SALES ANALYSIS USING SQL

PRESENTED BY :
SHRUTI PATEL
(cohort – 4)

INTRODUCTION

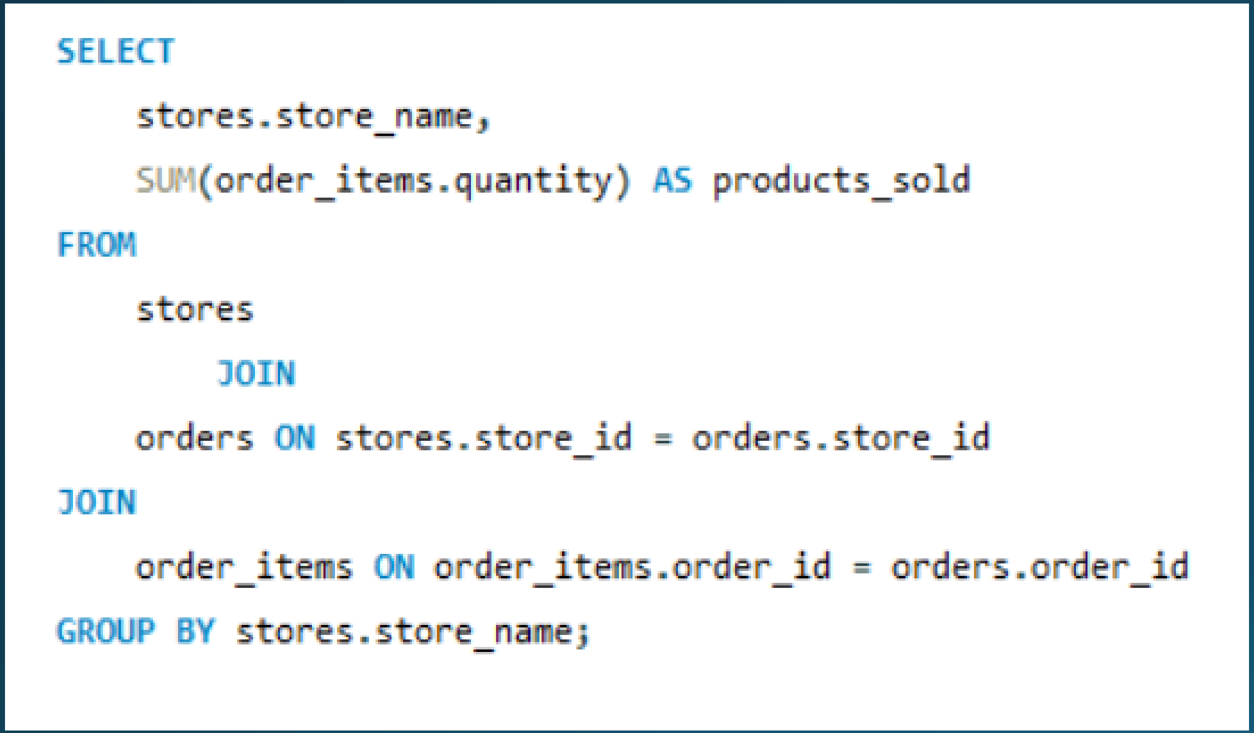
About Jenson USA:

Jenson USA is a well-known American retailer that specializes in bicycles, bike parts, and cycling gear. It offers a wide range of products online and serves cycling enthusiasts across the United States .

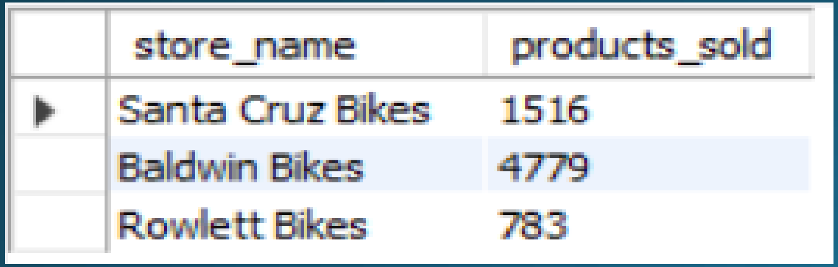
Why Analyse Jenson USA's Data ?

Analysing Jenson USA's sales data helps in :
Understanding product performance over time
Identifying top-selling products in each category
Monitoring trends in customer purchases
Making informed inventory and marketing decisions Improving overall business strategy and customer satisfaction.

Find the total number of product sold by each store along with the store name.



```
SELECT
    stores.store_name,
    SUM(order_items.quantity) AS products_sold
FROM
    stores
        JOIN
    orders ON stores.store_id = orders.store_id
JOIN
    order_items ON order_items.order_id = orders.order_id
GROUP BY stores.store_name;
```

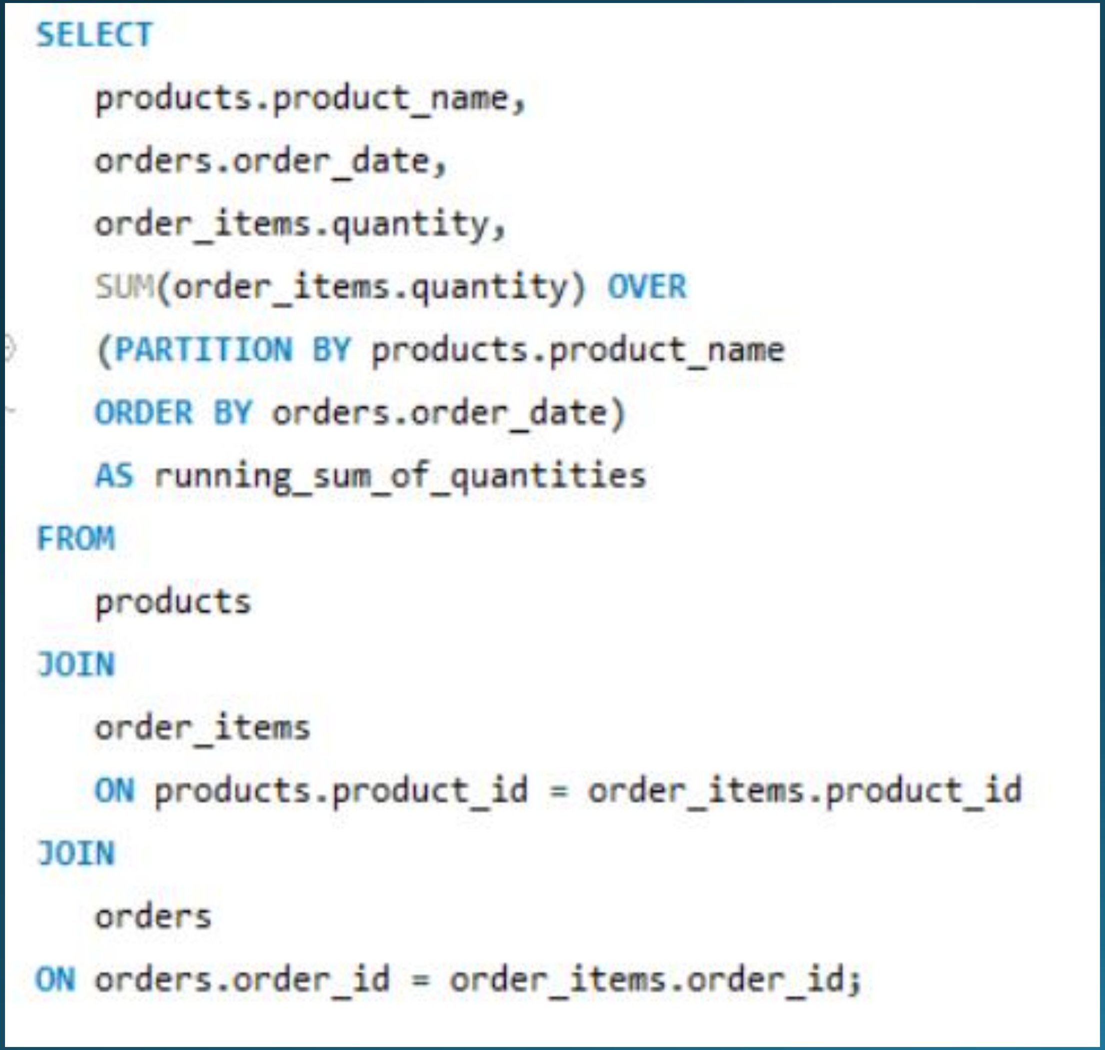


store_name	products_sold
Santa Cruz Bikes	1516
Baldwin Bikes	4779
Rowlett Bikes	783

