

BrightLight Data Analytics Coding Practical

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

The following questions are designed to help you build a strong foundation in basic SQL syntax. You are provided with a dataset named **retail_sales_dataset.csv**. Upload this dataset to your Snowflake account and use it to answer the questions below.

Please follow the instructions below carefully:

1. Write one SQL query per question.
Use proper formatting and indentation where necessary.
2. Each question tests a specific SQL concept.
Read the question carefully and apply only the concept being tested (e.g., SELECT, WHERE, GROUP BY, etc.).
3. Do not combine multiple SQL concepts unless instructed to.
For example, if the question asks for a SELECT DISTINCT, don't use WHERE unless specified.
4. Use the correct column names as provided in the dataset
5. Expected Output Columns are provided for each question.
Your query must return exactly those columns in the result.
6. Don't worry about the actual data.
Focus on getting the SQL syntax right.
7. Submit your completed SQL queries as a .sql

Table 1: Outlines the name and descriptions of the columns in the provided dataset
"retail_sales_dataset.csv"

Column Name	Description
Transaction_ID	Unique identifier for each transaction.
Date	The date on which the transaction occurred (format: YYYY-MM-DD).
Customer_ID	Unique identifier for the customer making the purchase.
Gender	Gender of the customer (e.g., Male, Female).
Age	Age of the customer at the time of the transaction.
Product_Category	Category of the product purchased (e.g., Beauty, Clothing, Electronics).
Quantity	Number of product units purchased in the transaction.
Price_per_Unit	Cost of a single unit of the product (in the currency used).
Total_Amount	Total amount spent for the transaction (Quantity × Price per Unit).

Questions

1. SELECT Statement

Q1. Display all columns for all transactions.

Expected output: All columns

```
1 SELECT *
2 FROM `brightlighttutorials.Practical1.Retail_Sales`
```

Press Alt+F1 for accessibility options.

Query results

Save results Open in ▾

Job information		Results		Chart	JSON	Execution details		Execution graph	
Row		Transaction ID	Date		Customer ID	Gender	Age	Product Category	Quantity
1		191	2023-10-18		CUST191	Male	64	Beauty	
2		204	2023-09-28		CUST204	Male	39	Beauty	
3		230	2023-04-23		CUST230	Male	54	Beauty	
4		232	2023-02-06		CUST232	Female	43	Beauty	
5		309	2023-12-23		CUST309	Female	26	Beauty	
6		310	2023-10-12		CUST310	Female	28	Beauty	
7		363	2023-06-03		CUST363	Male	64	Beauty	
8		371	2023-02-21		CUST371	Female	20	Beauty	
9		397	2023-03-10		CUST397	Female	30	Beauty	

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

Expected output: Transaction ID, Date, Customer ID

```
1 SELECT Transaction_ID, Date, Customer_ID
2 FROM `brightlighttutorials.Practical1.Retail_Sales`
```

Press Alt+F1 for accessibility options.

Query results

Save results Open in ▾

Job information		Results		Chart	JSON	Execution details		Execution graph	
Row		Transaction_ID	Date		Customer_ID				
1		191	2023-10-18		CUST191				
2		204	2023-09-28		CUST204				
3		230	2023-04-23		CUST230				
4		232	2023-02-06		CUST232				
5		309	2023-12-23		CUST309				
6		310	2023-10-12		CUST310				
7		363	2023-06-03		CUST363				
8		371	2023-02-21		CUST371				
9		397	2023-03-10		CUST397				
10		454	2023-02-22		CUST454				

2. SELECT DISTINCT Statement

Q3. Display all the distinct product categories in the dataset.

Expected output: Product Category

```
1 SELECT DISTINCT(Product_Category)
2 FROM `brightlighttutorials.Practical1.Retail_Sales`
```

Press Alt+F1 for accessibility options.

Query results

[Save results](#) [Open in](#)

Job information	Results	Chart	JSON	Execution details	Execution graph
Row	Product_Category				
1	Beauty				
2	Clothing				
3	Electronics				

Q4. Display all the distinct gender values in the dataset.

Expected output: Gender

```
1 SELECT DISTINCT(Gender)
2 FROM `brightlighttutorials.Practical1.Retail_Sales`
```

Press Alt+F1 for accessibility options.

Query results

[Save results](#) [Open in](#)

Job information	Results	Chart	JSON	Execution details	Execution graph
Row	Gender				
1	Male				
2	Female				

3. WHERE Clause

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

Expected output: All columns

```
1 SELECT *
2 FROM `brightlighttutorials.Practical1.Retail_Sales`
3 WHERE Age > 40
```

Press Alt+F1 for accessibility options.

Query results

[Save results](#) [Open in](#)

Job information	Results	Chart	JSON	Execution details	Execution graph
Row	Transaction_ID	Date	Customer_ID	Gender	Age
1	191	2023-10-18	CUST191	Male	64
2	230	2023-04-23	CUST230	Male	54
3	232	2023-02-06	CUST232	Female	43
4	363	2023-06-03	CUST363	Male	64
5	454	2023-02-22	CUST454	Female	46
6	512	2023-11-07	CUST512	Female	57
7	791	2023-12-05	CUST791	Female	51
8	825	2023-08-26	CUST825	Female	46
9	855	2023-09-01	CUST855	Male	54

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

Expected output: All columns

```

1 SELECT *
2 FROM `brightlighttutorials.Practical1.Retail_Sales`
3 WHERE Price_per_Unit BETWEEN 100 AND 500

```

Press Alt+F1 for accessibility options.

