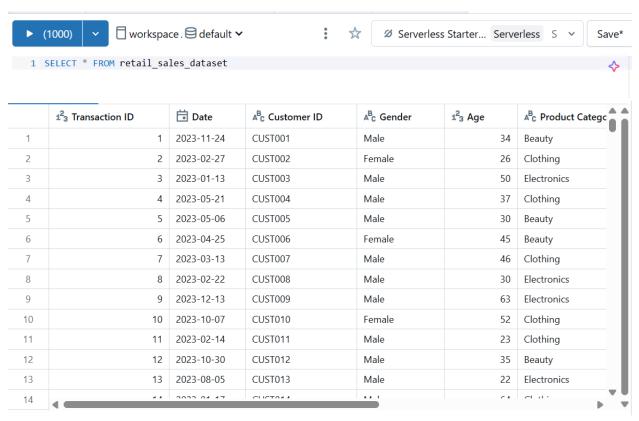
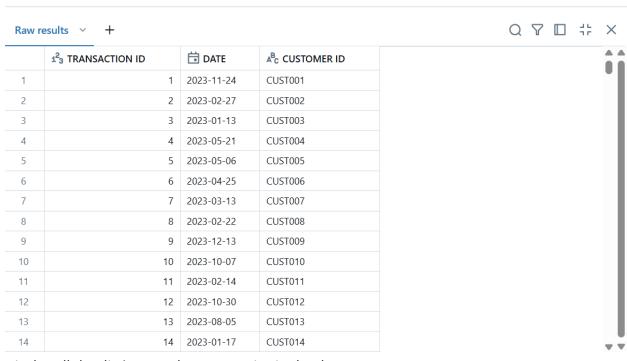
1. Display all columns for transactions.



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;
```



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



4. Display all the distinct gender values in the dataset

•	,	
3	SELECT	
4	DISTINCT Gender	
5	FROM retail_sales_dataset	ĺ



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

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

	1 ² ₃ Transaction ID	□ Date	ABC Customer ID	₄ ^B c Gender	1 ² ₃ Age	AB _C Product Catego
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	20	2022 00 40	CLICTOOO	<u>-</u> 1	40	ег

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 ² ₃ Age	ABC Product Category	1 ² ₃ Quantity	1 ² ₃ Price per Unit	1 ² ₃ 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	15
5	42	Electronics	4	500	20
6	19	Clothing	3	500	15
7	22	Clothing	3	300	91
8	50	Beauty	1	500	5
9	49	Clothing	1	300	3
10	28	Electronics	2	500	10
11	43	Beauty	1	500	5
12	39	Beauty	3	300	91
13	44	Electronics	4	300	12
14		n .	2	200	

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';
```

	ansaction ID	□ Date	A ^B _C Customer ID	A ^B _C Gender	1 ² ₃ Age	ABC 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	2022 00 02	CLICTORY	F I	20	D .

① 492 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';
```

	ansaction ID	□ Date	ABC Customer ID	A ^B C Gender	1 ² 3 Age	ABC Product Category
	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	2022 00 02	CLICTORT	F 1	30	D I

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
```

	A ^B C Gender	1 ² ₃ Age	ABC Product Category	1 ² ₃ Quantity	1 ² ₃ 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	4	20	n .	2	20

① 367 ms | 504 rows returned

Refreshed 2 seco

10. Count the total number of transactions.

```
Total_transaction ID`) AS TOTAL_TRANSACTIONS
FROM retail_sales_dataset;

123 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

+

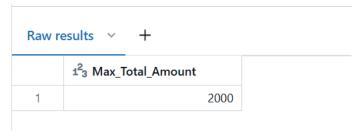
1.2 Average_Age

1

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

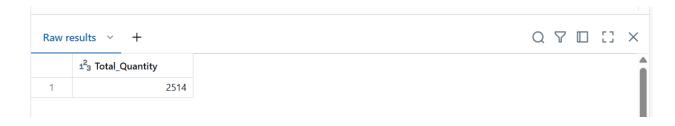
41.392

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



13. Find the total quantity of products sold

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



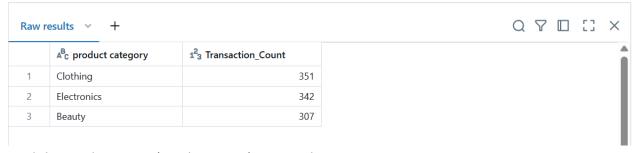
14. Find the minimum Price per unit in the dataset.

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



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`
```



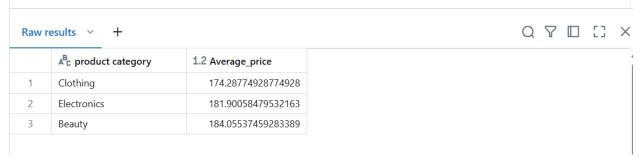
16. Find the total revenue (Total Amount) per gender.

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



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';
```



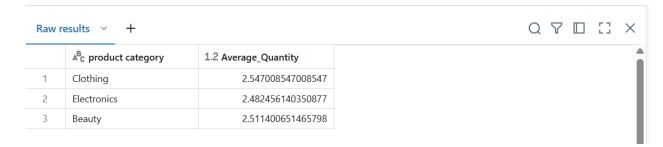
18. Find the total revenue per product category where total revenue is greater than 10 000.

```
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;
```



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

```
SELECT
product category,
savg(quantity) AS Average_Quantity
FROM retail_sales_dataset
GROUP BY `product category`
HAVING avg(quantity) > 2;
```



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 ² ₃ transaction id	1 ² ₃ Total Amount	AB _C 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:
```

