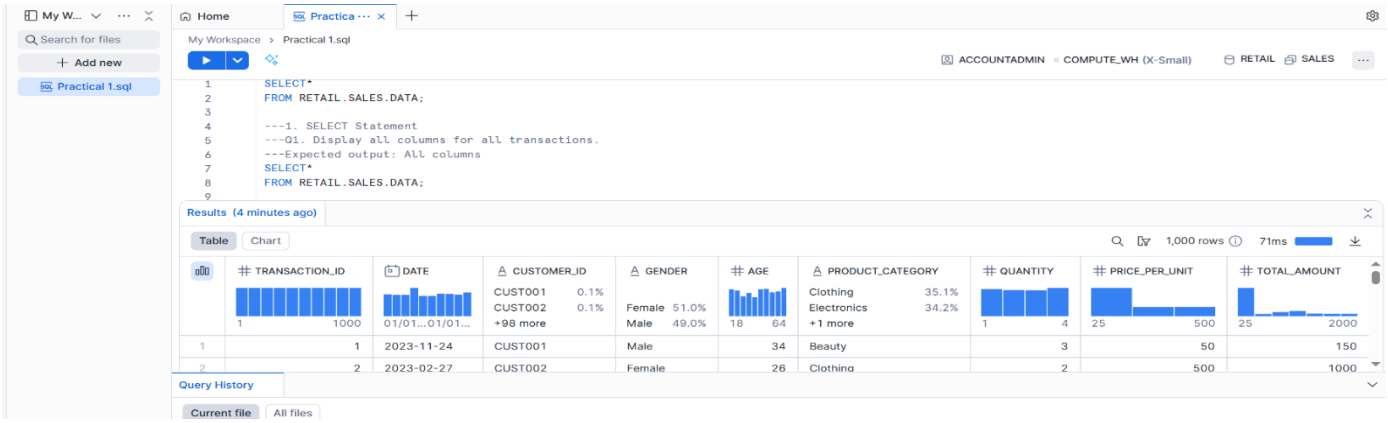
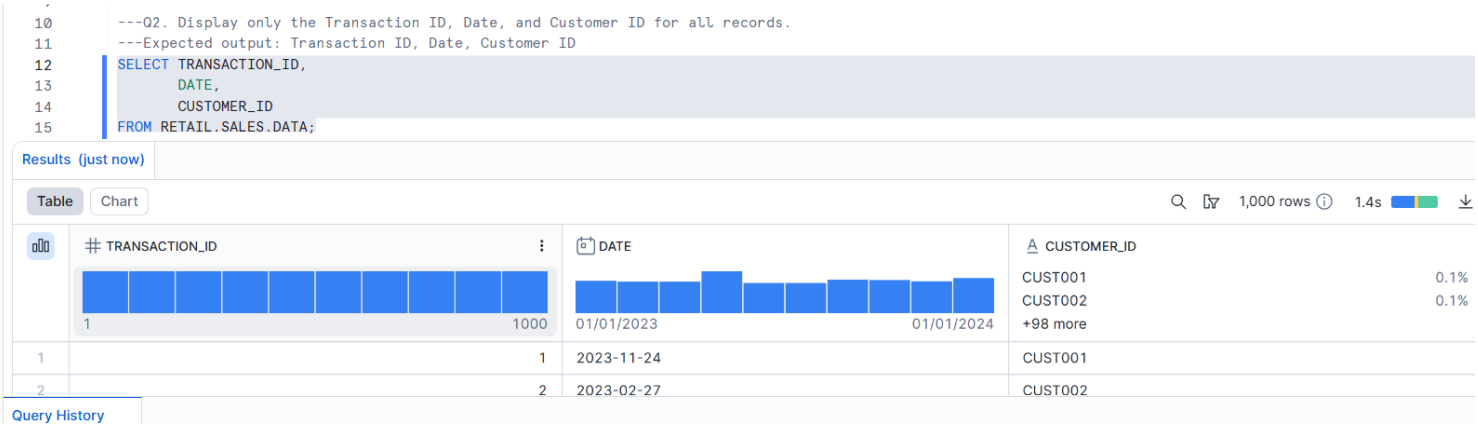
ODION OKONOFUA PRACTICAL 1

