SQL Project

Coded

Isha Shukla

17 April 2024

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 categorisation rules:

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

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

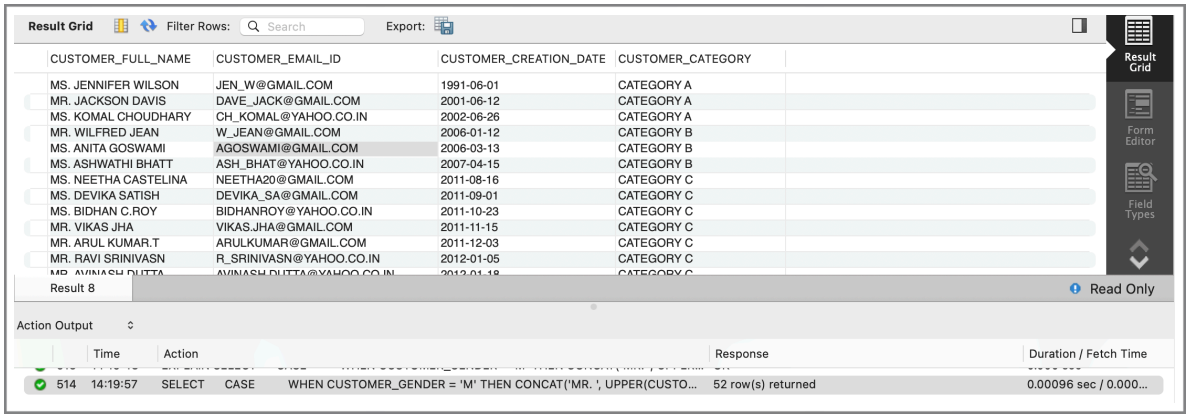
iii. If customer creation date year>= 2011 then category c

Hint: Use case statement, no permanent change in table required. [note: tables to be used -online_customer table]

Solution:

```sql
SELECT
    CASE
        WHEN CUSTOMER_GENDER = 'M' THEN CONCAT('MR. ', UPPER(CUSTOMER_FNAME), ' ', UPPER(CUSTOMER_LNAME))
        WHEN CUSTOMER_GENDER = 'F' THEN CONCAT('MS. ', UPPER(CUSTOMER_FNAME), ' ', UPPER(CUSTOMER_LNAME))
    END AS CUSTOMER_FULL_NAME,
    UPPER(CUSTOMER_EMAIL) AS CUSTOMER_EMAIL_ID,
    CUSTOMER_CREATION_DATE,
    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'
        ELSE 'CATEGORY C'
    END AS CUSTOMER_CATEGORY
FROM
    ONLINE_CUSTOMER;
```

Output:



CUSTOMER_FULL_NAME	CUSTOMER_EMAIL_ID	CUSTOMER_CREATION_DATE	CUSTOMER_CATEGORY
MS. JENNIFER WILSON	JEN_W@GMAIL.COM	1991-06-01	CATEGORY A
MR. JACKSON DAVIS	DAVE_JACK@GMAIL.COM	2001-06-12	CATEGORY A
MS. KOMAL CHOUDHARY	CH_KOMAL@YAHOO.CO.IN	2002-06-26	CATEGORY A
MR. WILFRED JEAN	W_JEAN@GMAIL.COM	2006-01-12	CATEGORY B
MS. ANITA GOSWAMI	AGOSWAMI@GMAIL.COM	2006-03-13	CATEGORY B
MS. ASHWATHI BHATT	ASH_BHAT@YAHOO.CO.IN	2007-04-15	CATEGORY B
MS. NEETHA CASTELINA	NEETHA20@GMAIL.COM	2011-08-16	CATEGORY C
MS. DEVIKA SATISH	DEVIKA_SA@GMAIL.COM	2011-09-01	CATEGORY C
MS. BIDHAN C.ROY	BIDHANROY@YAHOO.CO.IN	2011-10-23	CATEGORY C
MR. VIKAS JHA	VIKAS.JHA@GMAIL.COM	2011-11-15	CATEGORY C
MR. ARUL KUMAR.T	ARULKUMAR@GMAIL.COM	2011-12-03	CATEGORY C
MR. RAVI SRINIVASN	R_SRINIVASN@YAHOO.CO.IN	2012-01-05	CATEGORY C

52 rows returned.

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

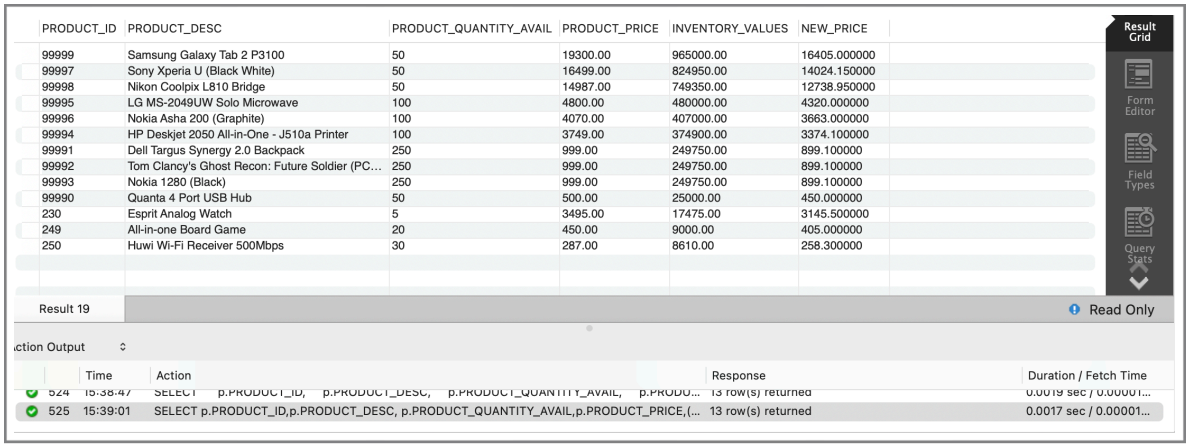
iii. If product price =< 10,000 then apply 10% discount

