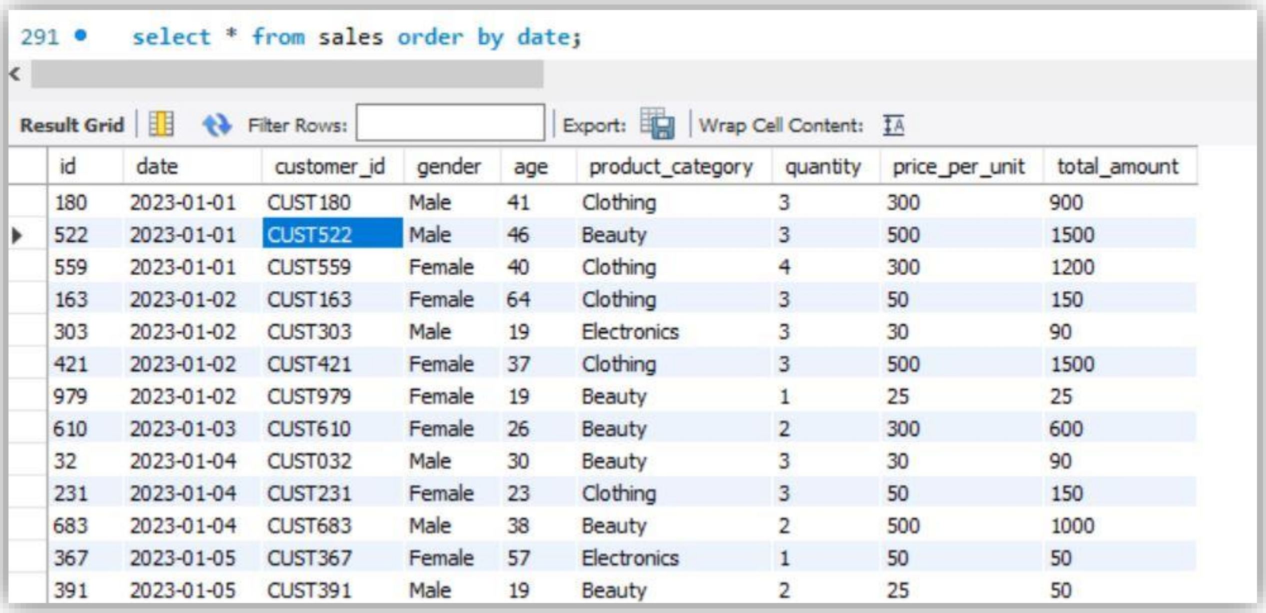
Accomplished by Aruzhan S.

Portfolio project 1
Retail Sales SQL project

The main goal of this project is to analyze sales in retail store. A data was taken from Kaggle and downloaded as one csv file. Then this data was transferred to MySQL. This work considers five main questions, which were already prepared and attached to data itself.

Firstly, *retail_sales* database was created. All downloaded data was transferred to *sales* table via "Table Data Import Wizard":



```
select * from sales order by date;
```

id	date	customer_id	gender	age	product_category	quantity	price_per_unit	total_amount
180	2023-01-01	CUST180	Male	41	Clothing	3	300	900
522	2023-01-01	CUST522	Male	46	Beauty	3	500	1500
559	2023-01-01	CUST559	Female	40	Clothing	4	300	1200
163	2023-01-02	CUST163	Female	64	Clothing	3	50	150
303	2023-01-02	CUST303	Male	19	Electronics	3	30	90
421	2023-01-02	CUST421	Female	37	Clothing	3	500	1500
979	2023-01-02	CUST979	Female	19	Beauty	1	25	25
610	2023-01-03	CUST610	Female	26	Beauty	2	300	600
32	2023-01-04	CUST032	Male	30	Beauty	3	30	90
231	2023-01-04	CUST231	Female	23	Clothing	3	50	150
683	2023-01-04	CUST683	Male	38	Beauty	2	500	1000
367	2023-01-05	CUST367	Female	57	Electronics	1	50	50
391	2023-01-05	CUST391	Male	19	Beauty	2	25	50

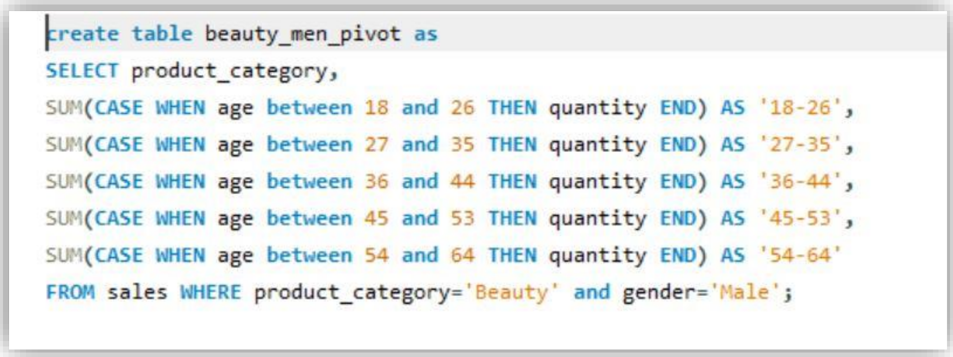
Figure 1. *sales* table

As it can be seen in Figure 1, initially, 9 columns were provided: id (transaction id), date (when transaction was happened), customer_id, gender, age, product_category (this store has three categories such as beauty, clothing and electronics), quantity (purchased quantity of products), price_per_unit and total_amount (total spending of each customer). A query in Figure 1 returns all data ordered by date.

Here are questions to answer:

1) How does customer age and gender influence their purchasing behavior?

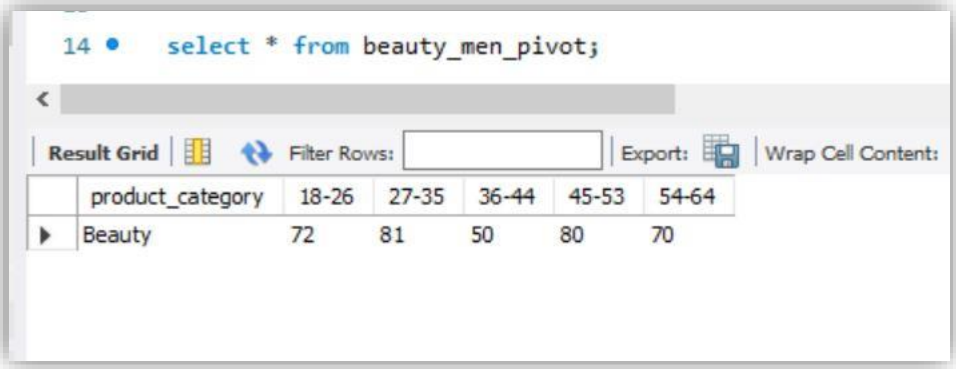
An answer to the first question considers relationship between customer age, gender and total purchased quantity. Five age groups were considered for each product category and gender. **CASE** statements allow to observe all age groups. Additionally, **SUM** function was applied to identify total quantity of purchased products for each age group.



```
create table beauty_men_pivot as
SELECT product_category,
SUM(CASE WHEN age between 18 and 26 THEN quantity END) AS '18-26',
SUM(CASE WHEN age between 27 and 35 THEN quantity END) AS '27-35',
SUM(CASE WHEN age between 36 and 44 THEN quantity END) AS '36-44',
SUM(CASE WHEN age between 45 and 53 THEN quantity END) AS '45-53',
SUM(CASE WHEN age between 54 and 64 THEN quantity END) AS '54-64'
FROM sales WHERE product_category='Beauty' and gender='Male';
```

Figure 2. *beauty_men_pivot* table

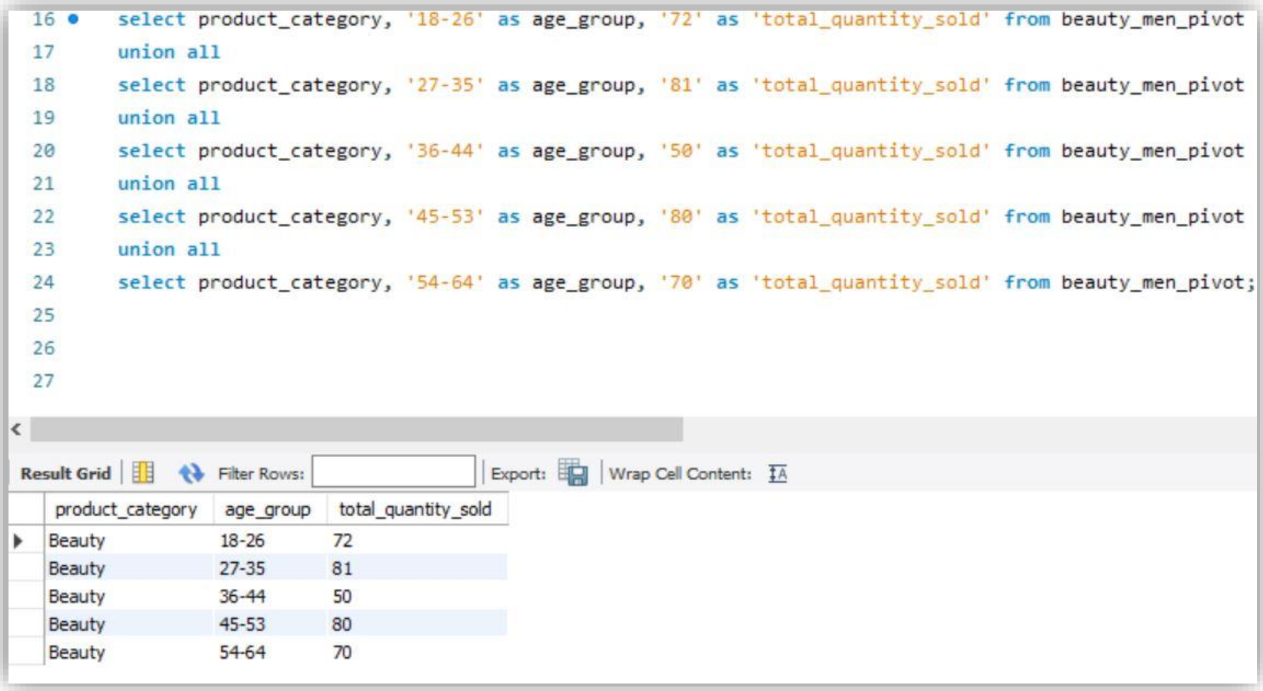
A result of the query in Figure 2 is displayed as pivot table:



```
select * from beauty_men_pivot;
```

product_category	18-26	27-35	36-44	45-53	54-64
Beauty	72	81	50	80	70

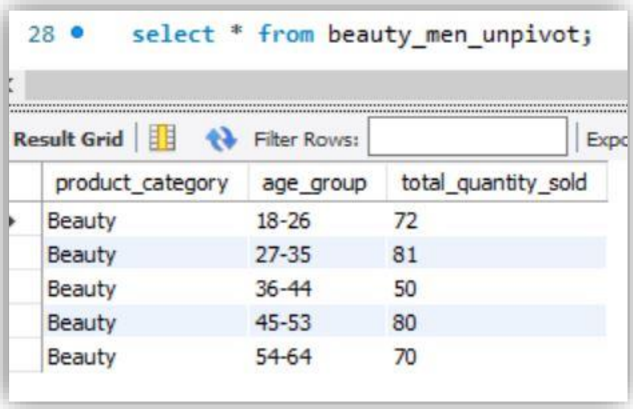
Therefore, unpivoting is needed:



```
select product_category, '18-26' as age_group, '72' as 'total_quantity_sold' from beauty_men_pivot
union all
select product_category, '27-35' as age_group, '81' as 'total_quantity_sold' from beauty_men_pivot
union all
select product_category, '36-44' as age_group, '50' as 'total_quantity_sold' from beauty_men_pivot
union all
select product_category, '45-53' as age_group, '80' as 'total_quantity_sold' from beauty_men_pivot
union all
select product_category, '54-64' as age_group, '70' as 'total_quantity_sold' from beauty_men_pivot;
```

product_category	age_group	total_quantity_sold
Beauty	18-26	72
Beauty	27-35	81
Beauty	36-44	50
Beauty	45-53	80
Beauty	54-64	70

UNION ALL operator enables to attach all results from SELECT statements. Figure 3 represents created table, *beauty_men_unpivot*.

