

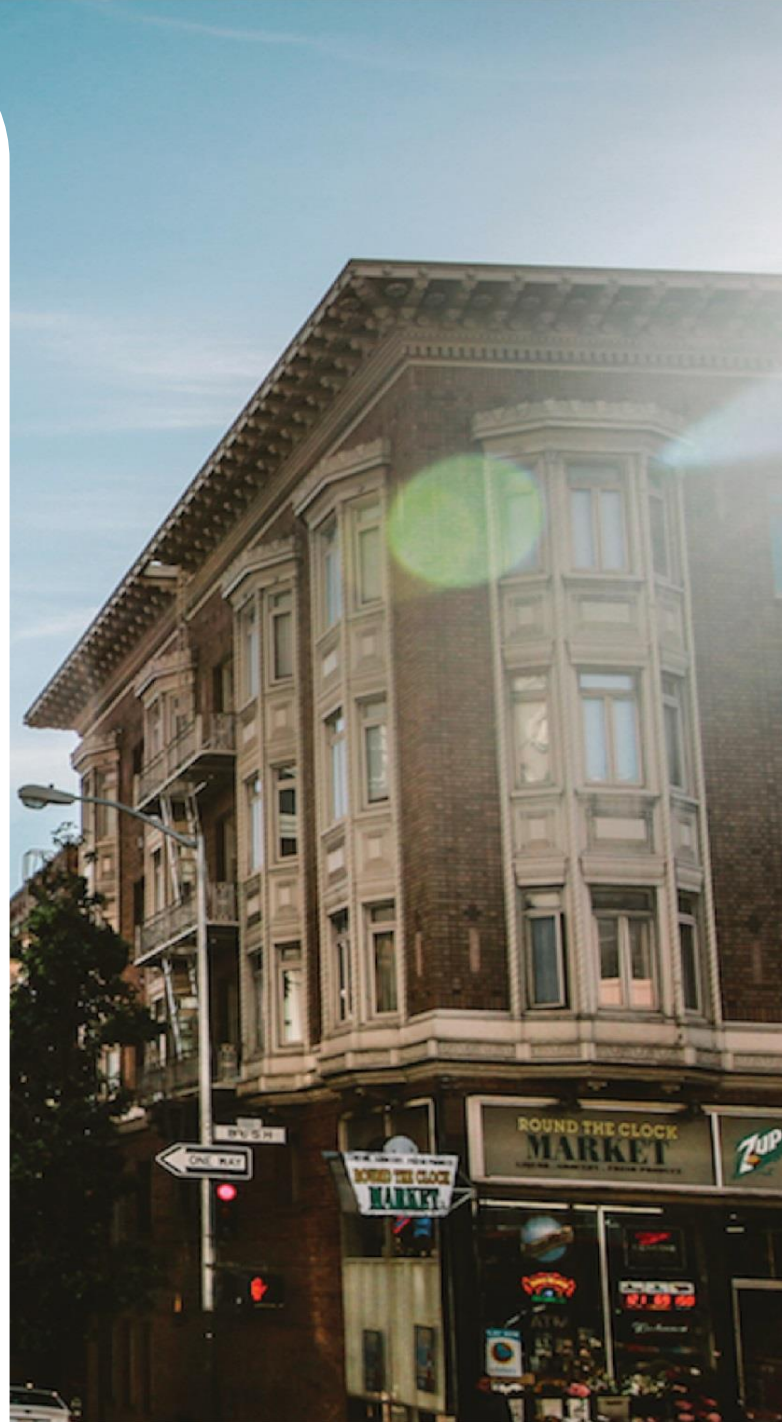
# SQL PROJECT 2024

---

---

**PGP DSBA PROGRAM**

**by: ABHISHEK K HIEMATH**



S NO	Clustering and Cleaning Ads	Page No.
1.1	<p>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:</p> <ul style="list-style-type: none"> <li>i. If customer creation date year &lt;2005 then category a</li> <li>ii. If customer creation date year &gt;=2005 and &lt;2011 then category b</li> <li>iii. If customer creation date year &gt;= 2011 then category c</li> </ul>	3 – 4
1.2	<p>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.</p> <ul style="list-style-type: none"> <li>i. If product price &gt; 20,000 then apply 20% discount</li> <li>ii. If product price &gt; 10,000 then apply 15% discount</li> <li>iii. If product price =&lt; 10,000 then apply 10% discount</li> </ul>	
1.3	<p>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.</p>	5
1.4	<p>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)</p>	6
1.5	<p>Write a query to display shipper name, city to which it is catering, number of customers catered by the shipper in the city and number of consignments delivered to that city for shipper dhl.</p>	7
1.6	<p>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'</p>	
1.7	<p>Write a query to display order_id and volume of biggest order (in terms of volume) that can fit in carton id 10</p>	8 – 9

