

Q1

```
1 -----  
2 --Q1 Display all columns for all transactions.  
3 --Expected output : ALL column  
4  
5   SELECT*  
6   FROM practical1.dataset.retail_sales;  
7  
8 -----  
9  
10  --Q2 Display the only the transaction ID ,Date , and Customer ID for all records  
11  --Expected output : Transaction ID ,Date ,Customer ID  
12  
13  
14  
15  
16  SELECT transaction_id,  
17    date,  
18    customer_id
```

→ Results ↵ Chart

The screenshot shows a database query results interface. At the top, there are tabs for 'Results' and 'Chart'. Below the tabs is a search bar and some filter icons. The main area displays a table with the following data:

# TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY
1	2023-11-24	CUST001	Male	34	Beauty	1
2	2023-02-27	CUST002	Female	26	Clothing	2
3	2023-01-13	CUST003	Male	50	Electronics	3

On the right side of the table, there is a 'Query Details' panel with the following information:

- Query duration: 456ms
- Rows: 1K
- Query ID: 01bfdbad-000c-b142-0...

Q2

```

10 --Q2 Display the only the transaction ID ,Date , and Customer ID for all records
11 --Expected output : Transaction ID ,Date ,Customer ID
12
13
14 SELECT Transaction_ID,Date,Customer_Id
15 FROM practical1.dataset.retail_sales;
16 -----
17
18
19
20 -----
21 --
22 --Q3. Display all the distinct product categories in the dataset.
23 --Expected output :Product.Category
24
25 SELECT DISTINCT product_category
26 FROM practical1.dataset.retail_sales;

```

Results

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

Query Details

- Query duration: 67ms
- Rows: 1K
- Query ID: 01bfd3-000c-b142-0...

Q3

```

21 --Q3. Display all the distinct product categories in the dataset.
22 --Expected output :Product.Category
23
24 SELECT DISTINCT product_category
25 FROM practical1.dataset.retail_sales;
26
27
28
29 -----
30 --
31 --Q4.Display all transactions all the distinct gender values in the dataset .

```

Results

PRODUCT_CATEGORY
Clothing
Beauty
Electronics

Query Details

- Query duration: 397ms
- Rows: 3
- Query ID: 01bfd3-000c-b142-0...

Q4.

```

30 --Q4.Display all transactions all the distinct gender values in the dataset .
31 --Expected output:Gender
32
33 SELECT DISTINCT Gender
34 FROM practical1.dataset.retail_sales;
35
36 -----

```

Results

GENDER
Male
Female

Query Details

- Query duration: 397ms
- Rows: 3
- Query ID: 01bfd3-000c-b142-0...

Q5

```
38 --Q5.Display all transactions where the Age is greater than 40 .
39 --Expected output:All columns
40
41
42 SELECT*
43 FROM practical1.dataset.retail_sales
44 WHERE AGE > 40;
45
46
--
```

↳ Results ↵ Chart

Query Details
Query duration
Rows
Query ID

	# TRANSACTION_ID	⌚ DATE	▲ CUSTOMER_ID	▲ GENDER	# AGE	▲ PRODUCT_CATEGORY	# QUANTITY
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	

Q6

```
48 --Q6.Display all transactions where the price per unit is between 100 and 500.
49 --Expected output:All columns
50
51
52 SELECT*
53 FROM practical1.dataset.retail_sales
54 WHERE price_per_unit BETWEEN 100 AND 500;
55
56
--
```

Q7.Display all transactions where the product category is either 'Beauty' or 'Electronics'
Expected

↳ Results ↵ Chart

Query Details
Query duration
Rows
Query ID

	# TRANSACTION_ID	⌚ DATE	▲ CUSTOMER_ID	▲ GENDER	# AGE	▲ PRODUCT_CATEGORY	# QUANTITY
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	

Q7

```

56
57 Q7.Display all transactions where the product category is either 'Beauty' or 'Electronics'
58 Expected
59 output: All Columns
60
61 SELECT*
62 FROM practical1.dataset.retail_sales
63 WHERE product_category IN ('Beauty', 'Electronics');
64

```

↳ Results ↵ Chart

	# TRANSACTION_ID	⌚ DATE	▲ CUSTOMER_ID	▲ GENDER	# AGE	▲ PRODUCT_CATEGORY	# QUANTIT
1	1	2023-11-24	CUST001	Male	34	Beauty	
2	5	2023-05-06	CUST005	Male	30	Beauty	
3	6	2023-04-25	CUST006	Female	45	Beauty	

