

1. Display all columns for transactions.

▶ (1000) workspace . default

1 SELECT \* FROM retail\_sales\_dataset

	Transaction ID	Date	Customer ID	Gender	Age	Product Category
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
8	8	2023-02-22	CUST008	Male	30	Electronics
9	9	2023-12-13	CUST009	Male	63	Electronics
10	10	2023-10-07	CUST010	Female	52	Clothing
11	11	2023-02-14	CUST011	Male	23	Clothing
12	12	2023-10-30	CUST012	Male	35	Beauty
13	13	2023-08-05	CUST013	Male	22	Electronics
14	14	2023-04-17	CUST014	Male	24	Clothing

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

```
5 SELECT
6 `TRANSACTION ID`,
7 `DATE`,
8 `CUSTOMER ID`
9 FROM retail_sales_dataset;
```

Raw results ▾ + 🔍 🏠 📏 ✕

	1 2 3 TRANSACTION ID	📅 DATE	A B C 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
9	9	2023-12-13	CUST009
10	10	2023-10-07	CUST010
11	11	2023-02-14	CUST011
12	12	2023-10-30	CUST012
13	13	2023-08-05	CUST013
14	14	2023-01-17	CUST014

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

```

11 SELECT
12 DISTINCT `PRODUCT CATEGORY`
13 FROM retail_sales_dataset
14

```

Raw results ▾ +

	A B C PRODUCT CATEGORY
1	Clothing
2	Electronics
3	Beauty

4. Display all the distinct gender values in the dataset

```

3 SELECT
4 DISTINCT Gender
5 FROM retail_sales_dataset

```

Raw results ▾ + 🔍 🏠 📏 ✕

	A B C Gender
1	Male
2	Female

5. Display all the transactions where the age is greater than 40.

```
7 SELECT *
8 FROM retail_sales_dataset
9 WHERE AGE > 40
```

	1 <sup>2</sup> <sub>3</sub> Transaction ID	📅 Date	A <sup>B</sup> <sub>C</sub> Customer ID	A <sup>B</sup> <sub>C</sub> Gender	1 <sup>2</sup> <sub>3</sub> Age	A <sup>B</sup> <sub>C</sub> 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
6	14	2023-01-17	CUST014	Male	64	Clothing
7	15	2023-01-16	CUST015	Female	42	Electronics
8	18	2023-04-30	CUST018	Female	47	Electronics
9	19	2023-09-16	CUST019	Female	62	Clothing
10	21	2023-01-14	CUST021	Female	50	Beauty
11	24	2023-11-29	CUST024	Female	49	Clothing
12	25	2023-12-26	CUST025	Female	64	Beauty
13	28	2023-04-23	CUST028	Female	43	Beauty
14	30	2023-08-18	CUST030	Female	43	Electronics

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

```
7 SELECT *
8 FROM retail_sales_dataset
9 WHERE retail_sales_dataset.`Price per Unit` BETWEEN 100 AND 500
```

	1 <sup>2</sup> <sub>3</sub> Age	A <sup>B</sup> <sub>C</sub> Product Category	1 <sup>2</sup> <sub>3</sub> Quantity	1 <sup>2</sup> <sub>3</sub> Price per Unit	1 <sup>2</sup> <sub>3</sub> Total Amount
1	26	Clothing	2	500	100
2	37	Clothing	1	500	50
3	63	Electronics	2	300	60
4	22	Electronics	3	500	150
5	42	Electronics	4	500	200
6	19	Clothing	3	500	150
7	22	Clothing	3	300	90
8	50	Beauty	1	500	50
9	49	Clothing	1	300	30
10	28	Electronics	2	500	100
11	43	Beauty	1	500	50
12	39	Beauty	3	300	90
13	44	Electronics	4	300	120
14	50	Beauty	2	300	60

🕒 991 ms | 396 rows returned

Refreshed 1 minute ago

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

```
13 SELECT *
14 FROM retail_sales_dataset
15 WHERE retail_sales_dataset.`Product Category` = 'Beauty'
16 OR retail_sales_dataset.`Product Category` = 'Electronics';
```

	Transaction ID	Date	Customer ID	Gender	Age	Product Category
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
6	9	2023-12-13	CUST009	Male	63	Electronics
7	12	2023-10-30	CUST012	Male	35	Beauty
8	13	2023-08-05	CUST013	Male	22	Electronics
9	15	2023-01-16	CUST015	Female	42	Electronics
10	18	2023-04-30	CUST018	Female	47	Electronics
11	21	2023-01-14	CUST021	Female	50	Beauty
12	25	2023-12-26	CUST025	Female	64	Beauty
13	26	2023-10-07	CUST026	Female	28	Electronics
14	27	2023-08-03	CUST027	Female	38	Beauty

492 ms | 649 rows returned

Refreshed 2 seconds ago

8. Display all transactions where the product category is not 'Clothing'

```
13 SELECT *
14 FROM retail_sales_dataset
15 WHERE NOT retail_sales_dataset.`Product Category` = 'Clothing' ;
```

	Transaction ID	Date	Customer ID	Gender	Age	Product Category
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
6	9	2023-12-13	CUST009	Male	63	Electronics
7	12	2023-10-30	CUST012	Male	35	Beauty
8	13	2023-08-05	CUST013	Male	22	Electronics
9	15	2023-01-16	CUST015	Female	42	Electronics
10	18	2023-04-30	CUST018	Female	47	Electronics
11	21	2023-01-14	CUST021	Female	50	Beauty
12	25	2023-12-26	CUST025	Female	64	Beauty
13	26	2023-10-07	CUST026	Female	28	Electronics
14	27	2023-08-02	CUST027	Female	30	Beauty

