

BrightLight Data Analytics Coding Practical

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

```
1 ---SELECT Statement
2 ---Q1. Display all columns for all transactions
3 SELECT *
4 FROM SALES.retailsales.salesretail;
5
```

↳ Results ⚡ Chart

#	TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	# PR
1	1	2023-11-24	CUST001	Male	34	Beauty	3	
2	2	2023-02-27	CUST002	Female	26	Clothing	2	
3	3	2023-01-13	CUST003	Male	50	Electronics	1	
4	4	2023-05-21	CUST004	Male	37	Clothing	1	
5	5	2023-05-06	CUST005	Male	30	Beauty	2	
6	6	2023-04-25	CUST006	Female	45	Beauty	1	
7	7	2023-03-13	CUST007	Male	46	Clothing	2	
8	8	2023-02-22	CUST008	Male	30	Electronics	4	
9	9	2023-12-13	CUST009	Male	63	Electronics	2	
10	10	2023-10-07	CUST010	Female	52	Clothing	4	
11	11	2023-02-14	CUST011	Male	23	Clothing	2	
12	12	2023-10-30	CUST012	Male	35	Beauty	3	
13	13	2023-08-05	CUST013	Male	22	Electronics	3	
14	14	2023-01-17	CUST014	Male	64	Clothing	4	
15	15	2023-01-16	CUST015	Female	42	Electronics	4	

```
6 ---Q2. Display only the Transaction ID, Date, and Customer ID for all records
7 SELECT Transaction_ID, Date, Customer_ID
8 FROM SALES.retailsales.salesretail;
9
```

↳ Results ⚡ Chart

#	TRANSACTION_ID	DATE	CUSTOMER_ID
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

```

10 ---SELECT DISTINCT Statement
11 ---Q3. Display all the distinct product categories in the dataset. Expected output: Product Category
12 SELECT DISTINCT Product_Category
13 FROM SALES.retailsales.salesretail;

```

↳ Results ⚡ Chart

A PRODUCT_CATEGORY

1	Clothing
2	Beauty
3	Electronics

```
15 ---Q4. Display all the distinct gender values in the dataset
```

```
16 SELECT DISTINCT gender
17 FROM SALES.retailsales.salesretail;
```

↳ Results ⚡ Chart

A GENDER

1	Male
2	Female

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

```
21 SELECT *
22 FROM SALES.retailsales.salesretail
23 WHERE Age > 40;
```

↳ Results ⚡ Chart

Q ⌂ ⌄ ⌁ ⌂ ⌃

# 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
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
9	19	2023-09-16	CUST019	Female	62	Clothing	2	25
0	21	2023-01-14	CUST021	Female	50	Beauty	1	500
1	24	2023-11-29	CUST024	Female	49	Clothing	1	300
2	25	2023-12-26	CUST025	Female	64	Beauty	1	50
3	28	2023-04-23	CUST028	Female	43	Beauty	1	500
4	29	2023-08-18	CUST029	Female	42	Electronics	1	30
5	31	2023-05-22	CUST031	Male	44	Electronics	4	200
6	32	2023-07-01	CUST032	Female	55	Beauty	1	100

```

25      ---Q6. Display all transactions where the Price per Unit is between 100 and 500.
26      | SELECT *
27      | FROM SALES.retailsales.salesretail
28      | WHERE Price_per_Unit BETWEEN 100 AND 500;
29

```

Results     

# TRANSACTION_ID	⌚ DATE	▲ CUSTOMER_ID	▲ GENDER	# AGE	▲ PRODUCT_CATEGORY	# QUANTITY	# PRICE_PER_UNIT	# TOTAL_AMOUNT
2	2023-02-27	CUST002	Female	26	Clothing	2	500	1000
4	2023-05-21	CUST004	Male	37	Clothing	1	500	500
9	2023-12-13	CUST009	Male	63	Electronics	2	300	600
13	2023-08-05	CUST013	Male	22	Electronics	3	500	1500
15	2023-01-16	CUST015	Female	42	Electronics	4	500	2000
16	2023-02-17	CUST016	Male	19	Clothing	3	500	1500
20	2023-11-05	CUST020	Male	22	Clothing	3	300	900
21	2023-01-14	CUST021	Female	50	Beauty	1	500	500
24	2023-11-29	CUST024	Female	49	Clothing	1	300	300
26	2023-10-07	CUST026	Female	28	Electronics	2	500	1000
28	2023-04-23	CUST028	Female	43	Beauty	1	500	500
30	2023-10-29	CUST030	Female	39	Beauty	3	300	900
31	2023-05-23	CUST031	Male	44	Electronics	4	300	1200
35	2023-08-05	CUST035	Female	58	Beauty	3	300	900
36	2023-06-24	CUST036	Male	52	Beauty	3	300	900

```

30      ---Q7. Display all transactions where the Product Category is either 'Beauty' or 'Electronics'.
31      | SELECT *
32      | FROM SALES.retailsales.salesretail
33      | WHERE Product_Category IN ('Beauty', 'Electronics');
34

