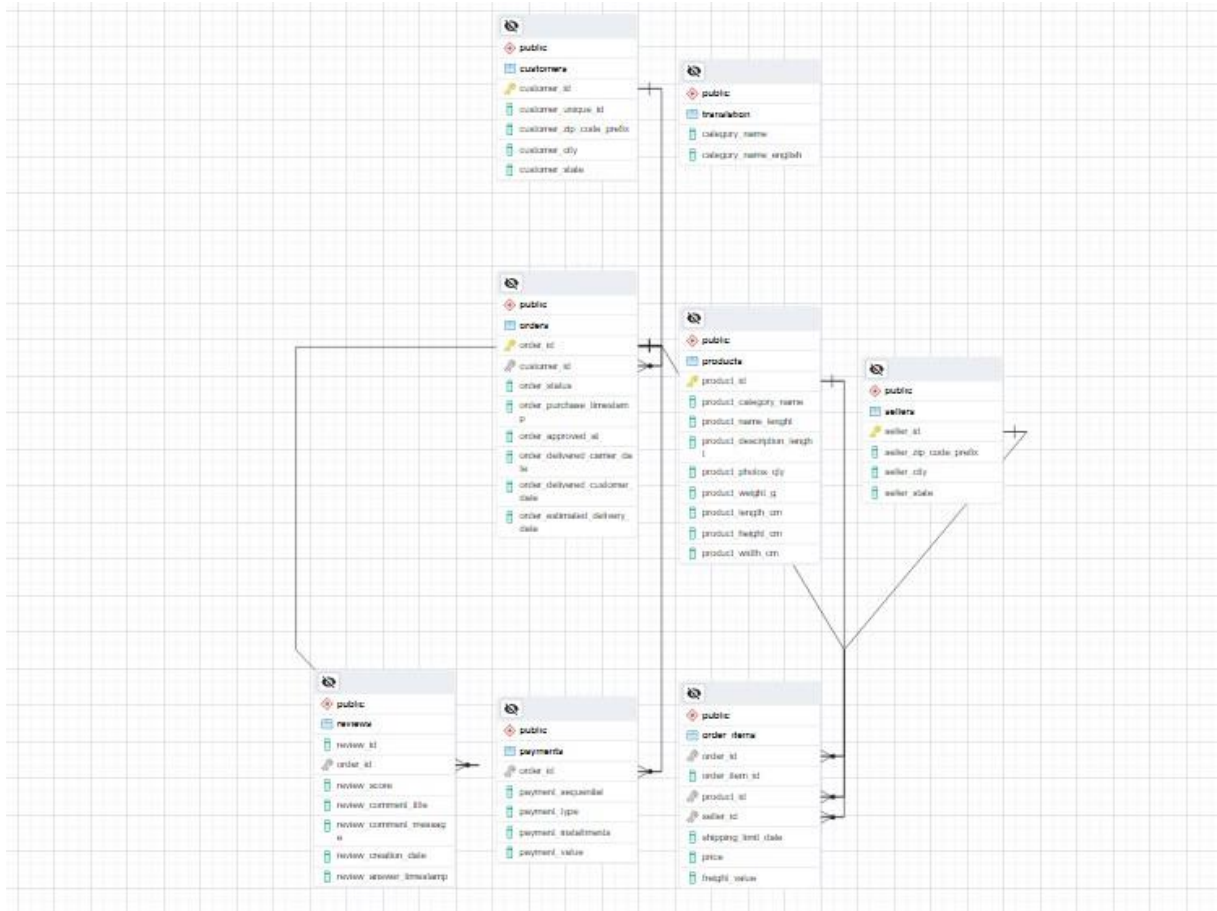


SQL PROJESİ



--Case 1 : Sipariş Analizi--

--Aylık olarak order dağılımını inceleyiniz. Tarih verisi için order_approved_at kullanılmalıdır.

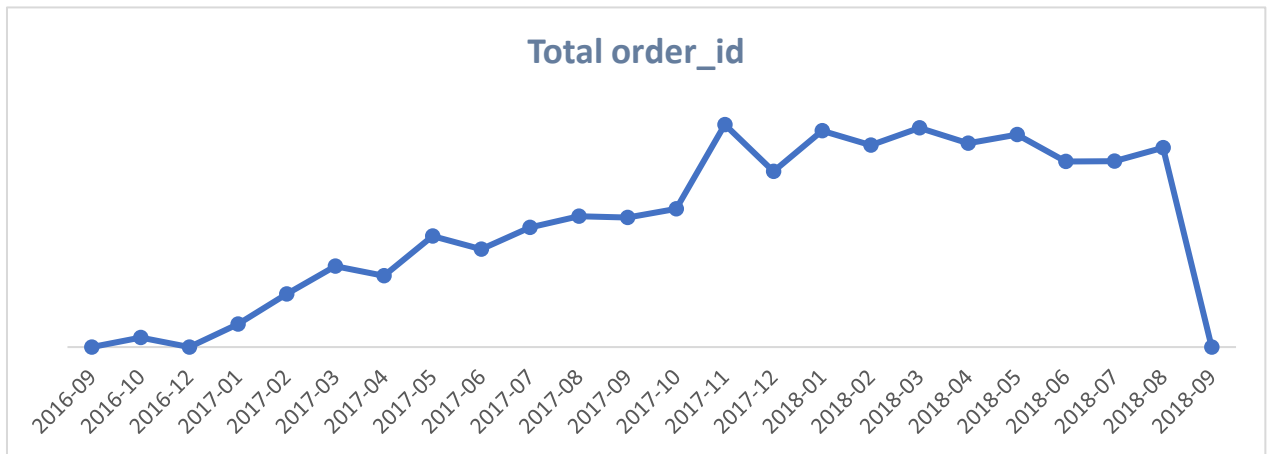
```
select
to_char(order_approved_at,'YYYY-MM')AS order_month,
count(order_id) as total
from orders
```

