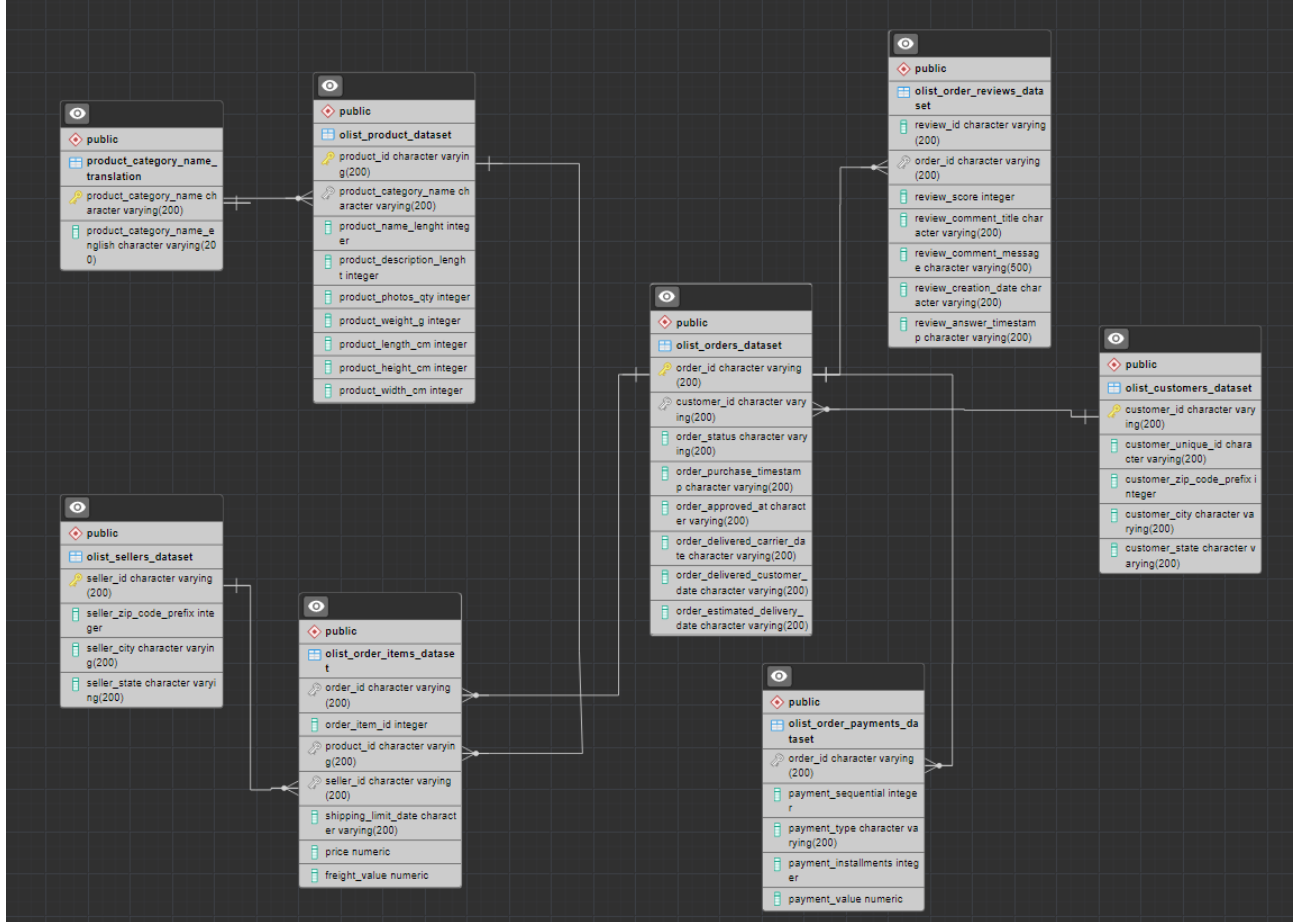


# SQL PROJESİ

Verilen linkteki veri setinden veri tabanını kullanılarak oluşturulan ERD şeması aşağıda verilmiştir.

Link: [Brazilian E-Commerce Public Dataset by Olist | Kaggle](https://www.kaggle.com/brazilian-e-commerce/public-dataset-by-olist)



## Case 1 : Sipariş Analizi

### Question 1 :

Aylık olarak order dağılımını inceleyiniz. Tarih verisi için order\_approved\_at kullanılmalıdır.

