

## Sindiswa Jachin Shongwe

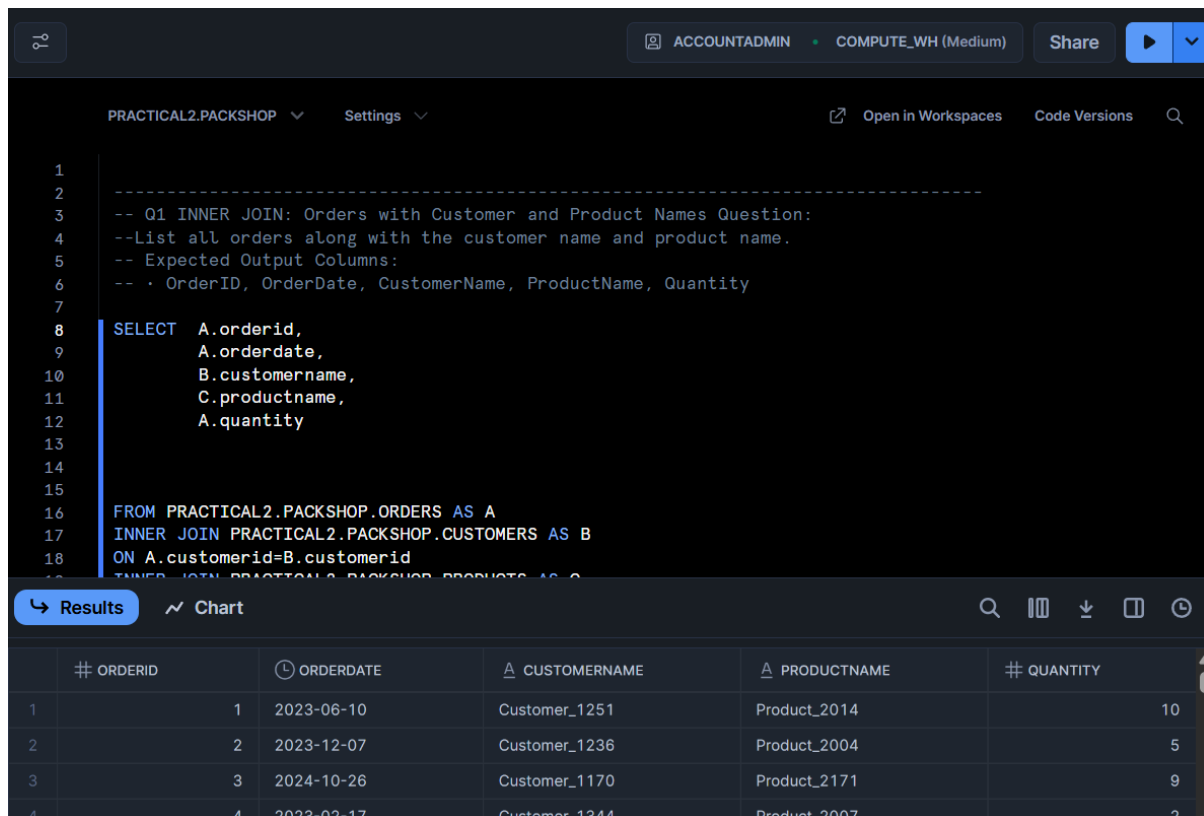
### Practical 2: BrightLight Data Analytics Coding Practical JOINS

-- Q1 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 IDE interface with a dark theme. At the top, there's a header bar with 'ACCOUNTADMIN', 'COMPUTE\_WH (Medium)', and a 'Share' button. Below the header, the main editor area displays a SQL query. The query is as follows:

```
1  
2  
3  -----  
4  -- Q1 INNER JOIN: Orders with Customer and Product Names Question:  
5  --List all orders along with the customer name and product name.  
6  -- Expected Output Columns:  
7  -- • OrderID, OrderDate, CustomerName, ProductName, Quantity  
8  
9  SELECT  A.orderid,  
10         A.orderdate,  
11         B.customername,  
12         C.productname,  
13         A.quantity  
14  
15  
16  FROM PRACTICAL2.PACKSHOP.ORDERS AS A  
17  INNER JOIN PRACTICAL2.PACKSHOP.CUSTOMERS AS B  
18  ON A.customerid=B.customerid  
19  INNER JOIN PRACTICAL2.PACKSHOP.PRODUCTS AS C
```

