

SQL JOINS: SNOWFLAKE

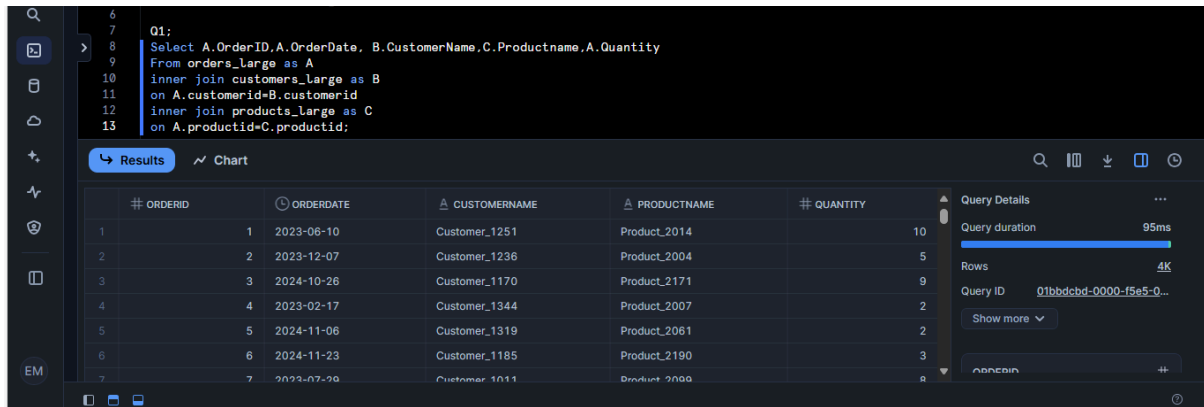
1. INNER JOIN: Orders with Customer and Product Names

Question:

List all orders along with the customer name and product name.

Expected Output Columns:

- OrderID, OrderDate, CustomerName, ProductName, Quantity



The screenshot shows a Snowflake SQL IDE interface. The query editor on the left contains the following SQL code:

```
Q1:
Select A.OrderID, A.OrderDate, B.CustomerName, C.Productname, A.Quantity
From orders_large as A
inner join customers_large as B
on A.customerid=B.customerid
inner join products_large as C
on A.productid=C.productid;
```

The results pane on the right displays a table with 7 rows and 5 columns: ORDERID, ORDERDATE, CUSTOMERNAME, PRODUCTNAME, and QUANTITY. The data is as follows:

#	ORDERID	ORDERDATE	CUSTOMERNAME	PRODUCTNAME	QUANTITY
1	1	2023-08-10	Customer_1251	Product_2014	10
2	2	2023-12-07	Customer_1236	Product_2004	5
3	3	2024-10-26	Customer_1170	Product_2171	9
4	4	2023-02-17	Customer_1344	Product_2007	2
5	5	2024-11-06	Customer_1319	Product_2061	2
6	6	2024-11-23	Customer_1185	Product_2190	3
7	7	2023-07-29	Customer_1011	Product_2009	8

On the far right, the 'Query Details' pane shows a query duration of 95ms and 4K rows. A 'Show more' button is visible below the details.

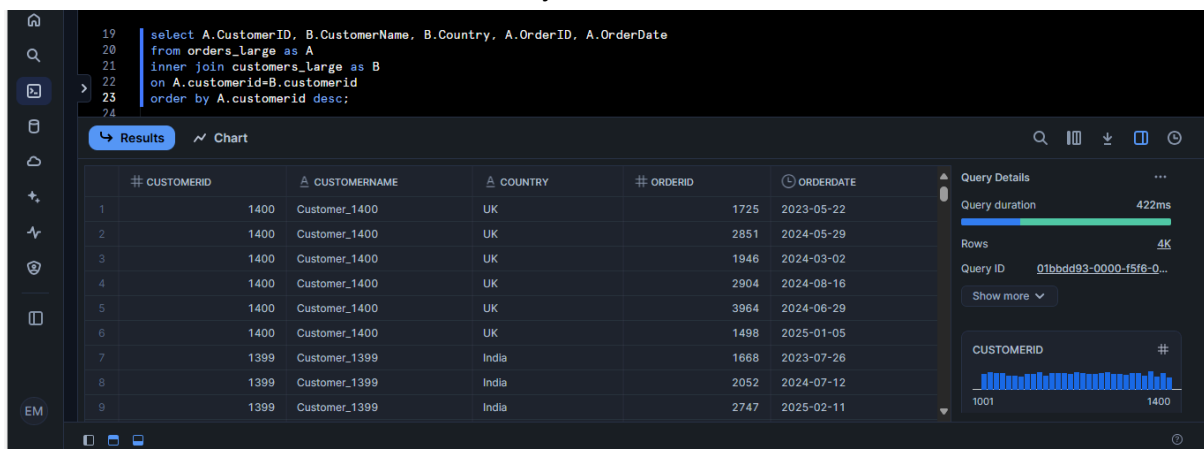
2. INNER JOIN: Customers Who Placed Orders

