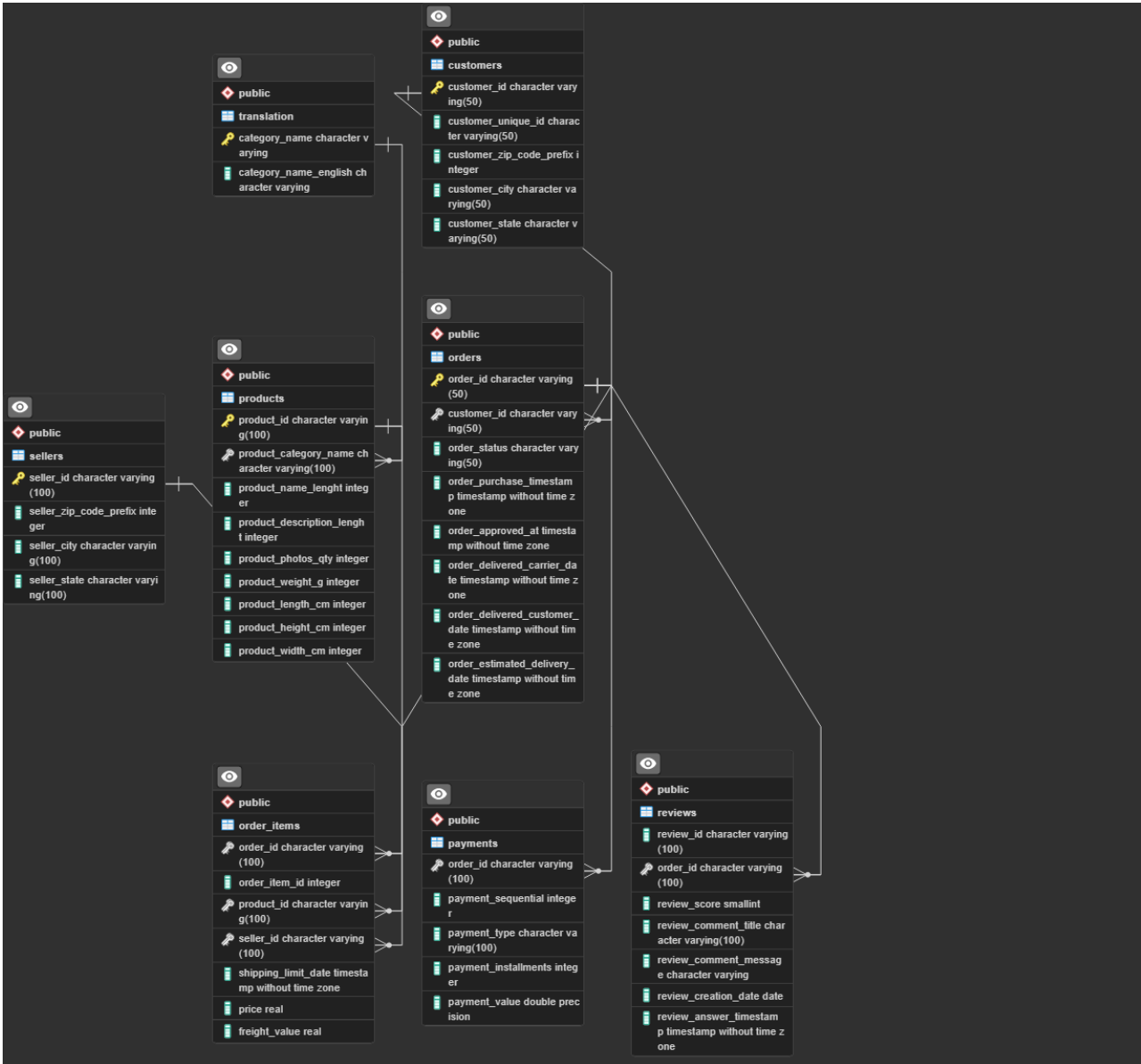


ERD Schema



CASE 1 : Sipariş Analizi

1: Aylık olarak order dağılımını inceleyiniz. Tarih verisi için order_approved_at kullanılmalıdır.

NOT:

İptal edilen ve mevcutta bulunmayan ürünleri filtrelemedim.

Order_approved_at kolonunda boş olan fakat gönderilmiş veya hazırlanmış siparişler olduğu için null kolonu çıkıyor.

SELECT

```
date_trunc('month',order_approved_at)::date AS months,  
COUNT(*)total_amount
```

FROM

```
orders
```

GROUP BY

```
1;
```

	months date	total_amount bigint
1	2016-09-01	1
2	2016-10-01	320
3	2016-12-01	1
4	2017-01-01	760
5	2017-02-01	1765
6	2017-03-01	2689
7	2017-04-01	2374
8	2017-05-01	3693
9	2017-06-01	3252
10	2017-07-01	3974
11	2017-08-01	4348
12	2017-09-01	4301
13	2017-10-01	4590

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.iptal edilen veya yolda olanları ayrı gösterebilirsin

