Triggers :-

Trigger for updating quantity of products in the inventory when customer adds product in their cart

```
mysql> create trigger nobita
    -> after insert
    -> on cart_items
    -> for each row
    -> UPDATE product set Quantity = Quantity - new.quantity where Product_ID = new.product_id;
Query OK, 0 rows affected (0.01 sec)
```

Trigger for deleting items from table cart and cart_items on checkout (place order)

```
mysql> create trigger trigcart
    -> after Update
    -> on orders
    -> for each row
    -> delete from cart where customer_id=new.Customer_ID;
Query OK, 0 rows affected (0.02 sec)

mysql> create trigger trigcartitems
    -> after Update
    -> on orders
    -> for each row
    -> delete from cart_items where customer_id=new.Customer_ID;
Query OK, 0 rows affected (0.02 sec)
```

Embedded Queries(other basic queries also written for CLI) :-

Query for Inserting order of a customer to be placed and then calculating total price of the products ordered

```
cur=database.cursor(buffered=True)
qry = ("INSERT into orders values(%s,%s,%s,%s,%s,%s,%s,%s,%s)")
cur.execute(qry , [newid , 'Pending' , add , datetime.today() , datetime.now().strftime("%H:%M:%S") , mop , discount , 0 , custid])
database.commit()
cur=database.cursor(buffered=True)
qry = ('''select sum(cart_items.quantity * product.Price) from cart_items, product, orders where cart_items.customer_id = %s and
    orders.Customer_ID = %s and cart_items.product_id = product.Product_ID''')
cur.execute(qry , [custid , custid])
for x in cur:
    amount = x[0]
cur=database.cursor(buffered=True)
qry = ("UPDATE orders set Amount = %s")
cur.execute(qry , [amount])
database.commit()
```

Query for deleting product from the cart of the customer

```
y=int(input("Enter how many products you want to delete : "))
for i in range(y):
    pid=input("Enter id of the product you want to delete : ")
    cur=database.cursor(buffered=True)
    q="SELECT product_id from cart_items where customer_id=%s"
    cur.execute(q,[custid])
    if not cur.fetchall():
        print("No product found")
    else:
        cur=database.cursor(buffered=True)
        q="SELECT quantity from cart_items where customer_id=%s and product_id=%s"
        cur.execute(q,[custid,pid])
        qty=0
        for x in cur:
            qty=x[0]
        cur=database.cursor(buffered=True)
        q="DELETE FROM cart_items where product_id=%s and customer_id=%s"
        cur.execute(q,[pid,custid])
        database.commit()
        cur=database.cursor(buffered=True)
        q="UPDATE cart SET no_of_items=no_of_items-%s where customer_id=%s"
        cur.execute(q,[qty,custid])
        database.commit()
        print("Product removed successfully")
```

OLAP Queries :-

Finding the number of customers of certain gender and the type

```
cur.execute('''select Customer_type , Gender , count(*) from customer group by Customer_type , Gender with rollup union
    select Customer_type , Gender , count(*) from customer group by Gender , Customer_type with rollup''')
```

Finding number of products of the same category

```
cur=database.cursor(buffered=True)
cur.execute('''select Name , Category_ID , sum(Quantity) from product group by Category_ID , name with rollup''')
```

Finding the number of products manufactured of every category by the manufacturer

```
cur=database.cursor(buffered=True)
cur.execute('''select name , Product_category , sum(Number_of_products) from manufacturer group by Product_category , Name with rollup''')
```

Finding the total sales of each admin after using join of admin table with sales_manager table

```
cur.execute(''' select a.username , s.Sales_manager_ID , sum(s.Payments) from admin as a INNER JOIN sales_manager as s
            on a.Admin_ID = s.Admin_ID group by a.username , s.Sales_manager_ID with rollup''')
```