

Practical exercise 1

1.

PRACTICALQ1.PUBLIC ▾ Settings ▾

Code Versions

```
1 | SELECT * FROM example_table;
2 |
3 | SELECT TRANSACTION_ID,DATE,CUSTOMER_ID FROM EXAMPLE_TABLE;
4 |
5 | SELECT DISTINCT PRODUCT_CATEGORY FROM EXAMPLE_TABLE;
6 |
```

↳ Results

#	TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER
1	1	2023-11-24	CUST001	Male
2	2	2023-02-27	CUST002	Female
3	3	2023-01-13	CUST003	Male
4	4	2023-05-21	CUST004	Male
5	5	2023-05-06	CUST005	Male
6	6	2023-04-25	CUST006	Female
7	7	2023-03-13	CUST007	Male
8	8	2023-02-22	CUST008	Male
9	9	2023-12-13	CUST009	Male
10	10	2023-10-07	CUST010	Female
11	11	2023-02-14	CUST011	Male
12	12	2023-10-30	CUST012	Male
13	13	2023-08-05	CUST013	Male

Query Details

Query duration 207ms

Rows 1K

Query ID 01bbd70b-0000-8f88-0...

Show more

Calculating stats...

```
3 | SELECT TRANSACTION_ID,DATE,CUSTOMER_ID FROM EXAMPLE_TABLE;
4 |
5 | SELECT DISTINCT PRODUCT_CATEGORY FROM EXAMPLE_TABLE;
6 |
```

Results

#	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

2.

PRACTICALQ1.PUBLIC ▾ Settings ▾

```
1 | SELECT * FROM example_table;
2 |
3 | SELECT TRANSACTION_ID,DATE,CUSTOMER_ID FROM EXAMPLE_TABLE;
4 |
5 | SELECT DISTINCT PRODUCT_CATEGORY FROM EXAMPLE_TABLE;
6 |
```

↳ Results ▾ Chart

	PRODUCT_CATEGORY
1	Beauty
2	Clothing
3	Electronics

3.

```
6 |
7 | SELECT DISTINCT GENDER FROM EXAMPLE_TABLE;
8 |
9 | SELECT *
10| FROM EXAMPLE_TABLE
11| WHERE AGE>40;
12|
13| SELECT *
14| FROM EXAMPLE_TABLE
15| WHERE PRICE_PER_UNIT BETWEEN 100 AND 500;
16|
17| SELECT *
18| FROM EXAMPLE_TABLE
19| WHERE PRODUCT_CATEGORY IN ('Beauty', 'Electronics');
20|
```

→ Results ▾ Chart

	GENDER
	Male
	Female

4.

```
8
9   | SELECT *
10  | FROM EXAMPLE_TABLE
11  | WHERE AGE>40;
12
13  | SELECT *
14  | FROM EXAMPLE_TABLE
15  | WHERE PRICE_PER_UNIT BETWEEN 100 AND 500;
16
17  | SELECT *
18  | FROM EXAMPLE_TABLE
19  | WHERE PRODUCT_CATEGORY IN ('Beauty', 'Electronics');
20
```

↳ Results

↗ Chart

	# TRANSACTION_ID	🕒 DATE	▲ CUSTOMER_ID	▲ GENDER	# AG
1	3	2023-01-13	CUST003	Male	
2	6	2023-04-25	CUST006	Female	
3	7	2023-03-13	CUST007	Male	
4	9	2023-12-13	CUST009	Male	
5	10	2023-10-07	CUST010	Female	
6	14	2023-01-17	CUST014	Male	
7	15	2023-01-16	CUST015	Female	
8	18	2023-04-30	CUST018	Female	
9	19	2023-09-16	CUST019	Female	
10	21	2023-01-14	CUST021	Female	

5.

```
12
13 | SELECT *
14 | FROM EXAMPLE_TABLE
15 | WHERE PRICE_PER_UNIT BETWEEN 100 AND 500;
16
17 | SELECT *
18 | FROM EXAMPLE_TABLE
19 | WHERE PRODUCT_CATEGORY IN ('Beauty', 'Electronics');
20
```

↳ Results ⚡ Chart

	# TRANSACTION_ID	⌚ DATE	▲ CUSTOMER_ID	▲ GENDER	# AG
1	2	2023-02-27	CUST002	Female	
2	4	2023-05-21	CUST004	Male	
3	9	2023-12-13	CUST009	Male	
4	13	2023-08-05	CUST013	Male	
5	15	2023-01-16	CUST015	Female	
6	16	2023-02-17	CUST016	Male	
7	20	2023-11-05	CUST020	Male	
8	21	2023-01-14	CUST021	Female	
9	24	2023-11-29	CUST024	Female	
10	26	2023-10-07	CUST026	Female	