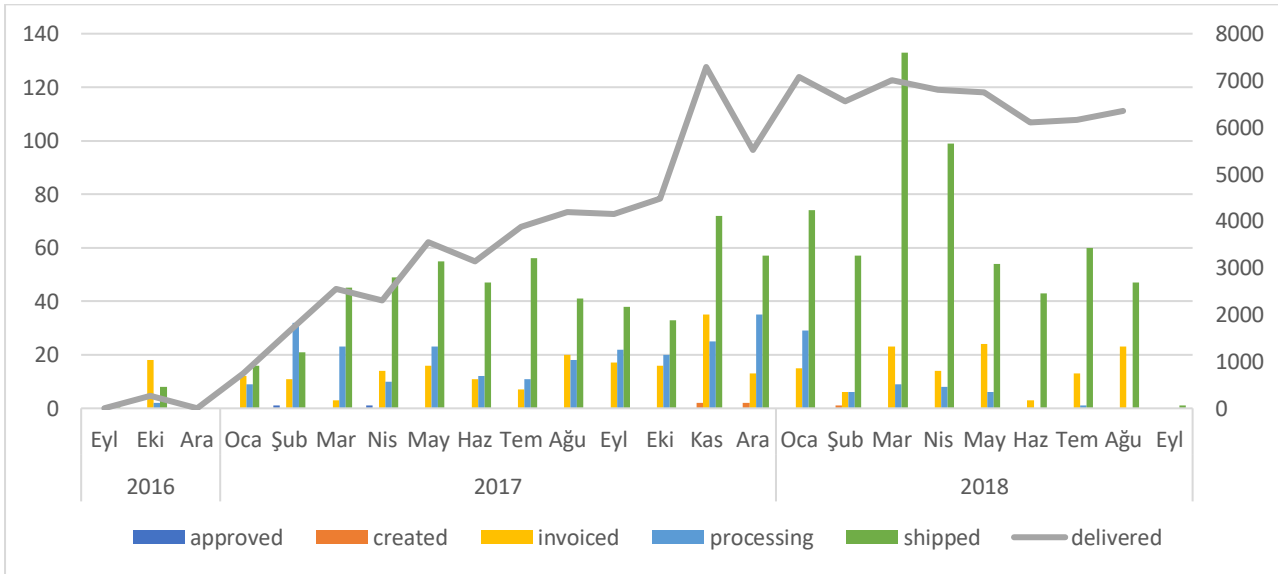
NOT:

Bir önceki soruda belirttiğim gibi , order_approved_at kolonu boş olan, fakat gönderilmiş veya hazırlanmış siparişler olduğu için order_purchase_timestamp kolonunun daha güvenilir olacağını düşündüm.

BAŞARILI SİPARİŞLER İÇİN;

```
SELECT
    date_trunc('month',order_purchase_timestamp)::date AS
months,
    order_status,
    COUNT (*) total_amount
FROM
    orders
WHERE
    order_status NOT IN ('unavailable','canceled')
GROUP BY
    1,2
ORDER BY
    1;
```

	months date	order_status character varying (50)	total_amount bigint
1	2016-09-01	delivered	1
2	2016-09-01	shipped	1
3	2016-10-01	delivered	265
4	2016-10-01	invoiced	18
5	2016-10-01	processing	2
6	2016-10-01	shipped	8
7	2016-12-01	delivered	1
8	2017-01-01	delivered	750
9	2017-01-01	invoiced	12
10	2017-01-01	processing	9



Elimizdeki verilere göre en çok 2017 Kasım ayında sipariş verilmiş.Diğer iki yılın Kasım verileri olmadığından bir karşılaştırma yapamıyoruz. Black Friday ve Black Friday'in dijital hali olan ,2012 yılından beri Brezilya'da Black Friday'in hemen ertesindeki Pazartesi kutlanan Cyber Monday gibi büyük indirim günlerinin Kasım ayında olmalarının siparişlerin artmasında etkileri olduğunu düşünüyorum.Bu iki büyük alışveriş festivalinin yanında Brezilya'da 12 kasım çocuk günü olarak kutlanıyor ve aileler bu tarihte çocuklarına hediyeler alıyor.Ayrıca Amerikan bayramı Halloween'in Brezilya versiyonu olan The Day of Saci de 31 kasımda gerçekleşiyor.Bu festival de kostüm,makyaj malzemesi tarzı ürünlerin siparişlerinin artmasına etki edebilir.

BAŞARISIZ SİPARİŞLER İÇİN;

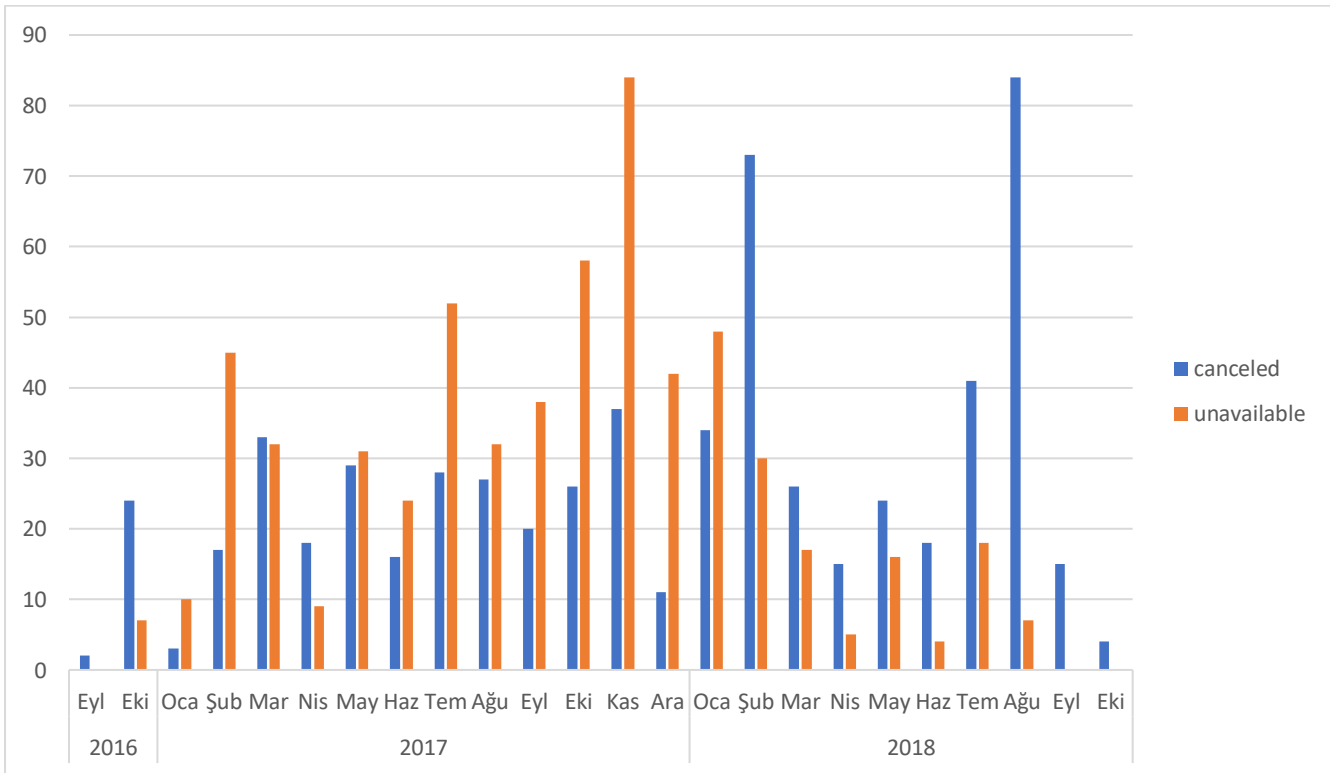
```
SELECT
    date_trunc('month',order_purchase_timestamp)::date as month,
    order_status,
```

```

COUNT (*) total_amount
FROM
orders
WHERE
order_status IN ('unavailable','canceled')
GROUP BY
1,2
ORDER BY
1;