### SQL QUERY

```
SELECT
    date_trunc('month', order_approved_at::date) AS month,
    COUNT(order_id) AS order
FROM olist_orders_dataset WHERE order_approved_at IS NOT NULL
GROUP BY 1
ORDER BY 1;
```

### YORUM:

2016 yılının Eylül ayından başlayarak 2018 yılının Eylül ayına kadar (Eylül ayı dahil) toplamda 24 aylık veri elde edilmiştir. Aylara ait sipariş sayıları incelendiğinde genel olarak 1000 siparişin üzerinde olduğu görülmektedir. Sadece üç ayda birer sipariş verilmiştir. Diğer yandan 2016 yılının Kasım ayında herhangi bir sipariş verilmemiştir. Bazı aylarda artış görülürken bazı aylarda azalış gözlemlenmektedir.

### Question 2 :

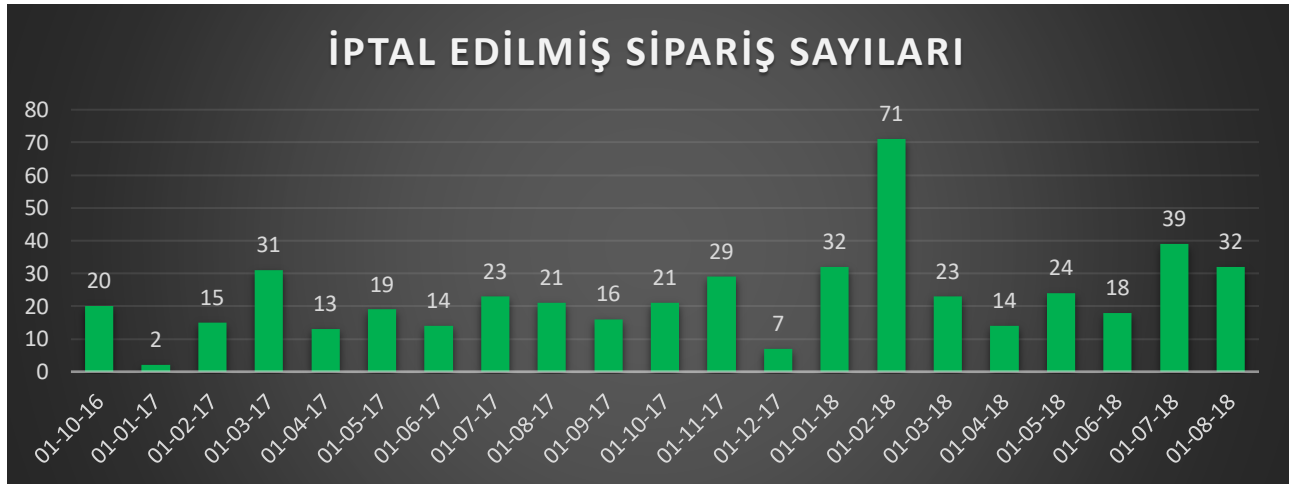
Aylık olarak order status kırılımında order sayılarını inceleyiniz.Sorgu sonucunda çıkan outputu excel ile görselleştiriniz. Dramatik bir düşüşün ya da yükselişin olduğu aylar var mı? Veriyi inceleyerek yorumlayınız.

