A project Evaluation Report for
Database Management Systems(UCS310)

E-commerce Management

Submitted By:
Yash Jindal (102103474)
Arpit Dubey (102103463)
Garvit Madaan (102103471)
Navjot Kaur (102103455)

Submitted to:
Dr. Palak Goyal

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

DEPARTMENT OF CSED
THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY, (A DEEMED TO BE UNIVERSITY), PATIALA, PUNJAB INDIA

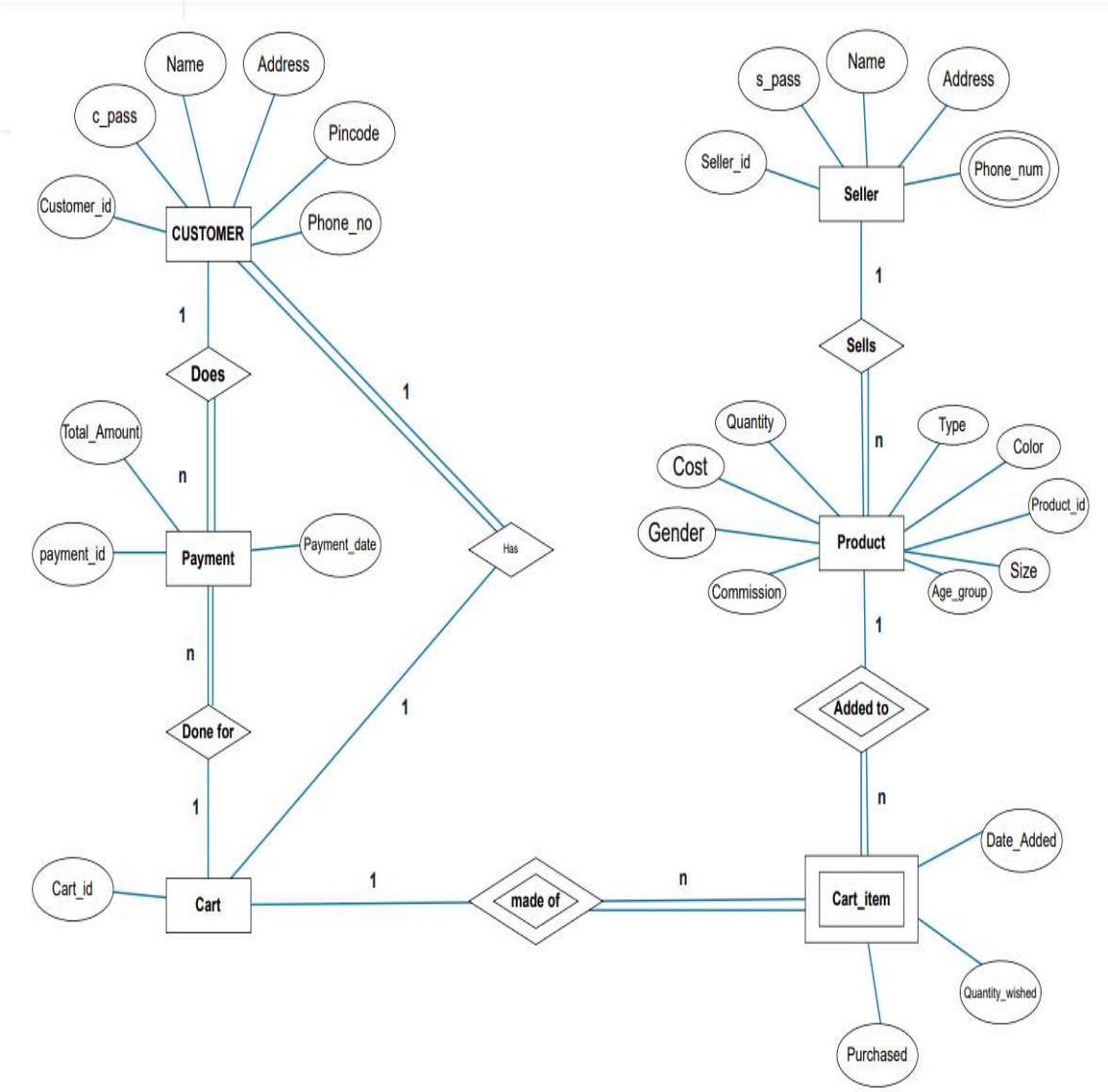
Jan-June 2023

Table of Contents

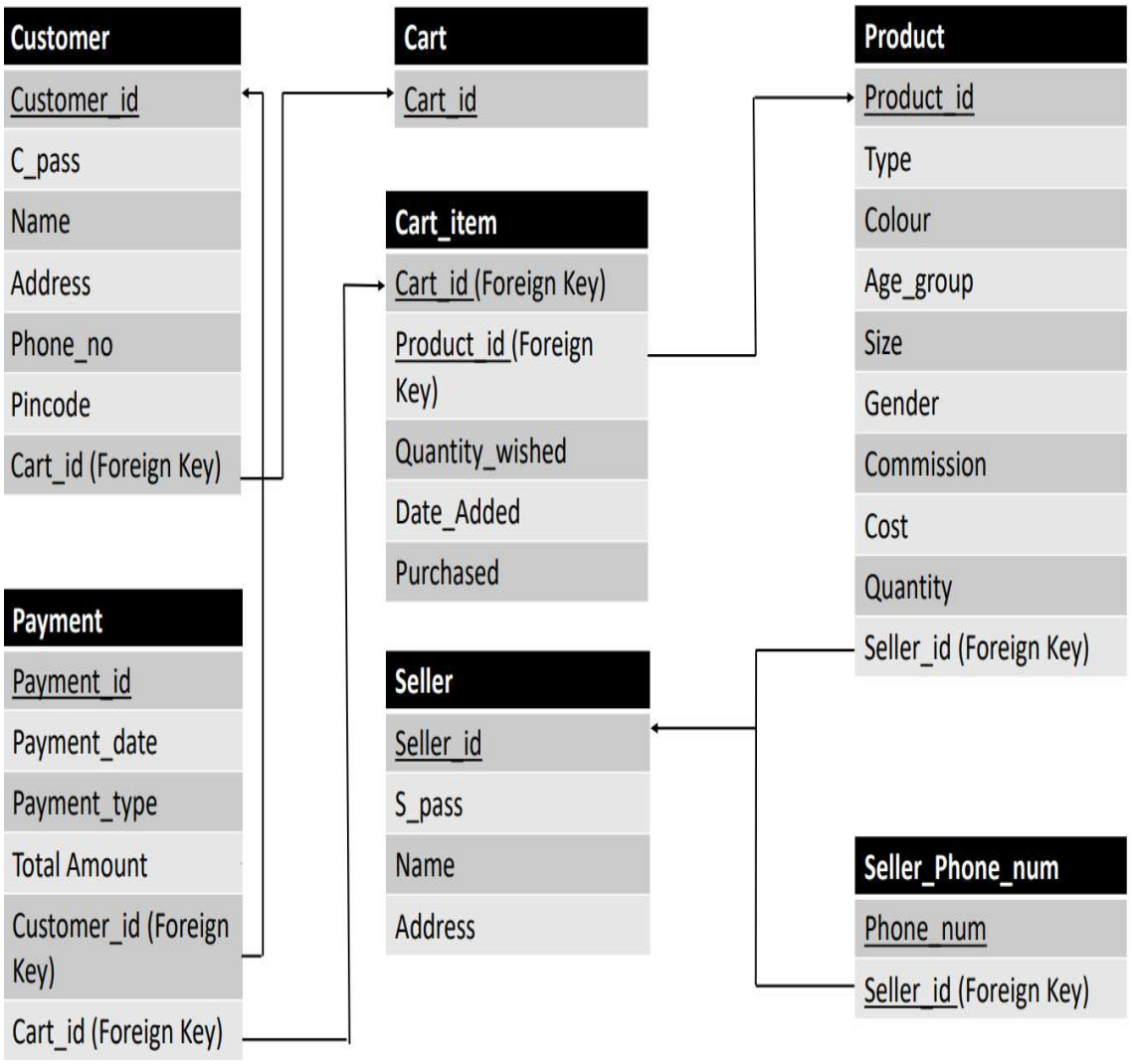
Mini World and Project Description

In this modern era of online shopping, no seller wants to be left behind, moreover, due to its simplicity the shift from an 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 small clothing sellers can sell their products online.

Entity Relation(ER) diagram



Relational Database Schema
(ER to tables)



Normalization

In database management systems (DBMS), normal forms are guidelines for designing and organizing database tables to minimize redundancy and maintain data integrity.

First Normal Form (1NF) is a basic requirement in database design that ensures that each table column contains only atomic values, which cannot be further subdivided. It is the first step in the normalization process, a technique used to organize data in a relational database.

