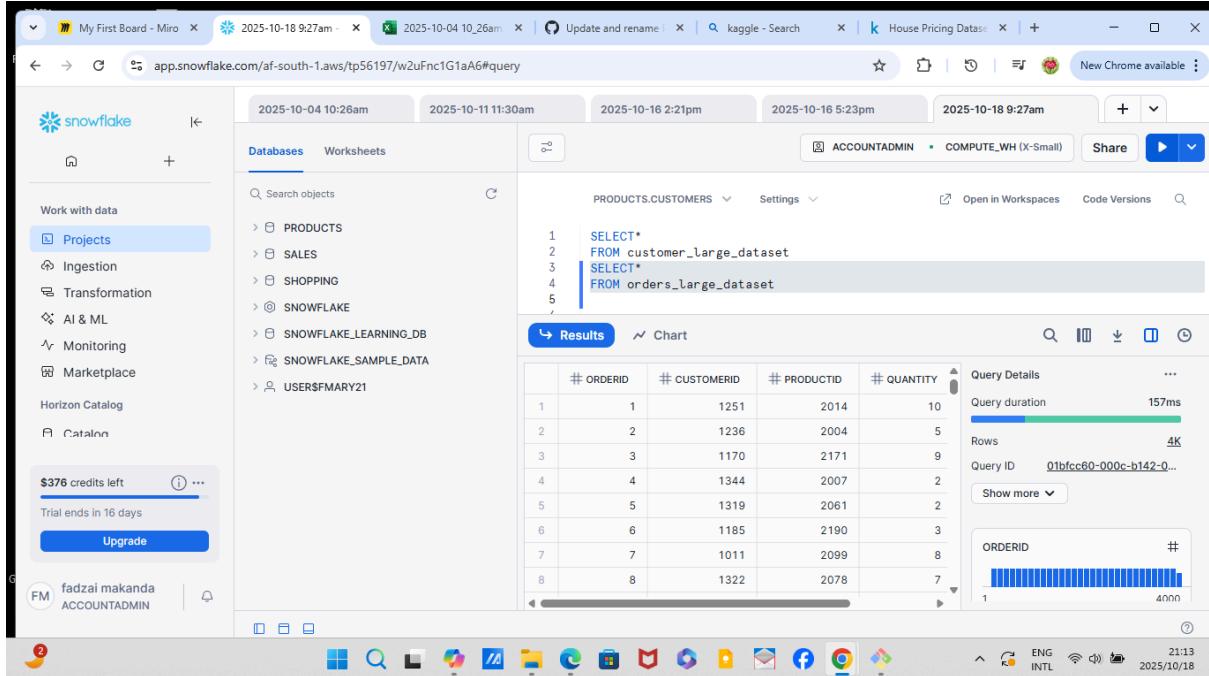


PRACTICAL 2

FADZAI MAKANDA



The screenshot shows the Snowflake web interface. On the left, there's a sidebar with 'Work with data' sections like Projects, Ingestion, Transformation, AI & ML, Monitoring, Marketplace, Horizon Catalog, and Catalog. A message says '\$376 credits left' and 'Trial ends in 16 days' with a 'Upgrade' button. The main area shows a timeline at the top with dates from 2025-10-04 to 2025-10-18. Below the timeline, the 'Databases' tab is selected, showing objects like PRODUCTS, SALES, SHOPPING, SNOWFLAKE, SNOWFLAKE_LEARNING_DB, SNOWFLAKE_SAMPLE_DATA, and USER\$FMARY21. A search bar is present. To the right, a query editor window displays the following SQL code:

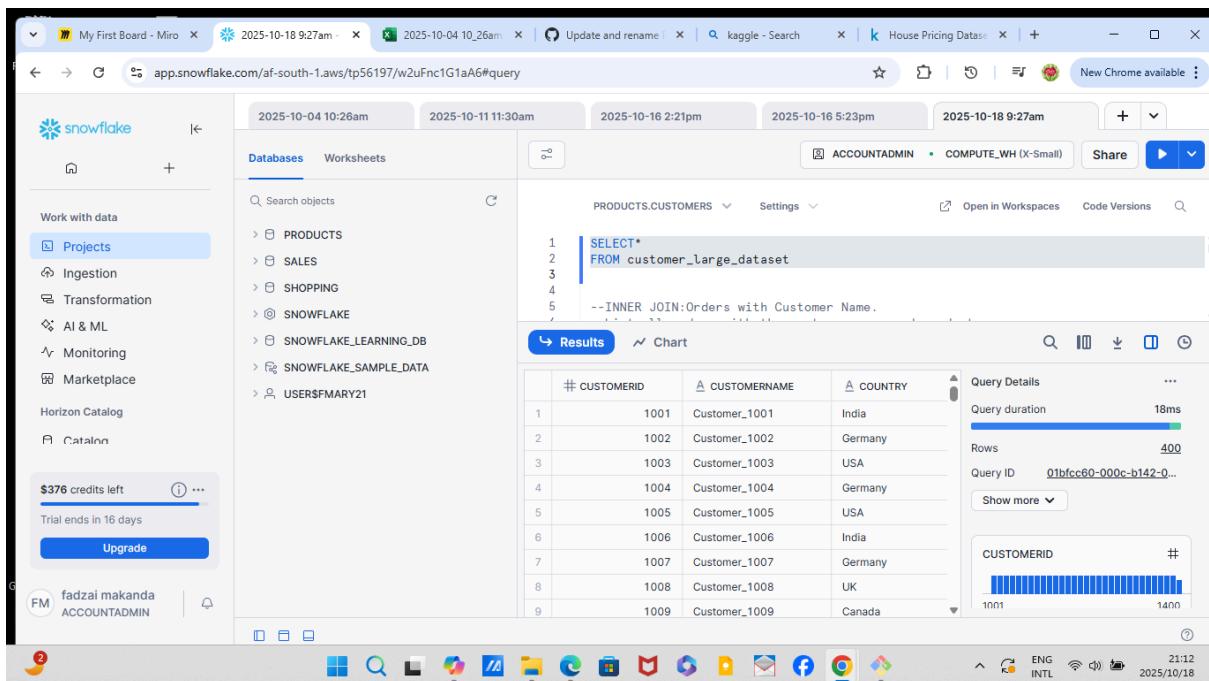
```
1 SELECT*
  2 FROM customer_large_dataset
  3 SELECT*
  4 FROM orders_large_dataset
```

The results section shows a table with columns ORDERID, CUSTOMERID, PRODUCTID, and QUANTITY. The data is as follows:

ORDERID	CUSTOMERID	PRODUCTID	QUANTITY
1	1	1251	2014
2	2	1236	2004
3	3	1170	2171
4	4	1344	2007
5	5	1319	2061
6	6	1185	2190
7	7	1011	2099
8	8	1322	2078

On the right, 'Query Details' show a duration of 157ms and 4K rows. A histogram for ORDERID is shown with values ranging from 1 to 4000.

ORDERS TABLE.



This screenshot is similar to the previous one but shows a different query. The timeline at the top has dates from 2025-10-04 to 2025-10-18. The 'Databases' tab is selected, showing the same list of datasets as before. The query editor window contains the following SQL code:

```
1 SELECT*
  2 FROM customer_large_dataset
  3
  4 --INNER JOIN:Orders with Customer Name.
```

The results section shows a table with columns CUSTOMERID, CUSTOMERNAME, and COUNTRY. The data is as follows:

CUSTOMERID	CUSTOMERNAME	COUNTRY
1001	Customer_1001	India
1002	Customer_1002	Germany
1003	Customer_1003	USA
1004	Customer_1004	Germany
1005	Customer_1005	USA
1006	Customer_1006	India
1007	Customer_1007	Germany
1008	Customer_1008	UK
1009	Customer_1009	Canada

On the right, 'Query Details' show a duration of 18ms and 400 rows. A histogram for CUSTOMERID is shown with values ranging from 1001 to 1400.

CUSTOMERS TABLE

The screenshot shows the Snowflake SQL interface. On the left, the sidebar includes 'Work with data' sections like Projects, Ingestion, Transformation, AI & ML, Monitoring, Marketplace, and Catalogs. A message indicates '\$376 credits left' and 'Trial ends in 16 days'. The main area displays a query:

```

1   FROM customer_large_dataset
2   SELECT*
3   FROM orders_large_dataset
4   SELECT*
5   FROM product_large_dataset

```

The results show a table with columns PRODUCTID, PRODUCTNAME, and PRICE, containing 200 rows. The first few rows are:

PRODUCTID	PRODUCTNAME	PRICE
1	Product_2001	833
2	Product_2002	1558
3	Product_2003	1398
4	Product_2004	1996
5	Product_2005	1146
6	Product_2006	694
7	Product_2007	156
8	Product_2008	811
9	Product_2009	342

The 'Query Details' panel shows a duration of 49ms and 200 rows.

ORDERS TABLE

The screenshot shows the Snowflake SQL interface. The sidebar is identical to the previous one. The main area displays a query:

```

14  SELECT A.orderid,
15    A.orderdate,
16    B.customername,
17    C.productid,
18

```

The results show a table with columns ORDERID, ORDERDATE, CUSTOMERNAME, and PRO, containing 320 rows. The first few rows are:

ORDERID	ORDERDATE	CUSTOMERNAME	PRO
1	2024-05-09	Customer_1001	
2	2024-05-26	Customer_1001	
3	2024-06-27	Customer_1001	
4	2024-10-07	Customer_1001	
5	2024-04-05	Customer_1001	
6	2024-07-18	Customer_1001	
7	2024-07-09	Customer_1001	
8	2023-05-18	Customer_1001	

The 'Query Details' panel shows a duration of 44ms and 320 rows.

