PES UNIVERSITY

DATABASE AND MANAGEMENT SYSTEM
UE18CS252

DBMS PROJECT REPORT

TOPIC : e-shopping system

Name : DIVYA G
SRN : PES1201801663
Class : 4 B

INTRODUCTION:

E-shopping commonly known as Electronic Commerce refers to the buying and selling of information, products and services via computer networks. It has become clear that most of the existing retailers use the online service as an extension of the service that they provide in house. This allows them to provide another convenient element for the customer that might not be able to visit a store for the products that they require. Evidently retailers see their presence on the internet as an opportunity to further meet the needs of their customers employing the motto of ‘If you can’t come to us, we’ll come to you’.

DBMS plays a very important role in E-shopping. It reduces the load on both customers and retailers.

It contains the following tables:

1) user : which stores the user name name and id.

2) customer : stores the details of the user like phone number, email id,pincode etc.

3) address : stores the address of the user.

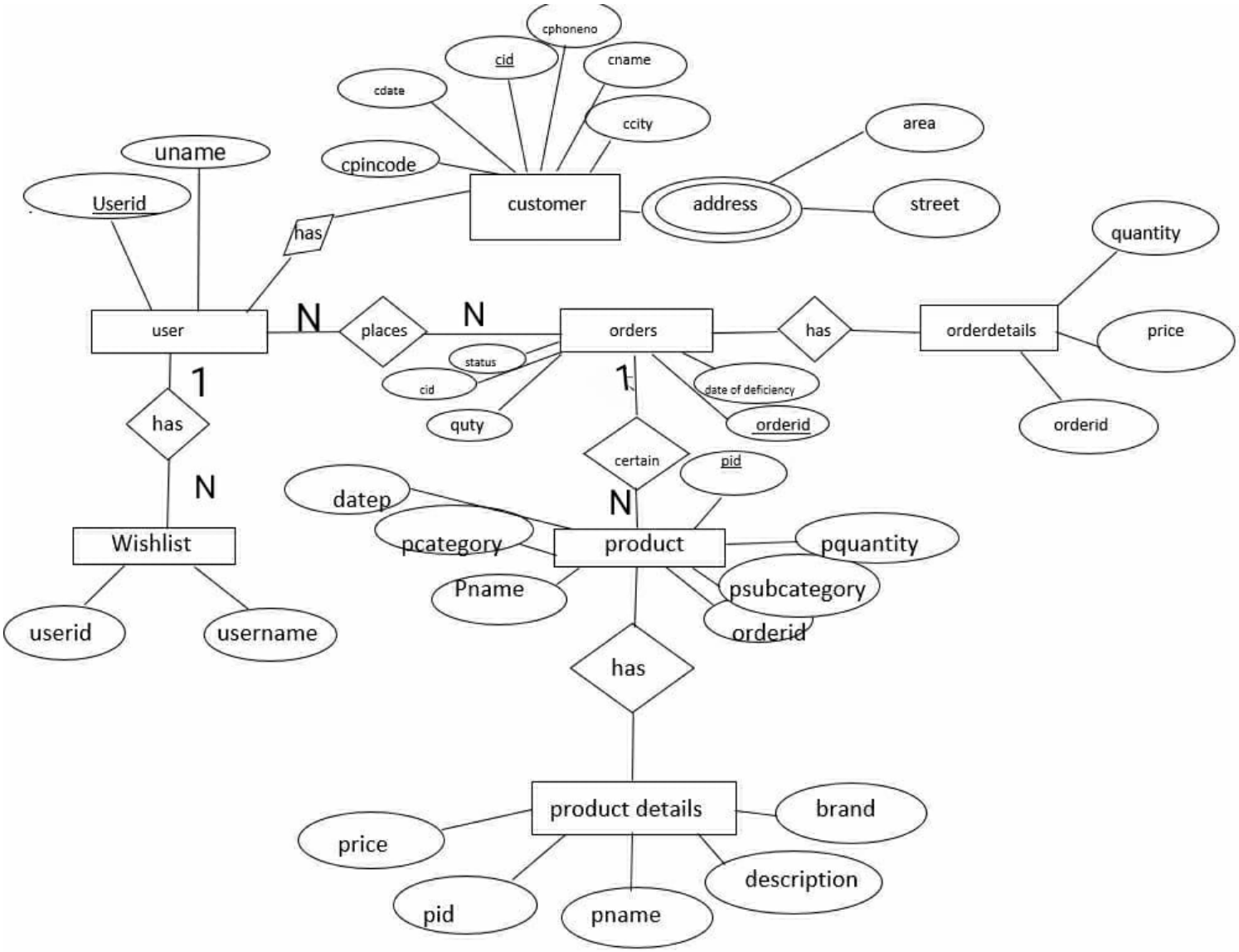
4) orders : stores customerid,date of delivery etc.

