

# BrightLearn Practical 1

NAME: Lubabalo Sesonasipho Sitama

DATE: 20/10/2025

---

## QUESTION 1:

PRACTICAL1.RETAIL\_SALES Settings

```
1 -----
2 -- Q1. Display all columns for all transactions.
3 Expected output: All columns
4 select *
5 FROM practical1.retail_sales.sales;
```

Results Chart

	# TRANSACTION_ID	🕒 DATE	🔍 CUSTOMER_ID	🔍 GENDER	# AGE	🔍 PRODUC
1	1	2023-11-24	CUST001	Male	34	Beauty
2	2	2023-02-27	CUST002	Female	26	Clothing
3	3	2023-01-13	CUST003	Male	50	Electronics
4	4	2023-05-21	CUST004	Male	37	Clothing
5	5	2023-05-06	CUST005	Male	30	Beauty
6	6	2023-04-25	CUST006	Female	45	Beauty
7	7	2023-03-13	CUST007	Male	46	Clothing

## Question 2:

PRACTICAL1.RETAIL\_SALES Settings

```
0 -----
7 -- Q2. Display only the Transaction ID, Date, and Customer ID for all records.
8 Expected output: Transaction ID, Date, Customer ID
9 select Transaction_ID,
10         Date,
11         Customer_ID
12 FROM practical1.retail_sales.sales;
```

Results 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

Query  
Query  
Rows  
Query  
Sho

### Question 3:

PRACTICAL1.RETAIL\_SALES Settings

```
10 select Transaction_ID,  
11       Date,  
12       Customer_ID  
13 FROM practical1.retail_sales.sales;  
14 -----  
15 -- Q3. Display all the distinct product categories in the dataset.  
16 Expected output: Product Category  
17 select DISTINCT Product_Category  
18 FROM practical1.retail_sales.sales;
```

Results Chart

	PRODUCT_CATEGORY
1	Clothing
2	Beauty
3	Electronics

### Question 4:

PRACTICAL1.RETAIL\_SALES Settings

```
15 -- Q3. Display all the distinct product categories in the dataset.  
16 Expected output: Product Category  
17 select DISTINCT Product_Category  
18 FROM practical1.retail_sales.sales;  
19 -----  
20 -- Q4. Display all the distinct gender values in the dataset.  
21 Expected output: Gender  
22 select DISTINCT Gender  
23 FROM practical1.retail_sales.sales;
```

Results Chart

	GENDER
1	Male
2	Female

### Question 5:

PRACTICAL1.RETAIL\_SALES Settings

```
20 Expected output: Gender  
21 select DISTINCT Gender  
22 FROM practical1.retail_sales.sales;  
23 -----  
24 -- Q5. Display all transactions where the Age is greater than 40.  
25 -- Expected output: All columns  
26 select *  
27 FROM practical1.retail_sales.sales  
28 WHERE Age > 40;
```

Results Chart

	# TRANSACTION_ID	🕒 DATE	🔍 CUSTOMER_ID	🔍 GENDER	# AGE	🔍 PRODUCT_CATEGORY
1	3	2023-01-13	CUST003	Male	50	Electronics
2	6	2023-04-25	CUST006	Female	45	Beauty
3	7	2023-03-13	CUST007	Male	46	Clothing
4	9	2023-12-13	CUST009	Male	63	Electronics
5	10	2023-10-07	CUST010	Female	52	Clothing

### Question 6:

PRACTICAL1.RETAIL\_SALES Settings

```
-- Q6. Display all transactions where the Price per Unit is between 100 and 500.
-- Expected output: All columns
select *
FROM practical1.retail_sales.sales
Where PRICE_PER_UNIT between 100 AND 500;
```

Results Chart

	# TRANSACTION_ID	🕒 DATE	🔍 CUSTOMER_ID	🔍 GENDER	# AGE	🔍 PRODUCT
1	2	2023-02-27	CUST002	Female	26	Clothing
2	4	2023-05-21	CUST004	Male	37	Clothing
3	9	2023-12-13	CUST009	Male	63	Electronics
4	13	2023-08-05	CUST013	Male	22	Electronics
5	15	2023-01-16	CUST015	Female	42	Electronics

Query De  
Query du  
Rows  
Query ID  
Show

### Question 7:

PRACTICAL1.RETAIL\_SALES Settings

```
-- Q7. Display all transactions where the Product Category is either 'Beauty' or
-- 'Electronics'.
-- Expected output: All columns
select *
FROM practical1.retail_sales.sales
WHERE PRODUCT_CATEGORY = 'Beauty' OR PRODUCT_CATEGORY = 'Electronics';
```

