PRACTICAL 1

Q1.

```sql
-- Q1. Display all columns for all transactions.

SELECT
      TRANSACTION_ID,
      DATE,
      CUSTOMER_ID,
      GENDER,
      AGE,
      PRODUCT_CATEGORY,
      QUANTITY,
      PRICE_PER_UNIT,
      TOTAL_AMOUNT,
FROM
  "RETAIL_SALES"."PUBLIC"."RETAIL_SALES_ANALYSIS";
```

Results (1 minute ago)

Table | Chart — 1,000 rows — 61ms

	TRANSACTION_ID	DATE	CUSTOMER_ID	GENDE	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT
	1 1000	01/0... 01/0...	CUST001 0.1% CUST002 0.1% +98 more	Fe... 51.0% Male 49.0%	18 64	Clothing 35.1% Electronics 34.2% +1 more	1 4	25 500
1	1	2023-11-24	CUST001	Male	34	Beauty	3	50
2	2	2023-02-27	CUST002	Female	26	Clothing	2	500
3	3	2023-01-13	CUST003	Male	50	Electronics	1	30
4	4	2023-05-21	CUST004	Male	37	Clothing	1	500
5	5	2023-05-06	CUST005	Male	30	Beauty	2	50
6	6	2023-04-25	CUST006	Female	45	Beauty	1	30
7	7	2023-03-13	CUST007	Male	46	Clothing	2	25
8	8	2023-02-22	CUST008	Male	30	Electronics	4	25

```sql
-- Q2. Display only the Transaction ID, Date, and Customer ID for all records.

SELECT
      TRANSACTION_ID,
      DATE,
      CUSTOMER_ID,
FROM
   RETAIL_SALES.PUBLIC.RETAIL_SALES_ANALYSIS;
```

Results (just now)

Table | Chart — 1,000 rows — 33ms

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

Workspace

```sql
-- Q3. Display all the distinct product categories in the dataset.

SELECT
DISTINCT
        PRODUCT_CATEGORY,
FROM
   RETAIL_SALES.PUBLIC.RETAIL_SALES_ANALYSIS;
```

Results (just now)

Table | Chart — 3 rows — 24ms

	PRODUCT_CATEGORY
1	Clothing
2	Beauty
3	Electronics

```sql
-- Q4. Display all the distinct gender values in the dataset
SELECT
DISTINCT
        GENDER,
FROM
    RETAIL_SALES.PUBLIC.RETAIL_SALES_ANALYSIS;
```

Results (just now)

Table | Chart — 2 rows — 24ms

	GENDER
1	Male
2	Female

```sql
-- Q5. Display all transactions where the Age is greater than 40.

SELECT
        TRANSACTION_ID,
        DATE,
        CUSTOMER_ID,
        GENDER,
        AGE,
        PRODUCT_CATEGORY,
        QUANTITY,
        PRICE_PER_UNIT,
        TOTAL_AMOUNT,
FROM
    RETAIL_SALES.PUBLIC.RETAIL_SALES_ANALYSIS
WHERE AGE > 40;
```

Results (just now)

Table | Chart — 534 rows — 36ms

	TRANSACTION_ID (3 … 1000)	DATE (01/0… 01/0…)	CUSTOMER_ID (CUST003 0.2%, CUST006 0.2%, +98 more)	GENDER (Fe… 51.3%, Male 48.7%)	AGE (41 … 64)	PRODUCT_CATEGORY (Clothing 36.3%, Electronics 34.8%, +1 more)	QUANTITY (1 … 4)	PRICE_PER_UNIT (25 … 500)
1	3	2023-01-13	CUST003	Male	50	Electronics	1	30
2	6	2023-04-25	CUST006	Female	45	Beauty	1	30
3	7	2023-03-13	CUST007	Male	46	Clothing	2	25
4	9	2023-12-13	CUST009	Male	63	Electronics	2	300
5	10	2023-10-07	CUST010	Female	52	Clothing	4	50
6	14	2023-01-17	CUST014	Male	64	Clothing	4	30
7	15	2023-01-16	CUST015	Female	42	Electronics	4	500
8	18	2023-04-30	CUST018	Female	47	Electronics	2	25

```sql
-- Q6. Display all transactions where the Price per Unit is between 100 and 500.
SELECT
        TRANSACTION_ID,
        DATE,
        CUSTOMER_ID,
        GENDER,
        AGE,
        PRODUCT_CATEGORY,
        QUANTITY,
        PRICE_PER_UNIT,
        TOTAL_AMOUNT,
FROM
    RETAIL_SALES.PUBLIC.RETAIL_SALES_ANALYSIS

WHERE PRICE_PER_UNIT BETWEEN 100 AND 500;
```