Question:

Which customers have placed at least one order?

Expected Output Columns:

- CustomerID, CustomerName, Country, OrderID, OrderDate



The screenshot shows a Snowflake SQL IDE interface. The query editor on the left contains the following SQL code:

```
select A.CustomerID, B.CustomerName, B.Country, A.OrderID, A.OrderDate
from orders_large as A
inner join customers_large as B
on A.customerid=B.customerid
order by A.customerid desc;
```

The results pane on the right displays a table with 9 rows and 5 columns: CUSTOMERID, CUSTOMERNAME, COUNTRY, ORDERID, and ORDERDATE. The data is as follows:

#	CUSTOMERID	CUSTOMERNAME	COUNTRY	ORDERID	ORDERDATE
1	1400	Customer_1400	UK	1725	2023-05-22
2	1400	Customer_1400	UK	2851	2024-05-29
3	1400	Customer_1400	UK	1946	2024-03-02
4	1400	Customer_1400	UK	2904	2024-08-16
5	1400	Customer_1400	UK	3964	2024-06-29
6	1400	Customer_1400	UK	1498	2025-01-05
7	1399	Customer_1399	India	1668	2023-07-26
8	1399	Customer_1399	India	2052	2024-07-12
9	1399	Customer_1399	India	2747	2025-02-11

On the far right, the 'Query Details' pane shows a query duration of 422ms and 4K rows. A 'Show more' button is visible below the details. Below the details, there is a histogram for the 'CUSTOMERID' column, showing a distribution of values from 1001 to 1400.

3. LEFT JOIN: All Customers and Their Orders

Question:

List all customers and any orders they might have placed. Include customers who have not placed any orders.

Expected Output Columns:

- CustomerID, CustomerName, Country, OrderID, OrderDate, ProductID, Quantity

The screenshot shows a SQL query editor with the following query:

```
26 select A.CustomerID, CustomerName, Country, OrderID, OrderDate, A.ProductID, Quantity
27 from orders_large as A
28 left join customers_large as B
29 on A.customerid=B.customerid
30 left join products_large as C
31 on A.productid=C.productid;
```

The results table displays the following data:

#	CUSTOMERID	CUSTOMERNAME	COUNTRY	#	ORDERID	ORDERDATE	#	PRODUCTID	QUANTITY
1	1251	Customer_1251	Germany	1	2023-06-10	2014	10		
2	1236	Customer_1236	Australia	2	2023-12-07	2004	5		
3	1170	Customer_1170	Germany	3	2024-10-26	2171	9		
4	1344	Customer_1344	Canada	4	2023-02-17	2007	2		
5	1319	Customer_1319	USA	5	2024-11-06	2061	2		
6	1185	Customer_1185	Australia	6	2024-11-23	2190	3		
7	1011	Customer_1011	Germany	7	2023-07-29	2099	8		
8	1322	Customer_1322	Australia	8	2023-12-06	2078	7		
9	1224	Customer_1224	Australia	9	2025-01-25	2043	7		

Query Details: Query duration 41ms, Rows 4K, Query ID 01bbddb3-0000-f5f6-0...

4. LEFT JOIN: Product Order Count

Question:

List all products and how many times each was ordered (if any).