Q8

```

67 Q8.Display all transactions where the Product Category is not 'Clothing'.
68 Expected output:All columns
69
70 SELECT*
71 FROM practical1.dataset.retail_sales
72 WHERE product_category NOT IN('Beauty', 'Electronics');
73
74 -----
-----
```

↳ Results ↵ Chart

	# TRANSACTION_ID	⌚ DATE	▲ CUSTOMER_ID	▲ GENDER	# AGE	▲ PRODUCT_CATEGORY	# QUANTIT
1	2	2023-02-27	CUST002	Female	26	Clothing	
2	3	2023-01-13	CUST003	Male	50	Electronics	
3	4	2023-05-21	CUST004	Male	37	Clothing	

Q8

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

```
78  
79     SELECT*  
80     FROM practical1.dataset.retail_sales  
81     WHERE QUANTITY >=3;  
82  
83 -----  
--  
84  
85 Q10. Count the total number of transactions.
```

↳ Results ⚡ Chart

	# TRANSACTION_ID	⌚ DATE	▲ CUSTOMER_ID	▲ GENDER	# AGE	▲ PRODUCT_CATEGORY	# QUANTIT
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	

Q9

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

```
78  
79     SELECT*  
80     FROM practical1.dataset.retail_sales  
81     WHERE QUANTITY >=3;  
82  
83 -----  
--  
84  
85 Q10. Count the total number of transactions.
```

↳ Results ⚡ Chart

	# TRANSACTION_ID	⌚ DATE	▲ CUSTOMER_ID	▲ GENDER	# AGE	▲ PRODUCT_CATEGORY	# QUANTIT	Query Details
1	1	2023-11-24	CUST001	Male	34	Beauty		Query duration
2	8	2023-02-22	CUST008	Male	30	Electronics		Rows
3	10	2023-10-07	CUST010	Female	52	Clothing		Query ID 01bfdbe

Q10

```

76 Q9.Display all transactions where the Quantity is greater than or equal to 3.
77 Expected output:All columns
78
79
80     SELECT*
81     FROM practical1.dataset.retail_sales
82     WHERE QUANTITY >=3;
83
84
85 Q10. Count the total number of transactions

```

Results

	# TRANSACTION_ID	⌚ DATE	▲ CUSTOMER_ID	▲ GENDER	# AGE	▲ PRODUCT_CATEGORY	# QUANTITY
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	

Q11

```

76 Q9.Display all transactions where the Quantity is greater than or equal to 3.
77 Expected output:All columns
78
79
80     SELECT*
81     FROM practical1.dataset.retail_sales
82     WHERE QUANTITY >=3;
83
84
85 Q10. Count the total number of transactions

```

Results

	# TRANSACTION_ID	⌚ DATE	▲ CUSTOMER_ID	▲ GENDER	# AGE	▲ PRODUCT_CATEGORY	# QUANTITY
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	

Q12

```

100    Q12.Find the total quantity of products sold .
101    Expected output:Total_Quantity
102
103    SELECT SUM(Quantity) AS Total_Quantity
104    FROM practical1.dataset.retail_sales;

```

↳ Results ↵ Chart

# TOTAL_QUANTITY	
1	2514

Query Details

- Query duration 22ms
- Rows 1
- Query ID 01bfdedb2-000c-b142-0...

(?)

Q13

```

108    Q13.Find the number of transactions per Product Category.
109    Expected output:Max_Total_Amount
110
111    SELECT MAX(Total_Amount)
112    FROM practical1.dataset.retail_sales;
113
114
115
116    --Q14.Find the minimum Price per Product category .
117    --Expected output:Min_Price_per_unit

```

↳ Results ↵ Chart

# MAX(TOTAL_AMOUNT)	
1	2000

Query Details

- Query duration 30ms
- Rows 1
- Query ID 01bfdedb4-000c-b142-0...

(?)

Q14

```

115
116    --Q14.Find the minimum Price per Product category .
117    --Expected output:Min_Price_per_unit
118
119    SELECT MIN(price_per_unit)
120    FROM practical1.dataset.retail_sales;

```

↳ Results ↵ Chart

# MIN(PRICE_PER_UNIT)	
1	25

Query Details

- Query duration 4ms
- Rows 1
- Query ID 01bfdedb7-000c-b142-0...

