SQL PROJECT

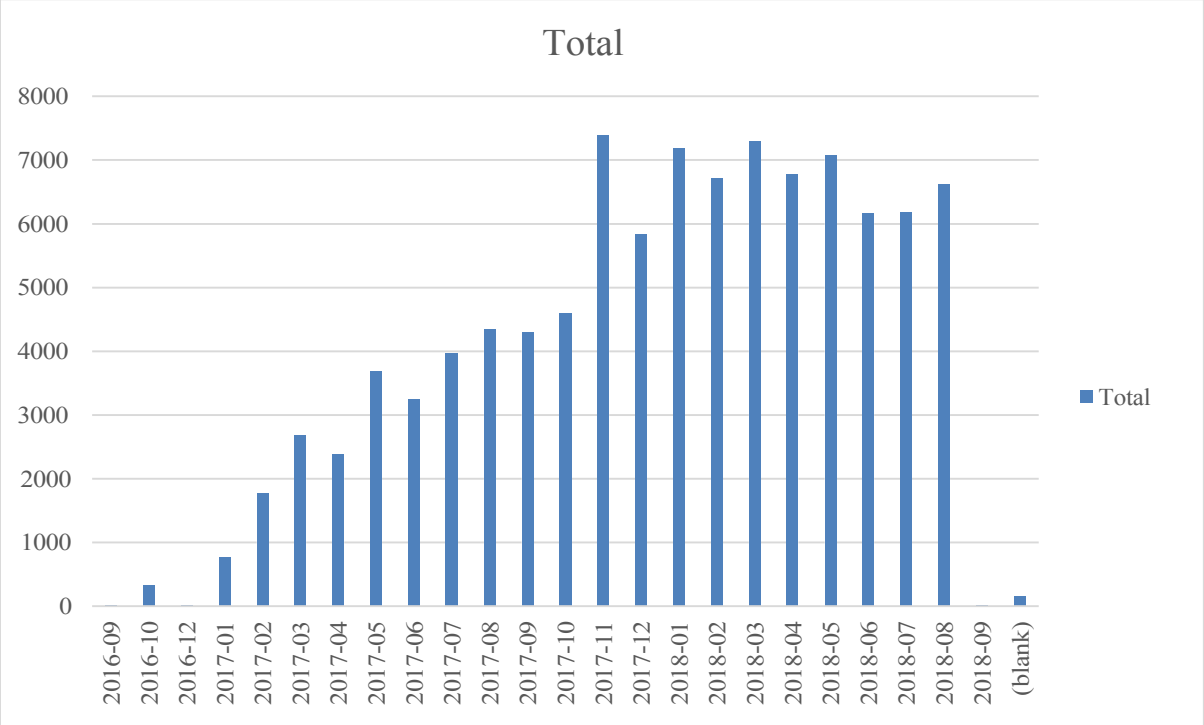
Case 1 : Order Analysis

Question 1 :

-Examine the order distribution on a monthly basis. For date data, order_approved_at should be used.

```
select
count (order_id),
to_char (order_approved_at,'YYYY-MM') as month
from orders
group by 2
order by 2
```

count	month
1	2016-09
320	2016-10
1	2016-12
760	2017-01
1765	2017-02
2689	2017-03
2374	2017-04
3693	2017-05
3252	2017-06
3974	2017-07



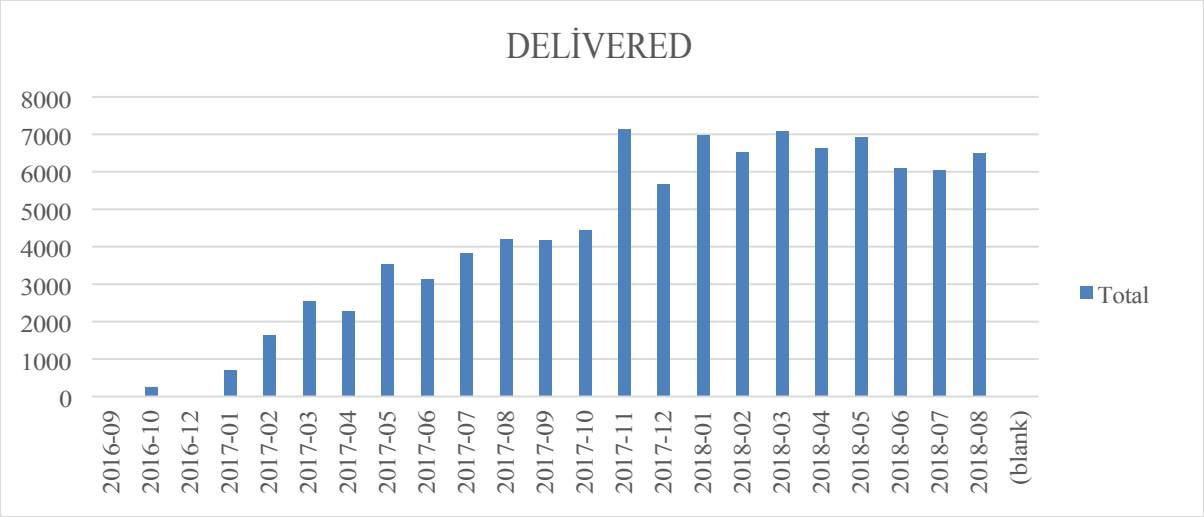
When we examine the order data table by month, we see that the data starts with very small values, then we can see that there is a general upward trend, but it can be seen that the number of orders peaked in November 2017. on this date, the company may have run a campaign, the effect of Black Friday, which is widely known around the world, may have been seen. in the following December, we can see a clear decline compared to November Christmas in December, so it can be expected that the number of orders will remain high, but it may have been a holiday period when people did not buy and give many gifts and spent time with each other. After that, we can see a general increase, albeit fluctuating, but there is a downward trend again in June and July. Although there is no clear indication of this situation, it can be said that the company is insufficient in terms of the campaigns it offers to its customers.

