

# SQL Project

## 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.

**Question – 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

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

**Solution No 1** – After the execution of query have received 52 rows as an output.

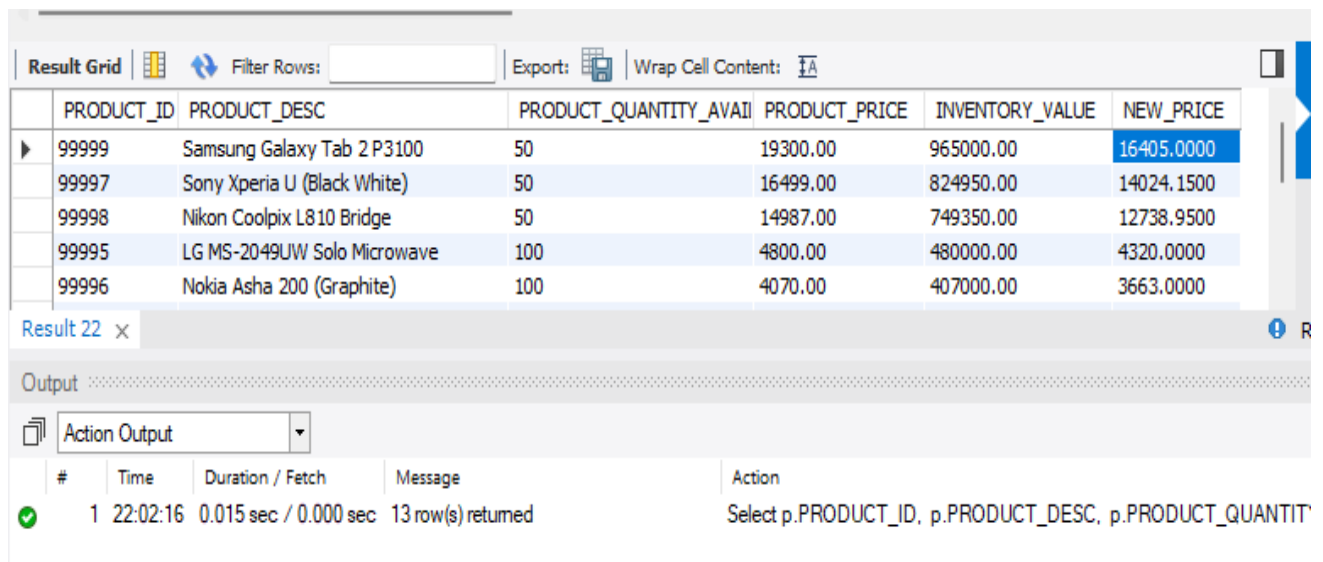
Result Grid					Filter Rows:	Export:	Wrap Cell Content:
	FullName	CUSTOMER_EMAIL	CUSTOMER_CREATION_DATE	CustomerCategory			
▶	Ms. JENNIFER WILSON	jen_w@gmail.com	1991-06-01	CATEGORY A			
	Mr. JACKSON DAVIS	dave_jack@gmail.com	2001-06-12	CATEGORY A			
	Ms. KOMAL CHOUDHARY	ch_komal@yahoo.co.IN	2002-06-26	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			
Result 21							
Output							
Action Output							
#	Time	Duration / Fetch	Message	Action			
✓ 1	21:55:37	0.922 sec / 0.000 sec	52 row(s) returned	SELECT CONCAT( (CASE WHEN CUSTOMER_			

**Question – 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 No 2** – After the execution of query have received 13 rows in final output.



The screenshot shows a database query result grid with the following columns: PRODUCT\_ID, PRODUCT\_DESC, PRODUCT\_QUANTITY\_AVAIL, PRODUCT\_PRICE, INVENTORY\_VALUE, and NEW\_PRICE. The results are sorted by INVENTORY\_VALUE in descending order. Below the grid, there is an 'Output' section showing the execution details of the query.

