# Problem Statement - SQL Project - Coded

**Project Problem Statement:**

You are hired by a chain of online retail stores **“Reliant retail limited”**. They provide you with “**orders**” database and seek answers to the following queries as the results from these queries will help the company in making data-driven decisions that will impact the overall growth of the online retail store.

1. Write a query to display customer full name with their title (mr/ms), both first name and last name are in upper case with customer email id, customer creation date and display customer’s category after applying below categorization rules:

i. If customer creation date year <2005 then category a

ii. If customer creation date year >=2005 and <2011 then category b

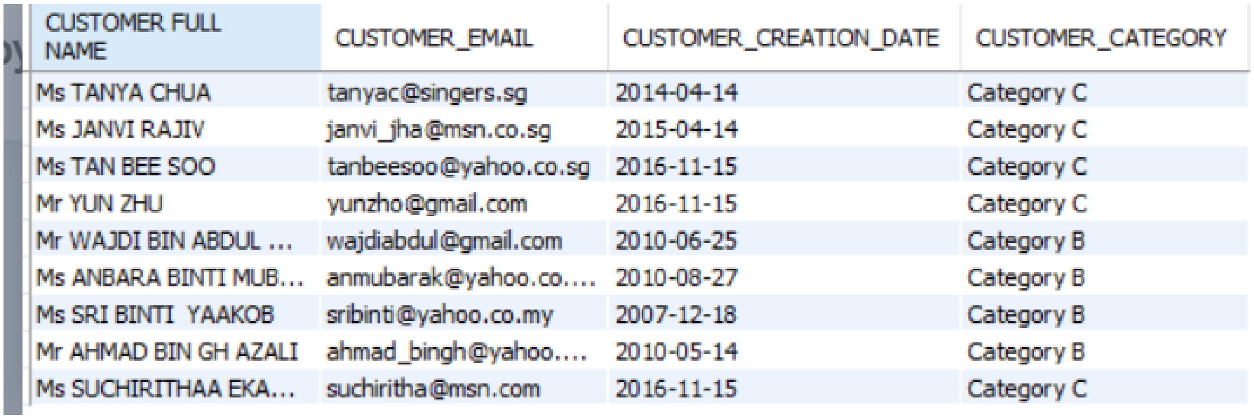
iii. If customer creation date year>= 2011 then category c

## SOLUTION

```
SELECT CONCAT(B.TITLE, ' ', B.CUSTOMER_FNAME, ' ',
B.CUSTOMER_LNAME) 'CUSTOMER FULL NAME', A.CUSTOMER_EMAIL,
A.CUSTOMER_CREATION_DATE, B.CUSTOMER_CATEGORY

FROM ONLINE_CUSTOMER A

INNER JOIN
```

```
          (SELECT
          CASE WHEN CUSTOMER_GENDER = 'F'
          THEN 'Ms'
          ELSE 'Mr'
          END AS Title,
          CUSTOMER_ID,
          UPPER(CUSTOMER_FNAME) CUSTOMER_FNAME,
          UPPER(CUSTOMER_LNAME) CUSTOMER_LNAME,
          CASE
          WHEN YEAR(CUSTOMER_CREATION_DATE) < 2005
          THEN 'Category A'
          WHEN YEAR(CUSTOMER_CREATION_DATE) >= 2005 AND
YEAR(CUSTOMER_CREATION_DATE) < 2011
          THEN 'Category B'
          WHEN YEAR(CUSTOMER_CREATION_DATE) >= 2011
          THEN 'Category C'
          ELSE 'UNKNOWN CATEGORY'
          END AS CUSTOMER_CATEGORY
          FROM online_customer) B
ON A.CUSTOMER_ID = B.CUSTOMER_ID
```

# OUTPUT



| CUSTOMER FULL NAME | CUSTOMER_EMAIL | CUSTOMER_CREATION_DATE | CUSTOMER_CATEGORY |
|---|---|---|---|
| Ms TANYA CHUA | tanyac@singers.sg | 2014-04-14 | Category C |
| Ms JANVI RAJIV | janvi_jha@msn.co.sg | 2015-04-14 | Category C |
| Ms TAN BEE SOO | tanbeesoo@yahoo.co.sg | 2016-11-15 | Category C |
| Mr YUN ZHU | yunzho@gmail.com | 2016-11-15 | Category C |
| Mr WAJDI BIN ABDUL ... | wajdiabdul@gmail.com | 2010-06-25 | Category B |
| Ms ANBARA BINTI MUB... | anmubarak@yahoo.co.... | 2010-08-27 | Category B |
| Ms SRI BINTI  YAAKOB | sribinti@yahoo.co.my | 2007-12-18 | Category B |
| Mr AHMAD BIN GH AZALI | ahmad_bingh@yahoo.... | 2010-05-14 | Category B |
| Ms SUCHIRITHAA EKA... | suchiritha@msn.com | 2016-11-15 | Category C |