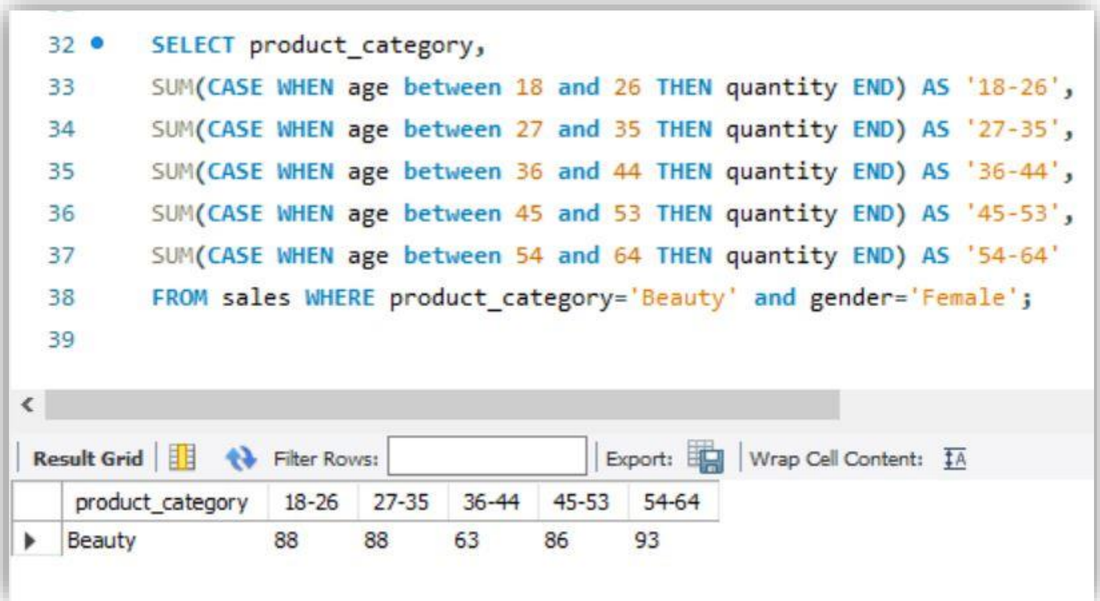


```sql
select * from beauty_men_unpivot;
```

product_category	age_group	total_quantity_sold
Beauty	18-26	72
Beauty	27-35	81
Beauty	36-44	50
Beauty	45-53	80
Beauty	54-64	70

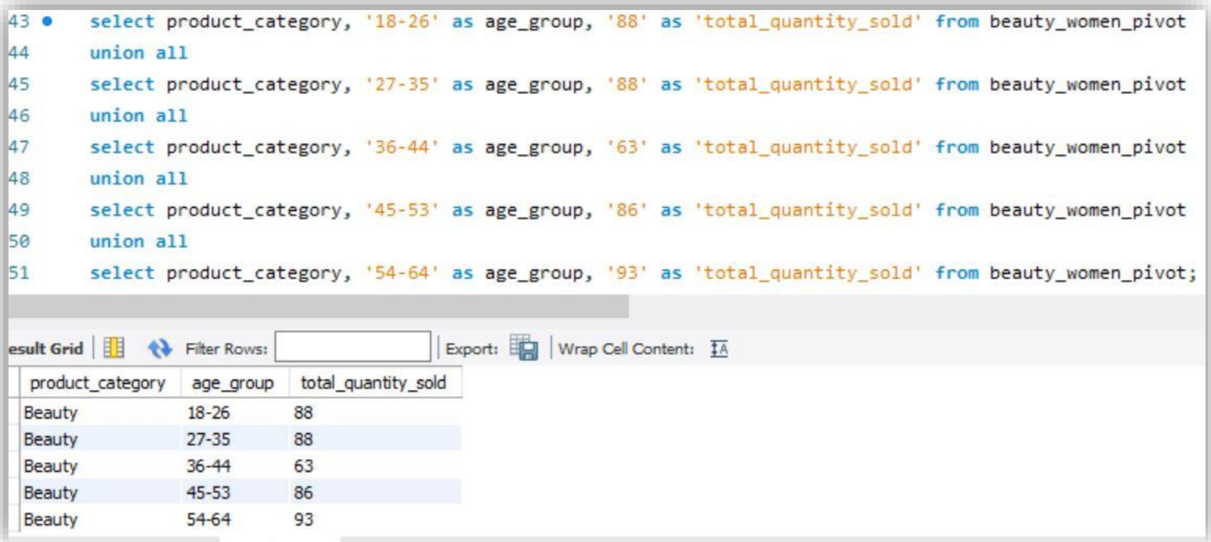
Figure 3. *beauty_men_unpivot* table

The same process was conducted for each product category and gender.



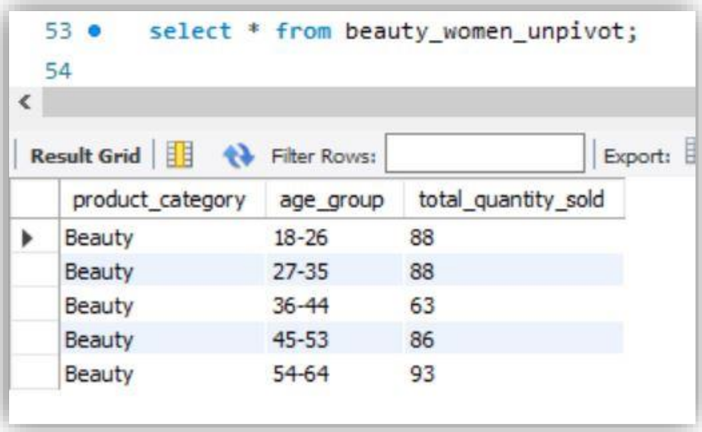
```sql
SELECT product_category,
SUM(CASE WHEN age between 18 and 26 THEN quantity END) AS '18-26',
SUM(CASE WHEN age between 27 and 35 THEN quantity END) AS '27-35',
SUM(CASE WHEN age between 36 and 44 THEN quantity END) AS '36-44',
SUM(CASE WHEN age between 45 and 53 THEN quantity END) AS '45-53',
SUM(CASE WHEN age between 54 and 64 THEN quantity END) AS '54-64'
FROM sales WHERE product_category='Beauty' and gender='Female';
```

product_category	18-26	27-35	36-44	45-53	54-64
Beauty	88	88	63	86	93



```sql
select product_category, '18-26' as age_group, '88' as 'total_quantity_sold' from beauty_women_pivot
union all
select product_category, '27-35' as age_group, '88' as 'total_quantity_sold' from beauty_women_pivot
union all
select product_category, '36-44' as age_group, '63' as 'total_quantity_sold' from beauty_women_pivot
union all
select product_category, '45-53' as age_group, '86' as 'total_quantity_sold' from beauty_women_pivot
union all
select product_category, '54-64' as age_group, '93' as 'total_quantity_sold' from beauty_women_pivot;
```

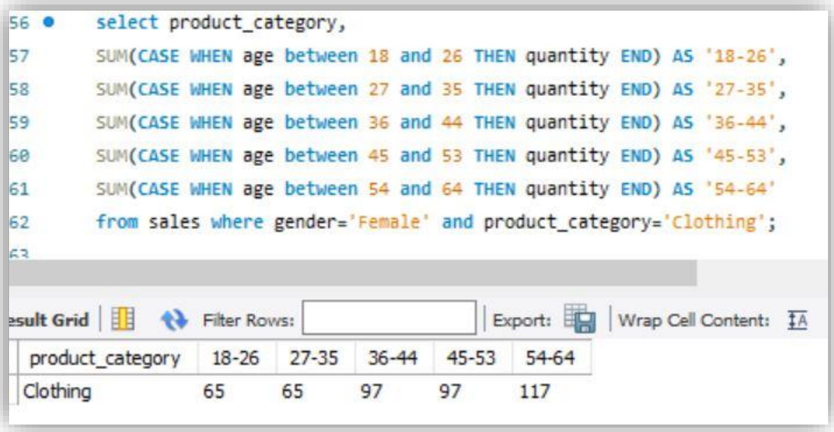
product_category	age_group	total_quantity_sold
Beauty	18-26	88
Beauty	27-35	88
Beauty	36-44	63
Beauty	45-53	86
Beauty	54-64	93



```
select * from beauty_women_unpivot;
```

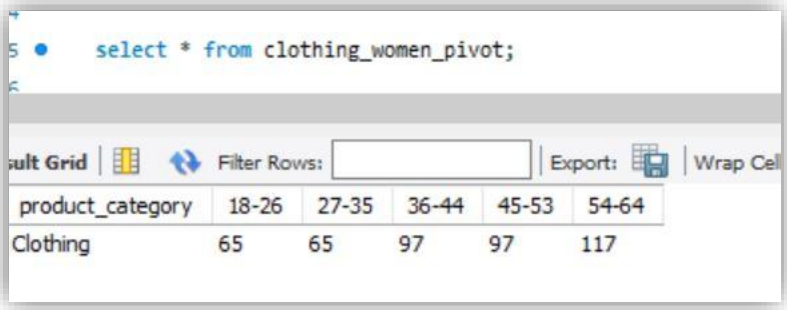
product_category	age_group	total_quantity_sold
Beauty	18-26	88
Beauty	27-35	88
Beauty	36-44	63
Beauty	45-53	86
Beauty	54-64	93

Figure 4. *beauty_women_unpivot* table



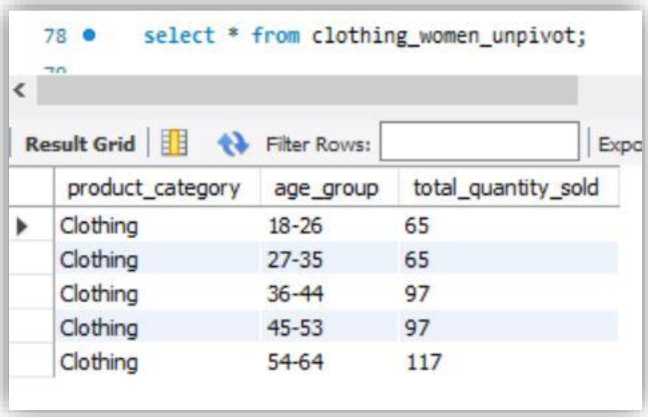
```
select product_category,
SUM(CASE WHEN age between 18 and 26 THEN quantity END) AS '18-26',
SUM(CASE WHEN age between 27 and 35 THEN quantity END) AS '27-35',
SUM(CASE WHEN age between 36 and 44 THEN quantity END) AS '36-44',
SUM(CASE WHEN age between 45 and 53 THEN quantity END) AS '45-53',
SUM(CASE WHEN age between 54 and 64 THEN quantity END) AS '54-64'
from sales where gender='Female' and product_category='Clothing';
```

product_category	18-26	27-35	36-44	45-53	54-64
Clothing	65	65	97	97	117



```
select * from clothing_women_pivot;
```

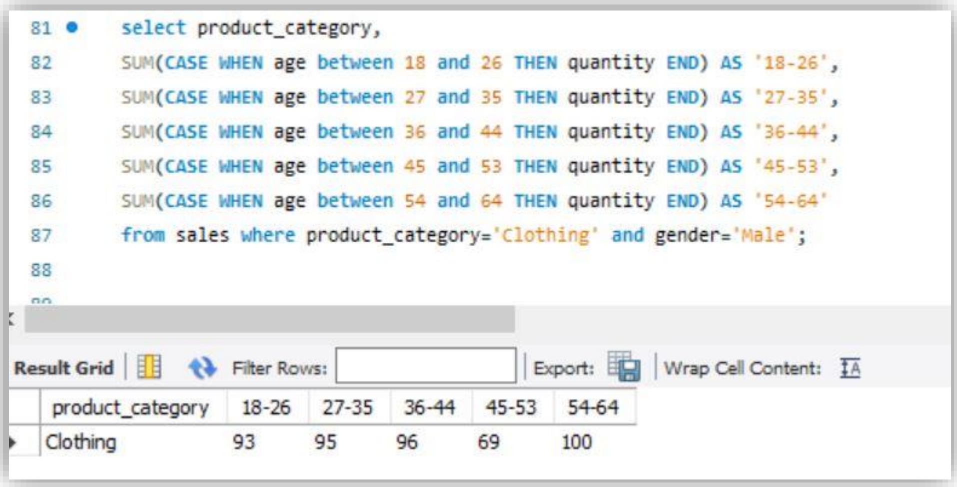
product_category	18-26	27-35	36-44	45-53	54-64
Clothing	65	65	97	97	117



```
select * from clothing_women_unpivot;
```

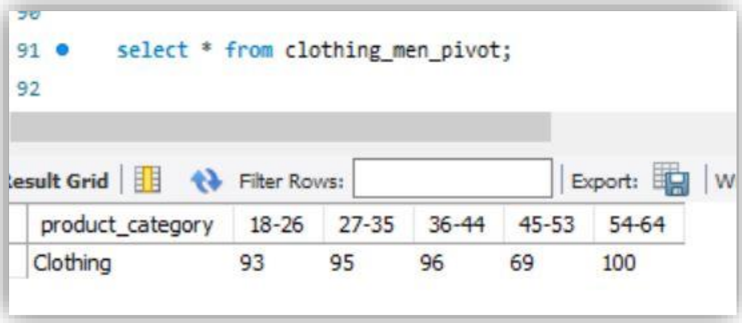
product_category	age_group	total_quantity_sold
Clothing	18-26	65
Clothing	27-35	65
Clothing	36-44	97
Clothing	45-53	97
Clothing	54-64	117