Results Chart

	# TRANSACTION_ID	🕒 DATE	🔍 CUSTOMER_ID	🔍 GENDER	# AGE	🔍 PRODUCT
1	1	2023-11-24	CUST001	Male	34	Beauty
2	3	2023-01-13	CUST003	Male	50	Electronics
3	5	2023-05-06	CUST005	Male	30	Beauty
4	6	2023-04-25	CUST006	Female	45	Beauty
5	8	2023-02-22	CUST008	Male	30	Electronics

Query De  
Query dur  
Rows  
Query ID  
Show n

### Question 8:

PRACTICAL1.RETAIL\_SALES Settings

```
40 select
41 FROM practical1.retail_sales.sales
42 WHERE PRODUCT_CATEGORY = 'Beauty' OR PRODUCT_CATEGORY = 'Electronics';
43 -----
44 -- Q8. Display all transactions where the Product Category is not 'Clothing'.
45 -- Expected output: All columns
46 select *
47 FROM practical1.retail_sales.sales
48 WHERE PRODUCT_CATEGORY != 'Clothing';
```

Results Chart

	# TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT
1	1	2023-11-24	CUST001	Male	34	Beauty
2	3	2023-01-13	CUST003	Male	50	Electronics
3	5	2023-05-06	CUST005	Male	30	Beauty
4	6	2023-04-25	CUST006	Female	45	Beauty
5	8	2023-02-22	CUST008	Male	30	Electronics

### Question 9:

PRACTICAL1.RETAIL\_SALES Settings

```
46 FROM practical1.retail_sales.sales
47 WHERE PRODUCT_CATEGORY != 'Clothing';
48 -----
49 -- Q9. Display all transactions where the Quantity is greater than or equal to 3.
50 -- Expected output: All columns
51 select *
52 FROM practical1.retail_sales.sales
53 WHERE QUANTITY >= 3;
```

Results Chart

	# TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT
1	1	2023-11-24	CUST001	Male	34	Beauty
2	8	2023-02-22	CUST008	Male	30	Electronics
3	10	2023-10-07	CUST010	Female	52	Clothing
4	12	2023-10-30	CUST012	Male	35	Beauty
5	13	2023-08-05	CUST013	Male	22	Electronics

### Question 10:

PRACTICAL1.RETAIL\_SALES Settings

```
53 WHERE QUANTITY >= 3;
54 -----
55 -- Q10. Count the total number of transactions.
56 -- Expected output: Total_Transactions
57 select COUNT (TRANSACTION_ID) AS Total_Transactions
58 FROM practical1.retail_sales.sales;
```

Results Chart

	# TOTAL_TRANSACTIONS
1	1000

### Question 11:

PRACTICAL1.RETAIL\_SALES ▾ Settings ▾

```
56 -- Expected output: Total_Transactions
57 select COUNT (TRANSACTION_ID) AS Total_Transactions
58 FROM practical1.retail_sales.sales;
59 -----
60 -- Q11. Find the average Age of customers.
61 -- Expected output: Average_Age
62 select AVG (AGE) AS Average_Age
63 FROM practical1.retail_sales.sales;|
64
```

↩ Results Chart

	# AVERAGE_AGE
1	41.392000

### Question 12:

PRACTICAL1.RETAIL\_SALES ▾ Settings ▾

```
60 -- Q11. Find the average Age of customers.
61 -- Expected output: Average_Age
62 select AVG (AGE) AS Average_Age
63 FROM practical1.retail_sales.sales;
64 -----
65 -- Q12. Find the total quantity of products sold.
66 -- Expected output: Total_Quantity
67 select SUM (QUANTITY) AS Total_Quantity
68 FROM practical1.retail_sales.sales;|
69
```

↩ Results Chart

	# TOTAL_QUANTITY
1	2514

### Question 13:

PRACTICAL1.RETAIL_SALES		Settings
<pre>-- Q12. Find the total quantity of products sold. 66 -- Expected output: Total_Quantity 67 select SUM (QUANTITY) AS Total_Quantity 68 FROM practical1.retail_sales.sales; 69 ----- 70 -- Q13. Find the maximum Total Amount spent in a single transaction. 71 -- Expected output: Max_Total_Amount 72 select max (TOTAL_AMOUNT) AS Max_Total_Amount 73 FROM practical1.retail_sales.sales;</pre>		
Results Chart		
# MAX_TOTAL_AMOUNT		
1		2000

### Question 14:

PRACTICAL1.RETAIL_SALES		Settings
<pre>-- Expected output: Max_Total_Amount 72 select max (TOTAL_AMOUNT) AS Max_Total_Amount 73 FROM practical1.retail_sales.sales; 74 ----- 75 -- Q14. Find the minimum Price per Unit in the dataset. 76 -- Expected output: Min_Price_per_Unit 77 select MIN (PRICE_PER_UNIT) AS Min_Price_Per_Unit 78 FROM practical1.retail_sales.sales; 79</pre>		
Results Chart		
# MIN_PRICE_PER_UNIT		
1		25

