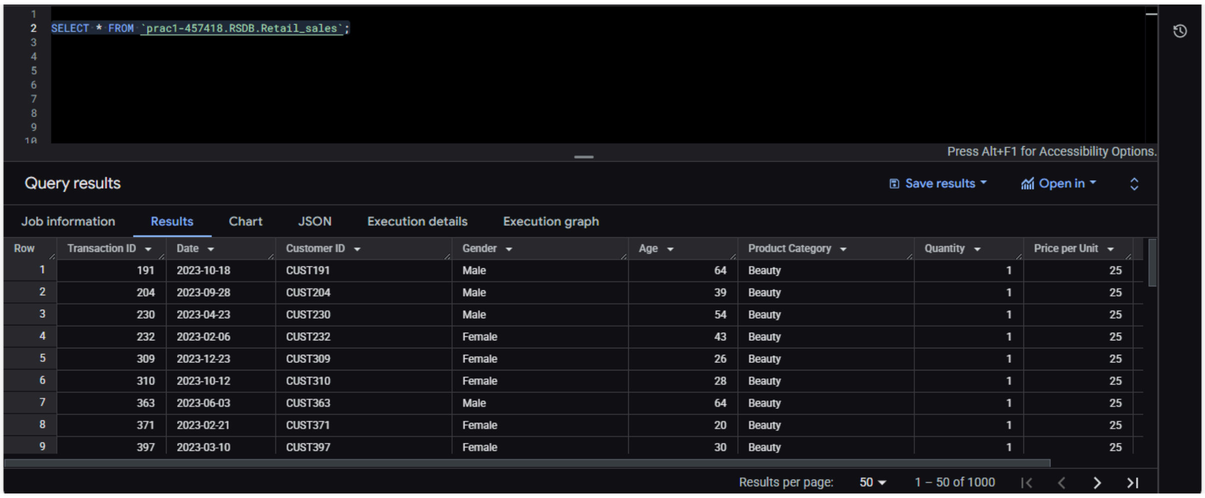
Practical 1: SQL Fundamentals (BigQuery)

1. SELECT Statement

Q1. Display all columns for all transactions.

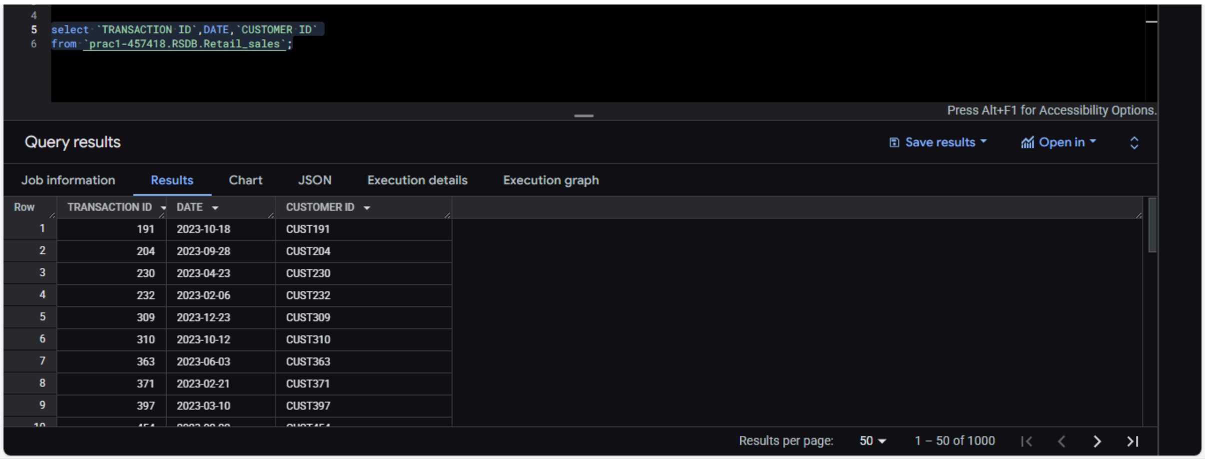
Expected output: All columns



```
SELECT * FROM `prac1-457418.RSDB.Retail_sales`;
```

Row	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit
1	191	2023-10-18	CUST191	Male	64	Beauty	1	25
2	204	2023-09-28	CUST204	Male	39	Beauty	1	25
3	230	2023-04-23	CUST230	Male	54	Beauty	1	25
4	232	2023-02-06	CUST232	Female	43	Beauty	1	25
5	309	2023-12-23	CUST309	Female	26	Beauty	1	25
6	310	2023-10-12	CUST310	Female	28	Beauty	1	25
7	363	2023-06-03	CUST363	Male	64	Beauty	1	25
8	371	2023-02-21	CUST371	Female	20	Beauty	1	25
9	397	2023-03-10	CUST397	Female	30	Beauty	1	25

Q2. Display only the Transaction ID, Date, and Customer ID for all records.
Expected output: Transaction ID, Date, Customer ID



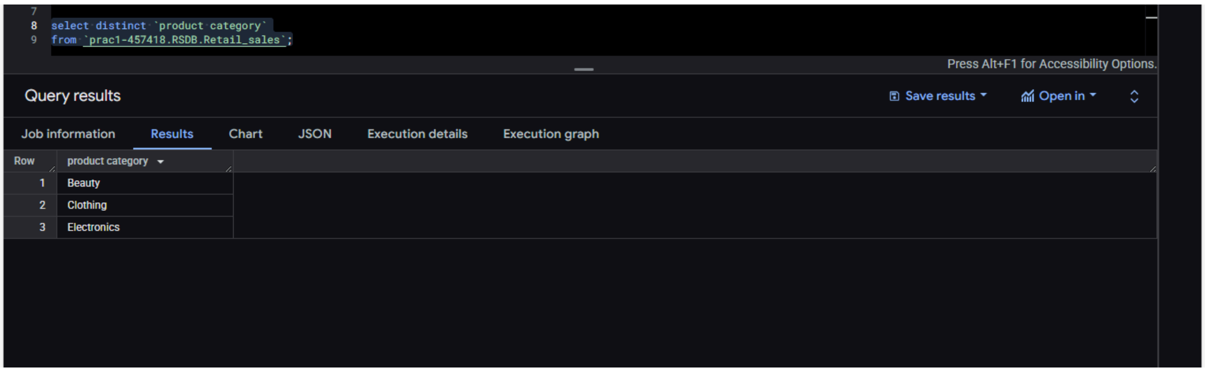
```
select `TRANSACTION ID`,DATE,`CUSTOMER ID`
from `prac1-457418.RSDB.Retail_sales`;
```

