

BrightLight Data Analytics

SQL JOIN Practice Instructions

General Guidelines:

1. Use the provided CSV files:
 - orders_large.csv
 - products_large.csv
 - customers_large.csv
2. Import the datasets into a SQL database system of your choice (e.g., MySQL, PostgreSQL, SQLite, or Snowflake).
3. Use the appropriate JOINS (INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL OUTER JOIN) to answer each question.
4. Ensure your query returns exactly the **expected columns** listed for each question.
5. Use aliases and format your SQL code for readability.
6. When using FULL OUTER JOIN, make sure your SQL engine supports it (e.g., MySQL does not support it natively — use UNION of LEFT JOIN and RIGHT JOIN if needed).

SQL JOIN Practice Questions

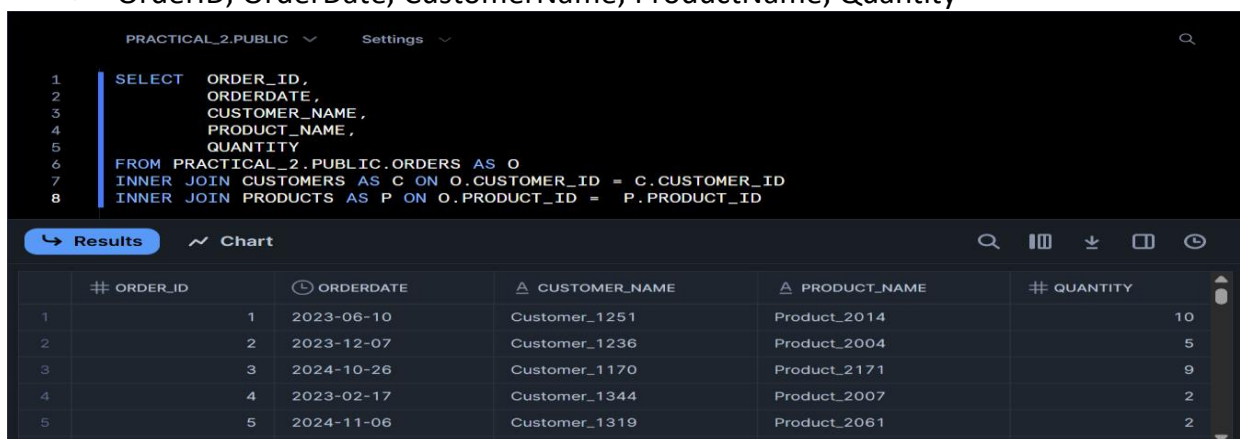
1. INNER JOIN: Orders with Customer and Product Names

Question:

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

Expected Output Columns:

- OrderID, OrderDate, CustomerName, ProductName, Quantity



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

```
1 SELECT ORDER_ID,  
2 ORDERDATE,  
3 CUSTOMER_NAME,  
4 PRODUCT_NAME,  
5 QUANTITY  
6 FROM PRACTICAL_2.PUBLIC.ORDERS AS O  
7 INNER JOIN CUSTOMERS AS C ON O.CUSTOMER_ID = C.CUSTOMER_ID  
8 INNER JOIN PRODUCTS AS P ON O.PRODUCT_ID = P.PRODUCT_ID
```

Below the query editor, the 'Results' tab is active, displaying a table with 5 rows and 6 columns. The columns are: #, ORDER_ID, ORDERDATE, CUSTOMER_NAME, PRODUCT_NAME, and # QUANTITY. The data is as follows:

#	ORDER_ID	ORDERDATE	CUSTOMER_NAME	PRODUCT_NAME	# QUANTITY
1	1	2023-06-10	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

2. INNER JOIN: Customers Who Placed Orders

Question:

Which customers have placed at least one order?

Expected Output Columns:

- CustomerID, CustomerName, Country, OrderID, OrderDate

PRACTICAL_2.PUBLIC ▾ Settings ▾

1 SELECT C.CUSTOMER_ID,
2 CUSTOMER_NAME,
3 COUNTRY,
4 ORDER_ID,
5 ORDERDATE
6 FROM PRACTICAL_2.PUBLIC.CUSTOMERS AS C
7 INNER JOIN ORDERS AS O ON O.CUSTOMER_ID = C.CUSTOMER_ID
8