### Question 15:

PRACTICAL1.RETAIL_SALES		Settings
<pre>-- 78 select MIN (PRICE_PER_UNIT) AS Min_Price_Per_Unit 79 FROM practical1.retail_sales.sales; 80 ----- 81 -- Q15. Find the number of transactions per Product Category. 82 -- Expected output: Product Category, Transaction_Count 82 select Product_Category, 83        COUNT (Transaction_ID) AS Transaction_Count 84 FROM practical1.retail_sales.sales 85 Group BY PRODUCT_CATEGORY;</pre>		
Results Chart		
	PRODUCT_CATEGORY	# TRANSACTION_COUNT
1	Beauty	307
2	Clothing	351
3	Electronics	342

### Question 16:

PRACTICAL1.RETAIL\_SALES ▾ Settings ▾

```
85 Group BY PRODUCT_CATEGORY;
86 -----
87 -- Q16. Find the total revenue (Total Amount) per gender.
88 -- Expected output: Gender, Total_Revenue
89 select Gender,
90         SUM(Quantity*Price_Per_Unit) AS Total_Revenue
91 FROM practical1.retail_sales.sales
92 Group By Gender;
93
```

↶ Results ▾ Chart

	A GENDER	# TOTAL_REVENUE
1	Male	223160
2	Female	232840

### Question 17:

PRACTICAL1.RETAIL\_SALES ▾ Settings ▾

```
91 FROM practical1.retail_sales.sales
92 Group By Gender;
93 -----
94 -- Q17. Find the average Price per Unit per product category.
95 -- Expected output: Product Category, Average_Price
96 select PRODUCT_CATEGORY,
97         AVG (Price_Per_Unit) AS Average_Price
98 FROM practical1.retail_sales.sales
99 Group By PRODUCT_CATEGORY;
```

↶ Results ▾ Chart

	A PRODUCT_CATEGORY	# AVERAGE_PRICE
1	Beauty	184.055375
2	Clothing	174.287749
3	Electronics	181.900585

### Question 18:

PRACTICAL1.RETAIL\_SALES ▾ Settings ▾ [Open in new window](#)

```
101 -- Q18. Find the total revenue per product category where total revenue is greater than
102 -- 10,000.
103 -- Expected output: Product Category, Total_Revenue
104 select PRODUCT_CATEGORY,
105         SUM (Quantity*Price_Per_Unit) AS Total_Revenue
106 FROM practical1.retail_sales.sales
107 Group By PRODUCT_CATEGORY
108 HAVING Total_Revenue > 10000;
109
```

↶ Results ▾ Chart 🔍

	A PRODUCT_CATEGORY	# TOTAL_REVENUE	Query Details
1	Beauty	143515	Query duration
2	Clothing	155580	Rows
3	Electronics	156905	Query ID <b>01bfd2</b>

### Question 19:

PRACTICAL1.RETAIL\_SALES Settings Open i

```
107
110 -- Q19. Find the average quantity per product category where the average is more than 2.
111 -- Expected output: Product Category, Average_Quantity
112 select PRODUCT_CATEGORY,
113        AVG (QUANTITY) AS Average_Quantity
114 FROM practical1.retail_sales.sales
115 Group By PRODUCT_CATEGORY
116 Having Average_Quantity > 2;
117
```

Results Chart

	PRODUCT_CATEGORY	AVERAGE_QUANTITY
1	Beauty	2.511401
2	Clothing	2.547009
3	Electronics	2.482456

Query Details

Query duration

Rows

Query ID 01bfd2

### Question 20:

PRACTICAL1.RETAIL\_SALES Settings Open i

```
118 -- Q20. Display a column called Spending_Level that shows 'High' if Total Amount > 1000,
119 -- otherwise 'Low'.
120 -- Expected output: Transaction ID, Total Amount, Spending_Level
121 select Transaction_ID,
122        Total_Amount,
123        CASE WHEN TOTAL_AMOUNT > 1000 THEN 'High'
124              ELSE 'Low'
125        END AS Spending_Level
126 FROM practical1.retail_sales.sales;
127
128
```

Results Chart

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

Query Details

Query duration

Rows

Query ID 01bfd2

### Question 21:

PRACTICAL1.RETAIL\_SALES Settings

```
131 -- Senior is Age >= 60
132 -- Expected output: Customer ID, Age, Age_Group
133 select Customer_ID,
134        Age,
135        CASE WHEN AGE < 30 THEN 'Youth'
136              WHEN AGE between 30 AND 59 THEN 'Adult'
137              WHEN AGE >= 60 THEN 'Senior'
138        END AS Age_Group
139 FROM practical1.retail_sales.sales;
140
141
142
```

