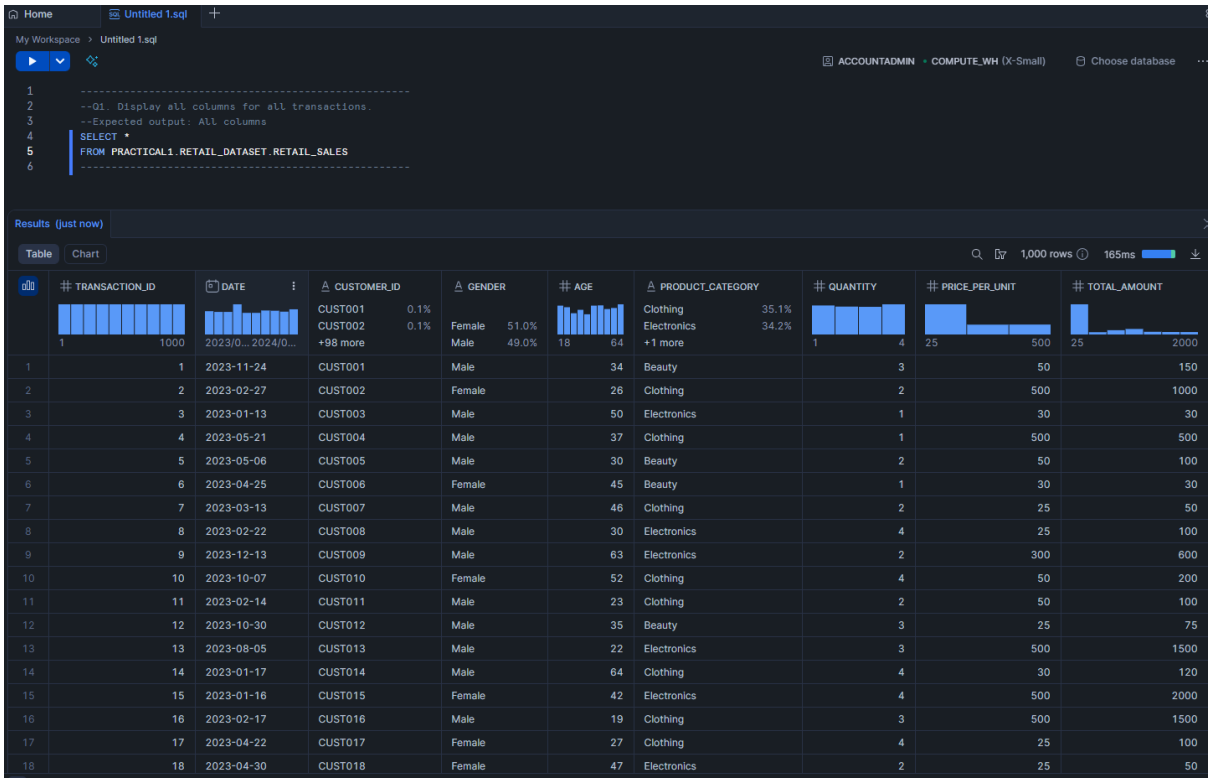
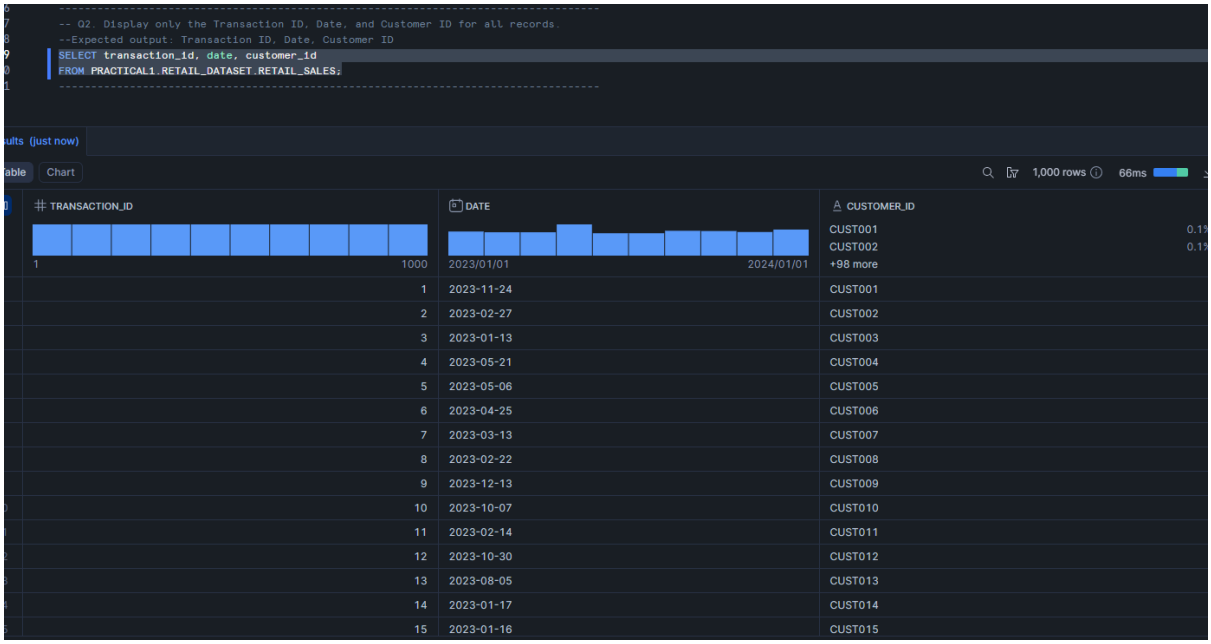
Practical 1