Figure 5. *clothing_women_unpivot* table



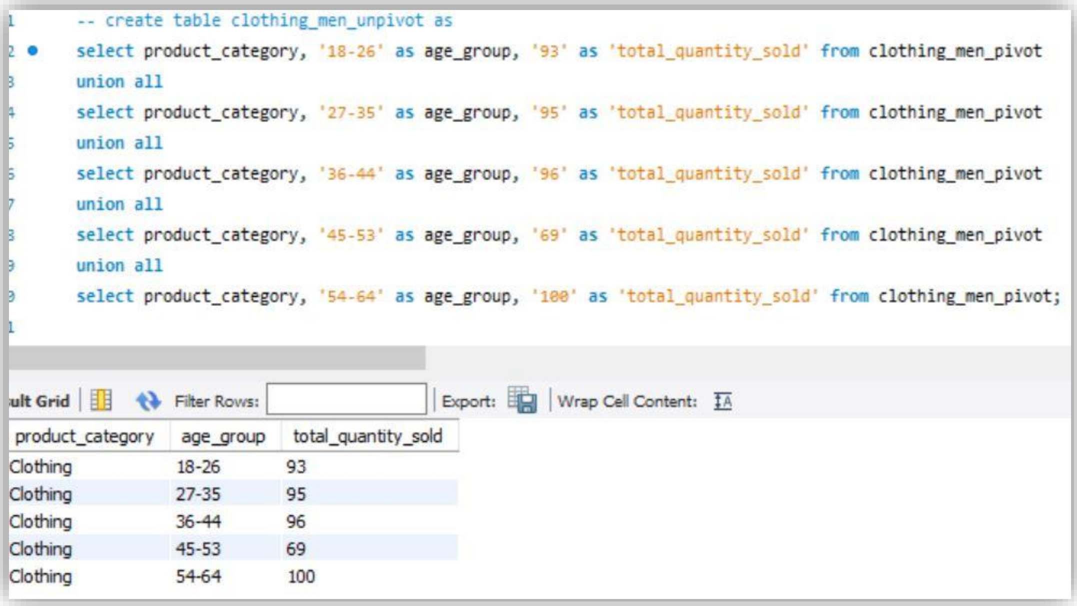
```sql
select product_category,
SUM(CASE WHEN age between 18 and 26 THEN quantity END) AS '18-26',
SUM(CASE WHEN age between 27 and 35 THEN quantity END) AS '27-35',
SUM(CASE WHEN age between 36 and 44 THEN quantity END) AS '36-44',
SUM(CASE WHEN age between 45 and 53 THEN quantity END) AS '45-53',
SUM(CASE WHEN age between 54 and 64 THEN quantity END) AS '54-64'
from sales where product_category='Clothing' and gender='Male';
```

product_category	18-26	27-35	36-44	45-53	54-64
Clothing	93	95	96	69	100



```sql
select * from clothing_men_pivot;
```

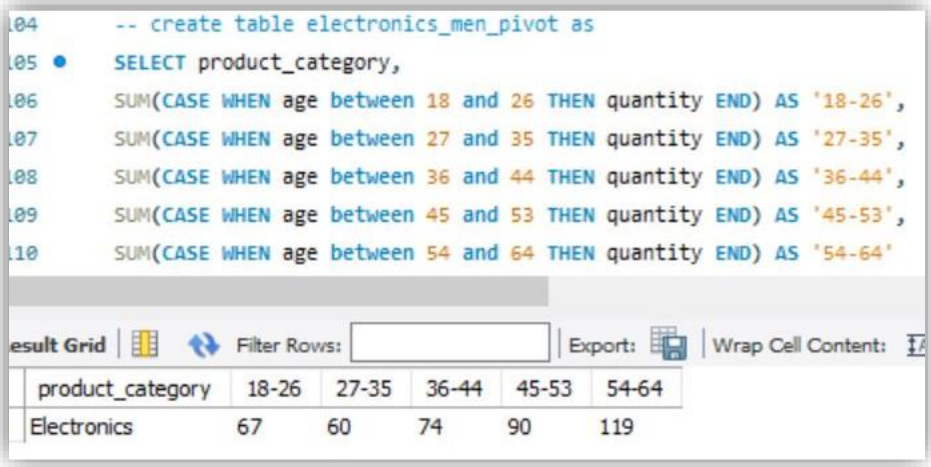
product_category	18-26	27-35	36-44	45-53	54-64
Clothing	93	95	96	69	100



```sql
-- create table clothing_men_unpivot as
select product_category, '18-26' as age_group, '93' as 'total_quantity_sold' from clothing_men_pivot
union all
select product_category, '27-35' as age_group, '95' as 'total_quantity_sold' from clothing_men_pivot
union all
select product_category, '36-44' as age_group, '96' as 'total_quantity_sold' from clothing_men_pivot
union all
select product_category, '45-53' as age_group, '69' as 'total_quantity_sold' from clothing_men_pivot
union all
select product_category, '54-64' as age_group, '100' as 'total_quantity_sold' from clothing_men_pivot;
```

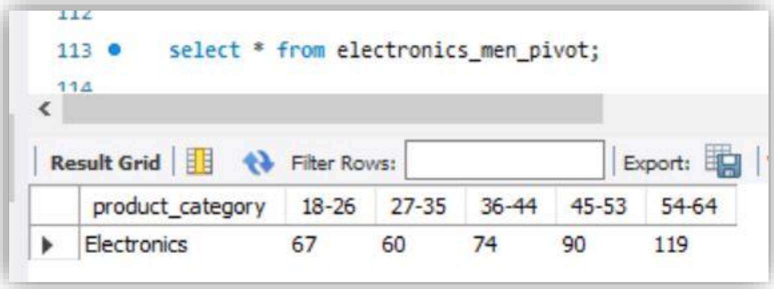
product_category	age_group	total_quantity_sold
Clothing	18-26	93
Clothing	27-35	95
Clothing	36-44	96
Clothing	45-53	69
Clothing	54-64	100

Figure 6. *clothing_men_unpivot* table



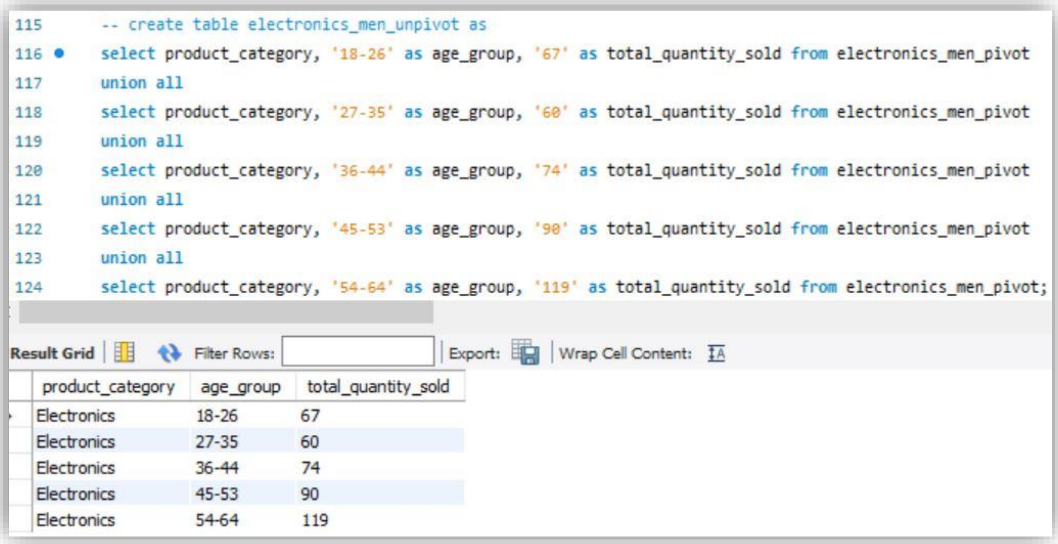
```
-- create table electronics_men_pivot as
SELECT product_category,
SUM(CASE WHEN age between 18 and 26 THEN quantity END) AS '18-26',
SUM(CASE WHEN age between 27 and 35 THEN quantity END) AS '27-35',
SUM(CASE WHEN age between 36 and 44 THEN quantity END) AS '36-44',
SUM(CASE WHEN age between 45 and 53 THEN quantity END) AS '45-53',
SUM(CASE WHEN age between 54 and 64 THEN quantity END) AS '54-64'
```

product_category	18-26	27-35	36-44	45-53	54-64
Electronics	67	60	74	90	119



```
select * from electronics_men_pivot;
```

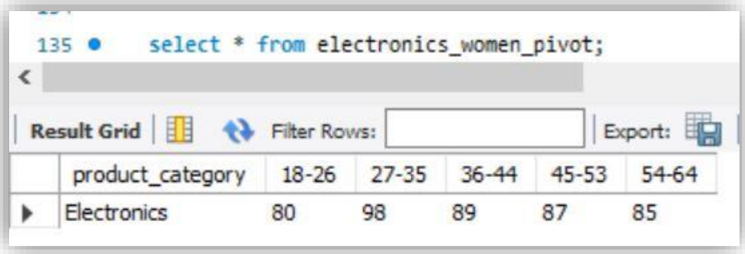
product_category	18-26	27-35	36-44	45-53	54-64
Electronics	67	60	74	90	119



```
-- create table electronics_men_unpivot as
select product_category, '18-26' as age_group, '67' as total_quantity_sold from electronics_men_pivot
union all
select product_category, '27-35' as age_group, '60' as total_quantity_sold from electronics_men_pivot
union all
select product_category, '36-44' as age_group, '74' as total_quantity_sold from electronics_men_pivot
union all
select product_category, '45-53' as age_group, '90' as total_quantity_sold from electronics_men_pivot
union all
select product_category, '54-64' as age_group, '119' as total_quantity_sold from electronics_men_pivot;
```

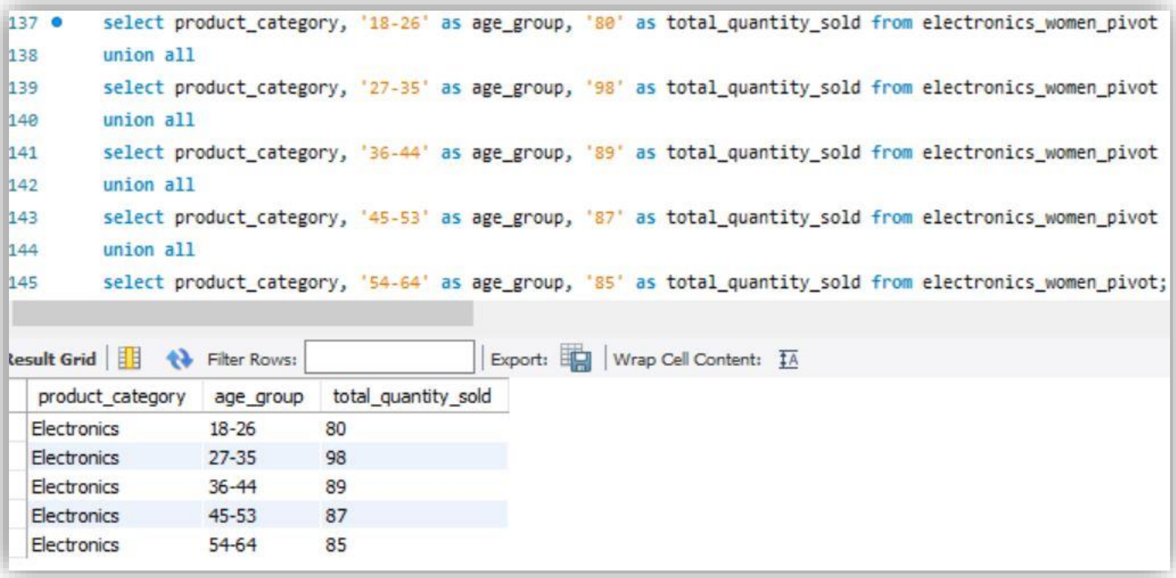
product_category	age_group	total_quantity_sold
Electronics	18-26	67
Electronics	27-35	60
Electronics	36-44	74
Electronics	45-53	90
Electronics	54-64	119