Results

Chart

🔍 📄 ⬇️ 📱 ⌛

	# CUSTOMER_ID	A CUSTOMER_NAME	A COUNTRY	# ORDER_ID	🕒 ORDERDATE
1	1251	Customer_1251	Germany	1	2023-06-10
2	1236	Customer_1236	Australia	2	2023-12-07
3	1170	Customer_1170	Germany	3	2024-10-26
4	1344	Customer_1344	Canada	4	2023-02-17
5	1319	Customer_1319	USA	5	2024-11-06
6	1185	Customer_1185	Australia	6	2024-11-23

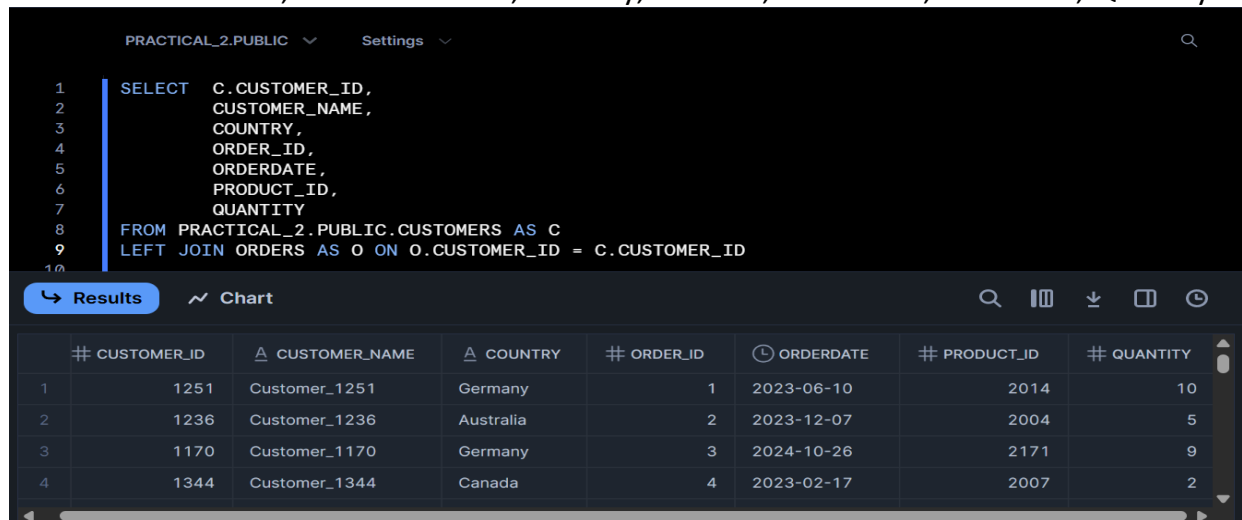
3. LEFT JOIN: All Customers and Their Orders

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 IDE interface with a query editor and a results pane. The query is a LEFT JOIN between the CUSTOMERS and ORDERS tables. The results pane shows a table with 7 columns: CUSTOMER_ID, CUSTOMER_NAME, COUNTRY, ORDER_ID, ORDERDATE, PRODUCT_ID, and QUANTITY. There are 4 rows of data.

```
1 SELECT C.CUSTOMER_ID,  
2        CUSTOMER_NAME,  
3        COUNTRY,  
4        ORDER_ID,  
5        ORDERDATE,  
6        PRODUCT_ID,  
7        QUANTITY  
8 FROM PRACTICAL_2.PUBLIC.CUSTOMERS AS C  
9 LEFT JOIN ORDERS AS O ON O.CUSTOMER_ID = C.CUSTOMER_ID  
10
```