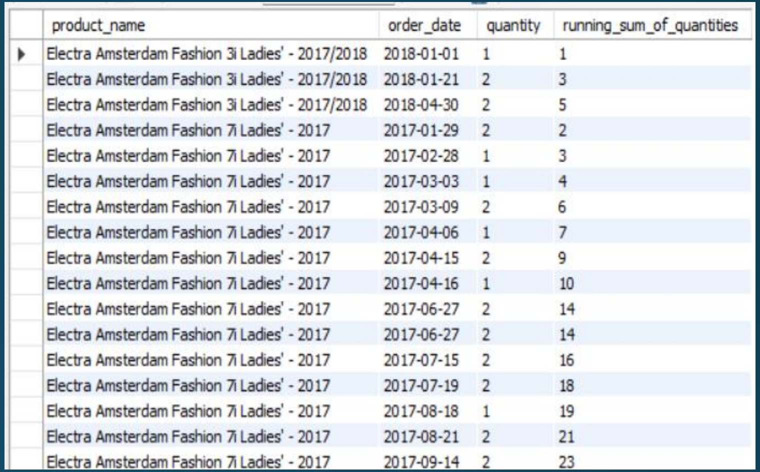
Key Takeaways:

1. Baldwin Bikes is the top-performing store with 4779 products sold.
2. Santa Cruz Bikes and Rowlett Bikes sold much fewer products in comparison.
3. There may be opportunities to:
   Study what makes Baldwin Bikes more successful (pricing, location, promotions).
   Improve sales strategies in the other two stores.

CALCULATE THE CUMULATIVE SUM OF QUANTITIES SOLD FOR EACH PRODUCT OVER TIME.



```sql
SELECT
    products.product_name,
    orders.order_date,
    order_items.quantity,
    SUM(order_items.quantity) OVER
    (PARTITION BY products.product_name
    ORDER BY orders.order_date)
    AS running_sum_of_quantities
FROM
    products
JOIN
    order_items
    ON products.product_id = order_items.product_id
JOIN
    orders
ON orders.order_id = order_items.order_id;
```

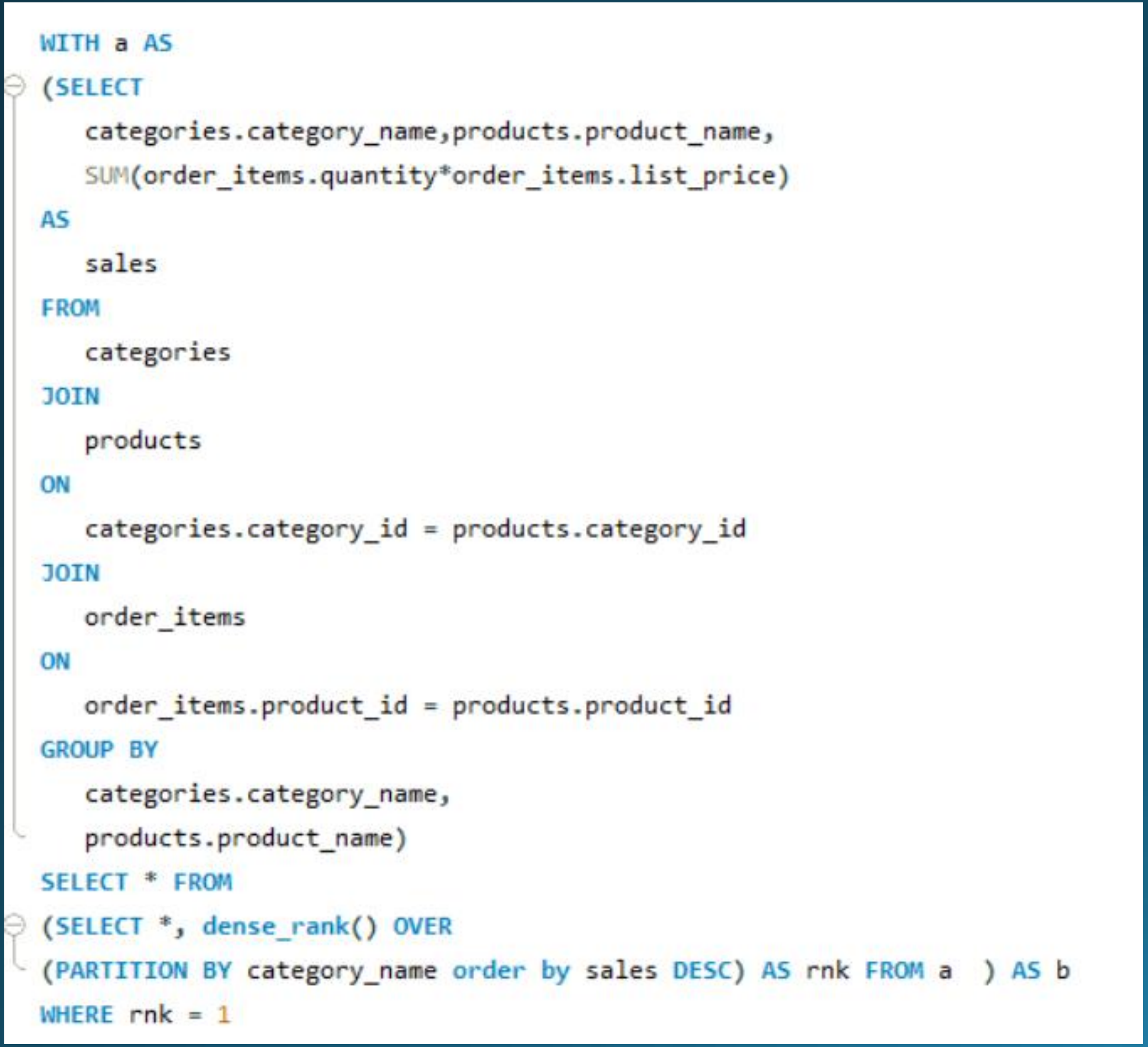


product_name	order_date	quantity	running_sum_of_quantities
Electra Amsterdam Fashion 3i Ladies' - 2017/2018	2018-01-01	1	1
Electra Amsterdam Fashion 3i Ladies' - 2017/2018	2018-01-21	2	3
Electra Amsterdam Fashion 3i Ladies' - 2017/2018	2018-04-30	2	5
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-01-29	2	2
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-02-28	1	3
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-03-03	1	4
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-03-09	2	6
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-04-06	1	7
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-04-15	2	9
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-04-16	1	10
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-06-27	2	14
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-06-27	2	14
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-07-15	2	16
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-07-19	2	18
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-08-18	1	19
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-08-21	2	21
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-09-14	2	23

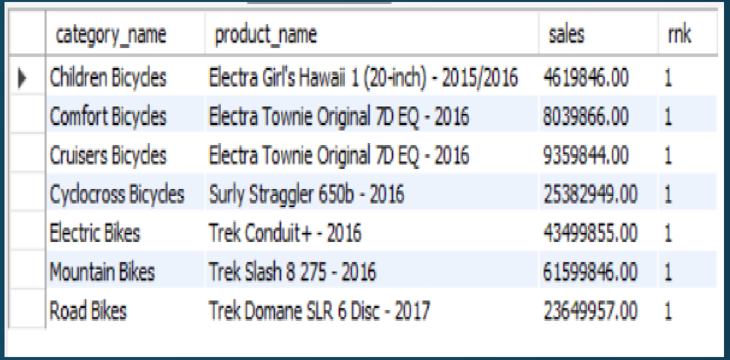
Key Takeaways:

1.The query calculates the cumulative quantity sold for each product over time.
2.It uses the SUM() OVER window function to show how sales grow.
3.Helps in tracking sales trends, forecasting demand, and managing inventory.
4.Useful for identifying best-selling periods of each product.

