

Practical 2: SQL JOIN Practice Instructions

1. INNER JOIN: Orders with Customer and Product Names

The screenshot shows a SQL IDE interface with a dark theme. On the left, a sidebar displays a database schema for 'BRIGHTLEARN_DATA_ANAL...' with tables 'CUSTOMERS', 'ORDERS', and 'PRODUCTS'. The main editor contains a SQL query for an INNER JOIN. The query is as follows:

```
--Question: List all orders along with the customer name and product name.
--Expected Output Columns:
--OrderID, OrderDate, CustomerName, ProductName, Quantity
SELECT
  o.OrderID,
  o.OrderDate,
  c.CustomerName,
  p.ProductName,
  o.Quantity
FROM orders o
INNER JOIN customers c ON o.CustomerID = c.CustomerID
INNER JOIN products p ON o.ProductID = p.ProductID;
```

Below the query, the 'Results' tab shows a table with 7 rows and 5 columns: ORDERID, ORDERDATE, CUSTOMERNAME, PRODUCTNAME, and QUANTITY.

ORDERID	ORDERDATE	CUSTOMERNAME	PRODUCTNAME	QUANTITY
1	2023-06-10	Customer_1251	Product_2014	10
2	2023-12-07	Customer_1236	Product_2004	5
3	2024-10-26	Customer_1170	Product_2171	9
4	2023-02-17	Customer_1344	Product_2007	2
5	2024-11-06	Customer_1319	Product_2061	2
6	2024-11-23	Customer_1185	Product_2190	3
7	2023-07-26	Customer_1011	Product_2085	6

2. INNER JOIN: Customers Who Placed Orders

The screenshot shows the same SQL IDE interface. The main editor contains a SQL query for an INNER JOIN between customers and orders. The query is as follows:

```
--2. INNER JOIN: Customers Who Placed Orders
--Question: Which customers have placed at least one order?
--Expected Output Columns:
--CustomerID, CustomerName, Country, OrderID, OrderDate
SELECT
  c.CustomerID,
  c.CustomerName,
  c.Country,
  o.OrderID,
  o.OrderDate
FROM customers c
INNER JOIN orders o ON c.CustomerID = o.CustomerID;
```

Below the query, the 'Results' tab shows a table with 7 rows and 5 columns: CUSTOMERID, CUSTOMERNAME, COUNTRY, ORDERID, and ORDERDATE.

CUSTOMERID	CUSTOMERNAME	COUNTRY	ORDERID	ORDERDATE
1251	Customer_1251	Germany	1	2023-06-10
1236	Customer_1236	Australia	2	2023-12-07
1170	Customer_1170	Germany	3	2024-10-26
1344	Customer_1344	Canada	4	2023-02-17
1319	Customer_1319	USA	5	2024-11-06
1185	Customer_1185	Australia	6	2024-11-23
1011	Customer_1011	Germany	7	2023-07-26

3. LEFT JOIN: All Customers and Their Orders

The screenshot shows a Snowflake SQL editor interface. On the left, a sidebar displays the database schema for 'BRIGHTLEARN_DATA_ANAL'. The main editor area contains a SQL query for a LEFT JOIN. The query is as follows:

```
--  
29 --3. LEFT JOIN: All Customers and Their Orders  
30 --Question: List all customers and any orders they might have placed. Include customers who have not  
31 --placed any orders.  
32 --Expected Output Columns:  
33 --CustomerID, CustomerName, Country, OrderID, OrderDate, ProductID, Quantity  
34  
35 SELECT  
36   c.CustomerID,  
37   c.CustomerName,  
38   c.Country,  
39   o.OrderID,  
40   o.OrderDate,  
41   o.ProductID,  
42   o.Quantity  
43 FROM customers c  
44 LEFT JOIN orders o ON c.CustomerID = o.CustomerID;
```

Below the query, the 'Results' tab is active, displaying a table with 7 columns: CUSTOMERID, CUSTOMERNAME, COUNTRY, ORDERID, ORDERDATE, PRODUCTID, and QUANTITY. The table contains 5 rows of 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	2024-10-24	2171	9
4	1344	Customer_1344	Canada	4	2023-02-17	2007	7
5	1316	Customer_1316	USA	5	2024-11-26	2061	1