Q15

```

124 -Q15.Find the number of transactions per Product Category .
125 --Expected output:Product category,Transaction_Count
126
127 | SELECT COUNT(product_category)
128 | FROM practical1.dataset.retail_sales
129 | GROUP BY transaction_id;
130 |
131 |
132 |
133 |
134 |
135 |
136 |
137 |
138 |
139

```

[↳ Results](#) [↗ Chart](#)

	# COUNT(PRODUCT_CATEGORY)	
1		1
2		1
3		1
4		1

Query Details
Query duration
Rows
Query ID 01b

Q16

```

131 --Q16.Find the total revenue (totalAmount) per gender .
132 --Expected output:Gender ,Total_Revenue
133
134 | SELECT SUM(total_amount)
135 | FROM practical1.dataset.retail_sales
136 | GROUP BY GENDER ;
137
138
139

```

[↳ Results](#) [↗ Chart](#)

	# SUM(TOTAL_AMOUNT)	
1		223160
2		232840

Query Details
Query duration
Rows
Query ID 01bfdbd9-000c

Q17

```

140    --Q17.Find the average Price per unit Per product category.
141    --Expected output :Product Category ,Average_Price
142
143    SELECT AVG (PRICE_PER_UNIT)
144    FROM practical1.dataset.retail_sales
145    GROUP BY product_category;
146
147
148
149    --Q18. Find the total revenue per product category where total revenue is greater than 10000.

```

↳ Results ⚡ Chart

	# AVG (PRICE_PER_UNIT)	Query Details
1	184.055375	Query duration
2	174.287749	Rows
3	181.900585	Query ID 01b

Q18

```

148
149    --Q18. Find the total revenue per product category where total revenue is greater than 10000.
150    --Expected output :Product Category ,Total_Revenue
151
152    SELECT Product_category ,SUM (TOTAL_AMOUNT) AS Total_Revenue
153    FROM practical1.dataset.retail_sales
154    GROUP BY product_category
155    HAVING COUNT (total_revenue)>10000;
156
157
158

```

↳ Results ⚡ Chart

!	Query Details
Aggregate functions cannot be nested. [SUM(DETAIL_SALES.TOTAL_AMOUNT)] nested in	Query duration
Rows	Query ID 01b

Q19

```

159 --Q19.Find the average quantity per product category where the average is more than 2 .
160 --Expected output:Product category ,Average_Quantity
161
162 SELECT Product_category,AVG(QUANTITY) AS Average_Quantity
163 FROM practical1.dataset.retail_sales
164 GROUP BY PRODUCT_CATEGORY
165 HAVING AVG (Quantity)>2;
166
167 -----
168
169 --Q20.Display a column called Spending _Level that shows 'High' if Total Amount >1000,otherwise 'Low'.
170 --Expected output:Transaction ID,Total Amount ,Spending_Level

```

↳ 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 0

Q20

```

169 --Q20.Display a column called Spending _Level that shows 'High' if Total Amount >1000,otherwise 'Low'
170 --Expected output:Transaction ID,Total Amount ,Spending_Level
171
172 SELECT Transaction_id,
173     total_amount,
174     CASE
175         WHEN TOTAL_AMOUNT>1000 THEN 'High'
176         ELSE AS Spending_Level
177     FROM practical1.dataset.retail_sales;
178
179 -----
180
181 --Q21.Display a new column called Age_Group that labels customers as:

```

↳ Results ⚡ Chart



Syntax error: unexpected 'AS'. (line 176)

Quer

Quer

Row

Quer

Q21

```
180  
181 --Q21.Display a new column called Age_Group that labels customers as:  
182 ~'Youth' if Age <30  
183 ~'Adult' if Age is between 30 and 59  
184 ~ 'Senior' if Age > =60  
185 --Expected output :Customer ID ,Age,Age_Group  
186  
187  
188 SELECT Customer_ID,  
189 Age,  
190 CASE  
191 WHEN Age < 30 THEN 'Youth'  
192 WHEN Age BETWEEN 30 AND 59 THEN 'Adult'  
193 ELSE AS Age_Group  
194 FROM practical1.dataset.retail_sales;  
195
```

↳ Results ↵ Chart

🔍 ⏪ ⏹ ⏴

Query Details

Query duration

Rows

Query ID [01bfdbe2-000c-b](#)



Syntax error: unexpected 'AS'. (line 193)