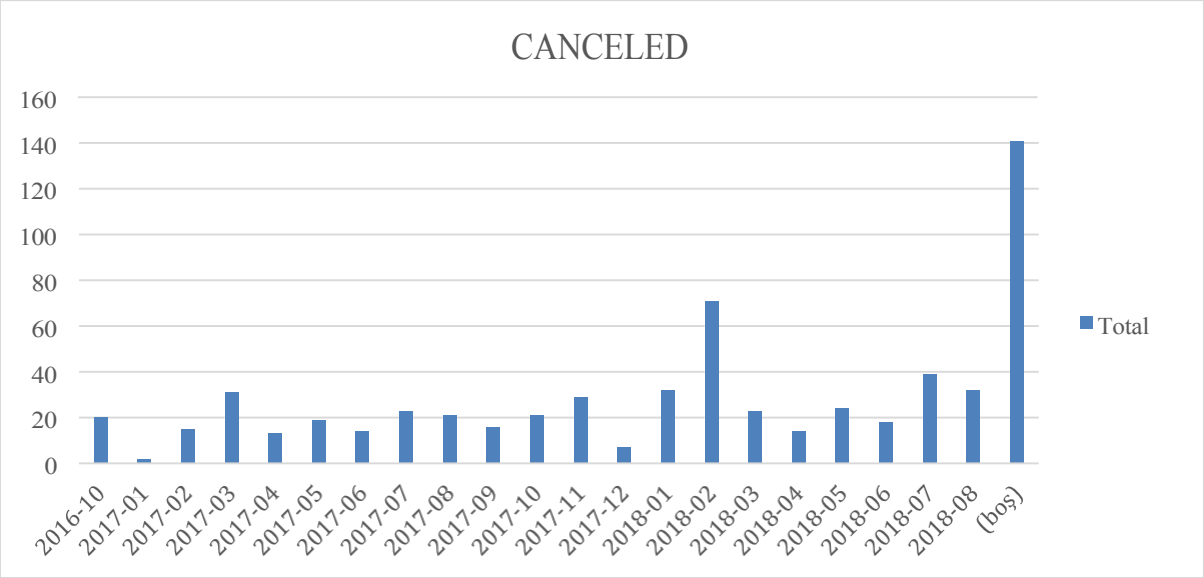
Question 2 :

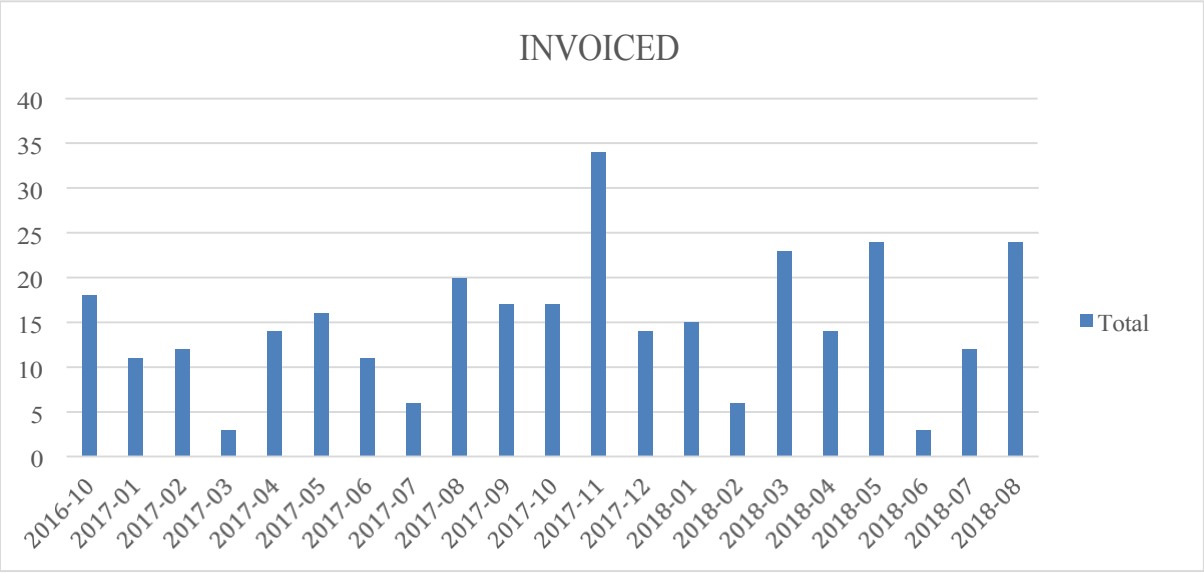
-Examine the number of orders in the order status breakdown on a monthly basis. Visualize the output of the query with excel. Are there any months with a dramatic decrease or increase? Analyze and interpret the data.

