Invoice Tablosu Analizi Raporu

Giriş

Bu raporda, PostgreSQL kullanarak invoice tablosu üzerinde gerçekleştirilen çeşitli analizler açıklanmaktadır. Çalışmada **NULL değer kontrolü, matematiksel işlemler, adres düzenleme ve tarih bazlı filtreleme** gibi işlemler gerçekleştirilmiştir. Her sorgunun amacı ve nasıl çalıştığı detaylandırılmıştır.

Sorgular ve Açıklamaları

1. Tüm Değerleri NULL Olan Kayıtları Bulma

Yöntem 1: Tüm Sütunları Tek Tek Kontrol Etme

```
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;
```

Nasıl Çalışır?

Bu sorgu, invoice tablosundaki tüm sütunları NULL olup olmadığını kontrol ederek kaç tane tamamen NULL olan satır bulunduğunu sayar. Eğer hiçbir satır NULL değilse, **çıktı: 0** olur.

Yöntem 2: Daha Kısa ve Optimize Çözüm (PostgreSQL IS DISTINCT FROM NULL Kullanımı)

```
SELECT COUNT(*)
FROM invoice
WHERE NOT(invoice IS DISTINCT FROM NULL);
```

Nasıl Çalışır?

- IS DISTINCT FROM NULL, PostgreSQL'e özgü bir işlemdir.
- Tablonun herhangi bir sütununun NULL dışında bir değer içerip içermediğini kontrol eder.
- Daha kısa ve optimize bir sorgu olup, yeni sütunlar eklense bile çalışmaya devam eder.

---

2. Total Değerlerinin İki Katını Almak ve Sıralamak

```
SELECT invoice_id, total AS eski_total, (total * 2) AS yeni_total
FROM invoice
ORDER BY yeni_total DESC;
```

Nasıl Çalışır?

- total sütunundaki her değerin **2 katı alınır** ve yeni_total adıyla yeni bir sütun oluşturulur.
- ORDER BY yeni_total DESC ifadesiyle sonuç **büyükten küçüğe** sıralanır.
- Bu analiz, satış toplamlarının nasıl değiştiğini anlamaya yardımcı olur.

---

3. Adres Kolonunu Düzenleme ve Belirli Tarihlere Göre Filtreleme

```
SELECT
   invoice_id,
   customer_id,
   billing_address,
   LEFT(billing_address, 3) || '...' || RIGHT(billing_address, 4) AS "Açık Adres",
   invoice_date
FROM invoice
WHERE TO_CHAR(invoice_date, 'YYYY-MM') = '2013-08';
```

Nasıl Çalışır?

- LEFT(billing_address, 3) fonksiyonu, adresin **ilk 3 karakterini alır**.
- RIGHT(billing_address, 4) fonksiyonu, adresin **son 4 karakterini alır**.
- || '...' || ifadesiyle **araya ... eklenerek** adresin kısaltılmış bir versiyonu oluşturulur.
- TO_CHAR(invoice_date, 'YYYY-MM') = '2013-08' ifadesiyle yalnızca **Ağustos 2013** tarihine ait faturalar seçilir.
- Bu yöntem, müşteri gizliliğini koruyarak adres verisini daha okunaklı hale getirir.

Sonuç

Bu çalışmada, **fatura verilerini analiz etmek için farklı SQL teknikleri kullanıldı**.

1. NULL değerleri kontrol ettik ve verinin bütünlüğünü doğruladık.
2. Fatura toplamlarını işleyerek, gelir analizine katkı sağladık.
3. Müşteri adreslerini düzenleyip, belirli tarih aralıklarına göre verileri süzdük.

Bu sorgular, veri temizliği ve analiz süreçlerinde **PostgreSQL'in güçlü fonksiyonlarını kullanarak veriye daha iyi hakim olmayı** sağlamaktadır.