

## Ödev-3 Çözümleri

**Soru** Her ülke (billing\_country) için kaç adet müşteri (customer\_id) olduğunu listeleyen bir sorgu yazınız. Sonuca ülke adı ve müşteri sayısı görünşün. Müşteri sayısına göre büyükten küçüğe sıralayınız.

### Çözüm

müşteri tablosunda ülke sütununu seçtim. count fonksiyonu ile müşteri id'lerini saydım ve bunları "customer\_count" isimli sütunda gösterdim. Group by ile ülkelere göre gruplandırdım, topladım ve order by ile ülkelerdeki müşteri sayılarını azalan sırada olacak şekilde tablomda sıraladım.

The screenshot shows the pgAdmin 4 interface. At the top, it displays the connection information: postgres/postgres@ödev1. Below the connection bar is a toolbar with various icons for database management. The main area is divided into two tabs: 'Query' (which is selected) and 'Query History'. The 'Query' tab contains the following SQL code:

```
1 ▾ SELECT
2     country,
3         COUNT(customer_id) AS customer_count
4 FROM Customer
5 GROUP BY country
6 ORDER BY customer_count DESC;
7
```

Below the code, the 'Data Output' tab is selected, showing the results of the query:

	country	customer_count
1	USA	13
2	Canada	8
3	France	5
4	Brazil	5
5	Germany	4
6	United Kingdom	3
7	Portugal	2
8	India	2
9	Czech Republic	2

At the bottom of the results pane, it says 'Showing rows: 1 to 9 of 24'. The status bar at the bottom of the window indicates 'Total rows: 24' and 'Query complete 00:00:00.077'.

**Soru** Her sanatçının (artist) toplam kaç albümü (album) olduğunu listeleyen bir sorgu yazınız. Sanatçı adı ve albüm sayısı görünüsün. Sonuçları albüm sayısına göre azalan şekilde sıralayınız.

### Çözüm

inner join kullanarak sadece albümü olan sanatçıları listeledim. Artist tablosuna “ar” takma adını verdim ve ar.name ile sanatçı isimlerini gösterdim. Count fonksiyonu ile albüm tablosundaki album\_id'leri saydım. Null değerleri saymaz çünkü album\_id NOT NULL alındır bu sayede sanatçının kaç albümü olduğunu bize verir. join kısmında album tablosundaki artist\_id değerleri ile artist tablosundaki artist\_id değerlerini eşleştirdim. Bu join sayesinde albümü olmayan sanatçılar tablomda yer almayacak. Order by ile azalan albüm sayısına göre sıraladım en çok albümü olan sanatçı en üstte yer alacak.

The screenshot shows the pgAdmin 4 interface. At the top, there's a toolbar with various icons for database management. Below it is a "Query History" section where a query has been run. The query itself is:

```
1 ✓ SELECT
2     ar.name AS artist_name,
3     COUNT(al.album_id) AS album_count
4 FROM artist AS ar
5 JOIN album AS al ON al.artist_id = ar.artist_id
6 GROUP BY ar.artist_id, ar.name
7 ORDER BY album_count DESC, artist_name ASC;
```

Below the query history, there are tabs for "Data Output", "Messages", and "Notifications". The "Data Output" tab is selected and displays the results of the query in a table format. The table has two columns: "artist\_name" and "album\_count". The data is as follows:

	artist_name	album_count
1	Iron Maiden	21
2	Led Zeppelin	14
3	Deep Purple	11
4	Metallica	10
5	U2	10
6	Ozzy Osbourne	6
7	Pearl Jam	5
8	Faith No More	4
9	Foo Fighters	4

At the bottom of the results pane, it says "Total rows: 204" and "Query complete 00:00:00.178".

**Soru** En uzun süreye (milliseconds) sahip ilk 10 parçayı (track), albüm (album) ve sanatçı (artist) bilgisi ile birlikte listeleyen bir sorgu yazınız.

### Çözüm

track tablosunda istediğim alanları seçtim. Albüm tablosundaki album\_id alanı ile track tablosundaki album\_id kısımlarını eşleştirdim bu sayede hangi parçanın hangi hangi albüme ait olduğu belli oldu.

artist tablosundaki artist\_id ile album tablosundaki artist\_id kısmını da eşleştirdim. Bu sayede parça albüm sanatçı alanım birleşmiş oldu. Order by ile süresine göre azalan sıraya koydum ve limit ifadesiyle en uzun süreye sahip 10 parçayı getirmiş oldum.

The screenshot shows the pgAdmin 4 interface. At the top, there's a toolbar with various icons for file operations, search, and database management. Below the toolbar, the title bar displays 'postgres/postgres@ödev1'. The main area is divided into two sections: 'Query' and 'Query History'. The 'Query' section contains the following SQL code:

```
1 SELECT
2     t.name      AS track_name,
3     t(milliseconds,
4     al.title    AS album_title,
5     ar.name     AS artist_name
6 FROM track AS t
7 JOIN album AS al ON t.album_id = al.album_id
8 JOIN artist AS ar ON al.artist_id = ar.artist_id
9 ORDER BY t(milliseconds) DESC
10 LIMIT 10;
11
```

Below the code, the 'Data Output' tab is selected, showing the results of the query in a grid format. The columns are labeled: track\_name, milliseconds, album\_title, and artist\_name. The data consists of 10 rows, each representing a track from the 'Battlestar Galactica' series. The 'milliseconds' column shows values ranging from 2927677 to 5286953.

	track_name	milliseconds	album_title	artist_name
1	Occupation / Precipice	5286953	Battlestar Galactica, Season 3	Battlestar Galactica
2	Through a Looking Glass	5088838	Lost, Season 3	Lost
3	Greetings from Earth, Pt...	2960293	Battlestar Galactica (Classic), Season 1	Battlestar Galactica (Classic)
4	The Man With Nine Lives	2956998	Battlestar Galactica (Classic), Season 1	Battlestar Galactica (Classic)
5	Battlestar Galactica, Pt. 2	2956081	Battlestar Galactica (Classic), Season 1	Battlestar Galactica (Classic)
6	Battlestar Galactica, Pt. 1	2952702	Battlestar Galactica (Classic), Season 1	Battlestar Galactica (Classic)
7	Murder On the Rising Star	2935894	Battlestar Galactica (Classic), Season 1	Battlestar Galactica (Classic)
8	Battlestar Galactica, Pt. 3	2927802	Battlestar Galactica (Classic), Season 1	Battlestar Galactica (Classic)
9	Take the Celestra	2927677	Battlestar Galactica (Classic), Season 1	Battlestar Galactica (Classic)

Total rows: 10 | Query complete 00:00:00.131