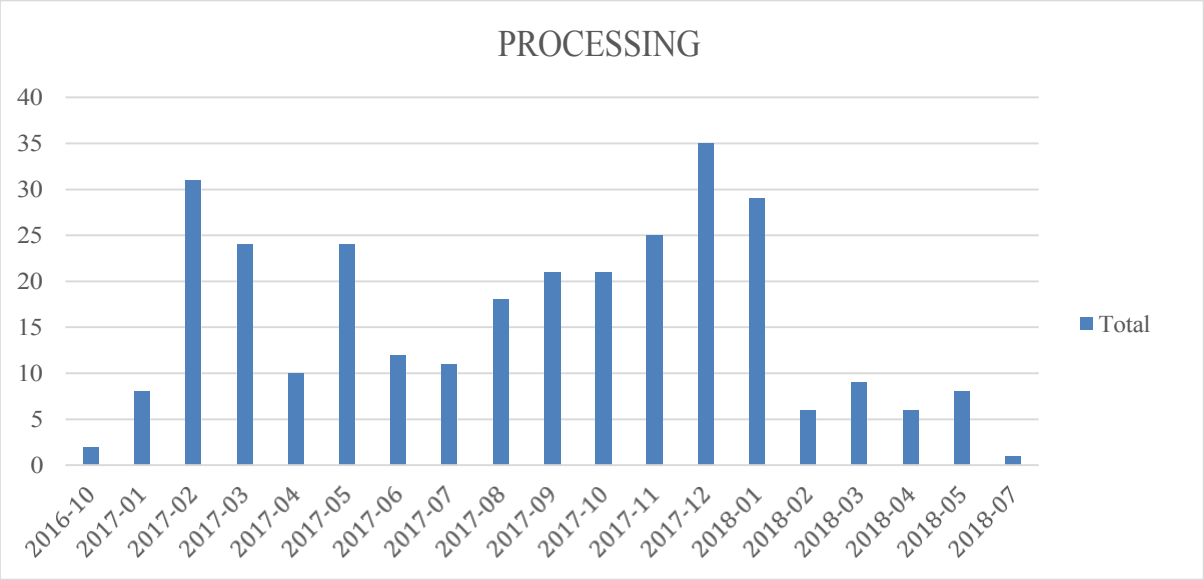
```
select
to_char(order_approved_at,'YYYY-MM') as month,
order_status,
count(order_id),
sum(Count(order_id)) OVER (PARTITION BY to_char(order_approved_at,'YYYY-MM')) AS
total
from orders
group by 1,2
order by 1
```

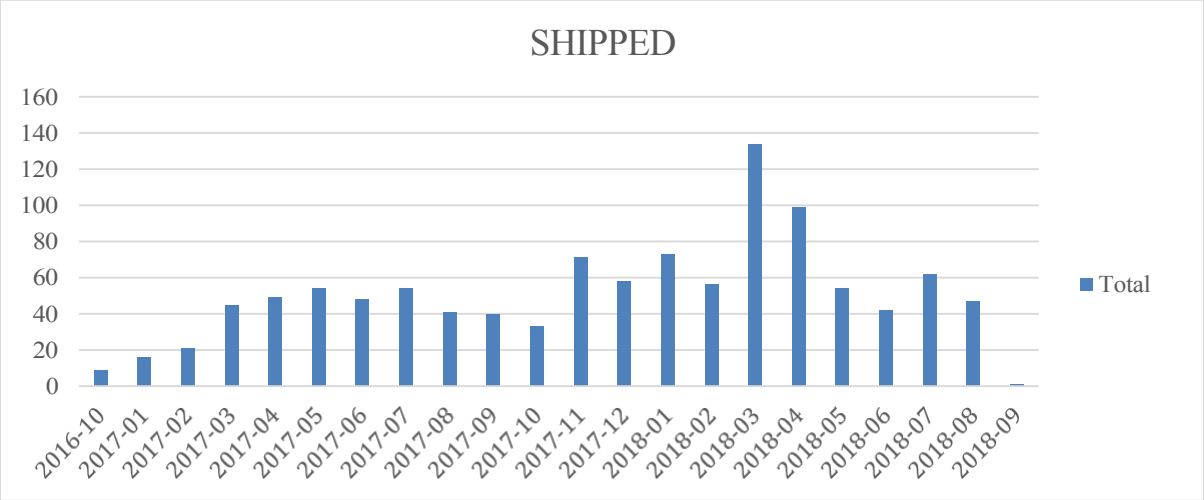
month	order_status	count	total
2016-09	delivered	1	1
2016-10	canceled	20	320
2016-10	delivered	265	320
2016-10	invoiced	18	320
2016-10	processing	2	320
2016-10	shipped	9	320
2016-10	unavailable	6	320
2016-12	delivered	1	1
2017-01	canceled	2	760
2017-01	delivered	715	760

