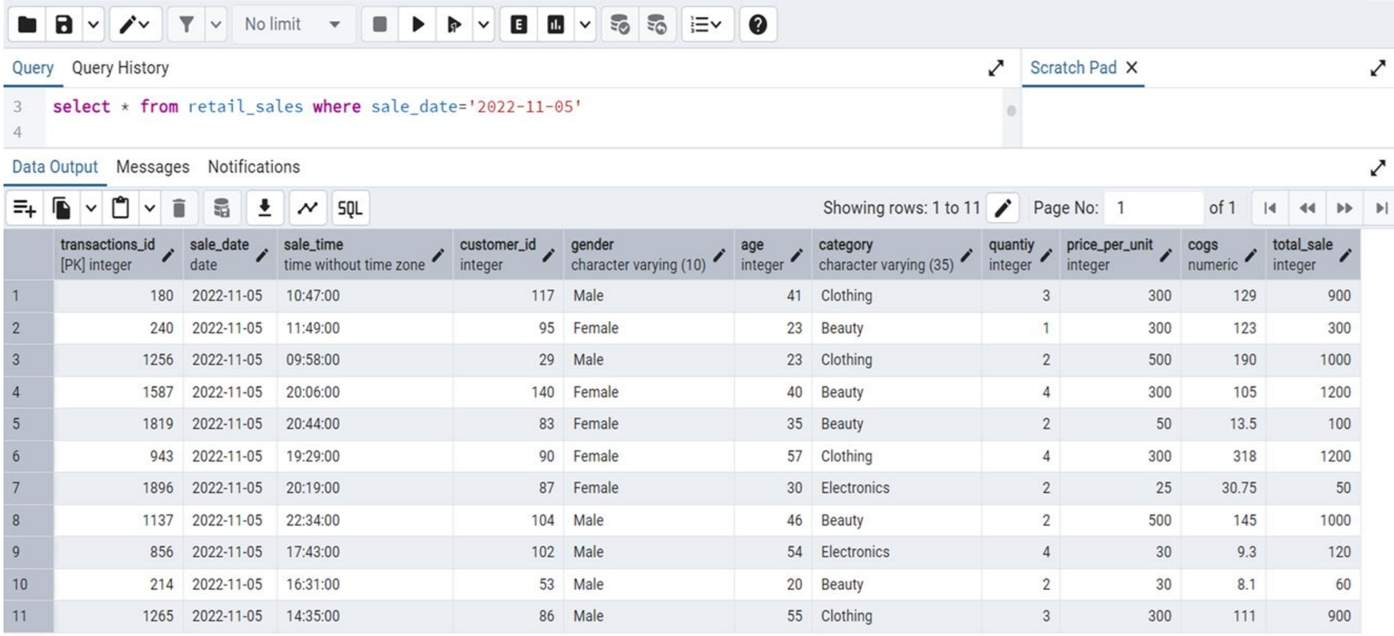
1. Write a SQL query to retrieve all columns for sales made on '2022-11-05

→ select * from retail_sales where sale_date='2022-11-05'



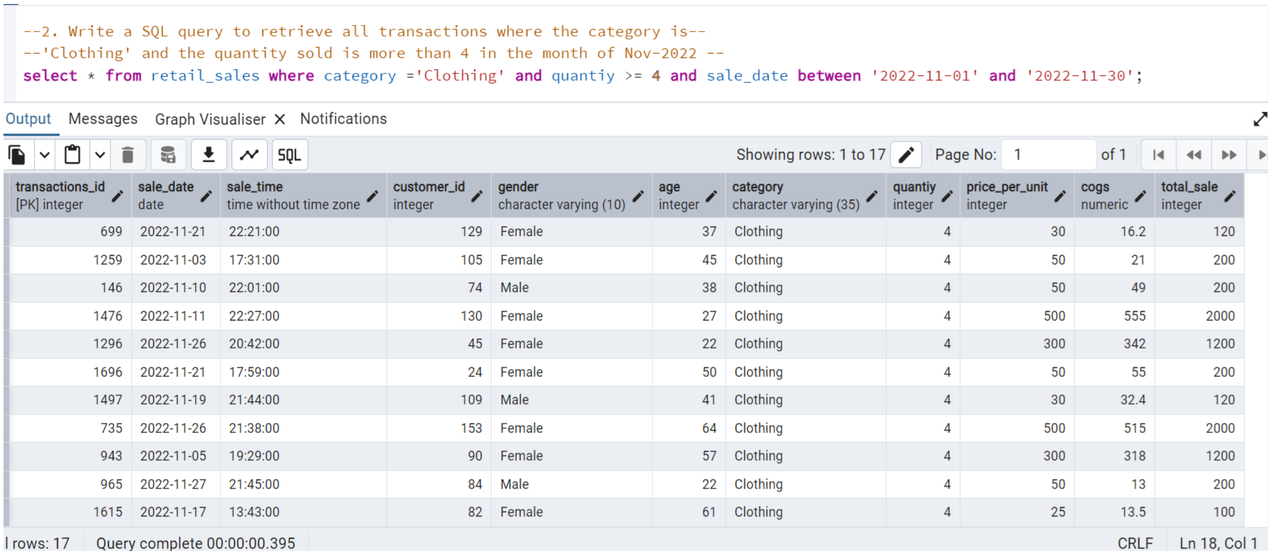
```
select * from retail_sales where sale_date='2022-11-05'
```