```

	month date	order_status character varying (50)	total_amount bigint
1	2016-09-01	canceled	2
2	2016-10-01	canceled	24
3	2016-10-01	unavailable	7
4	2017-01-01	canceled	3
5	2017-01-01	unavailable	10
6	2017-02-01	canceled	17
7	2017-02-01	unavailable	45
8	2017-03-01	canceled	33
9	2017-03-01	unavailable	32
10	2017-04-01	canceled	18



Elimizdeki verilere göre müşterilere en ‘yok’ denen ay , aynı zamanda en çok sipariş verilen kasım ayı.Buna yoğun siparişlere ürün yetiştirememek neden olmuş olabilir.En yüksek sipariş iptalinin olduğu dönem ise 2018 ağustos ayı.Bu 84 iptalin 53’ünün kategori bilgisi yok ve bu 53 ürünün 28’i de en çok sipariş verilen Sao Paolo bölgesinden.

3 :Ürün kategorisi kırılımında sipariş sayılarını inceleyiniz. Özel günlerde öne çıkan kategoriler nelerdir? Örneğin yılbaşı, sevgililer günü....Her kategoriden ne kadar sipariş var mesela yılbaşında

NOT:

-Sipariş verme tarihini Black Friday'e göre aldım.Çünkü önceki soru da gördüğümüz üzere en çok sipariş verilen ay Kasım ve sonraki soruda görebileceğimiz gibi ayın 24'ü en çok sipariş verilen gün.Bu tarihte bize Black Friday'i veriyor.Brezilya'da Black Friday indirimleri yaklaşık 1 hafta sürdüğünü öğrendim.Bu sebeple sorgudaki filtrelememi buna göre yaptım.

```
WITH BLACKFRIDAY AS (  
    SELECT  
        P.PRODUCT_CATEGORY_NAME,  
        T.CATEGORY_NAME_ENGLISH,  
        COUNT( O.ORDER_ID)BLACKFRIDAY  
    FROM ORDERS O  
    JOIN  
        ORDER_ITEMS OI ON O.ORDER_ID = OI.ORDER_ID  
    JOIN  
        PRODUCTS P ON OI.PRODUCT_ID = P.PRODUCT_ID  
    LEFT JOIN  
        TRANSLATION T ON P.PRODUCT_CATEGORY_NAME = T.CATEGORY_NAME  
    WHERE EXTRACT(MONTH FROM ORDER_PURCHASE_TIMESTAMP) = 11  
        AND EXTRACT(DAY FROM ORDER_PURCHASE_TIMESTAMP) BETWEEN 20 AND 29  
        AND ORDER_STATUS NOT IN ('unavailable','canceled')  
    GROUP BY  
        P.PRODUCT_CATEGORY_NAME,  
        T.CATEGORY_NAME_ENGLISH  
    ORDER BY  
        COUNT( O.ORDER_ID) DESC  
    )  
,REST AS (  
    SELECT  
        P.PRODUCT_CATEGORY_NAME,  
        T.CATEGORY_NAME_ENGLISH,  
        COUNT( O.ORDER_ID)SALES_NOT_BF  
    FROM ORDERS O  
    JOIN  
        ORDER_ITEMS OI ON O.ORDER_ID = OI.ORDER_ID  
    JOIN  
        PRODUCTS P ON OI.PRODUCT_ID = P.PRODUCT_ID  
    LEFT JOIN  
        TRANSLATION T ON P.PRODUCT_CATEGORY_NAME = T.CATEGORY_NAME  
    WHERE EXTRACT(MONTH FROM ORDER_PURCHASE_TIMESTAMP) != 11  
        AND EXTRACT(DAY FROM ORDER_PURCHASE_TIMESTAMP) NOT BETWEEN 20 AND 29  
        AND ORDER_STATUS NOT IN ('unavailable','canceled')  
    GROUP BY  
        P.PRODUCT_CATEGORY_NAME,  
        T.CATEGORY_NAME_ENGLISH  
    ORDER BY  
        COUNT( O.ORDER_ID) DESC  
    )  
,PERCENTAGE AS (  
    SELECT  
        BF.PRODUCT_CATEGORY_NAME,  
        BF.CATEGORY_NAME_ENGLISH,  
        ROUND (
```

```

(SUM(BF.BLACKFRIDAY) / SUM(BF.BLACKFRIDAY) OVER ()) * 100,2) AS BLACKFRIDAY_PERCENTAGE,
ROUND (
(SUM(REST.SALES_NOT_BF) / SUM(REST.SALES_NOT_BF) OVER ()) * 100,2) AS SALES_NOT_BF_PERCENTAGE
FROM
    BLACKFRIDAY BF
JOIN
    REST ON BF.PRODUCT_CATEGORY_NAME = REST.PRODUCT_CATEGORY_NAME
GROUP BY
    BF.PRODUCT_CATEGORY_NAME,
    BF.CATEGORY_NAME_ENGLISH,
    BF.BLACKFRIDAY,
    REST.SALES_NOT_BF
)
SELECT
    PRODUCT_CATEGORY_NAME,
    CATEGORY_NAME_ENGLISH,
    BLACKFRIDAY_PERCENTAGE-SALES_NOT_BF_PERCENTAGE INCREASE_PER
FROM
    PERCENTAGE
WHERE
    BLACKFRIDAY_PERCENTAGE-SALES_NOT_BF_PERCENTAGE>0
ORDER BY
    BLACKFRIDAY_PERCENTAGE-SALES_NOT_BF_PERCENTAGE DESC;