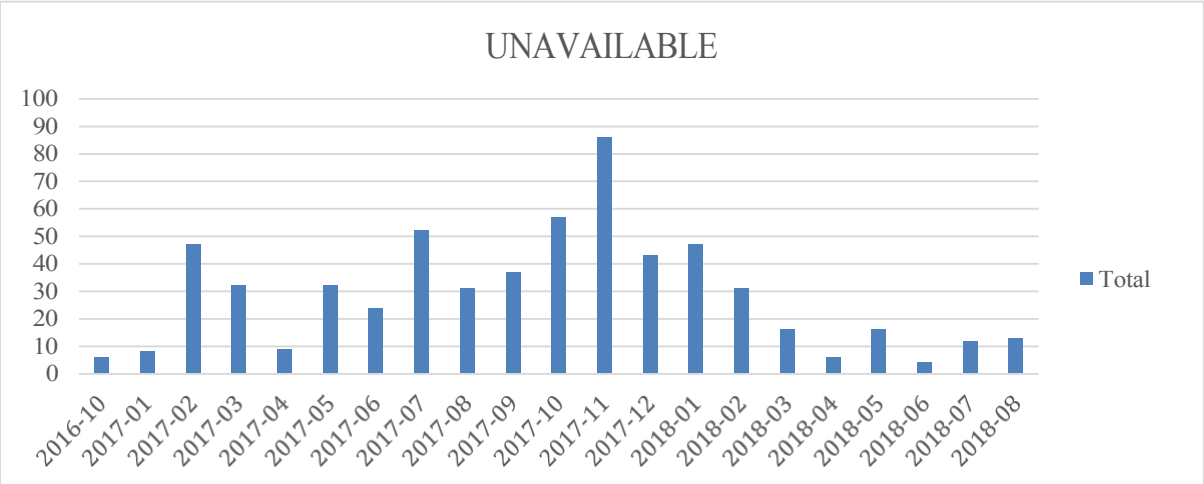

DELİVERED
8000
7000
6000
5000
4000
3000
2000
1000
0
2016-09
2016-10
2016-12
2017-01
2017-02
2017-03
2017-04
2017-05
2017-06
2017-07
2017-08
2017-09
2017-10
2017-11
2017-12
2018-01
2018-02
2018-03
2018-04
2018-05
2018-06
2018-07
2018-08
(blank)
Total


CANCELED
160
140
120
100
80
60
40
20
0
2016-10
2017-01
2017-02
2017-03
2017-04
2017-05
2017-06
2017-07
2017-08
2017-09
2017-10
2017-11
2017-12
2018-01
2018-02
2018-03
2018-04
2018-05
2018-06
2018-07
2018-08
(boş)
Total


INVOICED
40
35
30
25
20
15
10
5
0
2016-10
2017-01
2017-02
2017-03
2017-04
2017-05
2017-06
2017-07
2017-08
2017-09
2017-10
2017-11
2017-12
2018-01
2018-02
2018-03
2018-04
2018-05
2018-06
2018-07
2018-08
Total


PROCESSING
40
35
30
25
20
15
10
5
0
2016-10
2017-01
2017-02
2017-03
2017-04
2017-05
2017-06
2017-07
2017-08
2017-09
2017-10
2017-11
2017-12
2018-01
2018-02
2018-03
2018-04
2018-05
2018-07
Total


SHIPPED
160
140
120
100
80
60
40
20
0
2016-10
2017-01
2017-02
2017-03
2017-04
2017-05
2017-06
2017-07
2017-08
2017-09
2017-10
2017-11
2017-12
2018-01
2018-02
2018-03
2018-04
2018-05
2018-06
2018-07
2018-08
2018-09
Total


UNAVAILABLE
100
90
80
70
60
50
40
30
20
10
0
2016-10
2017-01
2017-02
2017-03
2017-04
2017-05
2017-06
2017-07
2017-08
2017-09
2017-10
2017-11
2017-12
2018-01
2018-02
2018-03
2018-04
2018-05
2018-06
2018-07
2018-08
Total

Order numbers were analyzed by examining the order status angle and creating a graph according to each situation.
The first graph shows the number of orders delivered, this graph is proportional to the Total orders graph and does not show a different situation.
When the canceled orders are examined, it seems that there are not many order cancellations, but there is an increase in February 2018 compared to other months, and there is a Rio carnival in Brazil in February. There may have been delivery problems or order cancelations due to this.
The low number of invoiced but not yet processed orders was highest in November 2017, when there were only the highest number of total orders, which may be due to the large number of orders received delaying processing.
When we look at the number of orders processed, it is slightly higher in the months with holidays such as Christmas and Carnival, as mentioned earlier, which may be due to special occasions such as these. For shipped orders, the highest amount is reached in March 2018, but there is no reason that could be related to this.
As for non-existing orders, the highest amount was observed in November 2017, which may be due to order intensity.