#	CUSTOMER_ID	CUSTOMER_NAME	COUNTRY	ORDER_ID	ORDERDATE	PRODUCT_ID	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

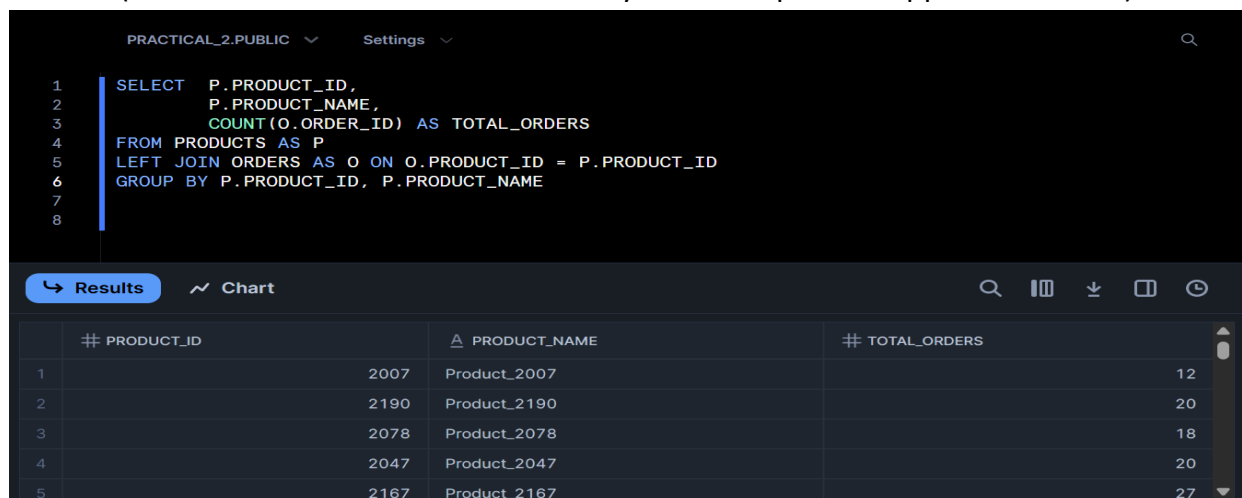
4. LEFT JOIN: Product Order Count

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 IDE interface with a query and its results. The query is a LEFT JOIN between the PRODUCTS and ORDERS tables, grouped by product. The results pane shows a table with 4 columns: PRODUCT_ID, PRODUCT_NAME, and TOTAL_ORDERS. There are 5 rows of data.

```
1 SELECT P.PRODUCT_ID,  
2        P.PRODUCT_NAME,  
3        COUNT(O.ORDER_ID) AS TOTAL_ORDERS  
4 FROM PRODUCTS AS P  
5 LEFT JOIN ORDERS AS O ON O.PRODUCT_ID = P.PRODUCT_ID  
6 GROUP BY P.PRODUCT_ID, P.PRODUCT_NAME  
7  
8
```

#	PRODUCT_ID	PRODUCT_NAME	TOTAL_ORDERS
1	2007	Product_2007	12
2	2190	Product_2190	20
3	2078	Product_2078	18
4	2047	Product_2047	20
5	2167	Product_2167	27

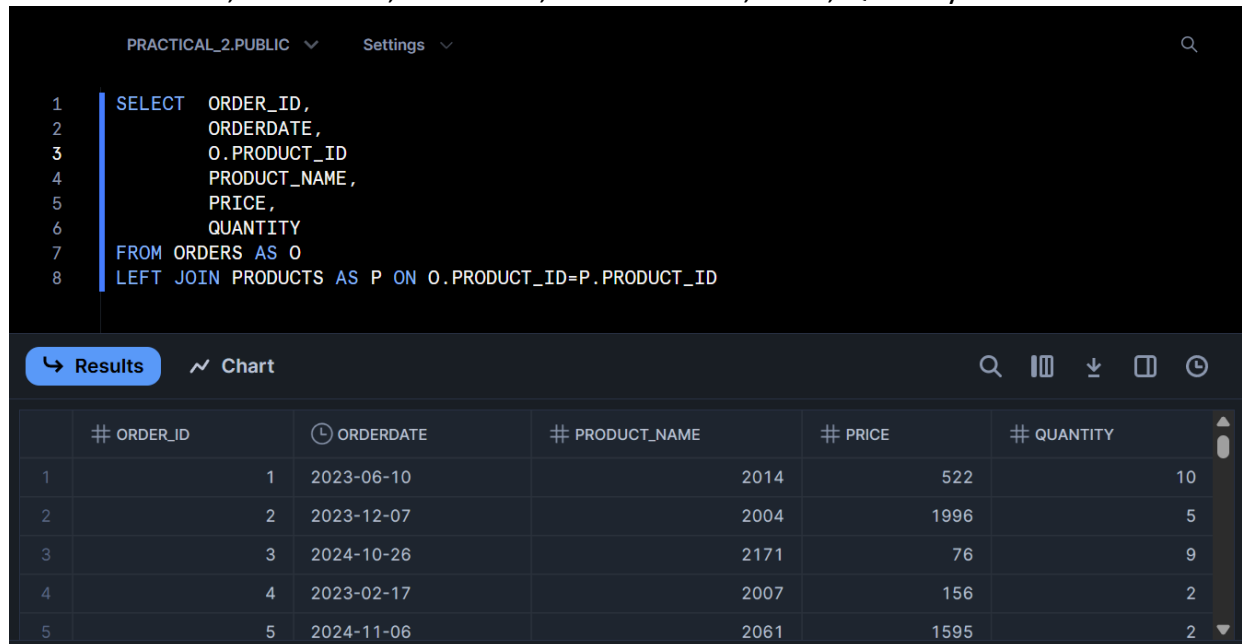
5. RIGHT JOIN: Orders with Product Info (Include Products Not Ordered)