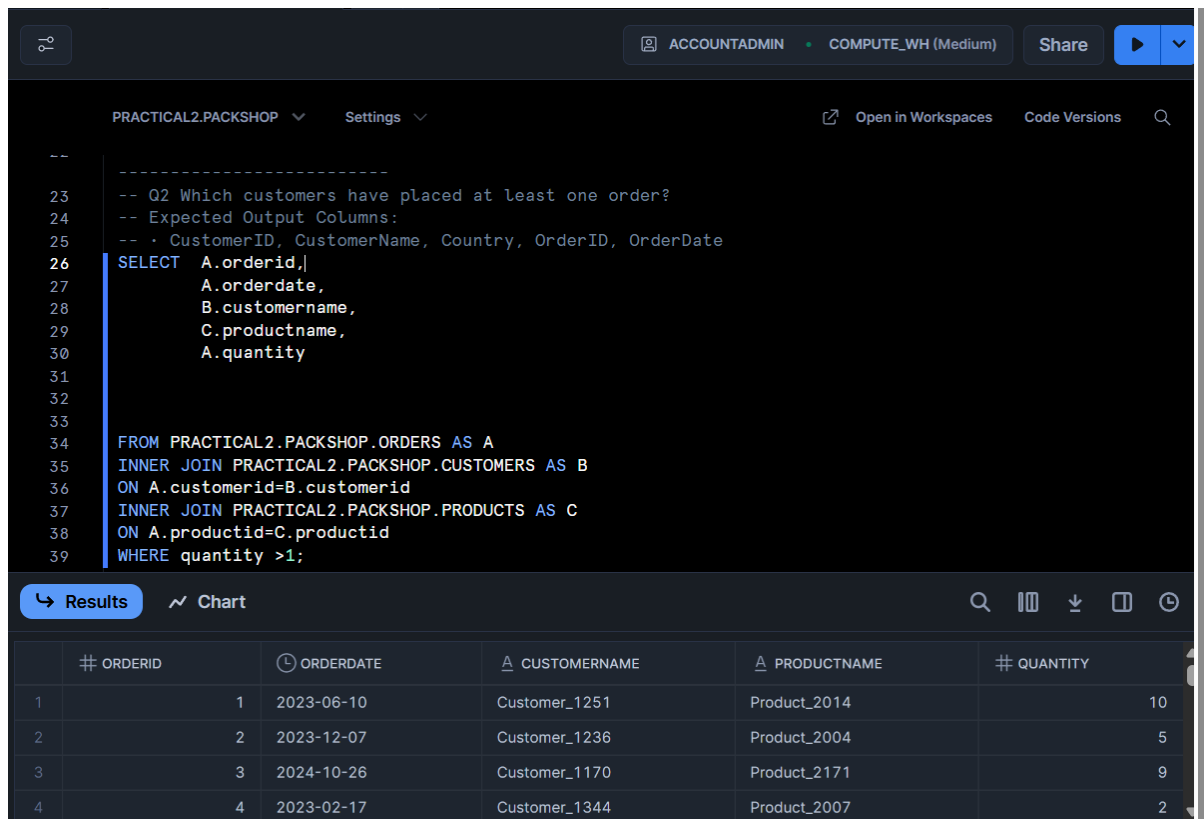
Below the editor, there's a 'Results' tab selected, showing a table with 5 columns: '# ORDERID', 'ORDERDATE', 'CUSTOMERNAME', 'PRODUCTNAME', and 'QUANTITY'. The table contains 4 rows of data:

#	ORDERID	ORDERDATE	CUSTOMERNAME	PRODUCTNAME	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

-- Q2 Which customers have placed at least one order?

-- Expected Output Columns:

-- • CustomerID, CustomerName, Country, OrderID, OrderDate.