Row	TRANSACTION ID	DATE	CUSTOMER ID
1	191	2023-10-18	CUST191
2	204	2023-09-28	CUST204
3	230	2023-04-23	CUST230
4	232	2023-02-06	CUST232
5	309	2023-12-23	CUST309
6	310	2023-10-12	CUST310
7	363	2023-06-03	CUST363
8	371	2023-02-21	CUST371
9	397	2023-03-10	CUST397

2. SELECT DISTINCT Statement

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

Expected output: Product Category

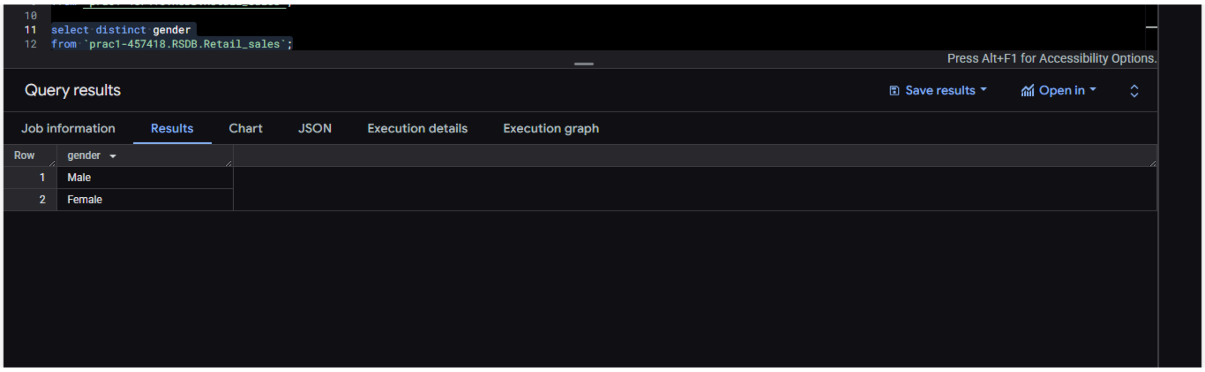


```
select distinct `product category`
from `prac1-457418.RSDB.Retail_sales`;
```

Row	product category
1	Beauty
2	Clothing
3	Electronics

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

Expected output: Gender



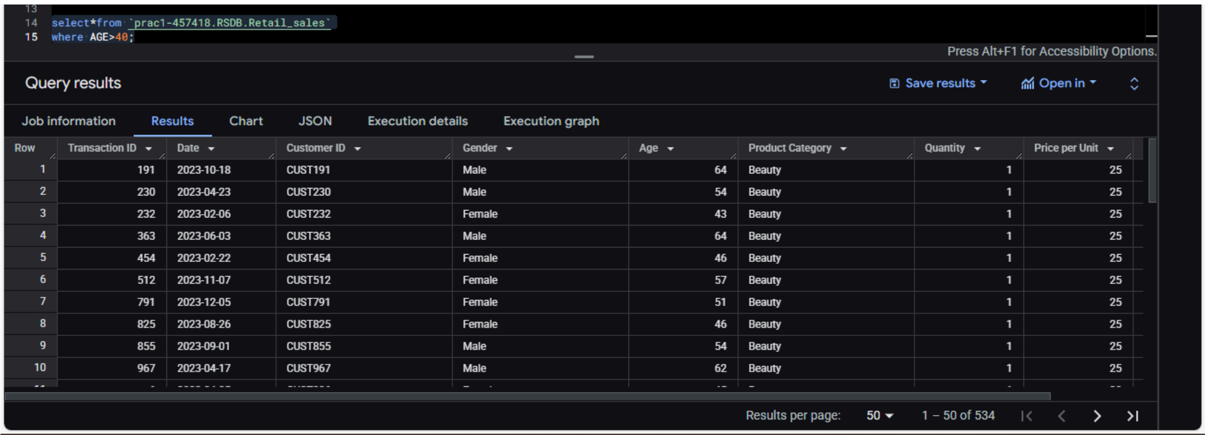
```
select distinct gender
from `prac1-457418.RSDB.Retail_sales`;
```