Showing rows: 1 to 11 Page No: 1 of 1

	transactions_id [PK] integer	sale_date date	sale_time time without time zone	customer_id integer	gender character varying (10)	age integer	category character varying (35)	quantiy integer	price_per_unit integer	cogs numeric	total_sale integer
1	180	2022-11-05	10:47:00	117	Male	41	Clothing	3	300	129	900
2	240	2022-11-05	11:49:00	95	Female	23	Beauty	1	300	123	300
3	1256	2022-11-05	09:58:00	29	Male	23	Clothing	2	500	190	1000
4	1587	2022-11-05	20:06:00	140	Female	40	Beauty	4	300	105	1200
5	1819	2022-11-05	20:44:00	83	Female	35	Beauty	2	50	13.5	100
6	943	2022-11-05	19:29:00	90	Female	57	Clothing	4	300	318	1200
7	1896	2022-11-05	20:19:00	87	Female	30	Electronics	2	25	30.75	50
8	1137	2022-11-05	22:34:00	104	Male	46	Beauty	2	500	145	1000
9	856	2022-11-05	17:43:00	102	Male	54	Electronics	4	30	9.3	120
10	214	2022-11-05	16:31:00	53	Male	20	Beauty	2	30	8.1	60
11	1265	2022-11-05	14:35:00	86	Male	55	Clothing	3	300	111	900

2. Write a SQL query to retrieve all transactions where the category is 'Clothing' and the quantity sold is more than 4 in the month of Nov-2022

→select * from retail_sales where category ='Clothing' and quantiy > 4 and sale_date between '2022-11-01' and '2022-11-30';



```
--2. Write a SQL query to retrieve all transactions where the category is--
--'Clothing' and the quantity sold is more than 4 in the month of Nov-2022 --
select * from retail_sales where category ='Clothing' and quantiy >= 4 and sale_date between '2022-11-01' and '2022-11-30';
```