TABLE SELLER_INFO

Column	Null?	Type
SELLER_ID	NOT NULL	VARCHAR2(6)
S_PASS	NOT NULL	VARCHAR2(10)
NAME	NOT NULL	VARCHAR2(20)
ADDRESS	NOT NULL	VARCHAR2(10)
PHONE_NUM	NOT NULL	NUMBER(10,0)

Download CSV

5 rows selected.

In the Table SELLER_INFO there is a multiple attribute named as Phone_num due to which this table is not in 1NF. To convert this table to 1NF Table is broken into 2 atomic tables which are shown below.

Created Atomic Units:

TABLE SELLER

Column	Null?	Type
SELLER_ID	NOT NULL	VARCHAR2(6)
S_PASS	NOT NULL	VARCHAR2(10)
NAME	NOT NULL	VARCHAR2(20)
ADDRESS	NOT NULL	VARCHAR2(10)

Download CSV

TABLE SELLER_PHONE_NUM

Column	Null?	Type
PHONE_NUM	NOT NULL	NUMBER(10,0)
SELLER_ID	NOT NULL	VARCHAR2(6)

Download CSV

Other tables are in :

- **1NF** as they do not contain any composite or multi-valued attribute.
- **2NF** as no non-primary key attribute is partially dependent on the primary key
- **3NF** as no non-key attribute is transitively dependent on the primary key.

Creating Tables

```
CREATE TABLE Cart
(
    Cart_id VARCHAR(7) NOT NULL,
    PRIMARY KEY(Cart_id)
);
```

TABLE CART

Column	Null?	Type
CART_ID	NOT NULL	VARCHAR2(7)

Download CSV

```
CREATE TABLE Seller
(
    Seller_id VARCHAR(6) NOT NULL,
    s_pass VARCHAR(10) NOT NULL,
    Name VARCHAR(20) NOT NULL,
    Address VARCHAR(10) NOT NULL,
    PRIMARY KEY (Seller_id)
);
```

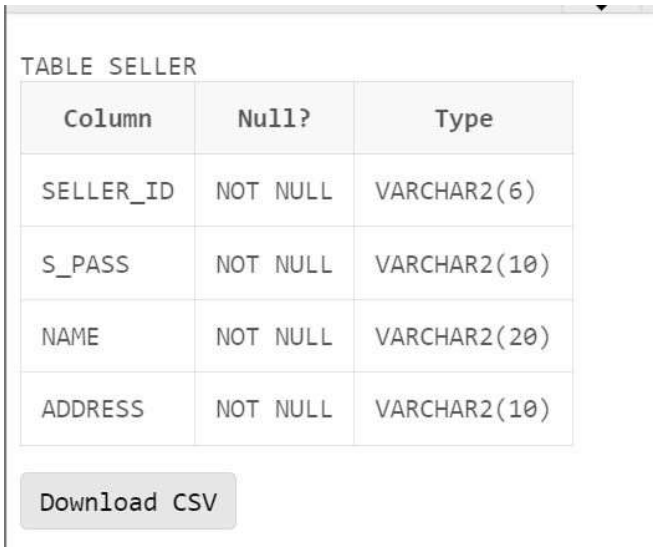


TABLE SELLER

Column	Null?	Type
SELLER_ID	NOT NULL	VARCHAR2(6)
S_PASS	NOT NULL	VARCHAR2(10)
NAME	NOT NULL	VARCHAR2(20)
ADDRESS	NOT NULL	VARCHAR2(10)

Download CSV

```
CREATE TABLE Customer
(
    Customer_id VARCHAR(6) NOT NULL,
    c_pass VARCHAR(10) NOT NULL,
    Name VARCHAR(20) NOT NULL,
    Address VARCHAR(20) NOT NULL,
    Pincode NUMBER(6) NOT NULL,
    Phone_number_s number(10) NOT NULL,
    PRIMARY KEY (Customer_id),
    Cart_id VARCHAR(7) NOT NULL,
    FOREIGN KEY(Cart_id) REFERENCES cart(Cart_id)
);
```