1.8	<p>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:</p> <p>a. For electronics and computer categories,</p> <p>i. If sales till date is zero then show 'no sales in past, give discount to reduce inventory',</p> <p>ii. If inventory quantity is less than 10% of quantity sold, show 'low inventory, need to add inventory',</p> <p>iii. If inventory quantity is less than 50% of quantity sold, show 'medium inventory, need to add some inventory',</p> <p>iv. If inventory quantity is more or equal to 50% of quantity sold, show 'sufficient inventory'</p> <p>b. For mobiles and watches categories,</p> <p>i. If sales till date is zero then show 'no sales in past, give discount to reduce inventory',</p> <p>ii. If inventory quantity is less than 20% of quantity sold, show 'low inventory, need to add inventory',</p> <p>iii. If inventory quantity is less than 60% of quantity sold, show 'medium inventory, need to add some inventory',</p> <p>iv. If inventory quantity is more or equal to 60% of quantity sold, show 'sufficient inventory'</p> <p>c. Rest of the categories,</p> <p>i. If sales till date is zero then show 'no sales in past, give discount to reduce inventory',</p> <p>ii. If inventory quantity is less than 30% of quantity sold, show 'low inventory, need to add inventory',</p> <p>iii. If inventory quantity is less than 70% of quantity sold, show 'medium inventory, need to add some inventory',</p> <p>iv. If inventory quantity is more or equal to 70% of quantity sold, show 'sufficient inventory'</p>	10 – 11
1.9	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)	12
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 pin code is not starting with "5"	13 – 14

<b>S No</b>	<b>Figures and Tables</b>	<b>Page No.</b>
<b>1</b>	<b>Fig-1: Tables in orders</b>	<b>5</b>
<b>2</b>	<b>Fig-2: Customers details categorized in A, B and C</b>	<b>6</b>
<b>3</b>	<b>Fig-3: New price table</b>	<b>8</b>
<b>4</b>	<b>Fig-4: product class above 100,000</b>	<b>9</b>
<b>5</b>	<b>Fig-5: Cancelled order</b>	<b>11</b>
<b>6</b>	<b>Fig-6: Shipping Details</b>	<b>12</b>
<b>8</b>	<b>Fig-7: Customers paid in cash with last name starts with G</b>	<b>14</b>
<b>9</b>	<b>Fig-8: Total volume</b>	<b>15</b>
<b>10</b>	<b>Fig-9: Inventory Status</b>	<b>18</b>
<b>11</b>	<b>Fig-10: Products sold</b>	<b>20</b>
<b>12</b>	<b>Fig-11: total quantity shipped</b>	<b>22</b>

# 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.

	Tables_in_orders
▶	address
	carton
	online_customer
	order_header
	order_items
	product
	product_class
	shipper

FIG: -1: Tables in orders

## Problem 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

**ANSWER:** QUERY EXPLANATION

1. **Column Concatenation (full\_name):**
  - The CONCAT function combines different values into the single string.
  - The code constructs a full name by concatenating:
    - A salutation ('Mr/Ms') based on the customer\_creation\_date.
    - The uppercase first name (CUSTOMER\_FNAME).
    - The uppercase last name (CUSTOMER\_LNAME).
  - The resulting column is named full\_name.
2. **Customer ID (CUSTOMER\_ID):**
  - This column simply includes the customer ID from the online\_customer table.
3. **Customer Email (CUSTOMER\_EMAIL):**
  - Similar to the customer ID, this column will include the email address of each customer.
4. **Customer Creation Date (CUSTOMER\_CREATION\_DATE):**
  - The original creation date of the customer is included in the column.
5. **Customer Category (customer\_category):**
  - This column categorizes customers based on their creation date:
    - If the creation date is before 2005, they are in 'Category A.'
    - If the creation date is between 2005 and 2011, they are in 'Category B.'
    - Otherwise, they fall into 'Category C.'

- By assuming when a customer joined, the company can adjust its marketing and customer service strategies. For instance, longtime customers (Category A) might appreciate loyalty rewards, while newer customers (Category C) could be targeted for special introductory offers.

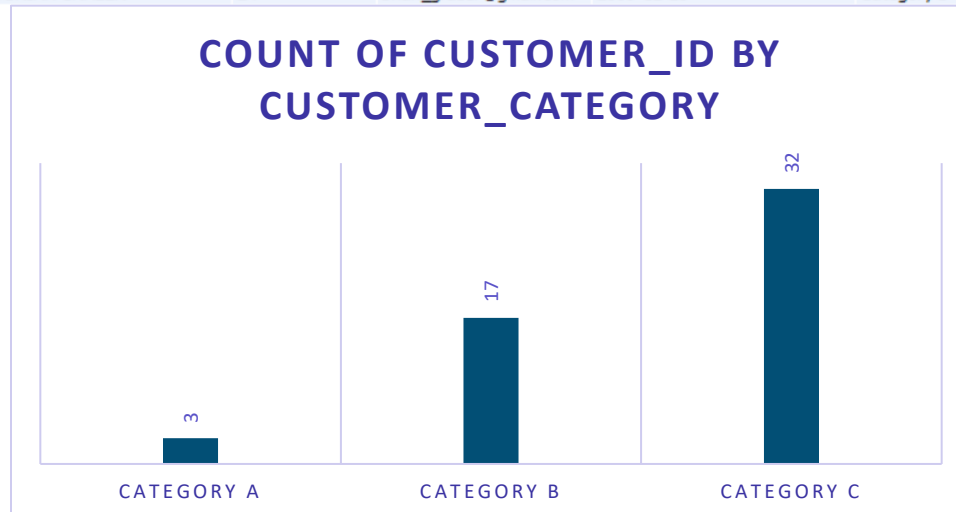
### Marketing segments:

- The segmented data can help in creating targeted marketing campaigns. Knowing which customers will fall into the which category can allow more personalized and effective promotions.
- Having email addresses alongside other details allows for quick extraction of data for personalized email campaigns aimed at different customer segments.