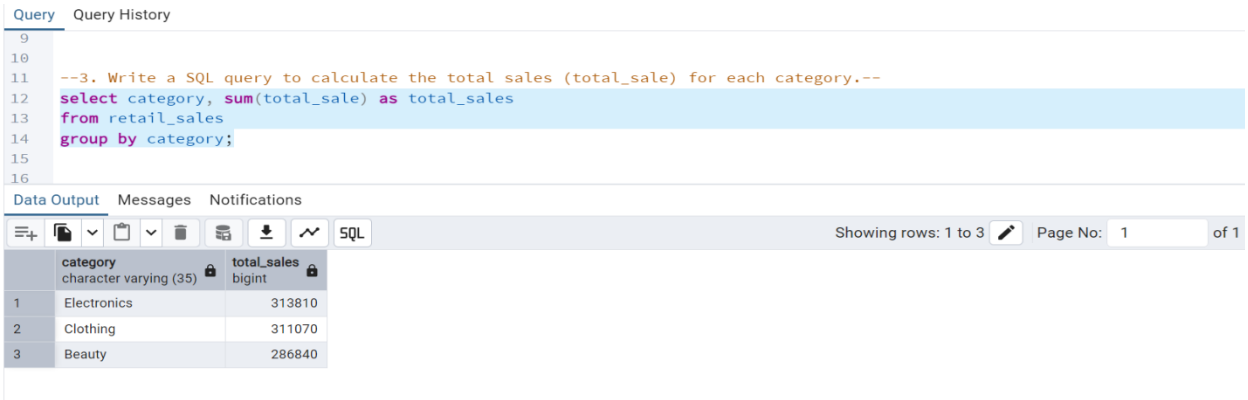
Showing rows: 1 to 17 Page No: 1 of 1

transactions_id [PK] integer	sale_date date	sale_time time without time zone	customer_id integer	gender character varying (10)	age integer	category character varying (35)	quantiy integer	price_per_unit integer	cogs numeric	total_sale integer
699	2022-11-21	22:21:00	129	Female	37	Clothing	4	30	16.2	120
1259	2022-11-03	17:31:00	105	Female	45	Clothing	4	50	21	200
146	2022-11-10	22:01:00	74	Male	38	Clothing	4	50	49	200
1476	2022-11-11	22:27:00	130	Female	27	Clothing	4	500	555	2000
1296	2022-11-26	20:42:00	45	Female	22	Clothing	4	300	342	1200
1696	2022-11-21	17:59:00	24	Female	50	Clothing	4	50	55	200
1497	2022-11-19	21:44:00	109	Male	41	Clothing	4	30	32.4	120
735	2022-11-26	21:38:00	153	Female	64	Clothing	4	500	515	2000
943	2022-11-05	19:29:00	90	Female	57	Clothing	4	300	318	1200
965	2022-11-27	21:45:00	84	Male	22	Clothing	4	50	13	200
1615	2022-11-17	13:43:00	82	Female	61	Clothing	4	25	13.5	100

l rows: 17 Query complete 00:00:00.395 CRLF Ln 18, Col 1

3. Write a SQL query to calculate the total sales (total_sale) for each category.

→ select category, sum(total_sale) as total_sales from retail_sales group by category;

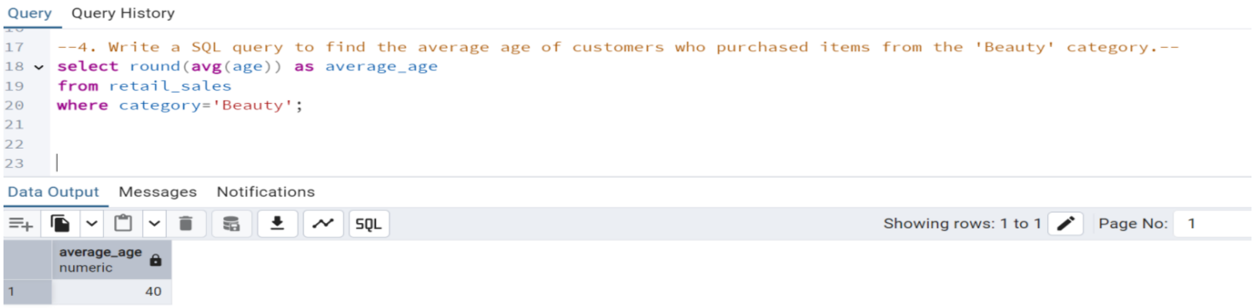


```sql
--3. Write a SQL query to calculate the total sales (total_sale) for each category.--
select category, sum(total_sale) as total_sales
from retail_sales
group by category;
```

	category character varying (35)	total_sales bigint
1	Electronics	313810
2	Clothing	311070
3	Beauty	286840

4. Write a SQL query to find the average age of customers who purchased items from the 'Beauty' category.

→select round(avg(age)) as average_age from retail_sales where category='Beauty';