Question 1



	TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT	TOTAL_AMOUNT
	1 – 1000	2023/0… 2024/0…	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
5	5	2023-05-06	CUST005	Male	30	Beauty	2	50	100
6	6	2023-04-25	CUST006	Female	45	Beauty	1	30	30
7	7	2023-03-13	CUST007	Male	46	Clothing	2	25	50
8	8	2023-02-22	CUST008	Male	30	Electronics	4	25	100
9	9	2023-12-13	CUST009	Male	63	Electronics	2	300	600
10	10	2023-10-07	CUST010	Female	52	Clothing	4	50	200
11	11	2023-02-14	CUST011	Male	23	Clothing	2	50	100
12	12	2023-10-30	CUST012	Male	35	Beauty	3	25	75
13	13	2023-08-05	CUST013	Male	22	Electronics	3	500	1500
14	14	2023-01-17	CUST014	Male	64	Clothing	4	30	120
15	15	2023-01-16	CUST015	Female	42	Electronics	4	500	2000
16	16	2023-02-17	CUST016	Male	19	Clothing	3	500	1500
17	17	2023-04-22	CUST017	Female	27	Clothing	4	25	100
18	18	2023-04-30	CUST018	Female	47	Electronics	2	25	50

Question 2



	TRANSACTION_ID	DATE	CUSTOMER_ID
	1 – 1000	2023/01/01 – 2024/01/01	CUST001 0.1% CUST002 0.1% +98 more
	1	2023-11-24	CUST001
	2	2023-02-27	CUST002
	3	2023-01-13	CUST003
	4	2023-05-21	CUST004
	5	2023-05-06	CUST005
	6	2023-04-25	CUST006
	7	2023-03-13	CUST007
	8	2023-02-22	CUST008
	9	2023-12-13	CUST009
	10	2023-10-07	CUST010
	11	2023-02-14	CUST011
	12	2023-10-30	CUST012
	13	2023-08-05	CUST013
	14	2023-01-17	CUST014
	15	2023-01-16	CUST015

Question 3

```sql
-- Q3. Display all the distinct product categories in the dataset.
-- Expected output: Product Category
SELECT DISTINCT product_category
FROM PRACTICAL1.RETAIL_DATASET.RETAIL_SALES;
```

Results (just now)

Table | Chart — 3 rows — 430ms

	PRODUCT_CATEGORY
1	Clothing
2	Beauty
3	Electronics

Question 4

```sql
--Q4. Display all the distinct gender values in the dataset.
--Expected output: Gender
SELECT DISTINCT gender
FROM PRACTICAL1.RETAIL_DATASET.RETAIL_SALES;
```