where order_approved_at IS NOT NULL

group by 1

order by 1

	order_month text	total bigint
1	2016-09	1
2	2016-10	320
3	2016-12	1
4	2017-01	760
5	2017-02	1765
6	2017-03	2689
7	2017-04	2374
8	2017-05	3693
9	2017-06	3252
10	2017-07	3974



Toplam sipariş sayıları 2017 yılının başlarından itibaren artış göstermiştir. 2017 yılının Kasım ayında maksimum seviyeye ulaşmıştır. Bunun sebebi yıl sonu yapılan indirimler ve yeni yılın yaklaşması olabilir.

--Aylık olarak order status kırılımında order sayılarını inceleyiniz. Sorgu sonucunda çıkan outputu excel ile görselleştiriniz. Dramatik bir düşüşün ya da yükselişin olduğu aylar var mı? Veriyi inceleyerek yorumlayınız.

```
select to_char(order_approved_at,'YYYY-MM')AS order_month,  
order_status,
```

count(order_id) as total_order

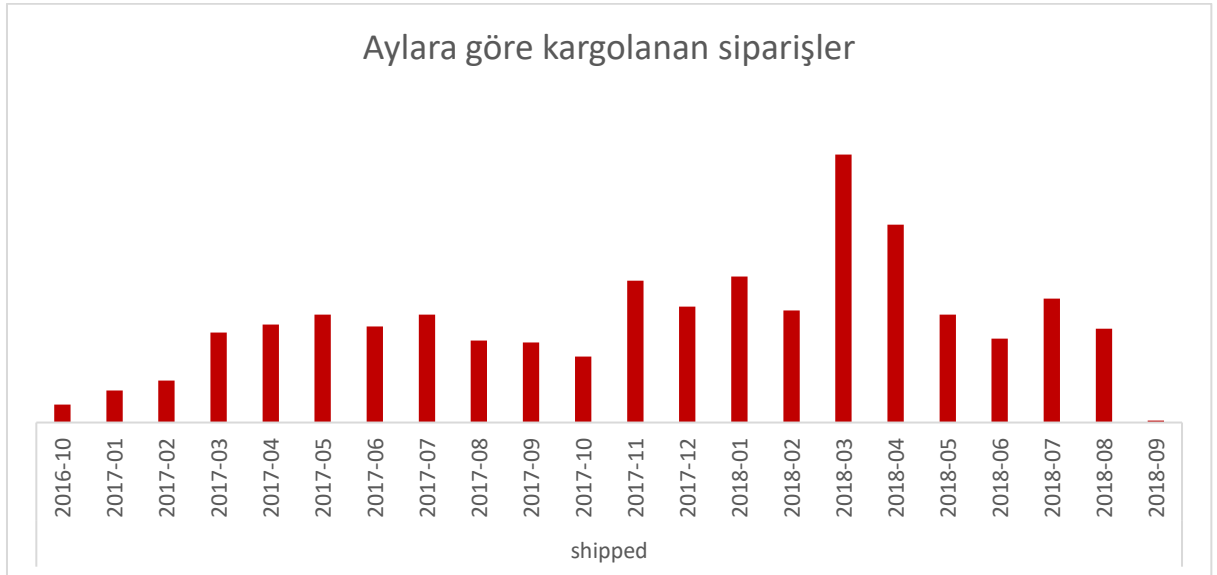
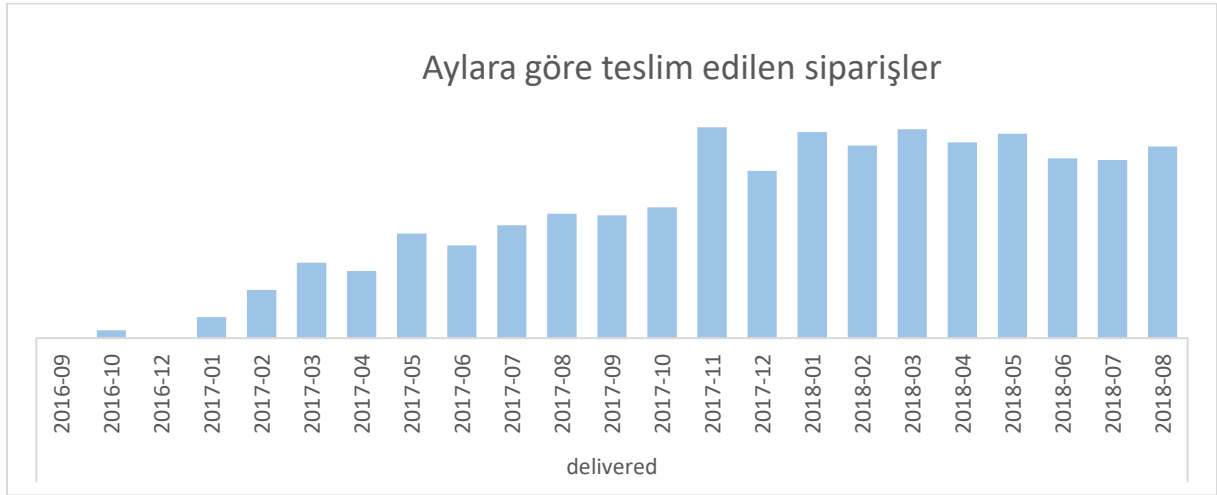
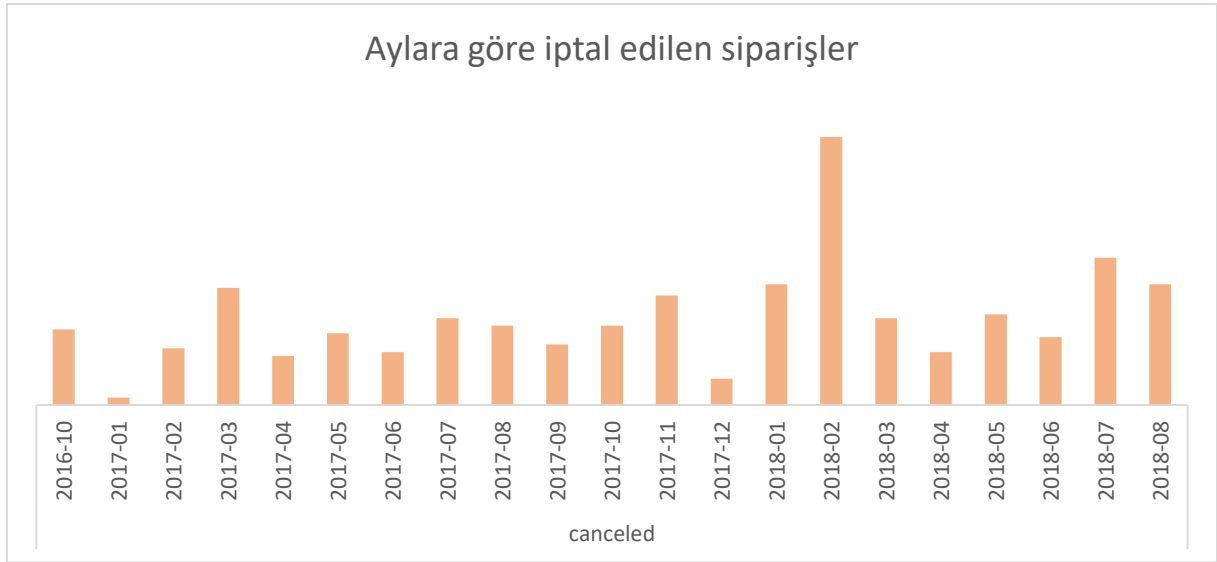
from orders

where order_approved_at IS NOT NULL

group by 1,2

order by 1,2

	order_month text	order_status character varying (50)	total_order bigint
1	2016-09	delivered	1