Question 3 :

-Examine the number of orders by product category. What are the categories that stand out on special occasions? For example, New Year, Valentine's Day...

```
select
count(distinct o.order_id) ,
category_name_english,
to_char(order_approved_at,'MM') as month
from orders o
left join order_items i on i.order_id=o.order_id
left join products p on i.product_id=p.product_id
left join translation t on t.category_name=p.product_category_name
group by 2,3
order by 1 desc
```

count	category_name_english	month
1135	health_beauty	8
1036	health_beauty	6
1025	bed_bath_table	7
1016	bed_bath_table	8
987	bed_bath_table	6
985	health_beauty	7
972	health_beauty	5
954	bed_bath_table	5
920	bed_bath_table	3
906	computers_accessories	2

Line Labels	1	2	3	4	5	6	7	8	9	10	11	12	(blank)	General Total
auto	291	410	435	434	446	412	454	550	109	161	255	278		550
baby	259	212	271	286	360	303	348	354	173	150	201	148		360
bed_bath_table	871	870	1088	1006	1140	1148	1199	1180	532	556	956	568		1199
computers_accessories	711	1098	946	668	793	674	737	778	261	333	527	301		1098
consoles_games			109					154		114	138			154
construction_tools_construction					125	120	123	170						170
cool_stuff	337	252	382	320	430	302	349	390	211	263	296	261		430
electronics	359	352	301	254	310	232	256	177			179	221		359
fashion_bags_accessories	161	147	204	155	223	176	161	230	122	121	198	133		230
food								126						126
furniture_decor	790	678	942	781	899	635	737	903	349	437	776	404		942
garden_tools	296	364	419	385	455	299	359	386	223	286	548	326		548
health_beauty	706	835	889	862	1079	1143	1095	1221	389	403	565	483		1221
home_appliances				116		108	117	116						117
housewares	374	486	607	652	931	918	820	953	254	246	413	310		953
luggage_accessories	132			102	124		127	129						132
office_furniture	149	162	293	192	161	122	197			107				293
perfumery	260	263	313	279	343	344	326	356	159	221	306	249		356
pet_shop	106	142	169	214	215	241	243	295						295
sports_leisure	718	800	979	790	833	703	838	918	461	486	604	509		979
stationery	414	175	222	193	197	203	263	259			162	253		414
telephony	385	470	517	427	487	382	362	418	180	265	377	273		517
toys	208	210	339	310	427	350	356	363	298	313	473	470		473
watches_gifts	380	369	514	586	788	609	682	642	269	319	462	371		788

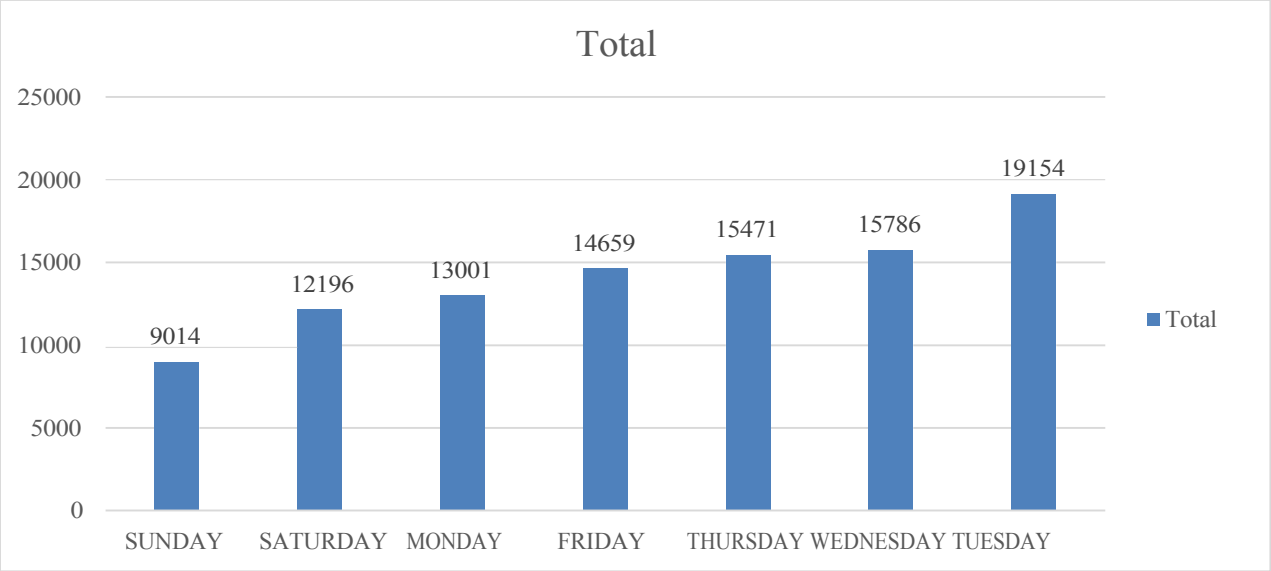
Question 3 was analyzed in terms of order quantities and months. Categories with more than 100 orders were included to reduce the crowding in the table. When analyzed by months, we see that the categories of electronics, luggage products and stationery products reached high numbers in January. This may have increased travel due to Christmas and increased stationery sales with the opening of schools. In February, we see that computers and accessories show the highest value. In March, sports equipment attracts attention, outdoor sports activities may have increased due to the summer season. Apart from this, it is seen that gardening products reached a high sales amount in November, which may be associated with the onset of spring and increased gardening activities.