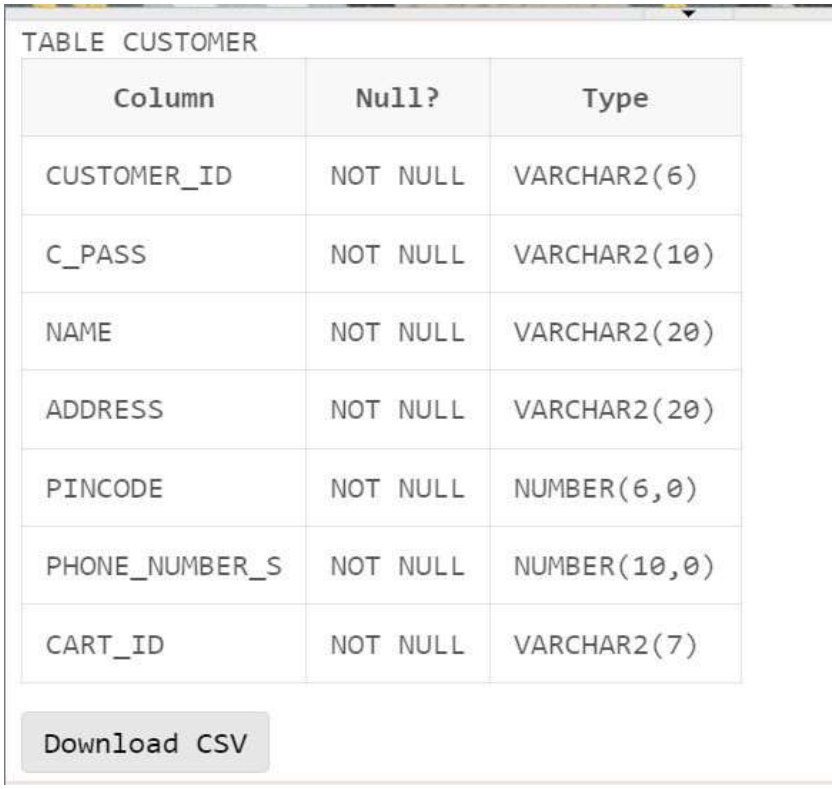


TABLE CUSTOMER

Column	Null?	Type
CUSTOMER_ID	NOT NULL	VARCHAR2(6)
C_PASS	NOT NULL	VARCHAR2(10)
NAME	NOT NULL	VARCHAR2(20)
ADDRESS	NOT NULL	VARCHAR2(20)
PINCODE	NOT NULL	NUMBER(6,0)
PHONE_NUMBER_S	NOT NULL	NUMBER(10,0)
CART_ID	NOT NULL	VARCHAR2(7)

Download CSV

CREATE TABLE **Seller_Phone_num**

(

Phone_num NUMBER(10) NOT NULL,

Seller_id VARCHAR(6) NOT NULL,

PRIMARY KEY (Phone_num, Seller_id),

FOREIGN KEY (Seller_id) REFERENCES Seller(Seller_id)

ON DELETE CASCADE

);

TABLE SELLER_PHONE_NUM

Column	Null?	Type
PHONE_NUM	NOT NULL	NUMBER(10,0)
SELLER_ID	NOT NULL	VARCHAR2(6)

Download CSV

```
CREATE TABLE Payment
  (
    payment_id VARCHAR(7) NOT NULL,
    payment_date DATE NOT NULL,
    Payment_type VARCHAR(10) NOT NULL,
    Customer_id VARCHAR(6) NOT NULL,
    Cart_id VARCHAR(7) NOT NULL,
    PRIMARY KEY (payment_id),
    FOREIGN KEY (Customer_id) REFERENCES Customer(Customer_id),
    FOREIGN KEY (Cart_id) REFERENCES Cart(Cart_id),
    total_amount numeric(6)
  );
```

TABLE PAYMENT

Column	Null?	Type
PAYMENT_ID	NOT NULL	VARCHAR2(7)
PAYMENT_DATE	NOT NULL	DATE
PAYMENT_TYPE	NOT NULL	VARCHAR2(10)
CUSTOMER_ID	NOT NULL	VARCHAR2(6)
CART_ID	NOT NULL	VARCHAR2(7)
TOTAL_AMOUNT	-	NUMBER(6,0)

Download CSV

```
CREATE TABLE Product
  (
    Product_id VARCHAR(7) NOT NULL,
    Type VARCHAR(7) NOT NULL,
    Color VARCHAR(15) NOT NULL,
    P_Size VARCHAR(2) NOT NULL,
    Gender CHAR(1) NOT NULL,
    Commission NUMBER(2) NOT NULL,
    Cost NUMBER(5) NOT NULL,
    Quantity NUMBER(2) NOT NULL,
    Seller_id VARCHAR(6),
    PRIMARY KEY (Product_id),
    FOREIGN KEY (Seller_id) REFERENCES Seller(Seller_id)
     ON DELETE SET NULL
  );
```