Hint: use case statement, no permanent change in table required. [note: tables to be used -product, order_items table]

Solution:

```
SELECT p.PRODUCT_ID,p.PRODUCT_DESC,
    p.PRODUCT_QUANTITY_AVAIL,p.PRODUCT_PRICE,
    (p.PRODUCT_QUANTITY_AVAIL*p.PRODUCT_PRICE) AS INVENTORY_VALUES,
    CASE
            WHEN PRODUCT_PRICE > 20000 THEN (PRODUCT_PRICE-(PRODUCT_PRICE*(20/100)))
            WHEN PRODUCT_PRICE > 10000 THEN (PRODUCT_PRICE-(PRODUCT_PRICE*(15/100)))
            ELSE (PRODUCT_PRICE-(PRODUCT_PRICE*(10/100)))
    END AS NEW_PRICE
FROM
    PRODUCT p
LEFT JOIN
    ORDER_ITEMS oi on p.PRODUCT_ID=oi.PRODUCT_ID
WHERE
    oi.PRODUCT_ID is NULL
ORDER BY
    INVENTORY_VALUES DESC;
```

Output:



PRODUCT_ID	PRODUCT_DESC	PRODUCT_QUANTITY_AVAIL	PRODUCT_PRICE	INVENTORY_VALUES	NEW_PRICE
99999	Samsung Galaxy Tab 2 P3100	50	19300.00	965000.00	16405.000000
99997	Sony Xperia U (Black White)	50	16499.00	824950.00	14024.150000
99998	Nikon Coolpix L810 Bridge	50	14987.00	749350.00	12738.950000
99995	LG MS-2049UW Solo Microwave	100	4800.00	480000.00	4320.000000
99996	Nokia Asha 200 (Graphite)	100	4070.00	407000.00	3663.000000
99994	HP Deskjet 2050 All-in-One - J510a Printer	100	3749.00	374900.00	3374.100000
99991	Dell Targus Synergy 2.0 Backpack	250	999.00	249750.00	899.100000
99992	Tom Clancy's Ghost Recon: Future Soldier (PC…	250	999.00	249750.00	899.100000
99993	Nokia 1280 (Black)	250	999.00	249750.00	899.100000
99990	Quanta 4 Port USB Hub	50	500.00	25000.00	450.000000
230	Esprit Analog Watch	5	3495.00	17475.00	3145.500000
249	All-in-one Board Game	20	450.00	9000.00	405.000000
250	Huwi Wi-Fi Receiver 500Mbps	30	287.00	8610.00	258.300000

Result Grid | Form Editor | Field Types | Query Stats

Result 19 | Read Only

ction Output

	Time	Action	Response	Duration / Fetch Time
524	15:38:47	SELECT p.PRODUCT_ID, p.PRODUCT_DESC, p.PRODUCT_QUANTITY_AVAIL, p.PRODU…	13 row(s) returned	0.0019 sec / 0.00001…
525	15:39:01	SELECT p.PRODUCT_ID,p.PRODUCT_DESC, p.PRODUCT_QUANTITY_AVAIL,p.PRODUCT_PRICE,(…	13 row(s) returned	0.0017 sec / 0.00001…

13 rows returned.

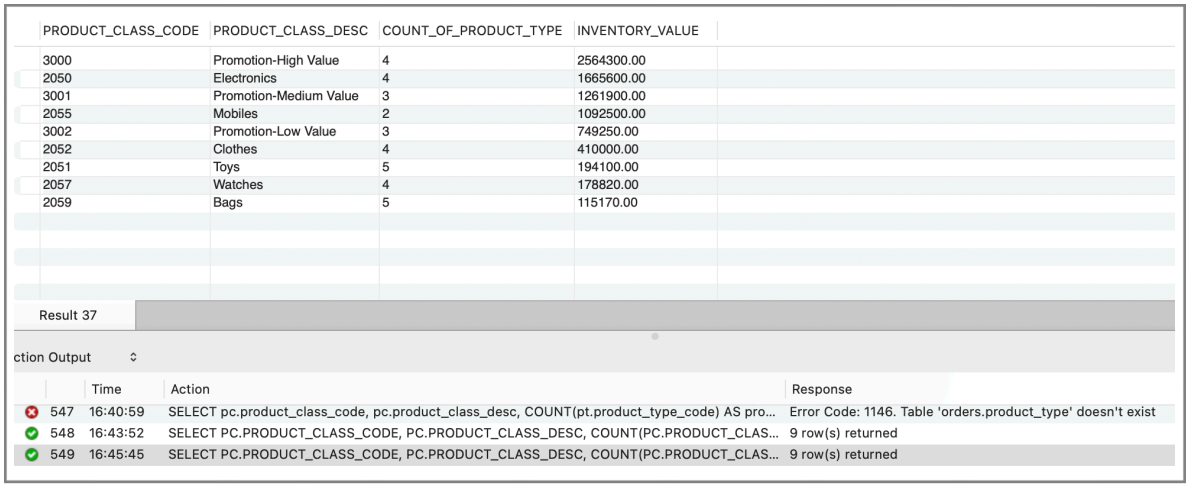
3. Write a query to display product_class_code, product_class_description, count of product type in each product class, and inventory value (p.product_quantity_avail*p.product_price). Information should be displayed for only those product_class_code that have more than 1,00,000 inventory value. sort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

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

Solution:

```
SELECT PC.PRODUCT_CLASS_CODE, PC.PRODUCT_CLASS_DESC, COUNT(PC.PRODUCT_CLASS_CODE) AS COUNT_OF_PRODUCT_TYPE,
    SUM((P.PRODUCT_QUANTITY_AVAIL*P.PRODUCT_PRICE)) AS INVENTORY_VALUE
FROM
    PRODUCT P
INNER JOIN
    PRODUCT_CLASS PC ON P.PRODUCT_CLASS_CODE = PC.PRODUCT_CLASS_CODE
GROUP BY
    PC.PRODUCT_CLASS_CODE, PC.PRODUCT_CLASS_DESC
HAVING
    INVENTORY_VALUE >100000
ORDER BY
    INVENTORY_VALUE DESC;
```