Expected Output Columns:

- ProductID, ProductName, TotalOrders (TotalOrders is the count of how many times the product appears in orders)

The screenshot shows a SQL query editor with the following query:

```
34 SELECT A.ProductID, ProductName, count(A.productid) as totalorders
35 from orders_large as A
36 left join products_large as C
37 on A.productid=C.productid
38 group by A.productid, C.productname;
```

The results table displays the following data:

#	PRODUCTID	PRODUCTNAME	#	TOTALORDERS
1	2014	Product_2014	22	
2	2004	Product_2004	24	
3	2171	Product_2171	15	
4	2007	Product_2007	12	
5	2061	Product_2061	18	
6	2190	Product_2190	20	
7	2099	Product_2099	26	
8	2078	Product_2078	18	
9	2043	Product_2043	22	

Query Details: Query duration 700ms, Rows 200, Query ID 01bbdd67-0000-f5ff-00...

5. RIGHT JOIN: Orders with Product Info (Include Products Not Ordered)

Question:

Find all orders along with product details, including any products that might not have been ordered.

Expected Output Columns:

- OrderID, OrderDate, ProductID, ProductName, Price, Quantity



The screenshot shows a SQL query editor with a query window on the left and a results window on the right. The query is a RIGHT JOIN between orders_large and products_large. The results window displays a table with 9 rows and 6 columns: #, ORDERID, ORDERDATE, #, PRODUCTID, and #, QUANTITY. The first 8 rows show data for orders and products, while the 9th row shows data for a product that has not been ordered.

```
43 right join orders_large as A
44 on C.productid=A.productid
45 right join customers_large as B
46 on A.customerid=B.customerid;
```

#	ORDERID	ORDERDATE	#	PRODUCTID	#, QUANTITY
1	1	2023-06-10	2014	Product_2014	522 10
2	2	2023-12-07	2004	Product_2004	1996 5
3	3	2024-10-26	2171	Product_2171	76 9
4	4	2023-02-17	2007	Product_2007	156 2
5	5	2024-11-06	2061	Product_2061	1595 2
6	6	2024-11-23	2190	Product_2190	1755 3
7	7	2023-07-29	2099	Product_2099	1674 8
8	8	2023-12-06	2078	Product_2078	333 7
9	9	2025-01-25	2043	Product_2043	1647 7

Query Details: Query duration 27ms, Rows 4K, Query ID 01bbddde-0000-f5f6-0...

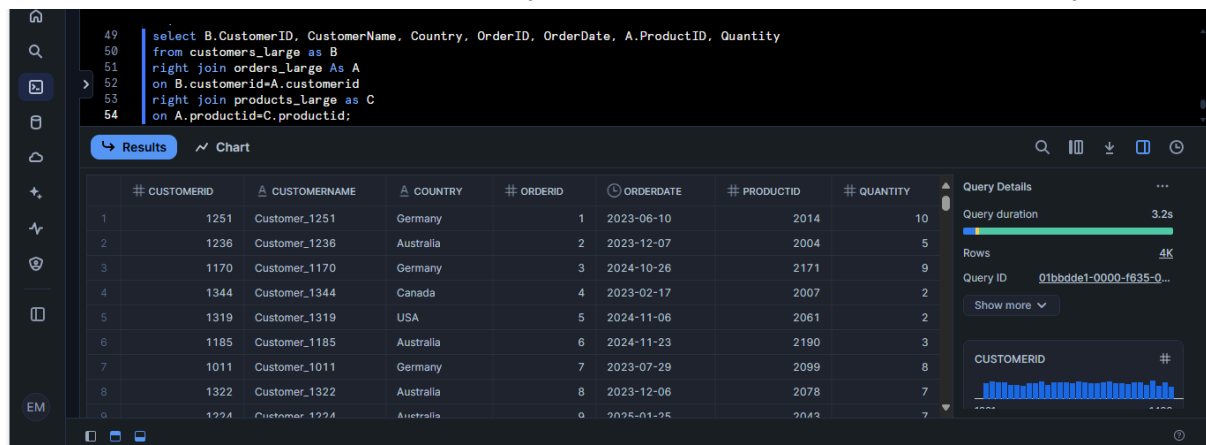
6. RIGHT JOIN: Customer Info with Orders (Include All Customers)

Question:

Which customers have made orders, and include customers even if they have never placed an order.