	PRODUCT_ID	PRODUCT_DESC	PRODUCT_QUANTITY_AVAIL	PRODUCT_PRICE	INVENTORY_VALUE	NEW_PRICE
▶	99999	Samsung Galaxy Tab 2 P3100	50	19300.00	965000.00	16405.0000
	99997	Sony Xperia U (Black White)	50	16499.00	824950.00	14024.1500
	99998	Nikon Coolpix L810 Bridge	50	14987.00	749350.00	12738.9500
	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

Result 22 x

Output



Action Output

#	Time	Duration / Fetch	Message	Action
✓ 1	22:02:16	0.015 sec / 0.000 sec	13 row(s) returned	Select p.PRODUCT_ID, p.PRODUCT_DESC, p.PRODUCT_QUANTIT

**Question – 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 No 3** – After the execution of query 9 rows returned.

Result Grid				
Filter Rows:		Export:  Wrap Cell Content: 		
PRODUCT_CLASS_CODE	PRODUCT_CLASS_DESC	Product_count	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	

Result 23 x

Output



Action Output

#	Time	Duration / Fetch	Message	Action
1	22:09:05	0.000 sec / 0.000 sec	9 row(s) returned	SELECT pc.PRODUCT_CLASS_CODE,

**Question – 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 No 4** - After the execution of query 5 rows returned.

Result Grid					
Filter Rows:		Export:  Wrap Cell Content: 			
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	

Result 24 x

Output

Action Output

#	Time	Duration / Fetch	Message	Action
1	22:22:39	0.000 sec / 0.000 sec	5 row(s) returned	SELECT OC.CUSTOMER_ID, Concat(OC.CUSTOMER_FNAME,'

**Question – 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 (9 Rows).

[note: tables to be used -shipper, online\_customer, addresss, order\_header]

**Solution No 5** - After the execution of query 9 rows returned.

The screenshot shows a database query result grid with the following data:

	SHIPPER_NAME	CITY	NUMBER_OF_CUSTOMERS	NUMBER_OF_CONSIGNMENTS
▶	DHL	Abington	1	1
	DHL	Amherst	1	1
	DHL	Bangalore	3	5
	DHL	Birmingham	1	1
	DHL	Brooklyn	1	1

Below the grid, the 'Output' section shows the 'Action Output' and a message: '9 row(s) returned'. The 'Action' column contains the SQL query: `SELECT S.SHIPPER_NAME, A.CITY, count(DISTINCT OC.CI`

**Question – 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 No 6** - After the execution of query 2 rows returned.

The screenshot shows a database query result grid with the following data:

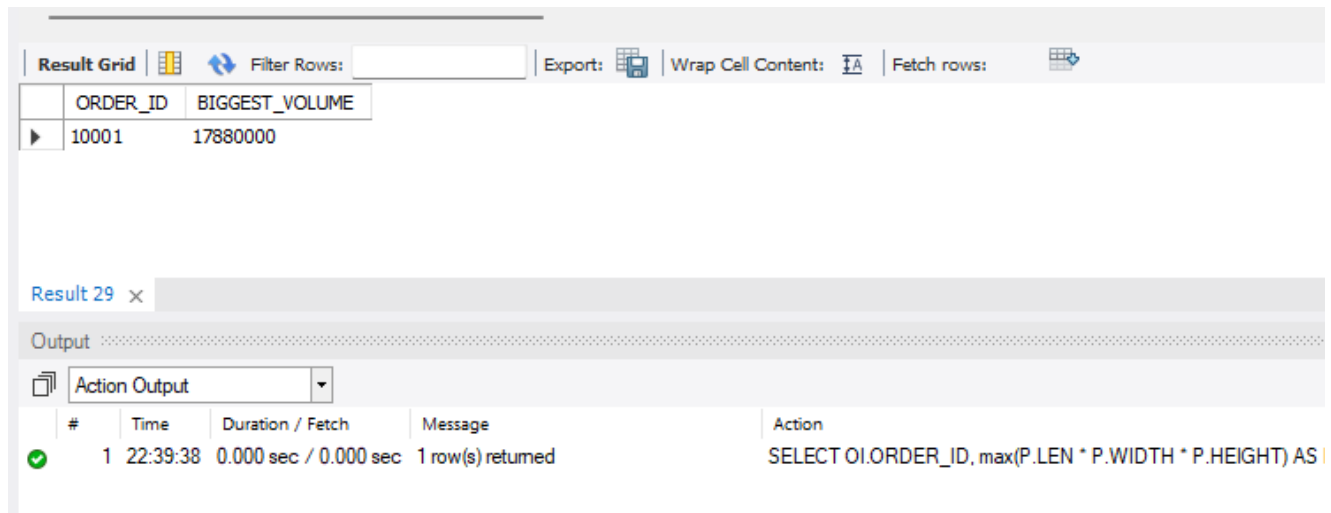
	CUSTOMER_ID	FULL_NAME	TOTAL_QUANTITY	TOTAL_VALUE
▶	24	Brian Grazer	483	1184734.00
	6	Anita Goswami	966	2369468.00