### Summary:

The provided query generates a list of customers with their full name, email address, creation date, and their categorized group based on the year of registration. This categorized view is Important for making strategic decisions regarding customer engagement and business growth strategies.

full_name	CUSTOMER_ID	CUSTOMER_EMAIL	CUSTOMER_CREATION_DATE	customer_category
Mr/Ms JENNIFER WILSON	1	jen_w@gmail.com	1991-06-01	Category A
Mr/Ms JACKSON DAVIS	2	dave_jack@gmail.com	2001-06-12	Category A
Mr/Ms KOMAL CHOUDHARY	3	ch_komal@yahoo.co.IN	2002-06-26	Category A
Mr/Ms WILFRED JEAN	4	w_jean@gmail.com	2006-01-12	Category B
Mr/Ms ANITA GOSWAMI	6	agoswami@gmail.com	2006-03-13	Category B
Mr/Ms ASHWATHI BHATT	7	ash_bhat@yahoo.co.IN	2007-04-15	Category B
Mr/Ms NEETHA CASTELINA	8	neetha20@gmail.com	2011-08-16	Category C
Mr/Ms DEVIKA SATISH	9	devika_sa@gmail.com	2011-09-01	Category C
Mr/Ms BIDHAN C.ROY	10	bidhanroy@yahoo.co.in	2011-10-23	Category C
Mr/Ms VIKAS JHA	11	vikas.jha@gmail.com	2011-11-15	Category C
Mr/Ms ARUL KUMAR.T	12	arulkumar@gmail.com	2011-12-03	Category C
Mr/Ms RAVI SRINIVASN	13	r_srinivasn@yahoo.co.in	2012-01-05	Category C
Mr/Ms AVINASH DUTTA	14	avinash.dutta@yahoo....	2012-01-18	Category C
Mr/Ms JYOTI SINHA	15	jyotisinha@gmail.com	2012-01-31	Category C
Mr/Ms VIJAY BOLLINENI	16	vbollineni@gmail.com	2012-02-06	Category C
Mr/Ms PRASAD SHETTY	17	pshetty@yahoo.co.in	2012-02-26	Category C
Mr/Ms SURESH BABU	18	sbabu@yahoo.co.in	2012-03-01	Category C
Mr/Ms BHARTI SUBHASH	19	bhartis@gmail.com	2012-03-28	Category C
Mr/Ms KESHAV JOG	20	kesjog@yahoo.co.in	2012-04-06	Category C
Mr/Ms RAMYA RAVINDER	5	ramya_r23@gmail.com	2006-02-12	Category B
Mr/Ms ALAN SILVESTRI	21	alan_silver@msn.com	2016-02-04	Category C
Mr/Ms ANDREW STANTON	22	andrew_stanton@yah...	2013-05-23	Category C
Mr/Ms ANNA PINNOCK	23	anna_pinnock@yahoo....	2013-01-18	Category C
Mr/Ms BRIAN GRAZER	24	brian_grazer@gmail.com	2009-12-28	Category B



**FIG: - 2: Customers details categorized in A, B and C**

## **Problem 2.**

**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**

**ANSWER:** Query Explanation

**1. Column Selection:**

- The query selects some columns from product table:
  - product\_id: The unique identifier for each product.
  - product\_desc: A description of the product.
  - product\_quantity\_avail: The available quantity of the product in the inventory.
  - product\_price: The price of the product.
  - inventory\_value: A calculated column representing total value of the product inventory (quantity multiplied by price).

**2. Conditional Calculation (new\_price):**

- The CASE expression calculates new price based on original product price:
  - If the product price is greater than the 20,000, the new price is 80% of the original price.
  - If the product price is between the 10,000 and 20,000, and the new price is 85% of the original price.
  - Otherwise (product price less than or equal to 10,000), the new price is 90% of the original price.
- The resulting column named as new\_price.

**3. Join and Filtering:**

- The query performed a left join between the product table (p) and the order\_items table (oi) using the product\_id.
- The WHERE clause filters out rows where there is no corresponding order (i.e., oi.order\_id IS NULL).
- This ensures that only products without any associated orders are included.

**4. Ordering Results:**

- The results are ordered by the inventory\_value in the descending order.

**Inventory Management:**

- Helps the business to identify products that are in stock but have not yet been sold. This can prompt further investigation into why these products aren't moving and what actions can be taken to sell those.