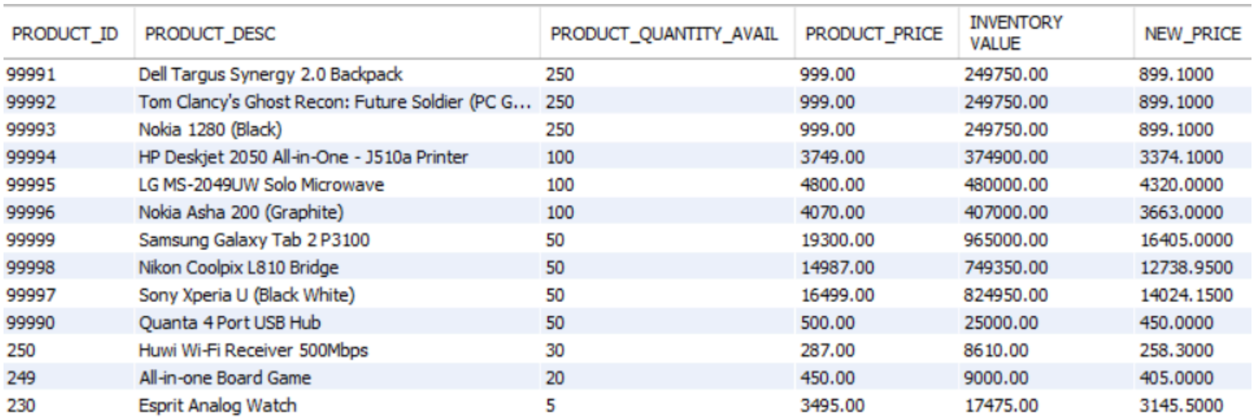
2. Write a query to display the following information for the products, which have not been sold:  product_id, product_desc, product_quantity_avail, product_price, inventory values(product_quantity_avail*product_price), new_price after applying discount as per the below criteria. Sort the output concerning the decreasing value of inventory_value.
i. If product price > 20,000 then apply 20% discount
ii. If product price > 10,000 then apply 15% discount
iii. If product price =< 10,000 then apply 10% discount
Hint: use case statement, no permanent change in table required. [note: tables to be used -product, order_items table]

## SOLUTION:

[NOTE: TABLES TO BE USED - PRODUCT, ORDER_ITEMS TABLE]

```
SELECT
p.PRODUCT_ID,
p.PRODUCT_DESC,
p.PRODUCT_QUANTITY_AVAIL,
p.PRODUCT_PRICE,
(p.PRODUCT_QUANTITY_AVAIL*p.PRODUCT_PRICE) 'INVENTORY VALUE',
CASE
WHEN p.PRODUCT_PRICE > 20000
THEN (p.PRODUCT_PRICE - (p.PRODUCT_PRICE * 0.20))
WHEN p.PRODUCT_PRICE > 10000
THEN (p.PRODUCT_PRICE - (p.PRODUCT_PRICE * 0.15))
WHEN PRODUCT_PRICE <= 10000
THEN (p.PRODUCT_PRICE - (p.PRODUCT_PRICE * 0.10))
END AS NEW_PRICE
FROM product p
LEFT JOIN ORDER_ITEMS o
ON p.PRODUCT_ID = o.PRODUCT_ID
WHERE o.ORDER_ID IS NULL
ORDER BY PRODUCT_QUANTITY_AVAIL DESC
```

# OUTPUT



| PRODUCT_ID | PRODUCT_DESC | PRODUCT_QUANTITY_AVAIL | PRODUCT_PRICE | INVENTORY VALUE | NEW_PRICE |
|---|---|---|---|---|---|
| 99991 | Dell Targus Synergy 2.0 Backpack | 250 | 999.00 | 249750.00 | 899.1000 |
| 99992 | Tom Clancy's Ghost Recon: Future Soldier (PC G... | 250 | 999.00 | 249750.00 | 899.1000 |
| 99993 | Nokia 1280 (Black) | 250 | 999.00 | 249750.00 | 899.1000 |
| 99994 | HP Deskjet 2050 All-in-One - J510a Printer | 100 | 3749.00 | 374900.00 | 3374.1000 |
| 99995 | LG MS-2049UW Solo Microwave | 100 | 4800.00 | 480000.00 | 4320.0000 |
| 99996 | Nokia Asha 200 (Graphite) | 100 | 4070.00 | 407000.00 | 3663.0000 |
| 99999 | Samsung Galaxy Tab 2 P3100 | 50 | 19300.00 | 965000.00 | 16405.0000 |
| 99998 | Nikon Coolpix L810 Bridge | 50 | 14987.00 | 749350.00 | 12738.9500 |
| 99997 | Sony Xperia U (Black White) | 50 | 16499.00 | 824950.00 | 14024.1500 |
| 99990 | Quanta 4 Port USB Hub | 50 | 500.00 | 25000.00 | 450.0000 |
| 250 | Huwi Wi-Fi Receiver 500Mbps | 30 | 287.00 | 8610.00 | 258.3000 |
| 249 | All-in-one Board Game | 20 | 450.00 | 9000.00 | 405.0000 |
| 230 | Esprit Analog Watch | 5 | 3495.00 | 17475.00 | 3145.5000 |

3.write a query to display product_class_code, product_class_description, count of product type in each product class, and inventory value (p.product_quantity_avail*p.product_price). Information should be displayed for only those product_class_code that have more than 1,00,000 inventory value. sort the output concerning the decreasing value of inventory_value.

