

Lefa Makhurubetshi

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

Question1

1

2

3

4

5

6

7

8

9

10

11

12

13

--Q1. Display all columns for all transactions.

--Expected output: All columns

SELECT *

FROM RETAIL.SALES.RETAIL_SALES_DATASET;

Ctrl+I to generate

Results (1 minute ago)

Table

Chart

1,000 rows

23ms

	#	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	
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	

Question 2

7

8

9

10

11

12

13

14

15

16

17

18

19

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

--Expected output: Transaction ID, Date, Customer ID

SELECT transaction_id,

date,

customer_id

FROM RETAIL.SALES.RETAIL_SALES_DATASET;

Ctrl+I to generate

Results (4 minutes ago)

Table

Chart

	#	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	
9	9	2023-12-13	CUST009	
10	10	2023-10-07	CUST010	

Question 3

```
17 -----
18 --Q3. Display all the distinct product categories in the dataset.
19 --Expected output: Product Category
20
21 SELECT DISTINCT product_category
22 FROM RETAIL.SALES.RETAIL_SALES_DATASET;
23
24 -----
25
26
27
28
29
30
31
32
33
34
35
```

Results (2 minutes ago)

Table Chart

	PRODUCT_CATEGORY
1	Electronics
2	Clothing
3	Beauty

Question 4

```
25
26 --Q4. Display all the distinct gender values in the dataset.
27 --Expected output: Gender
28
29 SELECT DISTINCT gender
30 FROM RETAIL.SALES.RETAIL_SALES_DATASET;
31
32 -----
```

Results (2 minutes ago)

Table Chart

	GENDER
1	Male
2	Female

Question 5

32

33

34

35

36

37

38

39

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

--Expected output: All columns

SELECT *

FROM RETAIL.SALES.RETAIL_SALES_DATASET

WHERE age > 40;

Ctrl-I to generate

Results (2 minutes ago)

Table

Chart

534 rows

67ms

	#	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	
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

40

41

42

43

44

45

46

47

--Q6. Display all transactions where the Price per Unit is between 100 and 500.

--Expected output: All columns

SELECT *

FROM RETAIL.SALES.RETAIL_SALES_DATASET

WHERE price_per_unit BETWEEN 100 AND 500;

Results (2 minutes ago)

Table

Chart

396 rows

1.4s

	#	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	
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	

Question 7

47
48
49
50
51
52
53
54
55

```
--07. Display all transactions where the Product Category is either 'Beauty' or 'Electronics'
--Expected output: ALL columns

SELECT *
FROM RETAIL.SALES.RETAIL_SALES_DATASET
WHERE product_category IN ('Beauty','Electronics');
```

Results (2 minutes ago)

TableChart

649 rows70ms

	#	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	
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	

Question 8

55
56
57
58
59
60
61
62
63

```
--08. Display all transactions where the Product Category is not 'Clothing'.
--Expected output: ALL columns

SELECT *
FROM RETAIL.SALES.RETAIL_SALES_DATASET
WHERE product_category NOT IN ('Clothing');
```

Results (2 minutes ago)

TableChart

649 rows70ms

	#	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	
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	

Question 9

63
64
65
66
67
68
69
70
71

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

SELECT *
FROM RETAIL.SALES.RETAIL_SALES_DATASET
WHERE quantity >= 3;
```

Results (2 minutes ago)

TableChart

504 rows73ms

#	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
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

Question 10

71
72
73
74
75
76
77
78

```
--Q10. Count the total number of transactions.
--Expected output: Total_Transactions

SELECT COUNT(transaction_id) AS Total_Transaction
FROM RETAIL.SALES.RETAIL_SALES_DATASET;
```

Results (2 minutes ago)

TableChart

1 row21ms

#	TOTAL_TRANSACTION
1	1000

Question 11

78
79
80
81
82
83
84
85

```
--Q11. Find the average Age of customers.
--Expected output: Average_Age

SELECT AVG(age) AS Average_Age
FROM RETAIL.SALES.RETAIL_SALES_DATASET;
```

Results (2 minutes ago)

TableChart

1 row433ms

#	AVERAGE_AGE
1	41.392000

Question 12

```
85 -----
86 --Q12. Find the total quantity of products sold.
87 --Expected output: Total_Quantity
88
89 SELECT SUM(quantity) AS Total_Quantity
90 FROM RETAIL.SALES.RETAIL_SALES_DATASET;
91
92 -----
```

Results (2 minutes ago)

Table Chart 1 row 64ms

#	TOTAL_QUANTITY
1	2514

Question 13

```
92 -----
93 --Q13. Find the maximum Total Amount spent in a single transaction.
94 --Expected output: Max_Total_Amount
95
96 SELECT MAX(TOTAL_AMOUNT) AS Max_Total_Amount
97 FROM RETAIL.SALES.RETAIL_SALES_DATASET;
98
99 -----
```

Results (2 minutes ago)

Table Chart 1 row 34ms

#	MAX_TOTAL_AMOUNT
1	2000

Question 14