Figure 7. *electronics_men_unpivot* table



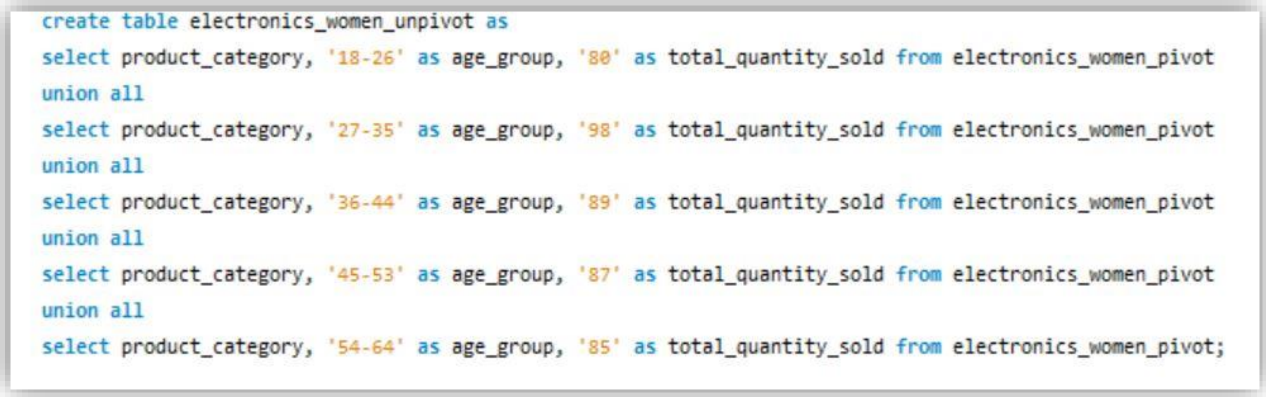
```
select * from electronics_women_pivot;
```

product_category	18-26	27-35	36-44	45-53	54-64
Electronics	80	98	89	87	85

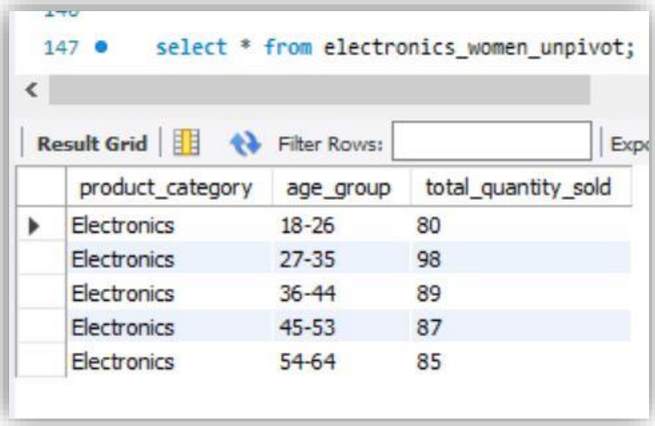


```
select product_category, '18-26' as age_group, '80' as total_quantity_sold from electronics_women_pivot
union all
select product_category, '27-35' as age_group, '98' as total_quantity_sold from electronics_women_pivot
union all
select product_category, '36-44' as age_group, '89' as total_quantity_sold from electronics_women_pivot
union all
select product_category, '45-53' as age_group, '87' as total_quantity_sold from electronics_women_pivot
union all
select product_category, '54-64' as age_group, '85' as total_quantity_sold from electronics_women_pivot;
```

product_category	age_group	total_quantity_sold
Electronics	18-26	80
Electronics	27-35	98
Electronics	36-44	89
Electronics	45-53	87
Electronics	54-64	85



```
create table electronics_women_unpivot as
select product_category, '18-26' as age_group, '80' as total_quantity_sold from electronics_women_pivot
union all
select product_category, '27-35' as age_group, '98' as total_quantity_sold from electronics_women_pivot
union all
select product_category, '36-44' as age_group, '89' as total_quantity_sold from electronics_women_pivot
union all
select product_category, '45-53' as age_group, '87' as total_quantity_sold from electronics_women_pivot
union all
select product_category, '54-64' as age_group, '85' as total_quantity_sold from electronics_women_pivot;
```



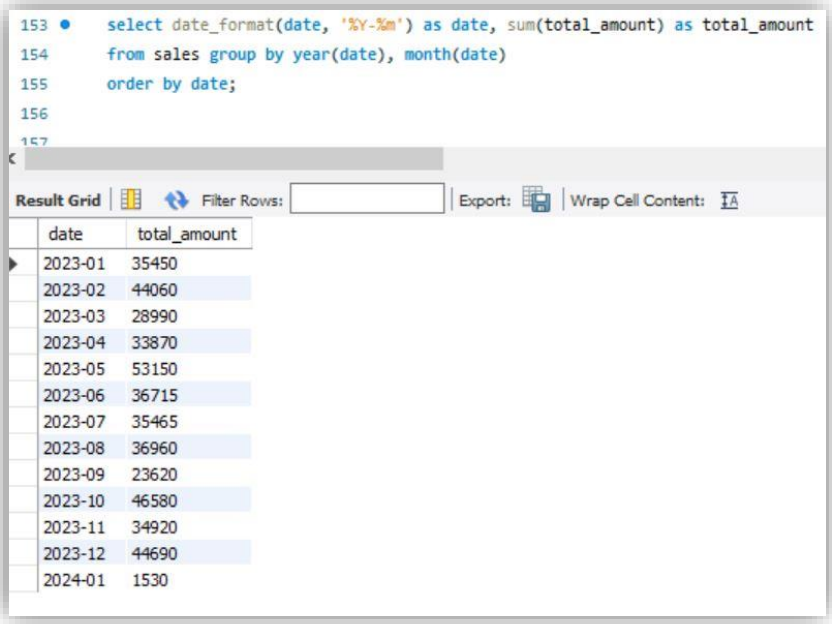
```
select * from electronics_women_unpivot;
```

product_category	age_group	total_quantity_sold
Electronics	18-26	80
Electronics	27-35	98
Electronics	36-44	89
Electronics	45-53	87
Electronics	54-64	85

Figure 8. *electronics_women_unpivot* table

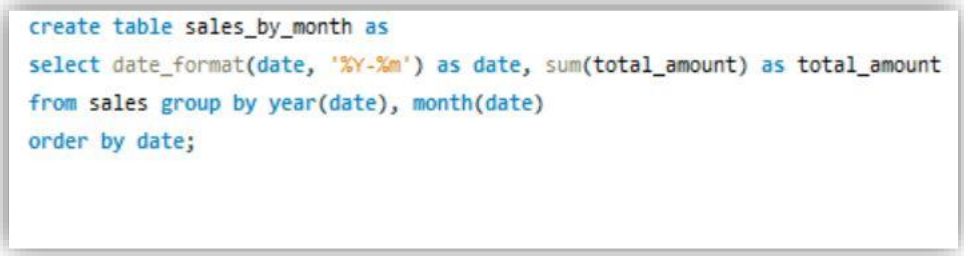
2) Are there discernible patterns in sales across different time periods?

Using **date_format()** and **sum** functions together by grouping data by year and month enables to determine monthly sales (Figure 9).



```sql
select date_format(date, '%Y-%m') as date, sum(total_amount) as total_amount
from sales group by year(date), month(date)
order by date;
```

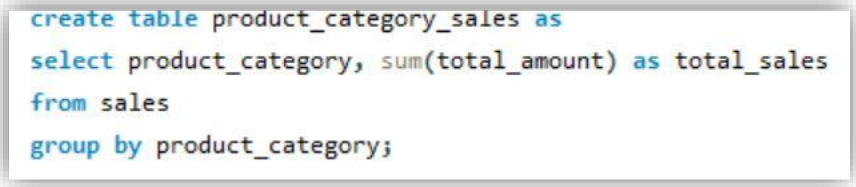
date	total_amount
2023-01	35450
2023-02	44060
2023-03	28990
2023-04	33870
2023-05	53150
2023-06	36715
2023-07	35465
2023-08	36960
2023-09	23620
2023-10	46580
2023-11	34920
2023-12	44690
2024-01	1530



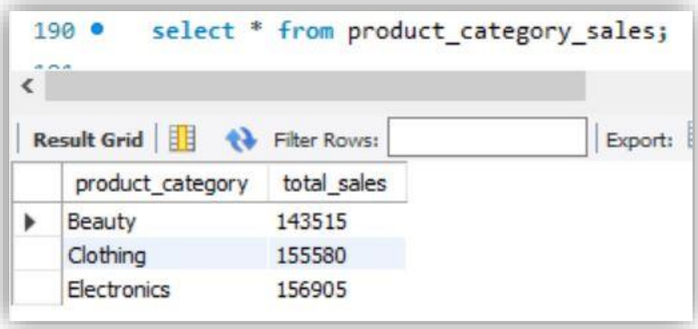
```sql
create table sales_by_month as
select date_format(date, '%Y-%m') as date, sum(total_amount) as total_amount
from sales group by year(date), month(date)
order by date;
```

Figure 9. *sales_by_month* table

3) Which product categories hold the highest appeal among customers? *Product_category_sales* table contains two columns such as ‘product_category’ and ‘total_sales’. As it is demonstrated (Figure 10), the highest sales were from electronics.



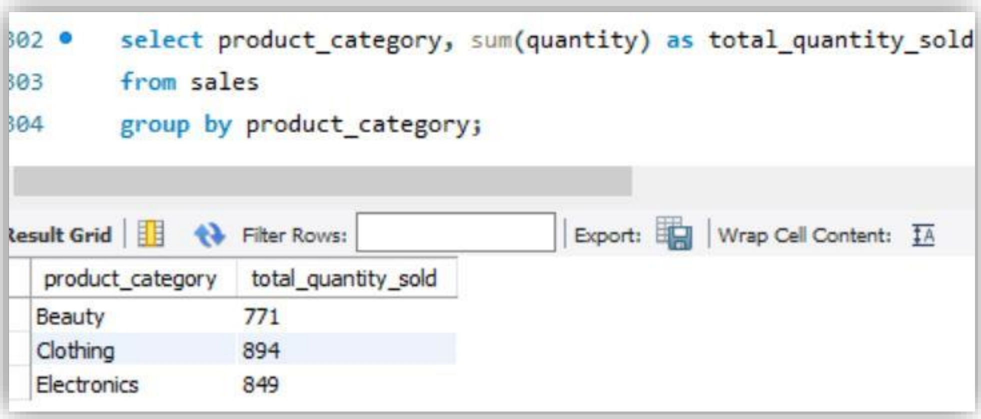
```sql
create table product_category_sales as
select product_category, sum(total_amount) as total_sales
from sales
group by product_category;
```



```sql
select * from product_category_sales;
```

product_category	total_sales
Beauty	143515
Clothing	155580
Electronics	156905

Figure 10. *product_category_sales* table

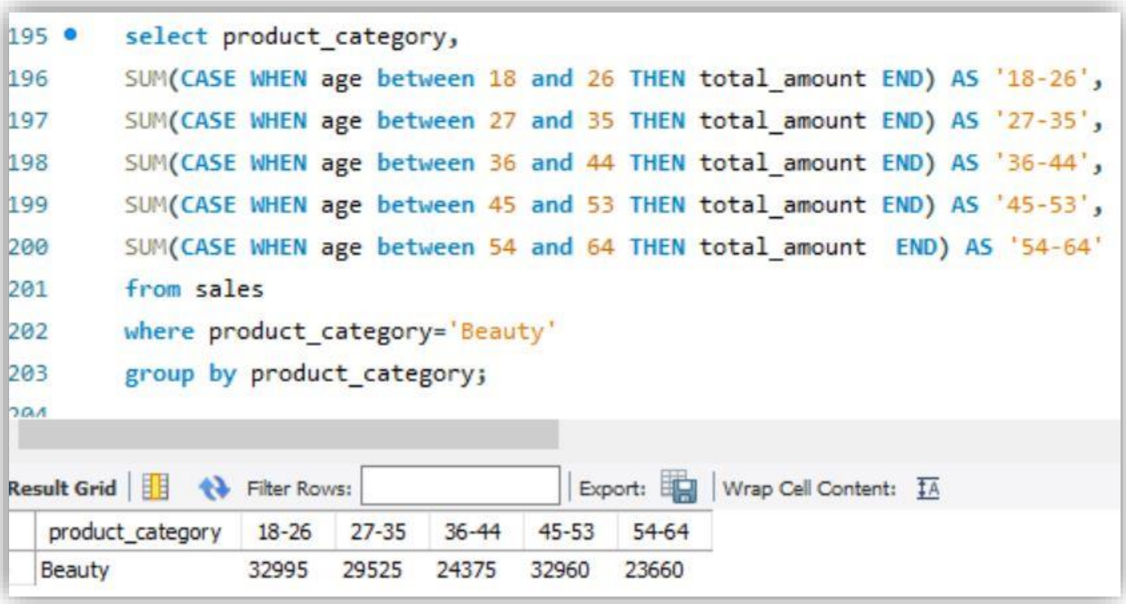


```sql
select product_category, sum(quantity) as total_quantity_sold
from sales
group by product_category;
```

product_category	total_quantity_sold
Beauty	771
Clothing	894
Electronics	849

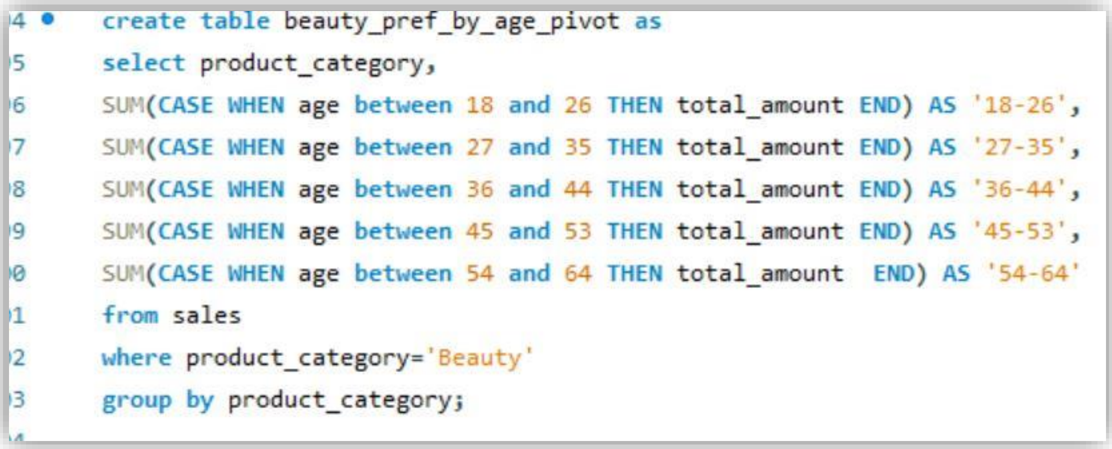
4) What are the relationships between age, spending, and product preferences?