TABLE PRODUCT

Column	Null?	Type
PRODUCT_ID	NOT NULL	VARCHAR2(7)
TYPE	NOT NULL	VARCHAR2(7)
COLOR	NOT NULL	VARCHAR2(15)
P_SIZE	NOT NULL	VARCHAR2(2)
GENDER	NOT NULL	CHAR(1)
COMMISSION	NOT NULL	NUMBER(2,0)
COST	NOT NULL	NUMBER(5,0)
QUANTITY	NOT NULL	NUMBER(2,0)
SELLER_ID	-	VARCHAR2(6)

```
CREATE TABLE Cart_item
(
   Quantity_wished NUMBER(1) NOT NULL,
   Date_Added DATE NOT NULL,
   Cart_id VARCHAR(7) NOT NULL,
   Product_id VARCHAR(7) NOT NULL,
   FOREIGN KEY (Cart_id) REFERENCES Cart(Cart_id),
   FOREIGN KEY (Product_id) REFERENCES Product(Product_id),
   Primary key(Cart_id,Product_id)
);

alter table Cart_item add purchased varchar(3) default 'NO';
```

TABLE CART_ITEM

Column	Null?	Type
QUANTITY_WISHED	NOT NULL	NUMBER(1,0)
DATE_ADDED	NOT NULL	DATE
CART_ID	NOT NULL	VARCHAR2(7)
PRODUCT_ID	NOT NULL	VARCHAR2(7)
PURCHASED	-	VARCHAR2(3)

Download CSV

Inserting Values

Inserting values in Cart table:

```
insert into Cart values('crt1011');
insert into Cart values('crt1012');
insert into Cart values('crt1013');
insert into Cart values('crt1014');
insert into Cart values('crt1015');
insert into Cart values('crt1016');
insert into Cart values('crt1017');
insert into Cart values('crt1018');
insert into Cart values('crt1019');
insert into Cart values('crt1020');
insert into Cart values('crt1021');
```

Inserting values in Customer Table:

```
insert into Customer
values('cid100','ABCM1235','rajat','G-453','632014',9854424845, 'crt1017');
insert into Customer
values('cid105','ABCM5298','garvit','G-567','632014',8765737723, 'crt1015');
insert into Customer
values('cid152','ABCM5893','garvit','h-789','985672',2983037419, 'crt1013');
insert into Customer
values('cid158','ABCM5847','yash','k-987','567453',4451617353, 'crt1011');
insert into Customer
values('cid245','ABCM1475','harshit','p-768','755421',8612262629, 'crt1021');
insert into Customer values('cid354','1256','arpit','g-567','632014',7105609455,
'crt1012');
```

insert into Customer values('cid654','258','garvit','h-987','345789',2775325567, 'crt1019');

insert into Customer values('cid758','3597','rajat','i-87','789534',2126568138, 'crt1020');
insert into Customer values('cid125','2587','navjot','o-981','978231',3202614672, 'crt1014');
insert into Customer values('cid025','2478','akshita','s-67','457992',6560837894, 'crt1016');
insert into Customer values('cid356','2574','lakshay','f-17','567435',7635620148, 'crt1018');

Inserting values in Seller:

insert into Seller values('sid100','272447','abhishek','a-908');
insert into Seller values('sid102','454243','luv','v-876');
insert into Seller values('sid104','706859','dwigt','s-301');
insert into Seller values('sid106','119150','shelby','z-301');
insert into Seller values('sid108','484148','tommy','c-301');
insert into Seller values('sid110','858649','randy','d-901');
insert into Seller values('sid112','824848','poly','e-706');
insert into Seller values('sid114','305906','simran','r-908');
insert into Seller values('sid116','710476','pranav','j-789');
insert into Seller values('sid118','892615','khyati','h-890');
insert into Seller values('sid120','334168','vineet','f-789');

Inserting Values in Seller_phone_num:

insert into Seller_Phone_num values('8700064719','sid102');
insert into Seller_Phone_num values('8700064789','sid102');
insert into Seller_Phone_num values('8700045695','sid106');

```
insert into Seller_Phone_num values('5286314785','sid108');
insert into Seller_Phone_num values('8700064719','sid110');
insert into Seller_Phone_num values('2546781235','sid102');
insert into Seller_Phone_num values('4567531598','sid104');
insert into Seller_Phone_num values('1524835678','sid112');
insert into Seller_Phone_num values('2548796851','sid118');
insert into Seller_Phone_num values('1245639875','sid116');
insert into Seller_Phone_num values('1478523698','sid108');
insert into Seller_Phone_num values('1245785555','sid114');
insert into Seller_Phone_num values('8700045695','sid100');
```