4. LEFT JOIN: Product Order Count

The screenshot shows a SQL IDE interface with a database explorer on the left and a query editor on the right. The database explorer shows a schema named 'PRACTICAL_3_SQLJOIN' with tables 'CUSTOMERS', 'ORDERS', and 'PRODUCTS'. The query editor contains a SQL query for a LEFT JOIN between 'customers' and 'orders'.

```
11 --B. LEFT JOIN: All Customers and Their Orders
12 --Question: List all customers and any orders they might have placed. Include customers who have not
13 --placed any orders.
14 --Expected Output Columns:
15 --CustomerID, CustomerName, Country, OrderID, OrderDate, ProductID, Quantity
16
17 SELECT
18     c.CustomerID,
19     c.CustomerName,
20     c.Country,
21     o.OrderID,
22     o.OrderDate,
23     o.ProductID,
24     o.Quantity
25 FROM customers c
26 LEFT JOIN orders o ON c.CustomerID = o.CustomerID;
```

The results pane shows 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	3171	9
4	1344	Customer_1344	Canada	4	2023-02-17	2007	2
5	1116	Customer_1116	USA	5	2024-11-24	2041	1

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

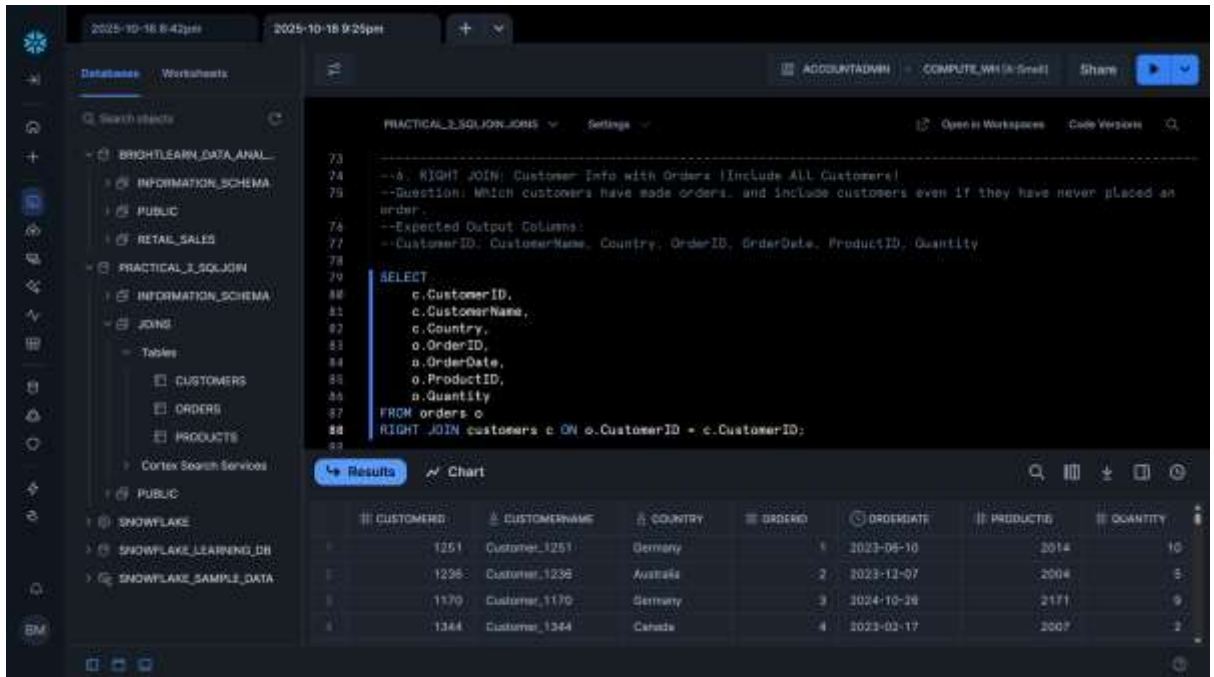
The screenshot shows a SQL IDE interface with a database explorer on the left and a query editor on the right. The database explorer shows a schema named 'PRACTICAL_3_SQLJOIN' with tables 'CUSTOMERS', 'ORDERS', and 'PRODUCTS'. The query editor contains a SQL query for a RIGHT JOIN between 'orders' and 'products'.

```
41 --B. RIGHT JOIN: Orders with Product Info (Include Products Not Ordered)
42 --Question: Find all orders along with product details, including any products that might not have been
43 --ordered.
44 --Expected Output Columns:
45 --OrderID, OrderDate, ProductID, ProductName, Price, Quantity
46
47 SELECT
48     o.OrderID,
49     o.OrderDate,
50     p.ProductID,
51     p.ProductName,
52     p.Price,
53     o.Quantity
54 FROM orders o
55 RIGHT JOIN products p ON o.ProductID = p.ProductID;
```

The results pane shows the following data:

#	ORDERID	ORDERDATE	PRODUCTID	PRODUCTNAME	PRICE	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	3171	Product_3171	76	9
4	4	2023-02-17	2007	Product_2007	158	2