In this part, the relationship between product preferences, age and total spending of customers was analyzed. First, *beauty_pref_by_age_pivot* table was created by applying **SUM** function, **CASE** statements (Figure 11). Grouping by product category enables to find total spending of that specific category for each age group.



```sql
select product_category,
SUM(CASE WHEN age between 18 and 26 THEN total_amount END) AS '18-26',
SUM(CASE WHEN age between 27 and 35 THEN total_amount END) AS '27-35',
SUM(CASE WHEN age between 36 and 44 THEN total_amount END) AS '36-44',
SUM(CASE WHEN age between 45 and 53 THEN total_amount END) AS '45-53',
SUM(CASE WHEN age between 54 and 64 THEN total_amount  END) AS '54-64'
from sales
where product_category='Beauty'
group by product_category;
```

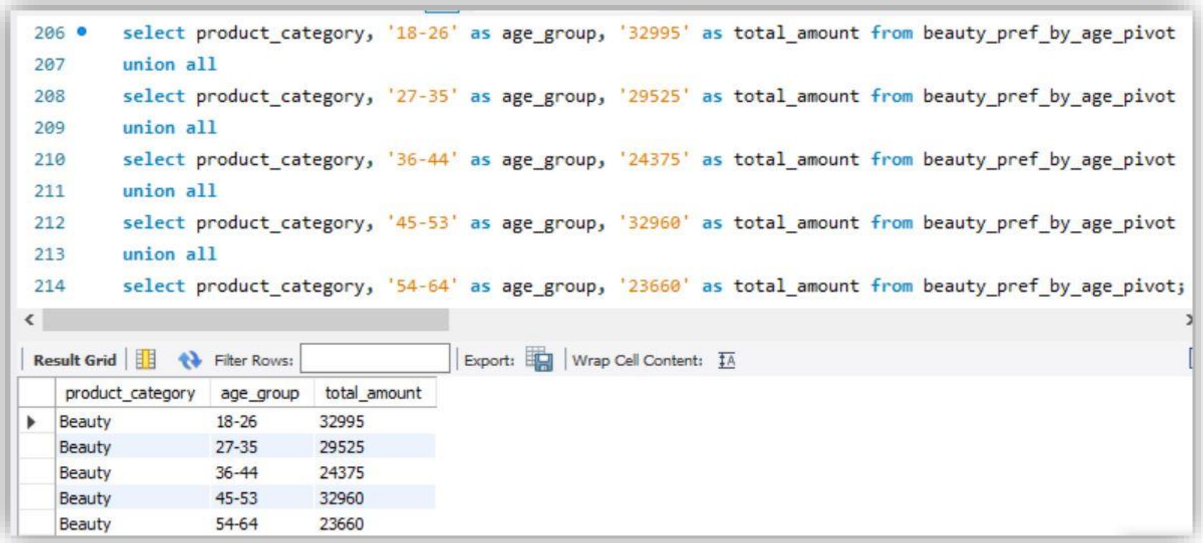
product_category	18-26	27-35	36-44	45-53	54-64
Beauty	32995	29525	24375	32960	23660



```sql
create table beauty_pref_by_age_pivot as
select product_category,
SUM(CASE WHEN age between 18 and 26 THEN total_amount END) AS '18-26',
SUM(CASE WHEN age between 27 and 35 THEN total_amount END) AS '27-35',
SUM(CASE WHEN age between 36 and 44 THEN total_amount END) AS '36-44',
SUM(CASE WHEN age between 45 and 53 THEN total_amount END) AS '45-53',
SUM(CASE WHEN age between 54 and 64 THEN total_amount  END) AS '54-64'
from sales
where product_category='Beauty'
group by product_category;
```

Figure 11. *beauty_pref_by_age_pivot* table

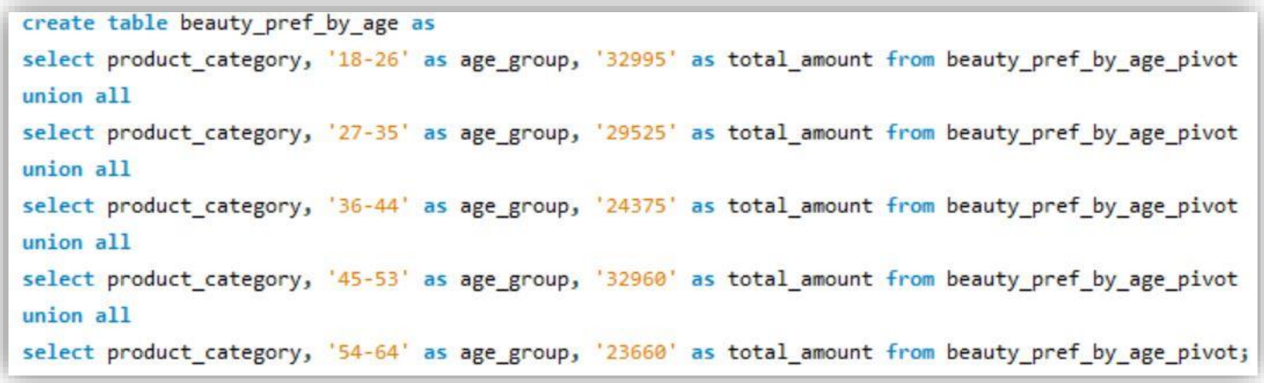
Then unpivoting was implemented:



```
select product_category, '18-26' as age_group, '32995' as total_amount from beauty_pref_by_age_pivot
union all
select product_category, '27-35' as age_group, '29525' as total_amount from beauty_pref_by_age_pivot
union all
select product_category, '36-44' as age_group, '24375' as total_amount from beauty_pref_by_age_pivot
union all
select product_category, '45-53' as age_group, '32960' as total_amount from beauty_pref_by_age_pivot
union all
select product_category, '54-64' as age_group, '23660' as total_amount from beauty_pref_by_age_pivot;
```

product_category	age_group	total_amount
Beauty	18-26	32995
Beauty	27-35	29525
Beauty	36-44	24375
Beauty	45-53	32960
Beauty	54-64	23660

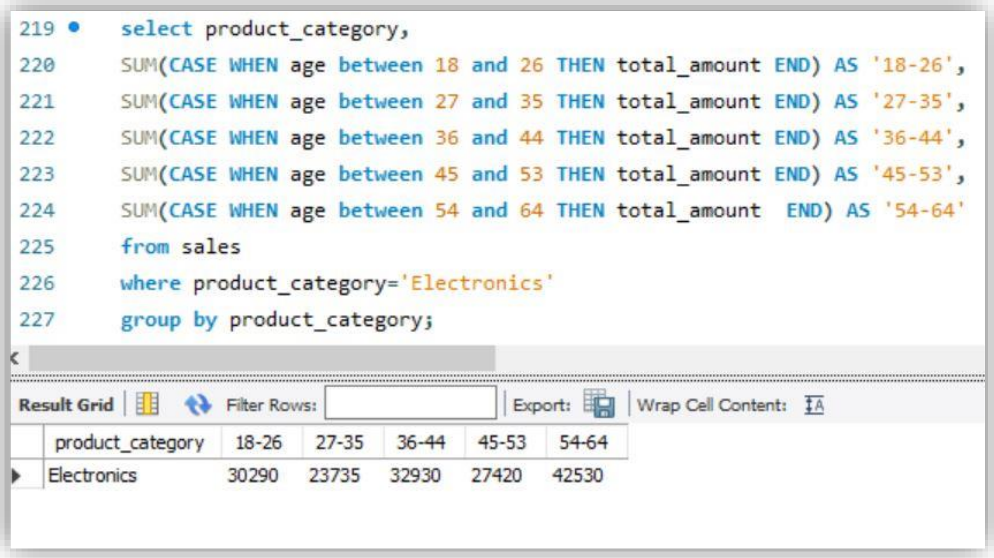
The next step is to create *beauty_pref_by_age* table, which shows spendings of each age group for beauty category (Figure 12):



```
create table beauty_pref_by_age as
select product_category, '18-26' as age_group, '32995' as total_amount from beauty_pref_by_age_pivot
union all
select product_category, '27-35' as age_group, '29525' as total_amount from beauty_pref_by_age_pivot
union all
select product_category, '36-44' as age_group, '24375' as total_amount from beauty_pref_by_age_pivot
union all
select product_category, '45-53' as age_group, '32960' as total_amount from beauty_pref_by_age_pivot
union all
select product_category, '54-64' as age_group, '23660' as total_amount from beauty_pref_by_age_pivot;
```

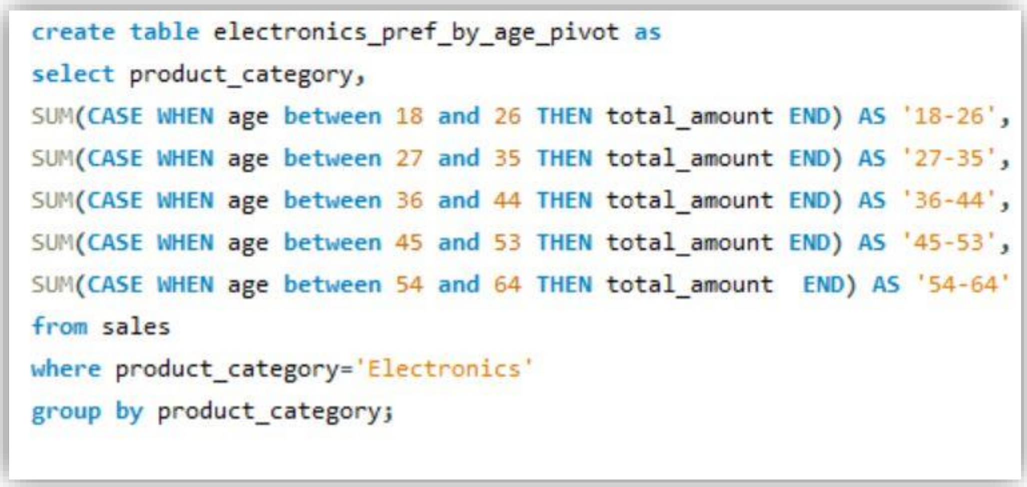
Figure 12. *beauty_pref_by_age* table

The same manipulations have been made for clothing and electronics categories.



```
select product_category,
SUM(CASE WHEN age between 18 and 26 THEN total_amount END) AS '18-26',
SUM(CASE WHEN age between 27 and 35 THEN total_amount END) AS '27-35',
SUM(CASE WHEN age between 36 and 44 THEN total_amount END) AS '36-44',
SUM(CASE WHEN age between 45 and 53 THEN total_amount END) AS '45-53',
SUM(CASE WHEN age between 54 and 64 THEN total_amount  END) AS '54-64'
from sales
where product_category='Electronics'
group by product_category;
```