QUESTION 1

```

15  SELECT A.orderid,
16    A.orderdate,
17    B.customername,
18    C.productid,
19    A.quantity
20   FROM orders_large_dataset AS A

```

#	CUSTOMERID	CUSTOMERNAME	COUNTRY	#	ORDERID	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
7	1011	Customer_1011	Germany		7	2023-07-29
8	1322	Customer_1322	Australia		8	2023-12-06
9	1224	Customer_1224	Australia		9	2025-01-25

Query Details
Query duration 19ms
Rows 4K
Query ID 01bfd045-000c-b142-0...
Show more

CUSTOMERID #
1001 1400

QUESTION 2

```

-- FROM CUSTOMER_LARGE_DATASET AS A
64  LEFT JOIN ORDERS_LARGE_DATASET AS B
65  ON A.CUSTOMERID = B.CUSTOMERID
66  LEFT JOIN PRODUCT_LARGE_DATASET AS C
67  ON C.PRODUCTID=C.PRODUCTID
68

```

#	CUSTOMERID	CUSTOMERNAME	COUNTRY	#	ORDERID	ORDERDATE	#	PRODUCTID	#	QUANTITY
1	1251	Customer_1251	Germany		1	2023-06-10		2001		10
2	1251	Customer_1251	Germany		1	2023-06-10		2002		10
3	1251	Customer_1251	Germany		1	2023-06-10		2003		10
4	1251	Customer_1251	Germany		1	2023-06-10		2004		10
5	1251	Customer_1251	Germany		1	2023-06-10		2005		10
6	1251	Customer_1251	Germany		1	2023-06-10		2006		10
7	1251	Customer_1251	Germany		1	2023-06-10		2007		10
8	1251	Customer_1251	Germany		1	2023-06-10		2008		10

Query Details
Query duration 805ms
Rows 800K
Query ID 01bfd062-000c-b142-0...
Show more

CUSTOMERID #
1001 1400

QUESTION 3

The screenshot shows a web browser window with multiple tabs open. The active tab is a Snowflake query results page titled "app.snowflake.com/af-south-1.aws/tp56197/w2uFnc1G1aA6#query". The query is a LEFT JOIN operation between the "PRODUCTS.CUSTOMERS" and "ORDERS_LARGE_DATASET" tables. The results table displays 200 rows, each containing a Product ID, Product Name, and Total Orders. The "Query Details" panel on the right shows a duration of 474ms and 200 rows.

#	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
8	2119	Product_2119	22
9	2182	Product_2182	17

QUESTION 4 - FADZAI MAKANDA

The screenshot shows a web browser window with multiple tabs open. The active tab is a Snowflake query results page titled "app.snowflake.com/af-south-1.aws/tp56197/w2uFnc1G1aA6#query". The query is identical to the one in the previous screenshot, performing a LEFT JOIN between "PRODUCTS.CUSTOMERS" and "ORDERS_LARGE_DATASET". The results table displays 200 rows, and the "Query Details" panel on the right shows a duration of 474ms and 200 rows.

#	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
8	2119	Product_2119	22
9	2182	Product_2182	17

QUESTION 4 - FADZAI MAKANDA

```

100      A.PRICE
101      FROM PRODUCT_LARGE_DATASET AS A
102      RIGHT JOIN ORDERS_LARGE_DATASET AS B
103      ON A.PRODUCTID = B.PRODUCTID;
104

```

# PRODUCTID	PRODUCTNAME	ORDERID	ORDERDATE	QUANTITY	PRICE
1	Product_2014	1	2023-06-10	10	522
2	Product_2004	2	2023-12-07	5	1996
3	Product_2171	3	2024-10-26	9	76
4	Product_2007	4	2023-02-17	2	156
5	Product_2061	5	2024-11-06	2	1595
6	Product_2190	6	2024-11-23	3	1755
7	Product_2099	7	2023-07-29	8	1674
8	Product_2078	8	2023-12-06	7	333
9	Product_2043	9	2025-01-25	7	1947

Query Details
Query duration: 615ms
Rows: 4K
Query ID: 01bfd09a-000c-b142-0...
Show more

PRODUCTID

QUESTION 5 - FADZAI MAKANDA

```

111
112      SELECT
113          A.ORDERID,
114          A.ORDERDATE,
115          A.PRODUCTID,
116          R.CUSTOMERID

```

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

Query Details
Query duration: 586ms
Rows: 4K
Query ID: 01bfd0b8-000c-b142-0...
Show more