Expected Output Columns:

- CustomerID, CustomerName, Country, OrderID, OrderDate, ProductID, Quantity



The screenshot shows a SQL query editor with a query window on the left and a results window on the right. The query is a RIGHT JOIN between customers_large and orders_large. The results window displays a table with 9 rows and 7 columns: #, CUSTOMERID, CUSTOMERNAME, COUNTRY, #, ORDERID, ORDERDATE, #, PRODUCTID, and #, QUANTITY. The first 8 rows show data for customers and orders, while the 9th row shows data for a customer that has not placed an order.

```
49 select B.CustomerID, CustomerName, Country, OrderID, OrderDate, A.ProductID, Quantity
50 from customers_large as B
51 right join orders_large As A
52 on B.customerid=A.customerid
53 right join products_large as C
54 on A.productid=C.productid;
```

#	CUSTOMERID	CUSTOMERNAME	COUNTRY	#	ORDERID	ORDERDATE	#	PRODUCTID	#, QUANTITY
1	1251	Customer_1251	Germany	1	2023-06-10	2014	10		
2	1236	Customer_1236	Australia	2	2023-12-07	2004	5		
3	1170	Customer_1170	Germany	3	2024-10-26	2171	9		
4	1344	Customer_1344	Canada	4	2023-02-17	2007	2		
5	1319	Customer_1319	USA	5	2024-11-06	2061	2		
6	1185	Customer_1185	Australia	6	2024-11-23	2190	3		
7	1011	Customer_1011	Germany	7	2023-07-29	2099	8		
8	1322	Customer_1322	Australia	8	2023-12-06	2078	7		
9	1224	Customer_1224	Australia	9	2025-01-25	2043	7		

Query Details: Query duration 3.2s, Rows 4K, Query ID 01bbdde1-0000-f635-0...

7. FULL OUTER JOIN: All Customers and All Orders

Question:

List all customers and orders, showing NULLs where customers have not ordered or where orders have no customer info.

Expected Output Columns:

- CustomerID, CustomerName, Country, OrderID, OrderDate, ProductID, Quantity

```
57 select A.CustomerID, CustomerName, Country, OrderID, OrderDate, A.ProductID, Quantity
58 from orders_large As A
59 full outer join customers_large As B
60 on A.customerid=B.customerid
61 full outer join products_large As C
62 on A.productid=C.productid;
```

#	CUSTOMERID	CUSTOMERNAME	COUNTRY	#	ORDERID	ORDERDATE	#	PRODUCTID	#	QUANTITY
1	1251	Customer_1251	Germany	1	2023-06-10			2014		10
2	1236	Customer_1236	Australia	2	2023-12-07			2004		5
3	1170	Customer_1170	Germany	3	2024-10-26			2171		9
4	1344	Customer_1344	Canada	4	2023-02-17			2007		2
5	1319	Customer_1319	USA	5	2024-11-06			2061		2
6	1185	Customer_1185	Australia	6	2024-11-23			2190		3
7	1011	Customer_1011	Germany	7	2023-07-29			2099		8
8	1322	Customer_1322	Australia	8	2023-12-06			2078		7
9	1224	Customer_1224	Australia	9	2025-01-25			2043		7

8. FULL OUTER JOIN: All Products and Orders

Question:

List all products and orders, showing NULLs where products were never ordered or orders are missing product info.

Expected Output Columns:

- ProductID, ProductName, Price, OrderID, OrderDate, CustomerID, Quantity

```
74 select A.ProductID, ProductName, Price, OrderID, OrderDate, CustomerID, Quantity
75 from orders_large as A
76 full outer join products_large as C
77 on A.productid= C.productid;
```

#	PRODUCTID	PRODUCTNAME	#	PRICE	#	ORDERID	ORDERDATE	#	CUSTOMERID	#	QUANTITY
1	2014	Product_2014		522	1	2023-06-10			1251		10
2	2004	Product_2004		1996	2	2023-12-07			1236		5
3	2171	Product_2171		76	3	2024-10-26			1170		9
4	2007	Product_2007		156	4	2023-02-17			1344		2
5	2061	Product_2061		1595	5	2024-11-06			1319		2
6	2190	Product_2190		1755	6	2024-11-23			1185		3
7	2099	Product_2099		1674	7	2023-07-29			1011		8
8	2078	Product_2078		333	8	2023-12-06			1322		7
9	2043	Product_2043		1047	9	2025-01-25			1224		7