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



The screenshot shows a SQL IDE interface with a query editor on the right and a results pane at the bottom. The query is a RIGHT JOIN between the CUSTOMERS and ORDERS tables, ensuring all customers are included.

```

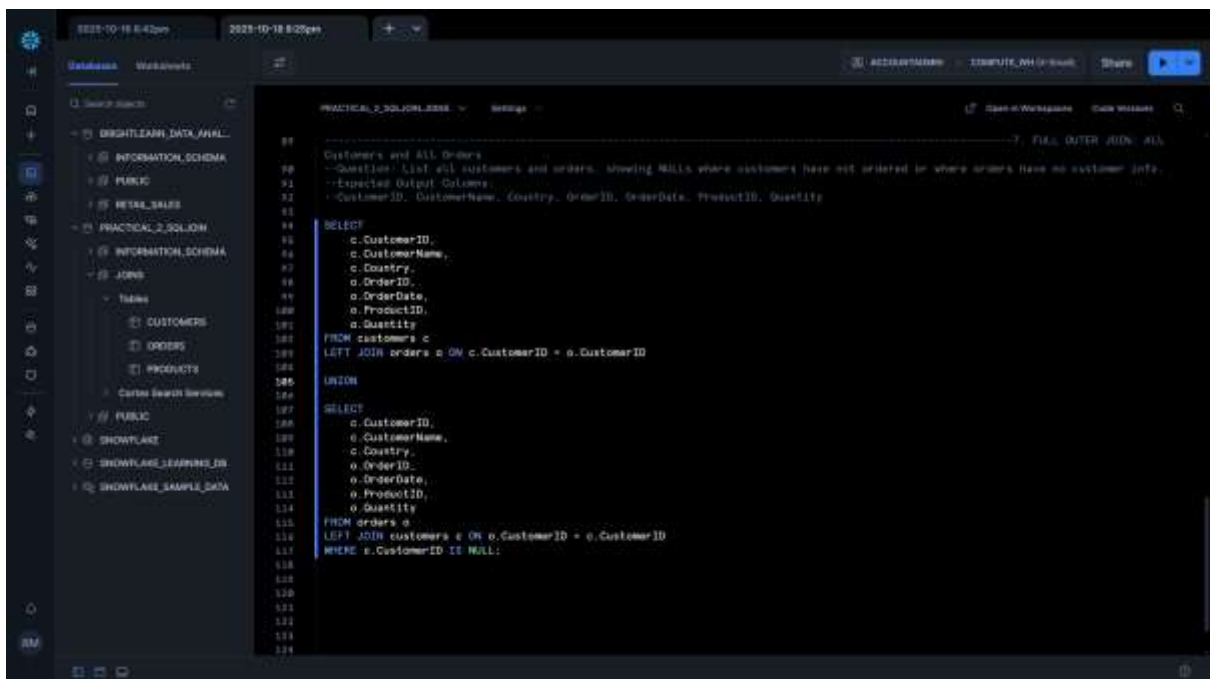
73
74 --6. RIGHT JOIN: Customer Info with Orders (Include All Customers)
75 --Question: Which customers have made orders, and include customers even if they have never placed an
76 --Expected Output Columns:
77 --CustomerID, CustomerName, Country, OrderID, OrderDate, ProductID, Quantity
78
79 SELECT
80     c.CustomerID,
81     c.CustomerName,
82     c.Country,
83     o.OrderID,
84     o.OrderDate,
85     o.ProductID,
86     o.Quantity
87 FROM orders o
88 RIGHT JOIN customers c ON o.CustomerID = c.CustomerID;
89

```

The results pane displays the following data:

	CUSTOMERID	CUSTOMERNAME	COUNTRY	ORDERID	ORDERDATE	PRODUCTID	QUANTITY
1	1251	Customer_1251	Germany	1	2023-06-10	2014	10
2	1296	Customer_1296	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 Syntax



The screenshot shows a SQL IDE interface with a query editor on the right. The query is a FULL OUTER JOIN between the CUSTOMERS and ORDERS tables, ensuring all data from both tables is included.

```

89
90 Customers and All Orders
91 --Question: List all customers and orders, showing NULLs where customers have not ordered or where orders have no customer info.
92 --Expected Output Columns:
93 --CustomerID, CustomerName, Country, OrderID, OrderDate, ProductID, Quantity
94
95 SELECT
96     c.CustomerID,
97     c.CustomerName,
98     c.Country,
99     o.OrderID,
100     o.OrderDate,
101     o.ProductID,
102     o.Quantity
103 FROM customers c
104 LEFT JOIN orders o ON c.CustomerID = o.CustomerID
105 UNION
106 SELECT
107     c.CustomerID,
108     c.CustomerName,
109     c.Country,
110     o.OrderID,
111     o.OrderDate,
112     o.ProductID,
113     o.Quantity
114 FROM orders o
115 LEFT JOIN customers c ON o.CustomerID = c.CustomerID
116 WHERE o.CustomerID IS NULL;
117
118
119
120
121
122
123
124

