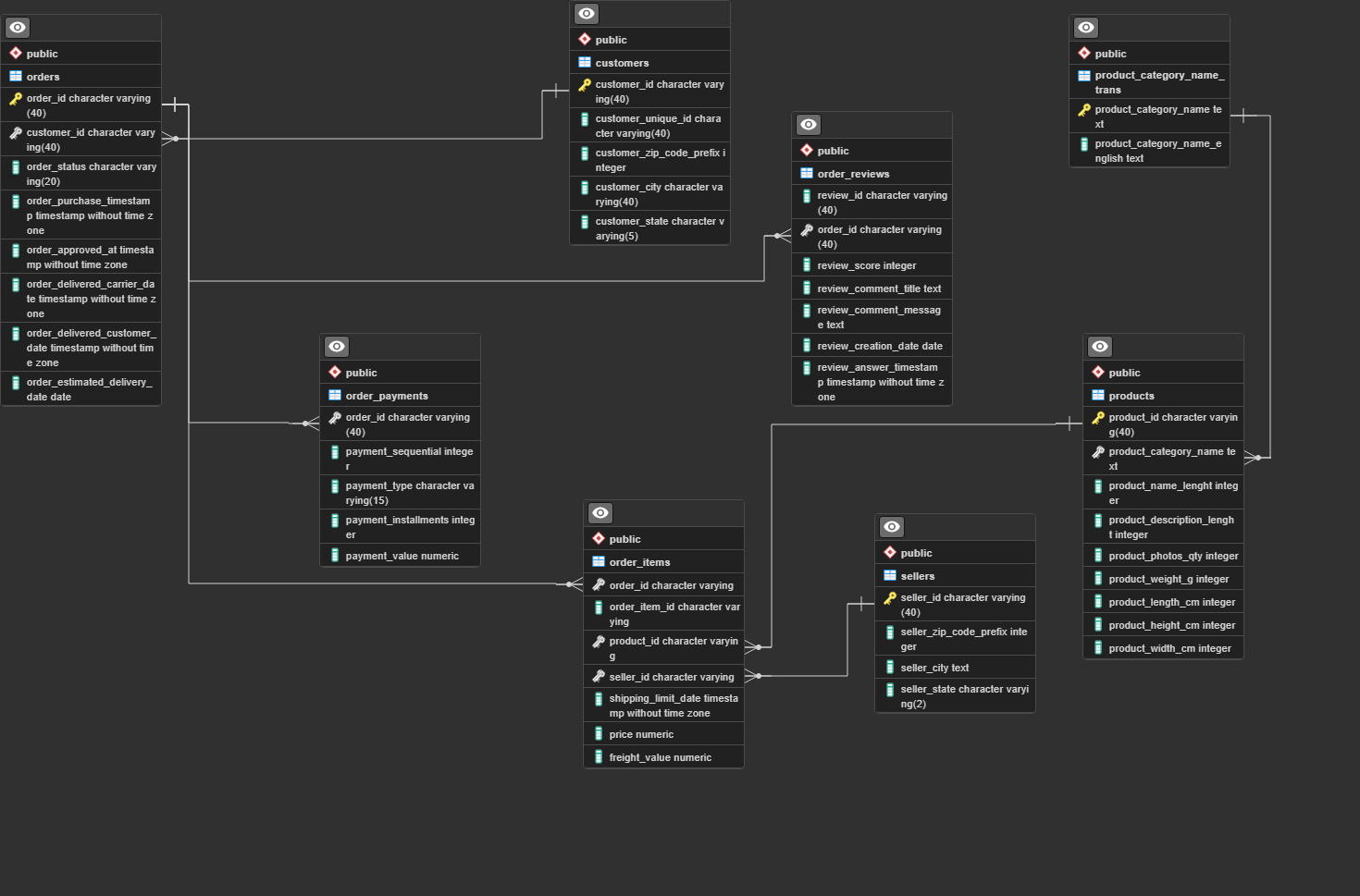
### SQL PROJE

Aşağıda Brazilian E-Commerce Public veri tabanının ERD görseli eklenmiştir.



Bu diagramda verilen tablolar arasında olabileceğini düşündüğüm bağlantıları hazırladım.

Örneğin orders tablosunda bulunan order\_id tekrar edilmediği için bu tabloda primary key olarak seçtim. Order\_items tablosunda bulunan order\_id bilgiside tekrar eden bilgileri tuttuğu için ERD diagramında orders ve order\_items tablolarını bağlarken one to many olacak şekilde orders tablosundaki order\_id ile order\_items tablosundaki order\_id bilgilerini şekildede görebileceğiniz gibi tablo ilişkilerini oluşturdum.

#### **Case 1 : Sipariş Analizi**

##### Question 1 :

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

SQL QUERY:

SELECT

COUNT (dıstınct orders.order\_id) siparis\_adet,

date\_trunc('month',orders.order\_approved\_at)::date AS tarih

FROM orders

WHERE orders.order\_approved\_at IS NOT null

GROUP BY tarih

ORDER BY tarih

Bunun için hazırladığım sorguda ay bilgisini kullanmak için date\_trunc kullandım, buradan gelen bilgide saat kısımları 00:00:00 olarak döndüğü için son olarak ::date kullanarak daha düzgün bir bilgi elde ettim.Çıkan veride; verilen siparişler incelendiğinde 2016 yılına ait çok fazla sipariş görünmemekte, takip eden yllarda sipariş sayıları giderek artmakta ve en çok siparişin alındığı yıl 2018’de toplam 53986 adet sipariş bulunmaktadır. 2017 yılıda 44973 ile en iyi 2.yıldır. 2016 yılında sadece 3 aylık veri bulunduğu için siparis adetleri 2017 ve 2018 e kıyasla çok azdır.

##### 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

date\_trunc('month',orders.order\_approved\_at)::date AS tarih,

orders.order\_status AS siparis\_durum,

COUNT (\*) AS siparis\_adet

FROM orders

WHERE orders.order\_status IS NOT null AND

orders.order\_approved\_at IS NOT null

GROUP BY tarih,siparis\_durum

ORDER BY tarih,siparis\_durum

Excel tablosunda order\_status te “delivered” için filtreleme yaptım ve teslim edilen siparişlerin ay bazlı durumuna baktığımızda 2018 yılının ilk 8 ayını 2017 deki verilerle karşılaştırdığımızda yüksek oradan bir artış olmuştur. Sipariş sayısı artışı içinde online alışveriş sitesinin tanınması , yapmış olduğu bir kampanya neden olmuş olabilir.Dramatik yükseliş ve artışın olduğu tarihi

2017 ve 2018 ocak ayında gözlemleyebiliriz.2016 yılındaki siparişleri veri az olduğu için değerlendirmeye almadım.

##### Question 3 :

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

SQL QUERY:

-Ürün kategorisi kırılımındaki sipariş sayıları için oluşturuduğum query,

SELECT product\_category\_name,

COUNT(\*)

FROM orders

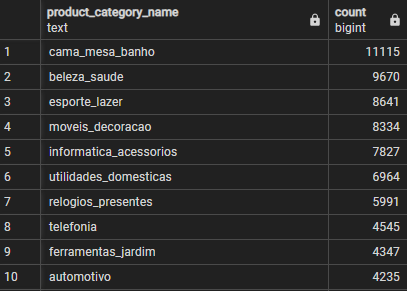
LEFT JOIN order\_items ON orders.order\_id = order\_items.order\_id

LEFT JOIN products ON order\_items.product\_id = products.product\_id

WHERE product\_category\_name IS NOT NULL

GROUP BY 1

ORDER BY 2 DESC



-Özel gün olarak Brezilya’da 12 Haziran tarihinde kutlanan sevgililer günü için yaptığım query,

SELECT

products.product\_category\_name,

COUNT (dıstınct orders.order\_id) adet

FROM orders

LEFT JOIN order\_items ON orders.order\_id = order\_items.order\_id

LEFT JOIN products ON order\_items.product\_id = products.product\_id

WHERE orders.order\_status = 'delivered' AND

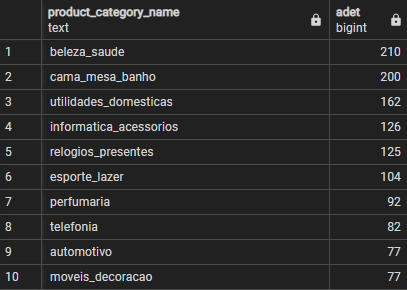
products.product\_category\_name IS NOT null AND

EXTRACT( DAY FROM orders.order\_approved\_at::date) BETWEEN 7 AND 12 AND

EXTRACT( MONTH FROM orders.order\_approved\_at::date)=6

GROUP BY products.product\_category\_name

ORDER BY adet DESC



-Özel gün olarak Brezilya’da 24 Aralık tarihinde kutlanan Noel için yaptığım query,

SELECT

products.product\_category\_name,

COUNT (distinct orders.order\_id) adet

FROM orders

LEFT JOIN order\_items ON orders.order\_id = order\_items.order\_id

LEFT JOIN products ON order\_items.product\_id = products.product\_id

WHERE orders.order\_status = 'delivered' AND

products.product\_category\_name IS NOT NULL AND

EXTRACT ( DAY FROM orders.order\_approved\_at::date) BETWEEN 20 AND 30 AND

EXTRACT ( MONTH FROM orders.order\_approved\_at::date)=12

GROUP BY products.product\_category\_name

ORDER BY adet 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.

SQL QUERY:

-Haftanın günleri bazında order sayılarının incelenmesi

SELECT

TO\_CHAR(order\_approved\_at::date, 'Day') hafta\_gün,

COUNT(order\_id) satis\_adet

FROM orders

WHERE order\_approved\_at IS NOT NULL

GROUP BY hafta\_gün

ORDER BY satis\_adet desc

- En çok satış adetinin olduğu gün Salı , en az satış yapılan gün Pazar günüdür.

-Ay günleri bazında order sayılarının incelenmesi

SELECT

TO\_CHAR(order\_approved\_at::date, 'DD') ay\_gün,

COUNT(order\_id) satis\_adet

FROM orders

WHERE order\_approved\_at IS NOT NULL

GROUP BY ay\_gün

ORDER BY satis\_adet desc

---Ayın günleri bazındada ayların 24’ ü ve 5’I en çok siparis verilen gündür.

#### **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.

SQL QUERY:

SELECT

COUNT(distinct orders.order\_id) AS siparis\_adet,

COUNT(distinct customers.customer\_unique\_id) AS musteri\_sayisi,

customers.customer\_city AS musteri\_sehir

FROM orders

LEFT JOIN customers ON orders.customer\_id = customers.customer\_id

GROUP BY musteri\_sehir

ORDER BY musteri\_sayisi DESC, siparis\_adet DESC

LIMIT 10

Sorgumu hazırlarken orders ve customers tablolarını joinledim. Customers tablosundaki customer\_unique\_id kullanma sebebim orders tablosundaki customer\_idler her yeni orderda değişiyor fakat aynı customer\_unique\_id ye ait olduğu için gerçek müşteri sayısını bulmak için distinct kullanarak customer\_unique\_id yi count ile saydırdım.

Sao Paulo şehrinde müşteri ve sipariş adetleri daha fazladır. Görselde top 10 u görüyoruz.

#### 

#### **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:

WITH hizli\_satici AS(

SELECT

COUNT(distinct orders.order\_id) AS siparis\_adet,

order\_items.seller\_id AS satici,

AVG(age(orders.order\_delivered\_customer\_date::DATE,orders.order\_purchase\_timestamp::DATE)) AS toplam\_teslim\_suresi

FROM orders

LEFT JOIN order\_items ON orders.order\_id = order\_items.order\_id

LEFT JOIN order\_reviews ON orders.order\_id = order\_reviews.order\_id

WHERE orders.order\_status ='delivered' AND order\_reviews.review\_score IS NOT null

GROUP BY satici

ORDER BY siparis\_adet DESC, toplam\_teslim\_suresi ASC, satici ASC

LIMIT 5

),

satici\_puan AS (

SELECT

COUNT(distinct orders.order\_id) AS siprais\_adet,

order\_items.seller\_id as satici,

SUM(order\_reviews.review\_score) AS puan

FROM orders

LEFT JOIN order\_items

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

LEFT JOIN order\_reviews ON orders.order\_id = order\_reviews.order\_id

WHERE orders.order\_status='delivered' AND order\_reviews.review\_score IS NOT null

GROUP BY satici

ORDER BY siprais\_adet DESC, puan DESC

LIMIT 5

)

SELECT

hizli\_satici.siparis\_adet,

hizli\_satici.satici,

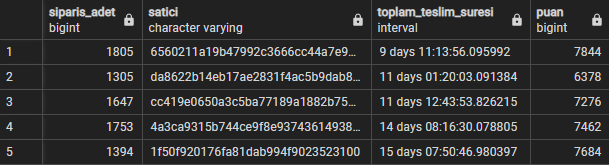
hizli\_satici.toplam\_teslim\_suresi,

satici\_puan.puan

FROM hizli\_satici

LEFT JOIN satici\_puan ON hizli\_satici.satici = satici\_puan.satici

ORDER BY toplam\_teslim\_suresi ASC, puan DESC,siparis\_adet DESC



Bu soru için önce with kullanarak en çok sipariş en kısa sürede ulaştıran ve siparis durumu delivered olan ilk 5 saticiyi tespit ediyorum. Sonrasında 2. with için ilk with te kullandığım sorgunun aynısını kullanarak bu sefer teslim edilmiş siparişlerdeki verilen review\_score alanını toplayarak satıcıya müşteriler tarafından verilen score bilgilerini topluyorum.

Sonrasında 2 tabloyu birleştirerek sipariş adet satıcı, teslim süresi ve toplam bir puan elde ediyorum.Sipariş adedi fazla olan kısa sürede teslim eden ve en yüksek scora sahip satici 1. sıradadır. Sonuçtan farkedileceği gibi sipariş adedi fazla olan ama daha düşük skora sahip satıcılar vardır. Sipariş adedi daha az olan teslimat süresi daha fazla olan ama puanı daha çok olan satici 4. sıradadır.

##### Question 2 :

-Hangi satıcılar daha fazla kategoriye ait ürün satışı yapmaktadır?

Fazla kategoriye sahip satıcıların order sayıları da fazla mı?

SQL QUERY:

SELECT

COUNT(distinct orders.order\_id) AS siparis\_adet,

order\_items.seller\_id as satici,

COUNT(distinct products.product\_category\_name) AS kat\_adet

FROM orders

LEFT JOIN order\_items ON orders.order\_id = order\_items.order\_id

LEFT JOIN products ON order\_items.product\_id = products.product\_id

GROUP BY satici

HAVING COUNT (distinct products.product\_category\_name)>=10

ORDER BY siparis\_adet DESC

A screenshot of a computer

Description automatically generated Having count kullanma sebebim kategori adeti 10 ve 10 dan fazla olan satıcı ve sipariş adetlerini alabilmek için yaptım.

Hazırladığım sorguda çıkan sonuç; en çok satışı yapan satıcı 23 adet kategori ile yapmıştır. Fakat sonraki satıcılar incelendiğinde daha az kategori ile daha fazla satış yapanlar olduğu gözlemlenmektedir. Her zaman kategorisi fazla olan daha çok satış yapar denemez.

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

SQL QUERY:

##### Question 1 :

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

WITH mtb AS(

SELECT

COUNT(distinct customers.customer\_unique\_id) AS musteri,

customers.customer\_state AS musteri\_bolge

op.payment\_installments AS taksit\_sayisi

FROM orders ord

LEFT JOIN customers ON ord.customer\_id = customers.customer\_id

LEFT JOIN order\_payments op ON ord.order\_id = op.order\_id

GROUP BY taksit\_sayisi,musteri\_bolge

ORDER BY taksit\_sayisi DESC

)

SELECT

SUM (mtb.musteri) AS musteri\_sayisi,

mtb.musteri\_bolge AS musteri\_bolgesi,

mtb.taksit\_sayisi AS taksit\_bilgisi

FROM mtb

GROUP BY musteri\_bolgesi,taksit\_bilgisi

ORDER BY taksit\_sayisi DESC, musteri\_sayisi DESC

Taksit sayısı 24 ile en fazla olan bölgeler sp,rj,rs,mg,df,ce,ro,go,pb,pr,ba dır. Fakat siparis adetleri çok düşüktür.Bu bölgeler aynı zamanda az taksit ile fazla sipariş oluşturmaktadır. Aşağıdaki tabloda da görüntülenmektedir. SP bölgesi 21508 tek çekim alış veriş yapmıştır.

Müşteriler daha çok taksit tutarını daha az tutmaktadır.

A screenshot of a computer

Description automatically generated

A screenshot of a black screen

Description automatically generated

##### 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

COUNT(distinct op.order\_id) AS siparis\_adet,

op.payment\_type as odeme\_tipi,

SUM (op.payment\_value ::money) AS toplam\_tutar

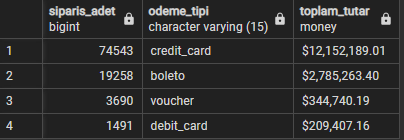
FROM order\_payments op

LEFT JOIN orders o ON op.order\_id = o.order\_id

WHERE o.order\_status='delivered' OR o.order\_status='invoiced'

GROUP BY odeme\_tipi

ORDER BY siparis\_adet DESC



Verilen datalar içerisinde en çok Kredi kartı ile ödeme yapılmış en az da debit card kullanılmıştır.

##### Question 3 :

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

-En çok taksit uygulanan kategoriler (24-15 taksit arasında)

SELECT

order\_payments.payment\_installments AS taksit\_adet,

products.product\_category\_name AS urun\_kategori

FROM orders

LEFT JOIN order\_payments ON orders.order\_id = order\_payments.order\_id

LEFT JOIN order\_items ON orders.order\_id = order\_items.order\_id

LEFT JOIN products ON order\_items.product\_id = products.product\_id

WHERE products.product\_category\_name IS NOT null AND

order\_payments.payment\_installments IS NOT null AND order\_payments.payment\_installments <=24 AND order\_payments.payment\_installments >=15

GROUP BY order\_payments.payment\_installments,products.product\_category\_name

A black screen with white text

Description automatically generated

En çok auto urunu , sağlık ve güzellik ürünleri ,ev konfor ürünleri gibi ürünlere fazla taksit yapıldığı görülmektedir.

-Sipariş adedi en fazla olan ve tek çekim olan ürün kategorileri

SELECT

COUNT (distinct orders.order\_id) AS adet,

order\_payments.payment\_installments AS taksit\_adet,

products.product\_category\_name AS urun\_kategori

FROM orders

LEFT JOIN order\_payments ON orders.order\_id = order\_payments.order\_id

LEFT JOIN order\_items ON orders.order\_id = order\_items.order\_id

LEFT JOIN products ON order\_items.product\_id = products.product\_id

WHERE products.product\_category\_name IS NOT null AND

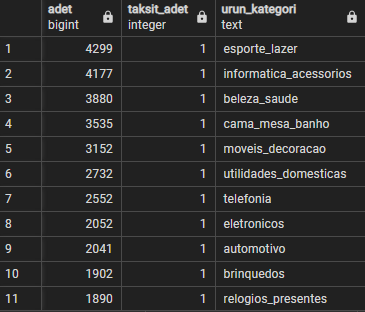
order\_payments.payment\_installments IS NOT null

and order\_payments.payment\_installments=1

GROUP BY order\_payments.payment\_installments,products.product\_category\_name

order by adet desc

limit 12



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

SQL QUERY:

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

WITH recency AS (

SELECT

customer\_id AS musteri,

EXTRACT(DAY FROM '2011-12-10 17:19:00'::DATE-MAX(invoice\_date)) AS recency

FROM tbl\_rfm

WHERE invoice\_date IS NOT null AND customer\_id IS NOT null AND unit\_price IS NOT null AND quantity IS NOT null

GROUP BY customer\_id

),

frequency AS (

SELECT

customer\_id AS musteri,

SUM ( quantity) AS frequency

--extract(DAY from '2011-12-10 17:19:00'::date-max(invoice\_date)) as recency

FROM tbl\_rfm

WHERE invoice\_date IS NOT null AND customer\_id IS NOT null AND unit\_price IS NOT null AND quantity IS NOT null

GROUP BY customer\_id

),

monetary AS(

SELECT

customer\_id AS musteri,

SUM ( quantity\*unit\_price) AS monetary

FROM tbl\_rfm

WHERE invoice\_date IS NOT null AND customer\_id IS NOT null AND unit\_price IS NOT null AND quantity IS NOT null

GROUP BY customer\_id

)

SELECT

recency.musteri,

recency.recency,

frequency.frequency,

monetary.monetary::numeric::money

FROM recency

JOIN frequency ON recency.musteri = frequency.musteri

JOIN monetary ON recency.musteri = monetary.muster

 Recency hesaplamasında son fatura tarihini baz alarak müşterilerin recency hesaplamasını yaptım. Frequency de aynı sorguyu kullanarak müşterilerin sipariş adetlerini hesapladım . Monetary olarakta yine aynı sorgumu kullanarak adet\*birim fiyat (unit\_price) hesaplamasıyla monetaryi buldum. Sonrasında with kullanarak 3 farklı CTE hazırladım. Sonrasında istenilen bilgilerin oluşacağı şekilde joinlerin kullanarak dataları birleştirdim. Görseli aşağıdaki gibidir