```

	product_category_name character varying (100)	category_name_english character varying	increase_per numeric
1	ferramentas_jardim	garden_tools	3.39
2	brinquedos	toys	2.65
3	moveis_decoracao	furniture_decor	2.40
4	cama_mesa_banho	bed_bath_table	1.99
5	perfumaria	perfumery	0.96
6	consoles_games	consoles_games	0.83
7	fashion_bolsas_e_acessorios	fashion_bags_accessories	0.74
8	telefonica	telephony	0.66
9	casa_conforto	home_comfort	0.22
10	moveis_cozinha_area_de_servico_jantar_e_jardim	kitchen_dining_laundry_garden_furniture	0.22

Bu sorgumda Black Friday’de satışı diğer günlere göre artan ürünleri ve artış oranlarını listeledim. Normal günlerde de zaten çok sipariş alan ürünlerin dışında daha keyfi ürünler olan toys,perfumery,consoles games,fashion bags and accessories kategorilerindeki artışlar görülüyor..Burada indirimlerin satışlara olumlu etkileri olduğunu söyleyebiliriz.

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.

NOT :

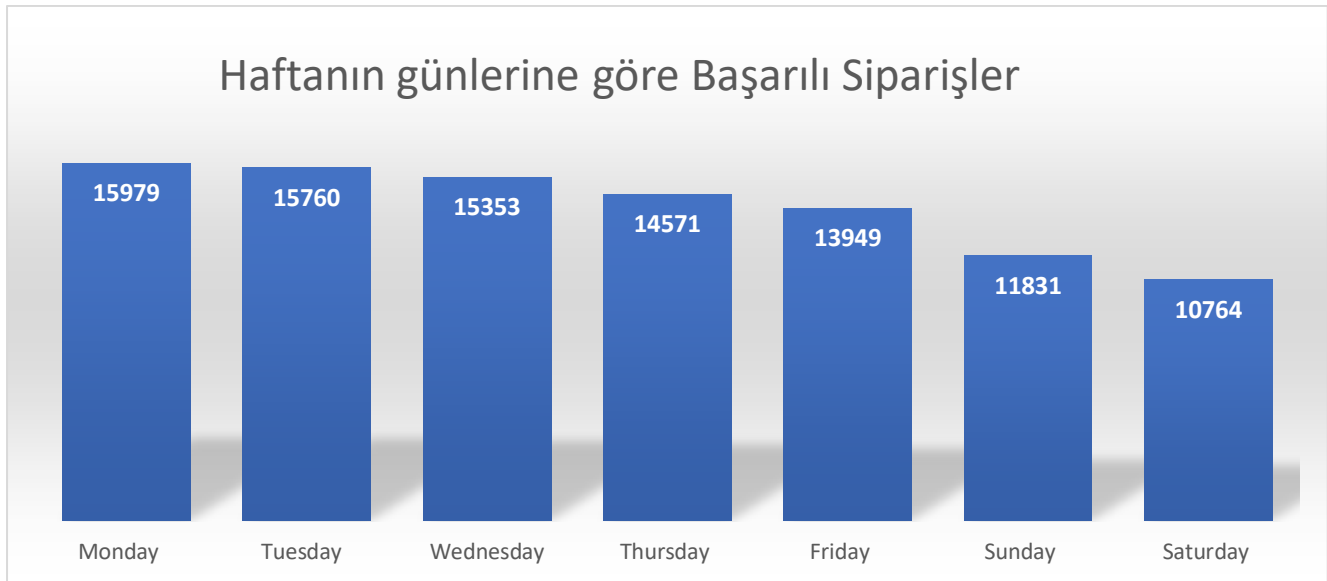
- Siparişin verildiği tarihten aldım.
- İptal ve mevcutta bulunmayan siparişleri filtreledim.
- Sorada vurgulanan order sayısı olduğu için to_char kullandım.

```

SELECT
    INITCAP(TO_CHAR(ORDER_PURCHASE_TIMESTAMP,'day'))DAYS,
    COUNT(DISTINCT ORDER_ID) TOTAL_ORDERS_PURCHASED
FROM
    ORDERS
WHERE
    ORDER_STATUS NOT IN ('canceled','unavailable')
GROUP BY
    1
ORDER BY
    2 DESC;

```

	days text	total_orders_purchased bigint
1	Monday	15979
2	Tuesday	15760
3	Wednesday	15353
4	Thursday	14571
5	Friday	13949
6	Sunday	11831
7	Saturday	10764



Bu sorguda başarılı siparişlerin daha çok hafta içi verildiğini görüyoruz. Bunun nedeni insanların teslim süresini de düşünerek yeni aldıkları ürünleri hafta sonları kullanmak istemeleri olabilir veya insanlar hafta içi günlerde daha az sosyalleştikleri için ürün incelemelerini ve siparişlerini bu günler de yapıyor olabilirler.

