Belinda Thando Mhlanga

Practical 1 - SQL Fundamentals (Snowflake-Basic SQL Syntax) Questions

Q1. Display all columns for all transactions.

```
SELECT *
FROM RETAIL.SALES.RETAIL_SALES;
 Ctrl+I to generate
```

Results (just now)

Table | Chart — 1,000 rows — 74ms

	TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT	TOTAL_AMOUNT
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

Question 2.

```
SELECT
    Transaction_ID,
    Date,
    Customer_ID
FROM RETAIL.SALES.RETAIL_SALES;
```

Results (just now)

Table | Chart — 1,000 rows

	TRANSACTION_ID	DATE	CUSTOMER_ID
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

Question 3

```
-- Q3. Display all the distinct product categories in the dataset.
SELECT DISTINCT Product_Category
FROM RETAIL.SALES.RETAIL_SALES;
```

Results (just now)

Table | Chart

	PRODUCT_CATEGORY
1	Clothing
2	Beauty
3	Electronics

Question 4

```
19  -- Q4. Display all the distinct gender values in the dataset.
20  SELECT DISTINCT Gender
21  FROM RETAIL.SALES.RETAIL_SALES;
22
```

Results (just now)

Table | Chart

	GENDER
1	Male
2	Female

Question 5

```
23  -- Q5. Display all transactions where the Age is greater than 40.
24  SELECT *
25  FROM RETAIL.SALES.RETAIL_SALES
26  WHERE Age > 40;
```

Results (just now)

Table | Chart — 534 rows — 26ms

	TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT	TOTAL_AMOUNT
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

Question 6

```
28  -- Q6. Display all transactions where the Price per Unit is between 100 and 500.
29  SELECT *
30  FROM RETAIL.SALES.RETAIL_SALES
31  WHERE Price_per_Unit BETWEEN 100 AND 500;
32
```

Results (just now)

Table | Chart — 396 rows — 68ms

	TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT	TOTAL_AMOUNT
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

Query History

Question 7

```sql
-- Q7. Display all transactions where the Product Category is either 'Beauty' or 'Electronics'.
SELECT *
FROM RETAIL.SALES.RETAIL_SALES
WHERE Product_Category IN ('Beauty', 'Electronics');
```

Results (just now)

Table | Chart — 649 rows — 76ms

	TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT	TOTAL_AMOUNT
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

Question 8

```sql
SELECT *
FROM RETAIL.SALES.RETAIL_SALES
WHERE Product_Category <> 'Clothing';
```

Results (just now)

Table | Chart — 649 rows — 71ms

	TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT	TOTAL_AMOUNT
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

Question 9

```sql
-- Q9. Display all transactions where the Quantity is greater than or equal to 3.
SELECT *
FROM RETAIL.SALES.RETAIL_SALES
WHERE Quantity >= 3;
```

Results (just now)

Table | Chart — 504 rows — 63ms

	TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT	TOTAL_AMOUNT
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

Question 10

```sql
-- Q10. Count the total number of transactions.
SELECT COUNT(*) AS Total_Transactions
FROM RETAIL.SALES.RETAIL_SALES;
```

Results (just now)

Table | Chart — 1 row — 20ms

	TOTAL_TRANSACTIONS
1	1000

Question 11

```sql
-- Q11. Find the average Age of customers.
SELECT AVG(Age) AS Average_Age
FROM RETAIL.SALES.RETAIL_SALES;
```

Results (just now)

Table | Chart — 1 row — 61ms

	AVERAGE_AGE
1	41.392000

Question 12

```sql
-- Q12. Find the total quantity of products sold.
SELECT SUM(Quantity) AS Total_Quantity
FROM RETAIL.SALES.RETAIL_SALES;
```

Results (just now)

Table | Chart — 1 row — 65ms

	TOTAL_QUANTITY
1	2514

Question 13

```sql
-- Q13. Find the maximum Total Amount spent in a single transaction.
SELECT MAX(Total_Amount) AS Max_Total_Amount
FROM RETAIL.SALES.RETAIL_SALES;
```

Results (just now)

Table | Chart — 1 row — 65ms

	TOTAL_QUANTITY
1	2514

Question 14

```sql
-- Q14. Find the minimum Price per Unit in the dataset.
SELECT MIN(Price_per_Unit) AS Min_Price_per_Unit
FROM RETAIL.SALES.RETAIL_SALES;
```

Results (just now)

Table | Chart — 1 row — 20ms

	MIN_PRICE_PER_UNIT
1	25

Question 15

```sql
-- Q15. Find the number of transactions per Product Category.
SELECT
    Product_Category,
    COUNT(*) AS Transaction_Count
FROM RETAIL.SALES.RETAIL_SALES
GROUP BY Product_Category;
```

Results (just now)

Table | Chart — 3 rows — 79ms

	PRODUCT_CATEGORY	TRANSACTION_COUNT
1	Clothing	351
2	Beauty	307
3	Electronics	342

Question 16

```sql
-- Q16. Find the total revenue (Total Amount) per gender.
SELECT
    Gender,
    SUM(Total_Amount) AS Total_Revenue
FROM RETAIL.SALES.RETAIL_SALES
GROUP BY Gender;
```

Results (just now)

Table | Chart — 2 rows — 89ms

	GENDER	TOTAL_REVENUE
1	Male	223160
2	Female	232840

Question 17

```sql
-- Q17. Find the average Price per Unit per product category.
SELECT
    Product_Category,
    AVG(Price_per_Unit) AS Average_Price
FROM RETAIL.SALES.RETAIL_SALES
GROUP BY Product_Category;
```

Results (just now)

Table | Chart — 3 rows — 77ms

	PRODUCT_CATEGORY	AVERAGE_PRICE
1	Beauty	184.055375
2	Clothing	174.287749
3	Electronics	181.900585

Question 18

```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.RETAIL_SALES
GROUP BY Product_Category
HAVING SUM(Total_Amount) > 10000;
```

Results (just now)

Table | Chart — 3 rows — 88ms

	PRODUCT_CATEGORY	TOTAL_REVENUE
1	Beauty	143515
2	Clothing	155580
3	Electronics	156905

Question 19

```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.RETAIL_SALES
GROUP BY Product_Category
HAVING AVG(Quantity) > 2;
```

Results (just now)

Table | Chart — 3 rows — 78ms

	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'.
SELECT
    Transaction_ID,
    Total_Amount,
    CASE
        WHEN Total_Amount > 1000 THEN 'High'
        ELSE 'Low'
    END AS Spending_Level
FROM RETAIL.SALES.RETAIL_SALES;
```

Results (just now)

Table | Chart | 1,000 rows | 76ms

	TRANSACTION_ID	TOTAL_AMOUNT	SPENDING_LEVEL
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

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
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.RETAIL_SALES;
```

Results (1 minute ago)

Table | Chart | 1,000 rows | 76ms

	TRANSACTION_ID	TOTAL_AMOUNT	SPENDING_LEVEL
1	1	150	Low
2	2	1000	Low
3	3	30	Low
4	4	500	Low
5	5	100	Low
6	6	30	Low