Row	gender
1	Male
2	Female

3. WHERE Clause

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

Expected output: All columns

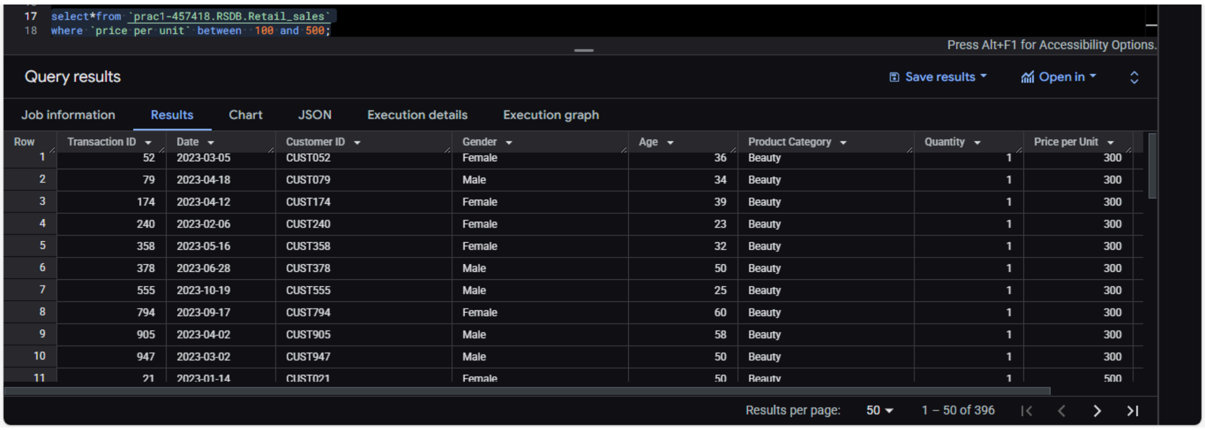


```
select*from `prac1-457418.RSDB.Retail_sales`
where AGE>40;
```

Row	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit
1	191	2023-10-18	CUST191	Male	64	Beauty	1	25
2	230	2023-04-23	CUST230	Male	54	Beauty	1	25
3	232	2023-02-06	CUST232	Female	43	Beauty	1	25
4	363	2023-06-03	CUST363	Male	64	Beauty	1	25
5	454	2023-02-22	CUST454	Female	46	Beauty	1	25
6	512	2023-11-07	CUST512	Female	57	Beauty	1	25
7	791	2023-12-05	CUST791	Female	51	Beauty	1	25
8	825	2023-08-26	CUST825	Female	46	Beauty	1	25
9	855	2023-09-01	CUST855	Male	54	Beauty	1	25
10	967	2023-04-17	CUST967	Male	62	Beauty	1	25

Results per page: 50 1 – 50 of 534

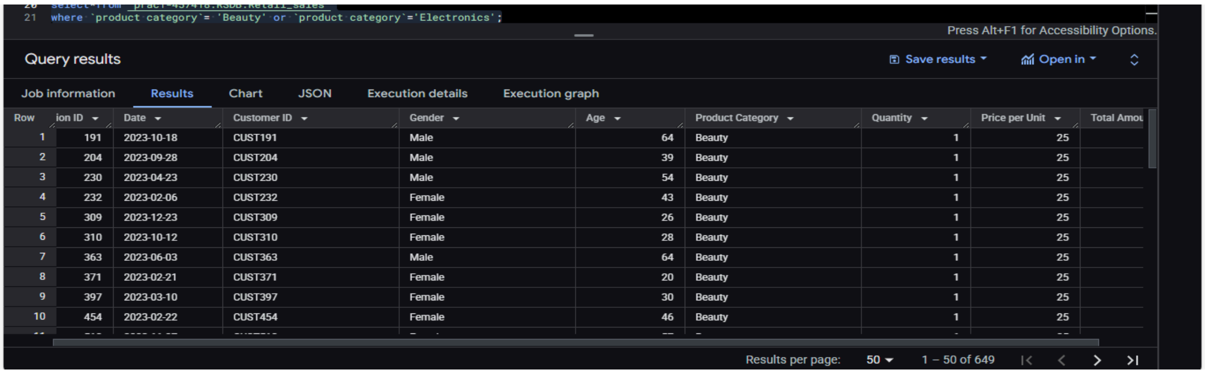
Q6. Display all transactions where the Price per Unit is between 100 and 500.
Expected output: All columns



```
select*from `prac1-457418.RSDB.Retail_sales`
where `price per unit` between  100 and 500;
```

Row	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit
1	52	2023-03-05	CUST052	Female	36	Beauty	1	300
2	79	2023-04-18	CUST079	Male	34	Beauty	1	300
3	174	2023-04-12	CUST174	Female	39	Beauty	1	300
4	240	2023-02-06	CUST240	Female	23	Beauty	1	300
5	358	2023-05-16	CUST358	Female	32	Beauty	1	300
6	378	2023-06-28	CUST378	Male	50	Beauty	1	300
7	555	2023-10-19	CUST555	Male	25	Beauty	1	300
8	794	2023-09-17	CUST794	Female	60	Beauty	1	300
9	905	2023-04-02	CUST905	Male	58	Beauty	1	300
10	947	2023-03-02	CUST947	Male	50	Beauty	1	300
11	21	2023-01-14	CUST021	Female	50	Beauty	1	500

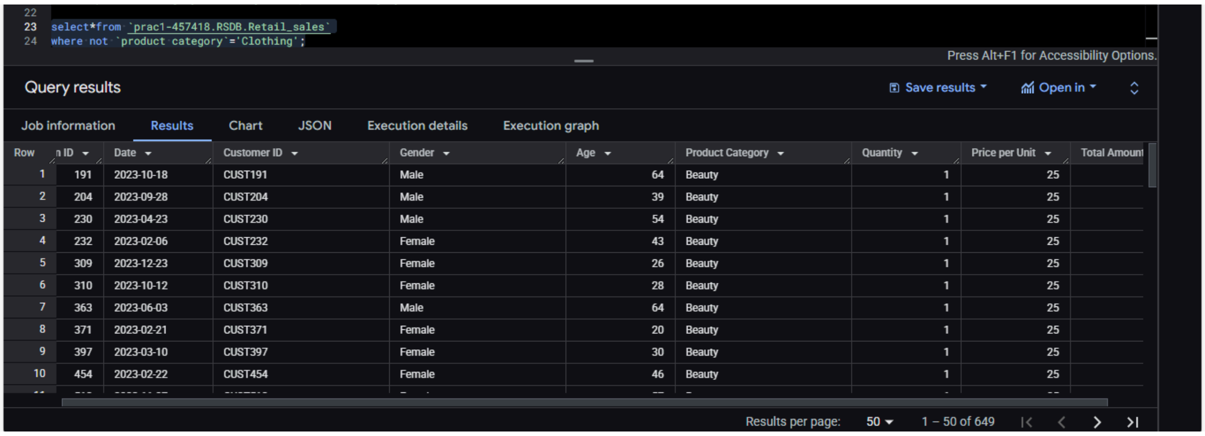
Q7. Display all transactions where the Product Category is either 'Beauty' or 'Electronics'.
 Expected output: All columns



```
select*from `prac1-457418.RSDB.Retail_sales`
where `product category`= 'Beauty' or `product category`='Electronics';
```