Output:



PRODUCT_CLASS_CODE	PRODUCT_CLASS_DESC	COUNT_OF_PRODUCT_TYPE	INVENTORY_VALUE
3000	Promotion-High Value	4	2564300.00
2050	Electronics	4	1665600.00
3001	Promotion-Medium Value	3	1261900.00
2055	Mobiles	2	1092500.00
3002	Promotion-Low Value	3	749250.00
2052	Clothes	4	410000.00
2051	Toys	5	194100.00
2057	Watches	4	178820.00
2059	Bags	5	115170.00

9 rows returned.

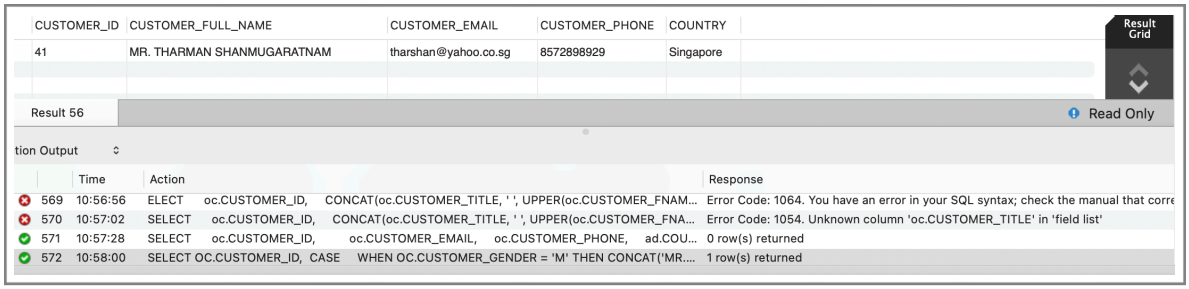
4. Write a query to display customer_id, full name, customer_email, customer_phone and country of customers who have cancelled all the orders placed by them(use sub-query)

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

Solution:

```sql
SELECT OC.CUSTOMER_ID,
    CASE
            WHEN OC.CUSTOMER_GENDER = 'M' THEN CONCAT('MR. ', UPPER(OC.CUSTOMER_FNAME), ' ', UPPER(OC.CUSTOMER_LNAME))
            WHEN OC.CUSTOMER_GENDER = 'F' THEN CONCAT('MS. ', UPPER(OC.CUSTOMER_FNAME), ' ', UPPER(OC.CUSTOMER_LNAME))
    END AS CUSTOMER_FULL_NAME,
    OC.CUSTOMER_EMAIL,OC.CUSTOMER_PHONE, A.COUNTRY
FROM
    ONLINE_CUSTOMER OC
JOIN
    ADDRESS A ON OC.ADDRESS_ID=A.ADDRESS_ID
WHERE
    OC.CUSTOMER_ID IN (
    SELECT OH.CUSTOMER_ID
    FROM
        ORDER_HEADER OH
    WHERE
        OH.ORDER_STATUS='Cancelled'
    GROUP BY
        OH.CUSTOMER_ID
    HAVING
        COUNT(*)=(SELECT COUNT(*) FROM ORDER_HEADER WHERE CUSTOMER_ID = oh.CUSTOMER_ID)
);
```

Output:



CUSTOMER_ID	CUSTOMER_FULL_NAME	CUSTOMER_EMAIL	CUSTOMER_PHONE	COUNTRY
41	MR. THARMAN SHANMUGARATNAM	tharshan@yahoo.co.sg	8572898929	Singapore

Result 56 — Read Only

tion Output

	Time	Action	Response
569	10:56:56	ELECT oc.CUSTOMER_ID, CONCAT(oc.CUSTOMER_TITLE, ' ', UPPER(oc.CUSTOMER_FNAM...	Error Code: 1064. You have an error in your SQL syntax; check the manual that corre
570	10:57:02	SELECT oc.CUSTOMER_ID, CONCAT(oc.CUSTOMER_TITLE, ' ', UPPER(oc.CUSTOMER_FNA...	Error Code: 1054. Unknown column 'oc.CUSTOMER_TITLE' in 'field list'
571	10:57:28	SELECT oc.CUSTOMER_ID, oc.CUSTOMER_EMAIL, oc.CUSTOMER_PHONE, ad.COU...	0 row(s) returned
572	10:58:00	SELECT OC.CUSTOMER_ID, CASE WHEN OC.CUSTOMER_GENDER = 'M' THEN CONCAT('MR....	1 row(s) returned

1 row returned.

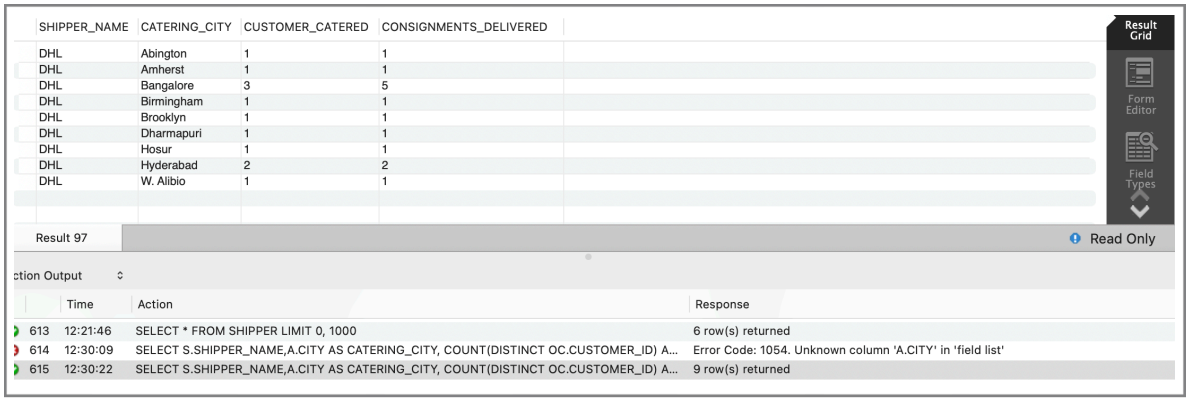
5. Write a query to display shipper name, city to which it is catering, number of customer catered by the shipper in the city and number of consignments delivered to that city for shipper dhl