Results (just now)

Table | Chart — 396 rows — 25ms

	TRANSACTION_ID (2 … 994)	DATE (01/0… 01/0…)	CUSTOMER_ID (CUST002 0.3%, CUST004 0.3%, +98 more)	GENDER (Fe… 51.8%, Male 48.2%)	AGE (18 … 64)	PRODUCT_CATEGORY (Electronics 35.1%, Clothing 34.3%, +1 more)	QUANTITY (1 … 4)	PRICE_PER_UNIT (300 … 500)
1	2	2023-02-27	CUST002	Female	26	Clothing	2	500
2	4	2023-05-21	CUST004	Male	37	Clothing	1	500
3	9	2023-12-13	CUST009	Male	63	Electronics	2	300
4	13	2023-08-05	CUST013	Male	22	Electronics	3	500
5	15	2023-01-16	CUST015	Female	42	Electronics	4	500
6	16	2023-02-17	CUST016	Male	19	Clothing	3	500
7	20	2023-11-05	CUST020	Male	22	Clothing	3	300

```sql
-- Q7. Display all transactions where the Product Category is either 'Beauty' or 'Electronics'.

SELECT
        TRANSACTION_ID,
        DATE,
        CUSTOMER_ID,
        GENDER,
        AGE,
        PRODUCT_CATEGORY,
        QUANTITY,
        PRICE_PER_UNIT,
        TOTAL_AMOUNT,
FROM RETAIL_SALES.PUBLIC.RETAIL_SALES_ANALYSIS
WHERE PRODUCT_CATEGORY IN ('Beauty', 'Electronics');
```

Results (just now)

Table | Chart — 649 rows — 30ms

	TRANSACTION_ID	DATE	CUSTOMER_	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT
	1 … 1000	01/0… 01/0…	CUST001 0.2% CUST003 0.2% +98 more	Fe… 51.8% Male 48.2%	18 … 64	Electronics 52.7% Beauty 47.3%	1 … 4	25 … 500
2	3	2023-01-13	CUST003	Male	50	Electronics	1	30
3	5	2023-05-06	CUST005	Male	30	Beauty	2	50
4	6	2023-04-25	CUST006	Female	45	Beauty	1	30
5	8	2023-02-22	CUST008	Male	30	Electronics	4	25
6	9	2023-12-13	CUST009	Male	63	Electronics	2	300
7	12	2023-10-30	CUST012	Male	35	Beauty	3	25
8	13	2023-08-05	CUST013	Male	22	Electronics	3	500
9	15	2023-01-16	CUST015	Female	42	Electronics	4	500

```sql
-- Q8. Display all transactions where the Product Category is not 'Clothing'.

SELECT
        TRANSACTION_ID,
        DATE,
        CUSTOMER_ID,
        GENDER,
        AGE,
        PRODUCT_CATEGORY,
        QUANTITY,
        PRICE_PER_UNIT,
        TOTAL_AMOUNT,
FROM RETAIL_SALES.PUBLIC.RETAIL_SALES_ANALYSIS
WHERE PRODUCT_CATEGORY <> 'Clothing';
```

Results (just now)