### SQL QUERY

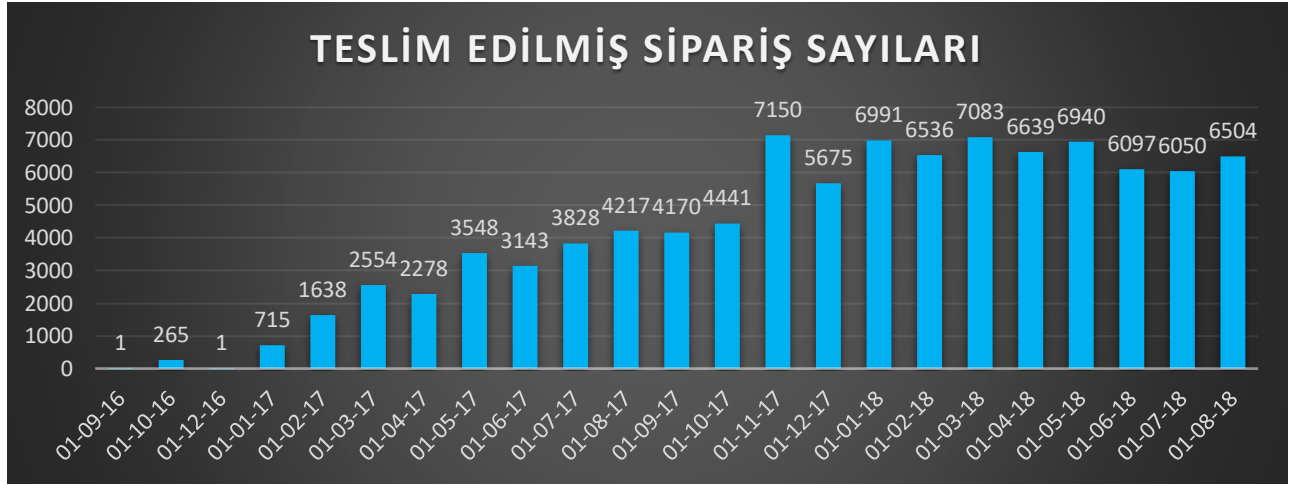
```
SELECT
    order_status,
    date_trunc('month', order_approved_at::date) AS month,
    COUNT(order_id) AS order
FROM olist_orders_dataset WHERE order_approved_at IS NOT NULL
GROUP BY 1, 2
ORDER BY 1, 2;
```

### YORUM:

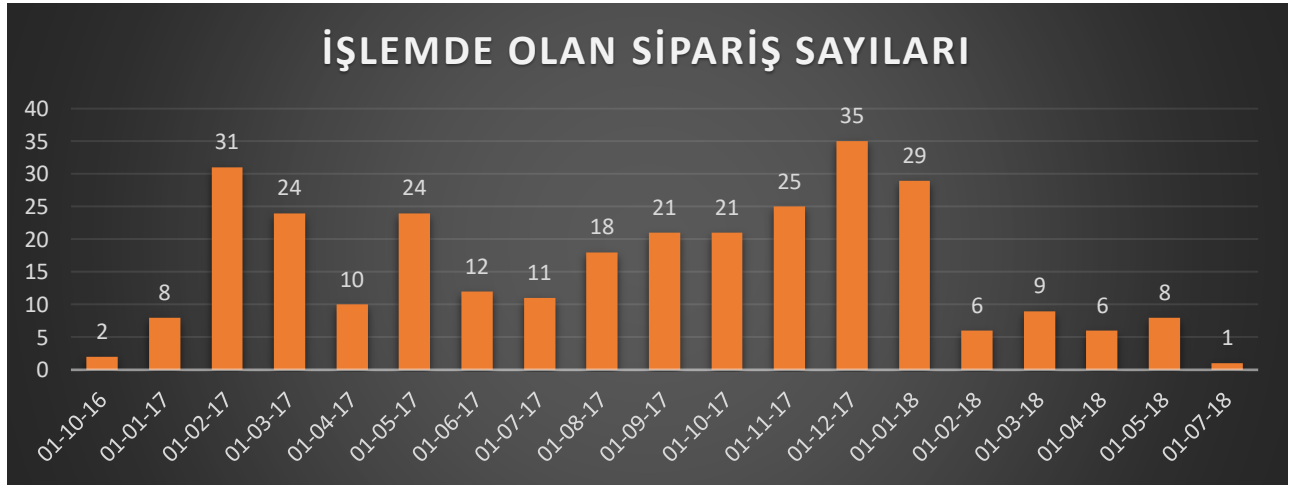
Aylık olarak sipariş durumu kırılımında sipariş sayıları incelenmiş ve artış azalışlarda dikkat çeken üç sipariş durumuna ait grafikler aşağıda verilmiştir.



İptal edilmiş sipariş sayılarının aylara göre değişim grafiği incelendiğinde Şubat 2018’de dramatik bir artış gözlenirken Aralık 2017 ve Mart 2018’de dramatik azalışlar görülmüştür.



Teslim edilmiş sipariş sayıları grafiği incelendiğinde aylara göre genel bir artışın olduğu görülmektedir. Eylül 2016 ve Aralık 2016 dönemlerinde yalnızca birer sipariş verilirken Kasım 2017’de dramatik bir artış görülmüştür.



İşlemde olan sipariş sayıları grafiği incelendiğinde Şubat 2017’de oldukça fazla bir artış gözlemlenirken, Şubat 2018’den itibaren önceki aylara kıyasla sipariş sayılarında düşüşler gözlemlenmiştir.