5) order_detail : stores the cost of the product, quantityof the product.

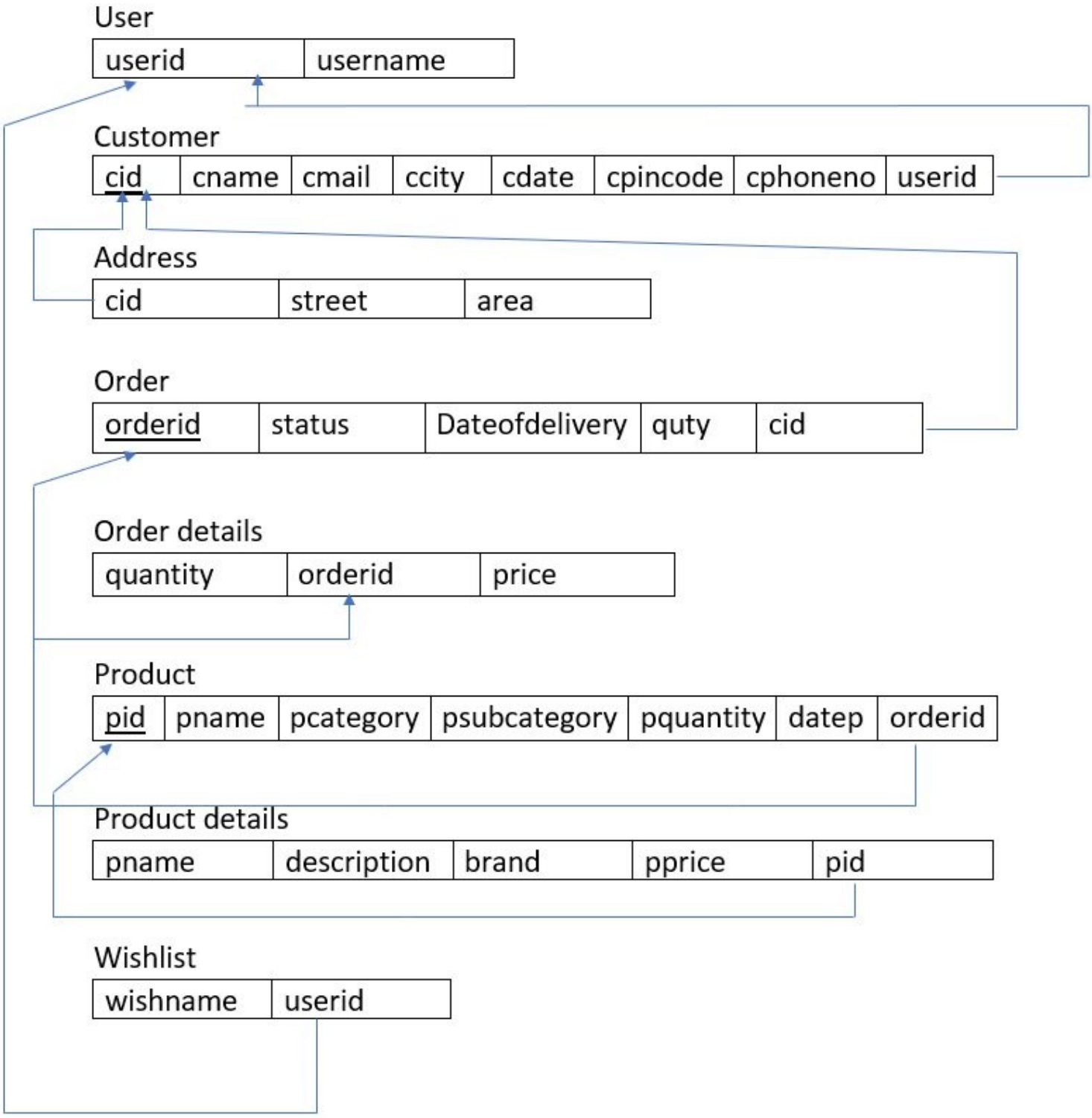
6)product : stores product name, category, sub category etc.