The screenshot shows a SQL IDE interface. At the top, there's a header with 'ACCOUNTADMIN' and 'COMPUTE\_WH (Medium)' roles, and a 'Share' button. Below the header, the database 'PRACTICAL2.PACKSHOP' is selected. The main area contains a SQL query with line numbers 23 to 39. The query is a SELECT statement with columns: A.orderid, A.orderdate, B.customername, C.productname, and A.quantity. It uses INNER JOINs for CUSTOMERS and PRODUCTS, and a WHERE clause for quantity > 1. Below the query editor, there's a 'Results' tab showing a table with 4 rows and 6 columns: #, ORDERID, ORDERDATE, CUSTOMERNAME, PRODUCTNAME, and QUANTITY. The results are as follows:

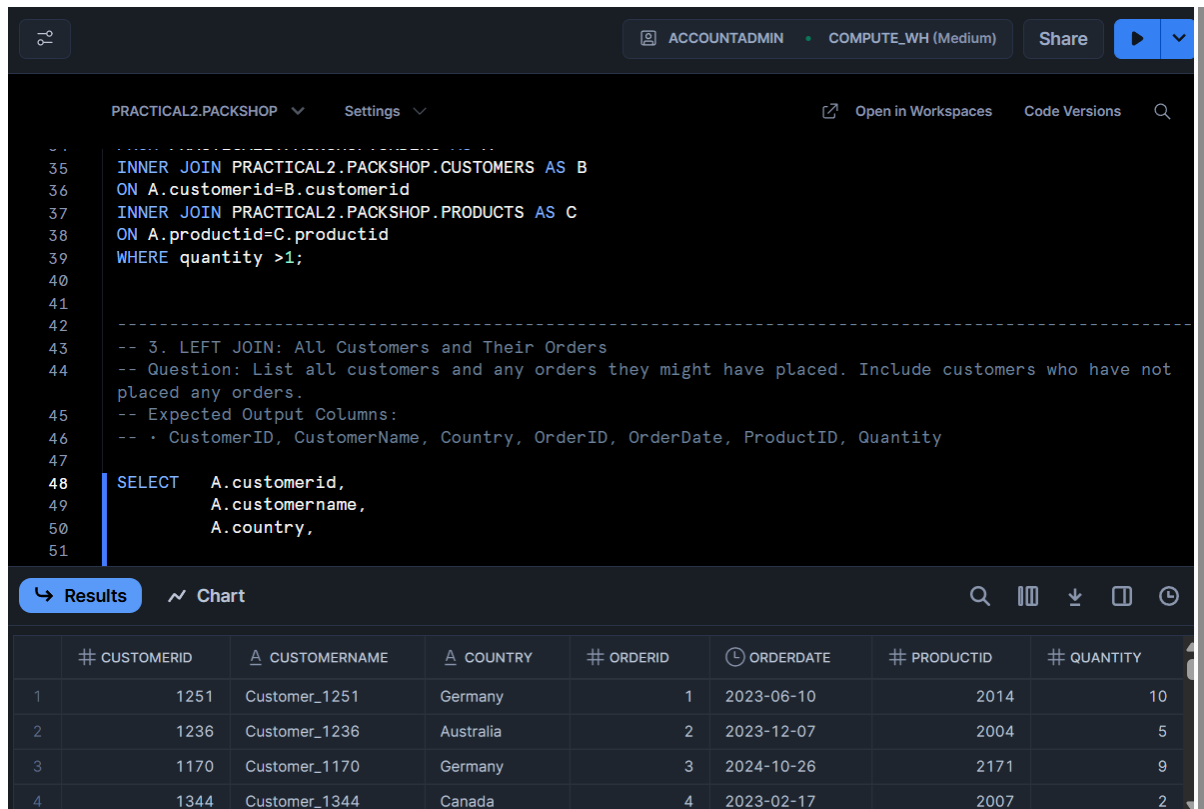
	#	ORDERID	ORDERDATE	CUSTOMERNAME	PRODUCTNAME	QUANTITY
1	1	2023-06-10	Customer_1251	Product_2014	10	
2	2	2023-12-07	Customer_1236	Product_2004	5	
3	3	2024-10-26	Customer_1170	Product_2171	9	
4	4	2023-02-17	Customer_1344	Product_2007	2	