2	2016-10	canceled	20
3	2016-10	delivered	265
4	2016-10	invoiced	18
5	2016-10	processing	2
6	2016-10	shipped	9
7	2016-10	unavailable	6
8	2016-12	delivered	1
9	2017-01	canceled	2
10	2017-01	delivered	715



Tabloları incelediğimizde 2018 Şubat ayında iptal edilen sipariş sayısının maximum seviyesinde olduğunu görüyoruz. Teslim edilen siparişlerde de 2018 Şubat ayında bir düşüş olduğunu görüyoruz. Yani sipariş müşteriye ulaşmadan iptal edilmiş. Bunun sebebi ürünün tedarik sürecinin uzun olması, kargo sürecinin uzun olması gibi sebepler olabilir. Bu sorun kısa sürede çözülmüş görünüyor. Çünkü

2018 Mart ayında kargolanan ve teslim edilen siparişler artmış ve iptal edilen siparişler azalmıştır. 2016 yılında iptal edilen siparişler teslim edilen ve kargolanan siparişlere göre daha fazla gözüküyor. 2016 yılı için elimizde sadece 2-3 aylık veri var. Alışveriş sitesi yeni kurulmuş olabilir ya da o aylara ait veriler sisteme kaydedilmemiş olabilir. Ama teslim edilen siparişlerde 2017-2018 yılında ciddi artışlar olduğu için sektöre 2016 yılında yeni giriş yapmış olduğunu düşünüyorum.