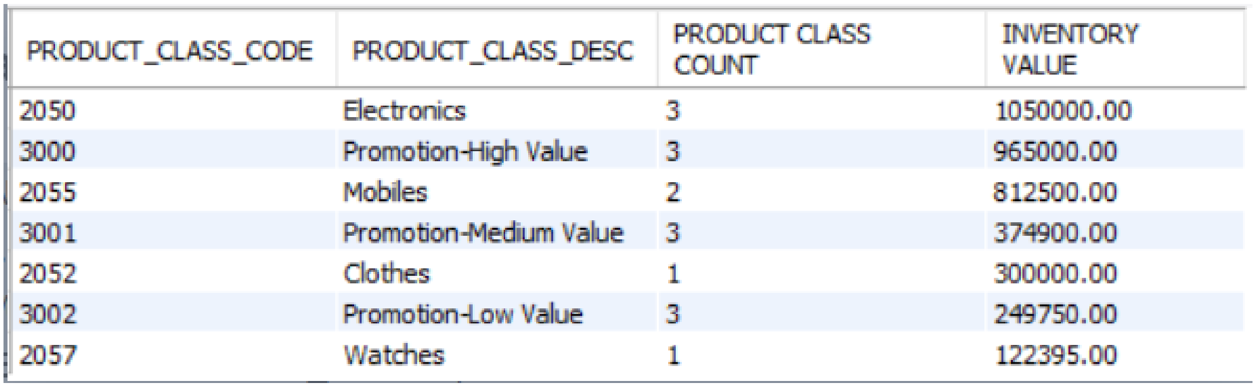
## SOLUTION

[note: tables to be used -product, product_class][9 ROWS)

```
SELECT

pr.PRODUCT_CLASS_CODE,

pc.PRODUCT_CLASS_DESC,

COUNT(pr.PRODUCT_CLASS_CODE) 'PRODUCT CLASS COUNT',

(pr.PRODUCT_QUANTITY_AVAIL*pr.PRODUCT_PRICE) 'INVENTORY VALUE'

FROM PRODUCT pr

INNER JOIN PRODUCT_CLASS pc

ON pr.PRODUCT_CLASS_CODE = pc.PRODUCT_CLASS_CODE
```

```
WHERE (pr.PRODUCT_QUANTITY_AVAIL*pr.PRODUCT_PRICE) > 100000

GROUP BY pr.PRODUCT_CLASS_CODE, pc.PRODUCT_CLASS_DESC

ORDER BY (pr.PRODUCT_QUANTITY_AVAIL*pr.PRODUCT_PRICE) DESC
```

## OUTPUT



| PRODUCT_CLASS_CODE | PRODUCT_CLASS_DESC | PRODUCT CLASS COUNT | INVENTORY VALUE |
|---|---|---|---|
| 2050 | Electronics | 3 | 1050000.00 |
| 3000 | Promotion-High Value | 3 | 965000.00 |
| 2055 | Mobiles | 2 | 812500.00 |
| 3001 | Promotion-Medium Value | 3 | 374900.00 |
| 2052 | Clothes | 1 | 300000.00 |
| 3002 | Promotion-Low Value | 3 | 249750.00 |
| 2057 | Watches | 1 | 122395.00 |

4. Write a query to display customer_id, full name, customer_email, customer_phone and country of customers who have canceled all the orders placed by them(use sub-query)

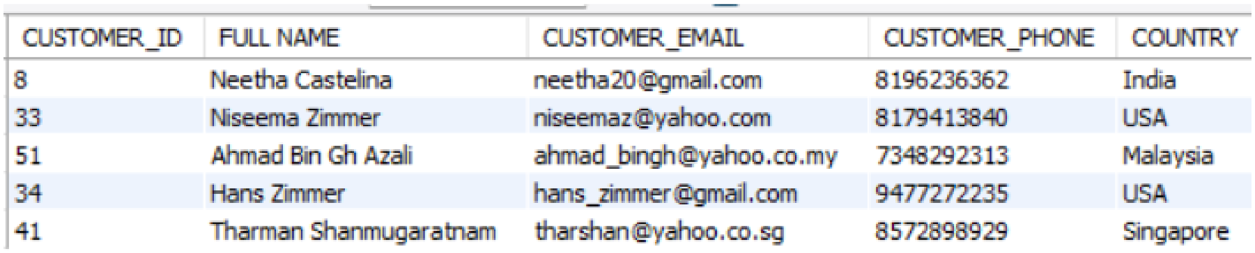
[note: tables to be used - online_customer, address, order_header]