CUSTOMERID

QUESTION 6- FADZAI MAKANDA

```

137     B.ORDERDATE,
138     B.PRODUCTID,
139     B.QUANTITY
140   FROM CUSTOMER_LARGE_DATASET AS A
141   FULL OUTER JOIN ORDERS_LARGE_DATASET AS B
142     ON A.CUSTOMERID = B.CUSTOMERID

```

#	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 538ms
Rows 4K
Query ID 01bfd0cb-000c-b142-0...
Show more

CUSTOMERID #
1001 1400

QUESTION 7 - FADZAI MAKANDA

```

146 ---8. FULL OUTER JOIN: All Products and Orders
147 ---Question:
148 ---List all products and orders, showing NULLs where products were never ordered or orders are missing product info
149 ---Expected Output Columns:
150 --- ProductID, ProductName, Price, OrderID, OrderDate, CustomerID, Quantity

```

#	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		1947		9	2025-01-25		1224		7

Query Details
Query duration 227ms
Rows 4K
Query ID 01bfd0d05-000c-b142-0...
Show more

PRODUCTID #
2001 2200

QUESTION 8 - FADZAI MAKANDA

```

SELECT*
FROM customer_large_dataset;

```

```

SELECT*
FROM orders_large_dataset;

```

```

SELECT*
FROM product_large_dataset;

```

--INNER JOIN:Orders with Customer Name.
--List all orders with the customer name and product name.
--Expected output columns: OrderID,OrderDate ,CustomerName, ProductName, Quantit

```
SELECT A.orderid,  
       A.orderdate,  
       B.customername,  
       C.productid,  
       A.quantity  
FROM orders_large_dataset AS A  
INNER JOIN customer_large_dataset AS B ON A.customerid = B.customerid  
INNER JOIN product_large_dataset AS C ON C.productid = C.productid
```

--INNER JOIN: customers who placed Orders
--Which customer have placed at least one order.
--Expected output columns : customer_id,Customer_name,Country,Order_id,Order-date

```
SELECT  
       A.CUSTOMERID,  
       A.CUSTOMERNAME,  
       A.COUNTRY,  
       B.ORDERID,  
       B.ORDERDATE  
FROM CUSTOMER_LARGE_DATASET A  
INNER JOIN ORDERS_LARGE_DATASET B  
ON A.CUSTOMERID = B.CUSTOMERID;
```

```
SELECT  
       A.customerid,  
       A.customername,  
       A.country,  
       B.orderid,  
       B.orderdate  
FROM customer_large_dataset A  
INNER JOIN orders_large_dataset B  
ON A.customerid = B.customerid  
WHERE B.quantity = 1;
```

-- 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

```
SELECT A.CUSTOMERID,
       A.CUSTOMERNAME,
       A.COUNTRY,
       B.ORDERID,
       B.ORDERDATE,
       C.PRODUCTID,
       B.QUANTITY
  FROM CUSTOMER_LARGE_DATASET AS A
 LEFT JOIN ORDERS_LARGE_DATASET AS B ON A.CUSTOMERID = B.CUSTOMERID
 LEFT JOIN PRODUCT_LARGE_DATASET AS C ON C.PRODUCTID = B.PRODUCTID;
```

--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)

```
SELECT
       A.PRODUCTID,
       A.PRODUCTNAME,
       COUNT(B.ORDERID) AS TOTALORDERS
  FROM PRODUCT_LARGE_DATASET AS A
 LEFT JOIN ORDERS_LARGE_DATASET AS B
    ON A.PRODUCTID = B.PRODUCTID
 GROUP BY
       A.PRODUCTID,
       A.PRODUCTNAME;
```

-- 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.PRODUCTID,
       A.PRODUCTNAME,
       B.ORDERID,
       B.ORDERDATE,
       B.QUANTITY,
```

```
A.PRICE  
FROM PRODUCT_LARGE_DATASET AS A  
RIGHT JOIN ORDERS_LARGE_DATASET AS B  
ON A.PRODUCTID = B.PRODUCTID;
```

--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  
B.CUSTOMERID,  
B.CUSTOMERNAME,  
B.COUNTRY,  
A.ORDERID,  
A.ORDERDATE,  
A.PRODUCTID,  
A.QUANTITY  
FROM ORDERS_LARGE_DATASET AS A  
RIGHT JOIN CUSTOMER_LARGE_DATASET AS B  
ON A.CUSTOMERID = B.CUSTOMERID;
```

--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,  
B.PRODUCTID,  
B.QUANTITY  
FROM CUSTOMER_LARGE_DATASET AS A  
FULL OUTER JOIN ORDERS_LARGE_DATASET AS B  
ON A.CUSTOMERID = B.CUSTOMERID
```

---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,  
       B.CUSTOMERID,  
       B.QUANTITY  
  FROM PRODUCT_LARGE_DATASET AS A  
FULL OUTER JOIN ORDERS_LARGE_DATASET AS B  
    ON A.PRODUCTID = B.PRODUCTID;
```