Results (just now)

Table | Chart — 2 rows — 77ms

	GENDER
1	Male
2	Female

Question 5

```sql
-- Q5. Display all transactions where the Age is greater than 40.
--Expected output: All columns
SELECT *
FROM PRACTICAL1.RETAIL_DATASET.RETAIL_SALES
WHERE age > 40;
```

Results (just now)

Table | Chart — 534 rows — 74ms

	TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT	TOTAL_AMOUNT
	3 – 1000	2023/0… – 2024/0…	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
1	3	2023-01-13	CUST003	Male	50	Electronics	1	30	30
2	6	2023-04-25	CUST006	Female	45	Beauty	1	30	30
3	7	2023-03-13	CUST007	Male	46	Clothing	2	25	50
4	9	2023-12-13	CUST009	Male	63	Electronics	2	300	600
5	10	2023-10-07	CUST010	Female	52	Clothing	4	50	200
6	14	2023-01-17	CUST014	Male	64	Clothing	4	30	120
7	15	2023-01-16	CUST015	Female	42	Electronics	4	500	2000
8	18	2023-04-30	CUST018	Female	47	Electronics	2	25	50
9	19	2023-09-16	CUST019	Female	62	Clothing	2	25	50
10	21	2023-01-14	CUST021	Female	50	Beauty	1	500	500
11	24	2023-11-29	CUST024	Female	49	Clothing	1	300	300
12	25	2023-12-26	CUST025	Female	64	Beauty	1	50	50
13	28	2023-04-23	CUST028	Female	43	Beauty	1	500	500
14	29	2023-08-18	CUST029	Female	42	Electronics	1	30	30
15	31	2023-05-23	CUST031	Male	44	Electronics	4	300	1200
16	33	2023-03-23	CUST033	Female	50	Electronics	2	50	100

Question 6

```sql
--Q6. Display all transactions where the Price per Unit is between 100 and 500.
--Expected output: All columns
SELECT *
FROM PRACTICAL1.RETAIL_DATASET.RETAIL_SALES
WHERE price_per_unit BETWEEN 100 AND 500;
```

Results (just now)

Table | Chart — 396 rows — 75ms

	TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT	TOTAL_AMOUNT
	2 – 994	2023/0… 2024/0…	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
5	15	2023-01-16	CUST015	Female	42	Electronics	4	500	2000
6	16	2023-02-17	CUST016	Male	19	Clothing	3	500	1500
7	20	2023-11-05	CUST020	Male	22	Clothing	3	300	900
8	21	2023-01-14	CUST021	Female	50	Beauty	1	500	500
9	24	2023-11-29	CUST024	Female	49	Clothing	1	300	300
10	26	2023-10-07	CUST026	Female	28	Electronics	2	500	1000
11	28	2023-04-23	CUST028	Female	43	Beauty	1	500	500
12	30	2023-10-29	CUST030	Female	39	Beauty	3	300	900
13	31	2023-05-23	CUST031	Male	44	Electronics	4	300	1200
14	35	2023-08-05	CUST035	Female	58	Beauty	3	300	900
15	36	2023-06-24	CUST036	Male	52	Beauty	3	300	900
16	42	2023-02-17	CUST042	Male	22	Clothing	3	300	900
17	43	2023-07-14	CUST043	Female	48	Clothing	1	300	300

Question 7

```sql
--Q7. Display all transactions where the Product Category is either 'Beauty' or
--'Electronics'. Expected output: All columns
SELECT *
FROM PRACTICAL1.RETAIL_DATASET.RETAIL_SALES
WHERE product_category = 'Beauty' OR product_category = 'Electronics';
```

Results (just now)