FIND THE PRODUCT WITH THE HIGHEST TOTAL SALES (QUANTITY * PRICE) FOR EACH CATEGORY.



```sql
WITH a AS
(SELECT
    categories.category_name,products.product_name,
    SUM(order_items.quantity*order_items.list_price)
AS
    sales
FROM
    categories
JOIN
    products
ON
    categories.category_id = products.category_id
JOIN
    order_items
ON
    order_items.product_id = products.product_id
GROUP BY
    categories.category_name,
    products.product_name)
SELECT * FROM
(SELECT *, dense_rank() OVER
(PARTITION BY category_name order by sales DESC) AS rnk FROM a  ) AS b
WHERE rnk = 1
```

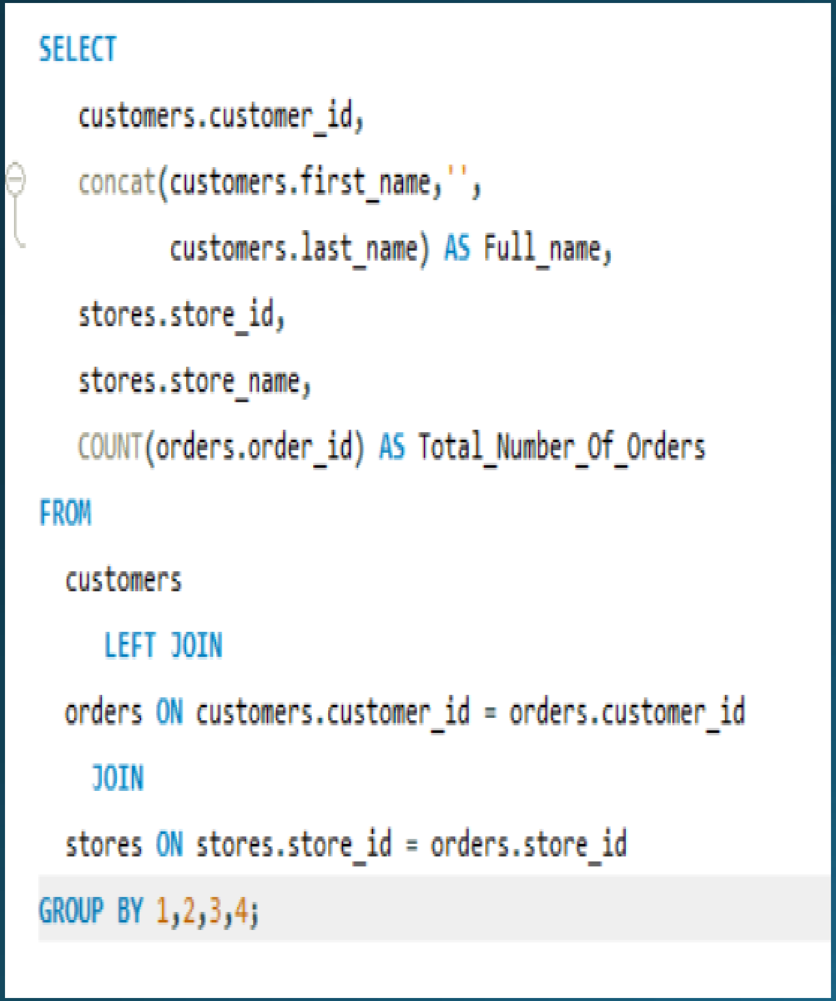


category_name	product_name	sales	rnk
Children Bicycles	Electra Girl's Hawaii 1 (20-inch) - 2015/2016	4619846.00	1
Comfort Bicycles	Electra Townie Original 7D EQ - 2016	8039866.00	1
Cruisers Bicycles	Electra Townie Original 7D EQ - 2016	9359844.00	1
Cyclocross Bicycles	Surly Straggler 650b - 2016	25382949.00	1
Electric Bikes	Trek Conduit+ - 2016	43499855.00	1
Mountain Bikes	Trek Slash 8 275 - 2016	61599846.00	1
Road Bikes	Trek Domane SLR 6 Disc - 2017	23649957.00	1

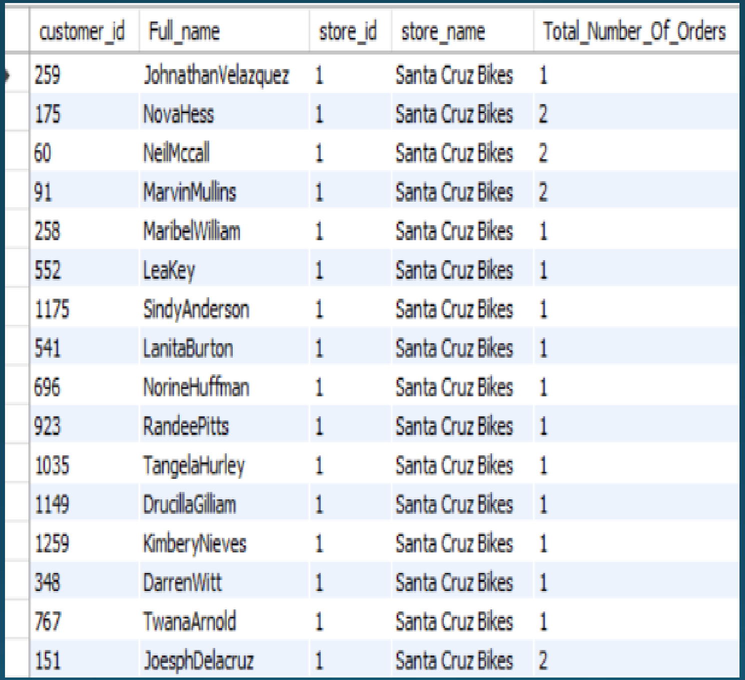
Key Insights:

1. This query finds the top-selling product (by revenue) in each category.
2. Revenue is calculated as: quantity × price.
3. It uses DENSE_RANK() to rank products by sales within each category.
4. Helps in identifying best performers in each product category for better strategy and focus.