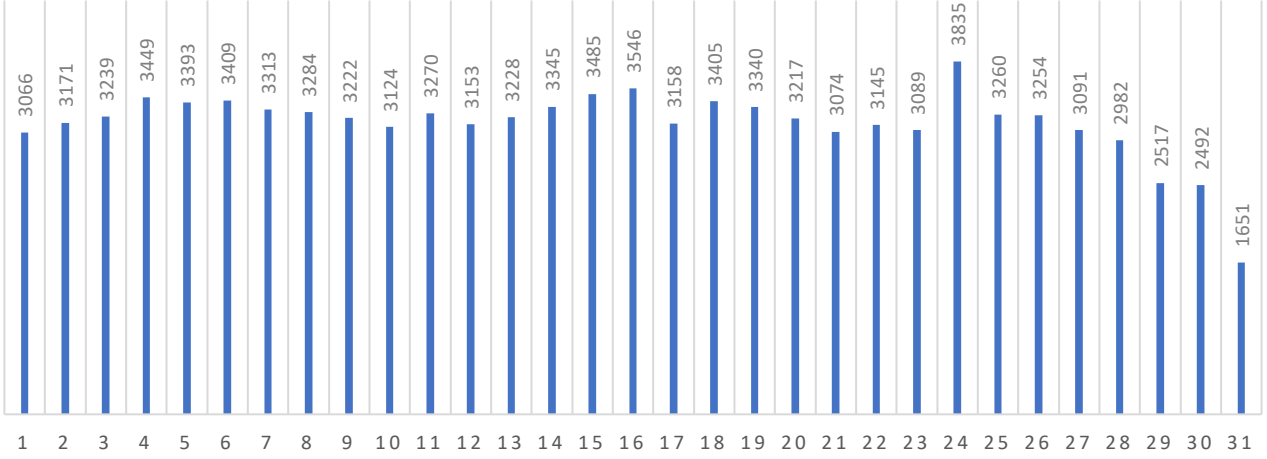
```

SELECT
    TO_CHAR(ORDER_PURCHASE_TIMESTAMP,'dd') DAYS_OFTHE_MONTHS,
    COUNT(DISTINCT ORDER_ID)
TOTAL_ORDERS_PURCHASED
FROM
    ORDERS
WHERE
    ORDER_STATUS NOT IN ('canceled','unavailable')
GROUP BY
    1
ORDER BY
    2 DESC;

```

	days_ofthe_months text	total_orders_purchased bigint
1	24	3835
2	16	3546
3	15	3485
4	04	3449
5	06	3409
6	18	3405
7	05	3393
8	14	3345
9	19	3340
10	07	3313

AYIN GÜNLERİNE GÖRE BAŞARILI SİPARİŞLER



Brezilya da Black Friday 22-29 kasım tarihleri arasında kutlanıyor. Ayrıca yılbaşı da ay sonu. Noel de ayın 24'ünde bu da gün içinde teslim edilebilecek yiyecek ve içecek tarzı siparişleri etkileyebilir. En çok sipariş verilen bölge olan Sao Paulo'nun kurtuluşu da ayın 25'inde. Tüm bu özel günlere baktığımızda en çok sipariş verilen günün ayın 24'nde diğer günlere göre yüksek olması beklenen bir durum.

CASE 2 : Müşteri Analizi

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.

NOTLAR:

1. İptal edilen ve mevcutta bulunmayan siparişleri denklemden çıkardım.

2. ROW_NUMBER() yerine RANK() kullandığımda bazı müşteriler farklı şehirlerden eşit sayıda sipariş verdikleri için o iki şehir de yazacaktı. Tek şehir görünmesi için Row kullandım.

```
WITH CITY_RANK AS(
    SELECT
        C.CUSTOMER_UNIQUE_ID,
        C.CUSTOMER_CITY,
        COUNT(O.ORDER_ID)TOTAL_ORDER,
        ROW_NUMBER() OVER (PARTITION BY C.CUSTOMER_UNIQUE_ID ORDER BY COUNT(O.ORDER_ID)
DESC) RANKING
    FROM
        CUSTOMERS C
    JOIN
        ORDERS O ON O.CUSTOMER_ID = C.CUSTOMER_ID
    WHERE
        O.ORDER_STATUS NOT IN ('unavailable','canceled')
    GROUP BY
        1,2
)
```

```

),TOTAL AS(
    SELECT
        CUSTOMER_UNIQUE_ID,
        CUSTOMER_CITY,
        SUM(TOTAL_ORDER)TOPLAM
    FROM
        CITY_RANK
    GROUP BY
        1,2
    ORDER BY
        SUM(TOTAL_ORDER) DESC
)
,COMBINE AS (
    SELECT
        CITY_RANK.CUSTOMER_UNIQUE_ID,
        CITY_RANK.CUSTOMER_CITY,
        TOTAL.TOPLAM
    FROM
        CITY_RANK
    JOIN
        TOTAL ON CITY_RANK.CUSTOMER_UNIQUE_ID = TOTAL.CUSTOMER_UNIQUE_ID
    WHERE
        CITY_RANK.RANKING = 1
)
SELECT
    CUSTOMER_CITY,
    SUM(TOPLAM)HOP
FROM COMBINE
GROUP BY
    1
ORDER BY
    2 DESC;

```

	customer_city character varying (50) 	hop numeric 
1	sao paulo	15293
2	rio de janeiro	6805
3	belo horizonte	2740
4	brasilia	2111
5	curitiba	1503
6	campinas	1420
7	porto alegre	1370
8	salvador	1234
9	guarulhos	1166
10	sao bernardo do campo	921
11	niteroi	839

CASE 3: Satıcı Analizi

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

Ortalama satış adetini 36 bulduğum için 36 siparişten fazla gönderim yapanları aldım.

Siparişin verildiği ve müşterinin eline ulaştığı zaman aralığını kullandım.

Tek order_id de farklı satıcılar olduğunu bulduğum için order_items tablosundaki siparişleri saydım.

```
WITH DELIVERY_TIMES AS(
    SELECT
        OI.SELLER_ID,
        ROUND(AVG(O.ORDER_DELIVERED_CUSTOMER_DATE::date -
O.ORDER_PURCHASE_TIMESTAMP::date),2)ORDER_DELIVERY_TIME
    FROM
        ORDERS O
    JOIN
        ORDER_ITEMS OI ON O.ORDER_ID = OI.ORDER_ID

    GROUP BY
        OI.SELLER_ID
    HAVING
        COUNT(DISTINCT OI.ORDER_ID) > 36
    ORDER BY
        ORDER_DELIVERY_TIME
)
SELECT
    DT.SELLER_ID,
    ORDER_DELIVERY_TIME,
    COUNT(DISTINCT OI.ORDER_ID)ORDER_TOTAL,
    ROUND(AVG(R.REVIEW_SCORE),2)REV_SCORE,
    COUNT(R.REVIEW_COMMENT_TITLE)COMMENT_COUNT,
    COUNT(R.REVIEW_COMMENT_MESSAGE)MESSAGE_COUNT
FROM
    DELIVERY_TIMES DT
JOIN
    ORDER_ITEMS OI ON DT.SELLER_ID = OI.SELLER_ID
JOIN
    REVIEWS R ON OI.ORDER_ID = R.ORDER_ID
GROUP BY
    DT.SELLER_ID,2
ORDER BY
    ORDER_DELIVERY_TIME
LIMIT
    5;
```

	seller_id character varying (100)	order_delivery_time numeric	order_total bigint	rev_score numeric	comment_count bigint	message_count bigint
1	d13e50eaa47b4cbe9eb81465865d8c...	4.93	69	4.81	9	12
2	6b90f847357d8981edd79a1eb1bf0acb	5.40	48	4.38	13	17
3	d566c37fa119d5e66c4e9052e83ee4...	5.71	71	4.72	18	23
4	138dbe45fc62f1e244378131a6801526	6.10	58	3.98	19	24
5	325f3178fb58e2a9778334621eecdbf9	6.28	46	4.46	0	13