**Giving away Discount:**

- The calculated new\_price based on different discount provides the business with information on how different pricing strategies can affect the inventory. This helps making strategic discount campaigns to clear out stock.



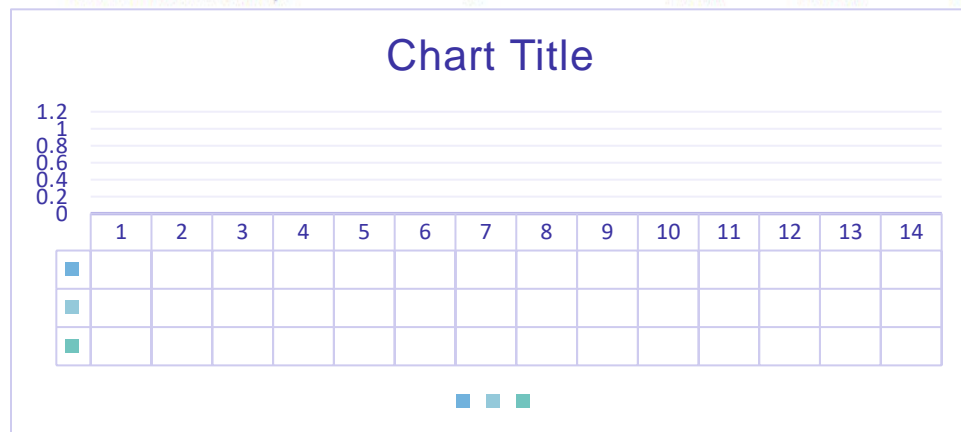
### Marketing and Promotions:

- o Knowing the inventory value helps in financial accounting and resource allocation.
- o Understanding which products have not sold can guide marketing strategies. For instance, these products could be highlighted in the promotions or sales campaigns to boost their visibility and sales.
- o By examining products that remain unsold and their respective inventory values, the business can better manage storage costs and operational efficiency.

### Summary:

The provided query lists unsold products along with their descriptions, available quantities, prices, inventory values, and discounted prices. Sorting by inventory value allows the business to prioritize actions on high-value inventory. This information is so important for driving sales, optimizing inventory levels, and improving the company's financial performance.

	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
	99994	HP Deskjet 2050 All-in-One - J510a Printer	100	3749.00	374900.00	3374.1000
	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
	99990	Quanta 4 Port USB Hub	50	500.00	25000.00	450.0000
	230	Esprit Analog Watch	5	3495.00	17475.00	3145.5000
	249	All-in-one Board Game	20	450.00	9000.00	405.0000
	250	Huwi Wi-Fi Receiver 500Mbps	30	287.00	8610.00	258.3000



**FIG: -3: New price table**

Here, zero product get 20% discount because none product\_decs price is greater than 20,000. Only 3 product\_decs get 15% discount and remains gets 10% discount (which are 10 product\_decs).



**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.**

**ANSWER:** Query Explanation

**1. Column Selection:**

- The query selects some of columns from two tables: PRODUCT (aliased as p) and PRODUCT\_CLASS (aliased as pc).
  - product\_class\_code: The code representing the product class.
  - product\_class\_desc: A description of the product class.
  - count\_product\_type: The count of product types within each class.
  - inventory\_value: The total inventory value for each product class (calculated the sum of product\_quantity\_avail \* product\_price).

**2. Joining Tables (JOIN):**

- The query starts an inner join between PRODUCT and PRODUCT\_CLASS tables based on product\_class\_code.
- This ensures that we combine information from both tables based on the common product class code.

**3. Grouping (GROUP BY):**

- The results are grouped by two the product\_class\_code and product\_class\_desc.
- This means that we'll have one row per unique product class.

**4. Filtering (HAVING):**

- The HAVING clause filters the results to include only those product classes where the inventory\_value (total value) is greater than 100,000.
- This helps focus on significant product classes.

**5. Ordering Results (ORDER BY):**

- The results are ordered by inventory\_value in descending order.
- This ensures that the most valuable product classes appear first in the report.

	product_class_code	product_class_desc	count_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

**FIG: - 4: product class above 100,000**

#### Identifying High-Value products:

- By filtering product classes with inventory values greater than 100,000, the query helps in identifying which product classes has significant inventory value. This is useful for inventory management and financial plans.
- Understanding which product classes have high inventory value can help in strategic decision-making & resource allocation, warehouse space planning, and ensuring that marketing efforts are to promote these high-value products.

#### Insights:

- High inventory value typically suggests a high potential for sales and revenue. This query can help the business focus on product classes that can significantly impact the company growth.
- By knowing the count of different product type within each high value class, the business can manage stock levels more efficiently, that important items are always available while avoiding overstocking less important items.

#### **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)**

##### **ANSWER:** Query Explanation

##### **1. Column Selection:**

- The query selects some columns from two tables: online\_customer (aliased as oc) and address (aliased as a).
  - customer\_id: The unique identifier for each customer.
  - full\_name: A concatenated column representing the full name of the customer (first name and last name).
  - customer\_email: The email address of each customer.
  - customer\_phone: The phone number of each customer.
  - country: The country associated with the customer's address.