# SOLUTION:

```
SELECT

oc.CUSTOMER_ID,

CONCAT(oc.CUSTOMER_FNAME, ' ', oc.CUSTOMER_LNAME) 'FULL NAME',

oc.CUSTOMER_EMAIL,

oc.CUSTOMER_PHONE,

ad.COUNTRY

FROM ONLINE_CUSTOMER oc

INNER JOIN ADDRESS ad
```

```
ON oc.ADDRESS_ID = ad.ADDRESS_ID

INNER JOIN ORDER_HEADER oh

ON oc.CUSTOMER_ID = oh.CUSTOMER_ID

WHERE oh.ORDER_STATUS = 'CANCELED'
```

# OUTPUT



| CUSTOMER_ID | FULL NAME | CUSTOMER_EMAIL | CUSTOMER_PHONE | COUNTRY |
|---|---|---|---|---|
| 8 | Neetha Castelina | neetha20@gmail.com | 8196236362 | India |
| 33 | Niseema Zimmer | niseemaz@yahoo.com | 8179413840 | USA |
| 51 | Ahmad Bin Gh Azali | ahmad_bingh@yahoo.co.my | 7348292313 | Malaysia |
| 34 | Hans Zimmer | hans_zimmer@gmail.com | 9477272235 | USA |
| 41 | Tharman Shanmugaratnam | tharshan@yahoo.co.sg | 8572898929 | Singapore |

5. Write a query to display shipper name, city to which it is catering, number of customer catered by the shipper in the city and number of consignments delivered to that city for shipper dhl

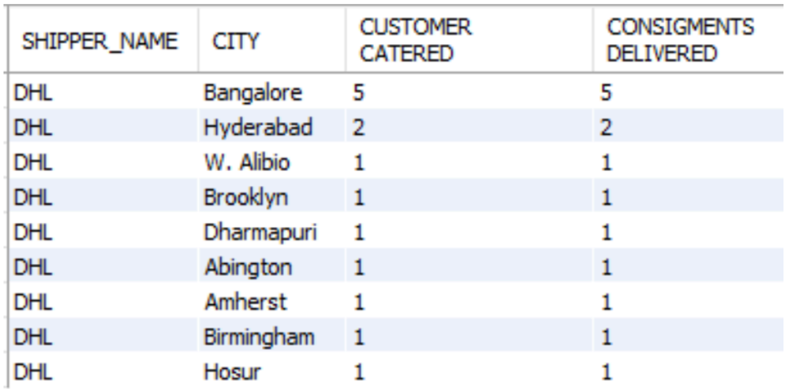
[note: tables to be used -shipper, online_customer, address, order_header]

# SOLUTION:

```
SELECT

sh.SHIPPER_NAME,

ad.CITY,

COUNT(oh.CUSTOMER_ID) 'CUSTOMER CATERED',

COUNT(oh.ORDER_STATUS) 'CONSIGNMENTS DELIVERED'

FROM SHIPPER sh

INNER JOIN ORDER_HEADER oh
```

```
ON sh.SHIPPER_ID = oh.SHIPPER_ID

INNER JOIN ONLINE_CUSTOMER oc

ON oh.CUSTOMER_ID = oc.CUSTOMER_ID

INNER JOIN ADDRESS ad

ON oc.ADDRESS_ID = ad.ADDRESS_ID

WHERE oh.ORDER_STATUS = 'SHIPPED'

AND sh.SHIPPER_NAME = 'DHL'

GROUP BY  sh.SHIPPER_NAME, ad.CITY
```

# OUTPUT



| SHIPPER_NAME | CITY | CUSTOMER CATERED | CONSIGMENTS DELIVERED |
|---|---|---|---|
| DHL | Bangalore | 5 | 5 |
| DHL | Hyderabad | 2 | 2 |
| DHL | W. Alibio | 1 | 1 |
| DHL | Brooklyn | 1 | 1 |
| DHL | Dharmapuri | 1 | 1 |
| DHL | Abington | 1 | 1 |
| DHL | Amherst | 1 | 1 |
| DHL | Birmingham | 1 | 1 |
| DHL | Hosur | 1 | 1 |

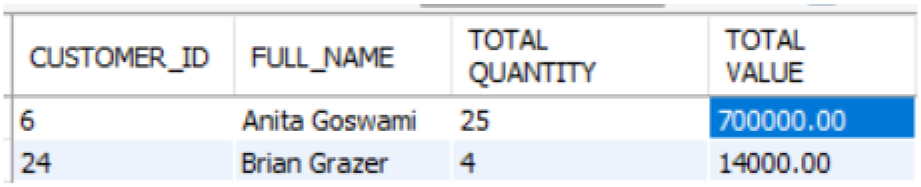
6.Write a query to display customer id, customer full name, total quantity and total value (quantity*price) shipped where mode of payment is cash and customer last name starts with 'g'

[note: tables to be used -online_customer, order_items, product, order_header]