7) product_details : stores the description,brand of the product

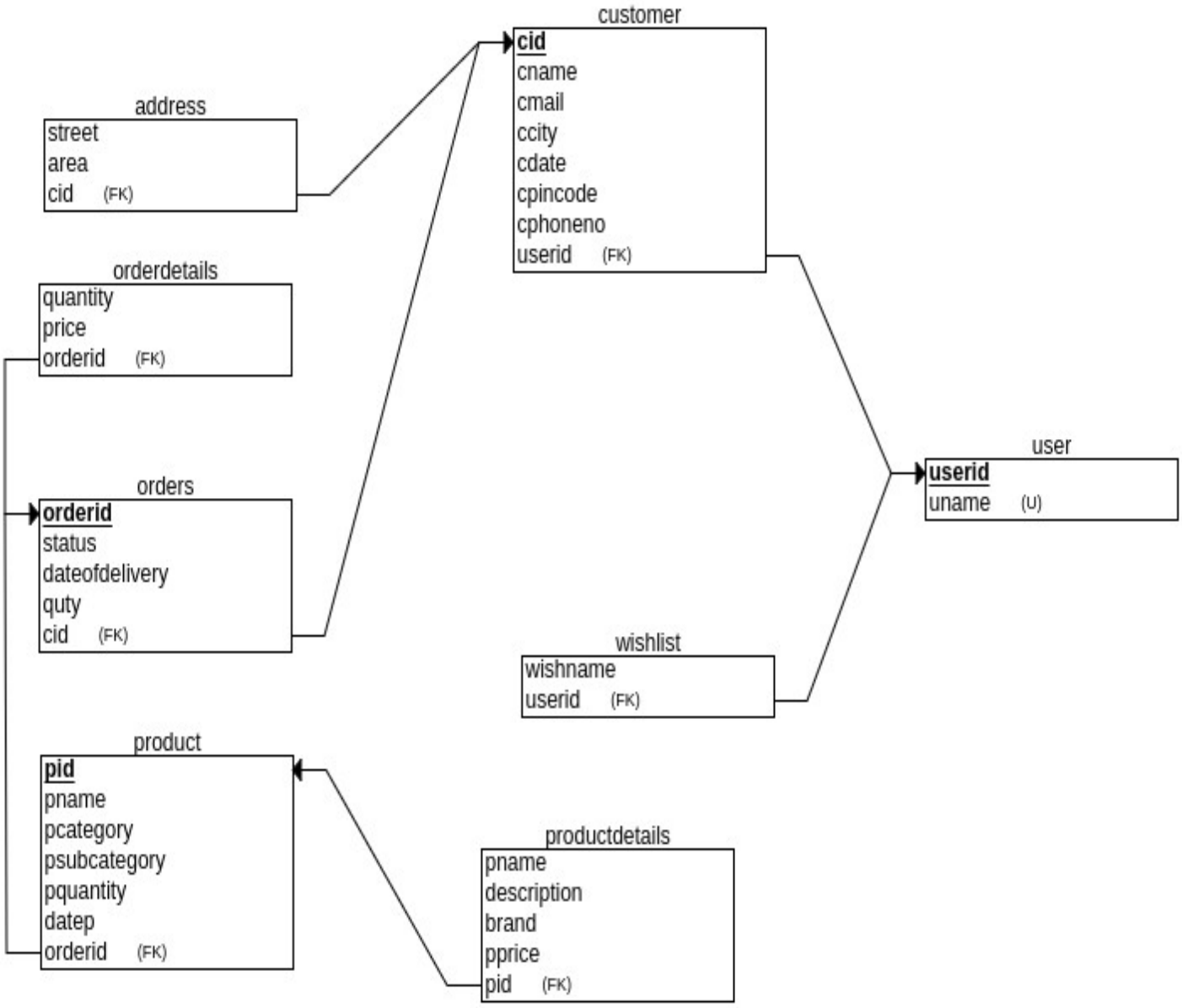
ER DIAGRAM:


cphoneno
cid
cname
cdate
ccity
area
uname
cpincode
customer
address
Userid
has
street
quantity
user
N
places
N
orders
has
orderdetails
price
status
cid
quty
date of deficiency
orderid
orderid
1
1
has
certain
pid
N
N
datep
Wishlist
pcategory
product
pquantity
Pname
psubcategory
orderid
userid
username
has
product details
brand
price
description
pid
pname