Row	ion ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Amou
1	191	2023-10-18	CUST191	Male	64	Beauty	1	25	
2	204	2023-09-28	CUST204	Male	39	Beauty	1	25	
3	230	2023-04-23	CUST230	Male	54	Beauty	1	25	
4	232	2023-02-06	CUST232	Female	43	Beauty	1	25	
5	309	2023-12-23	CUST309	Female	26	Beauty	1	25	
6	310	2023-10-12	CUST310	Female	28	Beauty	1	25	
7	363	2023-06-03	CUST363	Male	64	Beauty	1	25	
8	371	2023-02-21	CUST371	Female	20	Beauty	1	25	
9	397	2023-03-10	CUST397	Female	30	Beauty	1	25	
10	454	2023-02-22	CUST454	Female	46	Beauty	1	25	

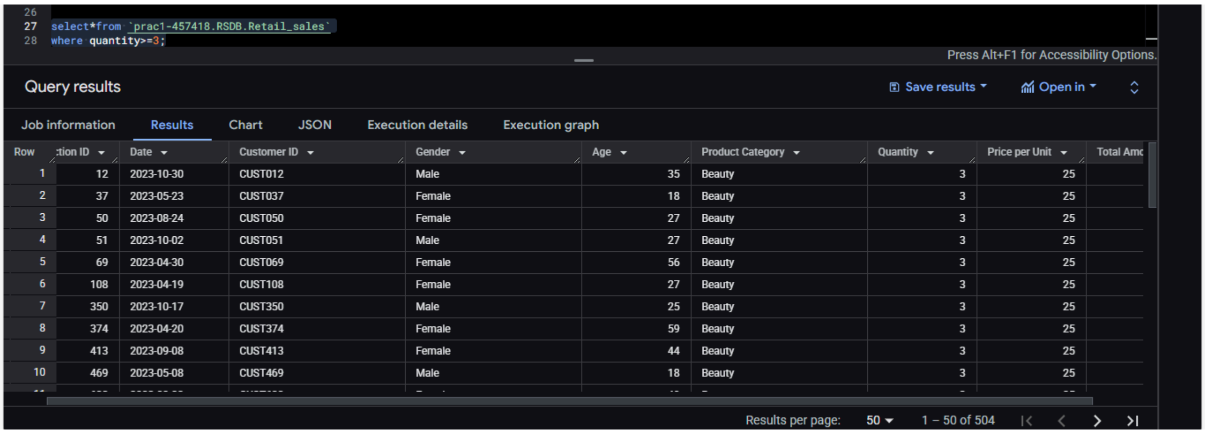
Q8. Display all transactions where the Product Category is not 'Clothing'.
Expected output: All columns



```
select*from `prac1-457418.RSDB.Retail_sales`
where not `product category`='Clothing';
```

Row	n ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Amount
1	191	2023-10-18	CUST191	Male	64	Beauty	1	25	
2	204	2023-09-28	CUST204	Male	39	Beauty	1	25	
3	230	2023-04-23	CUST230	Male	54	Beauty	1	25	
4	232	2023-02-06	CUST232	Female	43	Beauty	1	25	
5	309	2023-12-23	CUST309	Female	26	Beauty	1	25	
6	310	2023-10-12	CUST310	Female	28	Beauty	1	25	
7	363	2023-06-03	CUST363	Male	64	Beauty	1	25	
8	371	2023-02-21	CUST371	Female	20	Beauty	1	25	
9	397	2023-03-10	CUST397	Female	30	Beauty	1	25	
10	454	2023-02-22	CUST454	Female	46	Beauty	1	25	

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



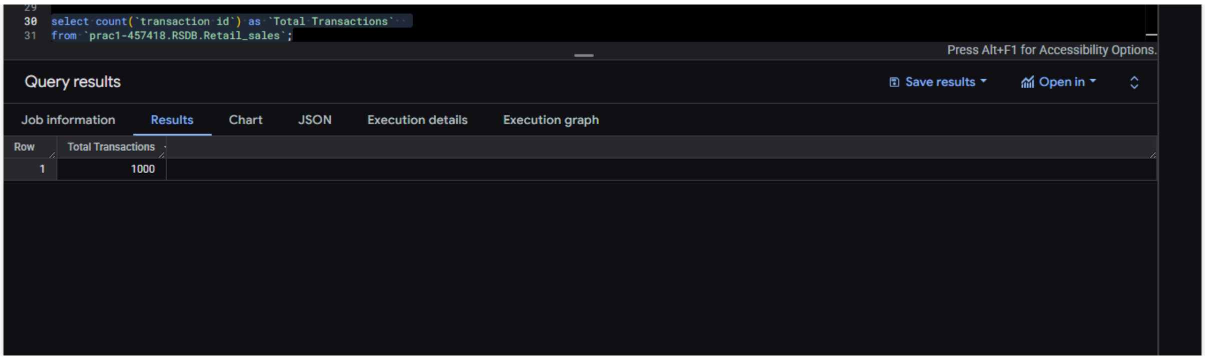
```sql
select*from `prac1-457418.RSDB.Retail_sales`
where quantity>=3;
```