Find the total number of orders placed by each customer per store.



```
SELECT
   customers.customer_id,
   concat(customers.first_name,'',
          customers.last_name) AS Full_name,
   stores.store_id,
   stores.store_name,
   COUNT(orders.order_id) AS Total_Number_Of_Orders
FROM
  customers
    LEFT JOIN
  orders ON customers.customer_id = orders.customer_id
    JOIN
  stores ON stores.store_id = orders.store_id
GROUP BY 1,2,3,4;
```

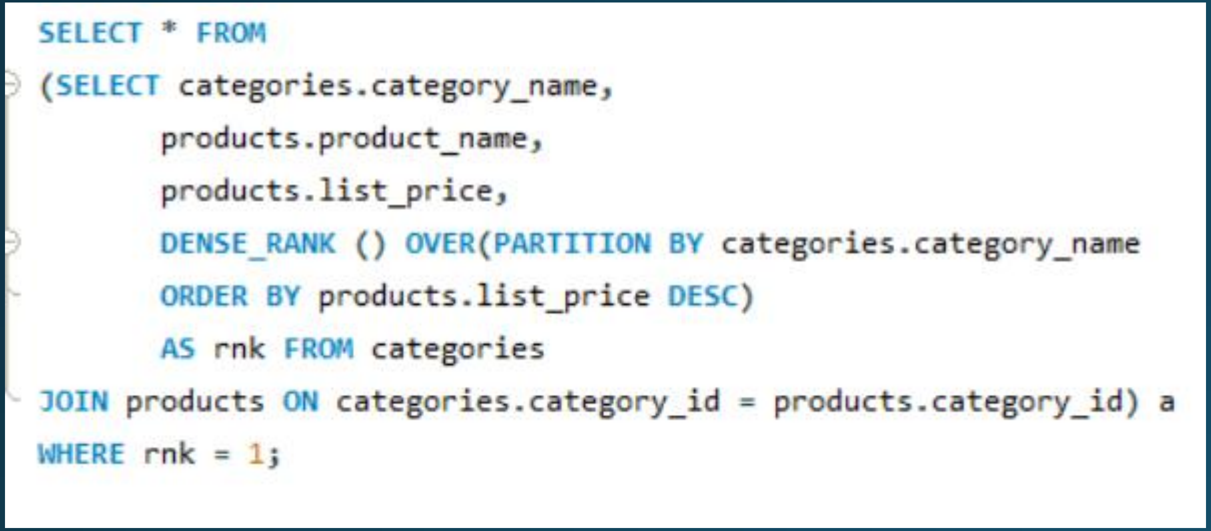


customer_id	Full_name	store_id	store_name	Total_Number_Of_Orders
259	JohnathanVelazquez	1	Santa Cruz Bikes	1
175	NovaHess	1	Santa Cruz Bikes	2
60	NeilMccall	1	Santa Cruz Bikes	2
91	MarvinMullins	1	Santa Cruz Bikes	2
258	MaribelWilliam	1	Santa Cruz Bikes	1
552	LeaKey	1	Santa Cruz Bikes	1
1175	SindyAnderson	1	Santa Cruz Bikes	1
541	LanitaBurton	1	Santa Cruz Bikes	1
696	NorineHuffman	1	Santa Cruz Bikes	1
923	RandeePitts	1	Santa Cruz Bikes	1
1035	TangelaHurley	1	Santa Cruz Bikes	1
1149	DrucillaGilliam	1	Santa Cruz Bikes	1
1259	KimberyNieves	1	Santa Cruz Bikes	1
348	DarrenWitt	1	Santa Cruz Bikes	1
767	TwanaArnold	1	Santa Cruz Bikes	1
151	JoesphDelacruz	1	Santa Cruz Bikes	2

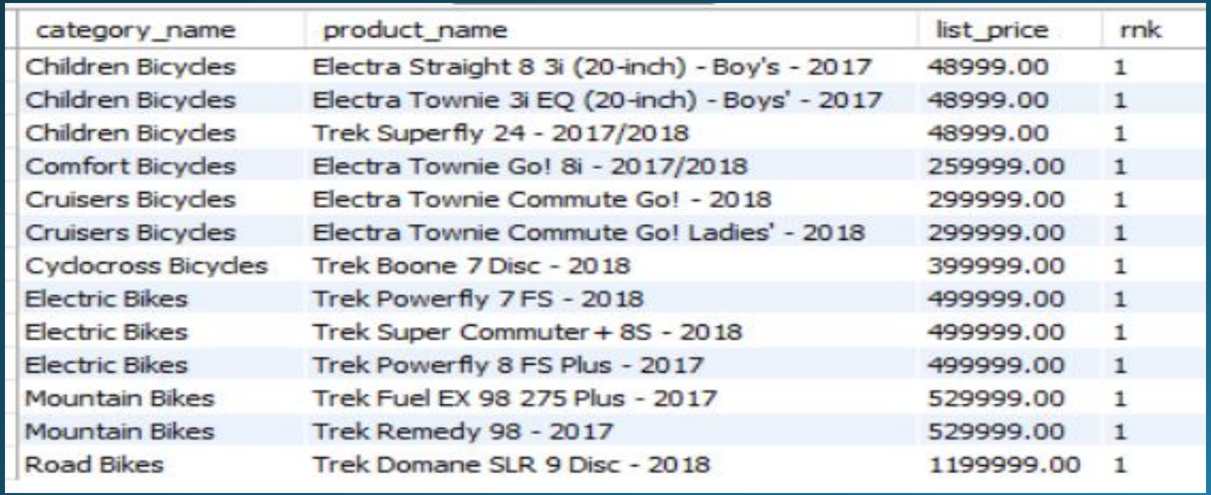
Key Insights:

1. This query finds the total number of orders placed by each customer in each store.
2. It uses COUNT(order_id) to calculate total orders.
3. A LEFT JOIN ensures even customers with zero orders are included.
4. Grouping by customer and store helps analyze behavior per location.
5. Supports identifying active vs. inactive customers and their store preferences.

FIND THE HIGHEST-PRICED PRODUCT FOR EACH CATEGORY NAME.



```
SELECT * FROM
(SELECT categories.category_name,
        products.product_name,
        products.list_price,
        DENSE_RANK () OVER(PARTITION BY categories.category_name
        ORDER BY products.list_price DESC)
        AS rnk FROM categories
JOIN products ON categories.category_id = products.category_id) a
WHERE rnk = 1;
```

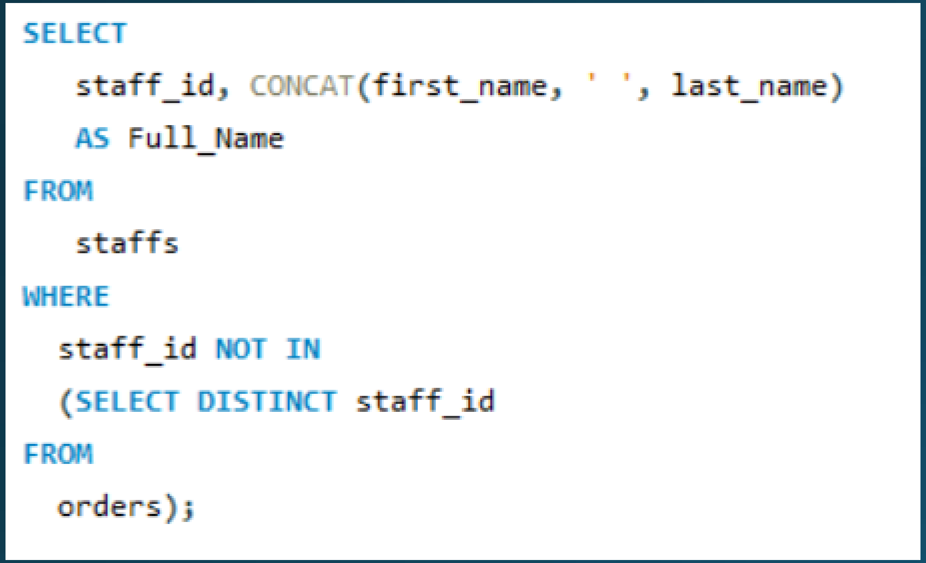


category_name	product_name	list_price	rnk
Children Bicycles	Electra Straight 8 3i (20-inch) - Boy's - 2017	48999.00	1
Children Bicycles	Electra Townie 3i EQ (20-inch) - Boys' - 2017	48999.00	1
Children Bicycles	Trek Superfly 24 - 2017/2018	48999.00	1
Comfort Bicycles	Electra Townie Go! 8i - 2017/2018	259999.00	1
Cruisers Bicycles	Electra Townie Commute Go! - 2018	299999.00	1
Cruisers Bicycles	Electra Townie Commute Go! Ladies' - 2018	299999.00	1
Cyclocross Bicycles	Trek Boone 7 Disc - 2018	399999.00	1
Electric Bikes	Trek Powerfly 7 FS - 2018	499999.00	1
Electric Bikes	Trek Super Commuter + 8S - 2018	499999.00	1
Electric Bikes	Trek Powerfly 8 FS Plus - 2017	499999.00	1
Mountain Bikes	Trek Fuel EX 98 275 Plus - 2017	529999.00	1
Mountain Bikes	Trek Remedy 98 - 2017	529999.00	1
Road Bikes	Trek Domane SLR 9 Disc - 2018	1199999.00	1