--Q3. 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 dark theme. The top bar includes a user profile icon, the username 'ACCOUNTADMIN', the database name 'COMPUTE\_WH (Medium)', a 'Share' button, and a dropdown menu. Below the top bar, there's a toolbar with 'Settings', 'Open in Workspaces', 'Code Versions', and a search icon. The main editor area contains a SQL query with line numbers 35 to 51. The query includes two INNER JOINs and a WHERE clause. Below the query, there's a comment block starting with '-- 3. LEFT JOIN: All Customers and Their Orders'. At the bottom, there's a 'Results' tab with a table of 4 rows and 8 columns. The table has columns: # CUSTOMERID, A CUSTOMERNAME, A COUNTRY, # ORDERID, ORDERDATE, # PRODUCTID, and # QUANTITY. The data rows show customer information and their orders.

```
35  -----
36  INNER JOIN PRACTICAL2.PACKSHOP.CUSTOMERS AS B
37  ON A.customerid=B.customerid
38  INNER JOIN PRACTICAL2.PACKSHOP.PRODUCTS AS C
39  ON A.productid=C.productid
40  WHERE quantity >1;
41
42  -----
43  -- 3. LEFT JOIN: All Customers and Their Orders
44  -- Question: List all customers and any orders they might have placed. Include customers who have not
45  -- placed any orders.
46  -- Expected Output Columns:
47  -- • CustomerID, CustomerName, Country, OrderID, OrderDate, ProductID, Quantity
48  SELECT  A.customerid,
49          A.customername,
50          A.country,
51
```

