TASC

1. CREATE TABLE USERS (UID VARCHAR(10)PRIMARY KEY,UNAME VARCHAR(20));

   CREATE TABLE PRODUCTS (PID VARCHAR(10) PRIMARY KEY,PNAME VARCHAR(20));

   CREATE TABLE ORDERS (OID VARCHAR(20)PRIMARY KEY,UNAME VARCHAR(20),PID VARCHAR(10)REFERENCES PRODUCTS(PID),PNAME VARCHAR(20),PRICE INTEGER,QUANTITY INTEGER);

   CREATE TABLE ORDER_DETAILS(OID VARCHAR(20) REFERENCES ORDERS(OID),ODATE DATE,ADDRESS VARCHAR(30));

2. SELECT PNAME,COUNT(PNAME)FROM ORDERS GROUP BY PNAME HAVING COUNT(PNAME)>50;

3. SELECT DISTINCT ORDERS.UNAME,ORDERS.OID, ORDER_DETAILS.ODATE FROM ORDERS JOIN ORDER_DETAILS ON ORDERS.OID=ORDER_DETAILS.OID ORDER BY ODATE DESC;

4. SELECT SUM(PRICE*QUANTITY)FROM ORDERS;

5. To handle concurrent transaction MySQL we can use shared locks. It allows multiple transactions to read same data at the same time but restricts any of them from writing or making changes to it.

6. CREATE TABLE SALES(PID VARCHAR(10) REFERENCES PRODUCTS(PID) ,QUANTITY_SOLD INTEGER ,DATE_SOLD DATE);

   SELECT MONTH(DATE_SOLD),PID,SUM(QUANTITY_SOLD) FROM SALES GROUP BY PID ,MONTH(DATE_SOLD) HAVING COUNT(MONTH(DATE_SOLD)) <=3 ORDER BY MONTH(DATE_SOLD) ASC ;

7. To secure sensitive data we can use data encryption method. By using this method we can convert the plain text to cipher text .

8. SELECT PNAME FROM PRODUCTS WHERE PID IN(SELECT PID FROM ORDERS);

   Here IN operator takes more time.so we can use the EXISTS operator instead.

   SELECT PNAME FROM ORDERS WHERE EXISTS (SELECT PID FROM PRODUCTS);

Submitted by,
KEERTHANA V S