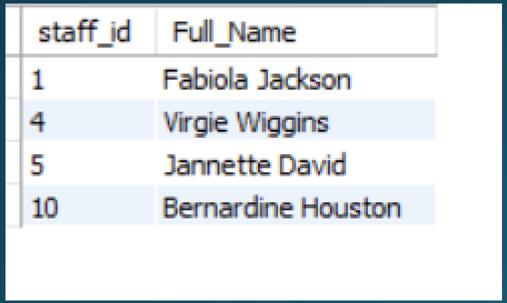
key insights:

1.Retrieve the highest-priced product for each product category.
2.Utilizes the DENSE_RANK() window function.
3.Data is partitioned by category_name to rank products within each category.
4.Products are ordered by list_price in descending order to rank highest first.
5.Only rows with rank = 1 are selected, giving the top-priced products per category.
6.categories and products tables are joined using(category_id.

FIND THE NAMES OF STAFF MEMBERS WHO HAVE NOT MADE ANY SALES.



```
SELECT
   staff_id, CONCAT(first_name, ' ', last_name)
   AS Full_Name
FROM
   staffs
WHERE
  staff_id NOT IN
  (SELECT DISTINCT staff_id
FROM
  orders);
```

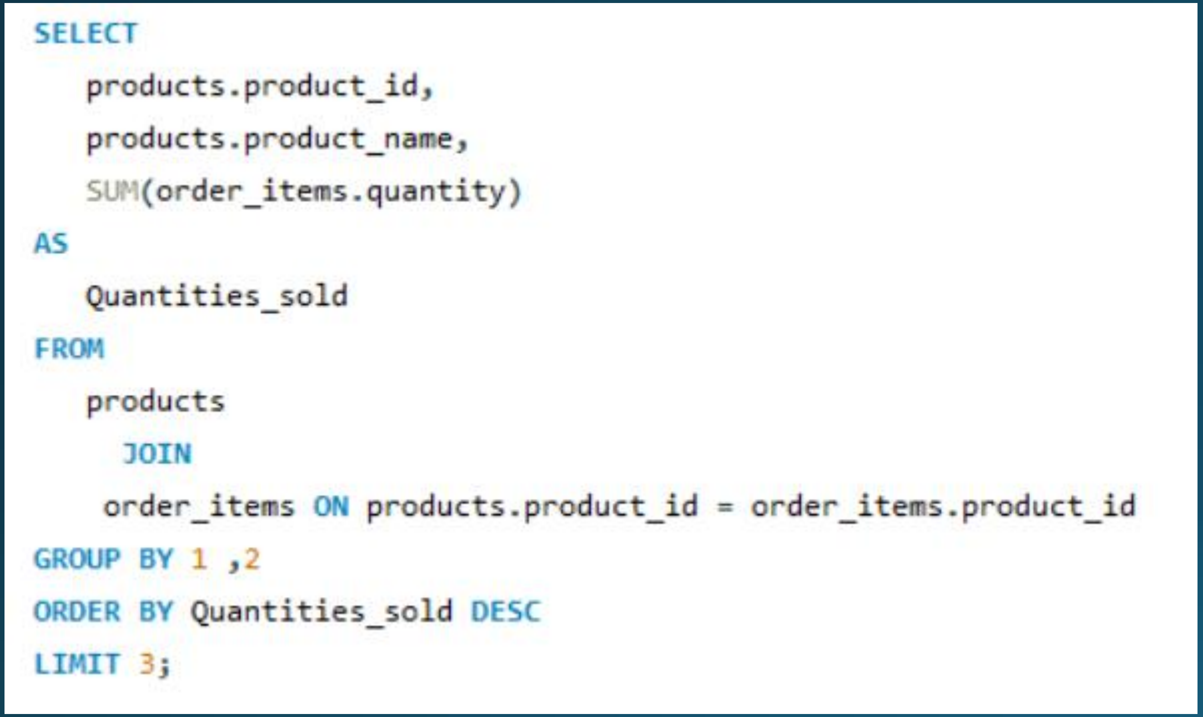


staff_id	Full_Name
1	Fabiola Jackson
4	Virgie Wiggins
5	Jannette David
10	Bernardine Houston

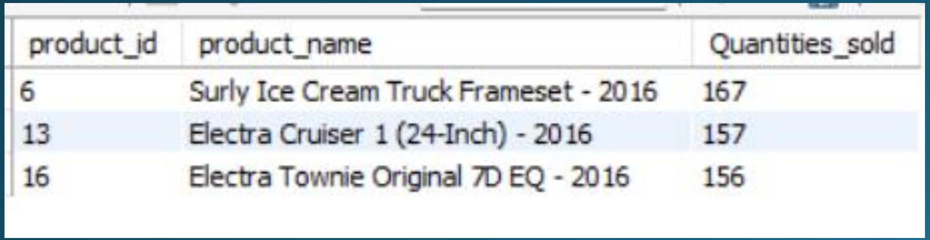
key insights:

1. Identify staff members who have made no sales.
2. Uses NOT IN with a subquery to exclude staff who appear in the orders table.
3. Selects distinct staff_ids from orders (those who made sales).
4. Selects from staffs table only those staff_ids not in the subquery result.
5. Returns a list of staff with no sales, showing their full names.

Find the top 3 most sold products in terms of quantity.



```
SELECT
   products.product_id,
   products.product_name,
   SUM(order_items.quantity)
AS
   Quantities_sold
FROM
   products
     JOIN
    order_items ON products.product_id = order_items.product_id
GROUP BY 1 ,2
ORDER BY Quantities_sold DESC
LIMIT 3;
```