# SOLUTION:

```
SELECT

oh.CUSTOMER_ID,

CONCAT(oc.CUSTOMER_FNAME, ' ',oc.CUSTOMER_LNAME) FULL_NAME,

SUM(oi.PRODUCT_QUANTITY) 'TOTAL QUANTITY',
(SUM(oi.PRODUCT_QUANTITY) * pr.PRODUCT_PRICE) 'TOTAL VALUE'
FROM ORDER_HEADER oh
INNER JOIN ORDER_ITEMS oi
ON oh.ORDER_ID = oi.ORDER_ID
INNER JOIN PRODUCT pr
ON oi.PRODUCT_ID = pr.PRODUCT_ID
INNER JOIN ONLINE_CUSTOMER oc
ON oh.CUSTOMER_ID = oc.CUSTOMER_ID
WHERE oh.ORDER_STATUS = 'SHIPPED'
AND oh.PAYMENT_MODE = 'CASH'
AND oc.CUSTOMER_LNAME LIKE 'G%'
GROUP BY oh.CUSTOMER_ID
```

## OUTPUT



| CUSTOMER_ID | FULL_NAME | TOTAL QUANTITY | TOTAL VALUE |
|---|---|---|---|
| 6 | Anita Goswami | 25 | 700000.00 |
| 24 | Brian Grazer | 4 | 14000.00 |

7.Write a query to display order_id and volume of biggest order (in terms of volume) that can fit in carton id 10

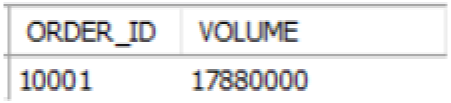
-- [note: tables to be used -carton, order_items, product]

## SOLUTION:

```
SELECT
          oi.ORDER_ID,
          MAX(pr.LEN * pr.WIDTH * pr.HEIGHT) VOLUME
```

```
  FROM PRODUCT pr
  INNER JOIN ORDER_ITEMS oi
  ON pr.PRODUCT_ID = oi.PRODUCT_ID
LEFT JOIN CARTON c
ON pr.LEN <= c.LEN
AND pr.WIDTH <= c.WIDTH
AND pr.HEIGHT <= c.HEIGHT
  WHERE c.CARTON_ID = 10
  GROUP BY oi.ORDER_ID
ORDER BY MAX(pr.LEN * pr.WIDTH * pr.HEIGHT) DESC
LIMIT 1
```

## OUTPUT



| ORDER_ID | VOLUME |
|---|---|
| 10001 | 17880000 |

8. Write a query to display product_id, product_desc, product_quantity_avail, quantity sold, and show inventory status of products as below as per below condition:

a. For electronics and computer categories,

i. If sales till date is zero then show 'no sales in past, give discount to reduce inventory',

ii. If inventory quantity is less than 10% of quantity sold, show 'low inventory, need to add inventory',

iii. If inventory quantity is less than 50% of quantity sold, show 'medium inventory, need to add some inventory',

iv. If inventory quantity is more or equal to 50% of quantity sold, show 'sufficient inventory'

b. For mobiles and watches categories,

i. If sales till date is zero then show 'no sales in past, give discount to reduce inventory',

ii. If inventory quantity is less than 20% of quantity sold, show 'low inventory, need to add inventory',

iii. If inventory quantity is less than 60% of quantity sold, show 'medium inventory, need to add some inventory',

iv. If inventory quantity is more or equal to 60% of quantity sold, show 'sufficient inventory'

c. Rest of the categories,

i. If sales till date is zero then show 'no sales in past, give discount to reduce inventory',

ii. If inventory quantity is less than 30% of quantity sold, show 'low inventory, need to add inventory',

iii. If inventory quantity is less than 70% of quantity sold, show 'medium inventory, need to add some inventory',

iv. If inventory quantity is more or equal to 70% of quantity sold, show 'sufficient inventory'

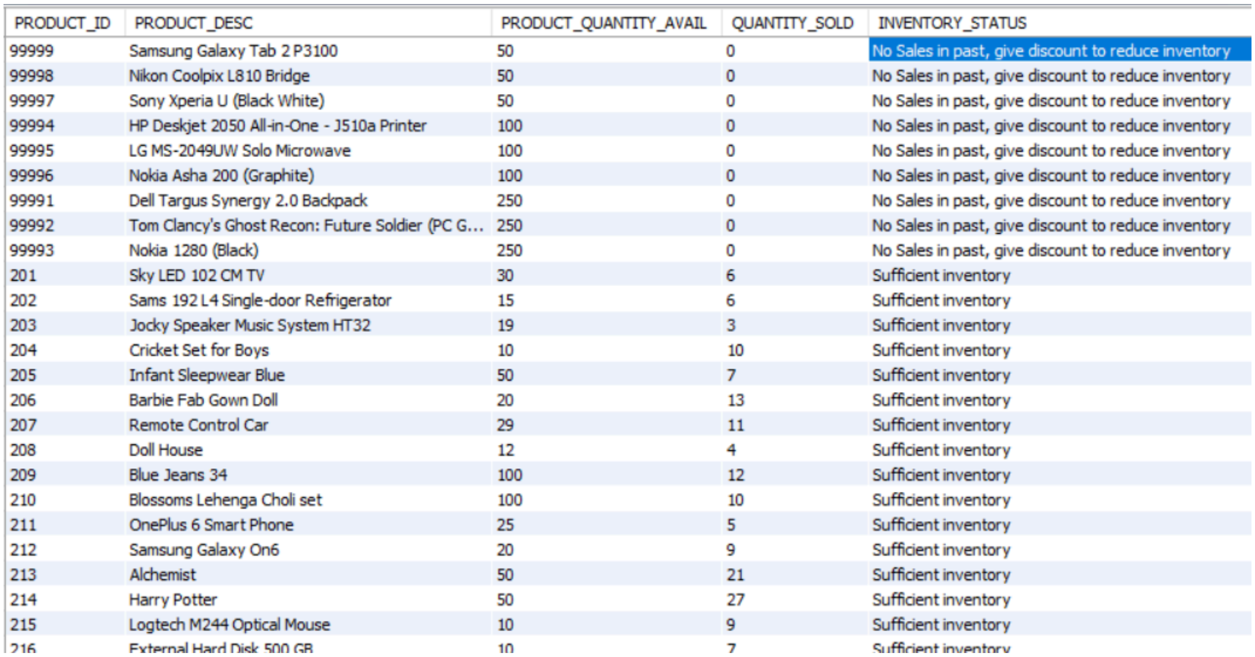
[note: tables to be used -product, product_class, order_items] (use sub-query)

