

# Chinook Ödev Çözümleri

## Bölüm 1: Temel Sorgulama ve Filtreleme

1. Tüm sanatçıları (artists tablosu) listeleyin.

```
SELECT * FROM artists;
```

2. Tüm müşterileri (customers tablosu) şirket adına göre alfabetik olarak sıralanmış şekilde listeleyin.

```
SELECT * FROM customers ORDER BY Company;
```

3. Ülkesi 'Brazil' olan tüm müşterilerin Ad, Soyad ve Şehir bilgilerini getirin.

```
SELECT FirstName, LastName, City FROM customers WHERE Country = 'Brazil';
```

4. Sadece 'Rock' türündeki parçaların (tracks) adlarını ve bestecilerini (Composer) listeleyin.

```
SELECT t.Name, t.Composer  
FROM tracks t  
JOIN genres g ON t.GenreId = g.GenreId  
WHERE g.Name = 'Rock';
```

5. Faturası (invoices) toplam tutarı 10'dan büyük olan faturaların InvoiceId ve Total bilgilerini, en yüksek tutardan en düşüğe doğru sıralayarak getirin.

```
SELECT InvoiceId, Total  
FROM invoices  
WHERE Total > 10  
ORDER BY Total DESC;
```

6. En uzun süreli (süresi en yüksek olan) 5 parçanın (tracks) adını ve milisaniye cinsinden süresini (Milliseconds) listeleyin.``sql

```
SELECT Name, Milliseconds
FROM tracks
ORDER BY Milliseconds DESC
LIMIT 5;
```

\*\*7. Adında 'Love' kelimesi geçen tüm parçaların (tracks) adlarını listeleyin.\*\*

```
``sql
SELECT Name
FROM tracks
WHERE Name LIKE '%Love%';
```

8. Müşterilerden (customers) adresi, posta kodu (PostalCode) veya telefonu (Phone) bilgisi olmayanları (NULL olanları) listeleyin.

```
SELECT *
FROM customers
WHERE Address IS NULL OR PostalCode IS NULL OR Phone IS NULL;
```

9. Şehri Berlin, Londra veya Paris olan müşterileri (customers) listeleyin (IN kullanarak).

```
SELECT *
FROM customers
WHERE City IN ('Berlin', 'London', 'Paris');
```

10. Fatura tarihi (InvoiceDate) 2010 yılı içinde olan tüm faturaları (invoices) listeleyin.

```
SELECT *
FROM invoices
WHERE STRFTIME('%Y', InvoiceDate) = '2010';
```

---

## Bölüm 2: Tabloları Birleştirme - JOIN'ler

11. Her parçanın (tracks) adını ve ait olduğu albümün (albums) başlığını yan yana listeleyin.

```
SELECT t.Name, a.Title
FROM tracks t
JOIN albums a ON t.AlbumId = a.AlbumId;
```

12. Her faturanın (invoices) ID'sini ve o faturayı oluşturan müşterinin (customers) adını ve soyadını birlikte gösterin.

```
SELECT i.InvoiceId, c.FirstName, c.LastName
FROM invoices i
JOIN customers c ON i.CustomerId = c.CustomerId;
```

13. Her parçanın adını, ait olduğu albümün başlığını ve o albümün sanatçısının (artists) adını tek bir sorguda birleştirerek listeleyin.

```
SELECT t.Name AS TrackName, al.Title AS AlbumTitle, ar.Name AS ArtistName
FROM tracks t
JOIN albums al ON t.AlbumId = al.AlbumId
JOIN artists ar ON al.ArtistId = ar.ArtistId;
```

14. Her müşterinin yaptığı alışverişin (faturanın) tarihini ve toplam tutarını, müşteri adı ve soyadıyla birlikte gösterin.

```
SELECT c.FirstName, c.LastName, i.InvoiceDate, i.Total
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId;
```

15. "Rock" türündeki tüm parçaların adlarını ve o parçaların ait olduğu albümlerin başlıklarını listeleyin.

```
SELECT t.Name AS TrackName, a.Title AS AlbumTitle
FROM tracks t
JOIN albums a ON t.AlbumId = a.AlbumId
JOIN genres g ON t.GenreId = g.GenreId
WHERE g.Name = 'Rock';
```

16. Her bir faturadaki (invoices) ürünlerin (invoice\_items) detaylarını (fatura ID, parça adı, birim fiyat, adet) gösteren bir liste oluşturun.

```
SELECT ii.InvoiceId, t.Name, ii.UnitPrice, ii.Quantity
FROM invoice_items ii
JOIN tracks t ON ii.TrackId = t.TrackId;
```

17. Hiç albümü olmayan sanatçıları (artists) listeleyin.

```
SELECT ar.*
FROM artists ar
LEFT JOIN albums al ON ar.ArtistId = al.ArtistId
WHERE al.AlbumId IS NULL;
```

18. Müşterilerin (customers) her birine destek olan çalışanların (employees) adını ve soyadını, müşteri adı ve soyadı ile birlikte listeleyin.

```
SELECT c.FirstName AS CustomerFirstName, c.LastName AS CustomerLastName,
e.FirstName AS EmployeeFirstName, e.LastName AS EmployeeLastName
FROM customers c
JOIN employees e ON c.SupportRepId = e.EmployeeId;
```