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

Solution:

```sql
SELECT S.SHIPPER_NAME,A.CITY AS CATERING_CITY, COUNT(DISTINCT OC.CUSTOMER_ID) AS CUSTOMER_CATERED,
    COUNT(*) AS CONSIGNMENTS_DELIVERED
FROM
    SHIPPER S
JOIN
    ORDER_HEADER OH ON S.SHIPPER_ID=OH.SHIPPER_ID
JOIN
    ONLINE_CUSTOMER OC ON OH.CUSTOMER_ID=OC.CUSTOMER_ID
JOIN
    ADDRESS A ON OC.ADDRESS_ID=A.ADDRESS_ID
WHERE
    S.SHIPPER_NAME = "DHL"
GROUP BY
    S.SHIPPER_NAME,A.CITY;
```

Output:



SHIPPER_NAME	CATERING_CITY	CUSTOMER_CATERED	CONSIGNMENTS_DELIVERED
DHL	Abington	1	1
DHL	Amherst	1	1
DHL	Bangalore	3	5
DHL	Birmingham	1	1
DHL	Brooklyn	1	1
DHL	Dharmapuri	1	1
DHL	Hosur	1	1
DHL	Hyderabad	2	2
DHL	W. Alibio	1	1

Result Grid

Form Editor

Field Types

Result 97

Read Only

ction Output

	Time	Action	Response
613	12:21:46	SELECT * FROM SHIPPER LIMIT 0, 1000	6 row(s) returned
614	12:30:09	SELECT S.SHIPPER_NAME,A.CITY AS CATERING_CITY, COUNT(DISTINCT OC.CUSTOMER_ID) A...	Error Code: 1054. Unknown column 'A.CITY' in 'field list'
615	12:30:22	SELECT S.SHIPPER_NAME,A.CITY AS CATERING_CITY, COUNT(DISTINCT OC.CUSTOMER_ID) A...	9 row(s) returned

9 rows returned.

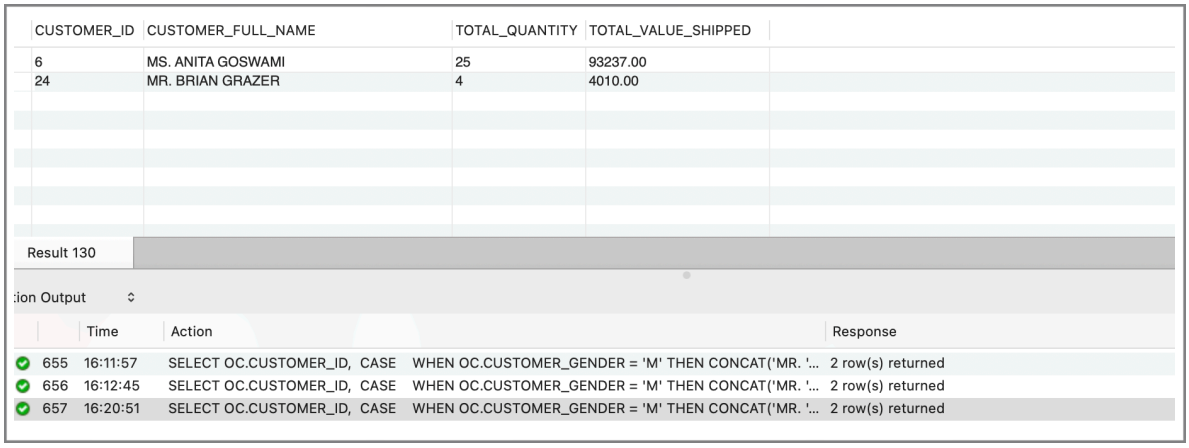
6. Write a query to display customer id, customer full name, total quantity and total value (quantity*price) shipped where mode of payment is cash and customer last name starts with 'g'

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

Solution:

```
SELECT OC.CUSTOMER_ID,
    CASE
        WHEN OC.CUSTOMER_GENDER = 'M' THEN CONCAT('MR. ', UPPER(OC.CUSTOMER_FNAME), ' ', UPPER(OC.CUSTOMER_LNAME))
        WHEN OC.CUSTOMER_GENDER = 'F' THEN CONCAT('MS. ', UPPER(OC.CUSTOMER_FNAME), ' ', UPPER(OC.CUSTOMER_LNAME))
    END AS CUSTOMER_FULL_NAME,
    SUM(OI.PRODUCT_QUANTITY) AS TOTAL_QUANTITY,
    SUM(OI.PRODUCT_QUANTITY*P.PRODUCT_PRICE) AS TOTAL_VALUE_SHIPPED
FROM
    ONLINE_CUSTOMER OC
JOIN
    ORDER_HEADER OH ON OC.CUSTOMER_ID = OH.CUSTOMER_ID
JOIN
    ORDER_ITEMS OI ON OH.ORDER_ID = OI.ORDER_ID
JOIN
    PRODUCT P ON OI.PRODUCT_ID = P.PRODUCT_ID
WHERE
    PAYMENT_MODE = 'Cash'
    AND CUSTOMER_LNAME LIKE 'G%'
GROUP BY
    OC.CUSTOMER_ID, OC.CUSTOMER_FNAME, OC.CUSTOMER_LNAME;
```

Output:



CUSTOMER_ID	CUSTOMER_FULL_NAME	TOTAL_QUANTITY	TOTAL_VALUE_SHIPPED
6	MS. ANITA GOSWAMI	25	93237.00
24	MR. BRIAN GRAZER	4	4010.00

Result 130

ion Output

	Time	Action	Response
655	16:11:57	SELECT OC.CUSTOMER_ID, CASE WHEN OC.CUSTOMER_GENDER = 'M' THEN CONCAT('MR. '...	2 row(s) returned
656	16:12:45	SELECT OC.CUSTOMER_ID, CASE WHEN OC.CUSTOMER_GENDER = 'M' THEN CONCAT('MR. '...	2 row(s) returned
657	16:20:51	SELECT OC.CUSTOMER_ID, CASE WHEN OC.CUSTOMER_GENDER = 'M' THEN CONCAT('MR. '...	2 row(s) returned

2 rows returned.