### Question 3 :

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

#### SQL QUERY

```
SELECT
    product_category_name AS category,
    EXTRACT('MONTH' FROM order_approved_at::date) AS month_number,
    COUNT(DISTINCT order_id) order_count
FROM olist_product_dataset
    LEFT JOIN olist_order_items_dataset USING(product_id)
    LEFT JOIN olist_orders_dataset USING(order_id)
WHERE order_approved_at IS NOT NULL AND product_category_name IS NOT NULL
GROUP BY 1,2;
```

#### YORUM:

Her kategorinin aylık bazda sipariş sayıları incelenmiştir. Birçok üründe bazı aylarda artış gözlemlense de çoğu ürün kategorisinde gerçekleşen artışın herhangi bir özel güne bağlı olmadığı görülmüştür.

Bebidas (içecek) ürün kategorisindeki sipariş sayısı en fazla Ocak ayında gerçekleşmiştir. Bu durumun nedeni olarak Brezilya'da yılbaşının Ocak ayında kutlanması gösterilebilir. Diğer yandan Climatizacao (klima) kategorisindeki en fazla sipariş sayısı ise Mart ayında gerçekleşmiştir. Brezilya'da Mart ayında yaz mevsimi yaşandığı için sipariş sayısının artması olağan görülmektedir.

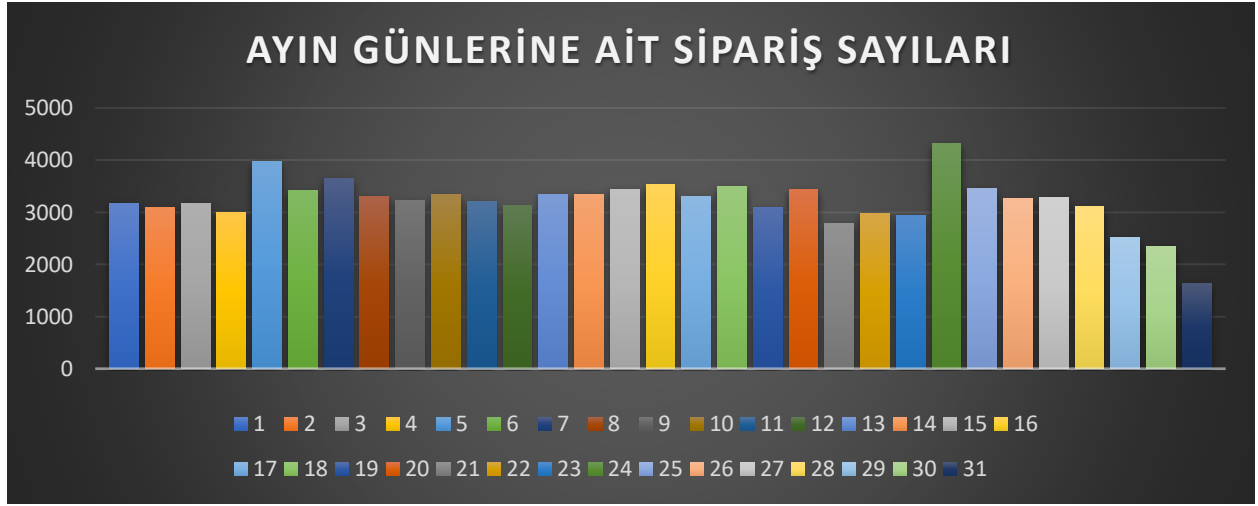
### Question 4 :

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.

#### SQL QUERY (AYIN GÜNLERİ)

```
SELECT
    EXTRACT(day FROM order_approved_at::date) AS days_of_month,
    COUNT(DISTINCT(order_id)) AS number_of_order
FROM olist_orders_dataset
WHERE order_approved_at IS NOT NULL
GROUP BY 1 ORDER BY 1;
```

## YORUM:



Excel'de elde edilen, ayın 31 gününe ait sipariş sayılarının gösterildiği grafik yukarıda verilmiştir. En fazla sipariş 24. gün verilirken, en az sipariş 31. gün verilmiştir. Genel olarak birbirine yakın sipariş sayıları görülse de ayın 5. günü ve 24. gününde diğer günlere kıyasla gözlemlenen artış daha fazladır. Diğer yandan bir önceki güne göre azalışların gerçekleştiği günler bulunmaktadır. Fakat bu azalışlar dramatik azalışlar değildir.