Table | Chart — 649 rows — 71ms

	TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT	TOTAL_AMOUNT
	1 – 1000	2023/0… 2024/0…	CUST001 0.2%, CUST003 0.2%, +98 more	Female 51.8%, Male 48.2%	18 – 64	Electronics 52.7%, Beauty 47.3%	1 – 4	25 – 500	25 – 2000
1	1	2023-11-24	CUST001	Male	34	Beauty	3	50	150
2	3	2023-01-13	CUST003	Male	50	Electronics	1	30	30
3	5	2023-05-06	CUST005	Male	30	Beauty	2	50	100
4	6	2023-04-25	CUST006	Female	45	Beauty	1	30	30
5	8	2023-02-22	CUST008	Male	30	Electronics	4	25	100
6	9	2023-12-13	CUST009	Male	63	Electronics	2	300	600
7	12	2023-10-30	CUST012	Male	35	Beauty	3	25	75
8	13	2023-08-05	CUST013	Male	22	Electronics	3	500	1500
9	15	2023-01-16	CUST015	Female	42	Electronics	4	500	2000
10	18	2023-04-30	CUST018	Female	47	Electronics	2	25	50
11	21	2023-01-14	CUST021	Female	50	Beauty	1	500	500
12	25	2023-12-26	CUST025	Female	64	Beauty	1	50	50
13	26	2023-10-07	CUST026	Female	28	Electronics	2	500	1000
14	27	2023-08-03	CUST027	Female	38	Beauty	2	25	50
15	28	2023-04-23	CUST028	Female	43	Beauty	1	500	500
16	29	2023-08-18	CUST029	Female	42	Electronics	1	30	30
17	30	2023-10-29	CUST030	Female	39	Beauty	3	300	900
18	31	2023-05-23	CUST031	Male	44	Electronics	4	300	1200
19	32	2023-01-04	CUST032	Male	30	Beauty	3	30	90

Question 8

```sql
--Q8. Display all transactions where the Product Category is not 'Clothing'.
--Expected output: All columns
SELECT *
FROM PRACTICAL1.RETAIL_DATASET.RETAIL_SALES
WHERE product_category NOT IN ('Clothing');
```

Results (just now)

Table | Chart — 649 rows — 75ms

	TRANSACTION_ID (1 … 1000)	DATE (2023/0… 2024/0…)	CUSTOMER_ID (CUST001 0.2%, CUST003 0.2%, +98 more)	GENDER (Female 51.8%, Male 48.2%)	AGE (18 … 64)	PRODUCT_CATEGORY (Electronics 52.7%, Beauty 47.3%)	QUANTITY (1 … 4)	PRICE_PER_UNIT (25 … 500)	TOTAL_AMOUNT (25 … 2000)
1	1	2023-11-24	CUST001	Male	34	Beauty	3	50	150
2	3	2023-01-13	CUST003	Male	50	Electronics	1	30	30
3	5	2023-05-06	CUST005	Male	30	Beauty	2	50	100
4	6	2023-04-25	CUST006	Female	45	Beauty	1	30	30
5	8	2023-02-22	CUST008	Male	30	Electronics	4	25	100
6	9	2023-12-13	CUST009	Male	63	Electronics	2	300	600
7	12	2023-10-30	CUST012	Male	35	Beauty	3	25	75
8	13	2023-08-05	CUST013	Male	22	Electronics	3	500	1500
9	15	2023-01-16	CUST015	Female	42	Electronics	4	500	2000
10	18	2023-04-30	CUST018	Female	47	Electronics	2	25	50
11	21	2023-01-14	CUST021	Female	50	Beauty	1	500	500
12	25	2023-12-26	CUST025	Female	64	Beauty	1	50	50
13	26	2023-10-07	CUST026	Female	28	Electronics	2	500	1000
14	27	2023-08-03	CUST027	Female	38	Beauty	2	25	50
15	28	2023-04-23	CUST028	Female	43	Beauty	1	500	500
16	29	2023-08-18	CUST029	Female	42	Electronics	1	30	30
17	30	2023-10-29	CUST030	Female	39	Beauty	3	300	900
18	31	2023-05-23	CUST031	Male	44	Electronics	4	300	1200

