Project- (SQLite & MYSQL)

1st part- Q1-Q6 comes under SQLite and queries should be executed in DB Browser. (Database - **Orders.db)**

2nd part- Q7-Q10 comes under MYSQL and the queries should be executed in MYSQL. (SQL Script -**orders.sql**)

1st Part

1. Write a query to Display the product details (product_class_code, product_id, product_desc, product_price,) as per the following criteria and sort them in descending order of category:
   a. If the category is 2050, increase the price by 2000
   b. If the category is 2051, increase the price by 500
   c. If the category is 2052, increase the price by 600.

   Hint: Use case statement, no permanent change in table required.

   (60 ROWS)[NOTE:PRODUCT TABLE]

ANSWER:

```
 SELECT product_class_code, product_id, product_desc, product_price,
 CASE product_class_code
        WHEN 2050 THEN product_price + 2000
        WHEN 2051 THEN product_price + 500
        WHEN 2052 THEN product_price + 600
   ELSE product_price
   END AS incr_price
   FROM Product
   ORDER BY product_class_code desc;
```

2. Write a query to display (product_class_desc, product_id, product_desc, product_quantity_avail) and Show inventory status of products as below as per their available quantity:

    a. For Electronics and Computer categories, if available quantity is <= 10, show 'Low stock', 11 <= qty <= 30, show 'In stock', >= 31, show 'Enough stock'

    b. For Stationery and Clothes categories, if qty <= 20, show 'Low stock', 21 <= qty <= 80, show 'In stock', >=81, show 'Enough stock'

    c. Rest of the categories, if qty <= 15 – 'Low Stock', 16 <= qty <= 50 – 'In Stock', >= 51 – 'Enough stock'

For all categories, if available quantity is 0, show 'Out of stock'.

Hint: Use case statement.

(60 ROWS)[NOTE : TABLES TO BE USED – product, product_class]

ANSWER:

```
SELECT pc.product_class_desc, product_id, product_desc, product_quantity_avail,
CASE
        WHEN product_class_desc = 'Electronics' OR product_class_desc =
'Computer' THEN
                CASE
                        WHEN product_quantity_avail = 0 THEN 'Out of stock'
                        WHEN product_quantity_avail <= 10 THEN 'Low stock'
                        WHEN product_quantity_avail BETWEEN 11 AND 30 THEN 'In
stock'
                        WHEN product_quantity_avail >= 31 THEN 'Enough stock'
                END
WHEN product_class_desc = 'Stationery' OR product_class_desc = 'Clothes' THEN
                CASE
WHEN product_quantity_avail = 0 THEN 'Out of stock'
WHEN product_quantity_avail <= 20 THEN 'Low stock'
WHEN product_quantity_avail BETWEEN 21 AND 80 THEN 'In stock'
                        WHEN product_quantity_avail >= 81 THEN 'Enough stock'
                END
```

```
          ELSE
                    CASE
                              WHEN product_quantity_avail = 0 THEN 'Out of stock'
                              WHEN product_quantity_avail <= 15 THEN 'Low stock'
                              WHEN product_quantity_avail BETWEEN 16 AND 50 THEN 'In stock'
                              WHEN product_quantity_avail >= 51 THEN 'Enough stock'
                    END
   END AS inventory_level
FROM product p INNER JOIN product_class pc
          ON p.product_class_code = pc.product_class_code;
```

3. Write a query to Show the count of cities in all countries other than USA & MALAYSIA, with more than 1 city, in the descending order of CITIES.
(2 rows)[NOTE :ADDRESS TABLE]

ANSWER:

```
select count(city)as count_cities ,country
from address where country not in ('USA','Malaysia')
group by country
having count(city) > 1
order by count_cities desc;
```

4. Write a query to display the customer_id,customer full name ,city,pincode,and order details (order id,order date, product class desc, product desc, subtotal(product_quantity * product_price)) for orders shipped to cities whose pin codes do not have any 0s in them. Sort the output on customer name, order date and subtotal.(52 ROWS)

[NOTE : TABLE TO BE USED - online_customer, address, order_header, order_items, product, product_class]

ANSWER:-

```
SELECT oc.customer_id, oc.customer_fname|| oc.customer_lname AS fullname,
a.city, a.pincode, oh.order_id, pc.product_class_desc, p.product_desc,
oi.product_quantity * p.product_price AS subtotal
FROM online_customer oc INNER JOIN address a
        ON oc.address_id = a.address_id
INNER JOIN order_header oh
        ON oc.customer_id = oh.customer_id
  AND oh.order_status = 'Shipped'
INNER JOIN order_items oi
        ON oh.order_id = oi.order_id
INNER JOIN product p
        ON oi.product_id = p.product_id
INNER JOIN product_class pc
        ON pc.product_class_code = p.product_class_code
where a.PINCODE not like "%0%"
ORDER BY fullname, oh.order_date, subtotal;
```

5. Write a Query to display product id,product description,totalquantity(sum(product quantity) for a given item whose product id is 201 and which item has been bought along with it maximum no. of times. Display only one record which has the maximum value for total quantity in this scenario.