	# CUSTOMERID	A CUSTOMERNAME	A 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

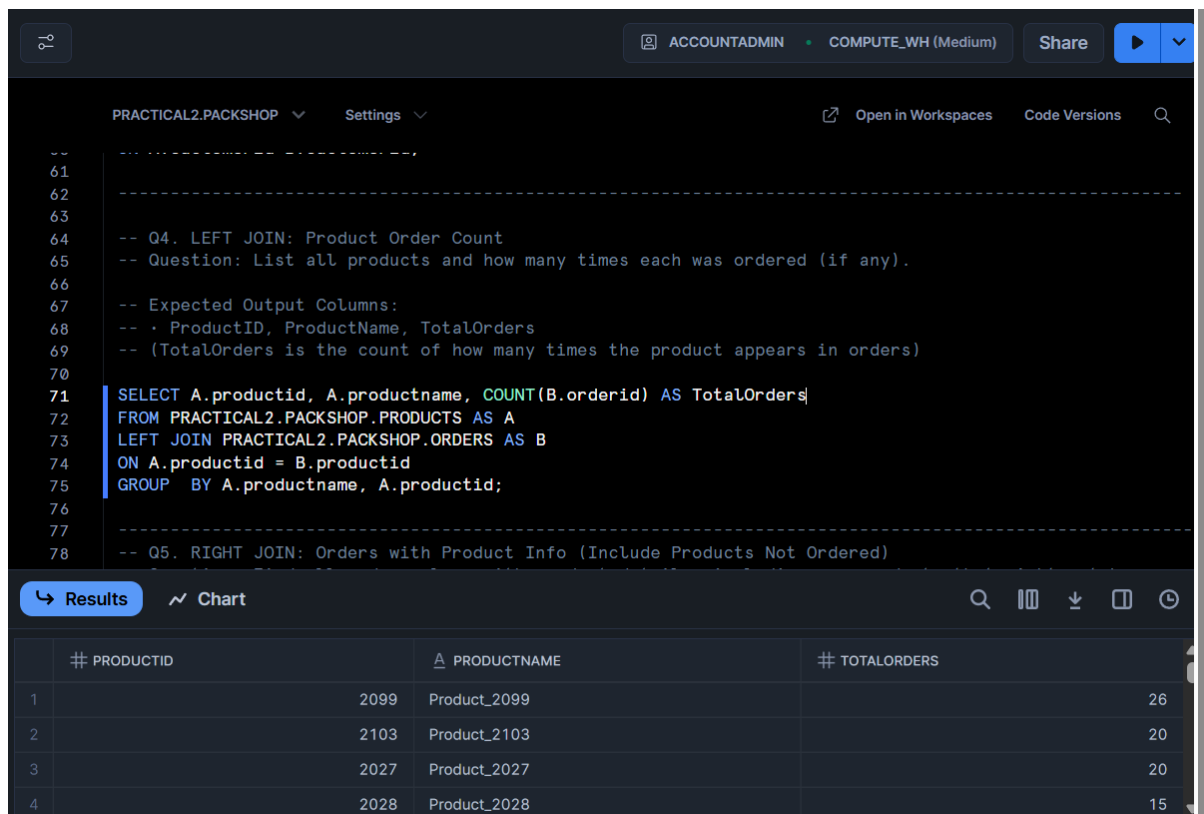
-- Q4. LEFT JOIN: Product Order Count

-- 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).



```
-- Q4. 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)

SELECT A.productid, A.productname, COUNT(B.orderid) AS TotalOrders
FROM PRACTICAL2.PACKSHOP.PRODUCTS AS A
LEFT JOIN PRACTICAL2.PACKSHOP.ORDERS AS B
ON A.productid = B.productid
GROUP BY A.productname, A.productid;
```

	# PRODUCTID	A PRODUCTNAME	# TOTALORDERS
1	2099	Product_2099	26
2	2103	Product_2103	20
3	2027	Product_2027	20
4	2028	Product_2028	15

-- Q5. 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 IDE interface with a dark theme. The top bar includes a tab labeled 'Practical2', a '+' icon, and a dropdown arrow. Below this, a toolbar shows 'ACCOUNTADMIN', 'COMPUTE\_WH (Medium)', a 'Share' button, and a play button. The main editor area displays a SQL query with line numbers 78 to 94. The query is a RIGHT JOIN between PRACTICAL2.PACKSHOP.PRODUCTS (AS A) and PRACTICAL2.PACKSHOP.ORDERS (AS B) on the condition A.productid=B.productid. The expected output columns are listed as OrderID, OrderDate, ProductID, ProductName, Price, and Quantity. Below the editor, there are tabs for 'Results' (selected) and 'Chart'. The 'Results' tab shows a table with 7 columns: # PRODUCTID, A PRODUCTNAME, # PRICE, # ORDERID, ORDERDATE, and # QUANTITY. The table contains 4 rows of data.

```
78 -- Q5. RIGHT JOIN: Orders with Product Info (Include Products Not Ordered)
79 -- Question: Find all orders along with product details, including any products that might not have
80 -- Expected Output Columns: • OrderID, OrderDate, ProductID, ProductName, Price, Quantity
81
82 SELECT A.productid,
83        A.productname,
84        A.price,
85
86        B.orderid,
87        B.orderdate,
88        B.quantity
89
90
91 FROM PRACTICAL2.PACKSHOP.PRODUCTS AS A
92 RIGHT JOIN PRACTICAL2.PACKSHOP.ORDERS AS B
93 ON A.productid=B.productid;
94
```