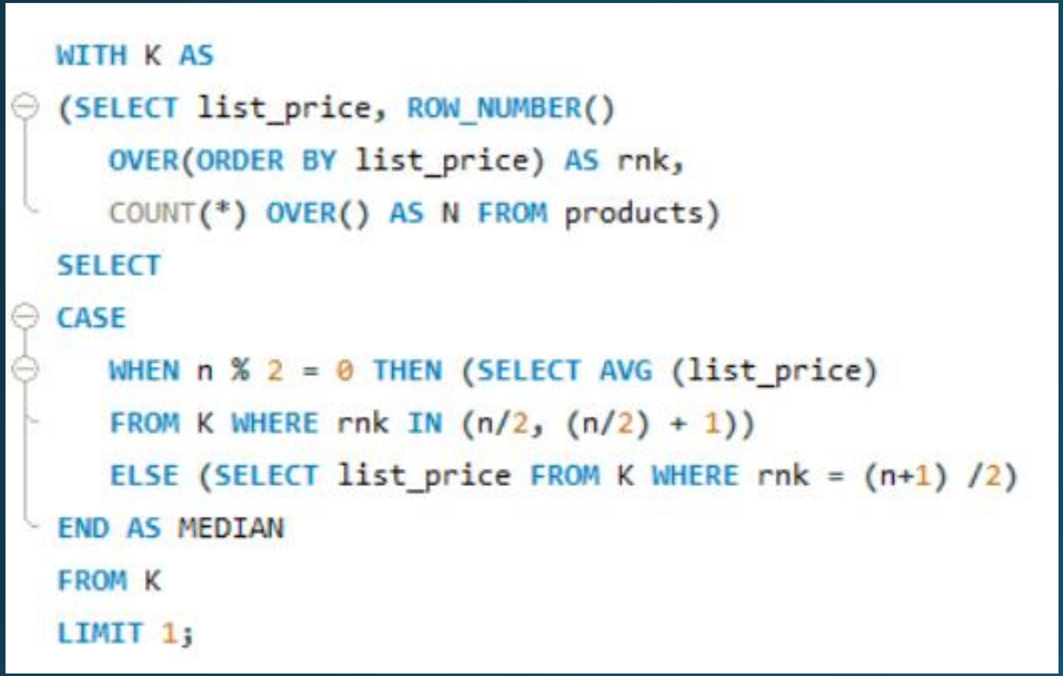


product_id	product_name	Quantities_sold
6	Surly Ice Cream Truck Frameset - 2016	167
13	Electra Cruiser 1 (24-Inch) - 2016	157
16	Electra Townie Original 7D EQ - 2016	156

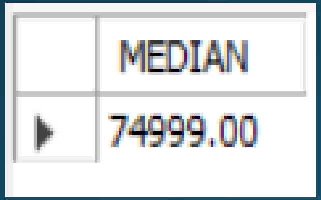
Key insights:

1. Retrieves the top 3 best-selling products based on total quantity sold.
2. Uses SUM(order_items.quantity) to calculate total quantity per product.
3. Joins products with order_items using product_id.
4. Groups by product_id and product_name.
5. Orders results in descending order of quantity sold.
6. Limits the output to 3 rows using LIMIT 3.

Find the median value of the price list.



```
WITH K AS
(SELECT list_price, ROW_NUMBER()
   OVER(ORDER BY list_price) AS rnk,
   COUNT(*) OVER() AS N FROM products)
SELECT
CASE
   WHEN n % 2 = 0 THEN (SELECT AVG (list_price)
   FROM K WHERE rnk IN (n/2, (n/2) + 1))
   ELSE (SELECT list_price FROM K WHERE rnk = (n+1) /2)
END AS MEDIAN
FROM K
LIMIT 1;
```

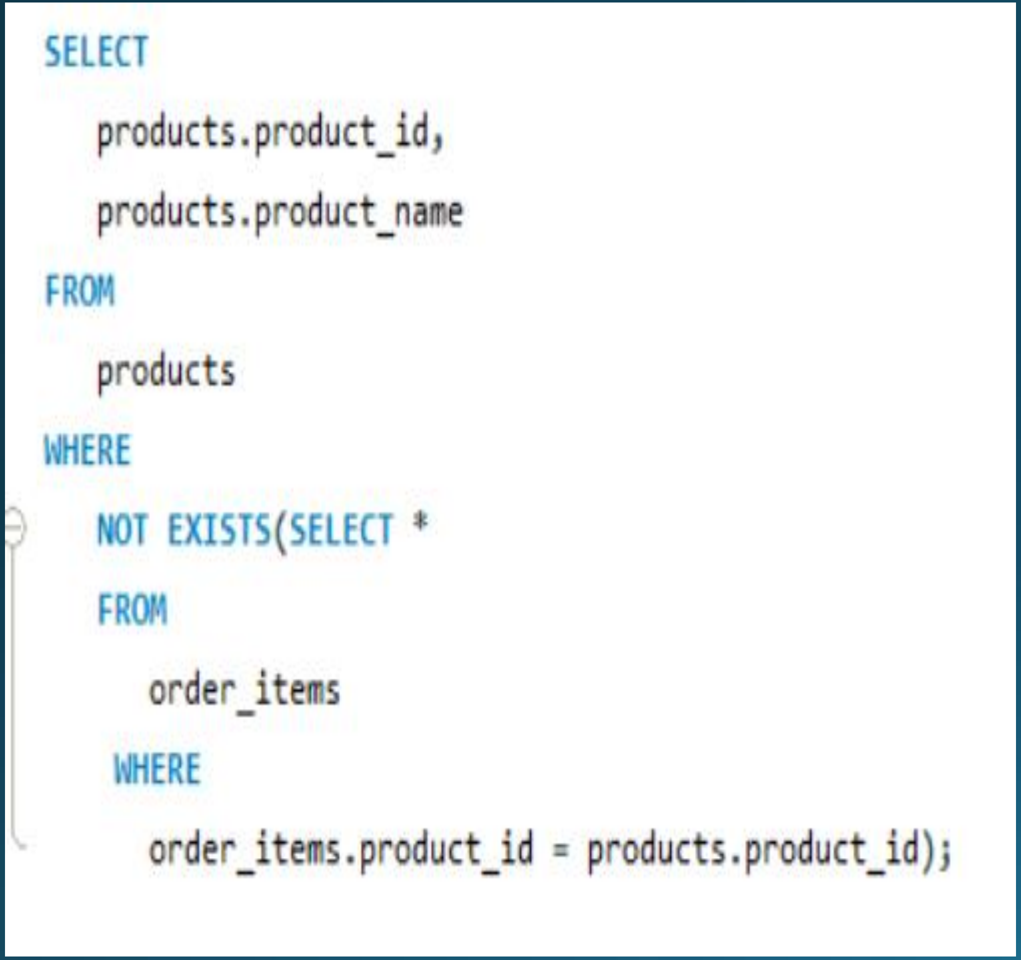


	MEDIAN
▶	74999.00

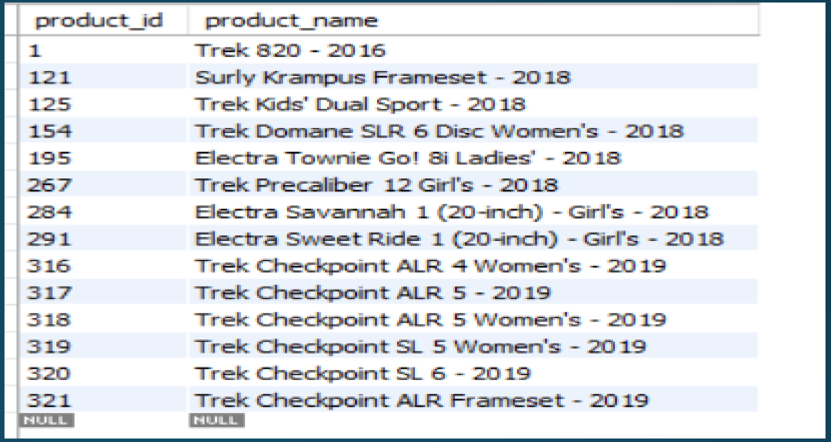
Key insights:

1.Calculates the median from the list_price column of the products table.
2.Uses ROW_NUMBER() to assign ranks based on ascending price.
3.Computes total count n using COUNT(*) OVER().
4.If n is even, takes the average of the two middle values.
5.If n is odd, selects the middle-ranked value.
6.Returns the median price from the derived table.