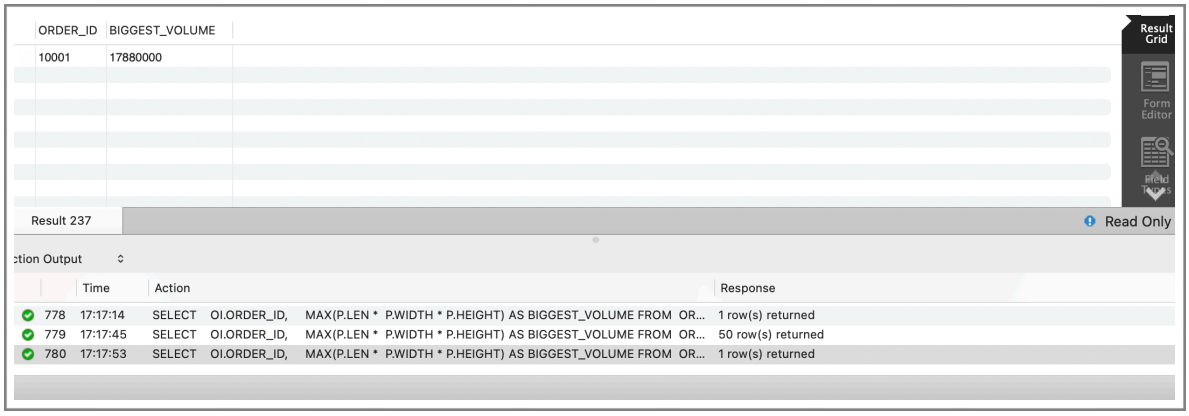
7. Write a query to display order_id and volume of biggest order (in terms of volume) that can fit in carton id 10

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

Solution:

```sql
SELECT
    OI.ORDER_ID,
    MAX(P.LEN *  P.WIDTH * P.HEIGHT) AS BIGGEST_VOLUME
FROM
    ORDER_ITEMS OI
JOIN
    PRODUCT P ON OI.PRODUCT_ID = P.PRODUCT_ID
JOIN
    CARTON C ON P.LEN <= C.LEN AND P.WIDTH <= C.WIDTH AND P.HEIGHT <= C.HEIGHT
WHERE
    C.CARTON_ID=10
GROUP BY
    OI.ORDER_ID
ORDER BY
    BIGGEST_VOLUME DESC
LIMIT 1;
```

Output:



1 row returned.

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'

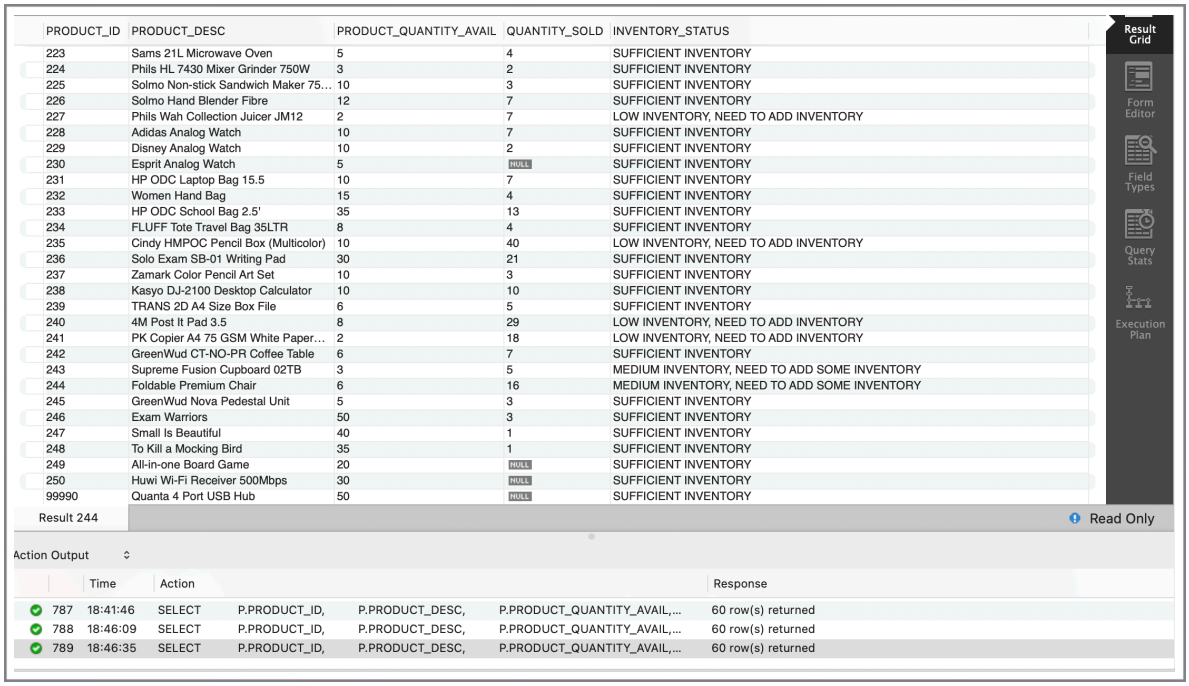
[note: tables to be used -product, product_class, order_items] (use sub-query)