## SQL QUERY (HAFTANIN GÜNLERİ)

```
WITH ref_table AS
```

```
(
```

```
SELECT to_char(order_approved_at::date,'day') AS days_of_week,  
       EXTRACT(isodow FROM order_approved_at::date) AS names_of_week,  
       COUNT(DISTINCT(order_id)) AS number_of_order
```

```
FROM olist_orders_dataset
```

```
WHERE order_approved_at IS NOT NULL
```

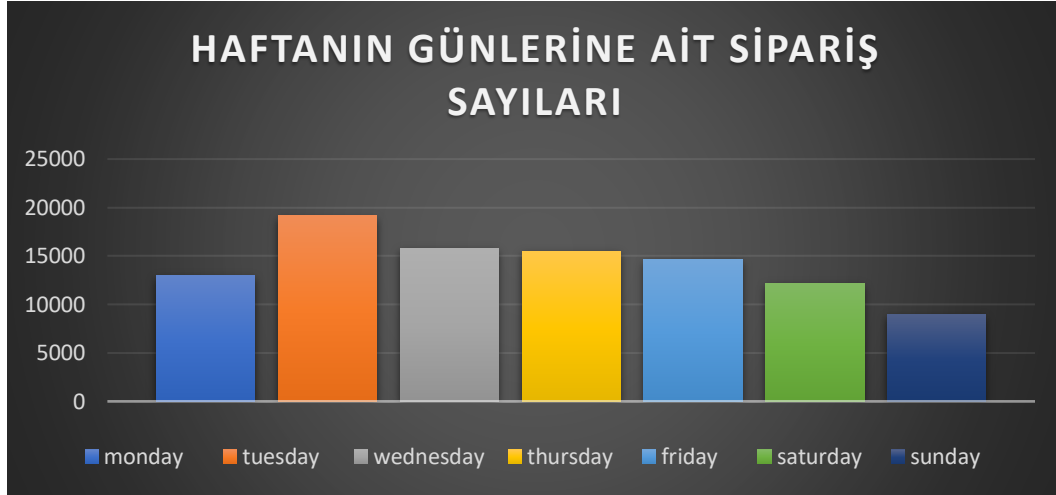
```
GROUP BY 1, 2 ORDER BY 2
```

```
)
```

```
SELECT days_of_week,  
       number_of_order
```

```
FROM ref_table;
```

#### YORUM:



Elde edilen grafik incelendiğinde en fazla siparişin Salı günü(19154) en az siparişin ise Pazar günü(9014) verildiği görülmektedir. Çarşamba ve Perşembe günü verilen sipariş sayıları birbirine çok yakındır. Sipariş sayıları haftasonu günlerinde hafta içi günlerine göre daha azdır.

#### Case 2 : Müşteri Analizi

##### Question 1 :

Hangi şehirlerdeki müşteriler daha çok alışveriş yapıyor?

#### SQL QUERY

```
SELECT
    customer_city,
    COUNT(DISTINCT order_id) AS number_of_order
FROM olist_customers_dataset
    JOIN olist_orders_dataset USING(customer_id)
GROUP BY 1 ORDER BY 2 DESC;
```

#### YORUM:

Müşterilerin yaptığı alışveriş sayıları şehir bazında elde edildiğinde en çok siparişin sırasıyla Sao Paulo(15540), Rio de Janeiro(6882), Belo Horizonte(2773) şehirlerinde verildiği görülmüştür.

## Question 2 :

Müşterinin şehri en çok sipariş verdiği şehir olarak belirleyip analizi ona göre yapınız.

### SQL QUERY

```
SELECT
    customer_id,
    customer_city,
    COUNT(DISTINCT customer_city) AS number_of_customer_city,
    ROW_NUMBER() OVER (PARTITION BY customer_id ORDER BY COUNT(DISTINCT customer_city))
    AS row_number
FROM olist_customers_dataset
GROUP BY 1;
```

### YORUM:

Sorgu sonucu elde edilen çıktı incelendiğinde tüm müşterilerin sadece bir şehirden sipariş verdiği görülmüştür. Bundan dolayı her müşterinin sipariş verdiği şehir en çok sipariş verdiği şehir olarak belirlenebilir.

