

Invoice Tablosu

invoice_id	customer_id	invoice_date	billing_address	billing_city	billing_state	billing_country	billingpostal_code	total
1	18	2017-01-03 00:00:00	627 Broadway	New York	NY	USA	10012-2612	15.84
2	30	2017-01-03 00:00:00	230 Elgin Street	Ottawa	ON	Canada	K2P 1L7	9.9
3	40	2017-01-05 00:00:00	8, Rue Hanovre	Paris	None	France	75002	1.98
4	18	2017-01-06 00:00:00	627 Broadway	New York	NY	USA	10012-2612	7.92

Sorular:

1- Invoice tablosunda, tüm değerleri NULL olan kayıtların sayısını bulmanız isteniyor. Bu işlemi tek bir sorgu ile yapmalısınız. Sorguyu yazdıktan sonra, PostgreSQL'in sol alt kısmındaki Row sayısını, SQL sorgunuzda yorum satırında belirtmeniz gerekmektedir.

```
-- Tüm sütunları NULL olan kayıtların sayısını bulma
SELECT COUNT(*) AS null_record_count
FROM Invoice
WHERE COALESCE(
    CAST(invoice_id AS text),
    CAST(customer_id AS text),
    CAST(invoice_date AS text),
    billing_address,
    billing_city,
    billing_state,
    billing_country,
    billingpostal_code,
    CAST(total AS text)
) IS NULL; -- Null olan satır sayısı = 0
```

Burada COALESCE fonksiyonunu kullanarak Invoice tablosu içindeki tüm sütunların NULL olup olmadığı kontrol ediliyor. Ayrıca COALESCE tüm sütunlardaki değerleri karşılaştırırken aynı datatype istediği için burada CAST fonksiyonu ile sayısal değerde

olan sütunlar text tipine dönüştürülmüştür. Eğer ki tüm satırlar NULL ise sonuç da NULL olarak dönecektir.

	null_record_count bigint	
1	0	

2- Koordinasyondaki kişiler, Total değerlerinde bir hata olduğunu belirtiyorlar. Bu değerlerin iki katını görmek ve eski versiyonlarıyla birlikte karşılaştırmak için bir sorgu yazmanız isteniyor. Ayrıca, verilerin daha rahat takip edilebilmesi için, tablonun yeni versiyonuna ait kolona göre büyükten küçüğe sıralama yapılması isteniyor.

```
-- Total değerlerinin iki katını alarak yeni bir sütun oluşturma ve sıralama
SELECT invoice_id, total AS eski_total, total * 2 AS yeni_total
FROM Invoice
ORDER BY yeni_total DESC;
```

Burada total değerinde hata olduğu için total değerinin iki katı alınarak Invoice tablosunda yeni total değeri elde edilmiştir. Ayrıca bu yeni total kolonu da ORDER BY ve DESC komutları ile büyükten küçüğe sıralanmıştır.

	invoice_id [PK] integer	eski_total numeric (10,2)	yeni_total numeric
1	404	25.86	51.72
2	299	23.86	47.72
3	194	21.86	43.72
4	96	21.86	43.72
5	89	18.86	37.72
6	201	18.86	37.72
7	88	17.91	35.82
8	313	16.86	33.72
9	306	16.86	33.72
10	103	15.86	31.72
11	208	15.86	31.72
12	193	14.91	29.82
13	5	13.86	27.72
14	26	13.86	27.72
15	362	13.86	27.72
16	124	13.86	27.72
17	180	13.86	27.72
18	173	13.86	27.72
19	131	13.86	27.72
20	166	13.86	27.72
Total rows: 412		Query complete 00:00:00.138	

3- . Adres kolonundaki verileri, soldan 3 karakter ve sağdan 4 karakter olarak birleştirmeniz ve "Açık Adres" olarak yazmanız isteniyor. Ayrıca, bu yeni açık adresi 2013 yılı ve 8. ay'a göre filtrelemeniz gerekiyor.

```
--Adres kolonunu kısaltma ve 2013 yılının 8. ayına göre filtreleme
SELECT invoice_id,
       CONCAT(LEFT(billing_address, 3), '...', RIGHT(billing_address, 4)) AS "Açık Adres",
       invoice_date
FROM Invoice
WHERE DATE_PART('year', invoice_date) = 2013
      AND DATE_PART('month', invoice_date) = 8;
```

Burada adres kolonundaki verileri filtrelemek için iç içe fonksiyonlar(nested functions) kullanılmıştır. LEFT fonksiyonu ile soldan 3 karakter, RIGHT fonksiyonu ile sağdan 4 karakter alınmış ve aralarına karışıklığı gidermek adına “...” konulmuştur. En son da bu iki veriyi birleştirerek yeni bir kolon elde etmek adına CONCAT fonksiyonu kullanılmıştır. Daha sonra DATE_PART fonksiyonu ile invoice_date sütunundan istenilen yıl ve ay bilgisine göre filtreleme yapılmıştır.

	invoice_id [PK] integer	Açık Adres text	invoice_date timestamp without time zone
1	378	3 C...reet	2013-08-02 00:00:00
2	379	Lij...20bg	2013-08-02 00:00:00
3	380	C/ ...o 85	2013-08-03 00:00:00
4	381	110...n PI	2013-08-04 00:00:00
5	382	Av....2170	2013-08-07 00:00:00
6	383	Rua... 155	2013-08-12 00:00:00
7	384	162...reet	2013-08-20 00:00:00