6.

```
16  
17 | SELECT *  
18 | FROM EXAMPLE_TABLE  
19 | WHERE PRODUCT_CATEGORY IN ('Beauty', 'Electronics');  
20  
21 | SELECT *  
22 | FROM EXAMPLE_TABLE  
23 | WHERE PRODUCT_CATEGORY Not IN ('Clothing');  
24  
25 | SELECT *  
26 | FROM EXAMPLE_TABLE  
27 | WHERE QUANTITY>=3;  
28  
29 | SELECT COUNT(TRANSACTION_ID) AS Total_transactions  
30 | FROM EXAMPLE_TABLE;  
31
```

↳ Results ↵ Chart

#	TRANSACTION_ID	⌚ DATE	▲ CUSTOMER_ID	▲ GENDER	# AC
1	1	2023-11-24	CUST001	Male	
2	3	2023-01-13	CUST003	Male	
3	5	2023-05-06	CUST005	Male	
4	6	2023-04-25	CUST006	Female	
5	8	2023-02-22	CUST008	Male	
6	9	2023-12-13	CUST009	Male	
7	12	2023-10-30	CUST012	Male	
8	13	2023-08-05	CUST013	Male	
9.	15	2023-01-16	CUST015	Female	

```
21 | SELECT *  
22 | FROM EXAMPLE_TABLE  
23 | WHERE PRODUCT_CATEGORY Not IN ('Clothing');  
24 |  
25 | SELECT *  
26 | FROM EXAMPLE_TABLE  
27 | WHERE QUANTITY>=3;  
28 |  
29 | SELECT COUNT(TRANSACTION_ID) AS Total_transactions  
30 | FROM EXAMPLE_TABLE;  
31 |
```

↳ Results

↗ Chart

#	TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AG
1	1	2023-11-24	CUST001	Male	
2	3	2023-01-13	CUST003	Male	
3	5	2023-05-06	CUST005	Male	
4	6	2023-04-25	CUST006	Female	
5	8	2023-02-22	CUST008	Male	

8.

```

23 | WHERE PRODUCT_CATEGORY NOT IN ('Clothing');
24 |
25 | SELECT *
26 | FROM EXAMPLE_TABLE
27 | WHERE QUANTITY>=3;
28 |
29 | SELECT COUNT(TRANSACTION_ID) AS Total_transactions
30 | FROM EXAMPLE_TABLE;
31 |
32 | SELECT AVG(AGE) AS Average_age
33 | FROM EXAMPLE_TABLE;
34 |
35 | SELECT SUM(Quantity) AS TOTAL_QUANTITY
36 | FROM EXAMPLE_TABLE;
37 |
38 | SELECT MAX(Total_Amount) AS MAX_TOTAL_AMOUNT
39 | FROM EXAMPLE_TABLE;
40 |
41 | SELECT MIN(PRICE_PER_UNIT) AS Min_price_per_unit
42 | FROM EXAMPLE_TABLE;

```

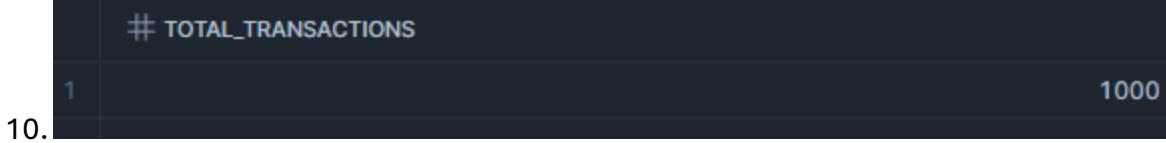
↳ Results

↗ Chart

	# TRANSACTION_ID	🕒 DATE	▲ CUSTOMER_ID	▲ GENDER	# AG
1.	1	2023-11-24	CUST001	Male	
2.	8	2023-02-22	CUST008	Male	
3.	10	2023-10-07	CUST010	Female	
4.	12	2023-10-30	CUST012	Male	
5.	13	2023-08-05	CUST013	Male	
6.	14	2023-01-17	CUST014	Male	
7.	15	2023-01-16	CUST015	Female	
8.	16	2023-02-17	CUST016	Male	
9.	17	2023-04-22	CUST017	Female	