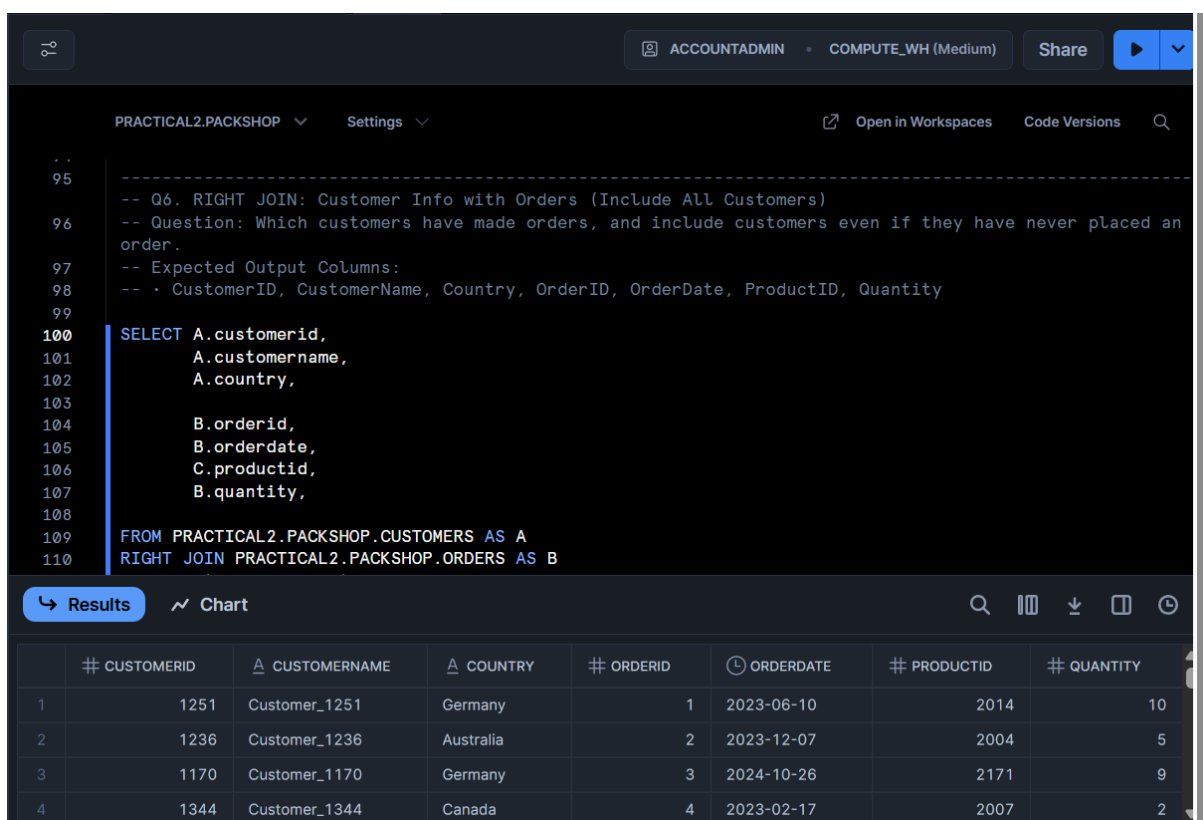
	# PRODUCTID	A PRODUCTNAME	# PRICE	# ORDERID	ORDERDATE	# QUANTITY
1	2014	Product_2014	522	1	2023-06-10	10
2	2004	Product_2004	1996	2	2023-12-07	5
3	2171	Product_2171	76	3	2024-10-26	9
4	2007	Product_2007	156	4	2023-02-17	2

## Q6. 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 IDE interface with a query editor and a results pane. The query editor contains a SQL query for a RIGHT JOIN between CUSTOMERS and ORDERS. The results pane shows a table with 8 columns: #, CUSTOMERID, CUSTOMERNAME, COUNTRY, ORDERID, ORDERDATE, PRODUCTID, and QUANTITY. The results table contains 4 rows of data.