Table | Chart — 649 rows — 32ms

	TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT
	1 … 1000	01/0… 01/0…	CUST001 0.2% CUST003 0.2% +98 more	Fe… 51.8% Male 48.2%	18 … 64	Electronics 52.7% Beauty 47.3%	1 … 4	25 … 500
1	1	2023-11-24	CUST001	Male	34	Beauty	3	50
2	3	2023-01-13	CUST003	Male	50	Electronics	1	30
3	5	2023-05-06	CUST005	Male	30	Beauty	2	50
4	6	2023-04-25	CUST006	Female	45	Beauty	1	30
5	8	2023-02-22	CUST008	Male	30	Electronics	4	25
6	9	2023-12-13	CUST009	Male	63	Electronics	2	300
7	12	2023-10-30	CUST012	Male	35	Beauty	3	25
8	13	2023-08-05	CUST013	Male	22	Electronics	3	500

```sql
-- Q9. Display all transactions where the Quantity is greater than or equal to 3.

SELECT
        TRANSACTION_ID,
        DATE,
        CUSTOMER_ID,
        GENDER,
        AGE,
        PRODUCT_CATEGORY,
        QUANTITY,
        PRICE_PER_UNIT,
        TOTAL_AMOUNT,
FROM    RETAIL_SALES.PUBLIC.RETAIL_SALES_ANALYSIS
WHERE   QUANTITY >= 3;
```

esults (just now)

Table | Chart — 504 rows — 54ms

	TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT
	1 … 1000	01/0… 01/0…	CUST001 0.2%, CUST008 0.2%, +98 more	Fe… 52.6%, Male 47.4%	18 … 64	Clothing 36.3%, Electronics 32.3%, +1 more	3 … 4	25 … 500
1	1	2023-11-24	CUST001	Male	34	Beauty	3	50
2	8	2023-02-22	CUST008	Male	30	Electronics	4	25
3	10	2023-10-07	CUST010	Female	52	Clothing	4	50
4	12	2023-10-30	CUST012	Male	35	Beauty	3	25
5	13	2023-08-05	CUST013	Male	22	Electronics	3	500
6	14	2023-01-17	CUST014	Male	64	Clothing	4	30
7	15	2023-01-16	CUST015	Female	42	Electronics	4	500
8	16	2023-02-17	CUST016	Male	19	Clothing	3	500

```sql
-- Q10. Count the total number of transactions.

SELECT
        COUNT(*) AS TOTAL_TRANSACTIONS
FROM    RETAIL_SALES.PUBLIC.RETAIL_SALES_ANALYSIS;
```

Results (just now)

Table | Chart — 1 row — 23ms

	TOTAL_TRANSACTIONS
1	1000

```sql
-- Q11. Find the average Age of customers.

SELECT
        AVG("AGE") AS AVERAGE_AGE
FROM    RETAIL_SALES.PUBLIC.RETAIL_SALES_ANALYSIS;
```

Results (just now)

Table | Chart — 1 row — 25ms

	TOTAL_TRANSACTIONS
1	1000

```sql
-- Q12. Find the total quantity of products sold.

SELECT
        SUM("QUANTITY") AS TOTAL_QUANTITY
FROM    RETAIL_SALES.PUBLIC.RETAIL_SALES_ANALYSIS;
```

Results (just now)

Table | Chart — 1 row — 26ms

	TOTAL_QUANTITY
1	2514

```sql
-- Q13. Find the maximum Total Amount spent in a single transaction.

SELECT
        MAX("TOTAL_AMOUNT") AS  MAX_TOTAL_AMOUNT
FROM    RETAIL_SALES.PUBLIC.RETAIL_SALES_ANALYSIS;
```

Results (just now)

Table | Chart — 1 row — 30ms

	MAX_TOTAL_AMOUNT
1	2000

```sql
-- Q14. Find the minimum Price per Unit in the dataset.

SELECT
        MIN("PRICE_PER_UNIT") AS MIN_PRICE_UNIT
FROM    RETAIL_SALES.PUBLIC.RETAIL_SALES_ANALYSIS;
```