## SOLUTION:

```
SELECT

pr.PRODUCT_ID,

pr.PRODUCT_DESC,

pr.PRODUCT_QUANTITY_AVAIL,

po.QUANTITY_SOLD,

CASE

WHEN pc.PRODUCT_CLASS_DESC IN('ELECTRONICS', 'COMPUTER')

THEN
```

```
     CASE WHEN po.QUANTITY_SOLD = 0
    THEN 'No Sales in past, give discount to reduce inventory'
    WHEN pr.PRODUCT_QUANTITY_AVAIL < (po.QUANTITY_SOLD * 0.10)
     THEN  'Low inventory, need to add inventory'
     WHEN pr.PRODUCT_QUANTITY_AVAIL >= (po.QUANTITY_SOLD * 0.50)
     THEN 'Sufficient inventory'
    END
WHEN pc.PRODUCT_CLASS_DESC IN('MOBILES', 'WATCHES')
THEN
     CASE WHEN po.QUANTITY_SOLD = 0
    THEN 'No Sales in past, give discount to reduce inventory'
     WHEN pr.PRODUCT_QUANTITY_AVAIL < (po.QUANTITY_SOLD * 0.20)
     THEN  'Low inventory, need to add inventory'
     WHEN pr.PRODUCT_QUANTITY_AVAIL >= (po.QUANTITY_SOLD * 0.60)
     THEN 'Sufficient inventory'
    END
ELSE
     CASE WHEN po.QUANTITY_SOLD = 0
    THEN 'No Sales in past, give discount to reduce inventory'
     WHEN pr.PRODUCT_QUANTITY_AVAIL < (po.QUANTITY_SOLD * 0.30)
     THEN  'Low inventory, need to add inventory'
     WHEN pr.PRODUCT_QUANTITY_AVAIL >= (po.QUANTITY_SOLD * 0.70)
     THEN 'Sufficient inventory'
    END

END INVENTORY_STATUS

FROM PRODUCT pr

INNER JOIN (

               SELECT

               pr.PRODUCT_ID,

               pr.PRODUCT_DESC,

               SUM(COALESCE(oi.PRODUCT_QUANTITY,0))
QUANTITY_SOLD

               FROM PRODUCT pr

               LEFT JOIN ORDER_ITEMS oi

               ON pr.PRODUCT_ID = oi.PRODUCT_ID

               GROUP BY pr.PRODUCT_ID, pr.PRODUCT_DESC) po

ON pr.PRODUCT_ID = po.PRODUCT_ID

INNER JOIN PRODUCT_CLASS pc

ON pr.PRODUCT_CLASS_CODE = pc.PRODUCT_CLASS_CODE
```