Row	:tion ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Amc
1	12	2023-10-30	CUST012	Male	35	Beauty	3	25	
2	37	2023-05-23	CUST037	Female	18	Beauty	3	25	
3	50	2023-08-24	CUST050	Female	27	Beauty	3	25	
4	51	2023-10-02	CUST051	Male	27	Beauty	3	25	
5	69	2023-04-30	CUST069	Female	56	Beauty	3	25	
6	108	2023-04-19	CUST108	Female	27	Beauty	3	25	
7	350	2023-10-17	CUST350	Male	25	Beauty	3	25	
8	374	2023-04-20	CUST374	Female	59	Beauty	3	25	
9	413	2023-09-08	CUST413	Female	44	Beauty	3	25	
10	469	2023-05-08	CUST469	Male	18	Beauty	3	25	

4. Aggregate Functions

Q10. Count the total number of transactions.

Expected output: Total_Transactions

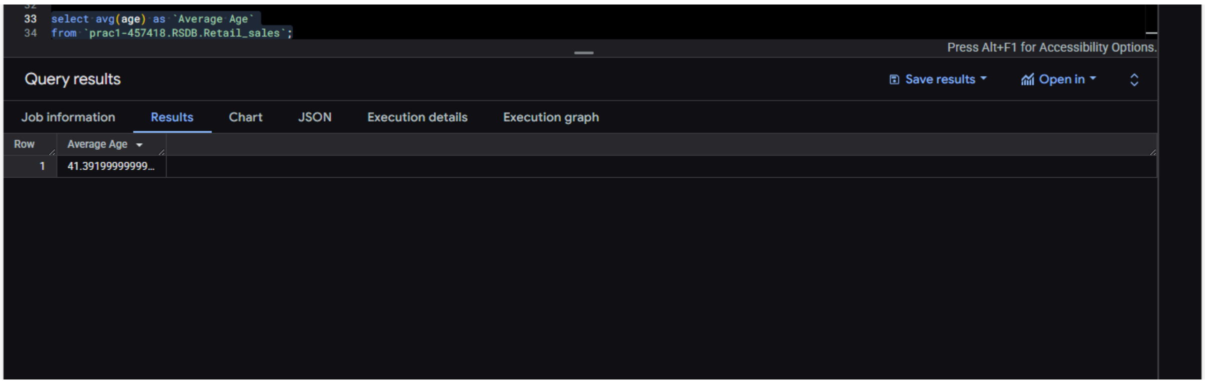


```sql
select count(`transaction id`) as `Total Transactions`
from `prac1-457418.RSDB.Retail_sales`;
```

Row	Total Transactions
1	1000

Q11. Find the average Age of customers.

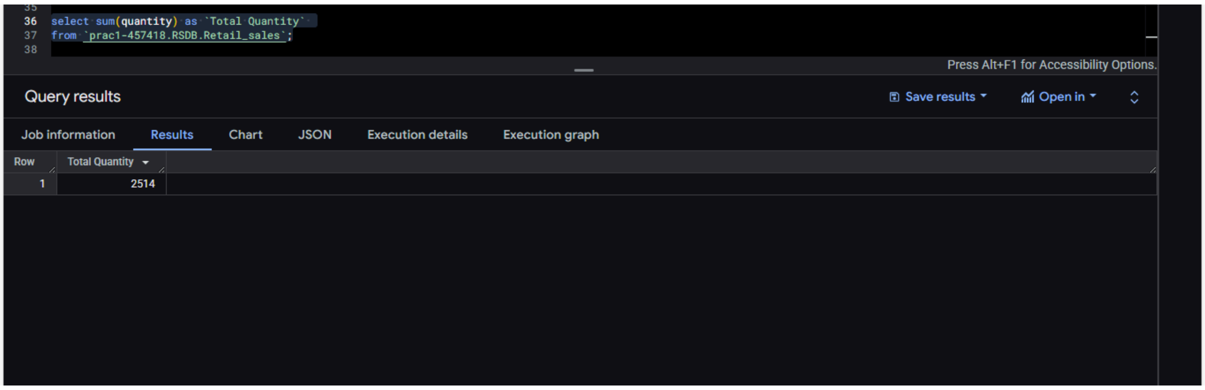
Expected output: Average_Age



```sql
select avg(age) as `Average Age`
from `prac1-457418.RSDB.Retail_sales`;
```

Row	Average Age
1	41.39199999999...

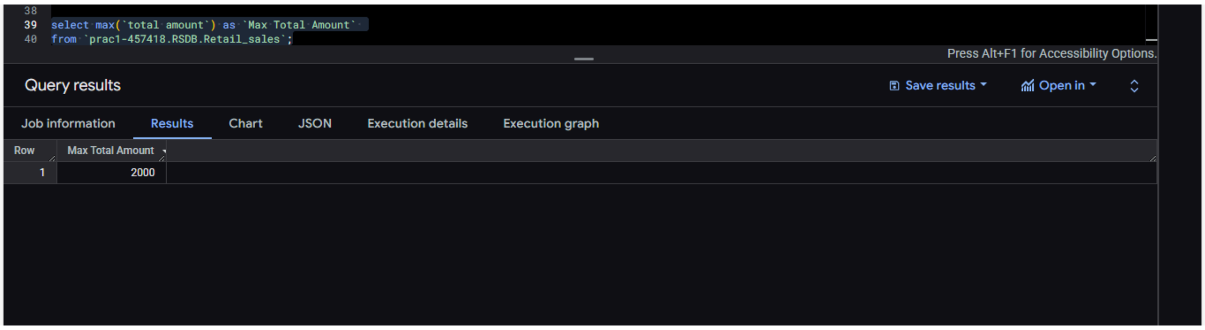
Q12. Find the total quantity of products sold.
Expected output: Total_Quantity



```
select sum(quantity) as `Total Quantity`
from `prac1-457418.RSDB.Retail_sales`;
```

