

Thatiso Sethema

## Practical 2 – SQL JOINS

### Question 1

The screenshot shows a Snowflake interface with a SQL query and its results. The query is an INNER JOIN between the orders and customers tables, filtered by customer ID.

```
SELECT orderid,
       orderdate,
       customername,
       productname,
       quantity
FROM practical2.joins.orders AS A
INNER JOIN practical2.joins.customers AS B
ON A.customerid = B.customerid
INNER JOIN practical2.joins.products AS C
ON A.productid = C.productid;
```

The results table shows 4,000 rows. The columns are: 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	2023-07-29	Customer_1011	Product_2099	8

### Question 2

The screenshot shows a Snowflake interface with a SQL query and its results. The query is an INNER JOIN between the orders and customers tables, filtered by customer ID, and includes the country column from the customers table.

```
SELECT A.customerid,
       A.customername,
       A.country,
       B.orderid,
       B.orderdate
FROM practical2.joins.customers AS A
INNER JOIN practical2.joins.orders AS B
ON A.customerid = B.customerid;
```

The results table shows 4,000 rows. The columns are: 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

## Question 3

The screenshot shows the Snowflake SQL editor with a query that performs a LEFT JOIN between the 'customers' table in the 'practical2.joins' schema and the 'orders' table in the same schema. The query selects customer details and order quantities.

```
31 -----
32 -- 3. LEFT JOIN: ALL Customers and Their Orders
33 -- Question:
34 -- List all customers and any orders they might have placed. Include customers who have
35 -- not placed any orders.
36 -- Expected Output Columns:
37 -- CustomerID, CustomerName, Country, OrderID, OrderDate, ProductID, Quantity
38 SELECT A.customerid,
39        A.customername,
40        A.country,
41        B.orderid,
42        B.orderdate,
43        B.productid,
44        B.quantity
45 FROM practical2.joins.customers AS A
46 LEFT JOIN practical2.joins.orders AS B
47 ON A.customerid = B.customerid;
```

The results table shows 4,000 rows. The data is as follows:

#	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

## Question 4

The screenshot shows the Snowflake SQL editor with a query that performs a LEFT JOIN between the 'products' table in the 'practical2.joins' schema and the 'orders' table in the same schema. The query uses COUNT to aggregate the total orders for each product.

```
47 -----
48 -- 4. LEFT JOIN: Product Order Count
49 -- Question:
50 -- List all products and how many times each was ordered (if any).
51 -- Expected Output Columns:
52 -- ProductID, ProductName, TotalOrders
53 -- (TotalOrders is the count of how many times the product appears in orders)
54 SELECT A.productid,
55        A.productname,
56        COUNT(B.orderid) AS totalorders
57 FROM practical2.joins.products AS A
58 LEFT JOIN practical2.joins.orders AS B
59 ON A.productid = B.productid
60 GROUP BY A.productid, A.productname;
```

The results table shows 200 rows. The data is as follows:

#	PRODUCTID	PRODUCTNAME	TOTALORDERS
1	2171	Product_2171	15
2	2177	Product_2177	20
3	2073	Product_2073	19
4	2089	Product_2089	20
5	2054	Product_2054	24
6	2019	Product_2019	17
7	2190	Product_2190	20

## Question 5

The screenshot shows the Snowflake web interface. The SQL editor contains the following query:

```
-- 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
SELECT A.orderid,
       A.orderdate,
       B.productid,
       B.productname,
       B.price,
       A.quantity
FROM practical2.joins.orders AS A
RIGHT JOIN practical2.joins.products AS B
ON A.productid = B.productid;
```

The results table shows 4,000 rows in 201ms. The data is as follows:

#	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		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

## Question 6

The screenshot shows the Snowflake web interface. The SQL editor contains the following query:

```
-- 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
SELECT A.customerid,
       A.customername,
       A.country,
       B.orderid,
       B.orderdate,
       C.productid,
       B.quantity
FROM practical2.joins.customers AS A
RIGHT JOIN practical2.joins.orders AS B
ON A.customerid = B.customerid
RIGHT JOIN practical2.joins.products AS C
ON B.productid = C.productid;
```

The results table shows 4,000 rows in 35ms. The data is as follows:

#	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

## Question 7

The screenshot shows the Snowflake SQL interface. The query editor contains the following SQL code:

```
-- 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

SELECT A.customerid,
       A.customername,
       A.country,
       B.orderid,
       B.orderdate,
       C.productid,
       B.quantity
FROM practical2.joins.customers AS A
FULL OUTER JOIN practical2.joins.orders AS B
ON A.customerid = B.customerid
FULL OUTER JOIN practical2.joins.products AS C
ON B.productid = C.productid;
```

The results table shows 4,000 rows in 28ms. The data is as follows:

#	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

## Question 8

The screenshot shows the Snowflake SQL interface. The query editor contains the following SQL code:

```
-- 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 A.productid,
       A.productname,
       A.price,
       B.orderid,
       B.orderdate,
       C.customerid,
       B.quantity
FROM practical2.joins.products AS A
FULL OUTER JOIN practical2.joins.orders AS B
ON A.productid = B.productid
FULL OUTER JOIN practical2.joins.customers AS C
ON B.customerid = C.customerid;
```

The results table shows 4,000 rows in 816ms. The data is as follows:

#	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
-	----	-----	----	-	-----	----	-