Question 4 :

-Examine the number of orders on the basis of days of the week (Monday, Thursday, ...) and days of the month (such as the 1st, 2nd of the month). Create and interpret a visual in excel with the output of the query you wrote.

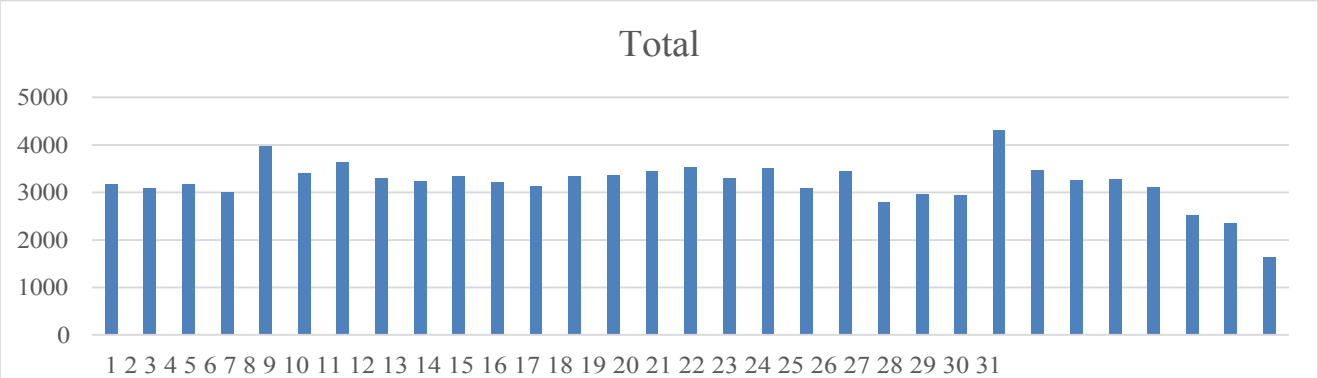
```
SELECT
count(order_id),
TO_CHAR(order_approved_at,'DAY') AS order_day
from orders
group by 2
```

count	order_day
14659	FRIDAY
13001	MONDAY
12196	SATURDAY
9014	SUNDAY
15471	THURSDAY
19154	TUESDAY
15786	WEDNESDAY




```
SELECT
count(order_id),
TO_CHAR(order_approved_at,'DD') AS order_day
from orders
group by 2
ORDER BY 2
ASC
```

count	order_day
3170	1
3086	2
3168	3
3000	4
3970	5
3409	6
3644	7
3306	8
3226	9
3333	10



When the order numbers are evaluated by days of the week, we see that the lowest order numbers come first on Sunday and then on Saturday. The source of this situation may be that people spend more time socializing or resting at the weekend and do not show much interest in shopping. Then comes Monday and Friday, the first and last days of the week, where we can say that people are partly busier and spend less time shopping. Mid-week days can be seen as the days when people spend the most time shopping online.

Looking at the days of the month graph, 3 days stand out. It is seen that the number of orders increased on the 5th and the 24th, which may be salary or advance days from today. On the 31st of the month, it seems largely low. It may be due to the decrease in people's budgets at the end of the month.

Case 2 : Customer Analysis

Question 1 :

-In which cities do customers shop more? Determine the city of the customer as the city where they place the most orders and analyze accordingly.

For example, Sibel orders from 3 different cities, 3 orders from Çanakkale, 8 orders from Muğla and 10 orders from Istanbul. Sibel's city is the city with the most orders You should select Istanbul and Sibel's orders should appear as 21 orders from Istanbul.

```
WITH customer_orders AS (
   SELECT
     c.customer_unique_id,
      c.customer_id,
      c.customer_city,
      COUNT(o.order_id) AS order_count
   FROM customers c
   LEFT JOIN orders o ON c.customer_id = o.customer_id
   GROUP BY c.customer_id, c.customer_city
),
```

```
ranked_orders AS (
   SELECT
      customer_unique_id,
       customer_id,
       customer_city,
                  order_count,
       ROW_NUMBER() OVER (PARTITION BY customer_unique_id ORDER BY
order_count DESC) AS row_num
   FROM customer_orders
)

SELECT
   customer_city,
   sum(order_count)
FROM ranked_orders
WHERE row_num = 1
group by customer_city
order by sum desc
```

Cities with the most orders;

customer_city	sum
sao paulo	14970
rio de janeiro	6614
belo horizonte	2670
brasilia	2066
curitiba	1462
campinas	1398
porto alegre	1326
salvador	1209
guarulhos	1151
sao bernardo do campo	908

For this question, the top 10 cities were selected from the output. The top 10 cities with the highest number of orders in the table also have the highest population density in Brazil. Here we can conclude that the number of orders is directly proportional to the population density. We can also see that the number of orders from Sao Paulo is quite high, more than twice that of the next city.