Solution:

```
SELECT
    P.PRODUCT_ID,
    P.PRODUCT_DESC,
    P.PRODUCT_QUANTITY_AVAIL,
    SUM(OI.PRODUCT_QUANTITY) AS QUANTITY_SOLD,
    CASE
        WHEN PC.PRODUCT_CLASS_CODE IN ('Electronics', 'Computer') THEN
            CASE
                WHEN SUM(OI.PRODUCT_QUANTITY) = 0 THEN 'NO SALES IN PAST, GIVE DISCOUNT TO REDUCE INVENTORY'
                WHEN P.PRODUCT_QUANTITY_AVAIL < (0.1 * SUM(OI.PRODUCT_QUANTITY)) THEN 'LOW INVENTORY, NEED TO ADD INVENTORY'
                WHEN P.PRODUCT_QUANTITY_AVAIL < (0.5 * SUM(OI.PRODUCT_QUANTITY)) THEN 'MEDIUM INVENTORY, NEED TO ADD SOME INVENTORY'
                ELSE 'SUFFICIENT INVENTORY'
            END
        WHEN PC.PRODUCT_CLASS_CODE IN ('Mobiles', 'Watches') THEN
            CASE
                WHEN SUM(OI.PRODUCT_QUANTITY) = 0 THEN 'NO SALES IN PAST, GIVE DISCOUNT TO REDUCE INVENTORY'
                WHEN P.PRODUCT_QUANTITY_AVAIL < (0.2 * SUM(OI.PRODUCT_QUANTITY)) THEN 'LOW INVENTORY, NEED TO ADD INVENTORY'
                WHEN P.PRODUCT_QUANTITY_AVAIL < (0.6 * SUM(OI.PRODUCT_QUANTITY)) THEN 'MEDIUM INVENTORY, NEED TO ADD SOME INVENTORY'
                ELSE 'SUFFICIENT INVENTORY'
            END
        ELSE
            CASE
                WHEN SUM(OI.PRODUCT_QUANTITY) = 0 THEN 'NO SALES IN PAST, GIVE DISCOUNT TO REDUCE INVENTORY'
```

```
                ELSE 'SUFFICIENT INVENTORY'
            END
        ELSE
            CASE
                WHEN SUM(OI.PRODUCT_QUANTITY) = 0 THEN 'NO SALES IN PAST, GIVE DISCOUNT TO REDUCE INVENTORY'
                WHEN P.PRODUCT_QUANTITY_AVAIL < (0.3 * SUM(OI.PRODUCT_QUANTITY)) THEN 'LOW INVENTORY, NEED TO ADD INVENTORY'
                WHEN P.PRODUCT_QUANTITY_AVAIL < (0.7 * SUM(OI.PRODUCT_QUANTITY)) THEN 'MEDIUM INVENTORY, NEED TO ADD SOME INVENTORY'
                ELSE 'SUFFICIENT INVENTORY'
            END
    END AS INVENTORY_STATUS
FROM
    PRODUCT P
JOIN
    PRODUCT_CLASS PC ON P.PRODUCT_CLASS_CODE = PC.PRODUCT_CLASS_CODE
LEFT JOIN
    ORDER_ITEMS OI ON P.PRODUCT_ID = OI.PRODUCT_ID
GROUP BY
    P.PRODUCT_ID, P.PRODUCT_DESC, P.PRODUCT_QUANTITY_AVAIL, PC.PRODUCT_CLASS_CODE;
```

Output:



PRODUCT_ID	PRODUCT_DESC	PRODUCT_QUANTITY_AVAIL	QUANTITY_SOLD	INVENTORY_STATUS
223	Sams 21L Microwave Oven	5	4	SUFFICIENT INVENTORY
224	Phils HL 7430 Mixer Grinder 750W	3	2	SUFFICIENT INVENTORY
225	Solmo Non-stick Sandwich Maker 75…	10	3	SUFFICIENT INVENTORY
226	Solmo Hand Blender Fibre	12	7	SUFFICIENT INVENTORY
227	Phils Wah Collection Juicer JM12	2	7	LOW INVENTORY, NEED TO ADD INVENTORY
228	Adidas Analog Watch	10	7	SUFFICIENT INVENTORY
229	Disney Analog Watch	10	2	SUFFICIENT INVENTORY
230	Esprit Analog Watch	5	NULL	SUFFICIENT INVENTORY
231	HP ODC Laptop Bag 15.5	10	7	SUFFICIENT INVENTORY
232	Women Hand Bag	15	4	SUFFICIENT INVENTORY
233	HP ODC School Bag 2.5'	35	13	SUFFICIENT INVENTORY
234	FLUFF Tote Travel Bag 35LTR	8	4	SUFFICIENT INVENTORY
235	Cindy HMPOC Pencil Box (Multicolor)	10	40	LOW INVENTORY, NEED TO ADD INVENTORY
236	Solo Exam SB-01 Writing Pad	30	21	SUFFICIENT INVENTORY
237	Zamark Color Pencil Art Set	10	3	SUFFICIENT INVENTORY
238	Kasyo DJ-2100 Desktop Calculator	10	10	SUFFICIENT INVENTORY
239	TRANS 2D A4 Size Box File	6	5	SUFFICIENT INVENTORY
240	4M Post It Pad 3.5	8	29	LOW INVENTORY, NEED TO ADD INVENTORY
241	PK Copier A4 75 GSM White Paper…	2	18	LOW INVENTORY, NEED TO ADD INVENTORY
242	GreenWud CT-NO-PR Coffee Table	6	7	SUFFICIENT INVENTORY
243	Supreme Fusion Cupboard 02TB	3	5	MEDIUM INVENTORY, NEED TO ADD SOME INVENTORY
244	Foldable Premium Chair	6	16	MEDIUM INVENTORY, NEED TO ADD SOME INVENTORY
245	GreenWud Nova Pedestal Unit	5	3	SUFFICIENT INVENTORY
246	Exam Warriors	50	3	SUFFICIENT INVENTORY
247	Small Is Beautiful	40	1	SUFFICIENT INVENTORY
248	To Kill a Mocking Bird	35	1	SUFFICIENT INVENTORY
249	All-in-one Board Game	20	NULL	SUFFICIENT INVENTORY
250	Huwi Wi-Fi Receiver 500Mbps	30	NULL	SUFFICIENT INVENTORY
99990	Quanta 4 Port USB Hub	50	NULL	SUFFICIENT INVENTORY