Question 9

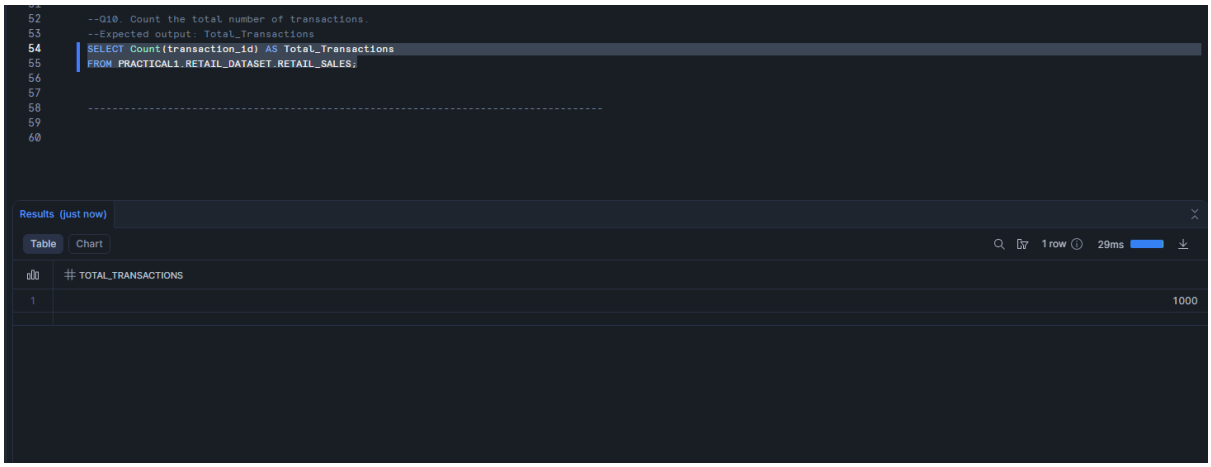
```sql
--Q9. Display all transactions where the Quantity is greater than or equal to 3.
--Expected output: All columns
SELECT *
FROM PRACTICAL1.RETAIL_DATASET.RETAIL_SALES
WHERE quantity >= 3;
```

Results (just now)