##### **2. Joining Tables (JOIN):**

- The query does an inner join between the online\_customer and address tables based on the address\_id.
- This ensures that we combine information from both tables related to customer data.

##### **3. Filtering (WHERE):**

- The WHERE clause filters the results to include only those customers whose customer\_id appears in a subquery.
- The subquery retrieves customer\_id values from the order\_header table where the order status is 'Cancelled'.
- The HAVING clause ensures that only customers with the same count of cancelled orders as their total order count are included.

Customer Service:

- Proactively reaching out to understand the reasons behind the cancellations.
- Improving the customer experience to retain these customers.

Sales and Marketing:

- Identifying important areas for improving in product offer or delivery services.
- Tailoring future marketing strategies to regain their business.
- Streamlining processes to minimize order cancellations and their associated costs.

	customer_id	full_name	customer_email	customer_phone	country
▶	41	Tharman Shanmugaratnam	tharshan@yahoo.co.sg	8572898929	Singapore

**FIG: -5: Cancelled order**

This SQL query retrieves customer data, including full names, emails, phone numbers, and associated countries, specifically for customers with only cancelled orders. The customer has canceled the order

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

**ANSWER:** Query Explanation

**1. Column Selection:**

- The query selects several columns from different tables:
  - shipper\_name: The name of the shipping company (in this case, 'DHL').
  - city\_catering: The city associated with the catering service (from the address table).
  - num\_customers\_catered: The count of distinct customer IDs served by DHL in each city.
  - num\_consignments\_delivered: The total number of consignments delivered by DHL in each city.

**2. Joining Tables (JOIN):**

- The query does multiple joins to combine data from different tables:
  - shipper (aliased as s) with order\_header (aliased as oh) based on the shipper ID.
  - order\_header with online\_customer (aliased as oc) based on the customer ID.
  - online\_customer with address (aliased as a) based on the address ID.
- These joins allow us to link information about shipments, customers, and their addresses.

**3. Filtering (WHERE):**

- The WHERE clause restricts the results to only those rows where the shipper's name is 'DHL'.
- We're interested in DHL's performance specifically.

**4. Grouping Results (GROUP BY):**

- The results are grouped by shipper\_name and city\_catering.
- This grouping helps aggregate data for each city where DHL operates.

#### Customer Service:

- Measures the working way of DHL in delivering consignments to various cities.
- Determines the effectiveness of DHL coverage and service in different regions.
- Highlights the number of customers served by DHL in each city, which can be useful for assessing customer satisfaction and identifying areas needing improvement.

#### Strategic Decision Making:

- Provides insights into cities with higher demand for DHL services, allowing better planning and allocation of resources.
- Helps in making decisions about continuing, expanding, or changing shipping providers in certain areas based on performance.
- Can be used to identify cities where there might be challenges in delivery and find out reasons for any inefficiencies.

	shipper_name	city_catering	num_customers_catered	num_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

**FIG: -6: Shipping Details**

This query helps Reliant Retail Limited analyze and improve its shipping operations, catering to growing demands in specific areas, and enhances overall customer satisfaction through reliable and efficient delivery services by understanding the contribution and reach of DHL.

**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'.**

**ANSWER:** Query Explanation

**1. Column Selection:**

- The query selects some of columns from different tables:
  - `customer_id`: The unique identifier for each customer.
  - `full_name`: A concated column represent the full name of the customer (first name and last name).
  - `total_quantity`: The total quantity of products orderes by each customer.
  - `total_value`: The total value of product ordered by each customer (calculated as the sum of product quantity multiplied by product price).

**2. Joining Tables (JOIN):**

- The query doesa multiple joins to combine data from different tables:
  - `online_customer` (aliased as `oc`) with `order_header` (aliased as `oh`) based on the customer ID.
  - `order_header` with `order_items` (aliased as `oi`) based on the order ID.
  - `order_items` with `product` (aliased as `p`) based on the product ID.
- These joins allow us to link information about customers, orders, and products.

**3. Filtering (WHERE):**

- The WHERE clause stops the results to only those rows where the payment mode is 'Cash' and the customer's last name starts with 'G'.
- We're interested in analyze cash transactions for customer with last names beginning with 'G'.

**4. Grouping Results (GROUP BY):**

- The results are grouped by `customer_id`, `CUSTOMER_FNAME`, and `CUSTOMER_LNAME`.
- This grouping helps aggregate data for each customer.

Sales Performance:

- Understand purchasing behavior and value brought in by a specific segment of the customer base.
- Evaluate sales performance by identifying which of these customers are high volume buyers in terms of quantity and value.

Payment Mode Analysis:

- Analyze the total business done in cash payments, particularly by customers with last names starting with 'G'.
- Targeted marketing campaigns or promotions for this specific group, especially if they represent a significant portion of sales.