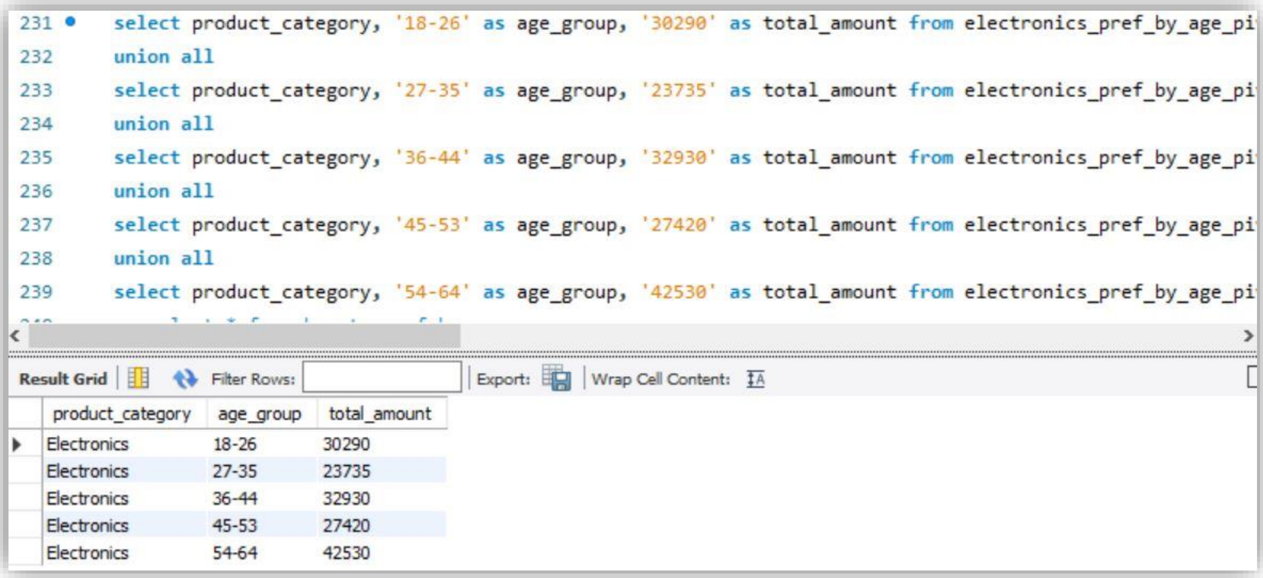
product_category	18-26	27-35	36-44	45-53	54-64
Electronics	30290	23735	32930	27420	42530



```
create table electronics_pref_by_age_pivot as
select product_category,
SUM(CASE WHEN age between 18 and 26 THEN total_amount END) AS '18-26',
SUM(CASE WHEN age between 27 and 35 THEN total_amount END) AS '27-35',
SUM(CASE WHEN age between 36 and 44 THEN total_amount END) AS '36-44',
SUM(CASE WHEN age between 45 and 53 THEN total_amount END) AS '45-53',
SUM(CASE WHEN age between 54 and 64 THEN total_amount  END) AS '54-64'
from sales
where product_category='Electronics'
group by product_category;
```

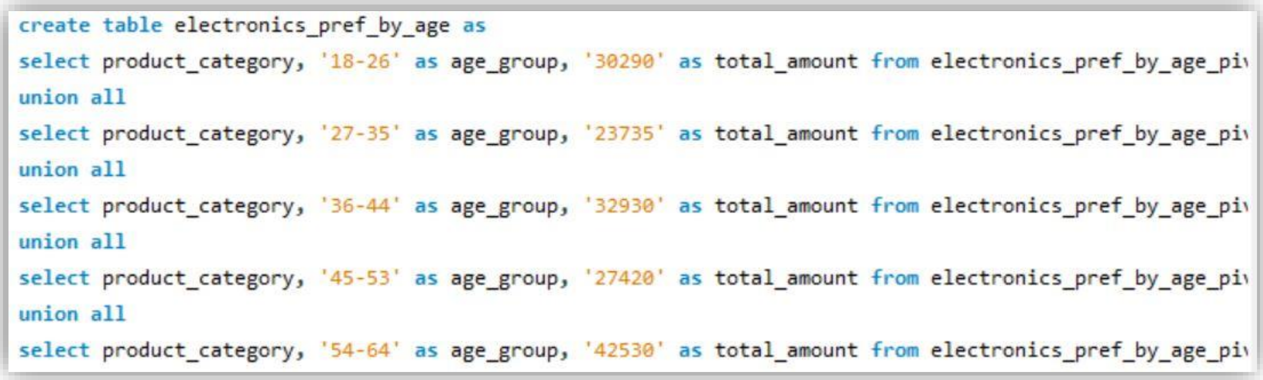
Figure 13. *electronics_pref_by_age_pivot* table

Unpivoting:



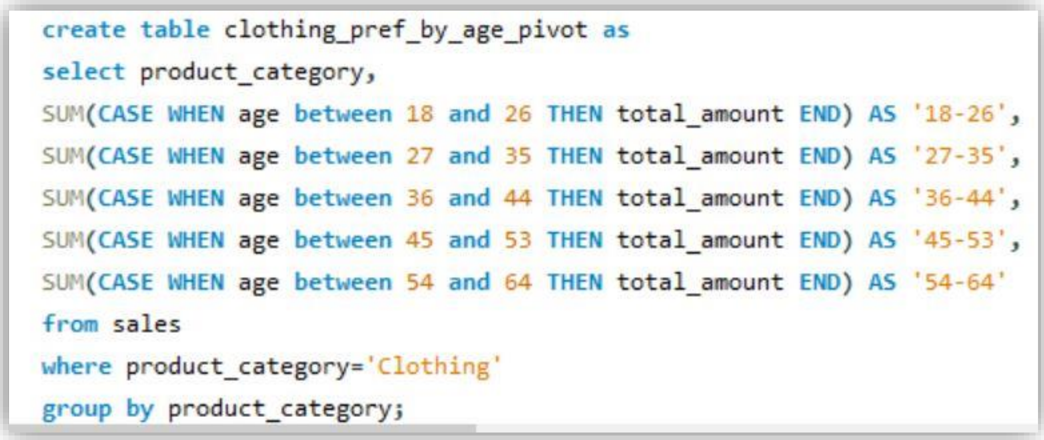
```
select product_category, '18-26' as age_group, '30290' as total_amount from electronics_pref_by_age_pi
union all
select product_category, '27-35' as age_group, '23735' as total_amount from electronics_pref_by_age_pi
union all
select product_category, '36-44' as age_group, '32930' as total_amount from electronics_pref_by_age_pi
union all
select product_category, '45-53' as age_group, '27420' as total_amount from electronics_pref_by_age_pi
union all
select product_category, '54-64' as age_group, '42530' as total_amount from electronics_pref_by_age_pi
```

product_category	age_group	total_amount
Electronics	18-26	30290
Electronics	27-35	23735
Electronics	36-44	32930
Electronics	45-53	27420
Electronics	54-64	42530



```
create table electronics_pref_by_age as
select product_category, '18-26' as age_group, '30290' as total_amount from electronics_pref_by_age_pi
union all
select product_category, '27-35' as age_group, '23735' as total_amount from electronics_pref_by_age_pi
union all
select product_category, '36-44' as age_group, '32930' as total_amount from electronics_pref_by_age_pi
union all
select product_category, '45-53' as age_group, '27420' as total_amount from electronics_pref_by_age_pi
union all
select product_category, '54-64' as age_group, '42530' as total_amount from electronics_pref_by_age_pi
```

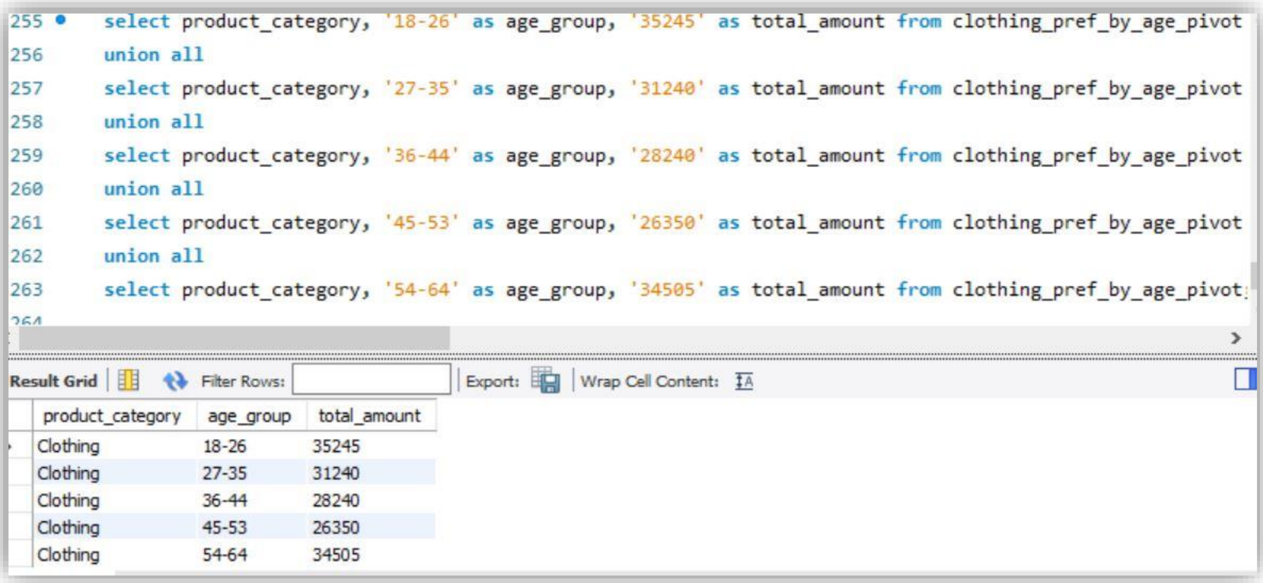
Figure 14. *electronics_pref_by_age* table



```sql
create table clothing_pref_by_age_pivot as
select product_category,
SUM(CASE WHEN age between 18 and 26 THEN total_amount END) AS '18-26',
SUM(CASE WHEN age between 27 and 35 THEN total_amount END) AS '27-35',
SUM(CASE WHEN age between 36 and 44 THEN total_amount END) AS '36-44',
SUM(CASE WHEN age between 45 and 53 THEN total_amount END) AS '45-53',
SUM(CASE WHEN age between 54 and 64 THEN total_amount END) AS '54-64'
from sales
where product_category='Clothing'
group by product_category;
```

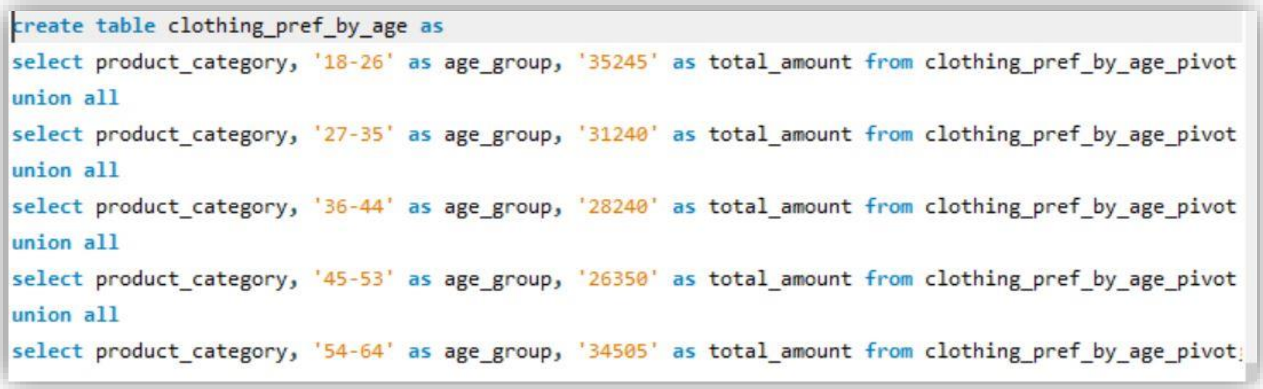
Figure 15. *clothing_pref_by_age_pivot* table

Unpivoting:



```sql
select product_category, '18-26' as age_group, '35245' as total_amount from clothing_pref_by_age_pivot
union all
select product_category, '27-35' as age_group, '31240' as total_amount from clothing_pref_by_age_pivot
union all
select product_category, '36-44' as age_group, '28240' as total_amount from clothing_pref_by_age_pivot
union all
select product_category, '45-53' as age_group, '26350' as total_amount from clothing_pref_by_age_pivot
union all
select product_category, '54-64' as age_group, '34505' as total_amount from clothing_pref_by_age_pivot;
```

product_category	age_group	total_amount
Clothing	18-26	35245
Clothing	27-35	31240
Clothing	36-44	28240
Clothing	45-53	26350
Clothing	54-64	34505