Row	Total Quantity
1	2514

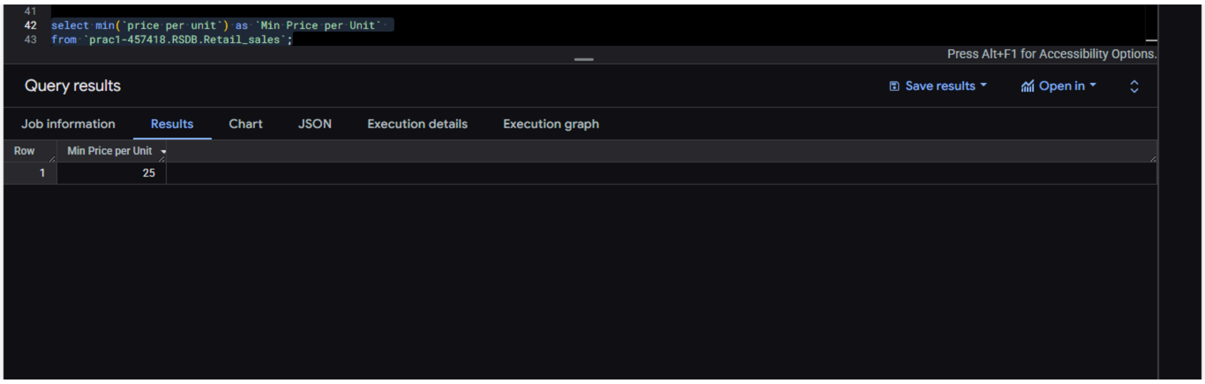
Q13. Find the maximum Total Amount spent in a single transaction.
Expected output: Max_Total_Amount



```
select max(`total amount`) as `Max Total Amount`
from `prac1-457418.RSDB.Retail_sales`;
```

Row	Max Total Amount
1	2000

Q14. Find the minimum Price per Unit in the dataset.
Expected output: Min_Price_per_Unit

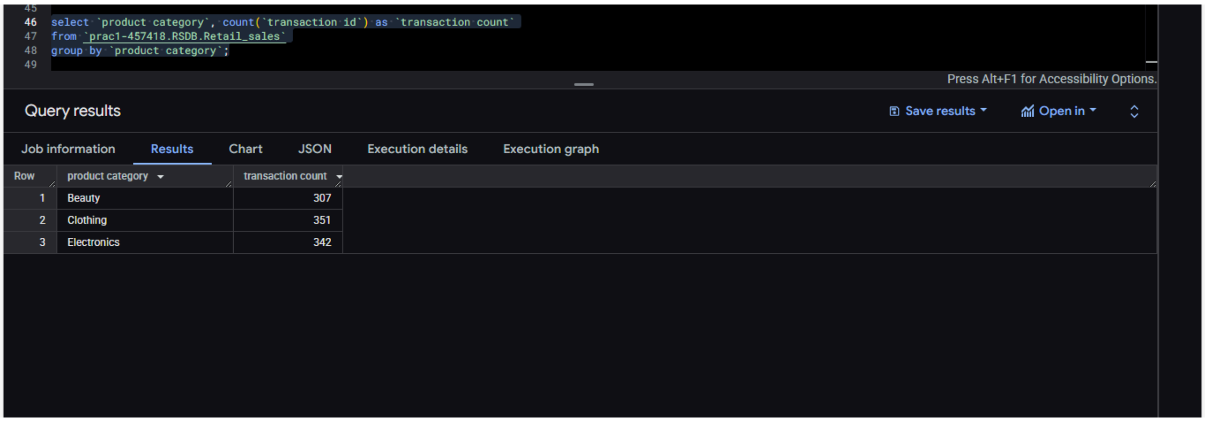


```
select min(`price per unit`) as `Min Price per Unit`
from `prac1-457418.RSDB.Retail_sales`;
```

Row	Min Price per Unit
1	25

5. GROUP BY Statement

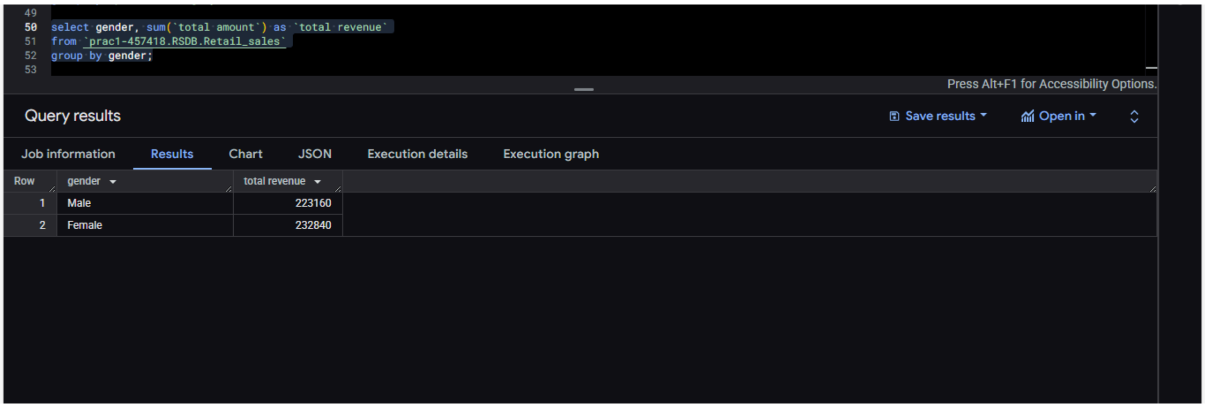
Q15. Find the number of transactions per Product Category.
Expected output: Product Category, Transaction_Count



```
select `product category`, count(`transaction id`) as `transaction count`
from `prac1-457418.RSDB.Retail_sales`
group by `product category`;
```