   (USE SUB-QUERY)(1 ROW)[NOTE : ORDER_ITEMS TABLE,PRODUCT TABLE]

   ANSWER:-

```
SELECT p.product_id, product_desc, SUM(product_quantity) AS tot_qty
FROM Order_Items oi, Product p
WHERE order_id IN
 (SELECT order_id FROM Order_Items
  WHERE product_id = 201)
 AND p.product_id != 201
AND oi.product_id = p.product_id
GROUP BY p.product_id, product_desc
ORDER BY tot_qty DESC LIMIT 1;
```

6. Write a query to display the customer_id,customer name, email and order details (order id, product desc,product qty, subtotal(product_quantity * product_price)) for all customers even if they have not ordered any item.(225 ROWS)
[NOTE : TABLE TO BE USED - online_customer, order_header, order_items, product]

ANSWER:

```
SELECT oc.customer_id, oc.customer_fname|| oc.customer_lname AS fullname,
customer_email, oh.order_id, p.product_desc, oi.product_quantity AS prod_qty,
oi.product_quantity * p.product_price AS subtotal
FROM online_customer oc LEFT JOIN order_header oh
        ON oc.customer_id = oh.customer_id
LEFT JOIN order_items oi
        ON oh.order_id = oi.order_id
LEFT JOIN product p
        ON oi.product_id = p.product_id
ORDER BY oc.customer_id, oh.order_id, p.product_desc;
```

IInd Part

7. Write a query to display carton id ,(len*width*height) as carton_vol and identify the optimum carton (carton with the least volume whose volume is greater than the total volume of all items(len * width * height * product_quantity)) for a given order whose order id is 10006 , Assume all items of an order are packed into one single carton (box) .(1 ROW)[NOTE : CARTON TABLE]

ANSWER:-

```
SELECT carton_id, (len * width * height) AS carton_vol FROM Carton
WHERE (len * width * height) >=
(
            SELECT SUM(len * width * height * product_quantity)
            FROM Order_Items oi INNER JOIN Product p
            ON oi.product_id = p.product_id
            WHERE order_id = 10006
)
ORDER BY carton_vol LIMIT 1;
```

8 . Write a query to display details (customer id,customer fullname,order id,product quantity) of customers who bought more than ten (i.e. total order qty) products per shipped order.

(11 ROWS)

[NOTE: TABLES TO BE USED - online_customer, order_header, order_items,]

```
SELECT oc.customer_id,
CONCAT(oc.customer_fname, ' ', oc.customer_lname) AS fullname,
oh.order_id, SUM(oi.product_quantity) AS tot_qty
FROM online_customer oc INNER JOIN order_header oh
            ON oc.customer_id = oh.customer_id
   AND oh.order_status = 'Shipped'
INNER JOIN order_items oi
            ON oh.order_id = oi.order_id
GROUP BY oc.customer_id, fullname, oh.order_id
HAVING tot_qty > 10;
```

9 . Write a query to display the order_id, customer id and cutomer full name of customers along with (product_quantity) as total quantity of products shipped for order ids > 10060. (6 ROWS)

[NOTE : TABLES TO BE USED - online_customer, order_header, order_items]

```
SELECT  oc.customer_id, oh.order_id,
CONCAT(oc.customer_fname, ' ', oc.customer_lname) AS fullname,
SUM(oi.product_quantity) AS tot_qty
FROM online_customer oc INNER JOIN order_header oh
```

```
            ON oc.customer_id = oh.customer_id
   AND oh.order_status = 'Shipped'
INNER JOIN order_items oi
            ON oh.order_id = oi.order_id
WHERE oh.order_id > 10060
GROUP BY oc.customer_id, fullname;
```

10. Write a query to display product class description ,total quantity(sum(product_quantity),

Total value (product_quantity * product price) and show which class of products have been shipped highest**(Quantity)** to countries outside India other than USA? Also show the total value of those items.

(1 ROWS)[NOTE:PRODUCT TABLE,ADDRESS TABLE,ONLINE_CUSTOMER TABLE,ORDER_HEADER TABLE,ORDER_ITEMS TABLE,PRODUCT_CLASS TABLE]

ANSWER:-

```
SELECT product_class_desc, SUM(oi.product_quantity) AS total_qty,
SUM(oi.product_quantity * p.product_price) AS total_value

FROM Address a inner join Online_Customer oc on oc.address_id=a.address_id
inner join Order_Header oh on  oc.customer_id = oh.customer_id
inner join Order_Items o on oh.order_id = oi.order_id
inner join Product p on  oi.product_id = p.product_id
inner join Product_class pc on p.product_class_code = pc.product_class_code

WHERE a.country != 'India'
AND a.country != 'USA'
AND order_status = "shipped"
GROUP BY product_class_desc
ORDER BY total_qty DESC limit 1;
```