

## 1.SORU

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.

```
SELECT COUNT(*) FROM invoice
WHERE invoice_id IS NULL
  AND customer_id IS NULL
  AND invoice_date IS NULL
  AND billing_address IS NULL
  AND billing_city IS NULL
  AND billing_state IS NULL
  AND billing_country IS NULL
  AND billingpostal_code IS NULL
  AND total IS NULL;

--Total Row = 0
```

- Bu sorguda count ifadesi kullanarak getireceğimiz şartlara göre toplam satır sayısını getirmek için kullanıyoruz.
- Daha sonra kullanacağımız "invoice" tablosunu belirttik.
- WHERE şartı ile de tek tek bütün kolonların NULL olann verilerini getirmek için AND ifadesiyle bağlayarak sonucu getirdik.
- Sonuç olarak bütün kolonları null olan hiçbir satır bulunmamakta...

## 2. SORU

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.

```
SELECT total, 2*total AS new_total FROM invoice
ORDER BY new_total DESC;
```

- Bu kısımda ilk olarak select ifadesi yanına total değerinin getirileceğini yazıyoruz. (kolonlar üzerinde aritmetik işlemler gerçekleştirebiliyoruz bu sayede) 2\*total diyerek total kolonundaki değerlerin 2 katını almış olduk ve alias kullanarak yeni oluşan bu ifadelere de new\_total kolonunda yer verdik.
- Daha sonra da Order by kullanarak yeni kolonu da DESC ifadesiyle artandan azalana doğru sıraladık.

### 3. SORU

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.

```
SELECT CONCAT(SUBSTRING(billing_address, 1,3),
              SUBSTRING(billing_address, LENGTH(billing_address)-3, 4)) AS acik_adres,
              invoice_date
FROM invoice
WHERE TO_CHAR(invoice_date, 'YYYY') = '2013'
      AND TO_CHAR(invoice_date, 'MM') = '08';
```

- Bu soruda birden fazla fonksiyon kullanmak zorundayız.
- İlk olarak Concat fonksiyonu ile birleştirme yapacağımız için onu yazıyoruz daha sonrasında concat fonksiyonuna birleştireceği parametreleri fonksiyon cinsinden vermeliyiz çünkü birleştireceğimiz ifadeleri substring ile istenildiği gibi ayıklamalıyız.
- Substring içinde billing\_address kolonun soldan 3 karakterini ilk kısımda, ikinci kısımda da sağdan yani adresin sonundan 4 karakter almak için gerekli olan ifadeyi yazıyoruz.
- Sağdan alma yaparken LENGTH fonksiyonu ile ifadenin boyutunu belirtip ondan 3 çıkararak başlangıç ifadesini veriyoruz. Daha sonra bu birleştirerek oluşturduğumuz ifadeye as kelimesi ile acik\_adres adını veriyoruz.
- To char fonksiyonu ile date kısmında şart ifadesini yazıyoruz. Bu fonksiyon sayesinde invoice\_date kolonunda yılı 2013 ve ay bilgisini de 8. Ay gelecek şekilde getiriyoruz.
- Sonuç da çıktımız aşağıdaki gibi oluyor.

	acik_adres text	invoice_date timestamp without time zone
1	3 Creet	2013-08-02 00:00:00
2	Lij20bg	2013-08-02 00:00:00
3	C/ o 85	2013-08-03 00:00:00
4	110n Pl	2013-08-04 00:00:00
5	Av.2170	2013-08-07 00:00:00
6	Rua 155	2013-08-12 00:00:00
7	162reet	2013-08-20 00:00:00