Case 3: Vendor Analysis

Question 1 :

-Who are the sellers who deliver orders to customers the fastest? Bring Top 5. Examine and comment on the number of orders of these sellers and the reviews and ratings of their products.

```
select
distinct oi.seller_id,
AVG(AGE(order_delivered_customer_date,order_purchase_timestamp)) as delivery_time,
count(o.order_id)as order_count,
AVG (r.review_score),
r.review_comment_message
from orders o
left join reviews r on r.order_id=o.order_id
left join order_items oi on o.order_id=oi.order_id
where order_status='delivered'
group by
oi.seller_id,o.order_delivered_customer_date,order_purchase_timestamp,r.review_score,r.re
view_comment_message
order by delivery_time asc
limit 5
```

seller_id	delivery_time	order_count	avg	review_comment_message
46dc3b2cc0980fb8ec44634e21d2718e	12:48:07	1	5.0	
f8db351d8c4c4c22c6835c19a46f01b0	18:45:10	1	3.0	
fdb9095204a334cd8872252ffec6f2db	20:31:39	1	3.0	
3b15288545f8928d3e65a8f949a28291	20:43:20	6	1.0	os produtos ainda nao foram entrgues
c847e075301870dd144a116762eaff9a	21:22:41	1	5.0	

The sellers who deliver their orders in the fastest way are as in the table. When we look at the table, we see that the number of orders is low. This may be a one-time situation, 4 sellers in the table sent one order and seem to have received 3 and 5 points. 4 sellers sent 6 orders, the delivery time is short, but the average score seems to be one, and in one comment it is mentioned that the order did not arrive. the delivery time of this sale may be incorrect.

Question 2 :
-Which sellers sell products from more categories? Do sellers with more categories also have more orders?

```
select
o.seller_id,
count(distinct product_category_name) as category_count,
count(distinct order_id) as order_count
from order_items o
left join products p on o.product_id=p.product_id
group by o.seller_id
order by category_count desc
limit 10
```

seller_id	category_count	order_count
b2ba3715d723d245138f291a6fe42594	27	337
955fee9216a65b617aa5c0531780ce60	23	1287
4e922959ae960d389249c378d1c939f5	23	420
1da3aeb70d7989d1e6d9b0e887f97c23	21	265
f8db351d8c4c4c22c6835c19a46f01b0	19	667
18a349e75d307f4b4cc646a691ed4216	17	121
6edacfd9f9074789dad6d62ba7950b9c	15	208
70a12e78e608ac31179aea7f8422044b	15	315
7178f9f4dd81dcef02f62acdf8151e01	14	203
8b28d096634035667e8263d57ba3368c	14	143

```
with cte as
(
select
o.seller_id,
count(distinct product_category_name) as category_count,
count(distinct order_id) as order_count
from order_items o
left join products p on o.product_id=p.product_id
group by o.seller_id
order by category_count desc
limit 10
)
select *,
order_count/category_count as rate
from cte
order by rate desc
```

seller_id	category_count	order_count	rate
955fee9216a65b617aa5c0531780ce60	23	1287	55
f8db351d8c4c4c22c6835c19a46f01b0	19	667	35
70a12e78e608ac31179aea7f8422044b	15	315	21
4e922959ae960d389249c378d1c939f5	23	420	18
7178f9f4dd81dcef02f62acdf8151e01	14	203	14
6edacfd9f9074789dad6d62ba7950b9c	15	208	13
1da3aeb70d7989d1e6d9b0e887f97c23	21	265	12
b2ba3715d723d245138f291a6fe42594	27	337	12
8b28d096634035667e8263d57ba3368c	14	143	10
18a349e75d307f4b4cc646a691ed4216	17	121	7

This question was analyzed with 2 outputs. In the first table, the sellers with the highest number of categories were ranked. In the second table, the number of orders of the sellers was divided by the number of categories and a ratio was obtained and a ranking was made. In one table, although the sellers changed in the ranking, the sellers maintained their place in the top 10. Here we can also see that the number of categories increases the number of orders.

Case 4 : Payment Analysis

Question 1 :

-In which region do users with a high number of installments live the most? Comment on this output.

```
with installment as (
    select
        c.customer_unique_id,
        c.customer_city,
        count(p.order_id) as installment_count
    from orders o
    left join customers c on o.customer_id=c.customer_id
    left join payments p on o.order_id=p.order_id
    where p.payment_installments > 1
    group by c.customer_unique_id, c.customer_city

)
select
            customer_city,
    count(customer_unique_id)
    from installment
group by 1
order by 2 desc
```

customer_city	count
sao paulo	6901
rio de janeiro	3543
belo horizonte	1450
brasilia	1028
salvador	707
porto alegre	678
curitiba	675
campinas	629
guarulhos	578
niteroi	422

In this question, the number of taxis with 2 or more taxis is included in the calculation and the top 10 cities are included in the table. In the table, we can see that Sao Paulo and Rio de Janeiro have the highest number of installments.