Query results

[Save results](#)

[Open in](#)

Row	Transaction_ID	Date	Customer_ID	Gender	Age	Product_Category	Quantity
1	52	2023-03-05	CUST052	Female	36	Beauty	
2	79	2023-04-18	CUST079	Male	34	Beauty	
3	174	2023-04-12	CUST174	Female	39	Beauty	
4	240	2023-02-06	CUST240	Female	23	Beauty	
5	358	2023-05-16	CUST358	Female	32	Beauty	
6	378	2023-06-28	CUST378	Male	50	Beauty	
7	555	2023-10-19	CUST555	Male	25	Beauty	
8	794	2023-09-17	CUST794	Female	60	Beauty	
9	905	2023-04-02	CUST905	Male	58	Beauty	

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

Expected output: All columns

```

1 SELECT *
2 FROM `brightlighttutorials.Practical1.Retail_Sales`
3 WHERE Product_Category IN('Beauty','Electronics')

```

Press Alt+F1 for accessibility options.

Query results

[Save results](#)

[Open in](#)

Row	Transaction_ID	Date	Customer_ID	Gender	Age	Product_Category	Quantity
1	191	2023-10-18	CUST191	Male	64	Beauty	
2	204	2023-09-28	CUST204	Male	39	Beauty	
3	230	2023-04-23	CUST230	Male	54	Beauty	
4	232	2023-02-06	CUST232	Female	43	Beauty	
5	309	2023-12-23	CUST309	Female	26	Beauty	
6	310	2023-10-12	CUST310	Female	28	Beauty	
7	363	2023-06-03	CUST363	Male	64	Beauty	
8	371	2023-02-21	CUST371	Female	20	Beauty	
9	397	2023-03-10	CUST397	Female	30	Beauty	

Q8. Display all transactions where the Product Category is **not** 'Clothing'.

Expected output: All columns

```

1 SELECT *
2 FROM `brightlighttutorials.Practical1.Retail_Sales`
3 WHERE Product_Category != 'Clothing'

```

Press Alt+F1 for accessibility options.

Query results

[Save results](#)

[Open in](#)

Row	Transaction_ID	Date	Customer_ID	Gender	Age	Product_Category	Quantity
1	191	2023-10-18	CUST191	Male	64	Beauty	
2	204	2023-09-28	CUST204	Male	39	Beauty	
3	230	2023-04-23	CUST230	Male	54	Beauty	
4	232	2023-02-06	CUST232	Female	43	Beauty	
5	309	2023-12-23	CUST309	Female	26	Beauty	
6	310	2023-10-12	CUST310	Female	28	Beauty	
7	363	2023-06-03	CUST363	Male	64	Beauty	
8	371	2023-02-21	CUST371	Female	20	Beauty	
9	397	2023-03-10	CUST397	Female	30	Beauty	

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

Expected output: All columns

```
1 SELECT *
2 FROM `brightlighttutorials.Practical1.Retail_Sales`
3 WHERE Quantity > 3
```

Press Alt+F1 for accessibility options.

Query results

[Save results](#) [Open in](#)

Row	Customer_ID	Gender	Age	Product_Category	Quantity	Price_per_Unit
1	CUST071	Female	51	Beauty	4	25
2	CUST114	Female	22	Beauty	4	25
3	CUST225	Female	57	Beauty	4	25
4	CUST276	Female	21	Beauty	4	25
5	CUST311	Female	32	Beauty	4	25
6	CUST406	Female	22	Beauty	4	25
7	CUST414	Male	48	Beauty	4	25
8	CUST492	Male	61	Beauty	4	25
9	CUST500	Female	60	Beauty	4	25

4. Aggregate Functions

Q10. Count the total number of transactions.

Expected output: Total_Transactions

```
1 SELECT COUNT(*) AS Total_Transactions
2 FROM `brightlighttutorials.Practical1.Retail_Sales`
```

Press Alt+F1 for accessibility options.

Query results

[Save results](#) [Open in](#)

Row	Total_Transactions
1	1000

Q11. Find the average Age of customers.

Expected output: Average_Age

```
1 SELECT AVG(Age) AS Average_Age
2 FROM `brightlighttutorials.Practical1.Retail_Sales`
```

Press Alt+F1 for accessibility options.

Query results

[Save results](#) [Open in](#)

Row	Average_Age
1	41.39199999999999

Q12. Find the total quantity of products sold.

Expected output: Total_Quantity

```
1 SELECT SUM(Quantity) As Total_Quantity
2 FROM `brightlighttutorials.Practical1.Retail_Sales`
```

Press Alt+F1 for accessibility options.