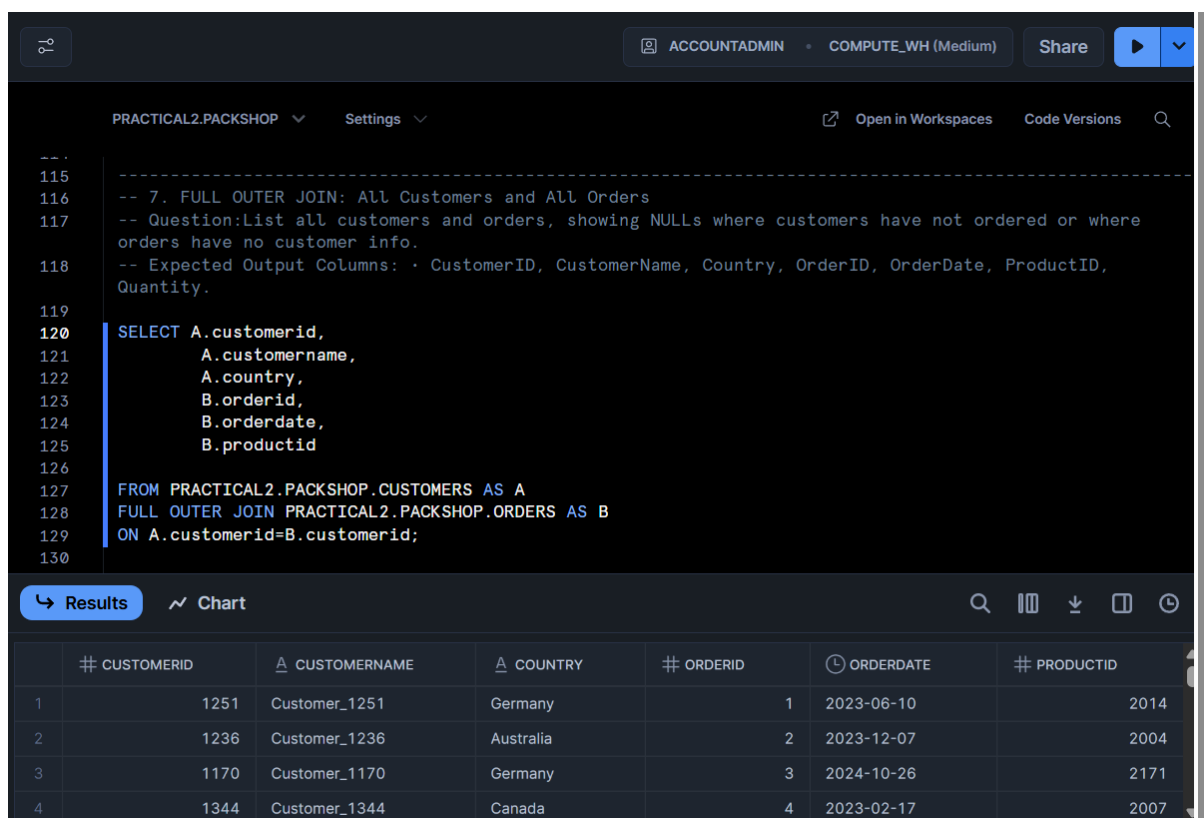
```
95 -----
96 -- Q6. RIGHT JOIN: Customer Info with Orders (Include All Customers)
97 -- Question: Which customers have made orders, and include customers even if they have never placed an
98 -- Expected Output Columns:
99 -- • CustomerID, CustomerName, Country, OrderID, OrderDate, ProductID, Quantity
100 SELECT A.customerid,
101        A.customername,
102        A.country,
103
104        B.orderid,
105        B.orderdate,
106        C.productid,
107        B.quantity,
108
109 FROM PRACTICAL2.PACKSHOP.CUSTOMERS AS A
110 RIGHT JOIN PRACTICAL2.PACKSHOP.ORDERS AS B
```

#	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

-- 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's a header with 'ACCOUNTADMIN', 'COMPUTE\_WH (Medium)', and a 'Share' button. Below the header, there's a toolbar with 'Open in Workspaces', 'Code Versions', and a search icon. The main area displays a SQL query with line numbers 115 to 130. The query is a FULL OUTER JOIN between CUSTOMERS and ORDERS. Below the query, there's a 'Results' tab and a 'Chart' tab. The 'Results' tab is active, showing a table with 7 columns: CUSTOMERID, CUSTOMERNAME, COUNTRY, ORDERID, ORDERDATE, and PRODUCTID. The table contains 4 rows of data.

```
-- .
115 -----
116 -- 7. FULL OUTER JOIN: All Customers and All Orders
117 -- Question: List all customers and orders, showing NULLs where customers have not ordered or where
118 -- Expected Output Columns: • CustomerID, CustomerName, Country, OrderID, OrderDate, ProductID,
119 -- Quantity.
120
121 SELECT A.customerid,
122        A.customername,
123        A.country,
124        B.orderid,
125        B.orderdate,
126        B.productid
127
128 FROM PRACTICAL2.PACKSHOP.CUSTOMERS AS A
129 FULL OUTER JOIN PRACTICAL2.PACKSHOP.ORDERS AS B
130 ON A.customerid=B.customerid;
```