REATIONAL SCHEMA



RELATIONAL SCHEMA:



I. IDENTIFYING FUNCTIONAL DEPENDENCIES

➔**FROM TABLE USERS**
userid -> {userid,uname} PRIMARY KEY

➔**FROM TABLE CUSTOMER**
cid -> {cid,cname,cmail,ccity,cdate,cpincode} PRIMARY KEY

➔**FROM TABLE ADDRESS**
cid -> {cid,street,area} FOREIGN KEY

➔**FROM TABLE ORDERS**
orderid -> {status,dateofdelivery,quty,orderid} PRIMARY KEY

➔**FROM TABLE ORDERDETAILS**
orderid -> {orderid,price,quantity} FOREIGN KEY

➔**FROM TABLE PRODUCT**
pid -> {pid,pname,pcategory,psubcategory,pquantity,datep} PRIMARY KEY

➔**FROM TABLE PRODUCTDETAILS**
pid -> {pid,pname,description,brand,pprice} FOREIGN KEY

➔**FROM TABLE WISHLIST**
userid -> {userid,wishname} FOREIGN KEY

II. IDENTIFYING KEYS OF RELATIONAL BASED ON FD'S

CANDIDATE KEYS:

Tablename	Attribute
User	userid
customer	cid
orders	orderid
product	productid

- userid is candidate key for table user
- cid is candidate key for table customer
- orderid is candidate key for table orders
- product is candidate key for table product

III. NORMALIZATION AND TESTING FOR LOSSLESS JOIN PROPERTY :

NORMALIZATION:

- **1-NF :** Every cell should have atomic values.
- **2-NF :** Not even one functional dependency should be partial.
- **3-NF :** Not even single functional dependency should have transitive functional dependency.

But our schema violates 1-NF in table named customer, in which the address of the customer is multi-valued. This multi-valued entity is split into two tables as base table and referencing table where the referencing table will have the multivalued attribute with primary key as foreign key.

LOSSLESS JOIN PROPERTY:

If the relation R can be splited into 2 relationship R1 and R2, then the natural join of R1 and R2 can get back relation R again. Therefore, there is no loss of data or rows.

In our schema, consider table named customer and address and perform join operation on them.

customer(cid,cname,cmail,ccity,cpincode,cphoneno) and address(cid,street,area) will be a lossless join because there is an attribute which is common and unique. Therefore there won’t be any loss of information.

IV. CREATING THE TABLES INCLUDING VALUES

CREATING TABLE AND CHECKING REFERENTIAL CONSTRAINT:

```
CREATE TABLE user (
        userid INT NOT NULL,
        uname VARCHAR NOT NULL,
        PRIMARY KEY (userid),
        UNIQUE (uname)
 );

CREATE TABLE customer (
        cname CHAR NOT NULL,
        cmail VARCHAR NOT NULL,
        ccity CHAR NOT NULL,
        cdate DATE NOT NULL,
        cpincode VARCHAR NOT NULL,
        cphoneno VARCHAR NOT NULL,
        cid INT NOT NULL,
        userid INT NOT NULL,
        PRIMARY KEY (cid),
        FOREIGN KEY (userid) REFERENCES user(userid)
);

CREATE TABLE address (
        street VARCHAR NOT NULL,
        area VARCHAR NOT NULL,
        cid INT NOT NULL,
        FOREIGN KEY (cid) REFERENCES customer(cid)
);

CREATE TABLE orders (
        status CHAR NOT NULL,
        orderid INT NOT NULL,
        dateofdelivery DATE NOT NULL,
        quty INT NOT NULL,
        cid INT NOT NULL,
        PRIMARY KEY (orderid),
        FOREIGN KEY (cid) REFERENCES customer(cid)
);
```

```
CREATE TABLE orderdetails (
	quantity INT NOT NULL,
	price FLOAT NOT NULL,
	orderid INT NOT NULL,
	FOREIGN KEY (orderid) REFERENCES orders(orderid)
);

CREATE TABLE product (
	pid INT NOT NULL,
	pname VARCHAR NOT NULL,
	pcategory VARCHAR NOT NULL,
	psubcategory VARCHAR NOT NULL,
	pquantity INT NOT NULL,
	datep DATE NOT NULL,
	orderid INT NOT NULL,
	PRIMARY KEY (pid),
	FOREIGN KEY (orderid) REFERENCES orders(orderid)
);

CREATE TABLE productdetails (
	pname VARCHAR NOT NULL,
	description VARCHAR NOT NULL,
	brand VARCHAR NOT NULL,
	pprice FLOAT NOT NULL,
	pid INT NOT NULL,
	FOREIGN KEY (pid) REFERENCES product(pid)
);

CREATE TABLE wishlist (
	wishname VARCHAR NOT NULL,
	userid INT NOT NULL,
	FOREIGN KEY (userid) REFERENCES user(userid)
);
```