## 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.

### SQL QUERY

```
SELECT
    seller_id,
    COUNT(DISTINCT order_id) AS number_of_order,
    (AVG(order_delivered_customer_date::timestamp - order_purchase_timestamp::timestamp)) AS
    duration
FROM olist_orders_dataset
    JOIN olist_order_items_dataset USING(order_id)
    JOIN olist_sellers_dataset USING(seller_id)
GROUP BY 1
    HAVING (AVG(order_delivered_customer_date::timestamp -
        order_purchase_timestamp::timestamp)) IS NOT NULL
ORDER BY 3 ASC
LIMIT 5;
```

#### YORUM:

SATICI_ID	SAYI	SÜRE	PUAN	YORUM
139157dd4daa45c25b0807ffff348363	1	1 day 05:08:25	4	YOK
5e063e85d44b0f5c3e6ec3131103a57e	1	1 day 06:55:12	5	YOK
6561d6bf844e464b4019442692b40e02	1	1 day 10:25:15	5	HARİKA
702835e4b785b67a084280efca355756	1	1 day 19:17:49	5	YOK
674207551483fec113276b67b0d871ff	1	1 day 20:52:08	5	YOK

Siparişleri müşterilere en hızlı ulaştıran satıcıların id numaraları yukarıdaki tabloda verilmiştir. Bunlardan ilki siparişini 1 gün 5 saat gibi kısa bir sürede müşteriye ulaştırmıştır. Bu satıcıların siparişleri için değerlendirmede yüksek puan verilse de genel olarak yorum yapılmamıştır. Sadece '6561d6bf844e464b4019442692b40e02' id numaralı satıcı için 'Harika' yorumu yapılmıştır.

#### Question 2 :

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

#### SQL QUERY

```
SELECT
    seller_id AS seller,
    COUNT(DISTINCT product_category_name) AS number_of_category,
    COUNT(DISTINCT order_id) AS number_of_order
FROM olist_product_dataset
JOIN olist_order_items_dataset USING(product_id)
WHERE product_category_name IS NOT NULL
GROUP BY 1
ORDER BY 2 DESC
LIMIT 5;
```

#### YORUM:

SATICI_ID	KATEGORİ SAYISI	SİPARİŞ SAYISI
"b2ba3715d723d245138f291a6fe42594"	27	337
"4e922959ae960d389249c378d1c939f5"	23	405
"955fee9216a65b617aa5c0531780ce60"	23	1287
"1da3aeb70d7989d1e6d9b0e887f97c23"	21	265
"f8db351d8c4c4c22c6835c19a46f01b0"	19	665

Ürün sattığı kategori sayısı en fazla olan 5 satıcının bilgileri yukarıdaki tabloda verilmiştir. 27 kategori ile en fazla kategoride ürün satan satıcının id numarası "b2ba3715d723d245138f291a6fe42594" şeklindedir. Bu satıcıların sipariş sayıları da az olmamakla birlikte kategori sayısındaki artışla paralel olarak ilişkilendirilememektedir. Tablodaki 5 satıcı arasında sipariş sayısı en fazla olan satıcı, ürün kategori sayısı 23 olan satıcılardan biridir.



#### Case 4 : Payment Analizi

##### Question 1 :

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

##### SQL QUERY

```
WITH ref_table AS
(
SELECT olist_customers_dataset.customer_id,
       payment_installments,
       customer_state
FROM olist_order_payments_dataset
     JOIN olist_orders_dataset USING(order_id)
     JOIN olist_customers_dataset USING(customer_id)
WHERE payment_installments>12
GROUP BY 1,2,3
ORDER BY 2 DESC
)
SELECT
       customer_state,
       COUNT(customer_state) AS number_of_customer_state
FROM ref_table
GROUP BY 1
ORDER BY 2 DESC;
```

##### YORUM:

Ülkemizde taksit sayısının üst limiti genellikle 12 olarak uygulanmaktadır. Bu sebeple analiz yaparken taksit sayısı için 12 baz alınmıştır. Sorgu sonucunda ödeme yaparken taksit sayısı fazla olan kullanıcıların daha çok 'SP'(Sao Paulo) bölgesinde yaşadığı belirlenmiştir. İkinci sırada ise 'RJ'(Rio de Janeiro) bölgesi bulunmaktadır.