Result Grid · Form Editor · Field Types · Query Stats · Execution Plan

Result 244 · Read Only

Action Output

	Time	Action				Response
787	18:41:46	SELECT	P.PRODUCT_ID,	P.PRODUCT_DESC,	P.PRODUCT_QUANTITY_AVAIL,...	60 row(s) returned
788	18:46:09	SELECT	P.PRODUCT_ID,	P.PRODUCT_DESC,	P.PRODUCT_QUANTITY_AVAIL,...	60 row(s) returned
789	18:46:35	SELECT	P.PRODUCT_ID,	P.PRODUCT_DESC,	P.PRODUCT_QUANTITY_AVAIL,...	60 row(s) returned

60 rows returned.

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)

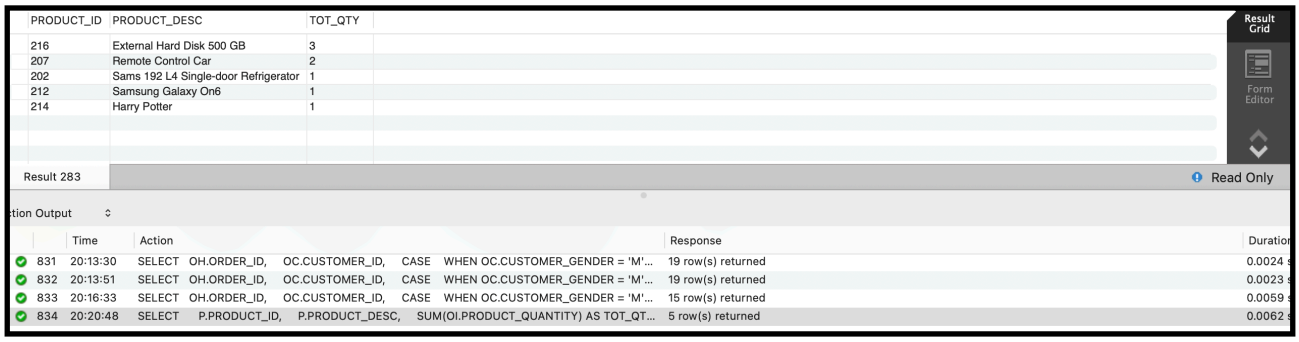
[note: tables to be used -order_items,product,order_header, online_customer, address]

Solution:

```sql
SELECT
    P.PRODUCT_ID,
    P.PRODUCT_DESC,
    SUM(OI.PRODUCT_QUANTITY) AS TOT_QTY
FROM
    ORDER_ITEMS OI
INNER JOIN
    PRODUCT P ON OI.PRODUCT_ID = P.PRODUCT_ID
WHERE
    OI.ORDER_ID IN (
        SELECT
            OI.ORDER_ID
        FROM
            ORDER_ITEMS OI
        JOIN
            ORDER_HEADER OH ON OI.ORDER_ID = OH.ORDER_ID
        JOIN
            ONLINE_CUSTOMER OC ON OH.CUSTOMER_ID = OC.CUSTOMER_ID
        JOIN
            ADDRESS A ON OC.ADDRESS_ID = A.ADDRESS_ID
        WHERE
            OI.PRODUCT_ID = 201
            AND OH.ORDER_STATUS = 'Shipped'
            AND A.CITY NOT IN ('New Delhi', 'Bangalore')
```

```
            AND A.CITY NOT IN ('New Delhi', 'Bangalore')
    )
    AND P.PRODUCT_ID != 201
GROUP BY
    P.PRODUCT_ID, P.PRODUCT_DESC
ORDER BY
    TOT_QTY DESC;
```

Output:



PRODUCT_ID	PRODUCT_DESC	TOT_QTY
216	External Hard Disk 500 GB	3
207	Remote Control Car	2
202	Sams 192 L4 Single-door Refrigerator	1
212	Samsung Galaxy On6	1
214	Harry Potter	1

Result 283 — Read Only

	Time	Action	Response	Duration
831	20:13:30	SELECT OH.ORDER_ID, OC.CUSTOMER_ID, CASE WHEN OC.CUSTOMER_GENDER = 'M'...	19 row(s) returned	0.0024 s
832	20:13:51	SELECT OH.ORDER_ID, OC.CUSTOMER_ID, CASE WHEN OC.CUSTOMER_GENDER = 'M'...	19 row(s) returned	0.0023 s
833	20:16:33	SELECT OH.ORDER_ID, OC.CUSTOMER_ID, CASE WHEN OC.CUSTOMER_GENDER = 'M'...	15 row(s) returned	0.0059 s
834	20:20:48	SELECT P.PRODUCT_ID, P.PRODUCT_DESC, SUM(OI.PRODUCT_QUANTITY) AS TOT_QT...	5 row(s) returned	0.0062 s

5 rows returned.