19. ABD'deki (USA) müşterilerin yaptığı tüm alışverişlerin fatura ID'lerini ve tarihlerini listeleyin.

```
SELECT i.InvoiceId, i.InvoiceDate
FROM invoices i
JOIN customers c ON i.CustomerId = c.CustomerId
WHERE c.Country = 'USA';
```

20. Hiçbir faturada satılmamış (yani invoice\_items tablosunda hiç yer almayan) parçaları (tracks) listeleyin.

```
SELECT t.*
FROM tracks t
LEFT JOIN invoice_items ii ON t.TrackId = ii.TrackId
WHERE ii.InvoiceLineId IS NULL;
```

---

### Bölüm 3: Gruplama, Agregasyon ve Alt Sorgular

21. Her türün (genres) toplam kaç parçaya (tracks) sahip olduğunu, parça sayısına göre çoktan aza doğru sıralayarak listeleyin.``sql

```
SELECT g.Name, COUNT(t.TrackId) AS NumberOfTracks
FROM genres g
JOIN tracks t ON g.GenreId = t.GenreId
GROUP BY g.Name
ORDER BY NumberOfTracks DESC;
```

\*\*22. Her sanatçının (artists) toplam kaç albümü olduğunu listeleyin ve en çok albüme sahip ilk 3 sanatçıyı gösterin.\*\*

```
``sql
SELECT ar.Name, COUNT(al.AlbumId) AS NumberOfAlbums
FROM artists ar
JOIN albums al ON ar.ArtistId = al.ArtistId
GROUP BY ar.Name
ORDER BY NumberOfAlbums DESC
LIMIT 3;
```

23. Her ülkedeki (customers) toplam müşteri sayısını bulun.

```
SELECT Country, COUNT(CustomerId) AS NumberOfCustomers
FROM customers
GROUP BY Country;
```

24. Her bir müşterinin toplam ne kadar harcama yaptığını (invoices tablosundaki Total sütununu toplayarak), en çok harcama yapandan en aza doğru sıralayarak listeleyin.

```
SELECT c.FirstName, c.LastName, SUM(i.Total) AS TotalSpending
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY TotalSpending DESC;
```

25. Her bir ülkenin toplam ne kadar ciro getirdiğini hesaplayın.

```
SELECT c.Country, SUM(i.Total) AS TotalRevenue
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.Country
ORDER BY TotalRevenue DESC;
```

26. 10'dan fazla albümü olan sanatçıları (artists) listeleyin.

```
SELECT ar.Name, COUNT(al.AlbumId) AS AlbumCount
FROM artists ar
JOIN albums al ON ar.ArtistId = al.ArtistId
GROUP BY ar.ArtistId
HAVING AlbumCount > 10;
```

27. Toplam süresi (milisaniye cinsinden) 1 saatten (3600000 milisaniye) daha uzun olan çalma listelerini (playlists) listeleyin.

```
SELECT p.Name, SUM(t.Milliseconds) AS TotalDuration
FROM playlists p
JOIN playlist_track pt ON p.PlaylistId = pt.PlaylistId
JOIN tracks t ON pt.TrackId = t.TrackId
GROUP BY p.PlaylistId
HAVING TotalDuration > 3600000;
```

28. Ortalama parça boyutundan (ortalama Bytes değeri) daha büyük boyuta sahip olan parçaları listeleysin.

```
SELECT Name, Bytes
FROM tracks
WHERE Bytes > (SELECT AVG(Bytes) FROM tracks);
```

29. “Queen” adlı sanatçının tüm parçalarını satın alan müşterilerin ad ve soyadlarını listeleysin.

```
SELECT c.FirstName, c.LastName
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
JOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId
JOIN tracks t ON ii.TrackId = t.TrackId
JOIN albums al ON t.AlbumId = al.AlbumId
JOIN artists ar ON al.ArtistId = ar.ArtistId
WHERE ar.Name = 'Queen'
GROUP BY c.CustomerId
HAVING COUNT(DISTINCT t.TrackId) = (
    SELECT COUNT(t2.TrackId)
    FROM tracks t2
    JOIN albums al2 ON t2.AlbumId = al2.AlbumId
    JOIN artists ar2 ON al2.ArtistId = ar2.ArtistId
    WHERE ar2.Name = 'Queen'
);
```

30. (Bonus) Her müzik türünün (genres) en çok satılan parçasını (tracks) bulun.

```
WITH TrackSales AS (
    SELECT
        TrackId,
        COUNT(InvoiceLineId) AS SalesCount
    FROM invoice_items
    GROUP BY TrackId
),
GenreTrackSales AS (
    SELECT
        g.Name AS GenreName,
        t.Name AS TrackName,
```

```
        ts.SalesCount,  
        RANK() OVER (PARTITION BY g.Name ORDER BY ts.SalesCount DESC) as  
RankNum  
    FROM TrackSales ts  
    JOIN tracks t ON ts.TrackId = t.TrackId  
    JOIN genres g ON t.GenreId = g.GenreId  
)  
SELECT  
    GenreName,  
    TrackName,  
    SalesCount  
FROM GenreTrackSales  
  
WHERE RankNum = 1;
```