```

↳ Results     

# TRANSACTION_ID	⌚ DATE	▲ CUSTOMER_ID	▲ GENDER	# AGE	▲ PRODUCT_CATEGORY	# QUANTITY	# PRICE_PER_UNIT	# TOTAL_AMOUNT
1	2023-11-24	CUST001	Male	34	Beauty	3	50	150
2	2023-01-13	CUST003	Male	50	Electronics	1	30	30
3	2023-05-06	CUST005	Male	30	Beauty	2	50	100
4	2023-04-25	CUST006	Female	45	Beauty	1	30	30
5	2023-02-22	CUST008	Male	30	Electronics	4	25	100
6	2023-12-13	CUST009	Male	63	Electronics	2	300	600
7	2023-10-30	CUST012	Male	35	Beauty	3	25	75
8	2023-08-05	CUST013	Male	22	Electronics	3	500	1500
9	2023-01-16	CUST015	Female	42	Electronics	4	500	2000
0	2023-04-30	CUST018	Female	47	Electronics	2	25	50
1	2023-01-14	CUST021	Female	50	Beauty	1	500	500
2	2023-12-26	CUST025	Female	64	Beauty	1	50	50
3	2023-10-07	CUST026	Female	28	Electronics	2	500	1000
4	2023-08-03	CUST027	Female	38	Beauty	2	25	50
5	2023-04-23	CUST028	Female	43	Beauty	1	500	500

```

35    ---Q8. Display all transactions where the Product Category is not 'Clothing'.
36    SELECT *
37    FROM SALES.retailsales.salesretail
38    WHERE Product_Category != 'Clothing';
39

```

Results ▾ Chart

# TRANSACTION_ID	⌚ DATE	▲ CUSTOMER_ID	▲ GENDER	# AGE	▲ PRODUCT_CATEGORY	# QUANTITY	# PRICE_PER_UNIT	# TOTAL_AMOUNT
1	2023-11-24	CUST001	Male	34	Beauty	3	50	150
3	2023-01-13	CUST003	Male	50	Electronics	1	30	30
5	2023-05-06	CUST005	Male	30	Beauty	2	50	100
6	2023-04-25	CUST006	Female	45	Beauty	1	30	30
8	2023-02-22	CUST008	Male	30	Electronics	4	25	100
9	2023-12-13	CUST009	Male	63	Electronics	2	300	600
12	2023-10-30	CUST012	Male	35	Beauty	3	25	75
13	2023-08-05	CUST013	Male	22	Electronics	3	500	1500
15	2023-01-16	CUST015	Female	42	Electronics	4	500	2000
18	2023-04-30	CUST018	Female	47	Electronics	2	25	50
21	2023-01-14	CUST021	Female	50	Beauty	1	500	500
25	2023-12-26	CUST025	Female	64	Beauty	1	50	50
26	2023-10-07	CUST026	Female	28	Electronics	2	500	1000
27	2023-08-03	CUST027	Female	38	Beauty	2	25	50
28	2023-04-23	CUST028	Female	43	Beauty	1	500	500

```

40    ---Q9. Display all transactions where the Quantity is greater than or equal to 3.
41    SELECT *
42    FROM SALES.retailsales.salesretail
43    WHERE Quantity >= 3;

```

Results ▾ Chart

# TRANSACTION_ID	⌚ DATE	▲ CUSTOMER_ID	▲ GENDER	# AGE	▲ PRODUCT_CATEGORY	# QUANTITY	# PRICE_PER_UNIT	# TOTAL_AMOUNT
1	2023-11-24	CUST001	Male	34	Beauty	3	50	150
8	2023-02-22	CUST008	Male	30	Electronics	4	25	100
10	2023-10-07	CUST010	Female	52	Clothing	4	50	200
12	2023-10-30	CUST012	Male	35	Beauty	3	25	75
13	2023-08-05	CUST013	Male	22	Electronics	3	500	1500
14	2023-01-17	CUST014	Male	64	Clothing	4	30	120
15	2023-01-16	CUST015	Female	42	Electronics	4	500	2000
16	2023-02-17	CUST016	Male	19	Clothing	3	500	1500
17	2023-04-22	CUST017	Female	27	Clothing	4	25	100
20	2023-11-05	CUST020	Male	22	Clothing	3	300	900
23	2023-04-12	CUST023	Female	35	Clothing	4	30	120
30	2023-10-29	CUST030	Female	39	Beauty	3	300	900
31	2023-05-23	CUST031	Male	44	Electronics	4	300	1200
32	2023-01-04	CUST032	Male	30	Beauty	3	30	90
34	2023-12-24	CUST034	Female	51	Clothing	3	50	150

```

45    ---4. Aggregate Functions
46    ---Q10. Count the total number of transactions.
47    SELECT COUNT(*) AS Total_Transactions
48    FROM SALES.retailsales.salesretail;
49