INSERTING VALUES:

```
INSERT into users values(1,'Akshatha');
INSERT into users values(2,'Pallavi');
INSERT into users values(3,'Harshitha');
INSERT into users values(4,'Akshatha01');
INSERT into users values(5,'Sonu');
INSERT into users values(6,'Bhavana');
INSERT into users values(7,'Harish');
INSERT into users values(8,'Bhuvan');
```

```
INSERT into users values(9,'Akash');
INSERT into users values(10,'Shreya');

INSERT INTO customer
values(1,111,'Akshatha','akshatha@gmail.com','Harihar','2020/03/15',577601,9632514
569);
INSERT INTO customer
values(2,112,'Pallavi','pallavi@gmail.com','Harihar','2020/02/10',577601,8563245698);
INSERT INTO customer
values(3,113,'Harshitha','harshitha@gmail.com','Harihar','2020/01/01',577601,6325896
412);
INSERT INTO customer
values(4,114,'Akshatha01','akshi123@gmail.com','Davangere','2019/05/23',577602,756
3256699);
INSERT INTO customer
values(5,115,'Sonu','sonusakki@gmail.com','Davangere','2019/09/03',577602,6254866
120);
INSERT INTO customer
values(6,116,'Bhavana','bhavana@gmail.com','Hubli','2018/04/25',580029,9445213354
);
INSERT INTO customer
values(7,117,'Harish','harishnaik@gmail.com','Mysuru','2019/10/16',570026,95481123
54);
INSERT INTO customer
values(8,118,'Bhuvan','bhuvanbhuvi@gmail.com','Bengaluru','2018/06/26',560098,987
5613341);
INSERT INTO customer
values(9,119,'Akash','akashhnh@gmail.com','Bengaluru','2018/06/26',560085,7215468
220);
INSERT INTO customer
values(10,120,'Shreya','shreyasheetal@gmail.com','Bengaluru','2019/10/05',560085,84
56213581);


INSERT INTO address values(111,'1st main 1st cross','Keshav nagar');
INSERT INTO address values(112,'1st main 1st cross','K R nagar');
INSERT INTO address values(113,'1st main 1st cross','Vidya nagar');
INSERT INTO address values(114,'2nd stage','Vinob nagar');
INSERT INTO address values(115,'1st cross','Dev raj urs Layout');
INSERT INTO address values(116,'3rd cross','Deshpande nagar');
INSERT INTO address values(117,'2nd cross','CFTRI layout');
INSERT INTO address values(118,'1st cross','RR nagar');
INSERT INTO address values(119,'3rd main road','Ittamadu');
```

```
INSERT INTO address values(120,'3rd cross','Patel nagar');


INSERT INTO orders values(111,'Not Delivered',101,'Pending',1);
INSERT INTO orders values(112,'Not Delivered',102,'Pending',1);
INSERT INTO orders values(113,'Delivered',103,'2020/02/01',1);
INSERT INTO orders values(114,'Delivered',104,'2019/06/23',2);
INSERT INTO orders values(115,'Delivered',105,'2019/10/03',1);
INSERT INTO orders values(116,'Delivered',106,'2018/05/25',1);
INSERT INTO orders values(117,'Delivered',107,'2019/11/16',2);
INSERT INTO orders values(118,'Delivered',108,'2018/08/26',1);
INSERT INTO orders values(119,'Delivered',109,'2018/07/26',1);