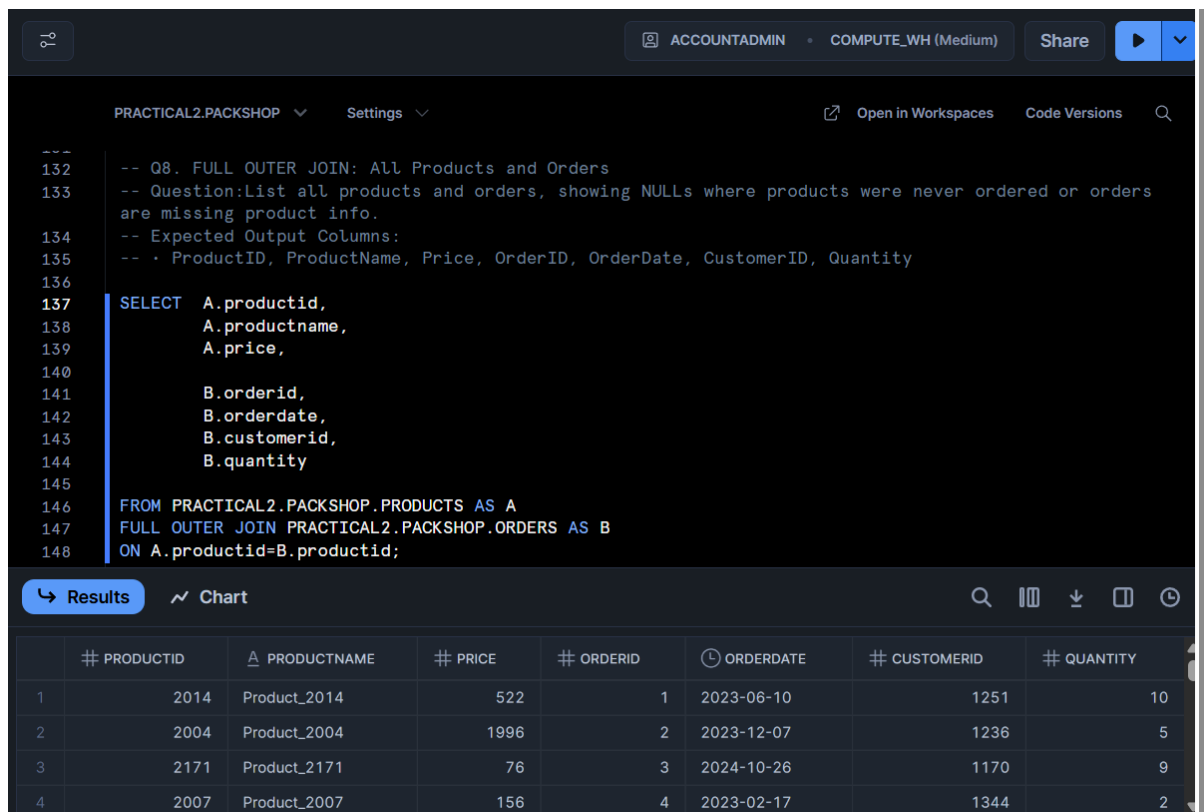
	# CUSTOMERID	A CUSTOMERNAME	A COUNTRY	# ORDERID	ORDERDATE	# PRODUCTID
1	1251	Customer_1251	Germany	1	2023-06-10	2014
2	1236	Customer_1236	Australia	2	2023-12-07	2004
3	1170	Customer_1170	Germany	3	2024-10-26	2171
4	1344	Customer_1344	Canada	4	2023-02-17	2007

-- Q8. 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 'ACCOUNTADMIN' and 'COMPUTE\_WH (Medium)' roles, a 'Share' button, and a dropdown menu. Below the header, the main area displays a SQL query. The query is a FULL OUTER JOIN between 'PRACTICAL2.PACKSHOP.PRODUCTS' (aliased as A) and 'PRACTICAL2.PACKSHOP.ORDERS' (aliased as B). The query selects columns: A.productid, A.productname, A.price, B.orderid, B.orderdate, B.customerid, and B.quantity. The results are shown in a table below the query.

```
-- Q8. 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,
       B.customerid,
       B.quantity
FROM PRACTICAL2.PACKSHOP.PRODUCTS AS A
FULL OUTER JOIN PRACTICAL2.PACKSHOP.ORDERS AS B
ON A.productid=B.productid;
```

	# PRODUCTID	A 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