Row	product category	transaction count
1	Beauty	307
2	Clothing	351
3	Electronics	342

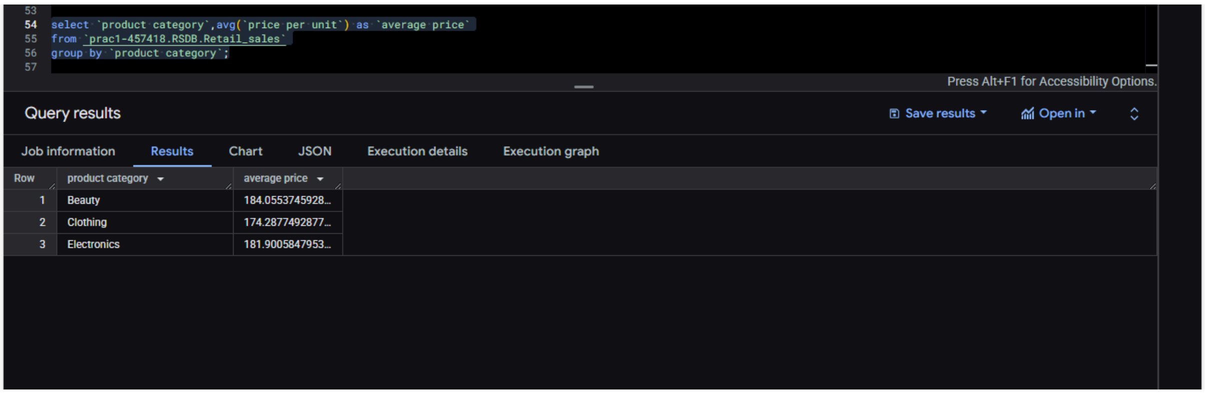
Q16. Find the total revenue (Total Amount) per gender.
Expected output: Gender, Total_Revenue



```
select gender, sum(`total amount`) as `total revenue`
from `prac1-457418.RSDB.Retail_sales`
group by gender;
```

Row	gender	total revenue
1	Male	223160
2	Female	232840

Q17. Find the average Price per Unit per product category.
Expected output: Product Category, Average_Price



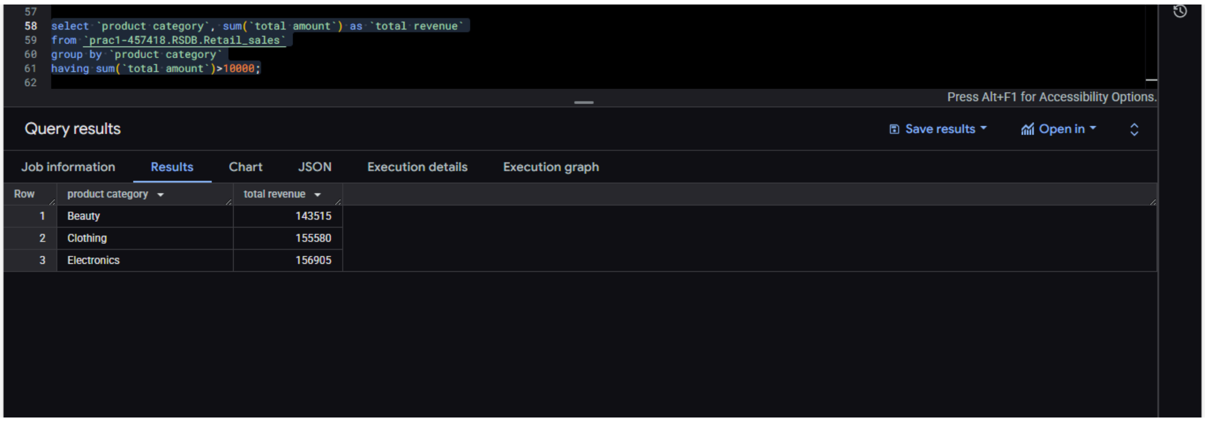
```
select `product category`,avg(`price per unit`) as `average price`
from `prac1-457418.RSDB.Retail_sales`
group by `product category`;
```

Row	product category	average price
1	Beauty	184.0553745928...
2	Clothing	174.2877492877...
3	Electronics	181.9005847953...

6. HAVING Clause

Q18. Find the total revenue per product category where total revenue is greater than 10,000.

Expected output: Product Category, Total_Revenue

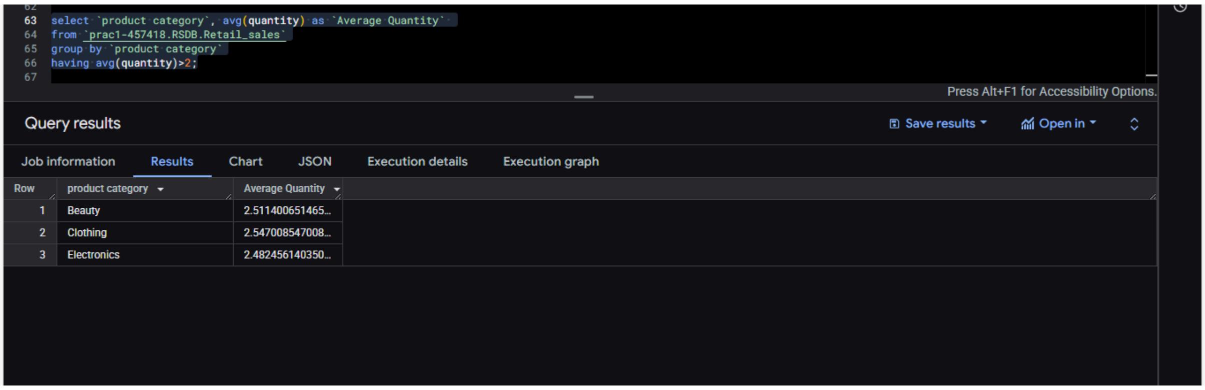


```
select `product category`, sum(`total amount`) as `total revenue`
from `prac1-457418.RSDB.Retail_sales`
group by `product category`
having sum(`total amount`)>10000;
```