Below the grid, the 'Output' section shows the 'Action Output' and a message: '2 row(s) returned'. The 'Action' column contains the SQL query: `SELECT OC.CUSTOMER_ID, concat(OC.CUSTOMER_FNAME, '', (`

**Question – 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 No 7** – After the execution of query 1 rows returned.



The screenshot shows a database interface with a 'Result Grid' at the top. It contains a single row with the columns 'ORDER\_ID' and 'BIGGEST\_VOLUME'. The values are '10001' and '17880000' respectively. Below the grid, there is an 'Output' section with a tab labeled 'Action Output'. This section displays a table with columns: '#', 'Time', 'Duration / Fetch', 'Message', and 'Action'. The first row shows a successful execution with a green checkmark, a time of 22:39:38, a duration of 0.000 sec, a message '1 row(s) returned', and the SQL query: 'SELECT OI.ORDER\_ID, max(P.LEN \* P.WIDTH \* P.HEIGHT) AS'.

	ORDER_ID	BIGGEST_VOLUME
▶	10001	17880000

#	Time	Duration / Fetch	Message	Action
1	22:39:38	0.000 sec / 0.000 sec	1 row(s) returned	SELECT OI.ORDER_ID, max(P.LEN * P.WIDTH * P.HEIGHT) AS

**Question – 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',

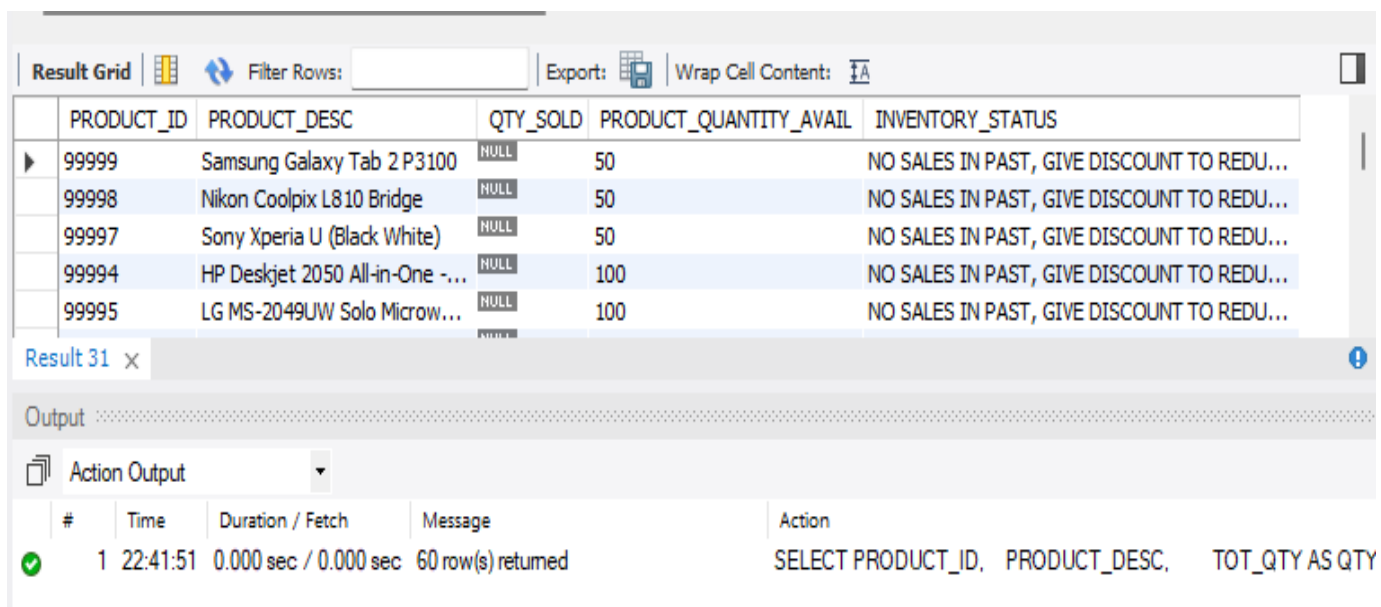
ii. If inventory quantity is less than 30% of quantity sold, show 'low inventory, need to add inventory',