Inserting values in Product:

```
insert into Product values('pid1001','jeans','white','26','M',5,3700,92,'sid102');
insert into Product values('pid1002','t-shirt','black','28','M',2,2900,47,'sid108');
insert into Product values('pid1003','dress','green','36','F',3,10000,67,'sid104');
insert into Product values('pid1004','t-shirt','white','38','F',8,8200,53,'sid106');
insert into Product values('pid1005','shirt','yellow','24','M',4,5900,58,'sid102');
insert into Product values('pid1006','shorts','red','42','F',7,8400,15,'sid108');
insert into Product values('pid1007','shorts','green','34','F',6,10000,34,'sid110');
insert into Product values('pid1008','top','white','36','F',6,7300,22,'sid112');
insert into Product values('pid1009','pants','grey','32','M',8,9500,11,'sid114');
insert into Product values('pid1010','suit','indigo','28','M',26,9700,35,'sid116');
insert into Product values('pid1011','t-shirt','red','26','M',12,2400,45,'sid118');
insert into Product values('pid1012','suit','blue','30','F',5,5700,10,'sid100');
insert into Product values('pid1013','shirt','red','22','F',14,1500,33,'sid120');
```

Inserting values in Payment:

```
insert into Payment
values('pmt1001',to_date('10-OCT-2007','dd-mon-yyyy'),'cod','cid152','crt1013',
NULL);
insert into Payment
values('pmt1002',to_date('10-NOV-2007','dd-mon-yyyy'),'online','cid158','crt101
1',NULL);
insert into Payment
values('pmt1003',to_date('10-JAN-2007','dd-mon-yyyy'),'cod','cid245','crt1021',
NULL);
insert into Payment
values('pmt1004',to_date('10-OCT-2007','dd-mon-yyyy'),'online','cid354','crt101
2',NULL);
insert into Payment
values('pmt1005',to_date('14-NOV-2007','dd-mon-yyyy'),'online','cid152','crt101
3',NULL);
insert into Payment
values('pmt1006',to_date('12-FEB-2007','dd-mon-yyyy'),'online','cid758','crt102
0',NULL);
insert into Payment
values('pmt1007',to_date('13-MAR-2007','dd-mon-yyyy'),'cod','cid125','crt1014'
,NULL);
insert into Payment
values('pmt1008',to_date('19-MAY-2007','dd-mon-yyyy'),'cod','cid356','crt1018',
NULL);
insert into Payment
values('pmt1009',to_date('07-JUN-2007','dd-mon-yyyy'),'cod','cid125','crt1014',
NULL);
```

Basic Queries

1. **If a customer wants to see details of the product present in the cart**

```
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'
    ))
and purchased='NO');
```

```
219
220     select * from product where product_id in(
221         select product_id from Cart_item where (Cart_id in (
222             select Cart_id from Customer where Customer_id='cid100'
223         ))
224     );
```

PRODUCT_ID	TYPE	COLOR	P_SIZE	GENDER	COMMISSION	COST	QUANTITY	SELLER_ID
pid1003	dress	green	36	F	3	10000	67	sid104

2. **If a customer wants to see order history**

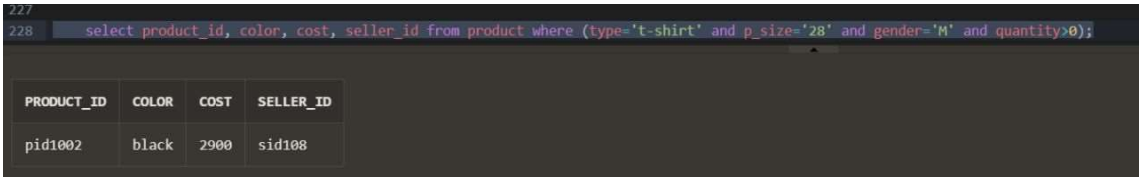
```
select product_id,Quantity_wished from Cart_item where
(purchased='Y' and Cart_id in (select Cart_id from customer where
Customer_id='cid152'));
```

```
226 select product_id,Quantity_wished from Cart_item where (purchased='N' and Cart_id in (select Cart_id from customer where Customer_id='cid152'));
```