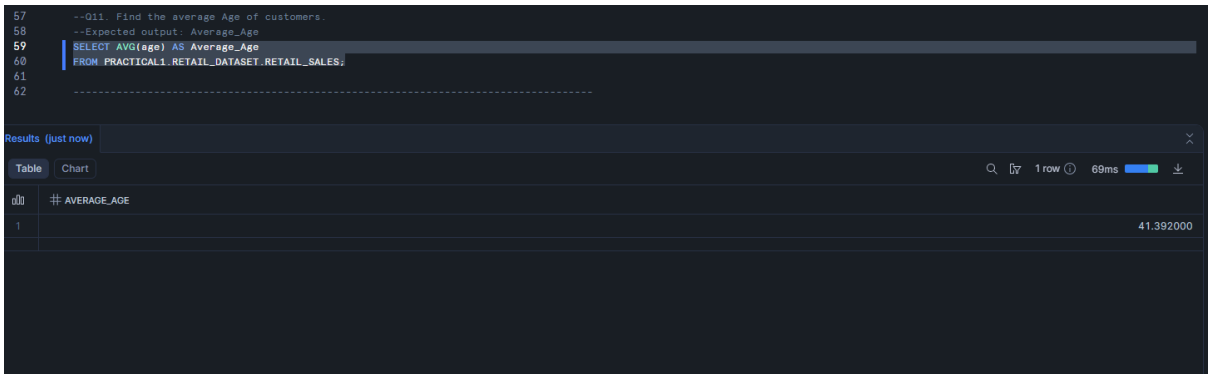
Table | Chart — 504 rows — 70ms

	TRANSACTION_ID (1 … 1000)	DATE (2023/0… 2024/0…)	CUSTOMER_ID (CUST001 0.2%, CUST008 0.2%, +98 more)	GENDER (Female 52.6%, Male 47.4%)	AGE (18 … 64)	PRODUCT_CATEGORY (Clothing 36.3%, Electronics 32.3%, +1 more)	QUANTITY (3 … 4)	PRICE_PER_UNIT (25 … 500)	TOTAL_AMOUNT (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
5	13	2023-08-05	CUST013	Male	22	Electronics	3	500	1500
6	14	2023-01-17	CUST014	Male	64	Clothing	4	30	120
7	15	2023-01-16	CUST015	Female	42	Electronics	4	500	2000
8	16	2023-02-17	CUST016	Male	19	Clothing	3	500	1500
9	17	2023-04-22	CUST017	Female	27	Clothing	4	25	100
10	20	2023-11-05	CUST020	Male	22	Clothing	3	300	900
11	23	2023-04-12	CUST023	Female	35	Clothing	4	30	120
12	30	2023-10-29	CUST030	Female	39	Beauty	3	300	900
13	31	2023-05-23	CUST031	Male	44	Electronics	4	300	1200
14	32	2023-01-04	CUST032	Male	30	Beauty	3	30	90
15	34	2023-12-24	CUST034	Female	51	Clothing	3	50	150
16	35	2023-08-05	CUST035	Female	58	Beauty	3	300	900
17	36	2023-06-24	CUST036	Male	52	Beauty	3	300	900
18	37	2023-05-23	CUST037	Female	18	Beauty	3	25	75

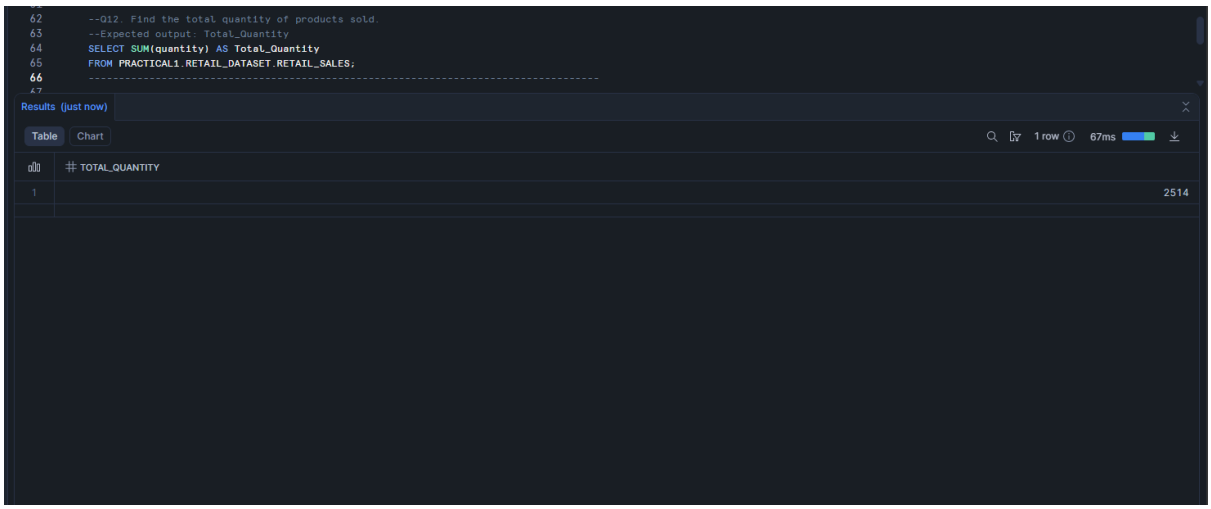
Question 10



Question 11



Question 12



Question 13

```
66  ------------------------------------------------------------------------------
67  --Q13. Find the maximum Total Amount spent in a single transaction.
68  --Expected output: Max_Total_Amount
69  SELECT MAX(total_amount) AS Max_Total_Amount
70  FROM PRACTICAL1.RETAIL_DATASET.RETAIL_SALES;
71
```

Results (just now)

Table | Chart — 1 row — 38ms

	MAX_TOTAL_AMOUNT
1	2000

Question 14

```
71  ------------------------------------------------------------------------------
72  --Q14. Find the minimum Price per Unit in the dataset.
73  --Expected output: Min_Price_per_Unit
74  SELECT MIN(total_amount) AS Max_Total_Amount
75  FROM PRACTICAL1.RETAIL_DATASET.RETAIL_SALES;
76   Ctrl+I to generate
```

Results (just now)

Table | Chart — 1 row — 27ms

	MAX_TOTAL_AMOUNT
1	25

Question 15

```
77  --Q15. Find the number of transactions per Product Category.
78  --Expected output: Product Category, Transaction_Count
79  SELECT product_category,
80         Count(transaction_id) AS Transaction_Count
81  FROM PRACTICAL1.RETAIL_DATASET.RETAIL_SALES
82  GROUP BY product_category;
83
84
```