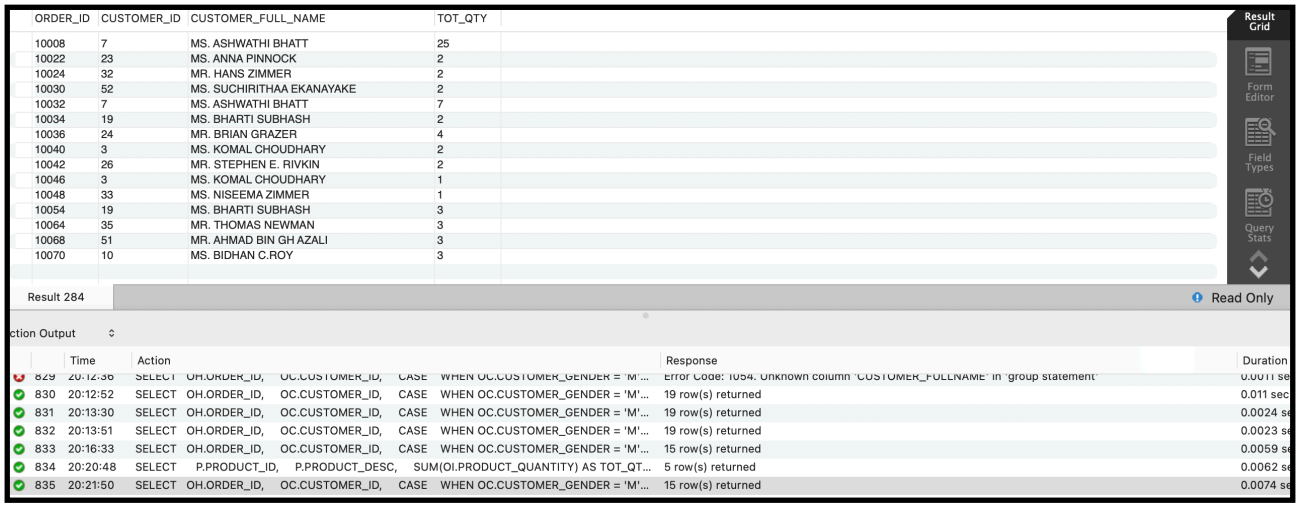
10. Write a query to display the order_id,customer_id and customer fullname and total quantity of products shipped for order ids which are even and shipped to address where pincode is not starting with "5"

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

Solution:

```sql
SELECT
    OH.ORDER_ID,
    OC.CUSTOMER_ID,
    CASE
        WHEN OC.CUSTOMER_GENDER = 'M' THEN CONCAT('MR. ', UPPER(OC.CUSTOMER_FNAME), ' ', UPPER(OC.CUSTOMER_LNAME))
        WHEN OC.CUSTOMER_GENDER = 'F' THEN CONCAT('MS. ', UPPER(OC.CUSTOMER_FNAME), ' ', UPPER(OC.CUSTOMER_LNAME))
    END AS CUSTOMER_FULL_NAME,
    SUM(OI.PRODUCT_QUANTITY) AS TOT_QTY
FROM
    ORDER_HEADER OH
INNER JOIN
    ONLINE_CUSTOMER OC ON OH.CUSTOMER_ID = OC.CUSTOMER_ID
INNER JOIN
    ADDRESS A ON OC.ADDRESS_ID = A.ADDRESS_ID
INNER JOIN
    ORDER_ITEMS OI ON OH.ORDER_ID = OI.ORDER_ID
WHERE
    OI.ORDER_ID % 2 = 0
    AND OH.ORDER_STATUS = 'Shipped'
    AND A.PINCODE NOT LIKE '5%'
GROUP BY
    OH.ORDER_ID, OC.CUSTOMER_ID,CUSTOMER_FULL_NAME;
```

Output:



ORDER_ID	CUSTOMER_ID	CUSTOMER_FULL_NAME	TOT_QTY
10008	7	MS. ASHWATHI BHATT	25
10022	23	MS. ANNA PINNOCK	2
10024	32	MR. HANS ZIMMER	2
10030	52	MS. SUCHIRITHAA EKANAYAKE	2
10032	7	MS. ASHWATHI BHATT	7
10034	19	MS. BHARTI SUBHASH	2
10036	24	MR. BRIAN GRAZER	4
10040	3	MS. KOMAL CHOUDHARY	2
10042	26	MR. STEPHEN E. RIVKIN	2
10046	3	MS. KOMAL CHOUDHARY	1
10048	33	MS. NISEEMA ZIMMER	1
10054	19	MS. BHARTI SUBHASH	3
10064	35	MR. THOMAS NEWMAN	3
10068	51	MR. AHMAD BIN GH AZALI	3
10070	10	MS. BIDHAN C.ROY	3

Result Grid · Form Editor · Field Types · Query Stats

Result 284 — Read Only

ction Output

	Time	Action	Response	Duration
829	20:12:36	SELECT OH.ORDER_ID, OC.CUSTOMER_ID, CASE WHEN OC.CUSTOMER_GENDER = 'M'...	Error Code: 1054. Unknown column 'CUSTOMER_FULLNAME' in 'group statement'	0.0011 se
830	20:12:52	SELECT OH.ORDER_ID, OC.CUSTOMER_ID, CASE WHEN OC.CUSTOMER_GENDER = 'M'...	19 row(s) returned	0.011 sec
831	20:13:30	SELECT OH.ORDER_ID, OC.CUSTOMER_ID, CASE WHEN OC.CUSTOMER_GENDER = 'M'...	19 row(s) returned	0.0024 se
832	20:13:51	SELECT OH.ORDER_ID, OC.CUSTOMER_ID, CASE WHEN OC.CUSTOMER_GENDER = 'M'...	19 row(s) returned	0.0023 se
833	20:16:33	SELECT OH.ORDER_ID, OC.CUSTOMER_ID, CASE WHEN OC.CUSTOMER_GENDER = 'M'...	15 row(s) returned	0.0059 se
834	20:20:48	SELECT P.PRODUCT_ID, P.PRODUCT_DESC, SUM(OI.PRODUCT_QUANTITY) AS TOT_QT...	5 row(s) returned	0.0062 se
835	20:21:50	SELECT OH.ORDER_ID, OC.CUSTOMER_ID, CASE WHEN OC.CUSTOMER_GENDER = 'M'...	15 row(s) returned	0.0074 se

15 rows returned.