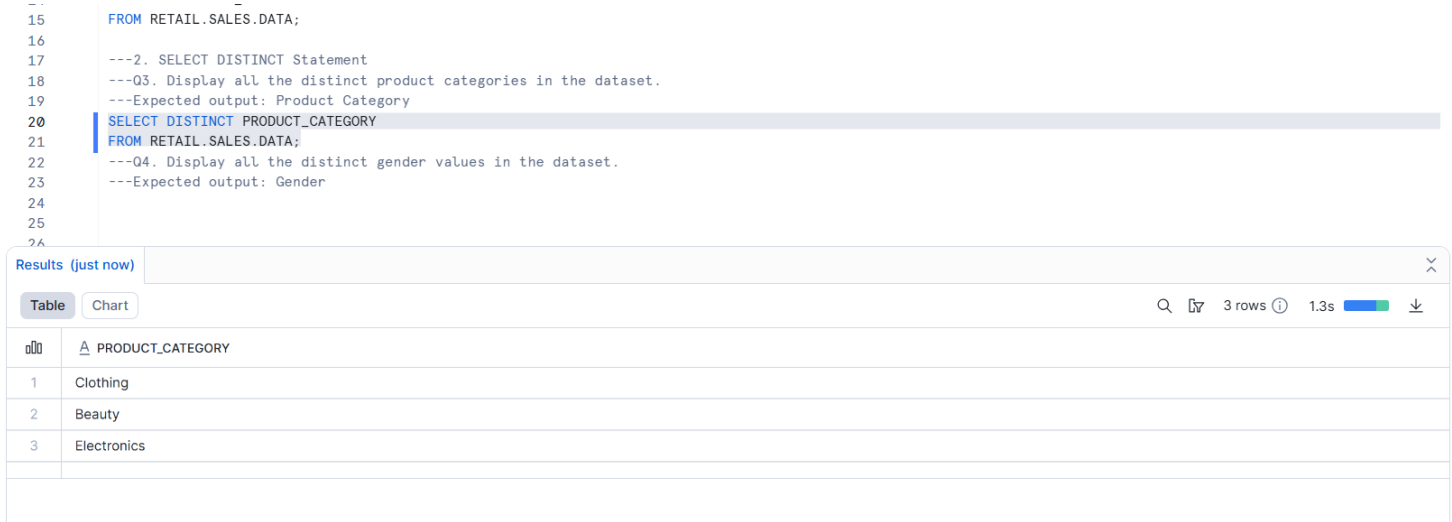
QUESTION 1



QUESTION 2



QUESTION 3



ODION OKONOFUA PRACTICAL 1

QUESTION 4

```
22
23  ---Q4. Display all the distinct gender values in the dataset.
24  ---Expected output: Gender
25  SELECT DISTINCT GENDER
26  FROM RETAIL.SALES.DATA;
```

Results (just now)

Table | Chart — 2 rows — 1.3s

	GENDER
1	Male
2	Female

QUESTION 5

```
31  SELECT*
32  FROM RETAIL.SALES.DATA
33  WHERE AGE>40;
```

esults (just now)

Table | Chart — 534 rows — 1.4s

	TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT	TOTAL_AMOUNT
	3 … 1000	01/01… 01/01…	CUST003 0.2% CUST006 0.2% +98 more	Female 51.3% Male 48.7%	41 … 64	Clothing 36.3% Electronics 34.8% +1 more	1 … 4	25 … 500	25 … 2000

ery History

QUESTION 6

```
36  ---Q6. Display all transactions where the Price per Unit is between 100 and 500.
37  ---Expected output: All columns
38  SELECT*
39  FROM RETAIL.SALES.DATA
40  WHERE PRICE_PER_UNIT BETWEEN 100 AND 500;
41
```

Results (just now)

Table | Chart — 396 rows — 1.4s

	TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT	TOTAL_AMOUNT
	2 … 994	01/01… 01/01…	CUST002 0.3% CUST004 0.3% +98 more	Female 51.8% Male 48.2%	18 … 64	Electronics 35.1% Clothing 34.3% +1 more	1 … 4	300 … 500	300 … 2000
1	2	2023-02-27	CUST002	Female	26	Clothing	2	500	1000
2	4	2023-05-21	CUST004	Male	37	Clothing	1	500	500
3	9	2023-12-13	CUST009	Male	63	Electronics	2	300	600
4	13	2023-08-05	CUST013	Male	22	Electronics	3	500	1500

Query History

QUESTION 7

```
42  ---Q7. Display all transactions where the Product Category is either 'Beauty' or
43  ---'Electronics'.
44  SELECT*
45  FROM RETAIL.SALES.DATA
46  WHERE PRODUCT_CATEGORY = 'BEAUTY' OR PRODUCT_CATEGORY ='ELECTRONICS';
47
48  ---Expected output: All columns
49  ---Q8. Display all transactions where the Product Category is not 'Clothing'.
50  ---Expected output: All columns
```