INSERT INTO orders values(120,'Delivered',110,'2019/11/05',1);


INSERT INTO order_detail values(101,1,1500.00);
INSERT INTO order_detail values(102,1,500.00);
INSERT INTO order_detail values(103,1,2500.00);
INSERT INTO order_detail values(104,2,4000.00);
INSERT INTO order_detail values(105,1,600.00);
INSERT INTO order_detail values(106,1,999.00);
INSERT INTO order_detail values(107,2,6000.00);
INSERT INTO order_detail values(108,1,3000.00);
INSERT INTO order_detail values(109,1,300.00);
INSERT INTO order_detail values(110,1,900.00);


INSERT INTO product values(101,1111,'Hair
straightner','Women','Accessories',1,'2020/03/15');
INSERT INTO product values(102,1112,'Top','Women','Clothes',1,'2020/02/10');
INSERT INTO product values(103,1113,'Bag','Women','Accessories',1,'2020/01/01');
INSERT INTO product values(104,1114,'Bluetooth speaker','Electronics','Audio
Equipment',2,'2019/05/23');
INSERT INTO product values(105,1115,'Shoes','Women','Footwear',1,'2019/09/03');
INSERT INTO product values(106,1116,'Makeup
kit','Women','Accessories',1,'2018/04/25');
INSERT INTO product values(107,1117,'Watch','Men','Accessories',2,'2019/10/16');
INSERT INTO product values(108,1118,'Watch','Women','Accessories',1,'2018/06/26');
INSERT INTO product values(109,1119,'Perfume','Men','Accessories',1,'2018/06/26');
INSERT INTO product values(110,1120,'Trimmer','Men','Accessories',1,'2019/10/05');
```

```
INSERT INTO product_detail values(1111,'Hair straightner','length : 22cm','brand : Philips',1500.00);
INSERT INTO product_detail values(1112,'Top','color : blue','brand : levis',500.00);
INSERT INTO product_detail values(1113,'Hair straightner','color : pale pink','brand : Guess',2500.00);
INSERT INTO product_detail values(1114,'Bluetooth speaker','Power : 80W','brand : JBL',4000.00);
INSERT INTO product_detail values(1115,'Shoes','Size : 24cm','brand : esse comfort',600.00);
INSERT INTO product_detail values(1116,'Makeup kit','Compact size','brand : Mac',999.00);
INSERT INTO product_detail values(1117,'Watch','Waterproof','brand : Fasttrack',6000.00);
INSERT INTO product_detail values(1118,'Watch','Sapphire crystal','brand : Titan',3000.00);
INSERT INTO product_detail values(1119,'Perfume','122ml','brand : Axe',300.00);
INSERT INTO product_detail values(1120,'Trimmer','Warranty : 1year','brand : Philips',900.00);

INSERT INTO wishlist values(1,'Jeans');
INSERT INTO wishlist values(2,'Shoes');
INSERT INTO wishlist values(1,'Bag');
INSERT INTO wishlist values(4,'Phant');
INSERT INTO wishlist values(5,'Lipstick');
INSERT INTO wishlist values(1,'Towel');
INSERT INTO wishlist values(7,'Bluetooth');
INSERT INTO wishlist values(8,'Iron Box');
INSERT INTO wishlist values(9,'Photo frame');
INSERT INTO wishlist values(10,'Watch');
```

