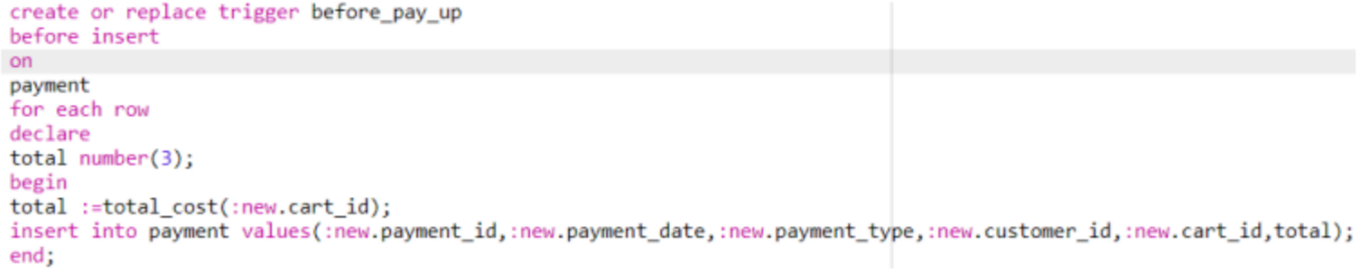
Trigger to update the total amount of user everytime he adds something to payment table



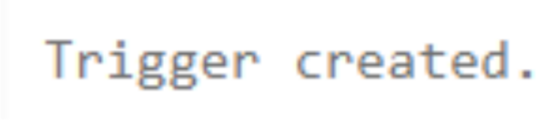
```
create or replace function total_cost(cId in varchar)
return number
is
total number(2) :=0;
begin
select sum(cost) into total from product,cart_item where product.product_id=cart_item.product_id and cart_id=cId;
return total;
end;
```



```
Function created.
```



```
create or replace trigger before_pay_up
before insert
on
payment
for each row
declare
total number(3);
begin
total :=total_cost(:new.cart_id);
insert into payment values(:new.payment_id,:new.payment_date,:new.payment_type,:new.customer_id,:new.cart_id,total);
end;
```



```
Trigger created.
```