## Question 2 :

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

### SQL QUERY

```
SELECT
    payment_type,
    COUNT(DISTINCT order_id) AS number_of_order,
    SUM(payment_value) AS total_payment
FROM olist_order_payments_dataset
JOIN olist_orders_dataset USING(order_id)
WHERE payment_value>0.00 AND order_status!='canceled' AND order_status!='unavailable'
GROUP BY 1
ORDER BY 2 DESC;
```

### YORUM:

ÖDEME TÜRÜ	BAŞARILI SİPARİŞ SAYISI	BAŞARILI ÖDEME TUTARI
Kredi Kartı	75618	12350042.56
Bilet	19539	2826802.30
Kupon	3745	349874.40
Banka Kartı	1515	212417.75

Ödeme türüne göre başarılı sipariş sayısı ve başarılı ödeme tutarı yukarıdaki tabloda verilmiştir. En fazla başarılı siparişin verildiği ödeme türü Kredi Kartı, en az başarılı siparişin verildiği ödeme türü ise Banka Kartıdır. Diğer yandan ödeme türüne göre başarılı sipariş sayısı arttıkça başarılı ödeme tutarının da arttığı görülmektedir.

### Question 3 :

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

#### SQL QUERY

```
WITH ref_table AS
(
SELECT
    product_category_name_english AS translate,
    product_category_name,
    payment_installments,
    CASE WHEN payment_installments=1 THEN 'tek_çekim' ELSE 'taksitli' END AS odeme_turu,
    order_id
FROM product_category_name_translation
    JOIN olist_product_dataset USING(product_category_name)
    JOIN olist_order_items_dataset USING(product_id)
    JOIN olist_order_payments_dataset USING(order_id)
)
SELECT
    translate,
    odeme_turu,
    COUNT( order_id) AS number_of_order
FROM ref_table GROUP BY 2,1
ORDER BY 3 DESC;
```

#### YORUM:

KATEGORİ	ÖDEME TÜRÜ	SİPARİŞ SAYISI
Yatak_Banyo_Masa	Taksitli	7133
Sağlık_Güzellik	Taksitli	5539
Bilgisayar_Aksesuarları	Tek Çekim	5027
Spor	Tek Çekim	4984
Yatak_Banyo_Masa	Tek Çekim	4690
Mobilya Dekorasyon	Taksitli	4503

Taksitle ödeme yapılan kategoriler incelendiğinde en çok siparişin sırasıyla 'Yatak\_Banyo\_Masa', 'Sağlık\_Güzellik' ve 'Mobilya Dekorasyon' kategorilerinden verildiği görülmektedir. Tek çekimle yapılan ödemelerde ise ilk üç sırayı 'Bilgisayar Aksesuarları', 'Spor' ve 'Yatak\_Banyo\_Masa' kategorileri almaktadır.

