

# Invoice ve Müzik Veritabanı Analizi

Bu rapor, bir müzik veritabanındaki **Invoice**, **Track**, **Playlist**, **Album** ve **Artist** tabloları üzerinde gerçekleştirilen SQL sorgularını ve bu sorguların nasıl çalıştığını açıklamaktadır.

---

## Soru 1: USA Ülkesine Ait 2009 Yılı Fatura Toplamı

Bu sorgu, **invoice** tablosundan **USA** ülkesine ait ve **2009** yılı içerisinde oluşturulmuş faturaların toplamını hesaplamaktadır.

### Sorgunun Adımları:

- billing\_country** sütununa göre "USA" filtresi uygulanır.
- invoice\_date** alanından yılı çıkartıp, **2009** yılına ait kayıtlar seçilir.
- total** sütununun toplamı hesaplanarak **total\_bill** olarak adlandırılır.

```
SELECT
    SUM(total) as total_bill
FROM invoice
WHERE billing_country = 'USA'
AND EXTRACT(YEAR FROM invoice_date) = 2009;
```

---

## Soru 2: Track, PlaylistTrack ve Playlist Tablolarında JOIN İşlemi

Bu sorgu, **track** tablosundaki tüm parçaları, **playlisttrack** ve **playlist** tabloları ile birleştirerek listelemektedir.

### Sorgunun Adımları:

- track** tablosu (**t**) ile **playlisttrack** tablosu (**pt**) **track\_id** üzerinden birleştirilir.
- playlisttrack** tablosu (**pt**) ile **playlist** tablosu (**p**) **playlist\_id** üzerinden birleştirilir.

```
SELECT t.*
FROM track t
INNER JOIN playlisttrack pt ON pt.track_id = t.track_id
INNER JOIN playlist p ON p.playlist_id = pt.playlist_id;
```

**Alternatif Yöntem:** **USING** ile daha kısa bir yazım:

```
SELECT t.*
FROM track t
JOIN playlisttrack pt USING (track_id)
JOIN playlist p USING (playlist_id);
```

---

## Soru 3: Track, Album ve Artist Tablolarında JOIN İşlemi

Bu sorgu, "Let There Be Rock" adlı albüme ait tüm parçaları ve sanatçı bilgisini listelemektedir. Sonuçlar, parça süresine göre büyükten küçüğe sıralanmaktadır.

### Sorgunun Adımları:

1. **track** tablosu (**t**) → Şarkıları içerir.
2. **album** tablosu (**a**) → Albüm bilgilerini içerir ve **album\_id** üzerinden **track** tablosu ile ilişkilidir.
3. **artist** tablosu (**ar**) → Sanatçıyı içerir ve **artist\_id** üzerinden **album** ile ilişkilidir.
4. Albüm adı "Let There Be Rock" olan parçalar filtrelenir.
5. Sonuçlar **milliseconds** sütununa göre büyükten küçüğe sıralanır.

```
SELECT
    t.track_id,
    t.name AS track_name,
    ar.name AS artist_name,
    a.title AS album_name,
    t.milliseconds
FROM track t
INNER JOIN album a ON a.album_id = t.album_id
INNER JOIN artist ar ON ar.artist_id = a.artist_id
WHERE a.title = 'Let There Be Rock'
ORDER BY t.milliseconds DESC;
```

**Alternatif Yöntem:** USING ile daha kısa bir yazım:

```
SELECT
    t.track_id,
    t.name AS track_name,
    ar.name AS artist_name,
    a.title AS album_name,
    t.milliseconds
FROM track t
JOIN album a USING (album_id)
JOIN artist ar USING (artist_id)
WHERE a.title = 'Let There Be Rock'
ORDER BY t.milliseconds DESC;
```

---