Row	product category	total revenue
1	Beauty	143515
2	Clothing	155580
3	Electronics	156905

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

Expected output: Product Category, Average_Quantity



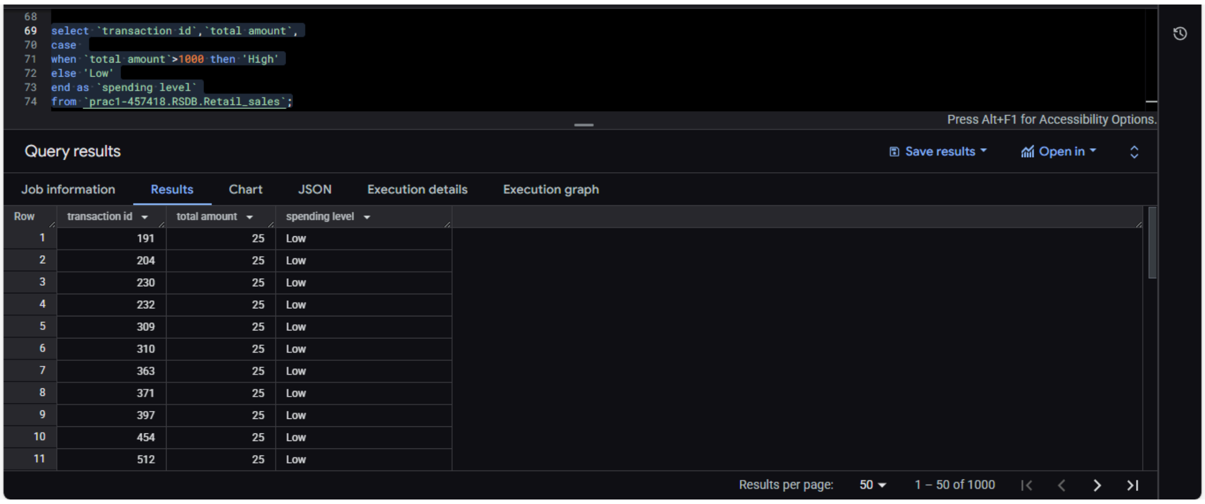
```
select `product category`, avg(quantity) as `Average Quantity`
from `prac1-457418.RSDB.Retail_sales`
group by `product category`
having avg(quantity)>2;
```

Row	product category	Average Quantity
1	Beauty	2.511400651465...
2	Clothing	2.547008547008...
3	Electronics	2.482456140350...

7. CASE Statement

Q20. Display a column called Spending_Level that shows 'High' if Total Amount > 1000, otherwise 'Low'.

Expected output: Transaction ID, Total Amount, Spending_Level



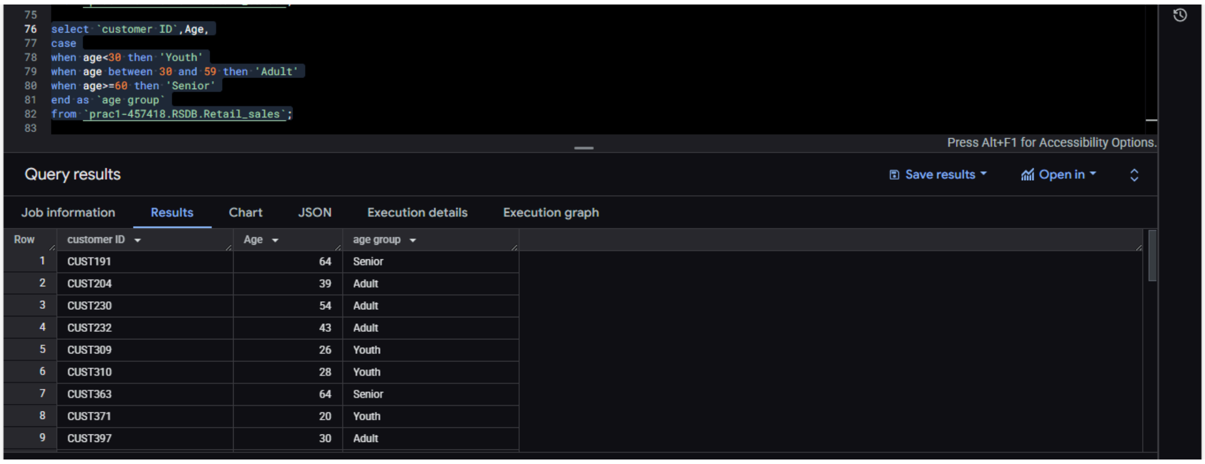
```
select `transaction id`,`total amount`,
case
when `total amount`>1000 then 'High'
else 'Low'
end as `spending level`
from `prac1-457418.RSDB.Retail_sales`;
```

Row	transaction id	total amount	spending level
1	191	25	Low
2	204	25	Low
3	230	25	Low
4	232	25	Low
5	309	25	Low
6	310	25	Low
7	363	25	Low
8	371	25	Low
9	397	25	Low
10	454	25	Low
11	512	25	Low

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 `prac1-457418.RSDB.Retail_sales`;
```

Row	customer ID	Age	age group
1	CUST191	64	Senior
2	CUST204	39	Adult
3	CUST230	54	Adult
4	CUST232	43	Adult
5	CUST309	26	Youth
6	CUST310	28	Youth
7	CUST363	64	Senior
8	CUST371	20	Youth
9	CUST397	30	Adult