LIST ALL PRODUCTS THAT HAVE NEVER BEEN ORDERED.(USE EXISTS)



```
SELECT
    products.product_id,
    products.product_name
FROM
    products
WHERE
    NOT EXISTS(SELECT *
    FROM
        order_items
     WHERE
        order_items.product_id = products.product_id);
```

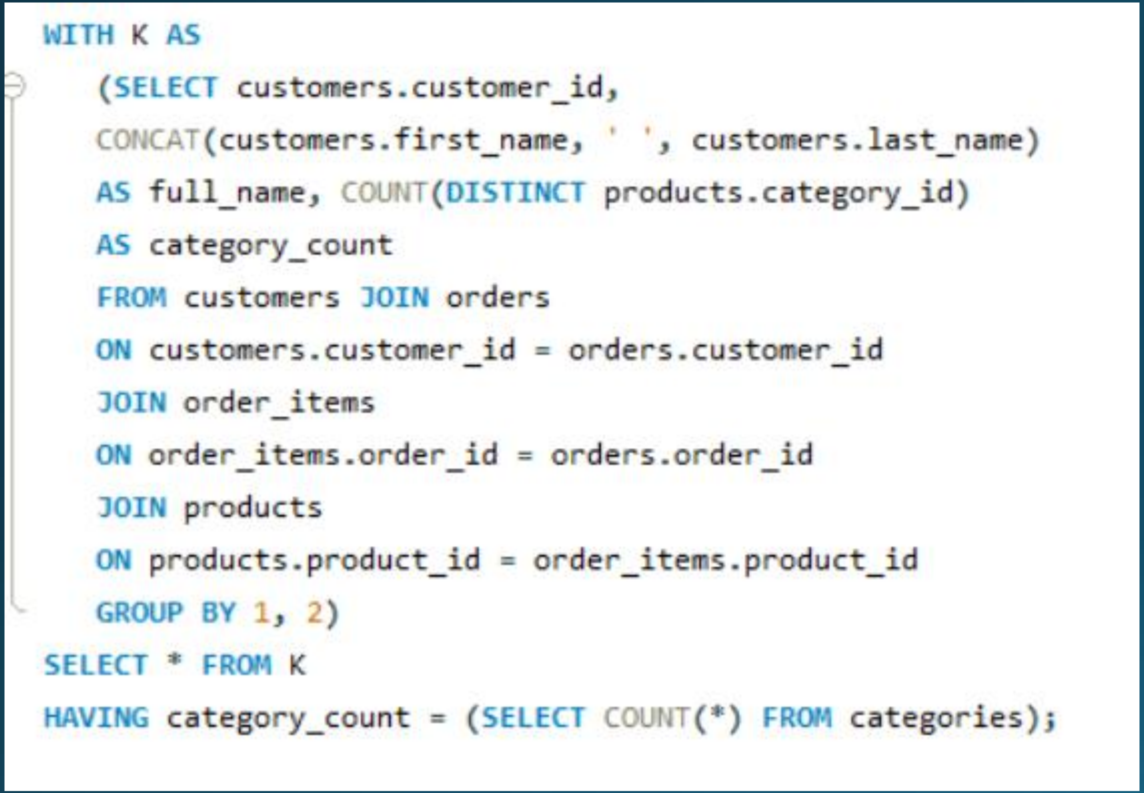


product_id	product_name
1	Trek 820 - 2016
121	Surly Krampus Frameset - 2018
125	Trek Kids' Dual Sport - 2018
154	Trek Domane SLR 6 Disc Women's - 2018
195	Electra Townie Go! 8i Ladies' - 2018
267	Trek Precaliber 12 Girl's - 2018
284	Electra Savannah 1 (20-inch) - Girl's - 2018
291	Electra Sweet Ride 1 (20-inch) - Girl's - 2018
316	Trek Checkpoint ALR 4 Women's - 2019
317	Trek Checkpoint ALR 5 - 2019
318	Trek Checkpoint ALR 5 Women's - 2019
319	Trek Checkpoint SL 5 Women's - 2019
320	Trek Checkpoint SL 6 - 2019
321	Trek Checkpoint ALR Frameset - 2019
NULL	NULL

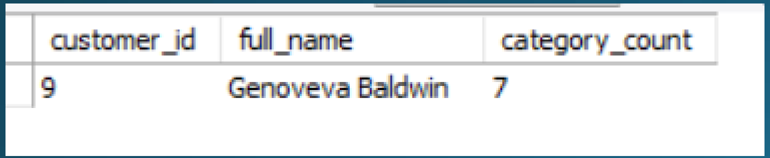
Key Insight:

This query identifies all products that have never been ordered by checking for the absence of matching entries in the order_items table. Using the NOT EXISTS clause ensures efficient filtering of only those products with zero sales history, helping businesses track inactive or unsold inventory.

Identify the customers who have ordered all types of products (i.e., from every category).



```
WITH K AS
    (SELECT customers.customer_id,
    CONCAT(customers.first_name, ' ', customers.last_name)
    AS full_name, COUNT(DISTINCT products.category_id)
    AS category_count
    FROM customers JOIN orders
    ON customers.customer_id = orders.customer_id
    JOIN order_items
    ON order_items.order_id = orders.order_id
    JOIN products
    ON products.product_id = order_items.product_id
    GROUP BY 1, 2)
SELECT * FROM K
HAVING category_count = (SELECT COUNT(*) FROM categories);
```