## OUTPUT



| PRODUCT_ID | PRODUCT_DESC | PRODUCT_QUANTITY_AVAIL | QUANTITY_SOLD | INVENTORY_STATUS |
|---|---|---|---|---|
| 99999 | Samsung Galaxy Tab 2 P3100 | 50 | 0 | No Sales in past, give discount to reduce inventory |
| 99998 | Nikon Coolpix L810 Bridge | 50 | 0 | No Sales in past, give discount to reduce inventory |
| 99997 | Sony Xperia U (Black White) | 50 | 0 | No Sales in past, give discount to reduce inventory |
| 99994 | HP Deskjet 2050 All-in-One - J510a Printer | 100 | 0 | No Sales in past, give discount to reduce inventory |
| 99995 | LG MS-2049UW Solo Microwave | 100 | 0 | No Sales in past, give discount to reduce inventory |
| 99996 | Nokia Asha 200 (Graphite) | 100 | 0 | No Sales in past, give discount to reduce inventory |
| 99991 | Dell Targus Synergy 2.0 Backpack | 250 | 0 | No Sales in past, give discount to reduce inventory |
| 99992 | Tom Clancy's Ghost Recon: Future Soldier (PC G... | 250 | 0 | No Sales in past, give discount to reduce inventory |
| 99993 | Nokia 1280 (Black) | 250 | 0 | No Sales in past, give discount to reduce inventory |
| 201 | Sky LED 102 CM TV | 30 | 6 | Sufficient inventory |
| 202 | Sams 192 L4 Single-door Refrigerator | 15 | 6 | Sufficient inventory |
| 203 | Jocky Speaker Music System HT32 | 19 | 3 | Sufficient inventory |
| 204 | Cricket Set for Boys | 10 | 10 | Sufficient inventory |
| 205 | Infant Sleepwear Blue | 50 | 7 | Sufficient inventory |
| 206 | Barbie Fab Gown Doll | 20 | 13 | Sufficient inventory |
| 207 | Remote Control Car | 29 | 11 | Sufficient inventory |
| 208 | Doll House | 12 | 4 | Sufficient inventory |
| 209 | Blue Jeans 34 | 100 | 12 | Sufficient inventory |
| 210 | Blossoms Lehenga Choli set | 100 | 10 | Sufficient inventory |
| 211 | OnePlus 6 Smart Phone | 25 | 5 | Sufficient inventory |
| 212 | Samsung Galaxy On6 | 20 | 9 | Sufficient inventory |
| 213 | Alchemist | 50 | 21 | Sufficient inventory |
| 214 | Harry Potter | 50 | 27 | Sufficient inventory |
| 215 | Logtech M244 Optical Mouse | 10 | 9 | Sufficient inventory |
| 216 | External Hard Disk 500 GB | 10 | 7 | Sufficient inventory |

9.Write a query to display product_id, product_desc and total quantity of products which are sold together with product id 201 and are not shipped to city bangalore and new delhi. Display the output in descending order concerning tot_qty.(use sub-query)

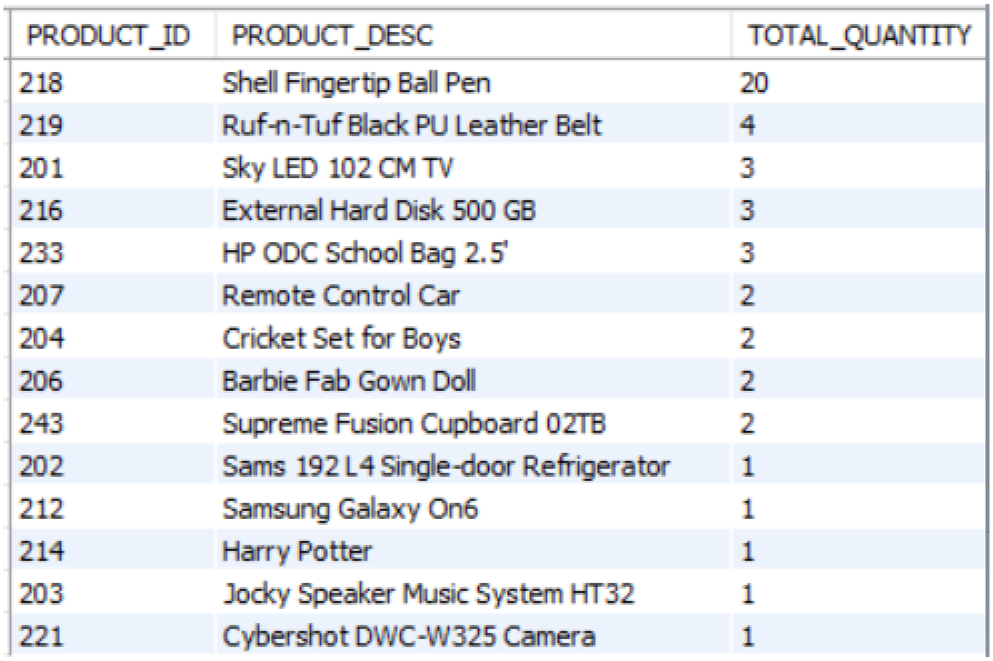
[note: tables to be used -order_items,product,order_header, online_customer, address]

# Solution:

```
SELECT

po.PRODUCT_ID,

pr.PRODUCT_DESC,

po.TOTAL_QUANTITY

FROM(

     SELECT

     oi.PRODUCT_ID,

     SUM(oi.PRODUCT_QUANTITY) TOTAL_QUANTITY

     FROM ORDER_ITEMS oi
```

```
    INNER JOIN ORDER_HEADER oh

    ON oi.ORDER_ID = oh.ORDER_ID

    INNER JOIN ONLINE_CUSTOMER oc

    ON oh.CUSTOMER_ID = oc.CUSTOMER_ID

    INNER JOIN ADDRESS ad

    ON oc.ADDRESS_ID = ad.ADDRESS_ID

    WHERE oi.ORDER_ID IN(SELECT ORDER_ID FROM ORDER_ITEMS WHERE
PRODUCT_ID = 201)

    AND ad.CITY NOT IN('BANGALORE', 'NEW DELHI')

     GROUP BY oi.PRODUCT_ID

) po

INNER JOIN PRODUCT  pr

ON po.PRODUCT_ID = pr.PRODUCT_ID

ORDER BY po.TOTAL_QUANTITY DESC
```

