QUERIES

1. Display the details of all the products of category electronics whose price<1000.

```
Select * from product where product.p_price<10000 and product.p_id in
        (select p_id from categorising where cat_name= 'electronics');
```

2. Display the details of customers who have some items in their cart. To notify and remind them that their cart has something so that we can notify the customer to buy the product.

```
        Select * from customer where c_phone in
                (select c_phone from makes where cart_id in
                        (select cart_id from consists_of where quantity>0)) ;
```

3. Display the order_id and customer-phone of the order which are delivered successfully.

```
        (select c_phone, o_id from have where o_id in
                (Select o_id from orders where o_status= 'Delivered'));
```

4. Display the details of the customers who have ordered more than 10 times.

```
        Select * from customer where c_phone in
                (select c_phone from owns where o_count>10);
```

5. customers who never queried ever

```
SELECT C.c_Name, C.c_email
 from Customer C
Where c_phone NOT IN(
SELECT C.c_phone
from Customer C, queries Q
WHERE C.c_Phone=Q.C_Phone
);
```

6. Suppliers who live in govindpuri

or supply produxts with price>30

```
SelectS_PHNO
From Supplies
Where s_PHONO in(

Select S.s_PHNO
 From supplies S, Product P
Where P.price>30 and S.P_id =P.P_id;
)
```

7. Fetch all products who are having the same price as product with id=10.

```
SELECT P2.p_id, P2.price FROM
PRODUCT P1, PRODUCT P2
WHERE P1.p_id= 10
```

8. suppliers of products ie out of stock

```
SELECT S_PHNO FROM supplies
WHERE P_id in(

SELECT P_id FROM product
WHERE available quantity=0
        )
```

9. #PRODUCT WHICH HAVE PRICE MORE THAN 1000 AND AVAILABLE QUANTITY >10

```
SELECT * FROM project.Product WHERE p_price > 1000 AND p_available_quantity  > 10;
```

10.#SELECT THE CUSTOMERS WHO HAVE PAID ONLINE

```
select * from customer where c_phone in
(select c_phone from have where o_id in
(select o_id from payment where payment_mode='online'));
```

Trigger

```
create table less_products(
amount int(4) not null
);
CREATE TRIGGER product_refill
after update on Product
for each row
if p_available_quantity<5 then
```

```
 insert into less_products(amount) value (p_id);
 end if
```

The above trigger works on updation of product. When customer buys products then the quantity of that product will decrease. Hence, when the quantity of a product becomes less than or equal to 5 then, the above trigger will add the id of that product in the table less products so that we can know which product is less in quantity and we can order it from the suppliers of that respective product.

ATTRIBUTES FOR INDEXING

1. Category_name → Index all the products by their category name.
2. Customer_phno → First sort all the customers in order of their phone numbers then Index all the customers by their phone numbers using primary indexing as the key i.e., phone number is the primary key(unique) and sorted. so that it’s easy to find a customer.
3. Employee_id → Index all the employees by their e_id using primary indexing after sorting or by secondary indexing without sorting.
4. Supplier_id → Index all the suppliers by their phone numbers.