--Ürün kategorisi kırılımında sipariş sayılarını inceleyiniz. Özel günlerde öne çıkan kategoriler nelerdir? Örneğin yılbaşı, sevgililer günü...

```
select category_name_english,  
count(o.order_id) as order_total  
from products as p join order_items as oi on p.product_id=oi.product_id  
join orders as o on o.order_id=oi.order_id  
join translation as t on p.product_category_name=t.category_name  
group by 1  
order by order_total desc
```

	category_name_english character varying	order_total bigint
1	bed_bath_table	11115
2	health_beauty	9670
3	sports_leisure	8641

Genel olarak ürün kategorisi kırılımında sipariş sayılarına baktığımızda 1.sırada ev dekorasyon ürünleri, 2.sırada sağlık-güzellik ürünleri, 3.sırada spor ürünlerinin yer aldığını olduğunu görmekteyiz.

--Özel günlerde öne çıkan kategoriler nelerdir? Örneğin yılbaşı, sevgililer günü

```
WITH tablo as(  
select product_category_name,  
to_char(order_approved_at,'MM') as order_date,  
count(o.order_id) as order_total  
from products as p join order_items as oi on p.product_id=oi.product_id  
join orders as o on o.order_id=oi.order_id  
where order_approved_at IS NOT NULL
```

```

group by 1,2)
select order_date,
order_total,
category_name_english
from tablo as tab
join translation t on t.category_name=tab.product_category_name
WHERE order_date in ('12')
-- WHERE order_date in ('05')
order by 1,2 desc

```

	order_date text	order_total bigint	category_name_english character varying
1	12	568	bed_bath_table
2	12	509	sports_leisure
3	12	483	health_beauty
4	12	470	toys

Aralık ayında En çok satış yapılan 4 ürünü incelediğimizde insanların genelde yeni yılda hayatlarında değişiklik yapmak istemesi sebebiyle ev dekorasyon ürünleri,spor ürünleri,sağlık-güzellik ürünlerinde sipariş sayılarının fazla olduğu görülmektedir. 4. Sırada yeni yılda çocuklara hediye alma durumu olduğu için oyuncak ürünlerinin de satışı fazla olduğu görülmektedir.

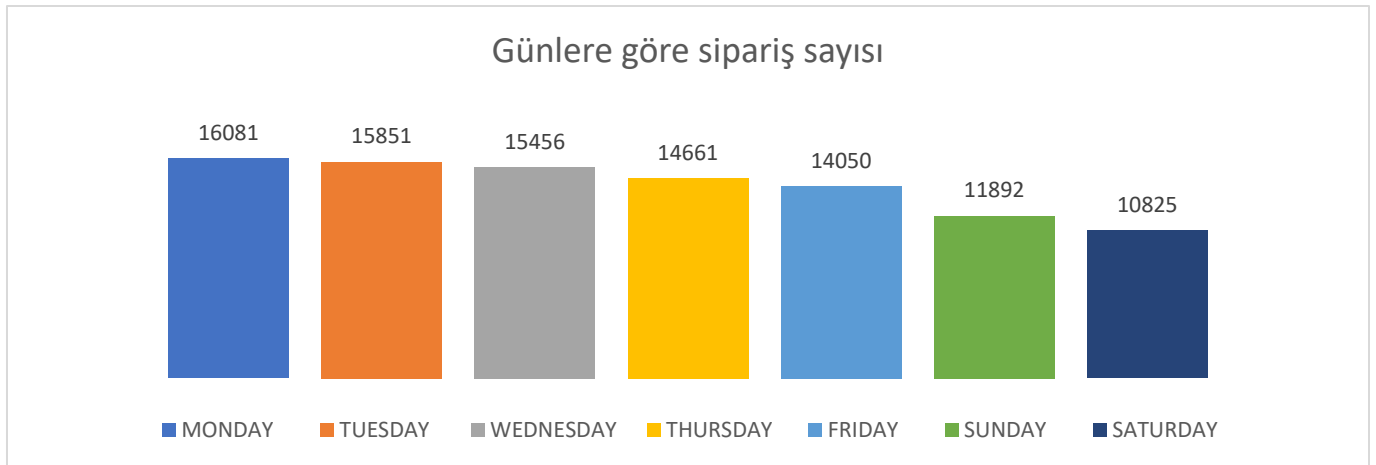
	order_date text	order_total bigint	category_name_english character varying
1	05	1140	bed_bath_table
2	05	1079	health_beauty
3	05	931	housewares
4	05	899	furniture_decor

Brezilya'da 14 Şubat günü sevgililer günü kutlanmıyor. Bunun yerine 12 Haziran'da sevgi günü adı verdikleri günü kutluyorlar. Sadece çiftler kutlama yapmıyor. Arkadaşlar,aile bireyleri kalabalık sofralarda bir araya geliyor.12 Hazirana yakın olan Mayıs ayındaki siparişlere baktığımızda ev dekorasyon ürünleri,sağlık-güzellik ürünleri,mutfak ürünleri,mobilya dekorasyon ürünleri siparişlerinin fazla olduğu görülmektedir. Kurulacak olan sofralar ve misafirler için mutfak ürünleri siparişinin ve mobilya dekorasyon ürünleri siparişinin arttığını düşünmekteyim. Mayıs ayındaki sipariş sayıları, hem anneler günü olması ve Haziran ayındaki sevgi gününün yaklaşması sebebiyle Aralık ayına göre iki kat fazla olabilir.

--Haftanın günleri(pazartesi, perşembe, ...) ve ay günleri (ayın 1'i,2'si gibi) bazında order sayılarını inceleyiniz. Yazdığınız sorgunun outputu ile excel'de bir görsel oluşturup yorumlayınız.

```
select
to_char(order_purchase_timestamp,'DAY') as days,
count(order_id) as total
from orders
where order_status != 'canceled'
group by 1
order by 2 desc
```

	days text	total bigint
1	MONDAY	16081
2	TUESDAY	15851
3	WEDNESD...	15456
4	THURSDAY	14661
5	FRIDAY	14050
6	SUNDAY	11892
7	SATURDAY	10825



Gün bazlı sipariş sayılarına baktığımızda en fazla siparişin pazartesi günü verildiğini görmekteyiz. Haftanın başı olması sebebiyle sipariş sayısı fazla olabilir. Haftasonuna doğru sipariş sayısı azalmaktadır.

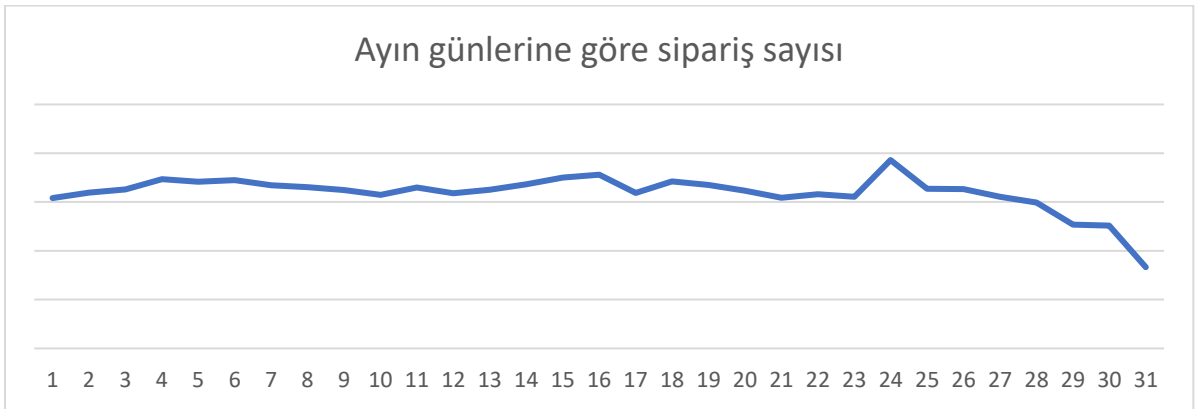
;

```

select
EXTRACT(DAY from order_purchase_timestamp) as monthdays,
count(order_id) as total
from orders
where order_status != 'canceled'
group by 1
order by 2 desc
;

```

	monthdays numeric	total bigint
1	24	3860
2	16	3558
3	15	3500
4	4	3469
5	6	3446
6	18	3420



Ayın günlerine göre sipariş sayısına baktığımızda en çok siparişin verildiği günler ayın ortası ve ayın 24'ü olduğunu görüyoruz. Maaşın yattığı günler olduğunu düşünüyorum.

--Case 2 : Müşteri Analizi --

--Hangi şehirlerdeki müşteriler daha çok alışveriş yapıyor? Müşterinin şehirini en çok sipariş verdiği şehir olarak belirleyip analizi ona göre yapınız.

```
WITH order_table AS (  
  select customer_unique_id,  
  customer_city,  
  count (DISTINCT order_id) as order_count  
  from orders as o join customers as c on o.customer_id=c.customer_id  
  group by 1,2  
)  
,  
city_table as(  
  select *,  
  ROW_NUMBER() OVER (PARTITION BY customer_unique_id ORDER BY order_count DESC ) as sr  
  from order_table  
)  
select customer_city,  
count(customer_unique_id) as total_customer  
from city_table  
where sr=1  
group by 1  
order by 2 DESC  
;
```

1	sao paulo	14965
2	rio de janeiro	6612
3	belo horizonte	2672
4	brasilia	2068
5	curitiba	1463

En çok sipariş verilen şehirler nüfusun fazla olduğu büyükşehirlerdir. Büyükşehirlerde ekonomik şartlar daha iyi olduğu için insanlar daha rahat alışveriş yapmaktadır. Bu sebeple sipariş miktarları daha fazla çıkmıştır.

--Case 3: Satıcı Analizi--

--Siparişleri en hızlı şekilde müşterilere ulaştıran satıcılar kimlerdir? Top 5 getiriniz. Bu satıcıların order sayıları ile ürünlerindeki yorumlar ve puanlamaları inceleyiniz ve yorumlayınız.

```
WITH table_data AS (
SELECT
o.order_id,
oi.seller_id,
(order_delivered_customer_date - order_approved_at) AS deliveredday,
r.review_score,
r.review_comment_message
FROM
orders AS o
JOIN order_items AS oi ON o.order_id = oi.order_id
JOIN sellers AS s ON oi.seller_id = s.seller_id
JOIN reviews AS r ON r.order_id = o.order_id
)
SELECT
seller_id,
COUNT(order_id) AS total_order,
AVG(deliveredday) AS avg_deliveredday,
ROUND(AVG(review_score),2) AS avg_score
FROM table_data
GROUP BY seller_id
HAVING COUNT (order_id) >30
ORDER BY avg_deliveredday asc
LIMIT 5
```

	seller_id character varying (100)	total_order bigint	avg_deliveredday interval	avg_score numeric
1	d13e50eaa47b4cbe9eb81465865d8c...	69	4 days 16:16:06.82353	4.81
2	6b90f847357d8981edd79a1eb1bf0a...	48	4 days 23:41:09.520833	4.38
3	6061155addc1e54b4cfb51c1c2a32a...	35	4 days 27:56:28.885714	4.71
4	d566c37fa119d5e66c4e9052e83ee4...	71	4 days 33:47:20.605634	4.72
5	18e694e0e48ed6f7aa3f24aade5fd697	34	5 days 15:43:51.764706	4.68

En hızlı sipariş yapan 5 müşteriyi bulmak için toplam sipariş sayısına 30 limiti getirdim. Çünkü hızlı teslimat yapan satıcının sadece birkaç ürün siparişi alması sağlıklı sonuç vermeyecektir. Teslimat günlerinin ve satıcı puanlarının ortalamasını aldım. İlk 5 satıcı ekteki gibidir. Satıcı yorumlarına baktığımızda da müşteriler hızlı teslimattan memnun olduklarını dile getirmişlerdir.

--Hangi satıcılar daha fazla kategoriye ait ürün satışı yapmaktadır?

--Fazla kategoriye sahip satıcıların order sayıları da fazla mı?

```
WITH tablo as (select s.seller_id,
oi.order_id,
category_name_english
from order_items as oi join products as p on oi.product_id=p.product_id
join sellers as s on oi.seller_id=s.seller_id
join translation as t on p.product_category_name=t.category_name
where product_category_name IS NOT NULL
)
select seller_id,
count (DISTINCT category_name_english) as total_category,
count (DISTINCT order_id) as total_order
from tablo
group by 1
order by 2 DESC
```

Daha fazla kategoriye göre satış yapan satıcılar aşağıdaki gibidir. Fazla kategoriye sahip müşteri daha fazla satış yapmamıştır. 27 kategoriye sahip müşteri 337 adet satış yaparken, 15 kategoriye sahip satıcı 315 satış yapabilmektedir. Sipariş sayısına göre sıralayıp baktığımızda 7 kategoriye sahip satıcı 1854 adet satış yapmıştır. Bunun sebebi bir satıcının diğer satıcılarda olmayan bir ürünü satıyor olması olabilir. Ya da ürünleri diğer satıcıya göre daha kaliteli olabilir, reklam ve pazarlama stratejileri gelişmiş olup daha fazla müşteriye ulaşmış olabilir.

	seller_id [PK] character varying (100)	total_category bigint	total_order bigint
1	b2ba3715d723d245138f291a6fe42...	27	337
2	4e922959ae960d389249c378d1c93...	23	405
3	955fee9216a65b617aa5c0531780c...	23	1287
4	1da3aeb70d7989d1e6d9b0e887f97...	21	265
5	f8db351d8c4c4c22c6835c19a46f01...	19	665
6	18a349e75d307f4b4cc646a691ed4...	17	121
7	70a12e78e608ac31179aea7f84220...	15	315
8	6edacfd9f9074789dad6d62ba7950...	15	208
9	2ff97219cb8622eaf3cd89b7d9c098...	14	89
10	7178f9f4dd81dcef02f62acdf8151e01	14	203
11	44073f8b7e41514de3b7815dd0237...	14	141
12	8b28d096634035667e8263d57ba33...	14	143

Sipariş sayısına göre sıraladığımızda,

	seller_id [PK] character varying (100)	total_category bigint	total_order bigint
1	6560211a19b47992c3666cc44a7e94c0	7	1854
2	4a3ca9315b744ce9f8e9374361493884	7	1806
3	cc419e0650a3c5ba77189a1882b7556a	4	1706
4	1f50f920176fa81dab994f9023523100	3	1404
5	da8622b14eb17ae2831f4ac5b9dab84a	4	1314

--Case 4 : Payment Analizi--

--Ödeme yaparken taksit sayısı fazla olan kullanıcılar en çok hangi bölgede yaşamaktadır?

--Bu çıktıyı yorumlayınız.

select customer_city,

```

customer_state,
payment_installments,
count( DISTINCT customer_unique_id) as count_customer
from payments as p
join orders as o on p.order_id=o.order_id
join customers as c on o.customer_id=c.customer_id
where payment_installments > 4
GROUP BY 1,2,3
HAVING count( DISTINCT customer_unique_id) >5
order by 3 DESC ,4

```

	customer_city character varying (50)	customer_state character varying (50)	payment_installments integer	count_customer bigint
1	rio de janeiro	RJ	15	6
2	rio de janeiro	RJ	12	10
3	sao paulo	SP	12	14
4	luis eduardo magalhaes	BA	10	6
5	itatiba	SP	10	6
6	lages	SC	10	6
7	linhares	ES	10	6

Ödeme yaparken taksit sayısını 4'ün üzerinde olacak şekilde ve müşteri sayısını 5 üzerinde olacak şekilde filtrelediğimizde yukarıdaki sonuç çıkmıştır. Taksit sayısının fazla olduğu bölgeler büyükşehirlerin bulunduğu bölgeler çıkmıştır. Büyükşehirlerde refah düzeyi hem düşük hem de yüksek bir çok insan yaşaması, bu bölgelerdeki sipariş sayısının fazla olması taksit sayısının fazla olmasını etkilemektedir. Genel olarak tabloya baktığımızda taksit sayısının fazla olduğu refah düzeyinin daha düşük olduğu şehirler bulunmaktadır.

--Ödeme tipine göre başarılı order sayısı ve toplam başarılı ödeme tutarını hesaplayınız. En çok kullanılan ödeme tipinden en az olana göre sıralayınız.

```

select DISTINCT payment_type,
COUNT(p.order_id) as total_order_id,

```

```

round(sum(p.payment_value::integer),2) as sum_payment
from orders as o join payments as p on o.order_id=p.order_id
where order_status ='delivered'
group by 1
order by 2 DESC,3

```

	payment_type character varying (100)	total_order_id bigint	sum_payment numeric
1	credit_card	74586	12101407.00
2	boleto	19191	2770059.00
3	voucher	5493	342972.00
4	debit_card	1486	208433.00

En çok kullanılan ödeme tipi kredi kartıdır.

--Tek çekimde ve taksitle ödenen siparişlerin kategori bazlı analizini yapınız.

--En çok hangi kategorilerde taksitle ödeme kullanılmaktadır?

```

WITH tablo as(
select category_name_english,
Count(DISTINCT o.order_id) as installment_order
from orders as o
JOIN payments as p on o.order_id=p.order_id
JOIN order_items as oi on p.order_id=oi.order_id
JOIN products as pr on oi.product_id=pr.product_id
JOIN translation t on pr.product_category_name=t.category_name
where payment_installments >1
Group by 1
order by 2 DESC
),
tablo2 as(

```

```

select category_name_english,
Count(DISTINCT o.order_id) as single_order
from orders as o
JOIN payments as p on o.order_id=p.order_id
JOIN order_items as oi on p.order_id=oi.order_id
JOIN products as pr on oi.product_id=pr.product_id
JOIN translation t on pr.product_category_name=t.category_name
where payment_installments =1
Group by 1
order by 2 DESC
)
select tablo.category_name_english,
tablo.installment_order,
tablo2.single_order
from tablo join tablo2 on tablo.category_name_english=tablo2.category_name_english
;

```

	category_name_english character varying	installment_order bigint	single_order bigint
1	bed_bath_table	5965	3535
2	health_beauty	5006	3880
3	watches_gifts	3794	1890
4	sports_leisure	3480	4299
5	furniture_decor	3353	3152
6	housewares	3197	2732
7	computers_accessories	2562	4177

En çok ev dekorasyon ürünleri, sağlık-güzellik ürünleri ve hediyeelik eşya kategorisinde taksit sayısı fazladır.

--Case 5 : RFM Analizi9 --

Aşağıdaki e_commerce_data_.csv doyasındaki veri setini kullanarak RFM analizi yapınız.
Recency hesaplarken bugünün tarihi değil en son sipariş tarihini baz alınız.

```
select * from rfm
```

```
;
```

```
--RECENCY
```

```
select max(invoicedate)
```

```
from rfm
```

```
--max inv date 2011-12-09
```

```
;
```

```
select customer_id,
```

```
max(invoicedate) as max_inv_date,
```

```
'2011-12-09'-max(invoicedate)::date as recency
```

```
from rfm
```

```
where customer_id IS NOT NULL
```

```
group by 1
```

```
--FREQUENCY
```

```
select customer_id,
```

```
count (DISTINCT invoiceno) as frequency
```

```
from rfm
```

```
where customer_id IS NOT NULL
```

```
group by 1
```

```
order by 2 DESC
```

```
--MONETARY
```

```
with tab as(select customer_id,
```

```
quantity*unitprice::numeric as price
```

```
from rfm
```

```
where customer_id IS NOT NULL and unitprice > 0)
```

```
select customer_id,
```

```
round(sum(tab.price),2) as monetary
```

```
from tab
```

```
group by 1
```

```
order by 2 desc
```

```
with recency as (select customer_id,
max(invoicedate) as max_inv_date,
'2011-12-09'-max(invoicedate)::date as recency
from rfm
where customer_id IS NOT NULL
group by 1),
frequency as(
select customer_id,
count (DISTINCT invoiceno) as frequency
from rfm
where customer_id IS NOT NULL
group by 1
order by 2 DESC
),
monetary as(with tab as(select customer_id,
quantity*unitprice::numeric as price
from rfm
where customer_id IS NOT NULL and unitprice > 0)
select customer_id,
round(sum(tab.price),2) as monetary
from tab
group by 1
order by 2 desc
)
select r.customer_id,
        recency,
        ntile(5) over(order by recency DESC) as recency_score,
        frequency,
        ntile(5) over(order by frequency) as frequency_score,
```

```

        monetary,
        ntile(5) over(order by monetary ASC) as monetary_score
from recency r
join frequency f on r.customer_id = f.customer_id
join monetary m on r.customer_id = m.customer_id

```

```

WITH recency AS (
    SELECT
        customer_id,
        MAX(invoicedate) AS max_inv_date,
        '2011-12-09'::DATE - MAX(invoicedate)::DATE AS recency
    FROM
        rfm
    WHERE
        customer_id IS NOT NULL
    GROUP BY
        customer_id
),

```

```

frequency AS (
    SELECT
        customer_id,
        COUNT(DISTINCT invoiceno) AS frequency
    FROM
        rfm
    WHERE
        customer_id IS NOT NULL
    GROUP BY
        customer_id
),

```

```

monetary AS (








```

```
WITH tab AS (  
    SELECT  
        customer_id,  
        quantity * unitprice::NUMERIC AS price  
    FROM  
        rfm  
    WHERE  
        customer_id IS NOT NULL  
        AND unitprice > 0  
)  
SELECT  
    customer_id,  
    ROUND(SUM(tab.price), 2) AS monetary  
FROM  
    tab  
GROUP BY  
    customer_id  
,  
scores AS (  
    SELECT  
        r.customer_id,  
        recency,  
        NTILE(5) OVER(ORDER BY recency DESC) AS recency_score,  
        frequency,  
        CASE  
            WHEN frequency BETWEEN 1 AND 4 THEN frequency  
            ELSE 5  
        END AS frequency_score,  
        monetary,  
        NTILE(5) OVER(ORDER BY monetary ASC) AS monetary_score  
FROM
```

```

    recency r
JOIN
    frequency f ON r.customer_id = f.customer_id
JOIN
    monetary m ON r.customer_id = m.customer_id
)
SELECT
    customer_id,
    recency_score::VARCHAR || '-' || frequency_score::VARCHAR || '-' || monetary_score::VARCHAR
AS RFM_Score
FROM
    scores

```

	customer_id character varying 	recency integer 	recency_score integer 	frequency bigint 	frequency_score integer 	monetary numeric 	monetary_score integer 
1	15165	373	1	1	2	487.75	3
2	15350	373	1	1	1	115.65	1
3	13747	373	1	1	2	79.60	1
4	14237	373	1	1	1	161.00	1
5	18011	373	1	1	1	102.79	1
6	17643	373	1	1	1	101.55	1
7	17908	373	1	1	1	243.28	2
8	16274	373	1	1	1	357.95	2
9	14729	373	1	1	1	313.49	2
10	17968	373	1	1	1	277.35	2

	customer_id character varying 	rfm_score text 
1	16274	1-1-2
2	17908	1-1-2
3	13747	1-1-1
4	12791	1-1-1
5	18011	1-1-1
6	14142	1-1-2
7	13065	1-1-1
8	17643	1-1-1
9	18074	1-1-3
10	15165	1-1-3
11	15350	1-1-1
12	14237	1-1-1
13	14729	1-1-2
14	17968	1-1-2
15	16583	1-1-1
16	16510	1-1-2
17	13108	1-1-2