Customer Service:

- Increase relationships with these customers by allowing them to pay their preferred payment method and implement services to meet their needs.
- Helps in managing inventory and understanding demand patterns from a particular group of customers.

	customer_id	full_name	total_quantity	total_value
▶	6	Anita Goswami	25	93237.00
	24	Brian Grazer	4	4010.00

**FIG: -7: Customers paid in cash with last name starts with G**

This query aids Reliant Retail Limited in understanding and assessing the purchasing behavior of a specific customer segment, facilitating more informed decision-making and targeted strategies to enhance customer satisfaction and increase sales. Anita Goswami and Brain Grazer are the two customers who's having high total\_value and total quantity

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

**ANSWER:** Query Explanation

### **1. Column Selection:**

- The query selects two columns:
  - order\_id: The unique identifier for each order.
  - total\_volume: A calculated value representing the total volume of products in the order. It will be computed by multiplying the product quantity (oi.product\_quantity) with the dimensions (LEN, WIDTH, and HEIGHT) of the associated carton.

### **2. Joining Tables (JOIN):**

- The query does three joins to combine data from different tables:
  - order\_items (aliased as oi) with product (aliased as p) based on the product ID.
  - product with carton (aliased as c) based on the carton ID.
  - This join allows us to link information about orders, products, and cartons.

### **3. Filtering (WHERE):**

- The WHERE clause restricts the results to only those rows where the order\_id appears in a subquery.
- The subquery retrieves order\_id values from the order\_items table where the carton\_id is 10.
- This ensures that we're considering only orders associated with carton ID 10.

### **4. Grouping Results (GROUP BY):**

- The results are grouped by order\_id.
- This grouping helps aggregate data for each order.

### **5. Ordering Results (ORDER BY and LIMIT):**

- The results are ordered by total\_volume in descending order.
- The LIMIT 1 ensures that only the order with the highest total volume is returned.

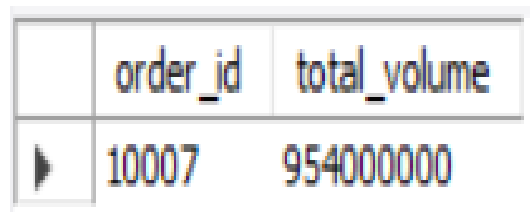


### Inventory and Logistics and Shipping:

- Helps in optimizing packaging processes by understanding the maximum volume that can fit into a specific carton size.
- Useful for determining appropriate packaging strategies for large orders.
- Assists in planning for shipping logistics by identifying orders that may require special handling or larger shipping containers.
- Helps in identifying high volume orders that could impact shipping costs and logistics planning.
- Aids in efficient resource allocation for packaging materials based on order volume.
- Ensures that the warehouse staff is better prepared to handle large volume orders.

### Customer Service:

- Allows for better tracking of large orders to know they are kept safe and delivered correctly, which can improve customer satisfaction.
- Provides insights into the types of products and volumes being ordered, which can guide future inventory and packaging decisions.



	order_id	total_volume
▶	10007	954000000

FIG: -8: Total volume

This order id has the highest total volume

**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'**

## **ANSWER:** Query Explanation

### 1. Column Selection:

- The query selects several columns from different tables:
  - PRODUCT\_ID: The unique identifier for each product.
  - PRODUCT\_DESC: A description of the product.
  - PRODUCT\_QUANTITY\_AVAIL: The available quantity of the product in inventory.
  - product\_quantity\_sold: The total quantity of the product sold (or 0 if no sales).
  - inventory\_status: A calculated column indicating the inventory status based on sales and available quantity.

### 2. Joining Tables (JOIN):

- The query joins the product table (aliased as p) with the product\_class table (aliased as pc) based on the product class code.
- This allows us to link information about products and their corresponding classes.

### 3. Left Join (LEFT JOIN):

- The query performs a left join with a subquery:
  - The subquery calculates the total quantity sold for each product using the order\_items table.
  - The left join ensures that even if there are no sales for a product, it still appears in the report with a quantity sold of 0.

### 4. Inventory Status (CASE Expressions):

- The CASE expressions determine the inventory status based on the product class and available quantity:
  - For product classes 'Electronics' and 'Computer':
    - If no sales, recommend giving a discount to reduce inventory.
    - If low inventory (less than 10% of sales), suggest adding inventory.
    - If medium inventory (less than 50% of sales), recommend adding some inventory.
    - Otherwise, consider inventory sufficient.
  - Similar rules apply for other product classes ('Mobiles', 'Watches', and others).

### Sales, Inventory Balance and Inventory Actions:

- Provides a comprehensive overview of both available quantities and sold quantities of each product.
- Recommends specific actions depending on the sales history and inventory levels:
- For products with no sales Suggests discounts to clear inventory.
- For products with low or medium inventory Indicates the importance to add inventory.
- For products with sufficient inventory Marks them as sufficiently stocked.