Results (just now)

Table | Chart — 0 rows — 705ms

TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT	TOTAL_AMOUNT

ODION OKONOFUA PRACTICAL 1

QUESTION 8

```
SELECT*
FROM RETAIL.SALES.DATA
WHERE PRODUCT_CATEGORY <>'CLOTHING';

---Q9. Display all transactions where the Quantity is greater than or equal to 3.
---Expected output: All columns
```

Results (just now)

Table | Chart — 1,000 rows — 2.2s

	TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT	TOTAL_AMOUNT
	1 … 1000	01/01…01/01…	CUST001 0.1%, CUST002 0.1%, +98 more	Female 51.0%, Male 49.0%	18 … 64	Clothing 35.1%, Electronics 34.2%, +1 more	1 … 4	25 … 500	25 … 2000
1	1	2023-11-24	CUST001	Male	34	Beauty	3	50	150
2	2	2023-02-27	CUST002	Female	26	Clothing	2	500	1000
3	3	2023-01-13	CUST003	Male	50	Electronics	1	30	30
4	4	2023-05-21	CUST004	Male	37	Clothing	1	500	500

Query History

Question 9

```
SELECT*
FROM RETAIL.SALES.DATA
WHERE QUANTITY>=3;
```

Results (just now)

Table | Chart — 504 rows — 1.0s

	TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT	TOTAL_AMOUNT
	1 … 1000	01/01…01/01…	CUST001 0.2%, CUST008 0.2%, +98 more	Female 52.6%, Male 47.4%	18 … 64	Clothing 36.3%, Electronics 32.3%, +1 more	3 … 4	25 … 500	75 … 2000
1	1	2023-11-24	CUST001	Male	34	Beauty	3	50	150
2	8	2023-02-22	CUST008	Male	30	Electronics	4	25	100
3	10	2023-10-07	CUST010	Female	52	Clothing	4	50	200
4	12	2023-10-30	CUST012	Male	35	Beauty	3	25	75

QUESTION 10

```
---4. Aggregate Functions
---Q10. Count the total number of transactions.
---Expected output: Total_Transactions
SELECT COUNT(TRANSACTION_ID) AS TOTAL_TRANSACTIONS
FROM RETAIL.SALES.DATA;
```

Results (just now)

Table | Chart — 1 row — 58ms

	TOTAL_TRANSACTIONS
1	1000

QUESTION 11

```
---Q11. Find the average Age of customers.
---Expected output: Average_Age
SELECT AVG(AGE) AS AVERAGE_AGE
FROM RETAIL.SALES.DATA;

---Q12. Find the total quantity of products sold.
---Expected output: Total_Quantity
---Q13. Find the maximum Total Amount spent in a single transaction.
---Expected output: Max_Total_Amount
```

Results (just now)

Table | Chart — 1 row — 730ms

	AVERAGE_AGE
1	41.392000

ODION OKONOFUA PRACTICAL 1

QUESTION 12

```
71  ---Q12. Find the total quantity of products sold.
72  ---Expected output: Total_Quantity
73
74  SELECT SUM(TOTAL_AMOUNT) AS TOTAL_QUANTITY
75  FROM RETAIL.SALES.DATA;
76
```

Results (just now)

Table | Chart — 1 row — 257ms

	TOTAL_QUANTITY
1	456000

QUESTION 13

```
    SELECT MAX(TOTAL_AMOUNT) AS MAX_TOTAL_AMOUNT
    FROM RETAIL.SALES.DATA;

    ---Q14. Find the minimum Price per Unit in the dataset.
    ---Expected output: Min_Price_per_Unit
```

ults (just now)

able | Chart — 1 row — 41ms

	MAX_TOTAL_AMOUNT
	2000

QUESTION 14

```
83  ---Q14. Find the minimum Price per Unit in the dataset.
84  ---Expected output: Min_Price_per_Unit
85  SELECT MIN(PRICE_PER_UNIT) AS MIN_PRICE_PER_UNIT
86  FROM RETAIL.SALES.DATA;
```

Results (just now)

Table | Chart — 1 row — 330ms

	MIN_PRICE_PER_UNIT
1	25

QUESTION 15

```
88  ---5. GROUP BY Statement
89  ---Q15. Find the number of transactions per Product Category.
90  ---Expected output: Product Category, Transaction_Count
91  SELECT Product_category,
92  COUNT(TRANSACTION_ID)AS TRANSACTION_COUNT
93  from retail.sales.data
94  GROUP BY PRODUCT_CATEGORY;
95
```

esults (just now)