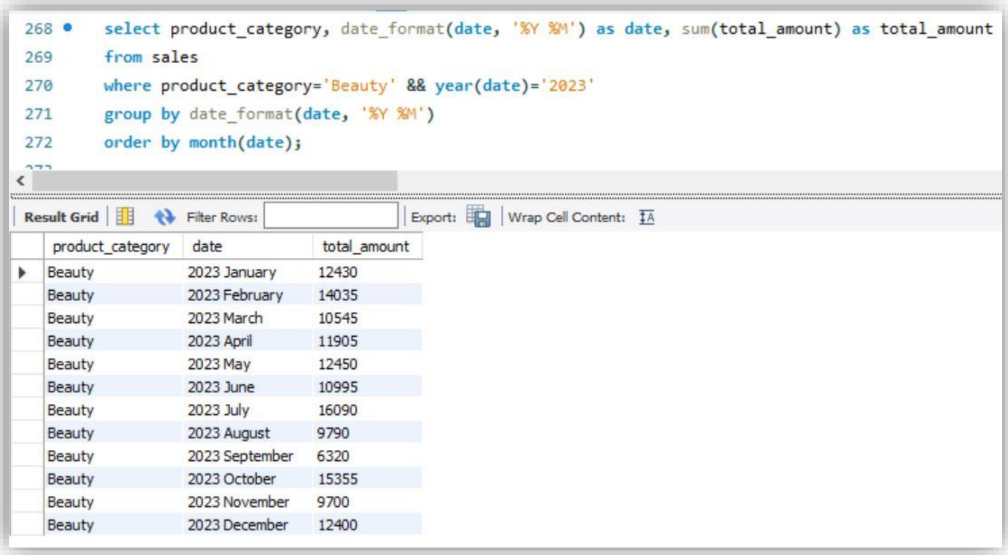


```sql
create table clothing_pref_by_age as
select product_category, '18-26' as age_group, '35245' as total_amount from clothing_pref_by_age_pivot
union all
select product_category, '27-35' as age_group, '31240' as total_amount from clothing_pref_by_age_pivot
union all
select product_category, '36-44' as age_group, '28240' as total_amount from clothing_pref_by_age_pivot
union all
select product_category, '45-53' as age_group, '26350' as total_amount from clothing_pref_by_age_pivot
union all
select product_category, '54-64' as age_group, '34505' as total_amount from clothing_pref_by_age_pivot;
```

Figure 16. *clothing_pref_by_age* table

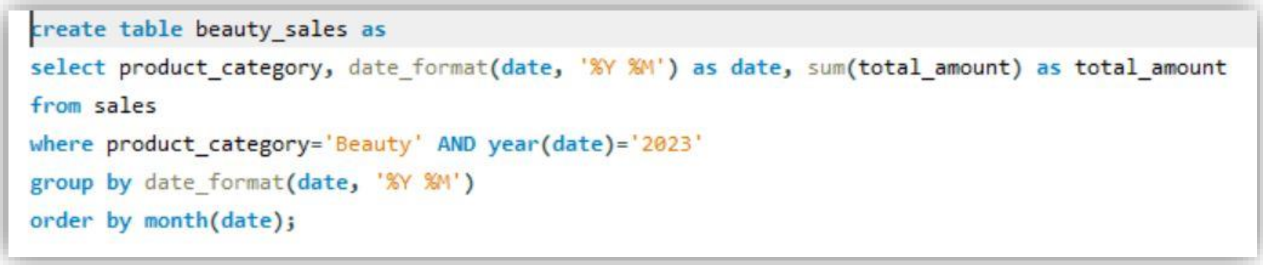
5) How do customers adapt their shopping habits during seasonal trends?

This section observes monthly sales of retail store for each month and category.



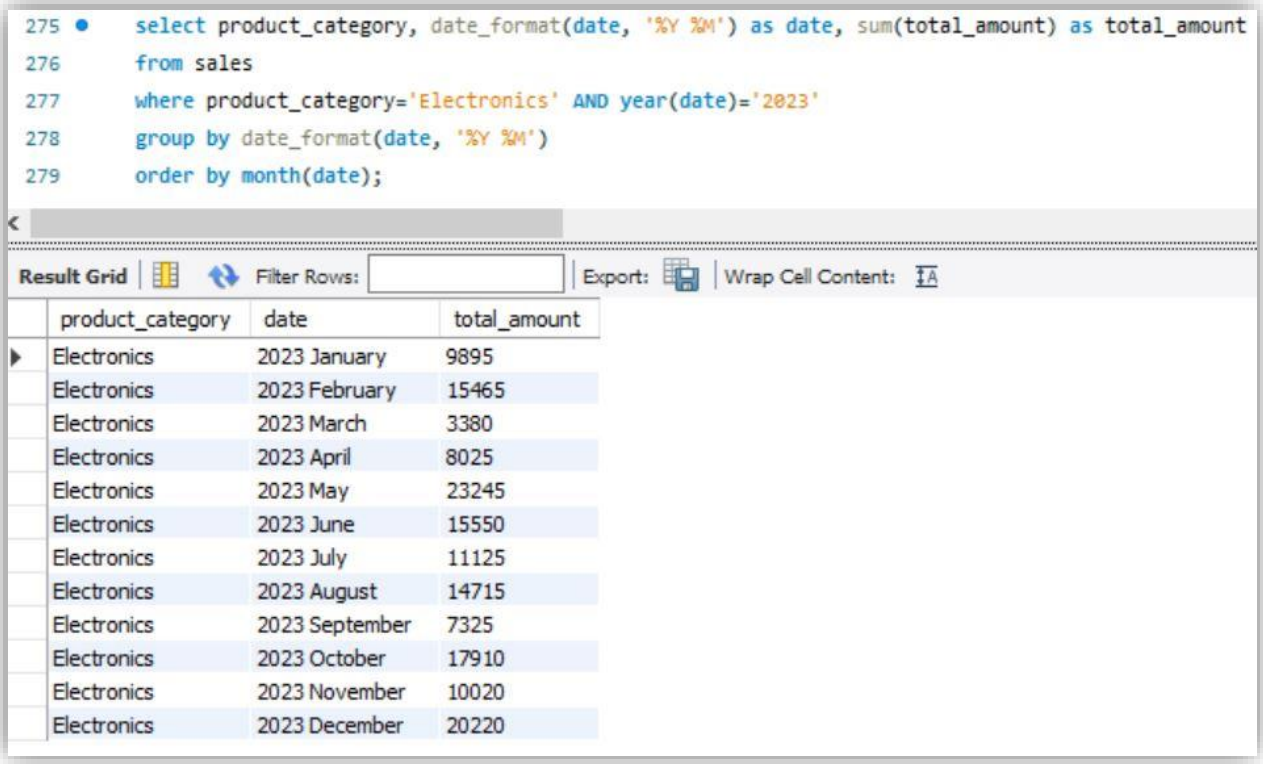
```sql
select product_category, date_format(date, '%Y %M') as date, sum(total_amount) as total_amount
from sales
where product_category='Beauty' && year(date)='2023'
group by date_format(date, '%Y %M')
order by month(date);
```

product_category	date	total_amount
Beauty	2023 January	12430
Beauty	2023 February	14035
Beauty	2023 March	10545
Beauty	2023 April	11905
Beauty	2023 May	12450
Beauty	2023 June	10995
Beauty	2023 July	16090
Beauty	2023 August	9790
Beauty	2023 September	6320
Beauty	2023 October	15355
Beauty	2023 November	9700
Beauty	2023 December	12400



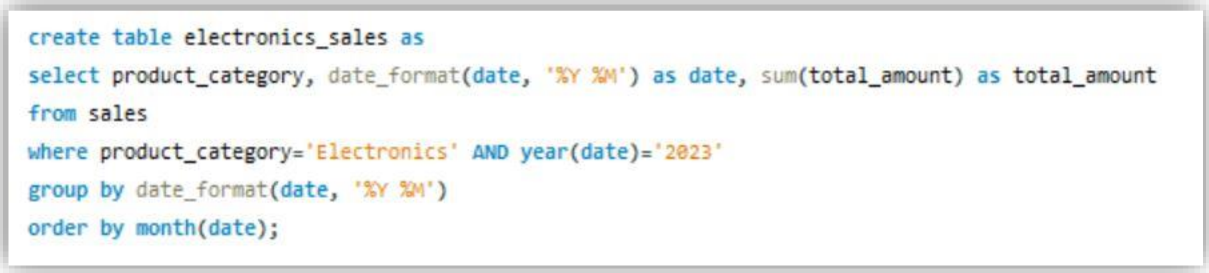
```sql
create table beauty_sales as
select product_category, date_format(date, '%Y %M') as date, sum(total_amount) as total_amount
from sales
where product_category='Beauty' AND year(date)='2023'
group by date_format(date, '%Y %M')
order by month(date);
```

Figure 17. *beauty_sales* table



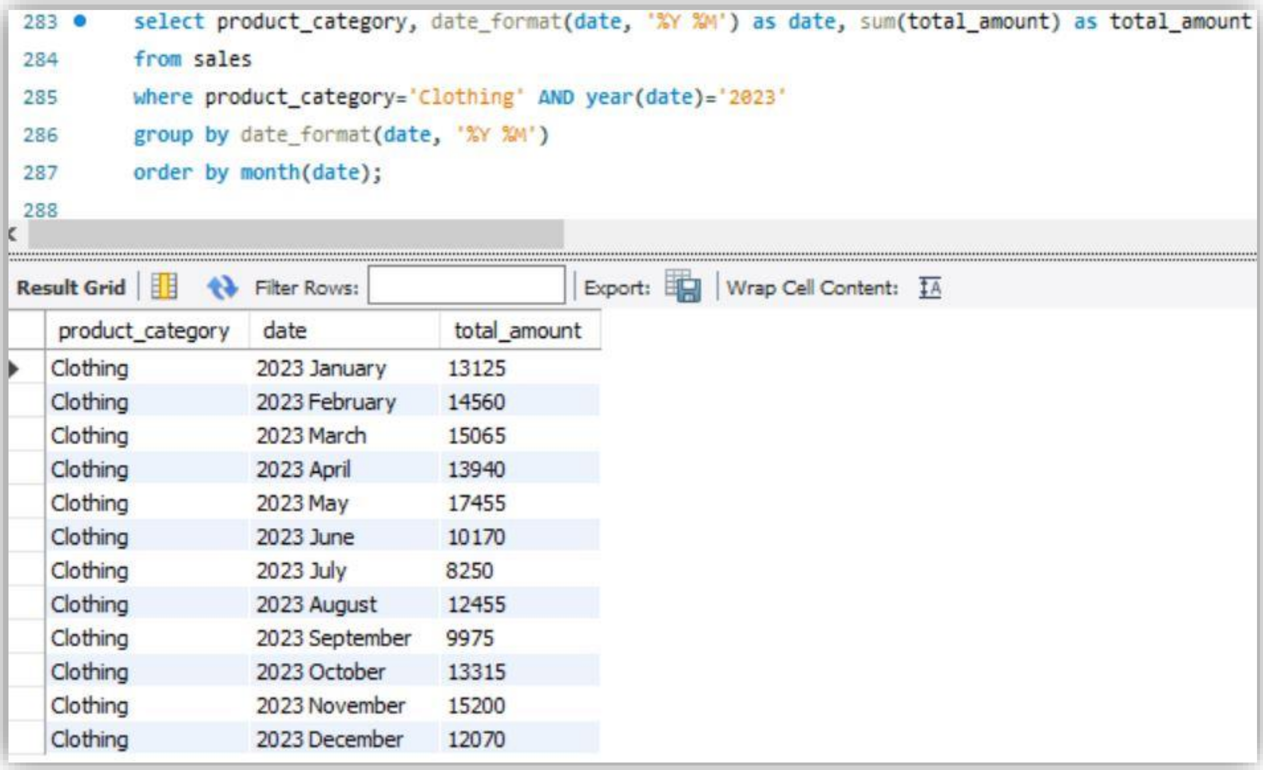
```sql
select product_category, date_format(date, '%Y %M') as date, sum(total_amount) as total_amount
from sales
where product_category='Electronics' AND year(date)='2023'
group by date_format(date, '%Y %M')
order by month(date);
```

product_category	date	total_amount
Electronics	2023 January	9895
Electronics	2023 February	15465
Electronics	2023 March	3380
Electronics	2023 April	8025
Electronics	2023 May	23245
Electronics	2023 June	15550
Electronics	2023 July	11125
Electronics	2023 August	14715
Electronics	2023 September	7325
Electronics	2023 October	17910
Electronics	2023 November	10020
Electronics	2023 December	20220



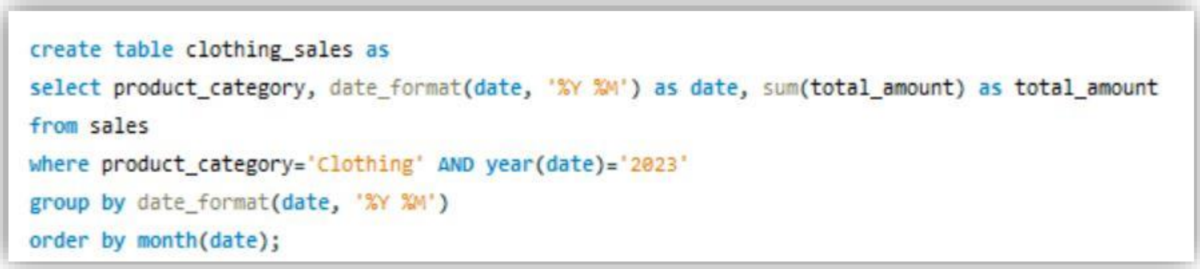
```
create table electronics_sales as
select product_category, date_format(date, '%Y %M') as date, sum(total_amount) as total_amount
from sales
where product_category='Electronics' AND year(date)='2023'
group by date_format(date, '%Y %M')
order by month(date);
```