Results (just now)

Table | Chart — 1 row — 26ms

	MIN_PRICE_UNIT
1	25

```sql
-- Q15. Find the number of transactions per Product Category.

SELECT
        PRODUCT_CATEGORY,
        COUNT(*) AS TRANSACTION_COUNT
FROM    RETAIL_SALES.PUBLIC.RETAIL_SALES_ANALYSIS
GROUP BY ("PRODUCT_CATEGORY");
```

Results (just now)

Table | Chart — 3 rows — 30ms

	PRODUCT_CATEGORY	TRANSACTION_COUNT
1	Clothing	351
2	Beauty	307
3	Electronics	342

```sql
-- Q16. Find the total revenue (Total Amount) per gender.

SELECT
        GENDER,
        SUM("TOTAL_AMOUNT") AS TOTAL_REVENUE
FROM    RETAIL_SALES.PUBLIC.RETAIL_SALES_ANALYSIS
GROUP BY ("GENDER");

```

Results (just now)

Table | Chart — 2 rows — 46ms

	GENDER	TOTAL_REVENUE
1	Male	223160
2	Female	232840

```sql
-- Q17. Find the average Price per Unit per product category.
SELECT
        PRODUCT_CATEGORY,
        AVG("PRICE_PER_UNIT") AS AVERAGE_PRICE
FROM    RETAIL_SALES.PUBLIC.RETAIL_SALES_ANALYSIS
GROUP BY (PRODUCT_CATEGORY);

```

Results (just now)

Table | Chart — 3 rows — 28ms

	PRODUCT_CATEGORY	AVERAGE_PRICE
1	Beauty	184.055375
2	Clothing	174.287749
3	Electronics	181.900585

```sql
--Q18. Find the total revenue per product category where total revenue is greater than 10,000.

SELECT
        PRODUCT_CATEGORY,
        SUM("TOTAL_AMOUNT") AS TOTAL_REVENUE
FROM    RETAIL_SALES.PUBLIC.RETAIL_SALES_ANALYSIS
GROUP BY ("PRODUCT_CATEGORY")
HAVING SUM("TOTAL_AMOUNT") > 10000;

```

Results (just now)

Table | Chart — 3 rows — 38ms

	PRODUCT_CATEGORY	TOTAL_REVENUE
1	Beauty	143515
2	Clothing	155580
3	Electronics	156905

Workspaces

```sql
-- Q19. Find the average quantity per product category where the average is more than 2.

SELECT
        PRODUCT_CATEGORY,
        AVG("QUANTITY") AS AVERAGE_QUANTITY
FROM RETAIL_SALES.PUBLIC.RETAIL_SALES_ANALYSIS
GROUP BY PRODUCT_CATEGORY
HAVING AVG("QUANTITY") > 2;
```

Results (just now)

Table | Chart — 3 rows — 31ms

	PRODUCT_CATEGORY	AVERAGE_QUANTITY
1	Beauty	2.511401
2	Clothing	2.547009
3	Electronics	2.482456

```sql
-- Q20. Display a column called Spending_Level that shows 'High' if Total Amount > 1000, otherwise 'Low'.
SELECT
        TRANSACTION_ID,
        TOTAL_AMOUNT,
        CASE
            WHEN "TOTAL_AMOUNT" > 1000 THEN 'HIGH'
            ELSE 'LOW'
        END AS SPENDING_LEVEL
FROM RETAIL_SALES.PUBLIC.RETAIL_SALES_ANALYSIS;
```

Results (just now)

Table | Chart — 1,000 rows — 30ms

	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
11	11	100	LOW

Query History

```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

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.PUBLIC.RETAIL_SALES_ANALYSIS;
```

Results (just now)

Table | Chart

1,000 rows | 27ms

	CUSTOMER_ID	AGE	AGE_GROUP
	CUST001 0.1% CUST002 0.1% +98 more	 18 64	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