OUTPUT OF THE ABOVE SCRIPT:

```
dbms=# select * from users;
 userid |    uname
--------+-------------
      1 | Akshatha
      2 | Pallavi
      3 | Harshitha
      4 | Akshatha01
      5 | Sonu
      6 | Bhavana
      7 | Harish
      8 | Bhuvan
      9 | Akash
     10 | Shreya
(10 rows)
```

```
dbms=# select * from customer;
 userid | cid |           cname            |           cmail            |       ccity        |   cdate    | cpincode |  cphoneno
--------+-----+----------------------------+----------------------------+--------------------+------------+----------+------------
      1 | 111 | Akshatha                   | akshatha@gmail.com         | Harihar            | 2020-03-15 | 577601   | 9632514569
      2 | 112 | Pallavi                    | pallavi@gmail.com          | Harihar            | 2020-02-10 | 577601   | 8563245698
      3 | 113 | Harshitha                  | harshitha@gmail.com        | Harihar            | 2020-01-01 | 577601   | 6325896412
      4 | 114 | Akshatha01                 | akshi123@gmail.com         | Davangere          | 2019-05-23 | 577602   | 7563256699
      5 | 115 | Sonu                       | sonusakki@gmail.com        | Davangere          | 2019-09-03 | 577602   | 6254866120
      6 | 116 | Bhavana                    | bhavana@gmail.com          | Hubli              | 2018-04-25 | 580029   | 9445213354
      7 | 117 | Harish                     | harishnaik@gmail.com       | Mysuru             | 2019-10-16 | 570026   | 9548112354
      8 | 118 | Bhuvan                     | bhuvanbhuvi@gmail.com      | Bengaluru          | 2018-06-26 | 560098   | 9875613341
      9 | 119 | Akash                      | akashhnh@gmail.com         | Bengaluru          | 2018-06-26 | 560085   | 7215468220
     10 | 120 | Shreya                     | shreyasheetal@gmail.com    | Bengaluru          | 2019-10-05 | 560085   | 8456213581
(10 rows)
```

```
dbms=# select * from address;
 cid |        street        |         area
-----+----------------------+----------------------
 111 | 1st main 1st cross   | Keshav nagar
 112 | 1st main 1st cross   | K R nagar
 113 | 1st main 1st cross   | Vidya nagar
 114 | 2nd stage            | Vinob nagar
 115 | 1st cross            | Dev raj urs Layout
 116 | 3rd cross            | Deshpande nagar
 117 | 2nd cross            | CFTRI layout
 118 | 1st cross            | RR nagar
 119 | 3rd main road        | Ittamadu
 120 | 3rd cross            | Patel nagar
(10 rows)
```

```
dbms=# select * from orders;
 cid |                                    status                                    | orderid | dateofdelivery | quty
-----+------------------------------------------------------------------------------+---------+----------------+------
 111 | Not Delivered                                                                |     101 | Pending        |    1
 112 | Not Delivered                                                                |     102 | Pending        |    1
 113 | Delivered                                                                    |     103 | 2020/02/01     |    1
 114 | Delivered                                                                    |     104 | 2019/06/23     |    2
 115 | Delivered                                                                    |     105 | 2019/10/03     |    1
 116 | Delivered                                                                    |     106 | 2018/05/25     |    1
 117 | Delivered                                                                    |     107 | 2019/11/16     |    2
 118 | Delivered                                                                    |     108 | 2018/08/26     |    1
 119 | Delivered                                                                    |     109 | 2018/07/26     |    1
 120 | Delivered                                                                    |     110 | 2019/11/05     |    1
(10 rows)
```

```
dbms=# select * from order_detail;
 orderid | quantity |  price
---------+----------+---------
     101 |        1 | 1500.00
     102 |        1 |  500.00
     103 |        1 | 2500.00
     104 |        2 | 4000.00
     105 |        1 |  600.00
     106 |        1 |  999.00
     107 |        2 | 6000.00
     108 |        1 | 3000.00
     109 |        1 |  300.00
     110 |        1 |  900.00
(10 rows)
```

```
dbms=# select * from product;
 orderid | pid  |       pname       |  pcategory  |  psubcategory   | pquantity |   datep
---------+------+-------------------+-------------+-----------------+-----------+------------
     101 | 1111 | Hair straightner  | Women       | Accessories     |         1 | 2020-03-15
     102 | 1112 | Top               | Women       | Clothes         |         1 | 2020-02-10
     103 | 1113 | Bag               | Women       | Accessories     |         1 | 2020-01-01
     104 | 1114 | Bluetooth speaker | Electronics | Audio Equipment |         2 | 2019-05-23
     105 | 1115 | Shoes             | Women       | Footwear        |         1 | 2019-09-03
     106 | 1116 | Makeup kit        | Women       | Accessories     |         1 | 2018-04-25
     107 | 1117 | Watch             | Men         | Accessories     |         2 | 2019-10-16
     108 | 1118 | Watch             | Women       | Accessories     |         1 | 2018-06-26
     109 | 1119 | Perfume           | Men         | Accessories     |         1 | 2018-06-26
     110 | 1120 | Trimmer           | Men         | Accessories     |         1 | 2019-10-05
(10 rows)
```

```
dbms=# select * from product_detail;
 pid  |       pname       |    description    |        brand          | pprice
------+-------------------+-------------------+-----------------------+---------
 1111 | Hair straightner  | length : 22cm     | brand : Philips       | 1500.00
 1112 | Top               | color : blue      | brand : levis         |  500.00
 1113 | Hair straightner  | color : pale pink | brand : Guess         | 2500.00
 1114 | Bluetooth speaker | Power : 80W       | brand : JBL           | 4000.00
 1115 | Shoes             | Size : 24cm       | brand : esse comfort  |  600.00
 1116 | Makeup kit        | Compact size      | brand : Mac           |  999.00
 1117 | Watch             | Waterproof        | brand : Fasttrack     | 6000.00
 1118 | Watch             | Sapphire crystal  | brand : Titan         | 3000.00
 1119 | Perfume           | 122ml             | brand : Axe           |  300.00
 1120 | Trimmer           | Warranty : 1year  | brand : Philips       |  900.00
(10 rows)
```

```
dbms=# select * from wishlist;
 userid |  wishname
--------+-------------
      1 | Jeans
      2 | Shoes
      1 | Bag
      4 | Phant
      5 | Lipstick
      1 | Towel
      7 | Bluetooth
      8 | Iron Box
      9 | Photo frame
     10 | Watch
(10 rows)
```