Figure 18. *electronics_sales* table



```
select product_category, date_format(date, '%Y %M') as date, sum(total_amount) as total_amount
from sales
where product_category='Clothing' AND year(date)='2023'
group by date_format(date, '%Y %M')
order by month(date);
```

product_category	date	total_amount
Clothing	2023 January	13125
Clothing	2023 February	14560
Clothing	2023 March	15065
Clothing	2023 April	13940
Clothing	2023 May	17455
Clothing	2023 June	10170
Clothing	2023 July	8250
Clothing	2023 August	12455
Clothing	2023 September	9975
Clothing	2023 October	13315
Clothing	2023 November	15200
Clothing	2023 December	12070

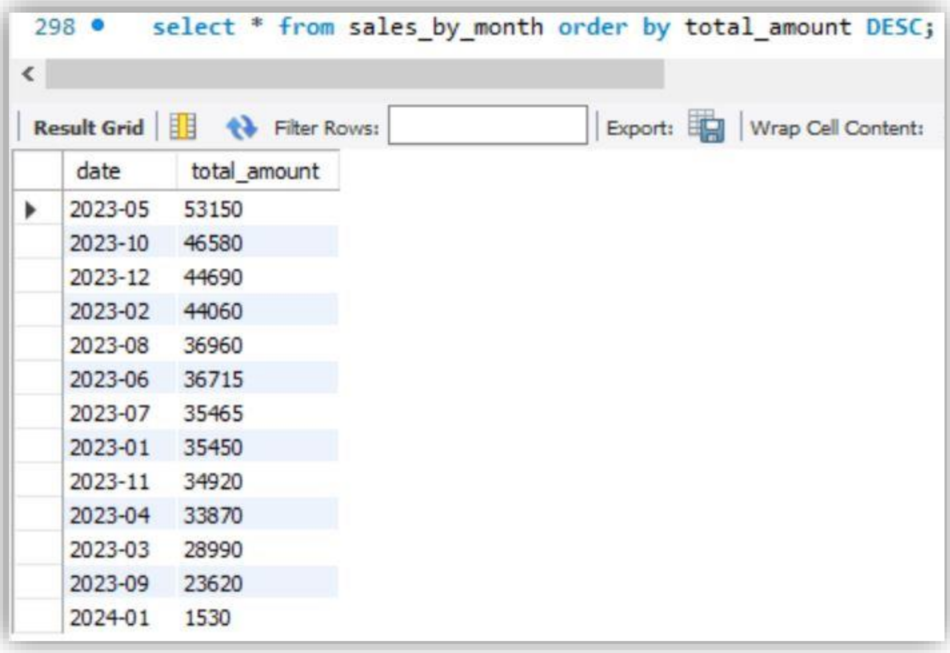


```
create table clothing_sales as
select product_category, date_format(date, '%Y %M') as date, sum(total_amount) as total_amount
from sales
where product_category='Clothing' AND year(date)='2023'
group by date_format(date, '%Y %M')
order by month(date);
```

Figure 19. *clothing_sales* table

RESULTS

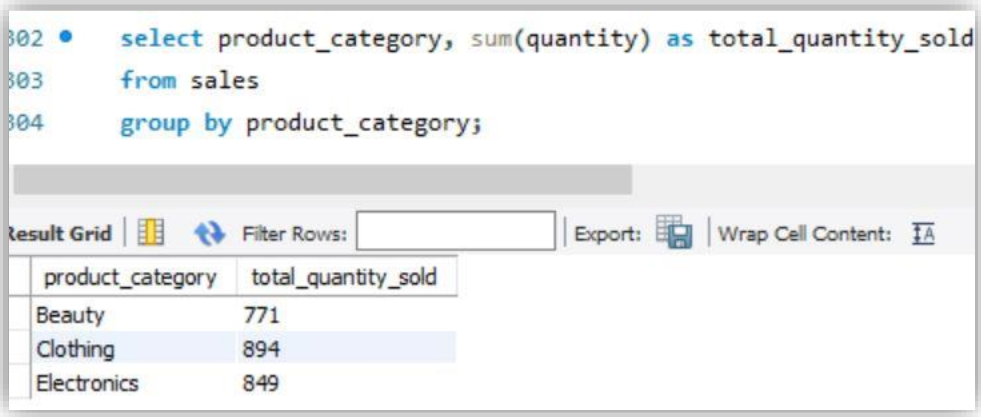
1) Figure 3 displays, men in age groups of 27-35 and 45-53 tended to buy more beauty products making up 81 and 80, respectively. Figure 4 shows, a highest quantity of sold beauty products was made by women aged 54-64. Also it can be seen that beauty products are in high demand for women who are 18-35 years old.
   In Figure 5-6, clothes have high demand for women and men who are in 54-64.
   Figure 7 demonstrates, 119 and 90 electronic products were sold by men aged 54-64 and 45-53, respectively. In Figure 8, women aged 27-35 bought more electronic products than women in other age groups.
2) The solution of second question shows, the highest sales of retail store were in May 2023. Also other higher sales were in October, December and January 2023.



```
select * from sales_by_month order by total_amount DESC;
```

date	total_amount
2023-05	53150
2023-10	46580
2023-12	44690
2023-02	44060
2023-08	36960
2023-06	36715
2023-07	35465
2023-01	35450
2023-11	34920
2023-04	33870
2023-03	28990
2023-09	23620
2024-01	1530

3) In terms of sales, popular product categories were electronics and clothing making up $156905 and $155580, respectively. The same tendency is noticed in terms of quantity sold. However, quantity of sold clothes is bit bigger than that in electronic products:



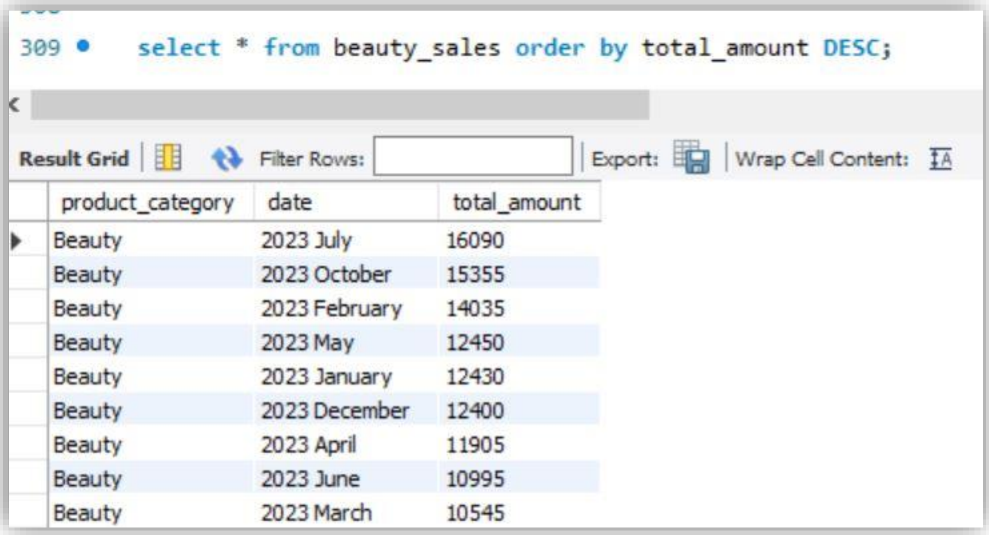
```
select product_category, sum(quantity) as total_quantity_sold
from sales
group by product_category;
```

product_category	total_quantity_sold
Beauty	771
Clothing	894
Electronics	849

4) Figure 12 illustrates, customers aged 18-26 and 45-53 spent significantly more money on beauty products making $32995 and $32960, correspondingly. In Figure 14, customers who are 54-64 years old have visibly high spending on electronics ($42530). Figure 16

demonstrates, clients in age groups of 18-26 and 54-64 bought clothes for $35245 and $34505, correspondingly.

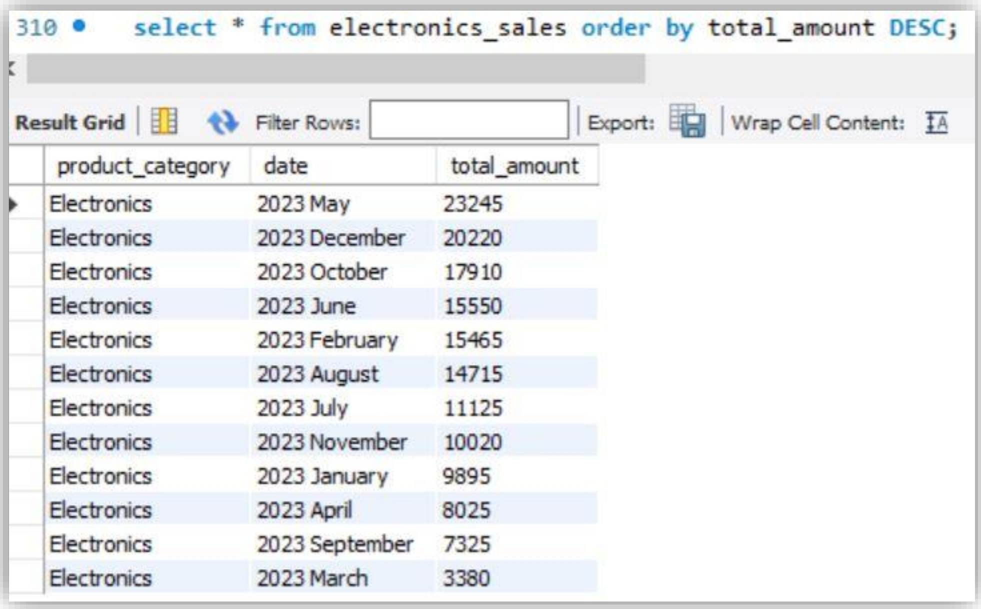
5) As it is shown in Figure 17, highest sales of beauty products were in July, October and February 2023:



```
select * from beauty_sales order by total_amount DESC;
```

product_category	date	total_amount
Beauty	2023 July	16090
Beauty	2023 October	15355
Beauty	2023 February	14035
Beauty	2023 May	12450
Beauty	2023 January	12430
Beauty	2023 December	12400
Beauty	2023 April	11905
Beauty	2023 June	10995
Beauty	2023 March	10545

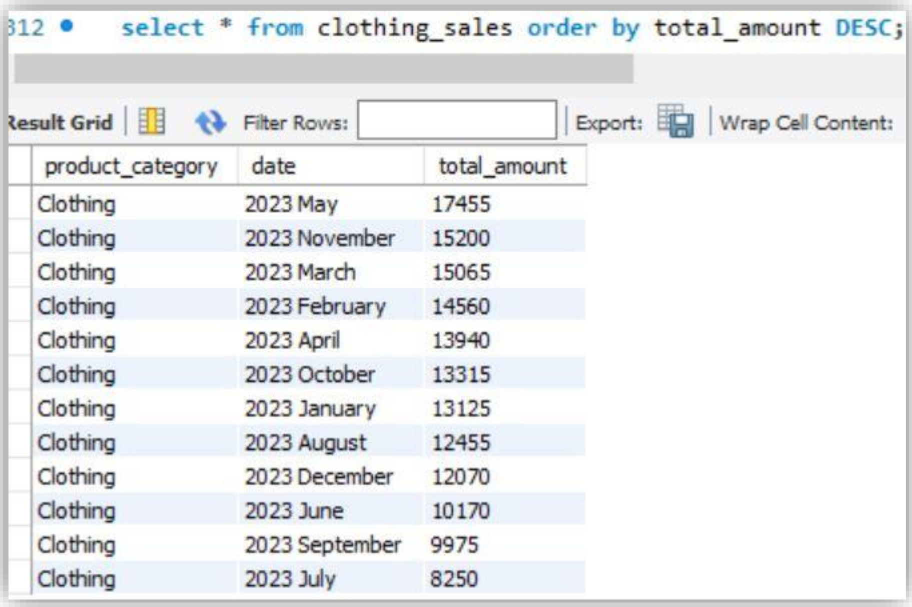
Figure 18 shows that highest sales from electronics noticeably were in May and December 2023.



```
select * from electronics_sales order by total_amount DESC;
```

product_category	date	total_amount
Electronics	2023 May	23245
Electronics	2023 December	20220
Electronics	2023 October	17910
Electronics	2023 June	15550
Electronics	2023 February	15465
Electronics	2023 August	14715
Electronics	2023 July	11125
Electronics	2023 November	10020
Electronics	2023 January	9895
Electronics	2023 April	8025
Electronics	2023 September	7325
Electronics	2023 March	3380

As it is seen in Figure 19, May, November and March 2023 had significantly high sales of clothing making $17455, $15200 and $15065, correspondingly:



```
select * from clothing_sales order by total_amount DESC;
```

product_category	date	total_amount
Clothing	2023 May	17455
Clothing	2023 November	15200
Clothing	2023 March	15065
Clothing	2023 February	14560
Clothing	2023 April	13940
Clothing	2023 October	13315
Clothing	2023 January	13125
Clothing	2023 August	12455
Clothing	2023 December	12070
Clothing	2023 June	10170
Clothing	2023 September	9975
Clothing	2023 July	8250