

## TurkStudentCo Data Science Bootcamp - 2. Hafta SQL Ödevi

### 1. NULL Değerli Kayıtların Sayısını Bulan Sorgu

```
SELECT COUNT(*) AS null_record_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;
```

#### Açıklama:

Bu sorgu Invoice tablosundaki tüm sütunları NULL olan kayıtların sayısını bulur. COUNT(\*) fonksiyonunda belirlenen koşulları sağlayan satırların sayısını döndürür.

### 2. Total Değerlerinin İki Katını Hesaplayan ve Karşılaştıran Sorgu

```
SELECT invoice_id, customer_id, total AS old_total, (total * 2) AS new_total  
FROM Invoice  
ORDER BY new_total DESC;
```

#### Açıklama:

Bu sorgu Invoice tablosundaki total sütununun eski ve iki katına çıkarılmış yeni versiyonunu gösterir ve sonuçları büyükten küçüğe sıralar.

- total AS old\_total: Mevcut toplam değerini "old\_total" olarak adlandırır.
- (total \* 2) AS new\_total: total değerini iki katına çıkarır ve "new\_total" olarak adlandırır.
- ORDER BY new\_total DESC: Yeni toplam değerine göre azalan sıralama yapar.

### 3. Adres Verilerini İşleyerek Açık Adres Oluşturan ve Belirtilen Tarihe Göre Filtreleme Yapan Sorgu

```
SELECT invoice_id,  
       customer_id,  
       CONCAT(LEFT(billing_address, 3), '...', RIGHT(billing_address, 4)) AS "Açık Adres",  
       invoice_date  
FROM Invoice  
WHERE EXTRACT(YEAR FROM invoice_date) = 2013  
      AND EXTRACT(MONTH FROM invoice_date) = 8;
```

#### Açıklama:

- LEFT(billing\_address, 3): Adresin ilk 3 karakterini alır.
- RIGHT(billing\_address, 4): Adresin son 4 karakterini alır.
- CONCAT(...): Bu değerleri birleştirerek "Açık Adres" oluşturur.
- EXTRACT(YEAR FROM invoice\_date) = 2013 AND EXTRACT(MONTH FROM invoice\_date) = 8: 2013 yılının 8. ayındaki kayıtları filtreler.

Bu sayede belirli bir tarih aralığında yer alan adresleri kısaltarak görüntülenir.