⌚ 423 ms | 649 rows returned Refreshed 2 seconds ago

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

```

13 | SELECT *
14 | FROM retail_sales_dataset
15 | WHERE Quantity >= 3 ;
16 |

```

	Gender	Age	Product Category	Quantity	Price per Unit
1	Male	34	Beauty	3	50
2	Male	30	Electronics	4	25
3	Female	52	Clothing	4	50
4	Male	35	Beauty	3	25
5	Male	22	Electronics	3	500
6	Male	64	Clothing	4	30
7	Female	42	Electronics	4	500
8	Male	19	Clothing	3	500
9	Female	27	Clothing	4	25
10	Male	22	Clothing	3	300
11	Female	35	Clothing	4	30
12	Female	39	Beauty	3	300
13	Male	44	Electronics	4	300
14	Male	30	Beauty	3	20

⌚ 367 ms | 504 rows returned Refreshed 2 seconds ago

10. Count the total number of transactions.

```

17 SELECT
18 COUNT(`Transaction ID`) AS TOTAL_TRANSACTIONS
19 FROM retail_sales_dataset;
20

```

	1.2 TOTAL_TRANSACTIONS
1	1000

11. Find the average age of customers.

```

21 SELECT
22 avg(Age) AS Average_Age
23 FROM retail_sales_dataset;

```

Raw results ▾ +

	1.2 Average_Age
1	41.392

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

```

21 SELECT
22 MAX(`Total Amount`) AS Max_Total_Amount
23 FROM retail_sales_dataset;

```

Raw results ▾ +

	1.2 Max_Total_Amount
1	2000

13. Find the total quantity of products sold

```

3 SELECT
4 SUM (QUANTITY) AS Total_Quantity
5 FROM retail_sales_dataset
6

```

Raw results	
	<sup>1</sup> <sub>3</sub> Total_Quantity
1	2514

14. Find the minimum Price per unit in the dataset.

```

7 SELECT
8 MIN (`price per unit`) AS Min_Price_Per_Unit
9 FROM retail_sales_dataset

```

Raw results	
	<sup>1</sup> <sub>3</sub> Min_Price_Per_Unit
1	25

15. Find the number of transactions per product category.

```

11 Select
12 `product category`,
13 count(`Transaction id`) as Transaction_Count
14 from retail_sales_dataset
15 GROUP BY `product category`

```

Raw results		
	<sup>A</sup> <sub>C</sub> product category	<sup>1</sup> <sub>3</sub> Transaction_Count
1	Clothing	351
2	Electronics	342
3	Beauty	307

16. Find the total revenue (Total Amount) per gender.

```

3 Select
4 gender,
5 sum (`total amount`) as Total_Revenue
6 from retail_sales_dataset
7 Group By gender;

```

Raw results		
	<sup>A</sup> <sub>C</sub> gender	<sup>1</sup> <sub>3</sub> Total_Revenue
1	Male	223160
2	Female	232840

17. Find the average price per unit per product category.

```
3 Select
4 `product category`,
5 avg(`price per unit`) as Average_price
6 from retail_sales_dataset
7 Group By `product category`;
```

Raw results ▾ +

	A <sup>B</sup> product category	1.2 Average_price
1	Clothing	174.28774928774928
2	Electronics	181.90058479532163
3	Beauty	184.05537459283389

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

```
9 SELECT
10 `product category`,
11 SUM(`total amount`) AS Total_Revenue
12 FROM retail_sales_dataset
13 GROUP BY `product category`
14 HAVING SUM(`total amount`) > 10000;
```

Raw results ▾ +

	A <sup>B</sup> product category	1 <sup>2</sup> Total_Revenue
1	Clothing	155580
2	Electronics	156905
3	Beauty	143515

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

```
16 SELECT
17 `product category`,
18 avg(quantity) AS Average_Quantity
19 FROM retail_sales_dataset
20 GROUP BY `product category`
21 HAVING avg(quantity) > 2;
```

Raw results ▾ +

	A <sup>B</sup> product category	1.2 Average_Quantity
1	Clothing	2.547008547008547
2	Electronics	2.482456140350877
3	Beauty	2.511400651465798



20. Display a column called Spending\_Level that shows 'High' if Total Amount > 1000 otherwise 'Low'

```
23 SELECT
24 `transaction id`,
25 `Total Amount`,
26 CASE
27 WHEN `Total Amount` > 1000 THEN 'High'
28 ELSE 'Low'
29 END as Spending_Level
30 FROM retail sales dataset:
```

	1 <sup>2</sup> <sub>3</sub> transaction id	1 <sup>2</sup> <sub>3</sub> Total Amount	A <sup>B</sup> <sub>C</sub> 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
10	10	200	Low
11	11	100	Low
12	12	75	Low
13	13	1500	High
14	14	120	Low

21. 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

```
32 SELECT
33 `customer id`,
34 age,
35 CASE
36 WHEN age < 30 THEN 'Youth'
37 WHEN age BETWEEN 30 AND 59 THEN 'Adult'
38 WHEN age >= 60 THEN 'Senior'
39 END as Age_Group
40 FROM retail sales dataset:
```

Raw results +

	customer id	age	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
11	CUST011	23	Youth
12	CUST012	35	Adult
13	CUST013	22	Youth
14	CUST014	64	Senior