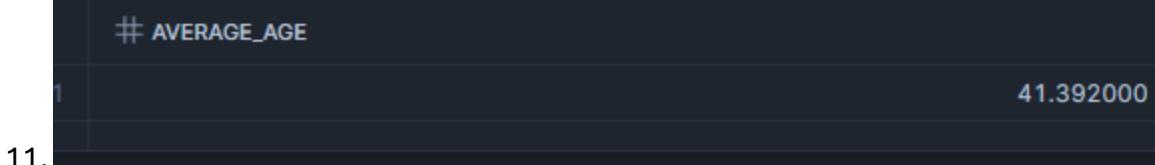
```
28 |     SELECT COUNT(TRANSACTION_ID) AS Total_transactions
29 |     FROM EXAMPLE_TABLE;
30 |
31 |     SELECT AVG(AGE) AS Average_age
32 |     FROM EXAMPLE_TABLE;
33 |
34 |     SELECT SUM(Quantity) AS TOTAL_QUANTITY
35 |     FROM EXAMPLE_TABLE;
36 |
37 |     SELECT MAX(Total_Amount) AS MAX_TOTAL_AMOUNT
38 |     FROM EXAMPLE_TABLE;
39 |
40 |     SELECT MIN(PRICE_PER_UNIT) AS Min_price_per_unit
41 |     FROM EXAMPLE_TABLE;
```

↳ Results ↗ Chart



```
31 |
32 |     SELECT AVG(AGE) AS Average_age
33 |     FROM EXAMPLE_TABLE;
34 |
35 |     SELECT SUM(Quantity) AS TOTAL_QUANTITY
36 |     FROM EXAMPLE_TABLE;
37 |
38 |     SELECT MAX(Total_Amount) AS MAX_TOTAL_AMOUNT
39 |     FROM EXAMPLE_TABLE;
40 |
41 |     SELECT MIN(PRICE_PER_UNIT) AS Min_price_per_unit
42 |     FROM EXAMPLE_TABLE;
```

↳ Results ↗ Chart



```
34  
35 | SELECT SUM(Quantity) AS TOTAL_QUANTITY  
36 | FROM EXAMPLE_TABLE;  
37  
38 | SELECT MAX(Total_Amount) AS MAX_TOTAL_AMOUNT  
39 | FROM EXAMPLE_TABLE;  
40  
41 | SELECT MIN(PRICE_PER_UNIT) AS Min_price_per_unit  
42 | FROM EXAMPLE_TABLE;  
43  
44 | SELECT Product_Category,  
45 | COUNT(Transaction_ID) AS TRANSACTION_COUNT  
46 | FROM EXAMPLE_TABLE  
47 | GROUP BY Product_Category;  
48
```

↳ Results ~ Chart

TOTAL_QUANTITY

12. 1 2514

```
37  
38 | SELECT MAX(Total_Amount) AS MAX_TOTAL_AMOUNT  
39 | FROM EXAMPLE_TABLE;  
40  
41 | SELECT MIN(PRICE_PER_UNIT) AS Min_price_per_unit  
42 | FROM EXAMPLE_TABLE;  
43  
44 | SELECT Product_Category,  
45 | COUNT(Transaction_ID) AS TRANSACTION_COUNT  
46 | FROM EXAMPLE_TABLE  
47 | GROUP BY Product_Category;  
48  
49 | SELECT Gender,  
50 | SUM(Total_amount) AS TOTAL_REVENUE  
51 | FROM EXAMPLE_TABLE  
52 | GROUP BY Gender;  
53
```

↳ Results ~ Chart

MAX_TOTAL_AMOUNT

13. 1 2000

```
SELECT MIN(PRICE_PER_UNIT) AS Min_price_per_unit
FROM EXAMPLE_TABLE;

SELECT Product_Category,
COUNT(Transaction_ID) AS TRANSACTION_COUNT
FROM EXAMPLE_TABLE
GROUP BY Product_Category;

SELECT Gender,
SUM(Total_amount) AS TOTAL_REVENU
FROM EXAMPLE_TABLE
GROUP BY Gender;

SELECT Product_Category,
AVG(PRICE_PER_UNIT) AS Average_price,
FROM EXAMPLE_TABLE
GROUP BY Product_Category;
```

```
SELECT Product_Catagory.
```

Results 

MIN_PRICE_PER_UNIT

25

14.

```
SELECT Product_Category,  
COUNT(Transaction_ID) AS TRANSACTION_COUNT  
FROM EXAMPLE_TABLE  
GROUP BY Product_Category;  
  
SELECT Gender,  
SUM(Total_amount) AS TOTAL_REVENU  
E FROM EXAMPLE_TABLE  
GROUP BY Gender;  
  
SELECT Product_Category,  
AVG(PRICE_PER_UNIT) AS Average_price,  
FROM EXAMPLE_TABLE  
GROUP BY Product_Category;
```

```
SELECT Product_Catagory.
```