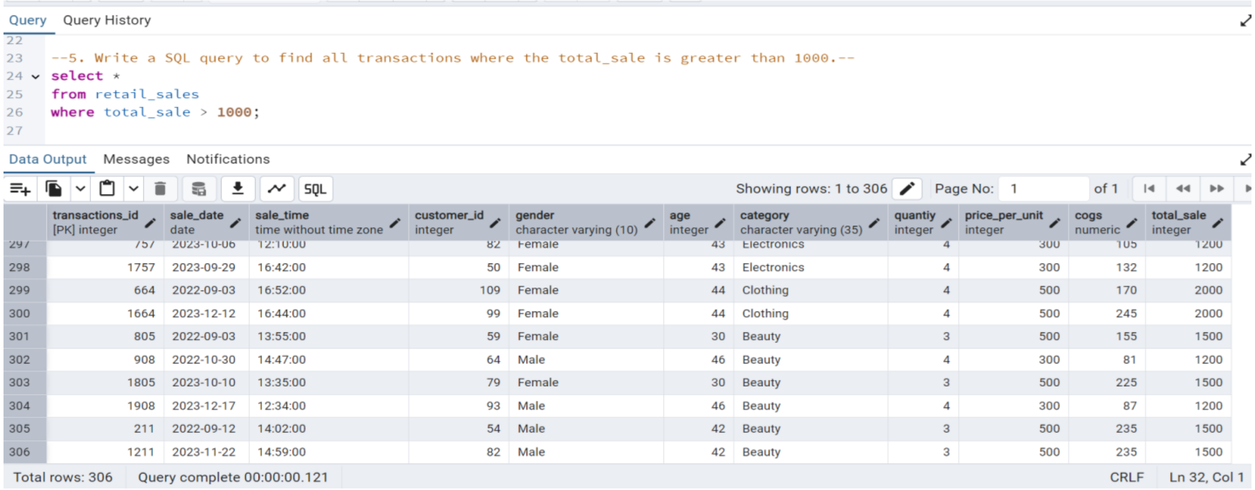


```sql
--4. Write a SQL query to find the average age of customers who purchased items from the 'Beauty' category.--
select round(avg(age)) as average_age
from retail_sales
where category='Beauty';
```

	average_age numeric
1	40

5. Write a SQL query to find all transactions where the total_sale is greater than 1000.

→select * from retail_sales where total_sale > 1000;



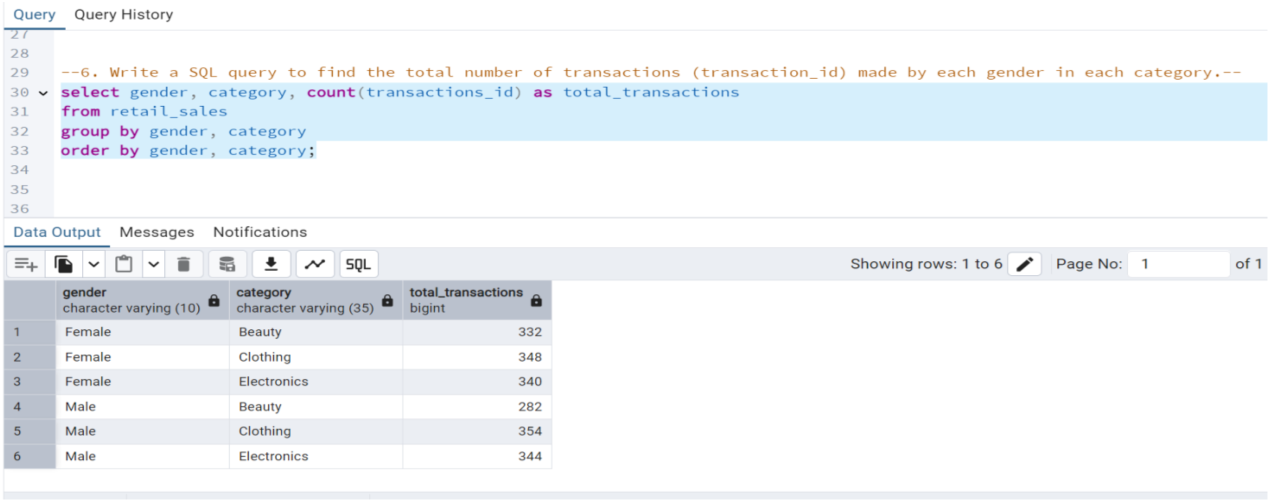
```sql
--5. Write a SQL query to find all transactions where the total_sale is greater than 1000.--
select *
from retail_sales
where total_sale > 1000;
```