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

NOT:

Kategorisi boş olmayanları ele aldım

İptal edilen ve mevcutta bulunmayan siparişleri denklemden çıkardım.

1 order_id de 1 den çok kategori olabiliyor.Örnek order_id'ler '005d9a5423d47281ac463a968b3936fb', '0c0aeabd6ae9d7a303af615334a220d0', '0f20a3721e8997ae2f771ec8cf0c4d3d' gibi....Bu sebeple distinct kullanmadan sayım yapmanın daha doğru olacağını düşündüm.

```

WITH CATEGORY_SALES AS (
    SELECT
        OI.SELLER_ID,
        P.PRODUCT_CATEGORY_NAME,
        COUNT(O.ORDER_ID)ORDER_TOTAL
    FROM
        ORDER_ITEMS OI
    JOIN
        ORDERS O ON OI.ORDER_ID = O.ORDER_ID
    JOIN
        PRODUCTS P ON P.PRODUCT_ID = OI.PRODUCT_ID
    WHERE
        ORDER_STATUS NOT IN ('unavailable','canceled')
    GROUP BY
        OI.SELLER_ID,
        P.PRODUCT_CATEGORY_NAME
    ORDER BY
        OI.SELLER_ID,
        ORDER_TOTAL DESC
)
SELECT
    SELLER_ID,
    COUNT(PRODUCT_CATEGORY_NAME)CATEGORY_COUNT,
    SUM(ORDER_TOTAL)ORDERS_SUM,
    ROUND((SUM(ORDER_TOTAL)/COUNT(DISTINCT PRODUCT_CATEGORY_NAME)),2) as
sales_rate_per_category
FROM
    CATEGORY_SALES
WHERE
    PRODUCT_CATEGORY_NAME IS NOT NULL
GROUP BY

```

SELLER_ID
ORDER BY
4 DESC;

	seller_id character varying (100)	category_count bigint	orders_sum numeric	sales_rate_per_category numeric
1	1f50f920176fa81dab994f9023523100	3	1930	643.33
2	ea8482cd71df3c1969d7b9473ff13abc	2	1202	601.00
3	7a67c85e85bb2ce8582c35f2203ad7...	2	1170	585.00
4	cc419e0650a3c5ba77189a1882b75...	4	1766	441.50
5	8581055ce74af1daba164fdbd55a40...	1	432	432.00
6	06a2c3af7b3aee5d69171b0e14f0ee...	1	405	405.00
7	620c87c171fb2a6dd6e8bb4dec959f...	2	789	394.50
8	da8622b14eb17ae2831f4ac5b9dab8...	4	1551	387.75
9	1025f0e2d44d7041d6cf58b6550e0bfa	4	1428	357.00
10	c70c1b0d8ca86052f45a432a38b739...	1	338	338.00

Bu sorguyu toplam satışa göre getirdiğimizde en fazla başarılı satış yapanların kategori sayıları , bu sorguda en çok kategoriye sahip olan satıcılardan oldukça az.Burdan anlayabileceğimiz satış adetinin kategori sayısında bağımsız olduğudur.Daha çok kategoriye sahip olanların daha fazla satış yaptığını söyleyemeyiz.Ayrıca kategori başına ürün satışlarında da bu belli oluyor.Listemizin ilk 10 sırasındaki satıcıların kategorileri diğer satıcılara göre az ama kategori başına satış adetleri fazla.

CASE 4 : Payment Analizi

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

ORTALAMA TAKSİT SAYISINI 3 DEN FAZLASINI ALDIM.

```
SELECT
    C.CUSTOMER_STATE,
    COUNT(DISTINCT O.ORDER_ID)TOTAL_DISTINCT_ORDER
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 > 3
GROUP BY
    1
ORDER BY
    2 DESC;
```

	customer_state character varying (50) 🔒	total_distinct_order bigint 🔒
1	SP	10300
2	RJ	3904
3	MG	3551
4	RS	1663
5	PR	1432
6	BA	1171
7	SC	1039
8	PE	655
9	ES	626
10	GO	624

Taksitli alışverişi daha çok kullanan bölgeler aynı zamanda ülkenin nüfusunun en yoğun olduğu ve alışverişin daha çok yapıldığı bölgeler, az olan bölgeler ise nüfusun az olduğu bölgeler. Burada taksit ve alışveriş adetlerinin nüfus ile doğru orantılı olduğunu görebiliyoruz.