## Inventory Management:

- Customizes the inventory status evaluation based on product categories:
- Electronics and Computer products require higher inventory edges for actions.
- Mobiles and Watches have slightly lower beginnings.
- Enables proactive decision-making to maintain adequate stock levels.
- Helps in preventing overstock or understock situations in the warehouse.
- Supports planning for promotional activities for non-performing inventory.
- Enhances operational efficiency by providing clear actions to be taken for different products.
- Facilitates collaboration between sales, marketing, and inventory management teams to align strategies with actual data.

PRODUCT_ID	PRODUCT_DESC	PRODUCT_QUANTITY_AVAIL	product_quantity_sold	inventory_status
99999	Samsung Galaxy Tab 2 P3100	50	0	No sales in past, give discount to reduce invent...
99998	Nikon Coolpix L810 Bridge	50	0	No sales in past, give discount to reduce invent...
99997	Sony Xperia U (Black White)	50	0	No sales in past, give discount to reduce invent...
99994	HP Deskjet 2050 All-in-One - J510a Printer	100	0	No sales in past, give discount to reduce invent...
99995	LG MS-2049UW Solo Microwave	100	0	No sales in past, give discount to reduce invent...
99996	Nokia Asha 200 (Graphite)	100	0	No sales in past, give discount to reduce invent...
99991	Dell Targus Synergy 2.0 Backpack	250	0	No sales in past, give discount to reduce invent...
99992	Tom Clancy's Ghost Recon: Future Soldier (PC G...	250	0	No sales in past, give discount to reduce invent...
99993	Nokia 1280 (Black)	250	0	No sales in past, give discount to reduce invent...
201	Sky LED 102 CM TV	30	6	Sufficient inventory
202	Sams 192 L4 Single-door Refrigerator	15	6	Sufficient inventory
203	Jockey 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
228	Adidas Analog Watch	10	7	Sufficient inventory
229	Disney Analog Watch	10	2	Sufficient inventory
230	Esprit Analog Watch	5	0	No sales in past, give discount to reduce invent...
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
236	Solo Exam SB-01 Writing Pad	30	21	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
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
235	Cindy HMPOC Pencil Box (Multicolor)	10	40	Low inventory, need to add inventory
237	Zamark Color Pencil Art Set	10	3	Sufficient inventory
241	PK Copier A4 75 GSM White Paper Ream	2	18	Low inventory, need to add 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	0	No sales in past, give discount to reduce invent...
250	Huwi Wi-Fi Receiver 500Mbps	30	0	No sales in past, give discount to reduce invent...
99990	Quanta 4 Port USB Hub	50	0	No sales in past, give discount to reduce invent...

FIG: -9: Inventory Status

**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)**

**Answer:** Query Explanation

**1. Column Selection:**

- The query selects three columns:
  - PRODUCT\_ID: The unique identifier for each product.
  - PRODUCT\_DESC: A description of the product.
  - tot\_qty: The total quantity of the product ordered across different orders.

**2. Joining Tables (JOIN):**

- The query performs a join between the order\_items table (aliased as oi) and the product table (aliased as p) based on the product ID.
- This allows us to link information about products and their corresponding order items.

**3. Filtering (WHERE):**

- The WHERE clause restricts the results to only those rows where the order ID appears in a subquery.
- The subquery retrieves order IDs from the order\_header table based on specific conditions related to customer addresses (excluding Bangalore and New Delhi).

**4. Additional Filtering (AND):**

- The query further filters out rows where the product ID is not equal to 201.
- This ensures that product 201 is excluded from the report.

**5. Grouping Results (GROUP BY):**

- The results are grouped by PRODUCT\_ID and PRODUCT\_DESC.
- This grouping helps aggregate data for each product.

**6. Ordering Results (ORDER BY):**

- The results are ordered by tot\_qty (total quantity) in descending order.
- This arranges the products based on their overall quantity ordered.

Targeted Marketing and Promotions:

- Identifies which products are frequently purchased together with the specific product (product ID 201), but not in 'Bangalore' or 'New Delhi'.
- Allows the marketing team to create targeted promotions for products often bought together with item 201.
- Helps in modifying marketing campaigns for customers not located in 'Bangalore' or 'New Delhi'.

Inventory Management:

- Provides information on the demand for products associated with item 201, helping in better inventory planning and stock management.
- Confirms popular items are effectively stocked in regions other than 'Bangalore' and 'New Delhi'.

- Supports strategic decisions for cross-selling and upselling by leveraging products that are commonly sold together.
- Helps in identifying likely bundles or packages to increase average order value.
- Provides insights allowing for efficient resource allocation in logistics and warehousing.
- Helps in optimizing supply chain operations to focus on regions with higher sales of associated products.

#### Customer Insights:

- Increases customer experience by understanding and get ahead their needs based on purchasing patterns.
- Allows for better segmentation and personalization of offers based on customer location and buying habits.