```

Results

2023-10-18 8:42pm2023-10-18 8:25pm

DatabaseWorkbooks

Search objects

BRIGHTLEARN_DATA_AWAL

INFORMATION_SCHEMA

PUBLIC

RETAIL_SALES

PRACTICAL_2_SOLID

INFORMATION_SCHEMA

JOBS

Customers

ORDERS

PRODUCTS

Custom Search Services

PUBLIC

SHOWPLATE

SHOWPLATE_LEARNING_DB

SHOWPLATE_SAMPLE_DATA

PRACTICAL_2_SOLIDLESSettings

Open in WorkbooksQuick Worksheet

100LEFT JOIN orders o ON o.CustomerID = o.CustomerID101102UNION103104105SELECT106o.CustomerID,107o.CustomerName,108o.Country,109o.OrderID,110o.OrderDate,111o.ProductID,112o.Quantity113FROM orders o114LEFT JOIN customers c ON o.CustomerID = c.CustomerID115WHERE c.CustomerID IS NULL;116117118

ResultsChart

Columns

CustomerIDCustomerNameCountryInvoiceOrderIDOrderDateProductIDQuantity

1	1004	Customer_1004	India	11	2024-11-01	2049	1
2	1166	Customer_1166	USA	17	2024-04-29	2047	5
3	1233	Customer_1233	USA	20	2023-06-06	2117	3
4	1166	Customer_1166	USA	46	2023-10-20	2017	7
5	1246	Customer_1246	Australia	142	2024-11-28	2034	1
6	1511	Customer_1511	Germany	7	2023-07-29	2069	8
7	1002	Customer_1002	India	42	2024-10-04	2116	6
8	1236	Customer_1236	Australia	49	2023-12-12	2076	4
9	1110	Customer_1110	Australia	81	2024-06-30	2119	9

8. FULL OUTER JOIN: All Products and Orders Syntax

SQL Query:

```

-- 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

SELECT
  p.ProductID,
  p.ProductName,
  p.Price,
  o.OrderID,
  o.OrderDate,
  o.CustomerID,
  o.Quantity
FROM products p
LEFT JOIN orders o ON p.ProductID = o.ProductID
UNION
SELECT
  p.ProductID,
  p.ProductName,
  p.Price,
  o.OrderID,
  o.OrderDate,
  o.CustomerID,
  o.Quantity
FROM orders o
LEFT JOIN products p ON o.ProductID = p.ProductID
WHERE p.ProductID IS NULL;

```

Results

2023-10-18 8:42pm2023-10-18 8:20pm

DatabaseWorkbooks

Search objects

DRIGHTLEARN_DATA_ANAL...

INFORMATION_SCHEMA

PUBLIC

RETAIL_SALES

PRACTICAL_2_SOLUTION

INFORMATION_SCHEMA

JOB

Tables

CUSTOMERS

ORDERS

PRODUCTS

Carto Search Services

PUBLIC

SHOWPLATE

SHOWPLATE_LEARNING_JOB

SHOWPLATE_SAMPLE_DATA

PRCTICAL_2_SOLUTION_JOBSettings

Open in WorkspaceQuick Edit

UNION

SELECT

p.ProductID,

p.ProductName,

p.Price,

o.OrderID,

o.CustomerID,

o.Quantity

FROM orders o

LEFT JOIN products p ON o.ProductID = p.ProductID

WHERE p.ProductID IS NULL;

ResultsChart

SearchFilterColumns

#	ID	PRODUCTID	PRODUCTNAME	PRICE	ORDERID	ORDERDATE	CUSTOMERID	QUANTITY
1	2078	Product_2078	333	8	2023-12-06	1022	7	
2	2092	Product_2092	000	19	2024-04-15	1219	3	
3	2117	Product_2117	1437	20	2023-06-08	1033	1	
4	2001	Product_2001	9999	30	2024-11-10	1101	3	
5	2030	Product_2030	1148	39	2023-01-22	1056	8	
6	2108	Product_2108	1223	42	2024-04-28	1052	1	
7	2057	Product_2057	787	48	2023-01-01	1039	1	
8	2046	Product_2046	1276	80	2024-11-27	1160	10	
9	2166	Product_2166	400	91	2023-10-17	1080	8	
10	2034	Product_2034	79	117	2024-09-24	1067	8	
11	2046	Product_2046	478	81	2024-02-22	1086	10	