Question 2 :

-Calculate the number of successful orders and total amount of successful payments by payment type. Rank from the most used payment type to the least used payment type.

```
select
payment_type,
count( o.order_id),
case
when order_status='unavailable' or order_status='canceled' then
'unsuccessful' else 'successful'
end as odeme_durumu
from orders o
left join payments p on o.order_id=p.order_id
group by 1,3
order by 2 desc
```

payment_type	count	payment_status
credit_card	75905	successful
boleto	19539	successful
voucher	5613	successful
debit_card	1516	successful
credit_card	890	unsuccessful
boleto	245	unsuccessful
voucher	162	unsuccessful
debit_card	13	unsuccessful
not_defined	3	unsuccessful

```
select
payment_type,
sum(p.payment_value::int),
case
when order_status='unavailable' or order_status='canceled' then
'unsuccessful' else 'successful'
end as odeme_durumu
from orders o
left join payments p on o.order_id=p.order_id
group by 1,3
order by 2 desc
```

payment_type	sum	payment_status
credit_card	12350350	successful
boleto	2826928	successful
voucher	349829	successful
debit_card	212430	successful
credit_card	192043	unsuccessful
boleto	42560	unsuccessful
voucher	29562	unsuccessful
debit_card	5572	unsuccessful
not_defined	0	unsuccessful

The number of successful orders by payment type is tabulated as follows: credit card, boleto, voucher and debit card. We can see that payment by credit card is about 3 times the size of the total of other payment methods at a high rate.

Likewise, the order amounts according to the payment method are as in the table. In this table, we can see the same ranking, and we can also see a considerably higher amount compared to the other 3 methods under credit card payments.

Question 3 :

Make a category-based analysis of orders paid in single check and installments. Which categories use installment payments the most?

```
with installment_table
as (
select
o.order_id,
product_category_name,
payment_installments,
CASE
WHEN payment_installments = '1' THEN
'single_cekim_siparis' WHEN payment_installments != '1'
THEN 'installment_siparis' end as installment
from order_items o
left join products p on o.product_id=p.product_id
left join payments pa on o.order_id=pa.order_id
        )
        select
        product_category_name,
        count(order_id),
        installment
        from installment_table
        where
        installment='installment_sip
        aris' group by 1,3
        order by 2 desc
```

category_name_english	count	installment
bed_bath_table	7133	installment_siparis
health_beauty	5539	installment_siparis
furniture_decor	4503	installment_siparis
watches_gifts	4012	installment_siparis
sports_leisure	3961	installment_siparis
housewares	3779	installment_siparis
computers_accessories	3055	installment_siparis
cool_stuff	2309	installment_siparis
garden_tools	2218	installment_siparis
toys	2153	installment_siparis

In the table, the 10 categories with the highest number of installment orders were added. When we look at these categories, we see that there are furniture, home products, computers, sports products, watches and beautiful categories. The relatively higher prices of products in these categories may have led people to shop in installments.

Case 5 : RFM Analysis

Perform RFM analysis using the data set in e_commerce_data_.csv below. When calculating recency, use the date of the last order, not today's date.

```
WITH max_o_d AS (

SELECT
	customer_id,
	MAX(order_approved_at) AS max_order_approved_at
	FROM orders
	WHERE order_status IN ('canceled', 'unavailable')
	GROUP BY customer_id
),

recency AS (

   SELECT
	customer_id,
	max_order_approved_at,
	('2018-08-22' - max_order_approved_at:: date) AS recency
   FROM max_o_d
```

```
frequency AS (

    SELECT
        customer_id,
        COUNT(DISTINCT order_id) AS frequency
    FROM orders
    WHERE order_status IN ('canceled', 'unavailable')
    GROUP BY customer_id
),

monetary AS (

    SELECT
        customer_id,
        SUM(payment_value) AS monetary
    FROM orders o
    LEFT JOIN payments p ON o.order_id = p.order_id
    WHERE order_status IN ('canceled', 'unavailable')
    GROUP BY customer_id
),

scores AS (

   SELECT
        r.customer_id,
        r.recency,
        ntile(5) OVER (ORDER BY recency DESC) AS recency_score,
        f.frequency,
        CASE WHEN f.frequency >= 1 AND f.frequency <= 4 THEN f.frequency ELSE 5 END
AS frequency_score,
        m.monetary,
        ntile(5) OVER (ORDER BY monetary) AS monetary_score
    FROM recency r
    LEFT JOIN frequency f ON r.customer_id = f.customer_id
    LEFT JOIN monetary m ON r.customer_id = m.customer_id
),
merge_mont_fre AS (
    SELECT
        customer_id AS customer_id,
        recency_score,
        frequency_score + monetary_score AS mon_fre_score
    FROM scores
),

rfm_score AS (
    SELECT
        customer_id,
        recency_score,
```

```
        ntile(5) OVER (ORDER BY mon_fre_score) AS mon_fre_score
    FROM merge_mont_fre
)

SELECT * FROM rfm_score;
```

customer_id	recency_score	mon_fre_score
002b5342c72978cf0aba6aae1f5d5293	1	1
deac5870962fbc4af64040f2cbfd57d9	5	1
019f5bb93ed18dd059051c3f81abe394	5	1
36ecc6ff2c4a59fd150e9524b12b9259	4	1
259e07ebcf2a6be5fcd98f01099eb7f4	2	1
556b350ff8b8954e863415703f9b58cd	3	1
ca970e8045a80d7bfb9b994b321439f8	3	1
ad8939b7dfebf1e59e4145ed73bb0e33	3	1
8840e68ca97a89b084ca61284b559d41	5	1
0b1899150b4ee7ae778d02ecb621324f	1	1
e8c2884e61a7c098543e33fcc3ba4be2	2	1
eb55392e91ca4fb066e8c3d099400dd1	2	1
43b2ae9283ff026230ccdeaa82986b62	3	1