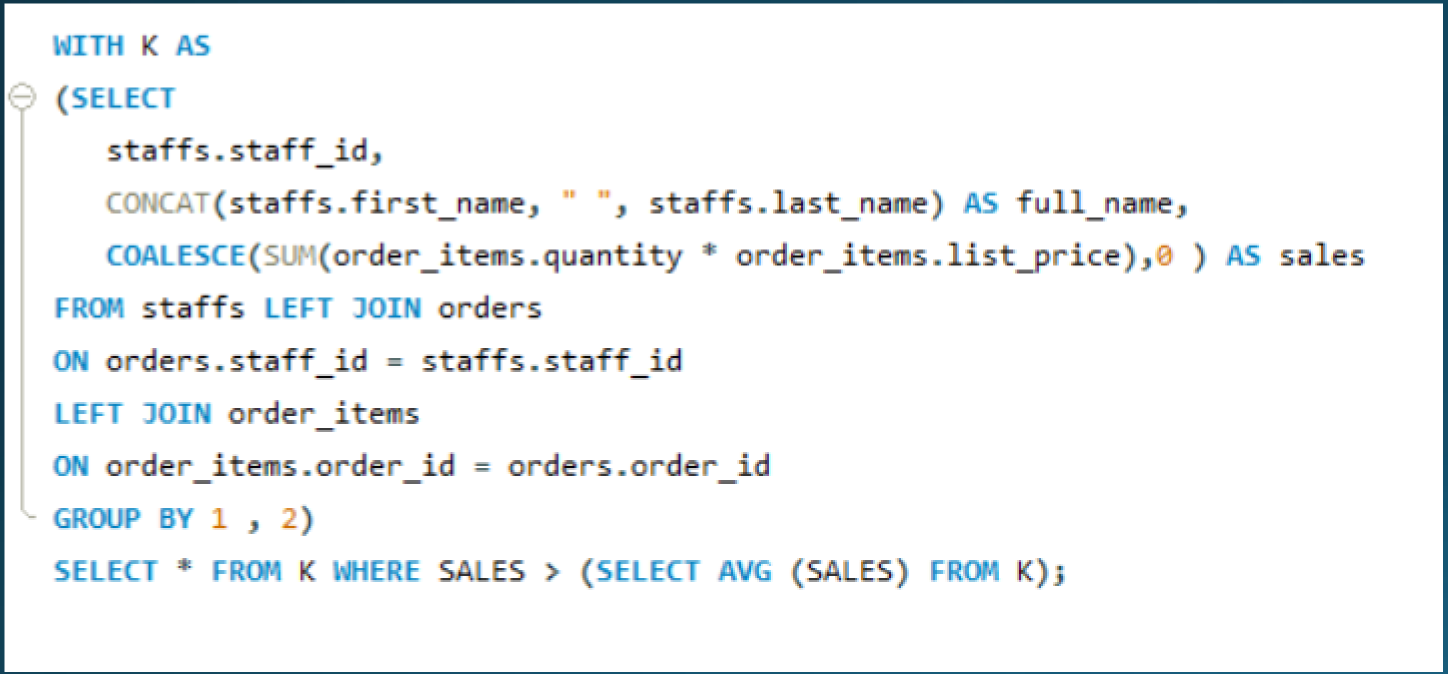


customer_id	full_name	category_count
9	Genoveva Baldwin	7

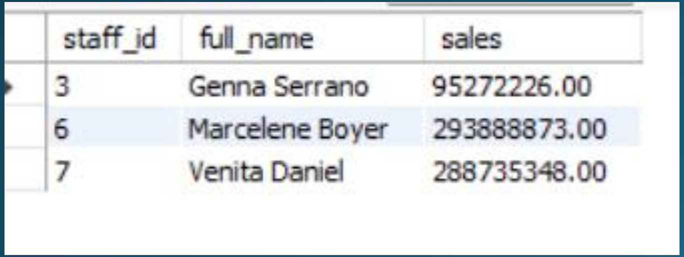
<u>Key Insight:</u>

This analysis identifies customers who have purchased products from every available category, indicating a high level of engagement and product interest. Such customers are valuable for loyalty programs, cross-selling opportunities, and targeted marketing due to their diverse purchasing behavior.

STAFF MEMBERS WHO HAVE MADE MORE SALES THAN THE AVERAGE NUMBER OF SALES BY ALL STAFF MEMBERS.



```sql
WITH K AS
(SELECT
   staffs.staff_id,
   CONCAT(staffs.first_name, " ", staffs.last_name) AS full_name,
   COALESCE(SUM(order_items.quantity * order_items.list_price),0 ) AS sales
FROM staffs LEFT JOIN orders
ON orders.staff_id = staffs.staff_id
LEFT JOIN order_items
ON order_items.order_id = orders.order_id
GROUP BY 1 , 2)
SELECT * FROM K WHERE SALES > (SELECT AVG (SALES) FROM K);
```

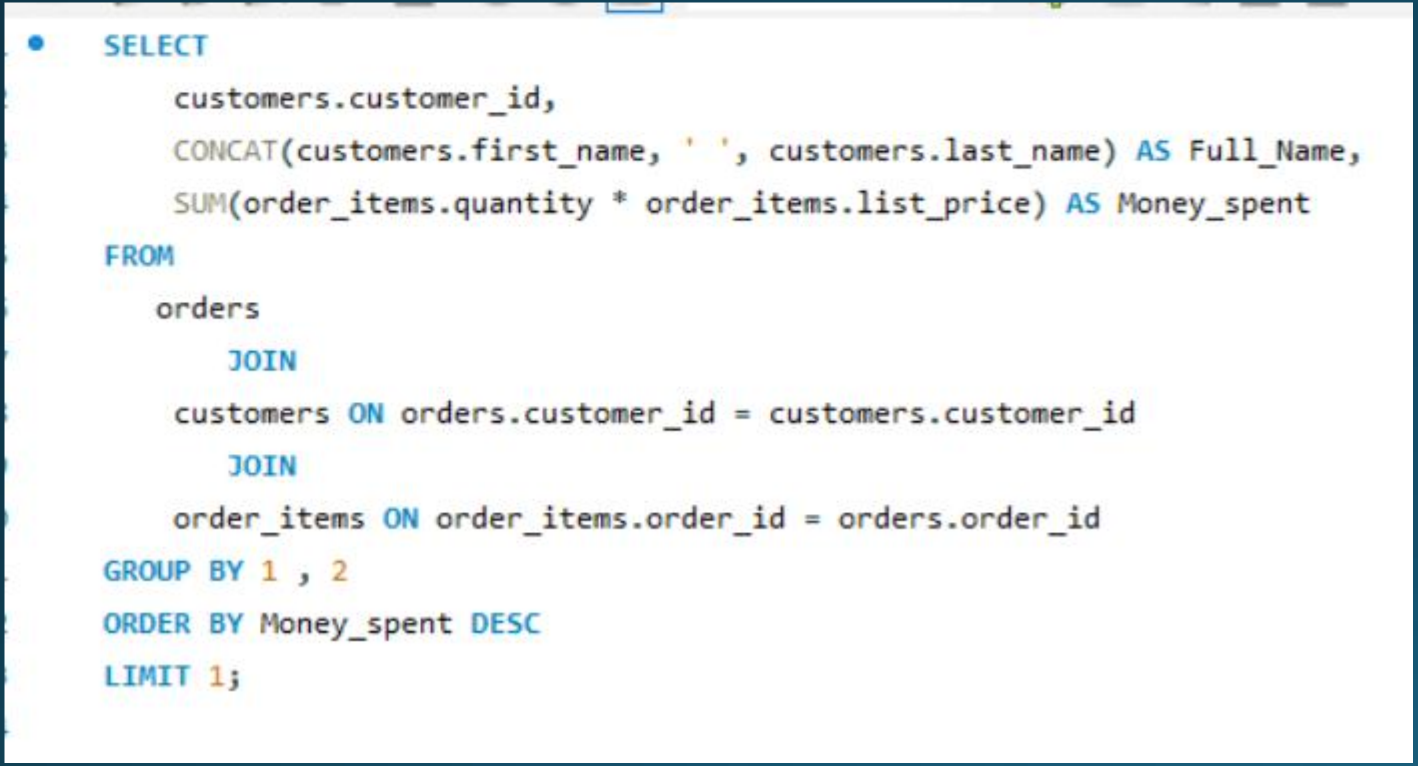


staff_id	full_name	sales
3	Genna Serrano	95272226.00
6	Marcelene Boyer	293888873.00
7	Venita Daniel	288735348.00

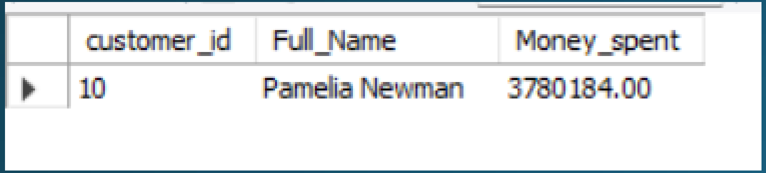
<u>Key Insight:</u>

This analysis highlights staff members who have generated above-average sales, helping to identify top-performing employees. Recognizing these individuals can guide performance rewards, training strategies, and sales optimization efforts, ultimately
driving higher revenue.

Find the customer who spent the most money on orders.



```
SELECT
    customers.customer_id,
    CONCAT(customers.first_name, ' ', customers.last_name) AS Full_Name,
    SUM(order_items.quantity * order_items.list_price) AS Money_spent
FROM
   orders
      JOIN
    customers ON orders.customer_id = customers.customer_id
      JOIN
    order_items ON order_items.order_id = orders.order_id
GROUP BY 1 , 2
ORDER BY Money_spent DESC
LIMIT 1;
```



```
customer_id  Full_Name       Money_spent
10           Pamelia Newman  3780184.00
```

Key Insight:

This analysis identifies the top-spending customer, which is crucial for recognizing high-value clients. Such insights help in crafting targeted loyalty programs, offering personalized services, and improving customer retention strategies by focusing on those who contribute significantly to revenue.

FOUND THESE INSIGHTS VALUABLE LIKE , SHARE , AND SAVE TO STAY AHEAD IN DATA ANALYTICS.