## RFM ANALIZI

```
WITH segment AS
(
    WITH last_scores AS
    (
        WITH scores AS
        (
            WITH recency AS
            (
                WITH last_date AS
                (
                    SELECT
                        customer_id,
                        MAX(invoice_date::date) AS last_invoice_date
                    FROM e_commerce_data
                    GROUP BY customer_id
                )
                SELECT
                    customer_id,
                    '2011-12-09 12:50:00'-last_invoice_date AS recency
                FROM last_date
                WHERE customer_id IS NOT NULL
            ),

            frequency AS
            (
                SELECT
                    customer_id,
                    COUNT(DISTINCT invoice_no) frequency
                FROM e_commerce_data
                GROUP BY customer_id
            ),

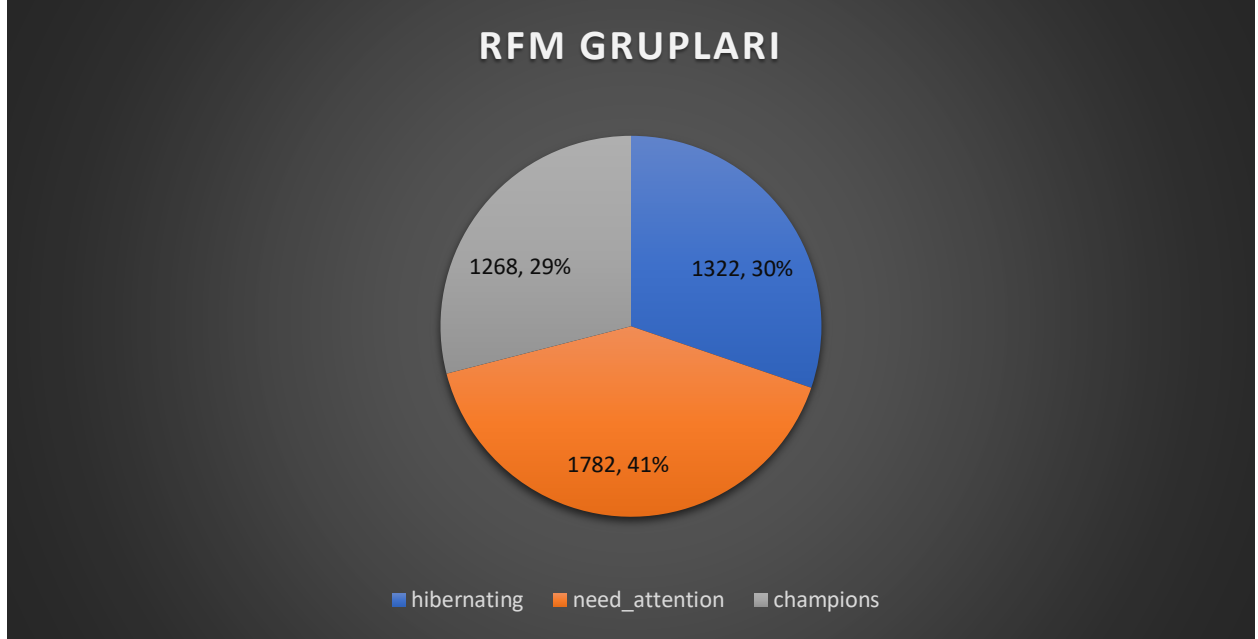
            monetary AS
            (
                SELECT
                    customer_id,
                    SUM(unit_price*quantity) AS monetary
                FROM e_commerce_data
                WHERE customer_id IS NOT NULL
                GROUP BY customer_id
            )
            SELECT *
            FROM recency
            JOIN frequency USING(customer_id)
            JOIN monetary USING(customer_id)
        )
    )
)
```

```

SELECT
    customer_id,
    recency,
    frequency,
    monetary,
    NTILE(4) OVER (ORDER BY recency DESC) AS rec_score,
    NTILE(4) OVER (ORDER BY frequency ASC) AS frq_score,
    NTILE(4) OVER (ORDER BY monetary ASC) AS mnt_score,
    NTILE(4) OVER (ORDER BY recency DESC)+NTILE(4) OVER (ORDER BY frequency ASC)+NTILE(4)
OVER (ORDER BY monetary ASC) AS rfm_score
    FROM scores order by rfm_score desc
)
SELECT *,
    CASE WHEN rfm_score<=12 AND rfm_score>=10 THEN 'champions'
    WHEN rfm_score<=9 AND rfm_score>=6 THEN 'need_attention'
    ELSE 'hibernating' END AS segments FROM last_scores
)
SELECT
    segments,
    COUNT(segments)
FROM segment
GROUP BY segments

```

## YORUM:



- ✓ Recency: Müşterilerin recency değerleri hesaplanırken güncel tarih yerine veri setindeki son fatura tarihi baz alınmıştır.
- ✓ Frequency: Müşterilerin kaçar tane sipariş verdiği bilgisi count(distinct invoice\_no) komutu ile elde edilmiştir.
- ✓ Monetary: Müşterilerin monetary değerleri bulunurken birim fiyat ile ürün adedi çarpılarak her müşteriye ait toplam fatura tutarı hesaplanmıştır.

Daha sonra tüm bu değerler için NTILE(4) komutu kullanılarak recency puanı, frequency puanı ve monetary puanı bulunmuştur. Bu puanlar toplanarak her bir müşteriye ait RFM skoru elde edilmiştir. Bu skorlar;

12-10='Champions'

9-6= 'Need\_Attention'

5-3= 'Hibernating'

şeklinde gruplandırılarak müşterilerin hangi gruba ait olduğu belirlenmiştir.

Müşterilerin %30'u Hibernating grubunda yer almaktadır. Bu gruptaki müşteriler yakın zamanda alışveriş yapmamış, alışveriş sıklıkları ve fatura tutarları az olan kişilerdir. Özellikle müşterilerin %41'ini oluşturan need\_attention grubu için çeşitli kampanyalar düzenlenerek yakın zamanda alışveriş yapmaları, alışveriş sıklıklarını ve fatura tutarlarını arttırmaları sağlanabilir. Bu sayede istenilen grup olan champions grubuna dâhil olabilirler.