Results (just now)

Table | Chart — 3 rows — 434ms

	PRODUCT_CATEGORY	TRANSACTION_COUNT
1	Beauty	307
2	Clothing	351
3	Electronics	342

Question 16

```
83  ------------------------------------------------------------------------------
84  --Q16. Find the total revenue (Total Amount) per gender.
85  --Expected output: Gender, Total_Revenue
86  SELECT gender,
87         SUM(total_amount) AS Total_Revenue
88  FROM PRACTICAL1.RETAIL_DATASET.RETAIL_SALES
89  GROUP BY gender;
90  ------------------------------------------------------------------------------
```

Results (just now)

Table | Chart — 2 rows — 75ms

	GENDER	TOTAL_REVENUE
1	Male	223160
2	Female	232840

Question 17

```
--Q17. Find the average Price per Unit per product category.
--Expected output: Product Category, Average_Price
SELECT product_category,
       AVG(price_per_unit) AS Average_Price
FROM PRACTICAL1.RETAIL_DATASET.RETAIL_SALES
GROUP BY product_category;
```

Results (just now) — Table | Chart — 3 rows — 88ms

	PRODUCT_CATEGORY	AVERAGE_PRICE
1	Beauty	184.055375
2	Clothing	174.287749
3	Electronics	181.900585

Question 18

```
--Q18. Find the total revenue per product category where total revenue is greater
--than 10,000. Expected output: Product Category, Total_Revenue
SELECT product_category,
       SUM(total_amount) AS Total_Revenue
FROM PRACTICAL1.RETAIL_DATASET.RETAIL_SALES
GROUP BY product_category
HAVING SUM(total_amount) > 10000;
```

Results (just now) — Table | Chart — 3 rows — 87ms

	PRODUCT_CATEGORY	TOTAL_REVENUE
1	Beauty	143515
2	Clothing	155580
3	Electronics	156905

Question 19

```
---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 PRACTICAL1.RETAIL_DATASET.RETAIL_SALES
GROUP BY product_category
HAVING AVG(quantity) > 2 ;
```

Results (1 minute ago) — Table | Chart — 3 rows — 93ms

	PRODUCT_CATEGORY	AVERAGE_QUANTITY
1	Beauty	2.511401
2	Clothing	2.547009
3	Electronics	2.482456

Question 20

```sql
--Q20. Display a column called Spending_Level that shows 'High' if Total Amount > 1000,
--otherwise 'Low'.
--Expected output: Transaction ID, Total Amount, Spending_Level
SELECT transaction_id,total_amount,
       CASE
           WHEN total_amount > 1000 THEN 'High'
           ELSE 'Low'
       END AS Spending_Level
FROM PRACTICAL1.RETAIL_DATASET.RETAIL_SALES;

--Q21. Display a new column called Age_Group that labels customers as:
-- • 'Youth' if Age < 30
-- • 'Adult' if Age is between 30 and 59
-- • 'Senior' if Age >= 60
-- Expected output: Customer ID, Age, Age_Group
```

Results (4 minutes ago)

Table | Chart — 1,000 rows — 1.1s

	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
5	5	100	Low
6	6	30	Low
7	7	50	Low
8	8	100	Low
9	9	600	Low
10	10	200	Low

Question 21

```sql
--Q21. Display a new column called Age_Group that labels customers as:
-- • 'Youth' if Age < 30
-- • 'Adult' if Age is between 30 and 59
-- • 'Senior' if Age >= 60
-- Expected output: Customer ID, Age, Age_Group
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 PRACTICAL1.RETAIL_DATASET.RETAIL_SALES;
```

Results (just now)

Table | Chart — 1,000 rows — 426ms

	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
5	CUST005	30	Adult
6	CUST006	45	Adult
7	CUST007	46	Adult
8	CUST008	30	Adult
9	CUST009	63	Senior
10	CUST010	52	Adult
11	CUST011	23	Youth
12	CUST012	35	Adult
13	CUST013	22	Youth
14	CUST014	64	Senior