Fakat bence aşağıda yazdığım sorgudan daha sağlıklı bilgi edinebiliriz. Bölgelerin pesin ve taksitli ödemelerine bakıp o bölgenin ekonomik durumu hakkında daha rahat yorum yapabiliriz. Örneğin yukarıda 3 taksitten fazla taksitle ödeme yapanlar SP bölgesinde en çok. Fakat aşağıda gördüğümüz gibi yüzde olarak 3 taksitten daha az kullanan bölge yani taksitsiz veya 3 taksitten az alışverişin daha fazla olduğu bölge.

```

WITH TOPLAM AS(
    SELECT
        C.CUSTOMER_STATE,
        COUNT(DISTINCT O.ORDER_ID)TOTAL_DISTINCT_ORDER
    FROM
        PAYMENTS P
    JOIN
        ORDERS O ON P.ORDER_ID = O.ORDER_ID
    JOIN
        CUSTOMERS C ON O.CUSTOMER_ID = C.CUSTOMER_ID
    GROUP BY
        1
    ORDER BY
        2 DESC
),
PESIN AS(
    SELECT
        C.CUSTOMER_STATE,
        COUNT(DISTINCT O.ORDER_ID)PESIN
    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 <3
    GROUP BY
        1

```

```

ORDER BY
    2 DESC
),
TAKSIT AS(
    SELECT
        C.CUSTOMER_STATE,
        COUNT(DISTINCT O.ORDER_ID)TAKSIT
    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 > 3
    GROUP BY
        1
    ORDER BY
        2 DESC
)

SELECT
    T.CUSTOMER_STATE,
    ROUND((PESIN * 1.0 / TOTAL_DISTINCT_ORDER * 1.0) * 100,2) LESS_THAN_3,
    ROUND((TAKSIT * 1.0 / TOTAL_DISTINCT_ORDER * 1.0) * 100,2) MORE_THAN_3
FROM
    TOPLAM T
JOIN
    PESIN P ON T.CUSTOMER_STATE = P.CUSTOMER_STATE
JOIN
    TAKSIT TA ON P.CUSTOMER_STATE = TA.CUSTOMER_STATE
ORDER BY
    MORE_THAN_3 DESC;

```

	customer_state character varying (50) 🔒	less_than_3 numeric 🔒	more_than_3 numeric 🔒		customer_state character varying (50) 🔒	less_than_3 numeric 🔒	more_than_3 numeric 🔒
1	PB	46.64	42.72	15	AP	58.82	32.35
2	AL	47.22	42.37	16	MA	58.37	31.73
3	SE	49.14	42.29	17	GO	58.61	30.89
4	AC	46.91	41.98	18	ES	58.63	30.79
5	RN	48.04	41.86	19	MG	58.95	30.52
6	PE	50.18	39.65	20	RS	60.03	30.42
7	CE	49.40	39.60	21	RJ	59.06	30.38
8	PI	52.73	38.79	22	MS	60.98	30.07
9	RO	54.15	37.15	23	SC	62.52	28.57
10	PA	55.59	34.87	24	PR	62.32	28.38
11	BA	54.41	34.64	25	RR	65.22	28.26
12	MT	56.34	33.74	26	DF	64.77	24.95
13	TO	58.21	33.57	27	SP	65.75	24.67
14	AM	56.08	32.43				

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

Her bir order tek bir ödeme yöntemiyle ödenmiş

```
SELECT
    P.PAYMENT_TYPE,
    COUNT(DISTINCT O.ORDER_ID)ORDER_COUNT,
    ROUND(SUM(P.PAYMENT_VALUE::decimal),2)TOTAL_PAYMENT
FROM
    PAYMENTS P
JOIN
    ORDERS O ON P.ORDER_ID = O.ORDER_ID
WHERE
    ORDER_STATUS NOT IN ('unavailable','canceled')
GROUP BY
    1
ORDER BY
    2 DESC;
```

	payment_type character varying (100) 🔒	order_count bigint 🔒	total_payment numeric 🔒
1	credit_card	75618	12350042.56
2	boleto	19539	2826802.30
3	voucher	3745	349874.40
4	debit_card	1515	212417.75

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

NOT:

Payment sequential kolonu aylık bazda değil ödeme bazlı taksit.Her ay veya belli dönemlerde değil ama ödemeyi bölerek yapılan ödemeler.Payment installments ise her ay düzenli ödeme olduğu için taksit kolonu olarak payment installments'ı aldım.Ayrıca github ve medium tarzı sitelerde de yaptığım araştırmalarda taksit için yine payment installments kullanılmış.

```
WITH PAYMENT_STYLE AS (
    SELECT
        PR.PRODUCT_CATEGORY_NAME,
        T.CATEGORY_NAME_ENGLISH,
        P.PAYMENT_INSTALLMENTS,
        COUNT(DISTINCT O.ORDER_ID) TOTAL_ORDERS,
        CASE
            WHEN PAYMENT_INSTALLMENTS <= 1 THEN 'Tek Çekim'
            ELSE 'Taksitli'
        END AS PAY_TYPE,
        DENSE_RANK() OVER (PARTITION BY CASE WHEN PAYMENT_INSTALLMENTS <= 1 THEN 'Tek Çekim'
            ELSE 'Taksitli'
        END ORDER BY COUNT(DISTINCT O.ORDER_ID) DESC) AS RANKING
    )
```

```

FROM
    PAYMENTS P
JOIN
    ORDERS O ON P.ORDER_ID = O.ORDER_ID
JOIN
    ORDER_ITEMS OI ON O.ORDER_ID = OI.ORDER_ID
JOIN
    PRODUCTS PR ON PR.PRODUCT_ID = OI.PRODUCT_ID
LEFT JOIN
    TRANSLATION T ON PR.PRODUCT_CATEGORY_NAME = T.CATEGORY_NAME
WHERE
    O.ORDER_STATUS NOT IN ('unavailable','canceled')
GROUP BY
    1,2,3
)

SELECT
    *
FROM
    PAYMENT_STYLE
WHERE
    RANKING BETWEEN 1 AND 5;

```

	product_category_name 	category_name_english 	payment_installments 	total_orders 	pay_type 	ranking 
1	beleza_saude	health_beauty	2	1187	Taksitli	1
2	cama_mesa_banho	bed_bath_table	2	1082	Taksitli	2
3	beleza_saude	health_beauty	3	1078	Taksitli	3
4	esporte_lazer	sports_leisure	2	1065	Taksitli	4
5	cama_mesa_banho	bed_bath_table	3	1062	Taksitli	5
6	esporte_lazer	sports_leisure	1	4272	Tek Çekim	1
7	informatica_acessorios	computers_accessories	1	4154	Tek Çekim	2
8	beleza_saude	health_beauty	1	3863	Tek Çekim	3
9	cama_mesa_banho	bed_bath_table	1	3529	Tek Çekim	4
10	moveis_decoracao	furniture_decor	1	3138	Tek Çekim	5

En çok taksit seçeneği kullanılan kategori health_beauty , en çok peşin ödenen kategori ise sport_leisure

RFM ANALİZİ

Burada amacımız kullanıcıların sipariş alışkanlıklarını anlamak ve değerli kullanıcıları belirlemek olduğu için eksi değerleri ve customerid kolonu boş olan kullanıcıları analize dahil etmiyorum.