PRODUCT_ID	QUANTITY_WISHED
pid1001	4
pid1006	1
pid1005	1

3. **Customer want to see filtered products on the basis of size, gender, type**

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

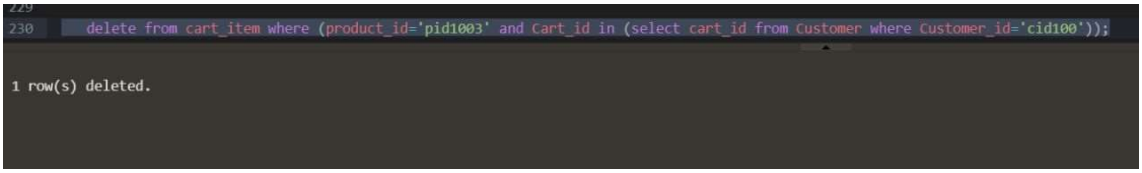


```
227
228     select product_id, color, cost, seller_id from product where (type='t-shirt' and p_size='28' and gender='M' and quantity>0);
```

PRODUCT_ID	COLOR	COST	SELLER_ID
pid1002	black	2900	sid108

4. **If customer want to modify the cart**

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



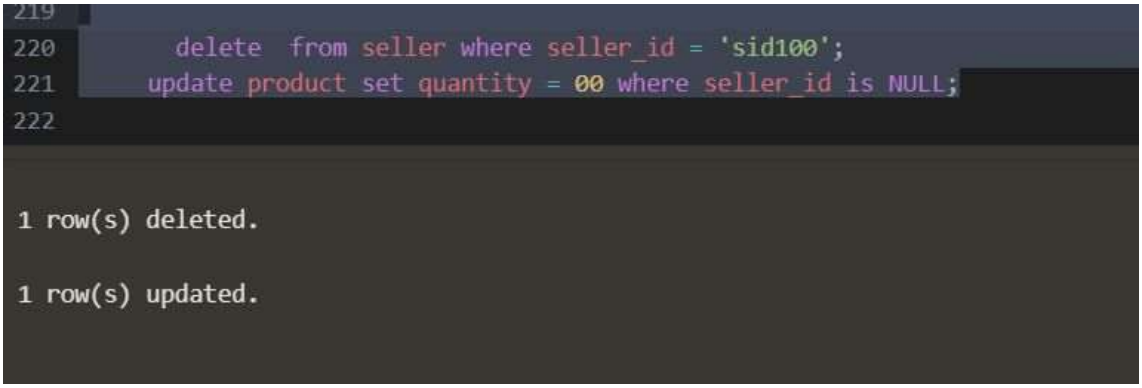
```
229
230     delete from cart_item where (product_id='pid1003' and Cart_id in (select cart_id from Customer where Customer_id='cid100'));

1 row(s) deleted.
```

5. **If a seller stop selling his product**

delete from seller where seller_id = 'sid100';

update product set quantity = 00 where seller_id is NULL;



```
219
220         delete  from seller where seller_id = 'sid100';
221       update product set quantity = 00 where seller_id is NULL;
222

1 row(s) deleted.

1 row(s) updated.
```

6. **If admin want to see what are the products added in the cart on a particular date**

   select product_id from cart_item where (purchased='N' and date_added='12-dec-2018');

```
234
235     select product_id from cart_item where (purchased='N' and date_added='17-feb-2007');
236
```

PRODUCT_ID
pid1003

7. **How much product was added in the cart on the particular date**

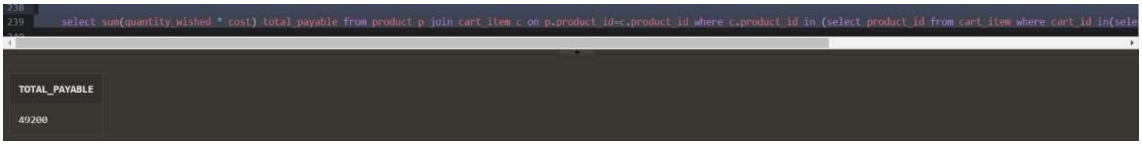
   select count(product_id) count_pid,date_added from Cart_item where purchased='N' group by(date_added);

```
237     select count(product_id) count_pid,date_added from Cart_item where purchased='N'  group by(date_added);
238
```