Results

Chart

PRODUCT_CATEGORY	TRANSACTION_COUNT
beauty	307
othing	351
electronics	342
15.	

```
48
49  SELECT Gender,
50    SUM(Total_amount) AS TOTAL_REVENUE
51  FROM EXAMPLE_TABLE
52  GROUP BY Gender;
53
54  SELECT Product_Category,
55    AVG(PRICE_PER_UNIT) AS Average_price,
56  FROM EXAMPLE_TABLE
57  GROUP BY Product_Category;
58
59  SELECT Product_Category,
60    SUM(Total_Amount) AS Total_revenue,
61  FROM EXAMPLE_TABLE,
62  GROUP BY Product_Category,
63  HAVING SUM(Total_Amount) > 10000;
64
65  SELECT Product_Category,
66    AVG(Quantity) AS AVERAGE_QUANTITY
67  FROM EXAMPLE_TABLE,
68  GROUP BY Product_Category,
69  HAVING AVG(Quantity) > 2;
70
71
```

→ Results ↗ Chart

GENDER	TOTAL_REVENUE
Male	223160
Female	232840

16.

```
| SELECT Product_Category,  
|   AVG(PRICE_PER_UNIT) AS Average_price,  
|   FROM EXAMPLE_TABLE  
|   GROUP BY Product_Category;  
  
SELECT Product_Category,  
      SUM(Total_Amount) AS Total_revenue,  
      FROM EXAMPLE_TABLE,  
      GROUP BY Product_Category,  
      HAVING SUM(Total_Amount) > 10000;  
  
SELECT Product_Category,  
      AVG(Quantity) AS AVERAGE_QUANTITY  
      FROM EXAMPLE_TABLE,  
      GROUP BY Product_Category,  
      HAVING AVG(Quantity) > 2;
```

Results  Chart

PRODUCT_CATEGORY	AVERAGE_PRICE
Beauty	184.055375
Clothing	174.287749
Electronics	181.900585

17.

```
58
59     SELECT Product_Category,
60             SUM(Total_Amount) AS Total_revenue,
61         FROM EXAMPLE_TABLE,
62     GROUP BY Product_Category,
63     HAVING SUM(Total_Amount) > 10000;
64
65     SELECT Product_Category,
66             AVG(Quantity) AS AVERAGE_QUANTITY
67         FROM EXAMPLE_TABLE,
68     GROUP BY Product_Category,
69     HAVING AVG(Quantity) > 2;
70
71
72     SELECT
73         Transaction_ID,
74         Total_amount,
75     FROM EXAMPLE_TABLE,
76     WHEN Total_amount > 1000 THEN 'HIGH'
77     ELSE 'LOW'
78     END AS TOTAL_REVENUE;
79
80     SELECT Customer_ID, Age,
81     FROM EXAMPLE_TABLE,
82     WHEN Age<30 THEN 'YOUTH'
83     ELSE 'ADULT' THEN 'ADULT';
84
```

↳ Results

↗ Chart



18.

Syntax error: unexpected 'HAVING'. (line 63)

```
65  SELECT Product_Category,
66    AVG(Quantity) AS AVERAGE_QUANTITY
67  FROM EXAMPLE_TABLE,
68  GROUP BY Product_Category,
69  HAVING AVG(Quantity) > 2;
70
71
72  SELECT
73    Transaction_ID,
74    Total_amount,
75  FROM EXAMPLE_TABLE,
76  WHEN Total_amount > 1000 THEN 'HIGH'
77  ELSE 'LOW'
78  END AS TOTAL_REVENUE;
79
80  SELECT Customer_ID, Age,
81  FROM EXAMPLE_TABLE,
82  WHEN Age<30 THEN 'YOUTH'
```

↳ Results ↵ Chart



Syntax error: unexpected 'HAVING'. (line 69)

19.

```
/1
72 | SELECT
73 | Transaction_ID,
74 | Total_amount,
75 | FROM EXAMPLE_TABLE,
76 | WHEN Total_amount > 1000 THEN 'HIGH'
77 | ELSE 'LOW'
78 | END AS TOTAL_REVENUE;
79
80 | SELECT Customer_ID, Age,
81 | FROM EXAMPLE_TABLE,
82 | WHEN Age<30 THEN 'YOUTH'
83 | WHEN Age BETWEEN 30 AND 59 THEN 'ADULT'
84 | ELSE 'SENIOR'
85 | END AS AGE_GROUP;
```

→ Results ↵ Chart



Syntax error: unexpected '>'. (line 76)

20.

```
80  SELECT Customer_ID, Age,  
81  FROM EXAMPLE_TABLE,  
82  WHEN Age<30 THEN 'YOUTH'  
83  WHEN Age BETWEEN 30 AND 59 THEN 'ADULT'  
84  ELSE 'SENIOR'  
85  END AS AGE_GROUP;  
86  
87  
88
```

↳ Results

↗ Chart



Syntax error: unexpected '<'. (line 82)

21.