Table | Chart — 3 rows — 1.8s

	PRODUCT_CATEGORY	TRANSACTION_COUNT
1	Beauty	307
2	Clothing	351
3	Electronics	342

ODION OKONOFUA PRACTICAL 1

QUESTION 16

```
96  ---Q16. Find the total revenue (Total Amount) per gender.
97  ---Expected output: Gender, Total_Revenue
98  SELECT GENDER,
99  SUM(TOTAL_AMOUNT) AS TOTAL_REVENUE
100 FROM RETAIL.SALES.DATA
101 GROUP BY GENDER;
102 ---Q17. Find the average Price per Unit per product category.
103 ---Expected output: Product Category, Average_Price
104
105  Ctrl+I to generate
106
```

Results (1 minute ago)

Table | Chart — 2 rows — 73ms

	GENDER	TOTAL_REVENUE
1	Male	223160
2	Female	232840

QUSETION 17

```
103 ---Q17. Find the average Price per Unit per product category.
104 ---Expected output: Product Category, Average_Price
105 SELECT PRODUCT_CATEGORY,
106 AVG(PRICE_PER_UNIT) AS AVERAGE_PRICE
107 FROM RETAIL.SALES.DATA
108 GROUP BY PRODUCT_CATEGORY;
```

Results (just now)

Table | Chart — 3 rows — 734ms

	PRODUCT_CATEGORY	AVERAGE_PRICE
1	Beauty	184.055375
2	Clothing	174.287749
3	Electronics	181.900585

QUESTION 18

```
110 ---6. HAVING Clause
111 ---Q18. Find the total revenue per product category where total revenue is greater than
112 ---10,000.
113 SELECT PRODUCT_CATEGORY,
114 SUM(TOTAL_AMOUNT) AS TOTAL_REVENUE
115 FROM RETAIL.SALES.DATA
116 GROUP BY PRODUCT_CATEGORY
117 HAVING SUM(TOTAL_AMOUNT) > 10000;
```

esults (just now)

Table | Chart — 3 rows — 1.8s

	PRODUCT_CATEGORY	TOTAL_REVENUE
1	Beauty	143515
2	Clothing	155580
3	Electronics	156905

QUESTION 19

ODION OKONOFUA PRACTICAL 1

```
---Expected output: Product Category, Total_Revenue
---Q19. Find the average quantity per product category where the average is more than 2.
---Expected output: Product Category, Average_Quantity
SELECT PRODUCT_CATEGORY,
AVG(QUANTITY) AS AVERAGE_QUANTITY
FROM RETAIL.SALES.DATA
GROUP BY PRODUCT_CATEGORY
HAVING AVG(QUANTITY)>2;
```

Results (just now)

Table | Chart — 3 rows — 851ms

	PRODUCT_CATEGORY	AVERAGE_QUANTITY
1	Beauty	2.511401
2	Clothing	2.547009
3	Electronics	2.482456

QUESTION 20

```
SELECT TRANSACTION_ID,
TOTAL_AMOUNT,
CASE
WHEN TOTAL_AMOUNT>1000 THEN 'HIGH'
ELSE 'LOW'
END AS SPENDING_LEVEL
FROM RETAIL.SALES.DATA;
```

Results (1 minute ago)

Table | Chart — 1,000 rows — 2.8s

	TRANSACTION_ID (1 – 1000)	TOTAL_AMOUNT (25 – 2000)	SPENDING_LEVEL (LOW 84.7%, HIGH 15.3%)
1	1	150	LOW
2	2	1000	LOW
3	3	30	LOW
4	4	500	LOW

uery History

QUESTION 21

```
SELECT CUSTOMER_ID,
AGE,
CASE
WHEN AGE<30 THEN 'YOUTH'
WHEN AGE BETWEEN 30 AND 59 THEN 'ADULT'
WHEN AGE>=60 THEN 'SENIOR'
END AS AGE_GROUP
FROM RETAIL.SALES.DATA;
```

esults (just now)

Table | Chart — 1,000 rows — 70ms

	CUSTOMER_ID (CUST001 0.1%, CUST002 0.1%, +98 more)	AGE (18 – 64)	AGE_GROUP (ADULT 63.4%, YOUTH 25.1%, +1 more)
1	CUST001	34	ADULT
2	CUST002	26	YOUTH
3	CUST003	50	ADULT
4	CUST004	37	ADULT