COUNT_PID	DATE_ADDED
1	08-MAR-07
1	09-APR-07
1	17-FEB-07
1	08-MAY-07
1	08-JUL-07
1	05-SEP-07
1	09-JUN-07
1	29-NOV-07
1	01-JAN-07

8. **If a customer wants to know the total price present in the cart**

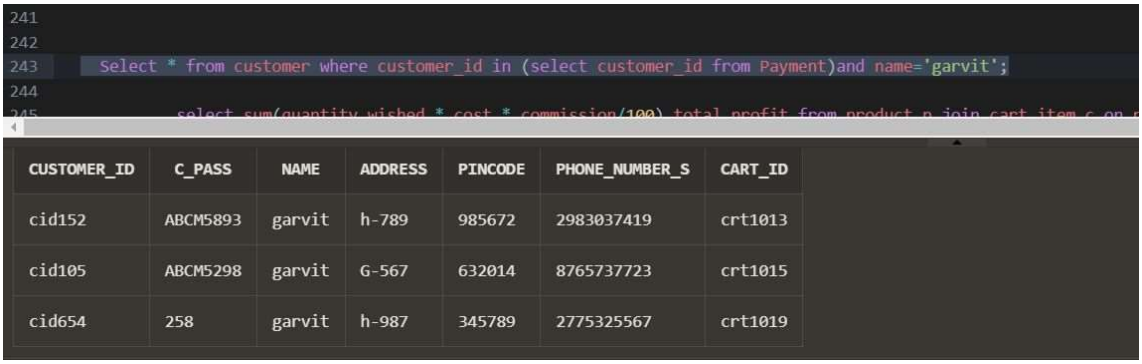
select sum(quantity_wished * cost) total_payable from product p join cart_item c on p.product_id=c.product_id where c.product_id in (select product_id from cart_item where cart_id in(select Cart_id from customer where customer_id='cid158') and purchased='N');



TOTAL_PAYABLE
49200

9. **Show the details of a specific customer who has purchased something**

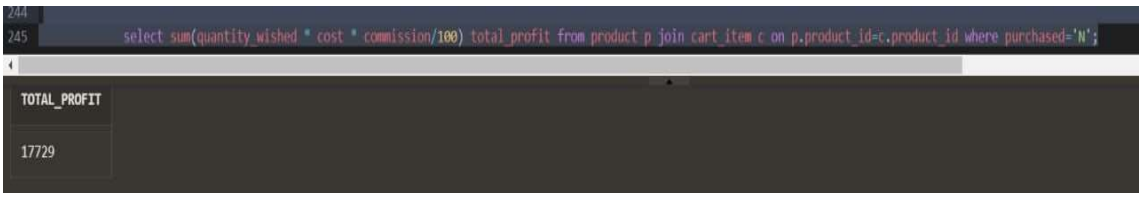
Select * from customer where customer_id in (select customer_id from Payment)and name ='garvit';



CUSTOMER_ID	C_PASS	NAME	ADDRESS	PINCODE	PHONE_NUMBER_S	CART_ID
cid152	ABCM5893	garvit	h-789	985672	2983037419	crt1013
cid105	ABCM5298	garvit	G-567	632014	8765737723	crt1015
cid654	258	garvit	h-987	345789	2775325567	crt1019

10. **Find total amount of product in all customers' cart**

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='N';



TOTAL_PROFIT
17729

PL/SQL Function

Procedure that returns the type of product with a cost less than the given cost

```
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;
```

```
Procedure created.
```

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;
```

```
Function created.
```

Function Execution:

```
declare
c number(2);
begin
c:=totalProducts('sid102');
dbms_output.put_line('Total products is : '|| c);
end;
```

```
Statement processed.
Total products is : 2
```

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

Procedure with Exceptional Handling

```
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;
```

```
Procedure created.
```

Triggers

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

Function to count number of cart items

```
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;
```

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 created.
```

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;
```

Trigger

```
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);
:new.total_amount := total;
end;
```

```
Trigger created.
```

Conclusion:

In conclusion, the e-commerce database management project successfully addressed the challenges of managing a large-scale online store's data and provided a reliable foundation for its operations. The developed database system effectively handled product inventory, order management, and customer information, enabling efficient processing of transactions.

The project's achievements include a well-structured database schema, a user-friendly interface, and essential features that contribute to an excellent user experience. The project team implemented various optimization techniques to enhance performance and ensure scalability as the business grows.

Overall, the e-commerce database management project has laid the groundwork for the successful operation of the online store, supporting its growth and enabling efficient management of critical business data.