### **SQL PROJE**

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

Aşağıda brasillianecommerce veri tabanının ERD görseli eklenmiştir.

#### 

#### **Case 1 : Sipariş Analizi**

##### Question 1 :

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

SQL Query:

select

extract(year from order\_approved\_at) as year ,

extract(month from order\_approved\_at) as month ,

count(order\_id) as order\_count

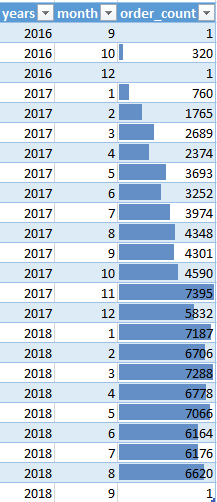
from orders

where order\_approved\_at is not null

group by year, month

order by year, month;

Tablo:



##### 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

to\_char(order\_approved\_at, 'yyyy-mm') as order\_month,

order\_status,

count(order\_id) as total\_orders

from orders

where order\_approved\_at ıs not null

group by order\_month, order\_status

order by order\_month, order\_status

Delivered grafiğini incelediğimizde 2017 yılının Kasım ayında ani bir sıçrama görünmektedir bunun sebebi kasım ayında uygulanan ‘’Kasım ayı indirimleri’’ olabilir.2017 Aralık ayında kritik sayılabilecek bir düşüş görünüyor. Buna Kasım ayındaki artıştan kaynaklı müşteri ihtiyaçlarının azalması sebep olmuş olabilir.

Canceled grafiğini incelediğimizde 2018 Şubat ayında normalin üstünde bir veri görüyoruz. Müşteri kaybı yaşanmaması için bu durumun sebeplerinin araştırılıp çözüm üretilmesi konusunda aksiyon alınabilir.

##### Question 3 :

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

WITH SPECIAL\_DAYS AS (

SELECT 'new\_year' AS special\_day, DATE '2016-01-01' AS day

UNION ALL

SELECT 'valentines\_day' AS special\_day, DATE '2016-02-14' AS day

UNION ALL

SELECT 'mothers\_day' AS special\_day, DATE '2016-05-08' AS day

UNION ALL

SELECT 'new\_year' AS special\_day, DATE '2017-01-01' AS day

UNION ALL

SELECT 'valentines\_day' AS special\_day, DATE '2017-02-14' AS day

UNION ALL

SELECT 'mothers\_day' AS special\_day, DATE '2017-05-14' AS day

UNION ALL

SELECT 'new\_year' AS special\_day, DATE '2018-01-01' AS day

UNION ALL

SELECT 'valentines\_day' AS special\_day, DATE '2018-02-14' AS day

UNION ALL

SELECT 'mothers\_day' AS special\_day, DATE '2018-05-13' AS day

),

ORDERS\_WITH\_SPECIAL\_DAYS AS (

SELECT o.order\_id,

o.customer\_id,

o.order\_purchase\_timestamp::DATE AS purchase\_date,

sd.special\_day

FROM orders o

JOIN SPECIAL\_DAYS sd ON o.order\_purchase\_timestamp::DATE = sd.day

),

ORDER\_ITEMS\_WITH\_SPECIAL\_DAYS AS (

SELECT ow.order\_id,

p.product\_id,

t.category\_name\_english AS product\_category\_name

FROM order\_items ow

JOIN products p ON ow.product\_id = p.product\_id

JOIN translation t ON p.product\_category\_name = t.category\_name

)

SELECT osd.special\_day,

oi.product\_category\_name,

COUNT(\*) AS order\_count

FROM ORDERS\_WITH\_SPECIAL\_DAYS osd

JOIN ORDER\_ITEMS\_WITH\_SPECIAL\_DAYS oi ON osd.order\_id = oi.order\_id

GROUP BY osd.special\_day, oi.product\_category\_name

ORDER BY osd.special\_day, order\_count DESC

Mother’s day:

1. bed\_bath\_table order\_count 36

2. watches\_gifts order\_count 33

3. sports\_leisure order\_count 32

Valentine’s day:

1. health\_beauty order\_count 46

2. bed\_bath\_table order\_count 42

3. furniture\_decor order\_count 40

New year

1. telephony order\_count 10

2. watches\_gifts order\_count 10

3. health\_beauty order\_count 9

Anneler Günü’nde en çok satılan kategoriler sırasıyla ev tekstili,hediyelik saat ve spor-hobi eşyaları kategorisidir. Anneler Günü için klasik hediye kategorilerindendir. Bu durumda beklenmedik bir satış öyküsü yoktur.

Sevgililer Günü’nde en çok satılan kategoriler sırasıyla sağlık-kozmetik, ev tekstili, ev dekorasyonu kategorileridir.Sevgililer Günü için dünyanın her yerinde tercih edilen kategorilerdir. Fakat burda beni şaşırtan durum aksesuar, giyim, pırlanta vs. Gibi ürünleri kapsayan bir kategori bulunmamıştır. Bu durum ülkenin gelir seviyesi ile alakalı olmakla birlikte gerekli araştırmalar yapılıp konu üzerine aksiyon alınabilir.

Yeni Yıl’da en çok satılan kategoriler sırasıyla telefon, hediyelik saat, sağlık-kozmetik kategorisidir. Bu durumun ortaya çıkmasında telefon ürünlerinde yapılan yıl sonu indirim ve kampanyaları etkili olmuş olabilir. Bu satış listesinde ev dekorasyon kategorisini görmemek (yeni yıl ağacı, ışıklandırma vs.) şaşırtıcı bir sonuçtur. Bu sonuç bu tarz ürünlerin hali hazırda satın alındığını gösterebilir.

##### 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.

Haftanın günleri;

select

to\_char(order\_approved\_at,'day') as day\_of\_week,

count(distinct order\_id) as order\_count

from orders

where order\_approved\_at is not null

group by 1

order by 2

Outputu incelediğimizde en fazla siparişin Salı günü verildiğini, en az siparişin ise Pazar günü verildiğini görüyoruz.Tatil günlerinde insanların e ticaret sitelerinden alışveriş yapma eğilimi olmayabilir.

Ayın günleri;

select

date\_part('day',order\_approved\_at) as day\_of\_month,

count(distinct order\_id) as order\_count

from orders

where order\_approved\_at is not null

group by 1

order by 1

Aylık siparişleri incelediğimizde en fazla siparişin ayın 24’ünde en az siparişin 31’inde verildiğini görüyoruz. İnsanların alışveriş yapma eğilimlerini de göz önüne alarak bu günlere özel kampanyalar yaratılıp daha fazla sipariş alınabilir. Aynı şey az sipariş verilen günlere de uygulanabilir böylece daha stabil bir satış grafiği elde edebilirz.

#### **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.**

Sql query;

with order\_counts as (

select o.customer\_id,

customer\_city,

count (order\_id) as order\_count

from orders as o

left join customers as c

on c.customer\_id = o.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\_city,

count (o.order\_id)

from orders as o

left join customer\_city as cc

on o.customer\_id = cc.customer\_id

group by 1

order by 2 desc

en çok sipariş veren şehirler;



#### **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 shortestsellerfullfilmenttime AS (

SELECT

s.seller\_id,

AVG(o.order\_delivered\_customer\_date - o.order\_purchase\_timestamp) AS average\_fulfillment\_days

FROM

orders o

JOIN order\_items oi ON o.order\_id = oi.order\_id

JOIN sellers s ON oi.seller\_id = s.seller\_id

WHERE

o.order\_delivered\_customer\_date IS NOT NULL

AND o.order\_purchase\_timestamp IS NOT NULL

GROUP BY

s.seller\_id

HAVING

COUNT(DISTINCT o.order\_id) > 33

ORDER BY

average\_fulfillment\_days ASC

LIMIT 5

)

SELECT

ssft.seller\_id,

ssft.average\_fulfillment\_days,

COUNT(DISTINCT o.order\_id) AS total\_orders,

COUNT(r.review\_id) AS total\_reviews,

AVG(r.review\_score) AS average\_review\_score

FROM

shortestsellerfullfilmenttime ssft

JOIN order\_items oi ON ssft.seller\_id = oi.seller\_id

JOIN orders o ON oi.order\_id = o.order\_id

LEFT JOIN reviews r ON oi.order\_id = r.order\_id

GROUP BY

ssft.seller\_id, ssft.average\_fulfillment\_days;

Outputu incelediğimizde ürünleri en hızlı şekilde teslim eden ilk 5 satıcıyı görüyoruz. Seller id’si ‘’d13’’ ile başlayan satıcı en fazla ürünü satıp en iyi puanlamayı alan satıcı olarak görünüyor. Bu konuda yaptığı çalışmalar araştırılıp diğer satıcılara öneri olarak sunulabilir. Böylece siteden yapılan alışveriş arttırılabilir.

##### Question 2 :

-Hangi satıcılar daha fazla kategoriye ait ürün satışı yapmaktadır?

Fazla kategoriye sahip satıcıların order sayıları da fazla mı?

SQL query:

select

s.seller\_id,

count(distinct p.product\_category\_name) as category\_count,

count(distinct o.order\_id) as order\_count

from sellers s

join order\_items ori on s.seller\_id = ori.seller\_id

join orders o on o.order\_id = ori.order\_id

join products p on p.product\_id = ori.product\_id

where p.product\_category\_name is not null

group by s.seller\_id

order by 2 desc;

|  |  |  |
| --- | --- | --- |
| **seller\_id** | **category\_count** | **order\_count** |
| b2ba3715d723d245138f291a6fe42594 | 27 | 337 |
| 955fee9216a65b617aa5c0531780ce60 | 23 | 1287 |
| 4e922959ae960d389249c378d1c939f5 | 23 | 405 |
| 1da3aeb70d7989d1e6d9b0e887f97c23 | 21 | 265 |
| f8db351d8c4c4c22c6835c19a46f01b0 | 19 | 665 |

|  |  |  |
| --- | --- | --- |
| **seller\_id** | **category\_count** | **order\_count** |
| 6560211a19b47992c3666cc44a7e94c0 | 7 | 1854 |
| 4a3ca9315b744ce9f8e9374361493884 | 7 | 1806 |
| cc419e0650a3c5ba77189a1882b7556a | 4 | 1706 |
| 1f50f920176fa81dab994f9023523100 | 3 | 1404 |
| da8622b14eb17ae2831f4ac5b9dab84a | 4 | 1314 |

Yukarıdaki tabloları incelediğimizde en fazla kategoriye sahip satıcının 337sipariş miktarı varken, alttaki tabloda görüldüğü gibi 7 kategoriye sahip bir satıcının 1854 sipariş miktarı ile en fazla sipariş miktarına sahip olduğunu görüyoruz. Yani ürün kategorisi ile satış miktarı arasında doğrusal bir bağlantı bulunmuyor.

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

##### Question 1 :

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

SQL query:

select c.customer\_state,

count(distinct o.order\_id) as total\_orders

from payments p

join orders o on p.order\_id = o.order\_id

join customers c on o.customer\_id = c.customer\_id

where p.payment\_installments >= 4

group by 1

order by 2 desc;

|  |  |
| --- | --- |
| **customer\_state** | **total\_orders** |
| SP | 10300 |
| RJ | 3904 |
| MG | 3551 |
| RS | 1663 |
| PR | 1432 |

Çıktıyı incelediğimizde en çok taksit kullanarak ödeme yapan ilk 5 bölgeyi görüyoruz. Bu ilk 5 bölge ülkenin gelir seviyesi en düşük yerlerinden biri olabilir. Bu bölgede reklam ve kampanya çalışmaları daha çok uygun fiyatlı ikame ürünler üzerine yapılabilir. Ya da satış arttırmak için taksitlendirme sayısının arttırılması faiz oranı düşürülmesi gibi hedefler koyulabilir.

##### 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

p.payment\_type,

COUNT(DISTINCT o.order\_id) AS successful\_order\_count,

SUM(p.payment\_value)::integer AS total\_successful\_payment\_amount

FROM

payments AS p

JOIN

orders AS o ON p.order\_id = o.order\_id

WHERE

o.order\_status != 'canceled' AND o.order\_status != 'unavailable'

GROUP BY

p.payment\_type

ORDER BY

successful\_order\_count DESC;

|  |  |  |
| --- | --- | --- |
| **payment\_type** | **successful\_order\_count** | **total\_successful\_payment\_amount** |
| credit\_card | 75618 | 12350043 |
| boleto | 19539 | 2826802 |
| voucher | 3745 | 349874 |
| debit\_card | 1515 | 212418 |

##### Question 3 :

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

SQL Query:

Tek Çekim Ödemelerin kategori bazlı analizi

SELECT

p.product\_category\_name,

COUNT(DISTINCT o.order\_id) AS single\_payment\_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

payments AS pay ON o.order\_id = pay.order\_id

WHERE

pay.payment\_installments = 1

GROUP BY

p.product\_category\_name

ORDER BY

single\_payment\_order\_count DESC;

|  |  |
| --- | --- |
| **product\_category\_name** | **single\_payment\_order\_count** |
| esporte\_lazer | 4299 |
| informatica\_acessorios | 4177 |
| beleza\_saude | 3880 |
| cama\_mesa\_banho | 3535 |
| moveis\_decoracao | 3152 |

Taksitle ödenen siparişlerin kategori bazlı analizi

SELECT payment\_installments,

product\_category\_name,

count(distinct o.order\_id) as order\_count

from orders as o

left join payments as p

on p.order\_id= o.order\_id

left join order\_items as oi

on oi.order\_id=o.order\_id

left join products as pr on pr.product\_id=oi.product\_id

where payment\_installments >1

group by 1,2

order by order\_count desc

|  |  |  |
| --- | --- | --- |
| **payment\_installments** | **product\_category\_name** | **order\_count** |
| 2 | beleza\_saude | 1194 |
| 2 | cama\_mesa\_banho | 1085 |
| 3 | beleza\_saude | 1081 |
| 2 | esporte\_lazer | 1071 |
| 3 | cama\_mesa\_banho | 1067 |
| 2 | informatica\_acessorios | 948 |
| 4 | cama\_mesa\_banho | 862 |
| 2 | moveis\_decoracao | 802 |
| 3 | esporte\_lazer | 781 |
| 5 | cama\_mesa\_banho | 736 |
| 8 | relogios\_presentes | 725 |

Taksitlendirilerek en az satın alınan kategoriler:

|  |  |  |
| --- | --- | --- |
| **payment\_installments** | **product\_category\_name** | **order\_count** |
| 24 | ferramentas\_jardim | 1 |
| 24 | moveis\_sala | 1 |
| 24 | casa\_conforto | 1 |
| 24 | eletrodomesticos | 1 |
| 24 | esporte\_lazer | 1 |

24 taksitle satılan bu kategorilerde sadece 1 satış var bu durum şirkete mali açıdan zarar veriyor olabilir üstünde durmakta fayda var.

#### **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.

Recency SQL query:

WITH recency\_scores AS (

SELECT

customerid,

DATE\_PART('day', CURRENT\_DATE - MAX(invoicedate)) AS recency\_score

FROM

rfm

WHERE

invoicedate IS NOT NULL

GROUP BY

customerid

)

SELECT

customerid,

recency\_score,

NTILE(10) OVER (ORDER BY recency\_score ASC) AS recency\_score\_rank

FROM

recency\_scores

ORDER BY

recency\_score\_rank;

Frequency SQL query:

WITH frequency\_scores AS (

SELECT

customerid,

COUNT(DISTINCT invoiceno) AS frequency\_score

FROM

rfm

WHERE

invoiceno IS NOT NULL

GROUP BY

customerid

)

SELECT

customerid,

frequency\_score,

NTILE(10) OVER (ORDER BY frequency\_score DESC) AS frequency\_score\_rank

FROM

frequency\_scores

ORDER BY

frequency\_score\_rank;

Monetary SQL query:

WITH monetary\_scores AS (

SELECT

customerid,

SUM(quantity \* unitprice) AS monetary\_score

FROM

rfm

WHERE

quantity > 0

AND unitprice > 0

AND quantity IS NOT NULL

AND unitprice IS NOT NULL

GROUP BY

customerid

)

SELECT

customerid,

monetary\_score,

11 - NTILE(10) OVER (ORDER BY monetary\_score DESC) AS monetary\_score\_rank

FROM

monetary\_scores

ORDER BY

RFM skoru SQL query:

WITH recency\_scores AS (

SELECT

customerid,

DATE\_PART('day', CURRENT\_DATE - MAX(invoicedate)) AS recency\_score

FROM

rfm

WHERE

invoicedate IS NOT NULL

GROUP BY

customerid

),

frequency\_scores AS (

SELECT

customerid,

COUNT(DISTINCT invoiceno) AS frequency\_score

FROM

rfm

WHERE

invoiceno IS NOT NULL

GROUP BY

customerid

),

monetary\_scores AS (

SELECT

customerid,

SUM(quantity \* unitprice) AS monetary\_score

FROM

rfm

WHERE

quantity > 0

AND unitprice > 0

AND quantity IS NOT NULL

AND unitprice IS NOT NULL

GROUP BY

customerid

),

rfm\_scores AS (

SELECT

r.customerid,

NTILE(10) OVER (ORDER BY r.recency\_score ASC) AS recency\_score\_rank,

NTILE(10) OVER (ORDER BY f.frequency\_score DESC) AS frequency\_score\_rank,

11 - NTILE(10) OVER (ORDER BY m.monetary\_score DESC) AS monetary\_score\_rank

FROM

recency\_scores r

JOIN frequency\_scores f ON r.customerid = f.customerid

JOIN monetary\_scores m ON r.customerid = m.customerid

)

SELECT

customerid,

CONCAT(recency\_score\_rank, '-', frequency\_score\_rank, '-', monetary\_score\_rank) AS rfm\_score

FROM

rfm\_scores

ORDER BY

customerid;

Aynı RFM değerlerine sahip olan müşteri sayıları SQL query:

WITH recency\_scores AS (

SELECT

customerid,

DATE\_PART('day', CURRENT\_DATE - MAX(invoicedate)) AS recency\_score

FROM

rfm

WHERE

invoicedate IS NOT NULL

GROUP BY

customerid

),

frequency\_scores AS (

SELECT

customerid,

COUNT(DISTINCT invoiceno) AS frequency\_score

FROM

rfm

WHERE

invoiceno IS NOT NULL

GROUP BY

customerid

),

monetary\_scores AS (

SELECT

customerid,

SUM(quantity \* unitprice) AS monetary\_score

FROM

rfm

WHERE

quantity > 0

AND unitprice > 0

AND quantity IS NOT NULL

AND unitprice IS NOT NULL

GROUP BY

customerid

),

rfm\_scores AS (

SELECT

r.customerid,

NTILE(10) OVER (ORDER BY r.recency\_score ASC) AS recency\_score\_rank,

NTILE(10) OVER (ORDER BY f.frequency\_score DESC) AS frequency\_score\_rank,

11 - NTILE(10) OVER (ORDER BY m.monetary\_score DESC) AS monetary\_score\_rank

FROM

recency\_scores r

JOIN frequency\_scores f ON r.customerid = f.customerid

JOIN monetary\_scores m ON r.customerid = m.customerid

)

SELECT

CONCAT(recency\_score\_rank, '-', frequency\_score\_rank, '-', monetary\_score\_rank) AS rfm\_score,

COUNT(customerid) AS customer\_count

FROM

rfm\_scores

GROUP BY

rfm\_score

ORDER BY

rfm\_score;

Outputları incelediğimizde:

RFM analizi ‘’1-1-10’’ olan müşteri sayısını toplam 82 kişi olarak görüyoruz. ‘’1-1-10’’ segmentindeki müşterilere özel teklifler veya promosyonlar sunarak sadakatlerini arttırabiliriz.

RFM analizi ‘’10-10-1’’ olan müşteri sayısını toplam 41 kişi olarak görüyoruz. ‘’10-10-1’’ segmentindeki müşteriler genellikle geçmişte sık alışveriş yapan ancak son zamanlarda alışveriş yapmayan ve genellikle düşük harcama yapan müşterilerdir. Bu müşterileri geri kazanmak için özel teklifler veya ‘’Geriye Dönüş’’ kampanyaları gibi stratejiler uygulanabilir. Ayrıca, müşteri sayısını arttırmak ve daha fazla harcama yapmalarını sağlamak için kişiselleştirilmiş pazarlama stratejileri uygulanabilir.

RFM analizi ‘’5-1-10’’ olan müşteri sayısını toplam 12 kişi olarak görüyoruz. Bu segmentteki müşteriler uzun süre önce alışveriş yapmış ancak yüksek miktarlarda harcama yapmış müşterilerdir. Bu müşterilere markamızı hatırlatmak ve tekrar alışveriş yapmalarını sağlamak için stratejiler geliştirilebilir. Örneğin daha önce yaptıkları alışveriş bilgilerine göre ürün inceleme ve öneriler kişisel olarak ulaşılıp paylaşılabilir.

RFM analizi ‘’2-1-10" olan müşteri sayısı toplam 45 kişi olarak görüyoruz. Bu segmentteki müşteriler son alışverişlerini biraz uzun süre önce yapmış ve bu alışverişlerinde düşük miktarlarda harcama yapmış. Bu müşteri segmenti, belki de yeni ürünler veya kampanyalar hakkında bilgilendirilerek tekrar alışveriş yapmaya teşvik edilebilir. Özel bir müşteri etkinliği düzenlemek veya sadakat programı kapsamında özel bir avantaj sunmak, bu müşterilerin ilgisini çekebilir.