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 the following query:

```
1 SELECT ORDER_ID,  
2 ORDERDATE,  
3 O.PRODUCT_ID  
4 PRODUCT_NAME,  
5 PRICE,  
6 QUANTITY  
7 FROM ORDERS AS O  
8 LEFT JOIN PRODUCTS AS P ON O.PRODUCT_ID=P.PRODUCT_ID
```

The results are displayed in a table with the following columns: # ORDER_ID, ORDERDATE, # PRODUCT_NAME, # PRICE, # QUANTITY.

	# ORDER_ID	ORDERDATE	# PRODUCT_NAME	# PRICE	# QUANTITY
1	1	2023-06-10	2014	522	10
2	2	2023-12-07	2004	1996	5
3	3	2024-10-26	2171	76	9
4	4	2023-02-17	2007	156	2
5	5	2024-11-06	2061	1595	2

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 the following query:

```
1 SELECT C.CUSTOMER_ID,  
2 CUSTOMER_NAME,  
3 COUNTRY,  
4 ORDER_ID,  
5 ORDERDATE,  
6 PRODUCT_ID,  
7 QUANTITY  
8 FROM ORDERS AS O  
9 RIGHT JOIN CUSTOMERS AS C ON O.CUSTOMER_ID = C.CUSTOMER_ID
```

The results are displayed in a table with the following columns: # CUSTOMER_ID, CUSTOMER_NAME, COUNTRY, # ORDER_ID, ORDERDATE, # PRODUCT_ID, # QUANTITY.

	# CUSTOMER_ID	CUSTOMER_NAME	COUNTRY	# ORDER_ID	ORDERDATE	# PRODUCT_ID	# 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

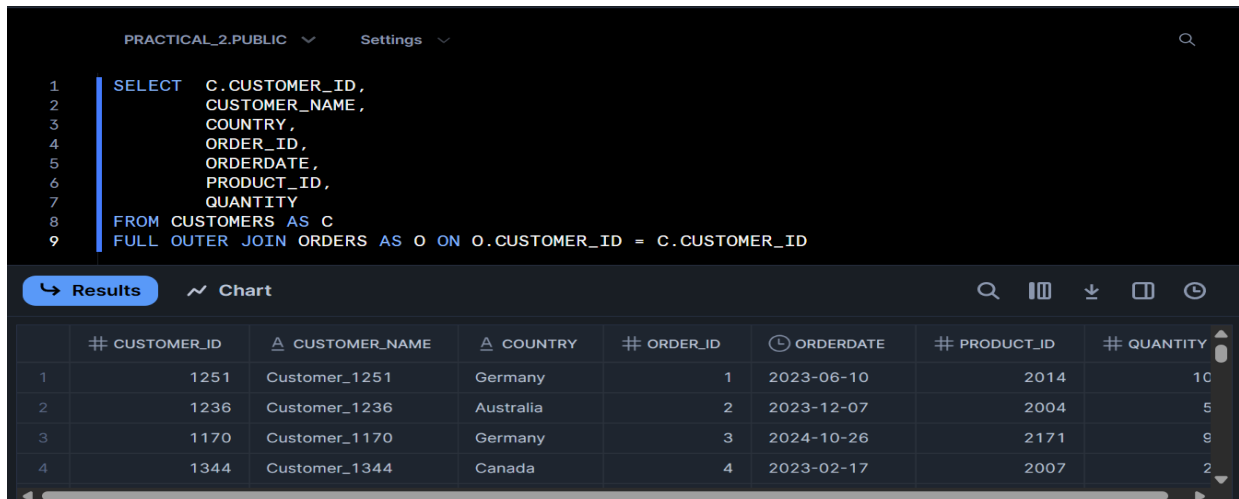
7. FULL OUTER JOIN: All Customers and All Orders