V. CHECK CONSTRAINTS

1) ALTER TABLE USERS ADD CONSTRAINT userid_min CHECK (userid>0);

2) ALTER TABLE ORDER_DETAIL ADD CONSTRAINT check_max_price CHECK (PRICE<10000);

VI. TRIGGERS

```
CREATE OR REPLACE FUNCTION tri()
RETURNS trigger AS
$$
BEGIN
IF NEW.price < 0 THEN
UPDATE order_detail SET price = 1000 WHERE orderid=NEW.orderid;

END IF;

RETURN NEW;
END;
$$
LANGUAGE PLPGSQL;

drop trigger trigg on order_detail;

CREATE TRIGGER trigg
AFTER INSERT OR UPDATE
ON order_detail
FOR EACH ROW
EXECUTE PROCEDURE tri();
```

OUTPUT:

```
dbms=# update order_detail set price=-1000 where orderid=101;
UPDATE 1
dbms=# select * from order_detail;
 orderid | quantity |  price
---------+----------+---------
     102 |        1 |  500.00
     103 |        1 | 2500.00
     104 |        2 | 4000.00
     105 |        1 |  600.00
     106 |        1 |  999.00
     107 |        2 | 6000.00
     108 |        1 | 3000.00
     109 |        1 |  300.00
     110 |        1 |  900.00
     101 |        1 | 1000.00
(10 rows)
```

VII. QUERIES

1) Retrieve username,phone,area ,city and wishname who have userid as 1.

select uname,cphoneno,area,ccity,wishname from users natural join customer natural join address natural join wishlist where wishlist.userid=1;

```
                                    ^
dbms=# select uname,cphoneno,area,ccity,wishname from users natural join customer natural joi
n address natural join wishlist where wishlist.userid=1;
  uname   |  cphoneno  |     area     |        ccity         | wishname
----------+------------+--------------+----------------------+----------
 Akshatha | 9632514569 | Keshav nagar | Harihar              | Jeans
 Akshatha | 9632514569 | Keshav nagar | Harihar              | Bag
 Akshatha | 9632514569 | Keshav nagar | Harihar              | Towel
(3 rows)
```

2) Retrieve username,phone,area ,city and wishname of all the users in wishlist.

select uname,cphoneno,area,ccity,wishname from users natural join customer natural join address natural join wishlist where wishlist.userid=users.userid;

OUTPUT:

```
dbms=# select uname,cphoneno,area,ccity,wishname from users natural join customer natural joi
n address natural join wishlist where wishlist.userid=users.userid;
   uname    |  cphoneno  |        area         |        ccity         |  wishname
------------+------------+---------------------+----------------------+-------------
Akshatha    | 9632514569 | Keshav nagar        | Harihar              | Jeans
Pallavi     | 8563245698 | K R nagar           | Harihar              | Shoes
Akshatha    | 9632514569 | Keshav nagar        | Harihar              | Bag
Akshatha01  | 7563256699 | Vinob nagar         | Davangere            | Phant
Sonu        | 6254866120 | Dev raj urs Layout  | Davangere            | Lipstick
Akshatha    | 9632514569 | Keshav nagar        | Harihar              | Towel
Harish      | 9548112354 | CFTRI layout        | Mysuru               | Bluetooth
Bhuvan      | 9875613341 | RR nagar            | Bengaluru            | Iron Box
Akash       | 7215468220 | Ittamadu            | Bengaluru            | Photo frame
Shreya      | 8456213581 | Patel nagar         | Bengaluru            | Watch
(10 rows)
```

3) List the name and phone number of customer who have not received the order.

select cname,cphoneno from customer natural join orders where orders.status='Not Delivered';

OUTPUT:

```
dbms=# select cname,cphoneno from customer natural join orders where orders.statu
s='Not Delivered';
            cname              |  cphoneno
-------------------------------+------------
 Pallavi                       | 8563245698
 Akshatha                      | 9632514569
(2 rows)
```

4) Retrieve the customer name who have pincode='577601';

select cname from customer where cpincode='577601';

OUTPUT: