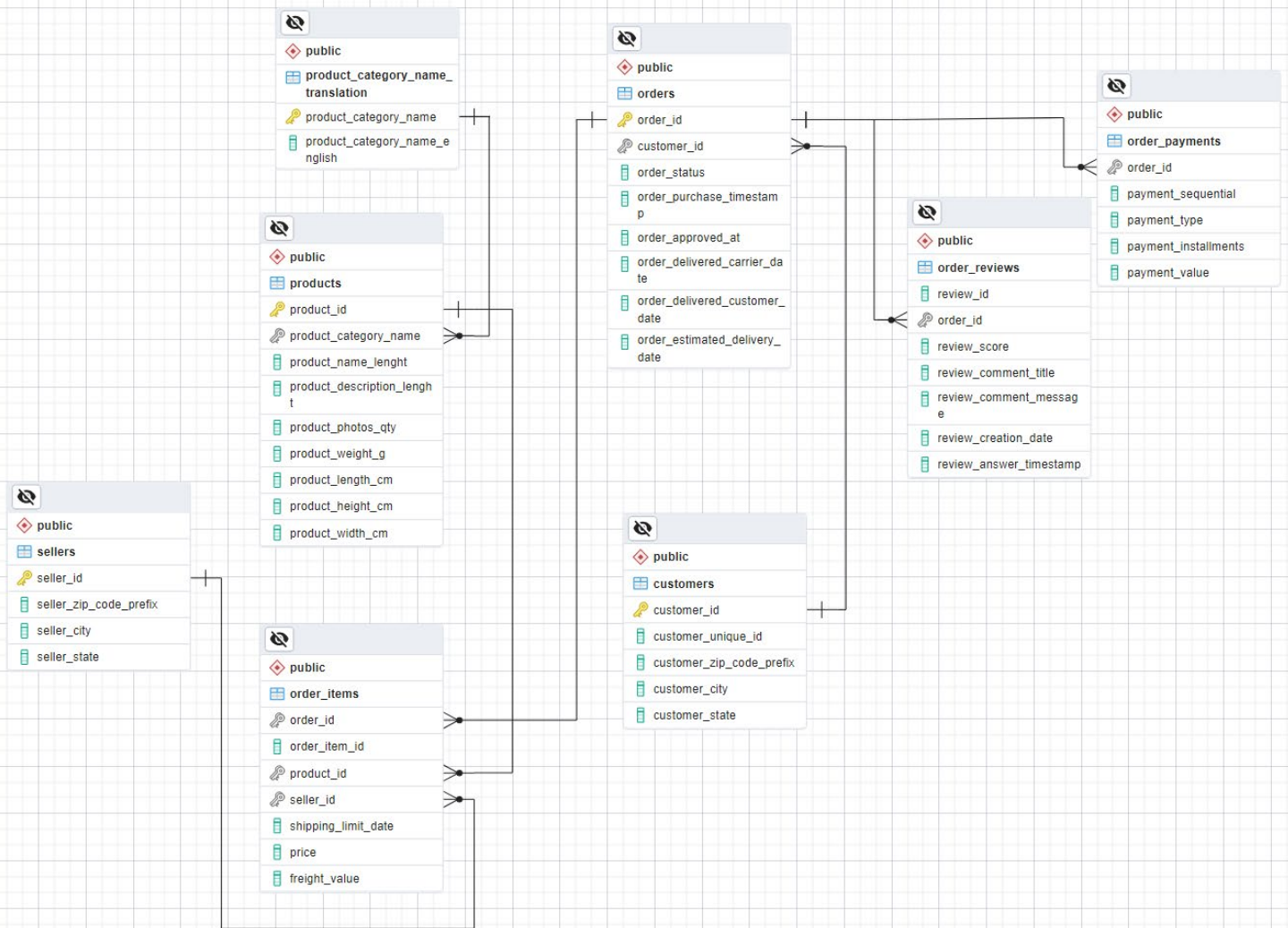


# SQL PROJE

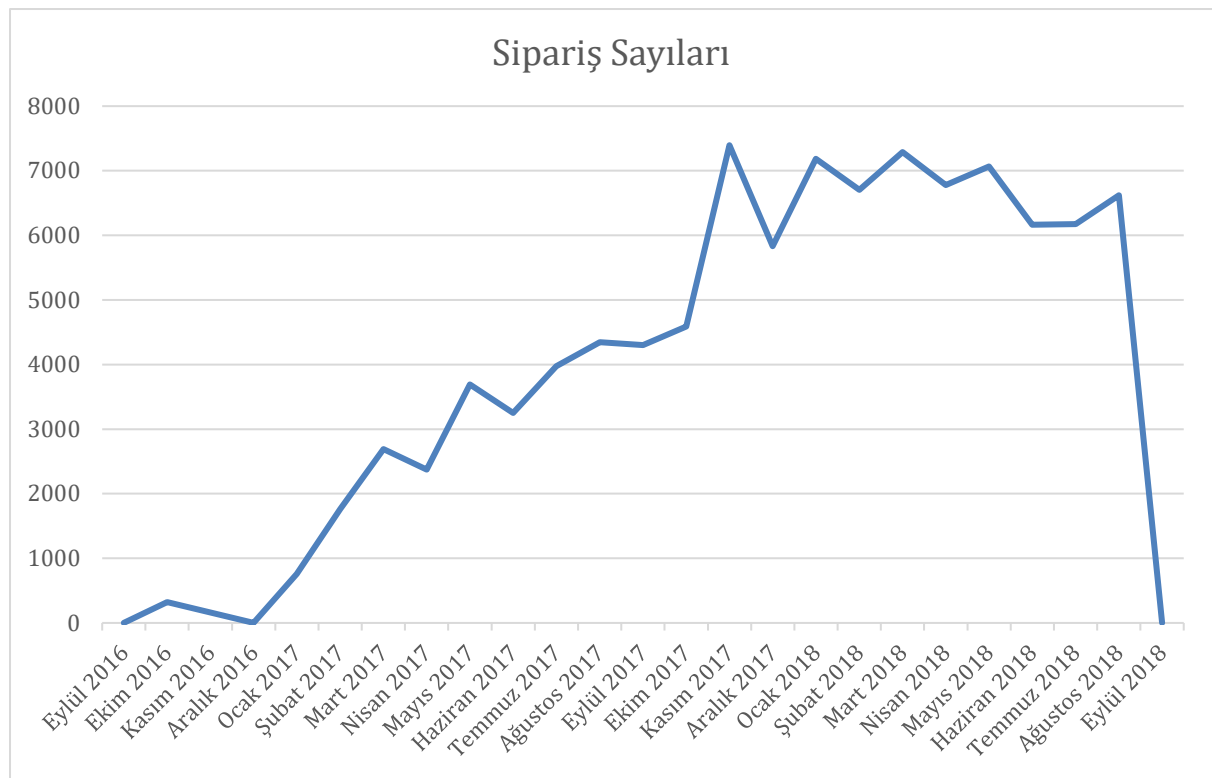


## CASE 1 : Sipariş Analizi

**Question 1:** -Aylık olarak order dağılımını inceleyiniz. Tarih verisi için order\_approved\_at kullanılmalıdır.

**Answer 1:**

```
SELECT
    TO_CHAR(order_approved_at, 'YYYY-MM') as Year_Month,
    COUNT(order_id) AS order_count
FROM
    orders
WHERE
    order_approved_at IS NOT NULL
GROUP BY
    1
ORDER BY
    1;
```



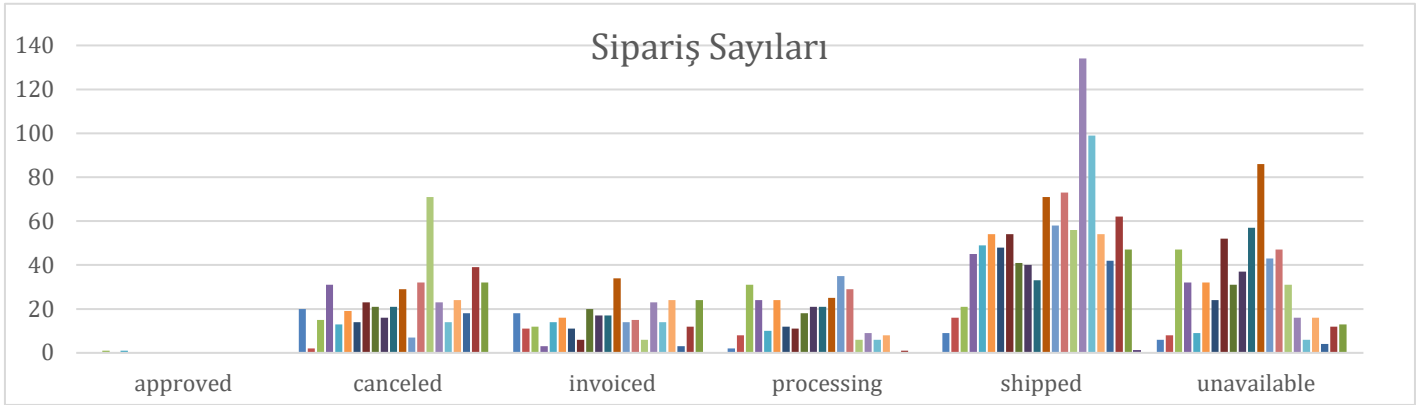
Sipariş sayılarındaki büyük dalgalanmalar dikkat çekiyor. Özellikle Kasım 2017 tarihindeki 7395 sipariş, bir önceki aya kıyasla belirgin bir artışı gösteriyor. Buna benzer aylarda gerçekleşen olayları ve kampanyaları incelemek, bu artışın sebebini anlamak açısından önemli olabilir. 2017 yılı için genel olarak sipariş sayılarında bir artış eğilimi var gibi görünüyor, ancak bazı aylarda düşüşler de yaşanmış. 2018 yılı içerisindeki ilk 8 ay ise ortalama olarak sipariş sayıları; artış ve düşüşlerle beraber ortalama 6500-7000 bandında seyretnmekte.

### Question 2:

-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

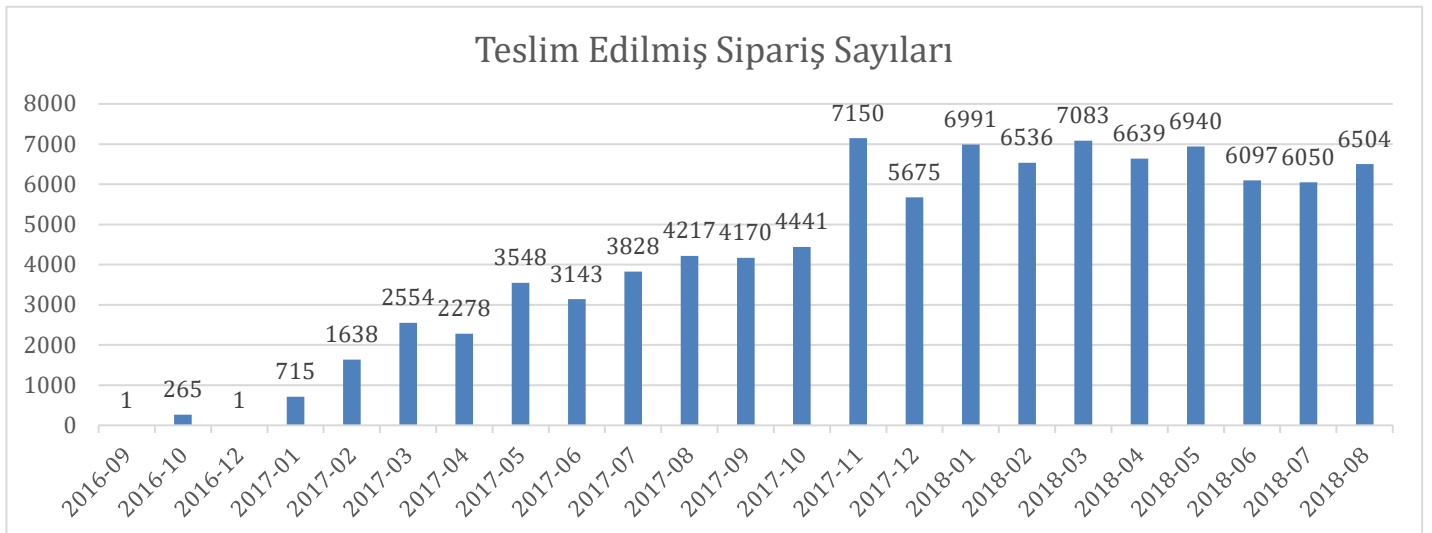
### Answer 2:

```
SELECT  
  TO_CHAR(order_approved_at, 'YYYY-MM') as Year_Month,  
  order_status,  
  COUNT(order_id) AS order_count  
FROM  
  orders  
GROUP BY  
  1,2  
ORDER BY  
  1,2
```



2018-Şubat ayında 71 iptal edilmiş sipariş göze çarpıyor. İptal sayıları genelde düşük olmakla birlikte, bu ayın öne çıktığını söyleyebiliriz.

Teslim edilen siparişlerde ise en yüksek toplam sipariş sayısı 7150 olmak üzere 2017 yılı Kasım ayında görünüyor. Toplamda 99281 sipariş var.



**Question 3:**

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

**Answer 3:**

```
SELECT
    pcn.product_category_name_english,
    TO_CHAR (o.order_purchase_timestamp, 'YYYY-MM-DD') as order_date,
    COUNT (oi.order_id) as order_count
FROM
    orders as o
JOIN
    order_items as oi
    ON o.order_id = oi.order_id
JOIN
    products as p
    ON oi.product_id = p.product_id
JOIN
    product_category_name_translation as pcn
    ON pcn.product_category_name = p.product_category_name
GROUP BY
    1,2
ORDER BY
    2,3 desc
```

Kategori bazında bir günde en çok sipariş verilen ürün 195 adet ile bed\_bath\_table (yatak banyo masası) olup 2017 yılın Kasım ayının son Cuma sına gelen 24 Kasım da sipariş almıştır. security\_and\_services (güvenlik ve hizmetler) ise bütün veri toplamında 2 sipariş ile en az sipariş verilen kategori olmuştur.

Tiradentes gününde (21 Nisan) 30 adet furniture\_decor (mobilya dekoru), Sevgililer gününde (14 Şubat) 46 adet ile heath\_beauty (sağlık ve güzellik) kategorilerinde sipariş verilmiştir.

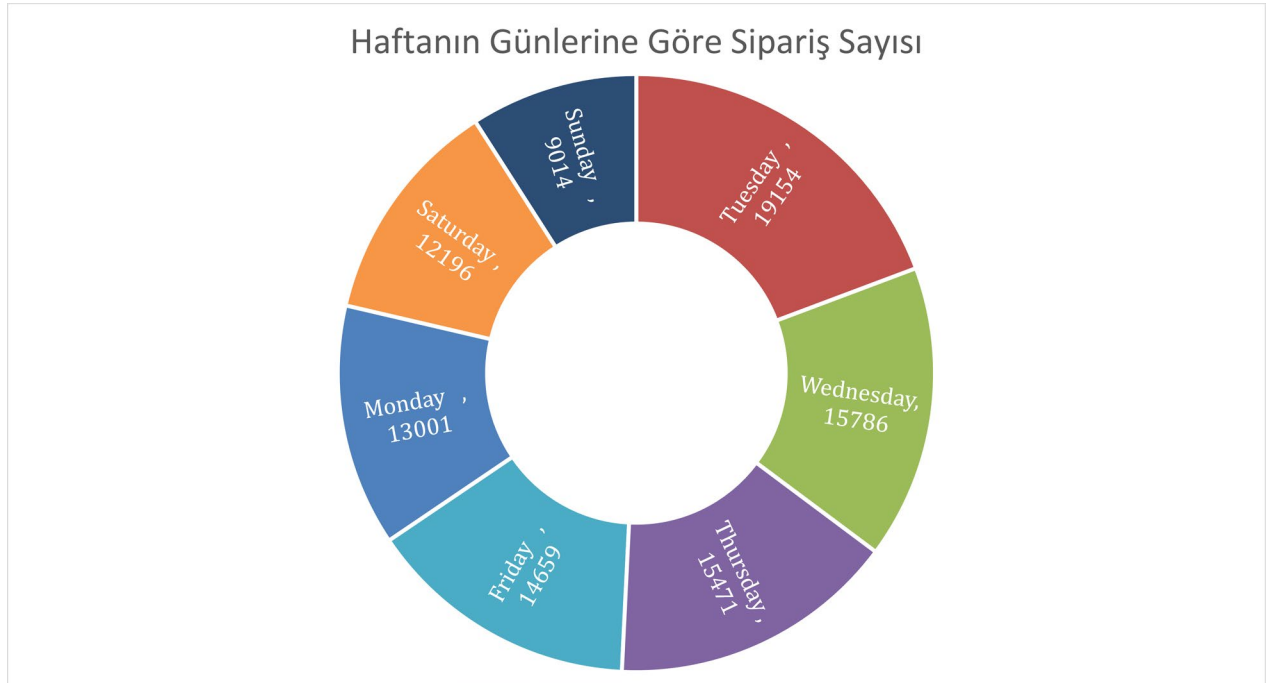
#### Question 4:

-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.

#### Answer 4:

4-A:

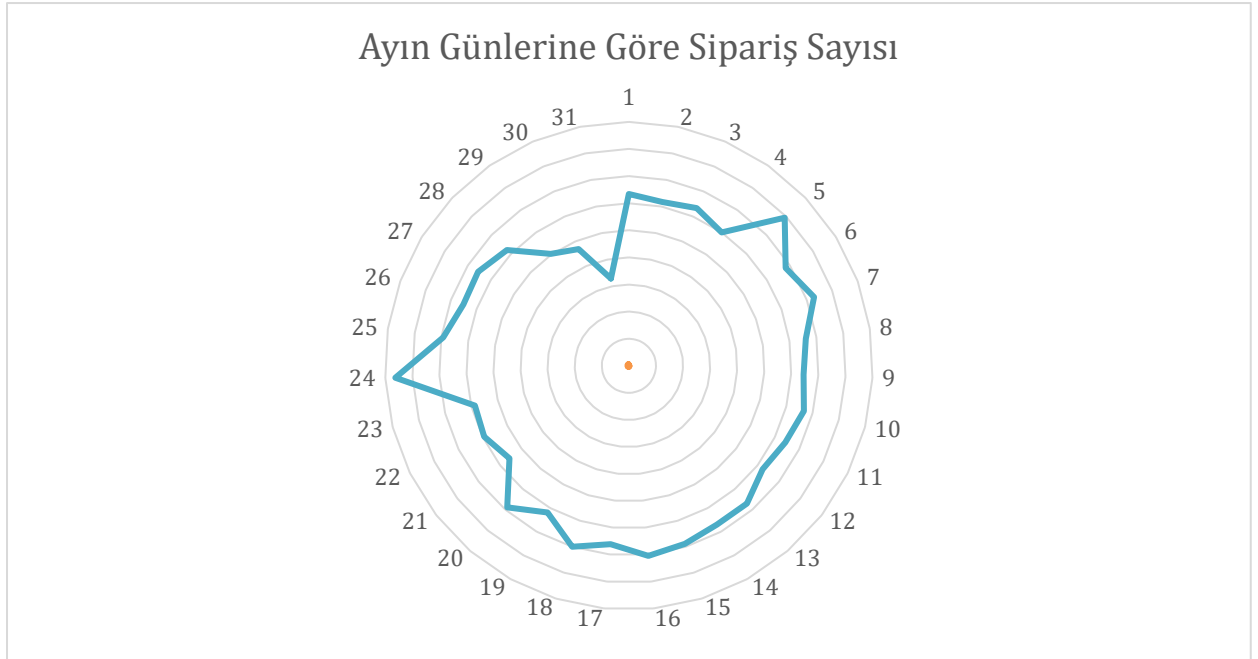
```
SELECT
    TO_CHAR(order_approved_at, 'Day') as "Day",
    COUNT (order_id) as order_count
FROM
    orders
WHERE
    order_approved_at IS NOT NULL
GROUP BY
    1
ORDER BY
    2 desc
```



Hafta içi günleri (Pazartesi'den Cuma'ya kadar) genellikle daha yüksek sipariş sayılarına sahiptir. Özellikle Salı günü, hafta içi günleri arasında en yüksek sipariş sayısına sahiptir. Hafta sonu günleri (Cumartesi ve Pazar), hafta içi günlerine kıyasla daha düşük sipariş sayılarına sahiptir. Bu durum, müşteri davranışlarının genellikle hafta içi günlerine odaklandığını gösteriyor olabilir. Salı günü diğer günlerden belirgin şekilde daha yüksek bir sipariş sayısına sahiptir. Bu durum, belki de özel bir promosyonun Salı günü yapıldığını veya belirli bir kampanyanın etkili olduğunu düşündürebilir.

4-B:

```
SELECT
    TO_CHAR(order_approved_at, 'DD') as "Days",
    COUNT (order_id) as order_count
FROM
    orders
WHERE
    order_approved_at IS NOT NULL
GROUP BY
    1
ORDER BY
    2 desc
```



Ayın 24'ü günündeki sipariş sayısı diğer günlerden belirgin şekilde daha yüksektir. Bu belirli kampanya dönemleri, özel günler veya indirim etkinlikleri gibi faktörlerle ilişkilidir. 29, 30 ve 31 günlerinde sipariş sayıları düşüş göstermekte. Bu durum müşterilerin ayın son gününde alışveriş yapmaya daha az eğilimli olduğunu gösteriyor.

## CASE 2 : Müşteri Analizi

### Question 1:

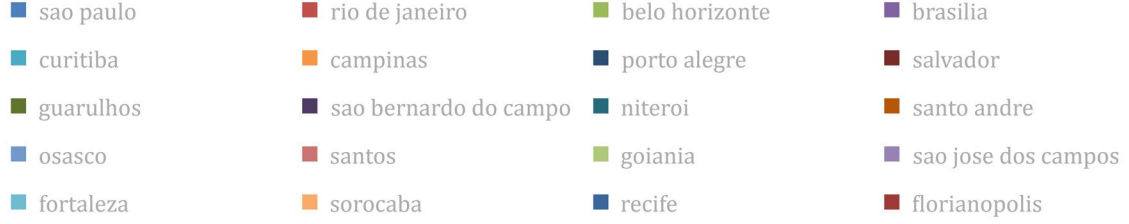
-Hangi şehirlerdeki müşteriler daha çok alışveriş yapıyor? Müşterinin şehrini en çok sipariş verdiği şehir olarak belirleyip analizi ona göre yapınız.

**Örneğin; Sibel Çanakkale'den 3, Muğla'dan 8 ve İstanbul'dan 10 sipariş olmak üzere 3 farklı şehirden sipariş veriyor. Sibel'in şehrini en çok sipariş verdiği şehir olan İstanbul olarak seçmelisiniz ve Sibel'in yaptığı siparişleri İstanbul'dan 21 sipariş vermiş şekilde görünmelidir.**

### Answer 1:

```
WITH order_counts as
(
    SELECT customer_unique_id,
           customer_city,
           COUNT(*) AS "total_order_count"
    FROM orders as o
    JOIN customers as c ON o.customer_id = c.customer_id
    GROUP BY 1,2
),
Customer_city_rn as
(
    SELECT row_number( ) OVER (PARTITION BY customer_id ORDER BY
order_count DESC) as rn,
           customer_id,
           customer_city
    FROM order_counts
),
customer_city as
(
    SELECT customer_id,
           customer_city
    FROM customer_city_rn
    WHERE rn = 1
)
SELECT cc.customer_ctiy,
       COUNT (o.order_id)
FROM orders as o
JOIN csutomer_city as cc
    ON o.customer_id = cc.customer_id
GROUP BY 1
ORDER BY 2 desc
```

## Şehirlere Göre Siparişlerin Dağılımı



## Müşterilerin En Çok Sipariş Verdiği Şehir

CUSTOMER UNIQUE ID	CITY	ORDER COUNT
8d50f5eadf50201ccdcedfb9e2ac8455	sao paulo	17
3e43e6105506432c953e165fb2acf44c	praia grande	9
1b6c7548a2a1f9037c1fd3ddfed95f33	ituiutaba	7
6469f99c1f9dfae7733b25662e7f1782	santos	7
ca77025e7201e3b30c44b472ff346268	recife	7
12f5d6e1cbf93dafd9dcc19095df0b3d	curitiba	6
63cfc61cee11cbe306bff5857d00bfe4	rio de janeiro	6
dc813062e0fc23409cd255f7f53c7074	garanhuns	6
f0e310a6839dce9de1638e0fe5ab282a	vitoria	6
de34b16117594161a6a89c50b289d35a	santo andre	6



## CASE 3 : Satıcı Analizi

### Question 1:

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

### Answer 1:

\* Satıcının sipariş sayısının en az 10 olacak şekilde listelenmesini isteyelim.

```
SELECT
    s.seller_id,
    s.seller_city,
    s.seller_state,
    COUNT(oi.order_id) AS order_count,
    ROUND(AVG(EXTRACT(EPOCH FROM (o.order_delivered_customer_date -
o.order_purchase_timestamp)) / 3600),2) AS avg_delivery_time_hours,
    COUNT(orf.review_comment_message) AS comment_count,
    ROUND(AVG(orf.review_score),2) AS avg_review_score
FROM
    sellers s
JOIN
    order_items oi ON s.seller_id = oi.seller_id
JOIN
    orders o ON oi.order_id = o.order_id
JOIN
    order_reviews orf ON o.order_id = orf.order_id
WHERE
    o.order_status = 'delivered'
GROUP BY
    1, 2, 3
HAVING
    COUNT(oi.order_id) > 10 *
ORDER BY
    avg_delivery_time_hours ASC
LIMIT 5;
```

Satıcılar genellikle SP bölgesinde yoğunlaşmış yani bu bölgenin ticaret faaliyetlerinde bir merkez olabileceğini gösteriyor. Satıcıların aldığı yorum sayıları 3 ile 6 arasında değişiyor. İnceleme puanları ise 4,46 ile 4,85 arasında değişiyor. Bu, müşterilerin genellikle olumlu değerlendirmeler yaptığını ve satıcıların güvenilir olduklarını gösteriyor.

**Question 2:**

-Hangi satıcılar daha fazla kategoriye ait ürün satışı yapmaktadır?  
Fazla kategoriye sahip satıcıların order sayıları da fazla mı?

**Answer 2:**

```
WITH SellerCategoryCounts AS (  
  SELECT  
    s.seller_id,  
    s.seller_city,  
    s.seller_state,  
    p.product_category_name,  
    COUNT(DISTINCT oi.order_id) AS order_count  
  FROM  
    sellers s  
  JOIN  
    order_items oi ON s.seller_id = oi.seller_id  
  JOIN  
    products p ON oi.product_id = p.product_id  
  GROUP BY  
    1, 2, 3, 4  
)  
  
SELECT  
  s.seller_id,  
  s.seller_city,  
  s.seller_state,  
  COUNT(DISTINCT scc.product_category_name) AS unique_category_count,  
  SUM(scc.order_count) AS total_order_count  
FROM  
  sellers s  
JOIN  
  SellerCategoryCounts scc ON s.seller_id = scc.seller_id  
GROUP BY  
  1, 2, 3  
ORDER BY  
  unique_category_count DESC;
```

## CASE 4 : Payment Analizi

### Question 1:

-Ödeme yaparken taksit sayısı fazla olan kullanıcılar en çok hangi bölgede yaşamaktadır? Bu çıktıyı yorumlayınız.

### Answer 1:

```
SELECT
    c.customer_state,
    COUNT(DISTINCT(c.customer_id)) as customer_count,
    op.payment_installments
FROM
    customers c
JOIN
    orders o ON c.customer_id = o.customer_id
JOIN
    order_payments op ON o.order_id = op.order_id
GROUP BY 3,1
ORDER BY 3 desc
```

Ödeme yaparken taksit sayısı fazla olan kullanıcıların çoğunlukla São Paulo (SP) bölgesinde yaşadığını söyleyebiliriz. İkinci sırada ise Rio de Janeiro (RJ) geliyor. Bu iki bölge, taksitli ödemelerin daha yaygın olduğu ve kullanıcıların bu ödeme seçeneğini tercih ettiği bölgeler olarak öne çıkıyor.

### Question 2:

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

### Answer 2:

```
SELECT
    op.payment_type,
    COUNT(DISTINCT o.order_id) AS successful_order_count,
    SUM(op.payment_value) AS total_successful_payment_amount
FROM
    order_payments op
JOIN
    orders o ON op.order_id = o.order_id
WHERE
    o.order_status = 'delivered' -- Sipariş başarıyla teslim edildiyse
GROUP BY
    1
ORDER BY
    2 DESC
```

**Question 3:**

-Tek çekimde ve taksitle ödenen siparişlerin kategori bazlı analizini yapınız. En çok hangi kategorilerde taksitle ödeme kullanılmaktadır?

**Answer 3:**

WITH single\_shot as

```
(
    SELECT
        pcnt.product_category_name_english,
        op.payment_installments,
        COUNT(DISTINCT oi.order_id) AS order_count
    FROM
        order_items oi
    JOIN
        order_payments op
        ON oi.order_id = op.order_id
    JOIN
        products p
        ON oi.product_id = p.product_id
    JOIN
        product_category_name_translation as pcnt
        ON p.product_category_name = pcnt.product_category_name
    WHERE
        op.payment_installments = 1 -- Tek çekim
    GROUP BY
        1,2
),
installment as
(
    SELECT
        pcnt.product_category_name_english,
        op.payment_installments,
        COUNT(DISTINCT oi.order_id) AS order_count
    FROM
        order_items oi
    JOIN
        order_payments op
        ON oi.order_id = op.order_id
    JOIN
        products p
        ON oi.product_id = p.product_id
    JOIN
        product_category_name_translation as pcnt
        ON p.product_category_name = pcnt.product_category_name
    WHERE
        op.payment_installments > 1 -- Taksit
    GROUP BY
        1,2
)
```

```

SELECT
    ss.product_category_name_english,
    ss.order_count as single_shot_order_count,
    SUM(ins.order_count) as installments_order_count
FROM single_shot as ss
JOIN installment as ins
    ON ss.product_category_name_english = ins.product_category_name_english
GROUP BY 1,2
ORDER BY 1

```

4299 adet tek seferlik siparişi ile sports\_leisure (spor eğlence) ve 5971 adet taksitli siparişi ile bad\_bath\_table (yatak banyo masası) kategorileri en yüksek sipariş sayılarına sahiptir.

## CASE 5 : RFM Analizi

### Question 1:

e\_commerce\_data\_.csv doyasındaki veri setini kullanarak RFM analizi yapınız.  
Recency hesaplarırken bugünün tarihi değil en son sipariş tarihini baz alınız.

### Question 1:

-- RECENCY --

```

SELECT
    CustomerID,
    MAX(InvoiceDate) AS LastPurchaseDate,
    EXTRACT(HOUR FROM('2011-12-09 12:50:00') - MAX(InvoiceDate)) AS Recency
FROM
    data
GROUP BY
    1

```

-- FREQUENCY --

```

SELECT
    customerID,
    COUNT(DISTINCT InvoiceNo) AS Frequency
FROM
    data
GROUP BY
    1

```

-- MONETARY --

```

SELECT

```

```
customerid,  
SUM(Quantity * UnitPrice) AS Monetary  
FROM  
data  
GROUP BY  
1  
----RFM---
```

```
WITH  
    recency as  
(  
        SELECT  
            CustomerID,  
            MAX(InvoiceDate) AS LastPurchaseDate,  
            ('2011-12-09 12:50:00') - MAX(InvoiceDate) AS Recency  
        FROM  
            data  
        GROUP BY  
            1  
    ),  
    frequency as  
(  
        SELECT  
            customerID,  
            COUNT(DISTINCT InvoiceNo) AS Frequency  
        FROM  
            data  
        GROUP BY  
            1  
    ),  
    monetary as  
(  
        SELECT  
            customerid,  
            SUM(Quantity * UnitPrice) AS Monetary  
        FROM  
            data  
        GROUP BY  
            1  
    )  
SELECT  
    r.customerid,  
    r.recency,  
    f.frequency,
```

```

        m.monetary
FROM
    recency as r
INNER JOIN frequency as f ON r.customerid = f.customerid
INNER JOIN monetary as m ON r.customerid = m.customerid

```

-----RFM SCORE-----

```

WITH
    recency AS
    (
        SELECT
            CustomerID,
            MAX(InvoiceDate::date) AS LastPurchaseDate,
            EXTRACT(DAY FROM ('2011-12-09 12:50:00'::timestamp -
            MAX(InvoiceDate::date))) AS RecencyInDays
        FROM
            data
        GROUP BY
            1
    ),
    frequency AS
    (
        SELECT
            customerID,
            COUNT(DISTINCT InvoiceNo) AS Frequency
        FROM
            data
        GROUP BY
            1
    ),
    monetary AS
    (
        SELECT
            customerid,
            SUM(Quantity * UnitPrice) AS Monetary
        FROM
            data
        GROUP BY
            1
    ),
    recency_quartiles AS
    (

```

```

SELECT
    PERCENTILE_CONT(0.20) WITHIN GROUP (ORDER BY RecencyInDays) AS
recency_quartile_1,
    PERCENTILE_CONT(0.40) WITHIN GROUP (ORDER BY RecencyInDays) AS
recency_quartile_2,
    PERCENTILE_CONT(0.60) WITHIN GROUP (ORDER BY RecencyInDays) AS
recency_quartile_3,
    PERCENTILE_CONT(0.80) WITHIN GROUP (ORDER BY
RecencyInDays) AS recency_quartile_4
FROM
    recency
),

frequency_quartiles AS
(
    SELECT
        PERCENTILE_CONT(0.20) WITHIN GROUP (ORDER BY Frequency) AS
frequency_quartile_1,
        PERCENTILE_CONT(0.40) WITHIN GROUP (ORDER BY Frequency) AS
frequency_quartile_2,
        PERCENTILE_CONT(0.60) WITHIN GROUP (ORDER BY Frequency) AS
frequency_quartile_3,
        PERCENTILE_CONT(0.80) WITHIN GROUP (ORDER BY
Frequency) AS frequency_quartile_4
    FROM
        frequency
),

monetary_quartiles AS
(
    SELECT
        PERCENTILE_CONT(0.20) WITHIN GROUP (ORDER BY Monetary) AS
monetary_quartile_1,
        PERCENTILE_CONT(0.40) WITHIN GROUP (ORDER BY Monetary) AS
monetary_quartile_2,
        PERCENTILE_CONT(0.60) WITHIN GROUP (ORDER BY Monetary) AS
monetary_quartile_3,
        PERCENTILE_CONT(0.80) WITHIN GROUP (ORDER BY Monetary)
AS monetary_quartile_4
    FROM
        monetary
)

SELECT
    r.CustomerID,
    CASE
        WHEN r.RecencyInDays < rq.recency_quartile_1 THEN 5
        WHEN r.RecencyInDays < rq.recency_quartile_2 THEN 4

```



```

        WHEN r.RecencyInDays < rq.recency_quartile_3 THEN 3
            WHEN r.RecencyInDays < rq.recency_quartile_4 THEN 2
        ELSE 1
    END AS RecencyScore,
    CASE
        WHEN f.Frequency < fq.frequency_quartile_1 THEN 1
        WHEN f.Frequency < fq.frequency_quartile_2 THEN 2
        WHEN f.Frequency < fq.frequency_quartile_3 THEN 3
            WHEN f.Frequency < fq.frequency_quartile_3 THEN 4
        ELSE 5
    END AS FrequencyScore,
    CASE
        WHEN m.Monetary < mq.monetary_quartile_1 THEN 1
        WHEN m.Monetary < mq.monetary_quartile_2 THEN 2
        WHEN m.Monetary < mq.monetary_quartile_3 THEN 3
            WHEN m.Monetary < mq.monetary_quartile_3 THEN 4
        ELSE 5
    END AS MonetaryScore,
    (
        CASE
            WHEN r.RecencyInDays < rq.recency_quartile_1 THEN 5
            WHEN r.RecencyInDays < rq.recency_quartile_2 THEN 4
            WHEN r.RecencyInDays < rq.recency_quartile_3 THEN 3
                WHEN r.RecencyInDays < rq.recency_quartile_4 THEN 2
            ELSE 1
        END +
        CASE
            WHEN f.Frequency < fq.frequency_quartile_1 THEN 1
            WHEN f.Frequency < fq.frequency_quartile_2 THEN 2
                WHEN f.Frequency < fq.frequency_quartile_3 THEN 3
                WHEN f.Frequency < fq.frequency_quartile_3 THEN 4
            ELSE 5
        END +
        CASE
            WHEN m.Monetary < mq.monetary_quartile_1 THEN 1
            WHEN m.Monetary < mq.monetary_quartile_2 THEN 2
            WHEN m.Monetary < mq.monetary_quartile_3 THEN 3
                WHEN m.Monetary < mq.monetary_quartile_3 THEN 4
            ELSE 5
        END
    ) AS TotalScore
FROM
    recency AS r
    INNER JOIN frequency AS f ON r.CustomerID = f.CustomerID
    INNER JOIN monetary AS m ON r.CustomerID = m.CustomerID
    CROSS JOIN recency_quartiles AS rq
    CROSS JOIN frequency_quartiles AS fq
    CROSS JOIN monetary_quartiles AS mq

```