Query results

Job information **Results** Chart JSON Execution details Execution graph

Row	Total_Quantity
1	2514

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

Expected output: Max_Total_Amount

```
1 SELECT MAX(Total_Amount) As Max_Total_Amount
2 FROM `brightlighttutorials.Practical1.Retail_Sales`
```

Press Alt+F1 for accessibility options.

Query results

Job information **Results** Chart JSON Execution details Execution graph

Row	Max_Total_Amount
1	2000

Q14. Find the minimum Price per Unit in the dataset.

Expected output: Min_Price_per_Unit

```
1 SELECT MIN(Price_per_Unit) As Min_Price_per_Unit
2 FROM `brightlighttutorials.Practical1.Retail_Sales`
```

Press Alt+F1 for accessibility options.

Query results

Job information **Results** Chart JSON Execution details Execution graph

Row	Min_Price_per_Unit
1	25

5. GROUP BY Statement

Q15. Find the number of transactions per Product Category.

Expected output: Product_Catgegory, Transaction_Count

```
1 SELECT Product_Catgegory,COUNT(Transaction_ID) AS Transaction_Count
2 FROM `brightlighttutorials.Practical1.Retail_Sales`
3 GROUP BY Product_Catgegory
```

Press Alt+F1 for accessibility options.

Query results

Job information **Results** Chart JSON Execution details Execution graph

Row	Product_Catgegory	Transaction_Count
1	Beauty	307
2	Clothing	351
3	Electronics	342

Q16. Find the total revenue (Total_Amount) per gender.

Expected output: Gender, Total_Revenue

```
1 SELECT Gender, SUM(Total_Amount) AS Total_Revenue
2 FROM `brightlighttutorials.Practical1.Retail_Sales`
3 GROUP BY Gender
```

Press Alt+F1 for accessibility options.

Query results

Job information **Results** Chart JSON Execution details Execution graph

Row	Gender	Total_Revenue
1	Male	223160
2	Female	232840

Q17. Find the average Price per Unit per product category.

Expected output: Product Category, Average_Price

```
1 SELECT Product_Category, AVG(Price_per_Unit) AS Average_Price
2 FROM `brightlighttutorials.Practical1.Retail_Sales`
3 GROUP BY Product_Category
```

Press Alt+F1 for accessibility options.

Query results

Job information **Results** Chart JSON Execution details Execution graph

Row	Product_Category	Average_Price
1	Beauty	184.0553745928...
2	Clothing	174.2877492877...
3	Electronics	181.9005847953...

6. HAVING Clause

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

Expected output: Product Category, Total_Revenue

```
1 SELECT Product_Category, SUM(Total_Amount) AS Total_Revenue
2 FROM `brightlighttutorials.Practical1.Retail_Sales`
3 GROUP BY Product_Category
4 HAVING SUM(Total_Amount) > 10000
```

Press Alt+F1 for accessibility options.

Query results

Job information **Results** Chart JSON Execution details Execution graph

Row	Product_Category	Total_Revenue
1	Beauty	143515
2	Clothing	155580
3	Electronics	156905

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

Expected output: Product Category, Average_Quantity

```
1 SELECT Product_Category, AVG(Quantity) AS Average_Quantity
2 FROM `brightlighttutorials.Practical1.Retail_Sales`
3 GROUP BY Product_Category
4 HAVING AVG(Quantity) > 2
```

Press Alt+F1 for accessibility options.

Query results

[Save results](#)

[Open in](#)

Job information		Results	Chart	JSON	Execution details	Execution graph
Row	Product_Category	Average_Quantity				
1	Beauty	2.511400651465...				
2	Clothing	2.547008547008...				
3	Electronics	2.482456140350...				

7. CASE Statement

Q20. Display a column called Spending_Level that shows 'High' if Total Amount > 1000, otherwise 'Low'.

Expected output: Transaction ID, Total Amount, Spending_Level

```
1 SELECT Transaction_ID, Total_Amount,
2      CASE
3          WHEN Total_Amount > 1000 THEN 'High'
4          ELSE 'Low'
5      END AS Spending_Level
6 FROM `brightlighttutorials.Practical1.Retail_Sales`
```

Press Alt+F1 for accessibility options.

Query results

[Save results](#)

[Open in](#)

Job information		Results	Chart	JSON	Execution details	Execution graph
Row	Transaction_ID	Total_Amount	Spending_Level			
1	191	25	Low			
2	204	25	Low			
3	230	25	Low			
4	232	25	Low			
5	309	25	Low			
6	310	25	Low			
7	363	25	Low			

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

```
1  SELECT Customer_ID, Age,
2      CASE
3          WHEN Age < 30 THEN 'Youth'
4          WHEN Age BETWEEN 30 AND 59 THEN 'Adult'
5          ELSE 'Senior'
6      END AS Age_Group
7  FROM `brightlighttutorials.Practical1.Retail_Sales`
```

Press Alt+F1 for accessibility

Query results

[Save results](#) [Open in](#)

Row	Customer_ID	Age	Age_Group
1	CUST191	64	Senior
2	CUST204	39	Adult
3	CUST230	54	Adult
4	CUST232	43	Adult
5	CUST309	26	Youth
6	CUST310	28	Youth
7	CUST363	64	Senior