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



The screenshot shows a SQL IDE interface. At the top, there are tabs for 'PRACTICAL_2.PUBLIC' and 'Settings'. Below the tabs is a search icon. The main area displays a SQL query:

```
1 SELECT C.CUSTOMER_ID,  
2       CUSTOMER_NAME,  
3       COUNTRY,  
4       ORDER_ID,  
5       ORDERDATE,  
6       PRODUCT_ID,  
7       QUANTITY  
8 FROM CUSTOMERS AS C  
9 FULL OUTER JOIN ORDERS AS O ON O.CUSTOMER_ID = C.CUSTOMER_ID
```

Below the query editor, there are two tabs: 'Results' (selected) and 'Chart'. The 'Results' tab shows a table with 8 columns: CUSTOMER_ID, CUSTOMER_NAME, COUNTRY, ORDER_ID, ORDERDATE, PRODUCT_ID, and QUANTITY. The table contains 4 rows of data:

	# CUSTOMER_ID	△ CUSTOMER_NAME	△ COUNTRY	# ORDER_ID	🕒 ORDERDATE	# PRODUCT_ID	# 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

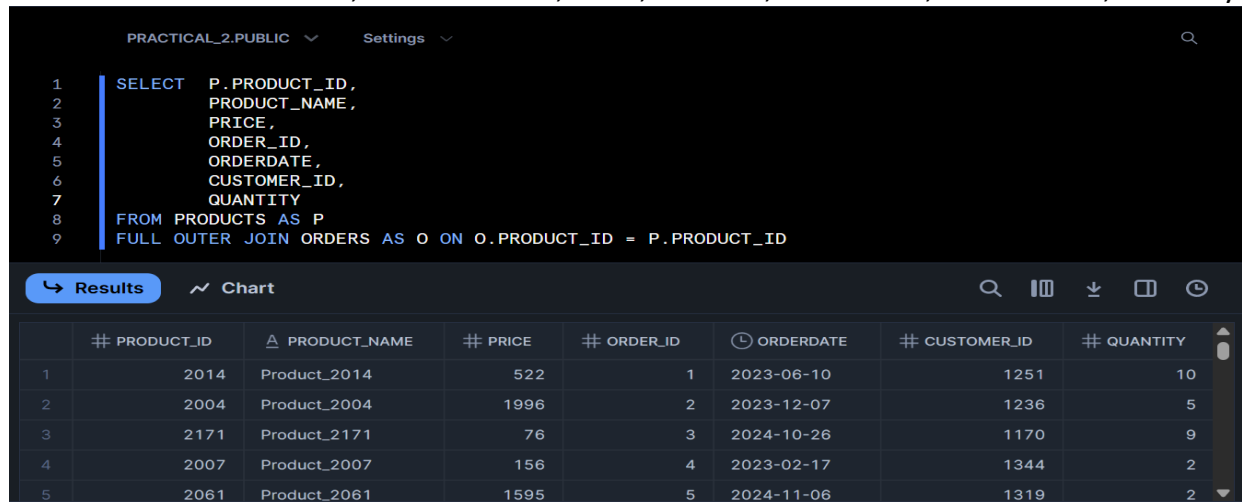
8.FULL OUTER JOIN: All Products and

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



The screenshot shows a SQL IDE interface. At the top, there's a header with 'PRACTICAL_2.PUBLIC' and 'Settings'. Below that, a SQL query is entered in a text area. The query is a SELECT statement that joins the 'PRODUCTS' table (aliased as 'P') with the 'ORDERS' table (aliased as 'O') using a FULL OUTER JOIN on the 'PRODUCT_ID' column. The selected columns are P.PRODUCT_ID, PRODUCT_NAME, PRICE, ORDER_ID, ORDERDATE, CUSTOMER_ID, and QUANTITY. Below the query editor, there's a 'Results' tab which is active, showing a table with 8 columns: # PRODUCT_ID, PRODUCT_NAME, PRICE, ORDER_ID, ORDERDATE, CUSTOMER_ID, and QUANTITY. The table contains 5 rows of data, representing the first 5 products and their corresponding orders.

```
1 SELECT P.PRODUCT_ID,  
2        PRODUCT_NAME,  
3        PRICE,  
4        ORDER_ID,  
5        ORDERDATE,  
6        CUSTOMER_ID,  
7        QUANTITY  
8 FROM PRODUCTS AS P  
9 FULL OUTER JOIN ORDERS AS O ON O.PRODUCT_ID = P.PRODUCT_ID
```

	# PRODUCT_ID	PRODUCT_NAME	PRICE	ORDER_ID	ORDERDATE	CUSTOMER_ID	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