```
98 -----
99
100 --Q14. Find the minimum Price per Unit in the dataset.
101 --Expected output: Min_Price_per_Unit
102
103 SELECT MIN(price_per_unit) AS Min_Price_per_Unit
104 FROM RETAIL.SALES.RETAIL_SALES_DATASET;
105
106 -----
```

Results (2 minutes ago)

Table Chart 1 row 22ms

#	MIN_PRICE_PER_UNIT
1	25

Question 15

```
106 -----
107 --Q15. Find the number of transactions per Product Category.
108 --Expected output: Product Category, Transaction_Count
109
110 SELECT product_category,
111        COUNT(transaction_id) AS Transaction_Count
112 FROM RETAIL.SALES.RETAIL_SALES_DATASET
113 GROUP BY product_category;
114
115
116
117
118
119
```

Results (2 minutes ago)

PRODUCT_CATEGORY	TRANSACTION_COUNT
Clothing	351
Beauty	307
Electronics	342

Question 16

```
115 -----
116 --Q16. Find the total revenue (Total Amount) per gender.
117 --Expected output: Gender, Total_Revenue
118
119 SELECT gender,
120        SUM(total_amount) AS Total_Revenue
121 FROM RETAIL.SALES.RETAIL_SALES_DATASET
122 GROUP BY gender;
123
124
125
126
127
128
```

Results (2 minutes ago)

GENDER	TOTAL_REVENUE
Male	223160
Female	232840

Question 17

```
124 -----
125 --Q17. Find the average Price per Unit per product category.
126 --Expected output: Product Category, Average_Price
127
128 SELECT product_category,
129        AVG(price_per_unit) AS Average_price
130 FROM RETAIL.SALES.RETAIL_SALES_DATASET
131 GROUP BY product_category;
132 Ctrl-I to generate
133 -----
134 --Q18. Find the total revenue per product category where total revenue is greater than 10,000.
135 --Expected output: Product Category, Total_Revenue
136
137
```

Results (2 minutes ago)

PRODUCT_CATEGORY	AVERAGE_PRICE
Clothing	174.287749
Beauty	184.055375
Electronics	181.900585

Question 18

```
133 -----
134 --Q18. Find the total revenue per product category where total revenue is greater than 10,000.
135 --Expected output: Product Category, Total_Revenue
136
137 SELECT
138     product_category,
139     SUM(total_amount) AS Total_Revenue
140 FROM RETAIL_SALES.RETAIL_SALES_DATASET
141 GROUP BY product_category
142 HAVING SUM(total_amount) > 10000;
143 -----
144
145 ***
```

Results (2 minutes ago)

PRODUCT_CATEGORY	TOTAL_REVENUE
Clothing	155580
Beauty	143515
Electronics	156905

Question 19

```
144 -----
145 --Q19. Find the average quantity per product category where the average is more than 2.
146 --Expected output: Product Category, Average_Quantity
147
148 SELECT
149     product_category,
150     AVG(quantity) AS Average_quantity
151 FROM RETAIL_SALES.RETAIL_SALES_DATASET
152 GROUP BY product_category
153 HAVING AVG(quantity) > 2;
154
155 Ctrl+I to generate
156
157 ***
```

Results (2 minutes ago)

PRODUCT_CATEGORY	AVERAGE_QUANTITY
Clothing	2.547009
Beauty	2.511401
Electronics	2.482456

Question 20

```
155 -----
156 --Q20. Display a column called Spending_Level that shows 'High' if Total Amount > 1000, otherwise 'Low'.
157 --Expected output: Transaction ID, Total Amount, Spending_Level
158
159 SELECT transaction_id,
160         total_amount,
161         CASE
162             WHEN total_amount > 1000 THEN 'High'
163             ELSE 'Low'
164         END AS Spending_Level
165 FROM RETAIL_SALES.RETAIL_SALES_DATASET;
166 -----
167
168 ***
```

Results (2 minutes ago)

TRANSACTION_ID	TOTAL_AMOUNT	SPENDING_LEVEL
1	150	Low
2	1000	Low
3	30	Low
4	500	Low
5	100	Low
6	30	Low
7	50	Low
8	100	Low
9	600	Low
10	200	Low
11	100	Low
12	75	Low
13	1500	High
14	120	Low

Question 21

167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183

--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 >= 60 THEN 'Senior'
WHEN Age BETWEEN 30 AND 59 THEN 'Adult'
ELSE 'Youth'
END AS Age_Group
FROM RETAIL.SALES.RETAIL_SALES_DATASET;

Results (2 minutes ago)

TableChart

1,000 rows55ms

	CUSTOMER_ID	AGE	AGE_GROUP
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