PRODUCT_ID	PRODUCT_DESC	tot_qty
218	Shell Fingertip Ball Pen	45
235	Cindy HMPOC Pencil Box (Multicolor)	25
240	4M Post It Pad 3.5	19
236	Solo Exam SB-01 Writing Pad	18
241	PK Copier A4 75 GSM White Paper Ream	16
244	Foldable Premium Chair	15
219	Ruf-n-Tuf Black PU Leather Belt	11
206	Barbie Fab Gown Doll	9
233	HP ODC School Bag 2.5'	8
207	Remote Control Car	7
242	GreenWud CT-NO-PR Coffee Table	7
216	External Hard Disk 500 GB	6
209	Blue Jeans 34	6
238	Kasyo DJ-2100 Desktop Calculator	5
243	Supreme Fusion Cupboard 02TB	5
228	Adidas Analog Watch	5
204	Cricket Set for Boys	5
231	HP ODC Laptop Bag 15.5	5
202	Sams 192 L4 Single-door Refrigerator	5
239	TRANS 2D A4 Size Box File	5
212	Samsung Galaxy On6	4
232	Women Hand Bag	4
227	Phils Wah Collection Juicer JM12	4

**FIG: -10: Products sold**

- Top-Selling Products: Identify products with the highest total quantities sold. For example, the Shell Fingertip Ball Pen and Cindy HMPOC Pencil Box have high quantities.
- Product Categories: Group products into categories like office supplies, electronics, clothing, and miscellaneous items to analyze sales trends within each category.
- Monitor stock levels for high-demand products to ensure timely restocking and avoid stockouts.
- Evaluate the pricing of products to ensure competitiveness and profitability.



**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 pin code is not starting with "5".**

**ANSWER:** Query Explanation

**1. Column Selection:**

- The query selects several columns:
  - ORDER\_ID: The unique identifier for each order.
  - CUSTOMER\_ID: The unique identifier for each customer.
  - full\_name: A concatenated column representing the full name of the customer (first name and last name).
  - total\_quantity\_shipped: The total quantity of products shipped in each order.

**2. Joining Tables (JOIN):**

- The query performs multiple joins to combine data from different tables:
  - order\_header (aliased as oh) with online\_customer (aliased as oc) based on the customer ID.
  - online\_customer with address (aliased as a) based on the address ID.
  - order\_items (aliased as oi) with order\_header based on the order ID.
- These joins allow us to link information about orders, customers, and their addresses.

**3. Filtering (WHERE):**

- The WHERE clause restricts the results to only those rows where:
  - The order ID is even (i.e.,  $oh.ORDER\_ID \% 2 = 0$ ).
  - The PIN code (from the address) does not start with '5' (i.e.,  $NOT (a.PINCODE LIKE '5\%')$ ).
- This filters out specific orders and addresses based on the given conditions.

**4. Grouping Results (GROUP BY):**

- The results are grouped by ORDER\_ID, CUSTOMER\_ID, and full\_name.
- This grouping helps aggregate data for each order and customer.

**Customer Analysis:**

- Identifies even-numbered orders, which can be useful for batch processing or targeting specific order sets.
- Helps in separating orders based on a specific sequence (even order IDs).
- Provides detailed information about customers who placed these even-numbered orders.
- Excludes orders to addresses where the pincode starts with '5', which may signify a particular region.
- Helps in focusing analysis on orders outside of this specific geographical area.
- Useful for regional performance analysis, logistics planning, and targeted marketing campaigns.

### Strategic Decisions:

- Calculates total quantities shipped per order, aiding in logistics and inventory management.
- Insights gathered can help in determining the load per batch, planning shipping resources, and managing warehouse operations.
- Facilitates efficient resource allocation and improves operational workflows.
- Data can support strategic decisions related to customer service, warehousing, and distribution.
- Helps in identifying patterns or trends in orders that can be leveraged for business improvement.

### Targeted Customer Service:

- Enables customer service teams to quickly access necessary information for any inquiries related to these specific orders.
- Enhances customer satisfaction by providing timely and accurate responses.

ORDER_ID	CUSTOMER_ID	full_name	total_quantity_shipped
10008	7	Ashwathi Bhatt	25
10022	23	Anna Pinnock	2
10024	32	Hans Zimmer	2
10028	23	Anna Pinnock	2
10030	52	Suchirithaa Ekanayake	2
10032	7	Ashwathi Bhatt	7
10034	19	Bharti Subhash	2
10036	24	Brian Grazer	4
10040	3	Komal Choudhary	2
10042	26	Stephen E. Rivkin	2
10044	39	Liz Mullane	3
10046	3	Komal Choudhary	1
10048	33	Niseema Zimmer	1
10052	17	Prasad Shetty	1
10054	19	Bharti Subhash	3
10058	28	Sayyed Faraj	2
10064	35	Thomas Newman	3
10068	51	Ahmad Bin Gh Azali	3
10070	10	Bidhan C.Roy	3

**Fig: -11: total quantity shipped**

- Identify customers with the highest total quantity shipped. For example, Ashwathi Bhatt and Bharti Subhash have multiple orders with significant quantities.
- Analyze how frequently each customer places orders. Customers like Anna Pinnock and Komal Choudhary have multiple entries, indicating regular transactions.
- Evaluate the total quantity shipped per order. Orders range from 1 to 7 units, which can help in understanding demand patterns.