**SQL – TAKE HOME LAB\_EXERCISE – 03**

**USE Orders SCHEMA:**

**PLEASE FIND LINK :DOWNLOAD ORDERS SCHEMA AND IMPORT IN MY SQL**

[**https://drive.google.com/open?id=15t6\_aO54J9iFPPirXLp9pUGcKGJ9NeYO**](https://drive.google.com/open?id=15t6_aO54J9iFPPirXLp9pUGcKGJ9NeYO)

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:**
   1. **If the category is 2050, increase the price by 2000**
   2. **If the category is 2051, increase the price by 500**
   3. **If the category is 2052, increase the price by 600.**

**(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 2050 THEN product\_price + 500

WHEN 2050 THEN product\_price + 600

ELSE product\_price

END AS incr\_price

FROM Product

ORDER BY product\_class\_code DESC;

1. **Write a Query to display the the product description, product class description and product price of all products which are shipped.(168 rows)**

**[NOTE : TABLE TO BE USED:PRODUCT\_CLASS,PRODUCT, ORDER\_ITEMS,ORDER\_HEADER]**

**ANSWER:**

SELECT product\_class\_desc, product\_desc, product\_price

FROM product\_class pc INNER JOIN product p

ON pc.product\_class\_code = p.product\_class\_code

INNER JOIN order\_items oi

ON p.product\_id = oi.product\_id

INNER JOIN order\_header oh

ON oi.order\_id = oh.order\_id

WHERE oh.order\_status = 'Shipped';

1. **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, CONCAT(oc.customer\_fname, ' ', oc.customer\_lname) AS fullname,

customer\_email, oh.order\_id, p.product\_desc, IFNULL(oi.product\_quantity,0) AS prod\_qty,

IFNULL(oi.product\_quantity \* p.product\_price, 0) 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;

1. **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, CONCAT(oc.customer\_fname, ' ', oc.customer\_lname) AS fullname,

a.city, a.pincode, oh.order\_id, oh.order\_date, pc.product\_class\_desc, p.product\_desc,

IFNULL(oi.product\_quantity \* p.product\_price, 0) AS subtotal

FROM online\_customer oc INNER JOIN address a

ON oc.address\_id = a.address\_id

AND CAST(a.pincode AS CHAR) NOT LIKE '%0%'

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

ORDER BY fullname, oh.order\_date, subtotal;

1. **Write a query to display (customer id,customer fullname,city) of customers from outside ‘Karnataka’ who haven’t bought any toys or books.(19 ROWS)**

**[NOTE : TABLES TO BE USED – online\_customer, address,**

**order\_header, order\_items, product, product\_class]**

**ANSWER:**

SELECT DISTINCT oc.customer\_id,

CONCAT(oc.customer\_fname, ' ', oc.customer\_lname) AS fullname, a.city

FROM online\_customer oc INNER JOIN address a

ON oc.address\_id = a.address\_id

AND a.state != 'Karnataka'

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

AND pc.product\_class\_desc NOT IN ('Toys', 'Books');

1. **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 order.**

**(11 ROWS)**

**[NOTE : TABLES TO BE USED - online\_customer, order\_header, order\_items,]**

**ANSWER:**

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;

1. **Write a query to display the customer full name and total order value(product\_quantity\*product\_price) of premium customers (i.e. the customers who bought items total worth > Rs. 1 lakh.)(2 ROWS)**

**[ NOTE : TABLES TO BE USED – ONLINE\_CUSTOMER,ORDER\_HEADER,**

**ORDER\_ITEMS,PRODUCT,]**

**ANSWER:**

SELECT oc.customer\_id,

CONCAT(oc.customer\_fname, ' ', oc.customer\_lname) AS fullname,

SUM(oi.product\_quantity \* p.product\_price) AS tot\_ord\_value

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

INNER JOIN product p

ON oi.product\_id = p.product\_id

GROUP BY oc.customer\_id, fullname

HAVING tot\_ord\_value > 100000;

1. **Write a query to display the customer id and cutomer full name of customers along with (product\_quantity) as total quantity of products ordered for order ids > 10060.(6 ROWS)**

**[NOTE : TABLES TO BE USED - online\_customer, order\_header, order\_items]**

**ANSWER:**

SELECT oc.customer\_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;

1. **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:**
   1. **For Electronics and Computer categories, if available quantity is < 10, show 'Low stock', 11 < qty < 30, show 'In stock', > 31, show 'Enough stock'**
   2. **For Stationery and Clothes categories, if qty < 20, show 'Low stock', 21 < qty < 80, show 'In stock', > 81, show 'Enough stock'**
   3. **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'.**

**(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;