## OUTPUT



| PRODUCT_ID | PRODUCT_DESC | TOTAL_QUANTITY |
|---|---|---|
| 218 | Shell Fingertip Ball Pen | 20 |
| 219 | Ruf-n-Tuf Black PU Leather Belt | 4 |
| 201 | Sky LED 102 CM TV | 3 |
| 216 | External Hard Disk 500 GB | 3 |
| 233 | HP ODC School Bag 2.5' | 3 |
| 207 | Remote Control Car | 2 |
| 204 | Cricket Set for Boys | 2 |
| 206 | Barbie Fab Gown Doll | 2 |
| 243 | Supreme Fusion Cupboard 02TB | 2 |
| 202 | Sams 192 L4 Single-door Refrigerator | 1 |
| 212 | Samsung Galaxy On6 | 1 |
| 214 | Harry Potter | 1 |
| 203 | Jocky Speaker Music System HT32 | 1 |
| 221 | Cybershot DWC-W325 Camera | 1 |

10.Write a query to display the order_id,customer_id and customer fullname and total quantity of products shipped for order ids which are even and shipped to address where pincode is not starting with "5"

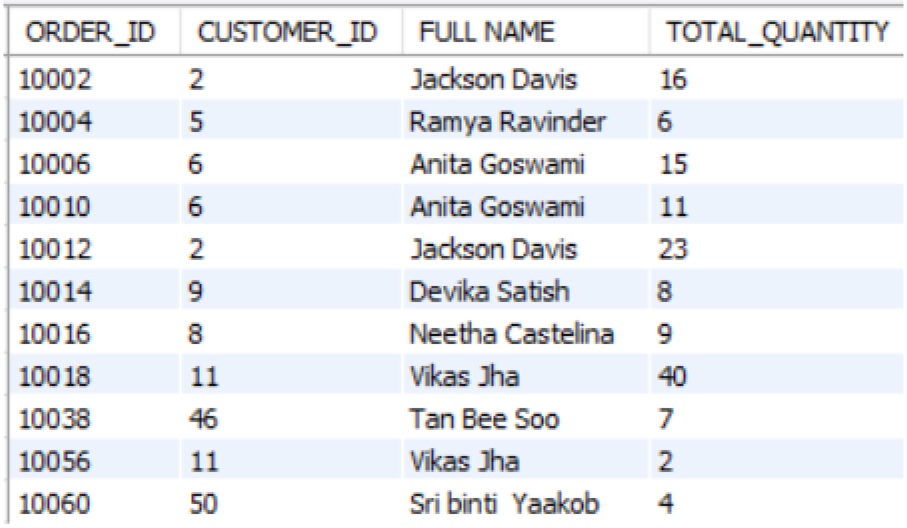
**Solution:**

[note: tables to be used - online_customer,order_header, order_items, address]

## Solution:

```
SELECT

oca.ORDER_ID,

oc.CUSTOMER_ID,

CONCAT(oc.CUSTOMER_FNAME, ' ', oc.CUSTOMER_LNAME) 'FULL NAME',

oca.TOTAL_QUANTITY

FROM (

          SELECT oh.ORDER_ID,oh.CUSTOMER_ID,

                  SUM(oi.PRODUCT_QUANTITY) TOTAL_QUANTITY

          FROM ORDER_HEADER oh

        INNER JOIN ORDER_ITEMS oi

        ON oh.ORDER_ID = oi.ORDER_ID

        INNER JOIN ONLINE_CUSTOMER oc

        ON oh.CUSTOMER_ID = oc.CUSTOMER_ID

        INNER JOIN ADDRESS ad

        ON oc.ADDRESS_ID = ad.ADDRESS_ID

        WHERE oh.ORDER_STATUS = 'SHIPPED'

        AND  oh.ORDER_ID % 2 = 0

        AND ad.PINCODE LIKE '5%'

        GROUP BY oh.ORDER_ID,oh.CUSTOMER_ID

        ) oca

INNER JOIN ONLINE_CUSTOMER oc

ON oca.CUSTOMER_ID = oc.CUSTOMER_ID
```

**OUTPUT**



| ORDER_ID | CUSTOMER_ID | FULL NAME | TOTAL_QUANTITY |
|---|---|---|---|
| 10002 | 2 | Jackson Davis | 16 |
| 10004 | 5 | Ramya Ravinder | 6 |
| 10006 | 6 | Anita Goswami | 15 |
| 10010 | 6 | Anita Goswami | 11 |
| 10012 | 2 | Jackson Davis | 23 |
| 10014 | 9 | Devika Satish | 8 |
| 10016 | 8 | Neetha Castelina | 9 |
| 10018 | 11 | Vikas Jha | 40 |
| 10038 | 46 | Tan Bee Soo | 7 |
| 10056 | 11 | Vikas Jha | 2 |
| 10060 | 50 | Sri binti  Yaakob | 4 |