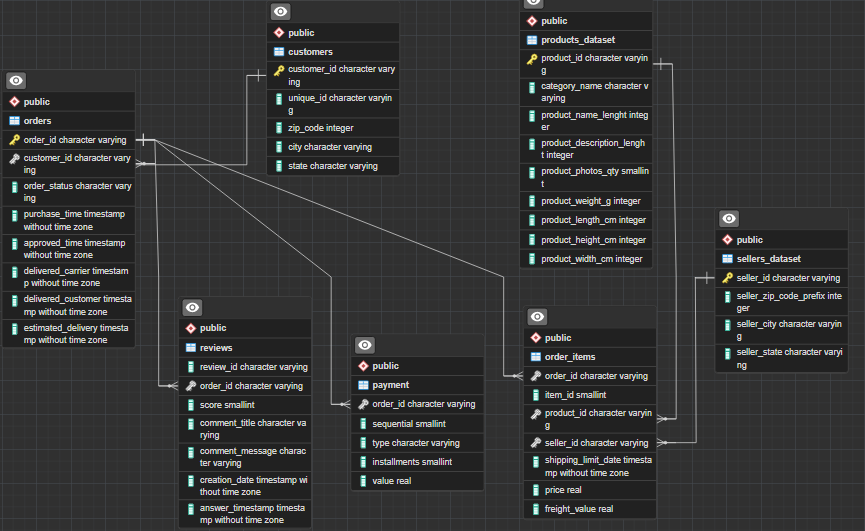
### SQL PROJE

Verilen linkteki veri setinden veri tabanını oluşturarak ERD oluşturunuz ve görsel olarak ekleyiniz. Geolocation tablosu projede kullanılmayacaktır.

Link : [Brazilian E-Commerce Public Dataset by Olist | Kaggle](https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce)

Aşağıda örnek olarak travel veri tabanının ERD görseli eklenmiştir.



#### **Örnek Case Çözümü**

-Aylık bazda toplam ödeme tutarı analizi yapıp excelde görselleştiriniz. Datayı yorumlayınız.

**SQL Query:**

**SELECT**

**(date\_trunc('month',o.order\_approved\_at))::date AS payment\_month,**

**(sum(payment\_value))::integer AS total\_payment\_value**

**FROM payments AS p**

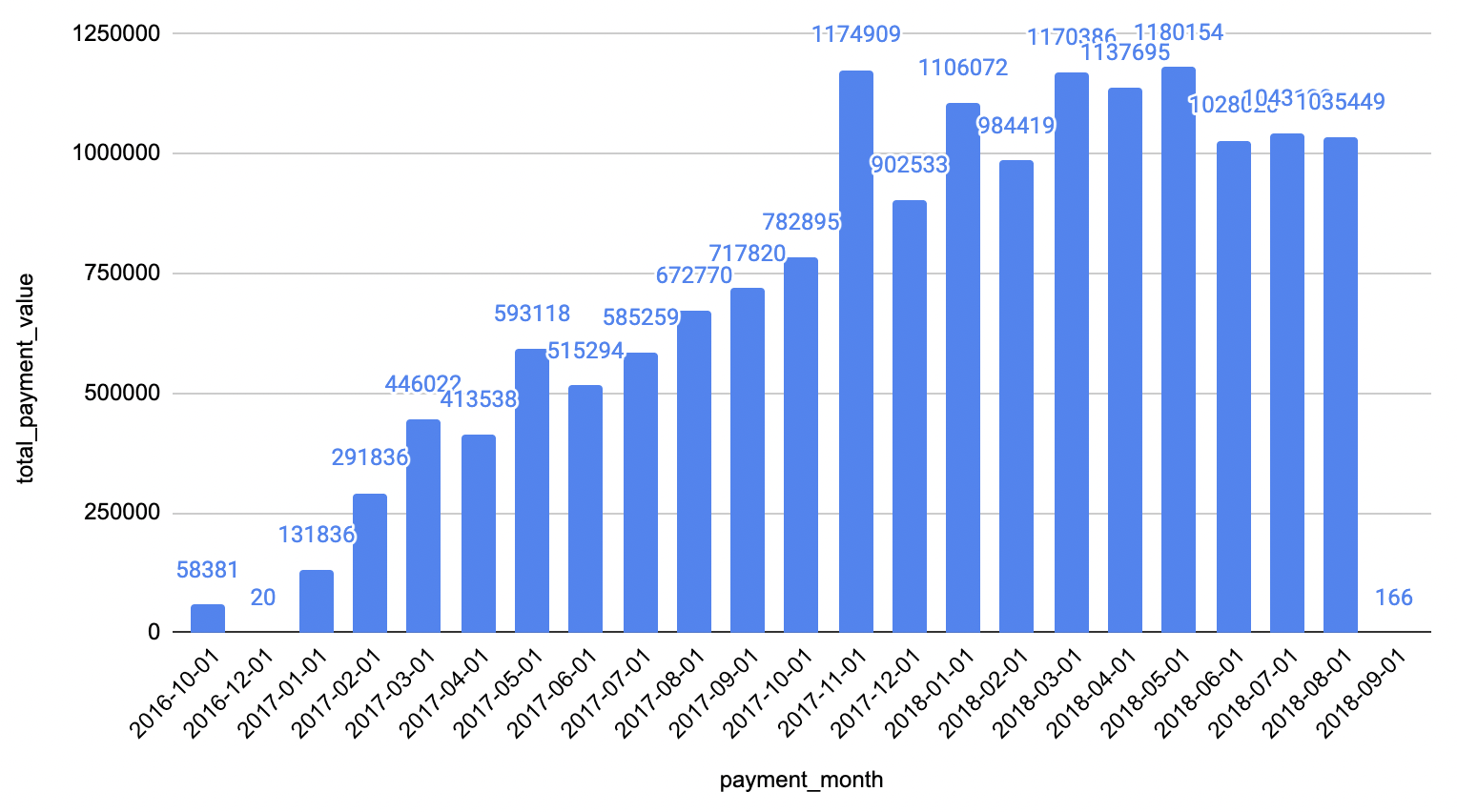
**LEFT JOIN orders AS o**

**ON o.order\_id = p.order\_id**

**WHERE o.order\_approved\_at IS NOT NULL**

**GROUP BY 1**

**ORDER BY payment\_month**



Çıktıyı incelediğimizde order sayılarında aylar içinde düzenli bir artış görüyoruz. Fakat Kasım 2017’de dramatik bir artış söz konusu, bunun sebebi ilgili ayda yapılan bir kampanya,özel bir gün (örneğin Türkiye’de bayram dönemlerinde maaşlara ek promosyon yattığında insanların daha fazla alışveriş yapması gibi) ya da tüm dünyada Kasım ayında uygulanan black friday etkili olmuş olabilir.

2018 yılında da Ocak ayında bir artış söz konusu, bunun sebebi yeni yılda yapılan maaş artışı ya da noel olabilir.

Elimizdeki veri ile yıl bazlı mevsim değerlendirmesi yapmak zor çünkü 2016 yılına ait veri çok az, 2017’de sezondan bağımsız bir artış görünüyor, 2018 yılında ise kış ve bahar ayları hemen hemen aynı verimlilikte geçmiş gibi görünüyor. Sezon incelemesi yapmak için verimiz pek yeterli değil.

#### 

#### 

#### 

#### 

#### **Case 1 : Sipariş Analizi**

##### Question 1 :

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

**SELECT**

**EXTRACT(MONTH FROM order\_approved\_at) AS order\_month,**

**COUNT(order\_id) AS order\_count**

**FROM**

**olist\_orders\_dataset**

**GROUP BY**

**order\_month**

**ORDER BY**

**order\_month;**

##### 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.

SELECT

EXTRACT(MONTH FROM order\_purchase\_timestamp) AS order\_month,

order\_status,

COUNT(order\_id) AS order\_count

FROM

olist\_orders\_dataset

GROUP BY

order\_month, order\_status

ORDER BY

order\_month, order\_status;

##### Question 3 :

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

WITH OrderCategories AS (

SELECT

EXTRACT(MONTH FROM orders.order\_purchase\_timestamp) AS order\_month,

product.product\_category\_name AS category\_name,

COUNT(order\_items.order\_item\_id) AS order\_count

FROM

olist\_orders\_dataset AS orders

INNER JOIN

olist\_order\_items\_dataset AS order\_items

ON

orders.order\_id = order\_items.order\_id

INNER JOIN

olist\_products\_dataset AS product

ON

order\_items.product\_id = product.product\_id

GROUP BY

order\_month, category\_name

)

SELECT

order\_month,

category\_name,

MAX(order\_count) AS max\_order\_count

FROM

OrderCategories

GROUP BY

order\_month, category\_name

ORDER BY

order\_month, max\_order\_count DESC;

##### 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.

**SELECT**

**EXTRACT(DOW FROM order\_purchase\_timestamp) AS day\_of\_week,**

**COUNT(order\_id) AS order\_count**

**FROM**

**olist\_orders\_dataset**

**GROUP BY**

**day\_of\_week**

**ORDER BY**

**day\_of\_week;**

#### **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.**

WITH CustomerCityOrderCounts AS (

SELECT

customer.customer\_id,

customer.customer\_city,

COUNT(orders.order\_id) AS order\_count

FROM

olist\_customers\_dataset AS customer

INNER JOIN

olist\_orders\_dataset AS orders

ON

customer.customer\_id = orders.customer\_id

GROUP BY

customer.customer\_id, customer.customer\_city

),

MaxOrderCity AS (

SELECT

customer\_id,

customer\_city,

MAX(order\_count) AS max\_order\_count

FROM

CustomerCityOrderCounts

GROUP BY

customer\_id, customer\_city

)

SELECT

customer.customer\_id,

MaxOrderCity.customer\_city AS most\_order\_city,

MaxOrderCity.max\_order\_count AS order\_count\_in\_most\_city

FROM

CustomerCityOrderCounts AS customer

INNER JOIN

MaxOrderCity

ON

customer.customer\_id = MaxOrderCity.customer\_id

WHERE

customer.order\_count = MaxOrderCity.max\_order\_count;

#### **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.

WITH DeliveryTimes AS (

SELECT

orders.order\_id,

orders.order\_purchase\_timestamp,

orders.order\_delivered\_customer\_date,

seller.seller\_id,

customer.customer\_unique\_id

FROM

olist\_orders\_dataset AS orders

INNER JOIN

olist\_order\_items\_dataset AS order\_items

ON

orders.order\_id = order\_items.order\_id

INNER JOIN

olist\_sellers\_dataset AS seller

ON

order\_items.seller\_id = seller.seller\_id

INNER JOIN

olist\_customers\_dataset AS customer

ON

orders.customer\_id = customer.customer\_id

),

DeliveryDurations AS (

SELECT

order\_id,

seller\_id,

customer\_unique\_id,

EXTRACT(EPOCH FROM (order\_delivered\_customer\_date - order\_purchase\_timestamp)) AS delivery\_duration

FROM

DeliveryTimes

)

SELECT

seller\_id,

AVG(delivery\_duration) AS average\_delivery\_duration

FROM

DeliveryDurations

GROUP BY

seller\_id

ORDER BY

average\_delivery\_duration

LIMIT 5;

##### Question 2 :

-Hangi satıcılar daha fazla kategoriye ait ürün satışı yapmaktadır?

Fazla kategoriye sahip satıcıların order sayıları da fazla mı?

WITH SellerCategoryCounts AS (

SELECT

seller.seller\_id,

COUNT(DISTINCT product.product\_category\_name) AS category\_count

FROM

olist\_sellers\_dataset AS seller

INNER JOIN

olist\_order\_items\_dataset AS order\_items

ON

seller.seller\_id = order\_items.seller\_id

INNER JOIN

olist\_orders\_dataset AS orders

ON

order\_items.order\_id = orders.order\_id

INNER JOIN

olist\_products\_dataset AS product

ON

order\_items.product\_id = product.product\_id

GROUP BY

seller.seller\_id

)

SELECT

seller.seller\_id,

sellercategorycounts.category\_count,

COUNT(orders.order\_id) AS order\_count

FROM

olist\_sellers\_dataset AS seller

LEFT JOIN

SellerCategoryCounts AS sellercategorycounts

ON

seller.seller\_id = sellercategorycounts.seller\_id

LEFT JOIN

olist\_order\_items\_dataset AS order\_items

ON

seller.seller\_id = order\_items.seller\_id

LEFT JOIN

olist\_orders\_dataset AS orders

ON

order\_items.order\_id = orders.order\_id

GROUP BY

seller.seller\_id, sellercategorycounts.category\_count

ORDER BY

sellercategorycounts.category\_count DESC;

#### **Case 4 : Payment Analizi**

##### Question 1 :

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

WITH TaksitSayilari AS (

SELECT

customer.customer\_zip\_code\_prefix,

payment.payment\_installments

FROM

olist\_order\_payments\_dataset AS payment

INNER JOIN

olist\_orders\_dataset AS orders

ON

payment.order\_id = orders.order\_id

INNER JOIN

olist\_customers\_dataset AS customer

ON

orders.customer\_id = customer.customer\_id

)

SELECT

customer\_zip\_code\_prefix,

AVG(payment\_installments) AS avg\_installments

FROM

TaksitSayilari

GROUP BY

customer\_zip\_code\_prefix

ORDER BY

avg\_installments DESC

LIMIT 1;

##### Question 2 :

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

**SELECT**

**payment.payment\_type AS payment\_type,**

**COUNT(orders.order\_id) AS successful\_order\_count,**

**SUM(payment.payment\_value) AS total\_successful\_payment\_amount**

**FROM**

**olist\_order\_payments\_dataset AS payment**

**INNER JOIN**

**olist\_orders\_dataset AS orders**

**ON**

**payment.order\_id = orders.order\_id**

**WHERE**

**orders.order\_status = 'delivered'**

**GROUP BY**

**payment.payment\_type**

**ORDER BY**

**successful\_order\_count DESC;**

##### Question 3 :

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

WITH OrderPaymentDetails AS (

SELECT

items.order\_id,

items.product\_id,

items.order\_item\_id,

items.price,

items.shipping\_limit\_date,

payment.payment\_type

FROM

olist\_order\_items\_dataset AS items

INNER JOIN

olist\_order\_payments\_dataset AS payment

ON

items.order\_id = payment.order\_id

WHERE

payment.payment\_type IN ('credit\_card', 'boleto')

)

SELECT

product.product\_category\_name AS product\_category,

payment\_details.payment\_type,

COUNT(\*) AS order\_count

FROM

OrderPaymentDetails AS payment\_details

INNER JOIN

olist\_products\_dataset AS product

ON

payment\_details.product\_id = product.product\_id

GROUP BY

product.product\_category\_name,

payment\_details.payment\_type

HAVING

COUNT(\*) > 0

ORDER BY

product.product\_category\_name,

order\_count DESC;

#### **Case 5 : RFM Analizi**

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

Veri seti bu linkten alınmıştır, veriyi tanımak için linke girip inceleyebilirsiniz.  
  
[E-Commerce Data](https://www.kaggle.com/datasets/carrie1/ecommerce-data)