**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]**

**select** product\_class\_code, product\_id, product\_desc, product\_price,

**case** **when** PRODUCT\_CLASS\_CODE = 2050 **then** product\_price +2000

**when** PRODUCT\_CLASS\_CODE = 2051 **then** product\_price +500

**when** PRODUCT\_CLASS\_CODE = 2052 **then** product\_price+ 600

**end**

**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]**

**select** product\_desc,product\_class\_desc,product\_price **from** ORDER\_HEADER

**join** order\_items **on** order\_items.order\_id = ORDER\_HEADER.order\_id

**join** product **on** product.product\_id = order\_items.product\_id

**join** PRODUCT\_CLASS **on** PRODUCT\_CLASS.product\_class\_code = product.product\_class\_code

**where** 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]**

**select** online\_customer.customer\_id,customer\_fname,customer\_email,ORDER\_HEADER.order\_id,product\_desc,product\_quantity,(product\_quantity \* product\_price) **as** total **from** online\_customer

**join** ORDER\_HEADER **on** ORDER\_HEADER.customer\_id = online\_customer.customer\_id

**join** order\_items **on** order\_items.order\_id = ORDER\_HEADER.order\_id

**join** product **on** product.product\_id = order\_items.product\_id

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]**

**select** online\_customer.customer\_id,**concat**(customer\_fname,' ',customer\_lname) **as** name,customer\_email,ORDER\_HEADER.order\_id,product\_desc,product\_quantity,(product\_quantity \* product\_price),pincode **as** total

**from** online\_customer

**join** address **on** address.address\_id = ONLINE\_CUSTOMER.address\_id

**join** ORDER\_HEADER **on** ORDER\_HEADER.customer\_id = online\_customer.customer\_id

**join** order\_items **on** order\_items.order\_id = ORDER\_HEADER.order\_id

**join** product **on** product.product\_id = order\_items.product\_id

**join** PRODUCT\_CLASS **on** PRODUCT\_CLASS.product\_class\_code = product.product\_class\_code

**where** pincode **not** **like** '%0%'**and** order\_status = 'Shipped'**order** **by** name,order\_date,total

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].**

**select** \* **from** ONLINE\_CUSTOMER

**join** address **on** address.address\_id = online\_customer.address\_id

**join** order\_header **on** order\_header.customer\_id = online\_customer.customer\_id

**join** order\_items **on** order\_items.order\_id = order\_header.order\_id

**join** product **on** product.product\_id = order\_items.product\_id

**where** state != 'Karnataka' **and** product\_class\_code **not** **in** (2051,2054) **and** order\_status = 'Shipped' **group** **by** online\_customer.customer\_id

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]**

**select** \*,**sum**(product\_quantity) **as** quantity **from** online\_customer

**left** **join** ORDER\_HEADER **on** ORDER\_HEADER.customer\_id = online\_customer.customer\_id

**inner** **join** order\_items **on** order\_items.order\_id = ORDER\_HEADER.order\_id

**where** order\_status = 'Shipped'

**group** **by** ONLINE\_CUSTOMER.customer\_id,order\_items.order\_id **having** quantity>=10 **order** **by** online\_customer.customer\_id

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]**

**select** **concat**(customer\_fname,' ',customer\_lname),**sum**(product\_quantity\*product\_price) worth **from** ORDER\_HEADER

**join** order\_items **on** order\_items.order\_id =order\_header.order\_id

**join** product **on** product.product\_id = order\_items.product\_id

**join** ONLINE\_CUSTOMER **on** ONLINE\_CUSTOMER.customer\_id = order\_header.customer\_id

**where** ORDER\_HEADER.order\_status = 'Shipped'

**group** **by** ONLINE\_CUSTOMER.customer\_id **having** worth > 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]**

**select** **concat**(customer\_fname,' ',customer\_lname),**sum**(product\_quantity) quant **from** ORDER\_HEADER

**join** order\_items **on** order\_items.order\_id =order\_header.order\_id

**join** ONLINE\_CUSTOMER **on** ONLINE\_CUSTOMER.customer\_id = order\_header.customer\_id

**where** order\_header.order\_id > 10060 **and** order\_status = 'shipped'

**group** **by** ONLINE\_CUSTOMER.customer\_id

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].**

**select** product\_desc,product\_class\_desc,product\_id,product\_quantity\_avail,product\_class.product\_class\_code,

**case**

**when** product\_quantity\_avail = 0 **then** 'Out of stock'

**when** product\_class.product\_class\_code **in** (2050,2053) **then**

**case** **when** product\_quantity\_avail < 10 **then** 'Low Stock'

**when** product\_quantity\_avail <30 **and** product\_quantity\_avail >11 **then** 'In Stock'

**else** 'Enough Stock' **end**

**when** product\_class.product\_class\_code **in** (2056,2052) **then**

**case** **when** product\_quantity\_avail < 20 **then** 'Low Stock'

**when** product\_quantity\_avail <80 **and** product\_quantity\_avail >21 **then** 'In Stock'

**else** 'Enough Stock' **end**

**else**

**case** **when** product\_quantity\_avail < 15 **then** 'Low Stock'

**when** product\_quantity\_avail <50 **and** product\_quantity\_avail >16 **then** 'In Stock'

**else** 'Enough Stock' **end**

**end** **as** someOthers

**from** product **join** PRODUCT\_CLASS **on** product\_class.product\_class\_code = product.product\_class\_code