```

Results ▾ Chart

TOTAL_TRANSACTIONS
1000

```

50    ---Q11. Find the average Age of customers.
51    SELECT AVG(Age) AS Average_Age
52    FROM SALES.retailsales.salesretail;
53

```

Results ▾ Chart

AVERAGE_AGE
41.392000

54	--Q12. Find the total quantity of products sold.
55	SELECT SUM(Quantity) AS Total_Quantity
56	FROM SALES.retailsales.salesretail;
57	
58	--Q13. Find the maximum Total Amount spent in a single transaction.
59	SELECT MAX(Total_Amount) AS Max_Total_Amount
60	FROM SALES.retailsales.salesretail;
61	
	Results
	Chart
	# TOTAL_QUANTITY
	2514
58	--Q13. Find the maximum Total Amount spent in a single transaction.
59	SELECT MAX(Total_Amount) AS Max_Total_Amount
60	FROM SALES.retailsales.salesretail;
61	
	Results
	Chart
	# MAX_TOTAL_AMOUNT
	2000
62	--Q14. Find the minimum Price per Unit in the dataset.
63	SELECT MIN(Price_per_Unit) AS Min_Price_per_Unit
64	FROM SALES.retailsales.salesretail;
65	
	Results
	Chart
	# MIN_PRICE_PER_UNIT
	25
66	--5. GROUP BY Statement
67	--Q15. Find the number of transactions per Product Category.
68	SELECT Product_Category, COUNT(*) AS Transaction_Count
69	FROM SALES.retailsales.salesretail
70	GROUP BY Product_Category;
71	
	Results
	Chart
	A PRODUCT_CATEGORY
	# TRANSACTION_COUNT
	Clothing
	351
	Beauty
	307
	Electronics
	342
72	--Q16. Find the total revenue (Total Amount) per gender.
73	SELECT Gender, SUM(Total_Amount) AS Total_Revenue
74	FROM SALES.retailsales.salesretail
75	GROUP BY Gender;
76	
	Results
	Chart
	A GENDER
	# TOTAL_REVENUE
	Male
	223160
	Female
	232840
77	--Q17. Find the average Price per Unit per product category.
78	SELECT Product_Category, AVG(Price_per_Unit) AS Average_Price
79	FROM SALES.retailsales.salesretail
80	GROUP BY Product_Category;
81	
	Results
	Chart
	A PRODUCT_CATEGORY
	# AVERAGE_PRICE
	Beauty
	184.055375
	Clothing
	174.287749
	Electronics
	181.900585

```

82    ---6. HAVING Clause
83    ---Q18. Find the total revenue per product category where total revenue is greater than 10,000.
84    | SELECT Product_Category, SUM(Total_Amount) AS Total_Revenue
85    | FROM SALES.retailsales.salesretail
86    | GROUP BY Product_Category
87    | HAVING SUM(Total_Amount) > 10000;

```

↳ Results ↵ Chart

🔍 ⌂ ⌄ ⌁ ⌂

▲ PRODUCT_CATEGORY	# TOTAL_REVENUE
Beauty	143515
Clothing	155580
Electronics	156905

```

89    ---Q19. Find the average quantity per product category where the average is more than 2. |
90    | SELECT Product_Category, AVG(Quantity) AS Average_Quantity
91    | FROM SALES.retailsales.salesretail
92    | GROUP BY Product_Category
93    | HAVING AVG(Quantity) > 2;
94
95    ---7. CASE Statement

```

↳ Results ↵ Chart

🔍 ⌂ ⌄ ⌁ ⌂

▲ PRODUCT_CATEGORY	# AVERAGE_QUANTITY
Beauty	2.511401
Clothing	2.547009
Electronics	2.482456

```

96    ---Q20. Display a column called Spending_Level that shows 'High' if Total Amount > 1000, otherwise 'Low'.
97    | SELECT Transaction_ID, Total_Amount,
98    | CASE
99    |     WHEN Total_Amount > 1000 THEN 'High'
100   |     ELSE 'Low'
101   | END AS Spending_Level
102   | FROM SALES.retailsales.salesretail;
103

```

↳ Results ↵ Chart

# 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
8	8	100 Low
9	9	600 Low
0	10	200 Low
1	11	100 Low
2	12	75 Low
3	13	1500 High
4	14	120 Low

```
104 ---Q21. Display a new column called Age_Group that labels customers as:'Youth' if Age < 30,'Adult' if Age is between 30 and  
105 59,'Senior' if Age >= 60  
106 SELECT Customer_ID, Age,  
107 CASE  
108     WHEN Age < 30 THEN 'Youth'  
109     WHEN Age BETWEEN 30 AND 59 THEN 'Adult'  
110     ELSE 'Senior'  
111 END AS Age_Group  
112 FROM SALES.retailsales.salesretail;
```

↳ Results ↳ Chart

	A CUSTOMER_ID	# AGE	A 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