	transactions_id [PK] integer	sale_date date	sale_time time without time zone	customer_id integer	gender character varying (10)	age integer	category character varying (35)	quantiy integer	price_per_unit integer	cogs numeric	total_sale integer
297	757	2023-10-06	12:10:00	82	Female	43	Electronics	4	300	105	1200
298	1757	2023-09-29	16:42:00	50	Female	43	Electronics	4	300	132	1200
299	664	2022-09-03	16:52:00	109	Female	44	Clothing	4	500	170	2000
300	1664	2023-12-12	16:44:00	99	Female	44	Clothing	4	500	245	2000
301	805	2022-09-03	13:55:00	59	Female	30	Beauty	3	500	155	1500
302	908	2022-10-30	14:47:00	64	Male	46	Beauty	4	300	81	1200
303	1805	2023-10-10	13:35:00	79	Female	30	Beauty	3	500	225	1500
304	1908	2023-12-17	12:34:00	93	Male	46	Beauty	4	300	87	1200
305	211	2022-09-12	14:02:00	54	Male	42	Beauty	3	500	235	1500
306	1211	2023-11-22	14:59:00	82	Male	42	Beauty	3	500	235	1500

Total rows: 306   Query complete 00:00:00.121   CRLF   Ln 32, Col 1

6. Write a SQL query to find the total number of transactions (transaction_id) made by each gender in each category.

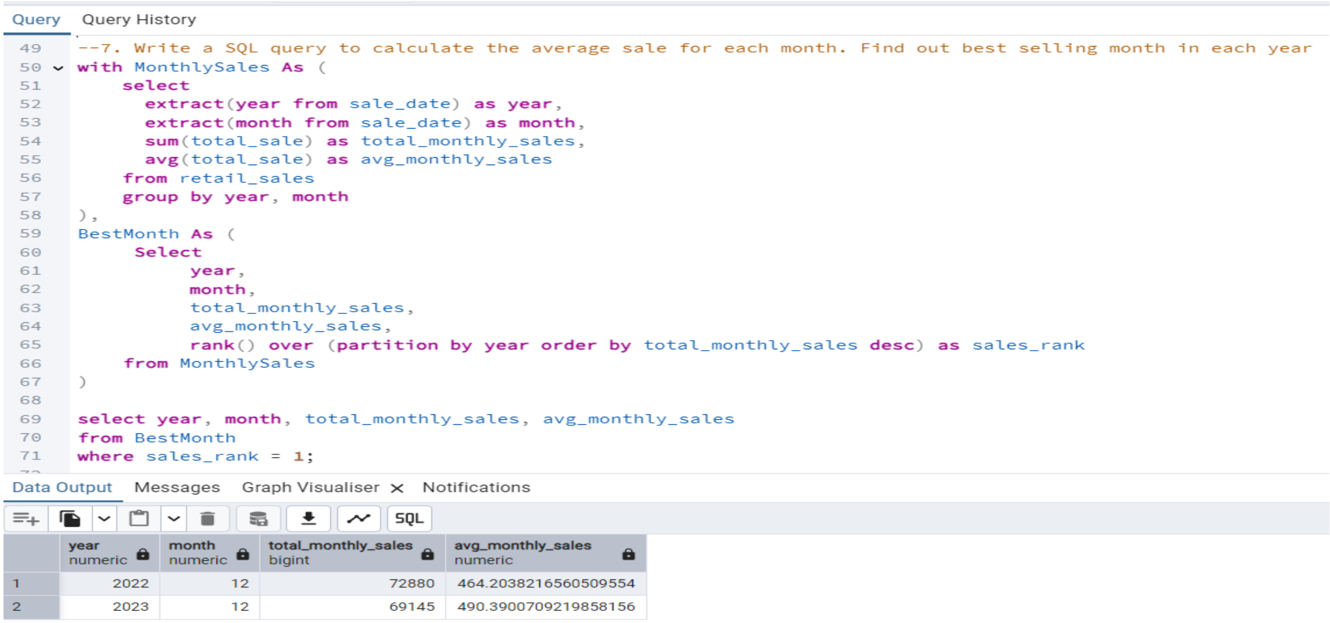
→select gender, category, count(transactions_id) as total_transactions from retail_sales group by gender, category order by gender, category;



```
--6. Write a SQL query to find the total number of transactions (transaction_id) made by each gender in each category.--
select gender, category, count(transactions_id) as total_transactions
from retail_sales
group by gender, category
order by gender, category;
```

	gender character varying (10)	category character varying (35)	total_transactions bigint
1	Female	Beauty	332
2	Female	Clothing	348
3	Female	Electronics	340
4	Male	Beauty	282
5	Male	Clothing	354
6	Male	Electronics	344