Results Chart

	CUSTOMER_ID	AGE	AGE_GROUP
1	CUST001	34	Adult
2	CUST002	26	Youth
3	CUST003	50	Adult
4	CUST004	37	Adult

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

-- Expected output: All columns

```
select *  
  
FROM practical1.retail_sales.sales;
```

-----  
-- 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 practical1.retail_sales.sales;
```

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

-- Expected output: Product Category

```
select DISTINCT Product_Category  
  
FROM practical1.retail_sales.sales;
```

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

-- Expected output: Gender

```
select DISTINCT Gender  
  
FROM practical1.retail_sales.sales;
```

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

-- Expected output: All columns

```
select *  
  
FROM practical1.retail_sales.sales  
  
WHERE Age > 40;
```

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

-- Expected output: All columns

```

select *
FROM practical1.retail_sales.sales
Where PRICE_PER_UNIT between 100 AND 500;

-----

-- Q7. Display all transactions where the Product Category is either 'Beauty' or
-- 'Electronics'.
-- Expected output: All columns
select *
FROM practical1.retail_sales.sales
WHERE PRODUCT_CATEGORY = 'Beauty' OR PRODUCT_CATEGORY = 'Electronics';

-----

-- Q8. Display all transactions where the Product Category is not 'Clothing'.
-- Expected output: All columns
select *
FROM practical1.retail_sales.sales
WHERE PRODUCT_CATEGORY != 'Clothing';

-----

-- Q9. Display all transactions where the Quantity is greater than or equal to 3.
-- Expected output: All columns
select *
FROM practical1.retail_sales.sales
WHERE QUANTITY >= 3;

-----

-- Q10. Count the total number of transactions.
-- Expected output: Total_Transactions
select COUNT (TRANSACTION_ID) AS Total_Transactions
FROM practical1.retail_sales.sales;

-----

-- Q11. Find the average Age of customers.
-- Expected output: Average_Age
select AVG (AGE) AS Average_Age

```

```
FROM practical1.retail_sales.sales;
```

```
-----
```

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

```
-- Expected output: Total_Quantity
```

```
select SUM (QUANTITY) AS Total_Quantity
```

```
FROM practical1.retail_sales.sales;
```

```
-----
```

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

```
-- Expected output: Max_Total_Amount
```

```
select max (TOTAL_AMOUNT) AS Max_Total_Amount
```

```
FROM practical1.retail_sales.sales;
```

```
-----
```

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

```
-- Expected output: Min_Price_per_Unit
```

```
select MIN (PRICE_PER_UNIT) AS Min_Price_Per_Unit
```

```
FROM practical1.retail_sales.sales;
```

```
-----
```

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

```
-- Expected output: Product Category, Transaction_Count
```

```
select Product_Category,
```

```
       COUNT (Transaction_ID) AS Transaction_Count
```

```
FROM practical1.retail_sales.sales
```

```
Group BY PRODUCT_CATEGORY;
```

```
-----
```

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

```
-- Expected output: Gender, Total_Revenue
```

```
select Gender,
```

```
       SUM(Quantity*Price_Per_Unit) AS Total_Revenue
```

```
FROM practical1.retail_sales.sales
```

```
Group By Gender;
```

```
-----
```

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

-- Expected output: Product Category, Average\_Price

```
select PRODUCT_CATEGORY,  
       AVG (Price_Per_Unit) AS Average_Price  
FROM practical1.retail_sales.sales  
Group By PRODUCT_CATEGORY;
```

-----  
-- Q18. Find the total revenue per product category where total revenue is greater than

-- 10,000.

-- Expected output: Product Category, Total\_Revenue

```
select PRODUCT_CATEGORY,  
       SUM (Quantity*Price_Per_Unit) AS Total_Revenue  
FROM practical1.retail_sales.sales  
Group By PRODUCT_CATEGORY  
HAVING Total_Revenue > 10000;
```

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

-- Expected output: Product Category, Average\_Quantity

```
select PRODUCT_CATEGORY,  
       AVG (QUANTITY) AS Average_Quantity  
FROM practical1.retail_sales.sales  
Group By PRODUCT_CATEGORY  
Having Average_Quantity > 2;
```

-----  
-- Q20. Display a column called Spending\_Level that shows 'High' if Total Amount > 1000,

-- otherwise 'Low'.

-- Expected output: Transaction ID, Total Amount, Spending\_Level

```
select Transaction_ID,  
       Total_Amount,  
       CASE WHEN TOTAL_AMOUNT > 1000 THEN 'High'  
            ELSE 'Low'
```

```

END AS Spending_Level

FROM practical1.retail_sales.sales;

-----

-- 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 < 30 THEN 'Youth'
      WHEN AGE between 30 AND 59 THEN 'Adult'
      WHEN AGE >= 60 THEN 'Senior'
END AS Age_Group
FROM practical1.retail_sales.sales;

```