RECENCY :

```
SELECT
    CUSTOMERID,
    '2011-12-10' -
    MAX(INVOICEDATE::date)TIME_PASSED
FROM
    RFM
WHERE
    CUSTOMERID IS NOT NULL
    AND UNITPRICE * QUANTITY > 0
GROUP BY
    CUSTOMERID
ORDER BY
    CUSTOMERID;
```

	customerid integer	time_passed integer
1	12346	326
2	12347	3
3	12348	76
4	12349	19
5	12350	311
6	12352	37
7	12353	205
8	12354	233
9	12355	215
10	12356	23

FREQUENCY :

```
SELECT
    CUSTOMERID,
    COUNT (DISTINCT INVOICENO)TOTAL_INVOICE
FROM
    RFM
WHERE
    CUSTOMERID IS NOT NULL
    AND UNITPRICE * QUANTITY > 0
GROUP BY
    CUSTOMERID
ORDER BY
    2 DESC;
```

	customerid integer	total_invoice bigint
1	12748	209
2	14911	201
3	17841	124
4	13089	97
5	14606	93
6	15311	91
7	12971	86
8	14646	73
9	16029	63
10	13408	62

MONETARY:

```
SELECT
    CUSTOMERID,
    ROUND(SUM(UNITPRICE * QUANTITY)::decimal,2)
FROM
    RFM
WHERE CUSTOMERID IS NOT NULL
    AND UNITPRICE * QUANTITY > 0
GROUP BY
    CUSTOMERID
ORDER BY
    2;
```

	customerid integer	round numeric
1	16738	3.75
2	14792	6.20
3	16454	6.90
4	17956	12.75
5	16878	13.30
6	15823	15.00
7	17763	15.00
8	13307	15.00
9	16093	17.00
10	16953	20.80

RFM :

```
WITH RFM_VALUES AS (
    SELECT
        CUSTOMERID,
        '2011-12-10' - MAX(INVOICEDATE::date) RECENCY,
        COUNT (DISTINCT INVOICENO) FREQUENCY,
        ROUND(SUM(UNITPRICE * QUANTITY)::decimal,2) MONETARY
    FROM
        RFM
    WHERE CUSTOMERID IS NOT NULL
        AND UNITPRICE * QUANTITY > 0
    GROUP BY
        CUSTOMERID)
,RFM_RATES AS (
    SELECT
        CUSTOMERID,
        RECENCY,
        NTILE(5) OVER (ORDER BY RECENCY DESC) RECENCY_SCORE,
        FREQUENCY,
        NTILE(5) OVER (ORDER BY FREQUENCY) FREQUENCY_SCORE,
        MONETARY,
        NTILE(5) OVER (ORDER BY MONETARY) MONETARY_SCORE
    FROM
        RFM_VALUES)
,RFM_ANALYSIS AS (
    SELECT
```

```

CUSTOMERID,
REGENCY_SCORE,
FREQUENCY_SCORE,
MONETARY_SCORE,
CONCAT(REGENCY_SCORE,FREQUENCY_SCORE,MONETARY_SCORE) RFM_SCORE
FROM
RFM_RATES
ORDER BY
CUSTOMERID)
SELECT *,
CASE
    WHEN (REGENCY_SCORE = 1 OR REGENCY_SCORE = 2) AND (FREQUENCY_SCORE = 1 OR
FREQUENCY_SCORE = 2) THEN 'Hibernating'
    WHEN (REGENCY_SCORE = 1 OR REGENCY_SCORE = 2) AND (FREQUENCY_SCORE = 3 OR
FREQUENCY_SCORE = 4) THEN 'At Risk'
    WHEN (REGENCY_SCORE = 1 OR REGENCY_SCORE = 2) AND FREQUENCY_SCORE = 5 THEN 'Cant
Loose'
    WHEN REGENCY_SCORE = 3 AND (FREQUENCY_SCORE = 1 OR FREQUENCY_SCORE = 2) THEN
'About_to_Sleep'
    WHEN REGENCY_SCORE = 3 AND FREQUENCY_SCORE = 3 THEN 'Need_Attention'
    WHEN (REGENCY_SCORE = 3 OR REGENCY_SCORE = 4) AND (FREQUENCY_SCORE = 4 OR
FREQUENCY_SCORE = 5) THEN 'Loyal_Customers'
    WHEN REGENCY_SCORE = 4 AND FREQUENCY_SCORE = 1 THEN 'Promising'
    WHEN REGENCY_SCORE = 5 AND FREQUENCY_SCORE = 1 THEN 'New_Customers'
    WHEN (REGENCY_SCORE = 4 OR REGENCY_SCORE = 5) AND (FREQUENCY_SCORE = 2 OR
FREQUENCY_SCORE = 3) THEN 'Potential_Loyalists'
    WHEN REGENCY_SCORE = 5 AND (FREQUENCY_SCORE = 4 OR FREQUENCY_SCORE = 5) THEN
'Champions'
END AS CLIENT_SEGMENT
FROM
RFM_ANALYSIS;

```

	customerid integer	recency_score integer	frequency_score integer	monetary_score integer	rfm_score text	client_segment text
1	12346	1	2	5	125	Hibernating
2	12347	5	5	5	555	Champions
3	12348	2	4	4	244	At Risk
4	12349	4	2	4	424	Potential_Loyalists
5	12350	1	2	2	122	Hibernating
6	12352	3	5	5	355	Loyal_Customers
7	12353	1	1	1	111	Hibernating
8	12354	1	1	4	114	Hibernating
9	12355	1	1	2	112	Hibernating
10	12356	4	4	5	445	Loyal_Customers

NOT:

NTILE() eşit böldüğü için frequency değerleri aynı olmasına rağmen bir noktadan itibaren frequency scoreları farklı çıkabiliyor.