7. Write a SQL query to calculate the average sale for each month. Find out best selling month in each year

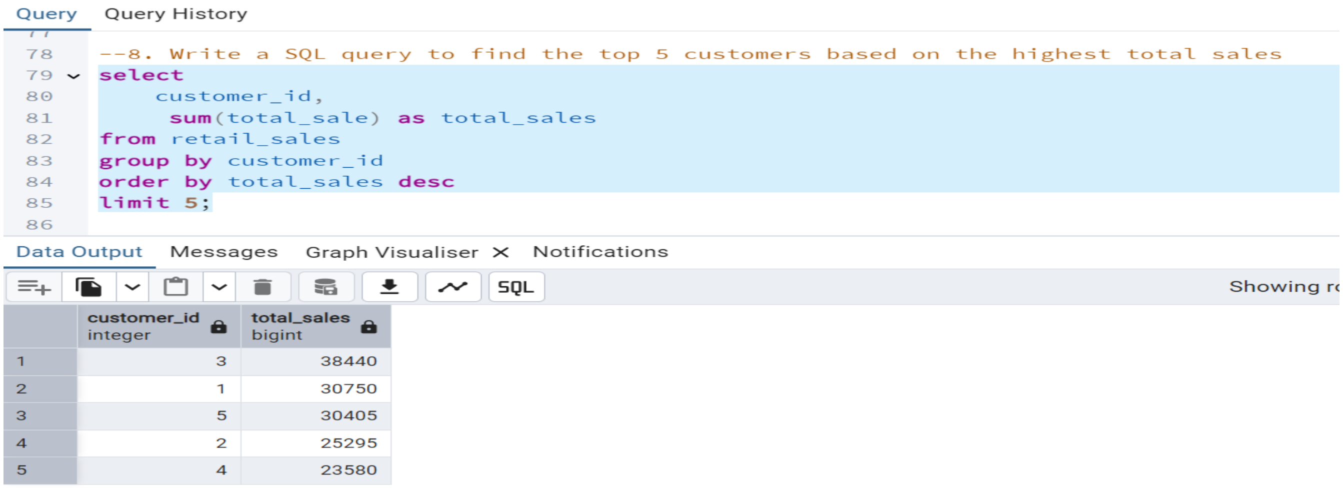


```
--7. Write a SQL query to calculate the average sale for each month. Find out best selling month in each year
with MonthlySales As (
    select
      extract(year from sale_date) as year,
      extract(month from sale_date) as month,
      sum(total_sale) as total_monthly_sales,
      avg(total_sale) as avg_monthly_sales
    from retail_sales
    group by year, month
),
BestMonth As (
     Select
          year,
          month,
          total_monthly_sales,
          avg_monthly_sales,
          rank() over (partition by year order by total_monthly_sales desc) as sales_rank
    from MonthlySales
)

select year, month, total_monthly_sales, avg_monthly_sales
from BestMonth
where sales_rank = 1;
```

	year numeric	month numeric	total_monthly_sales bigint	avg_monthly_sales numeric
1	2022	12	72880	464.2038216560509554
2	2023	12	69145	490.3900709219858156