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 No 8** – After the execution of query have received 60 rows as an output.



	PRODUCT_ID	PRODUCT_DESC	QTY_SOLD	PRODUCT_QUANTITY_AVAIL	INVENTORY_STATUS
▶	99999	Samsung Galaxy Tab 2 P3100	NULL	50	NO SALES IN PAST, GIVE DISCOUNT TO REDU...
	99998	Nikon Coolpix L810 Bridge	NULL	50	NO SALES IN PAST, GIVE DISCOUNT TO REDU...
	99997	Sony Xperia U (Black White)	NULL	50	NO SALES IN PAST, GIVE DISCOUNT TO REDU...
	99994	HP Deskjet 2050 All-in-One -...	NULL	100	NO SALES IN PAST, GIVE DISCOUNT TO REDU...
	99995	LG MS-2049UW Solo Microw...	NULL	100	NO SALES IN PAST, GIVE DISCOUNT TO REDU...

Result 31 x

Output

Action Output

#	Time	Duration / Fetch	Message	Action
✓ 1	22:41:51	0.000 sec / 0.000 sec	60 row(s) returned	SELECT PRODUCT_ID, PRODUCT_DESC, TOT_QTY AS QTY

**Question – 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 No 9** - After the execution of query have received 6 rows as an output.

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	ORDER_ID	PRODUCT_ID	PRODUCT_DESC	TOT_QTY
▶	10014	207	Remote Control Car	2
	10014	216	External Hard Disk 500 GB	2
	10014	202	Sams 192 L4 Single-door Refrigerator	1
	10014	212	Samsung Galaxy On6	1
	10014	214	Harry Potter	1

Result 33

×

Output

Action Output

#	Time	Duration / Fetch	Message	Action
✓ 1	22:45:48	0.000 sec / 0.000 sec	6 row(s) returned	SELECT S.ORDER_ID, S.PRODUCT_ID, S.PRODUCT_DESC, S

**Question – 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 No 10** - After the execution of query have received 15 rows as an output.

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	ORDER_ID	CUSTOMER_ID	FULLNAME	TOT_QTY
▶	10008	7	Ashwathi Bhatt	25
	10022	23	Anna Pinnock	2
	10024	32	Hans Zimmer	2
	10030	52	Suchirithaa Ekanayake	2
	10032	7	Ashwathi Bhatt	7

Result 34 x

Output

Action Output

#	Time	Duration / Fetch	Message	Action
✓ 1	22:47:14	0.000 sec / 0.000 sec	15 row(s) returned	SELECT OH.ORDER_ID, OC.CUSTOMER_ID, CONCAT (OC.CI