QUERIES

Q1) Showing the details of warehouses and the watches which are stored in them.

```
select * from warehouse,warehouse_batch,batch where
warehouse.warehouse_id=warehouse_batch.warehouse_id and
warehouse_batch.batch_id=batch.batch_id;
```

Q2)To show the bill details along with customers and products that were bought sorted by total_amount of the bill and their bill_ids.

```
select bill.bill_id,customer_id,total_amount,name,quantity,mrp from
bill,bill_product,product where bill.bill_id=bill_product.bill_id and
bill_product.product_id=product.product_id order by bill.total_amount,bill.bill_id;;
```

Q3)To find all the products whose production costs are more than the mrp of products containing “paneer” in their name.

```
select p.*,t.name,t.mrp from product as p,product as t where p.cost>t.mrp and t.name
like '%paneer%';
```

Q4)To find the average capacity of a warehouse in a particular pincode.

```
select pincode,avg(capacity) from warehouse,district_warehouse where
warehouse.warehouse_id=district_warehouse.warehouse_id group by pincode;
```

Q5)To show the department and the number of employees in a particular department.

```
select department,count(department) from executive group by department;
```

Q6)To show all the warehouses and their details which are not under any district_hq

```
select * from warehouse where warehouse_id not in (select warehouse_id from
district_warehouse);
```

Q7)To show which collective has provided which material, on what date and its quantity.

```
select collective.collective_id,name,no_of_members,raw_id,supply_date,quantity
from collective inner join collective_rawmaterial on
collective.collective_id=collective_rawmaterial.collective_id;
```

Q8) To find the profit of each product between given fixed dates

```
select product.product_id,(t2.s1*product.mrp-t2.c*product.cost) as yearly_profit from
product inner join (select sum(batch.quantity) as c,batch.product_id as pid,sale.s1
from batch inner join(select sum(bill_product.quantity) as s1,product_id from
bill_product group by bill_product.product_id) as sale on
batch.product_id=sale.product_id where batch.production_date between '2021-01-01'
and '2022-011-01' group by batch.product_id) t2 on product.product_id=t2.pid;
```

Q9) To find the number of expired products and the loss incurred

```
select sum(batch.quantity) as
expired_qty,batch.product_id,sum(batch.quantity)*product.cost as loss from
batch,product where expiry_date<'2023-01-01' and
product.product_id=batch.product_id group by batch.product_id order by
batch.product_id;
```

Q10)The sales of a particular retail store(say store with id 58) product wise

```
select product_id,sum(quantity) from bill_product,bill where bill.store_id=58 and
bill.bill_id=bill_product.bill_id group by product_id order by product_id;
```

Q11) To update the product cost of products whose profit margin is 10 Rs.

```
update product set cost=cost+10,mrp=cost+cost*0.5 where (mrp-cost)=10;
```

Q12)To add a phone number of retailer

```
#does not work, foreign_key constraint fails.
insert into retailer_phone(store_id,phone) values(125,921029379);

#does not work, duplicate phone number.
insert into retailer_phone(store_id,phone) values(25,9555274333);

#inserting phone number
(just to avoid duplicate numbers)
delete from retailer_phone where phone=857613591;
insert into retailer_phone(store_id,phone) values(1,857613591);
```

Q13) Updating batch quantity when used for a particular batch(say batch 69)

```
update batch set quantity=quantity-100 where batch_id=69;
```

Q14)Update bill id

```
#duplicate entry for primary key
update bill set bill_id=139 where bill_id=140;
```

Q15)Adding a new customer

```
#does not work as name cannot be NULL.
 insert into customer(name,age,gender,phone) values(NULL,32,'male',921234567);
insert into customer(age,gender,phone) values(32,'male',921234567);

#does not work due to unique phone number
insert into customer(name,age,gender,phone) values('abc',32,'male',921234567);
delete from customer where phone=921234567;
#works after running this
```

Q16)Creating a view to see the details of employees while safeguarding their personal information.

```
create view employees as select emp_id,name,department,hq_id from executive;
```