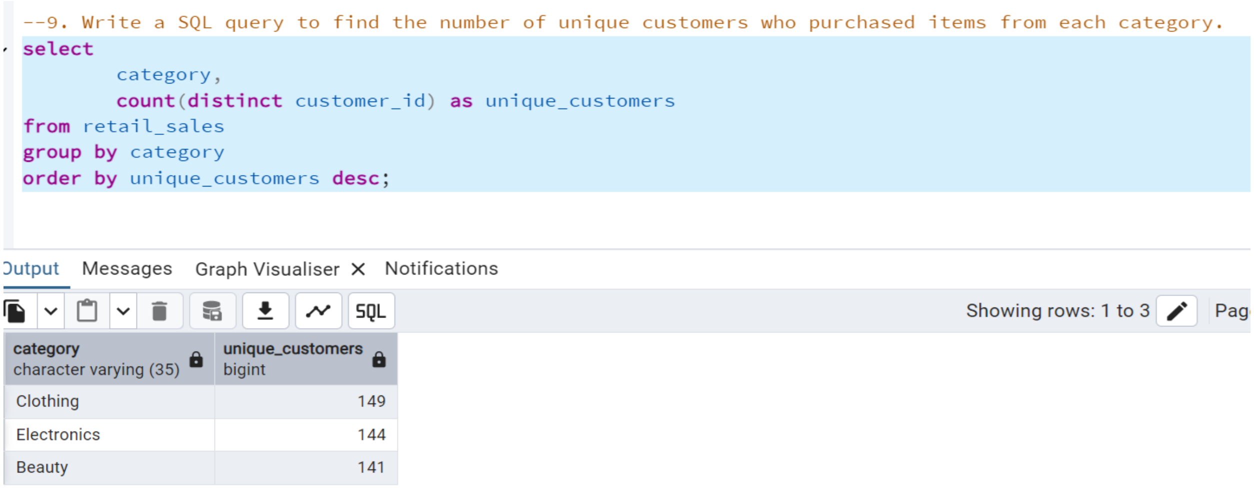
8. Write a SQL query to find the top 5 customers based on the highest total sales



```sql
--8. Write a SQL query to find the top 5 customers based on the highest total sales
select
    customer_id,
     sum(total_sale) as total_sales
from retail_sales
group by customer_id
order by total_sales desc
limit 5;
```

	customer_id integer	total_sales bigint
1	3	38440
2	1	30750
3	5	30405
4	2	25295
5	4	23580

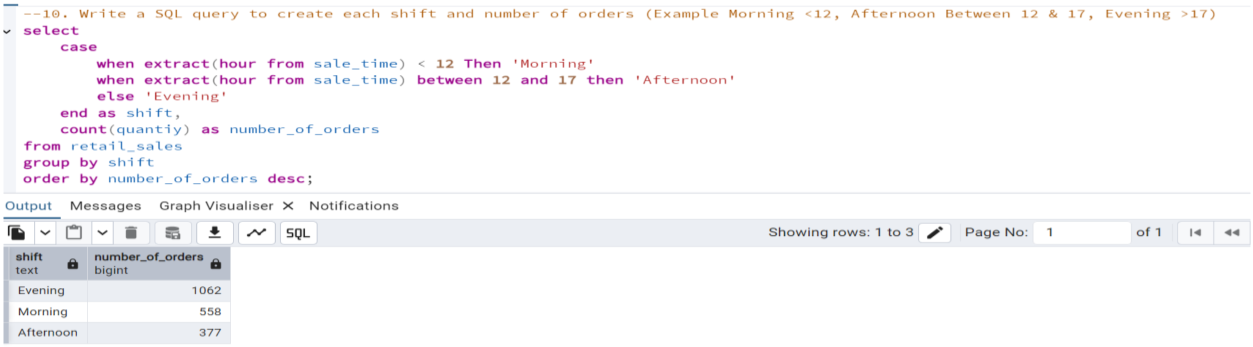
9. Write a SQL query to find the number of unique customers who purchased items from each category.



```sql
--9. Write a SQL query to find the number of unique customers who purchased items from each category.
select
        category,
        count(distinct customer_id) as unique_customers
from retail_sales
group by category
order by unique_customers desc;
```

category character varying (35)	unique_customers bigint
Clothing	149
Electronics	144
Beauty	141

10. Write a SQL query to create each shift and number of orders (Example Morning <12, Afternoon Between 12 & 17, Evening >17)



```sql
--10. Write a SQL query to create each shift and number of orders (Example Morning <12, Afternoon Between 12 & 17, Evening >17)
select
    case
        when extract(hour from sale_time) < 12 Then 'Morning'
        when extract(hour from sale_time) between 12 and 17 then 'Afternoon'
        else 'Evening'
    end as shift,
    count(quantiy) as number_of_orders
from retail_sales
group by shift
order by number_of_orders desc;
```

shift text	number_of_orders bigint
Evening	1062
Morning	558
Afternoon	377