

Ödev-2 Çözümleri

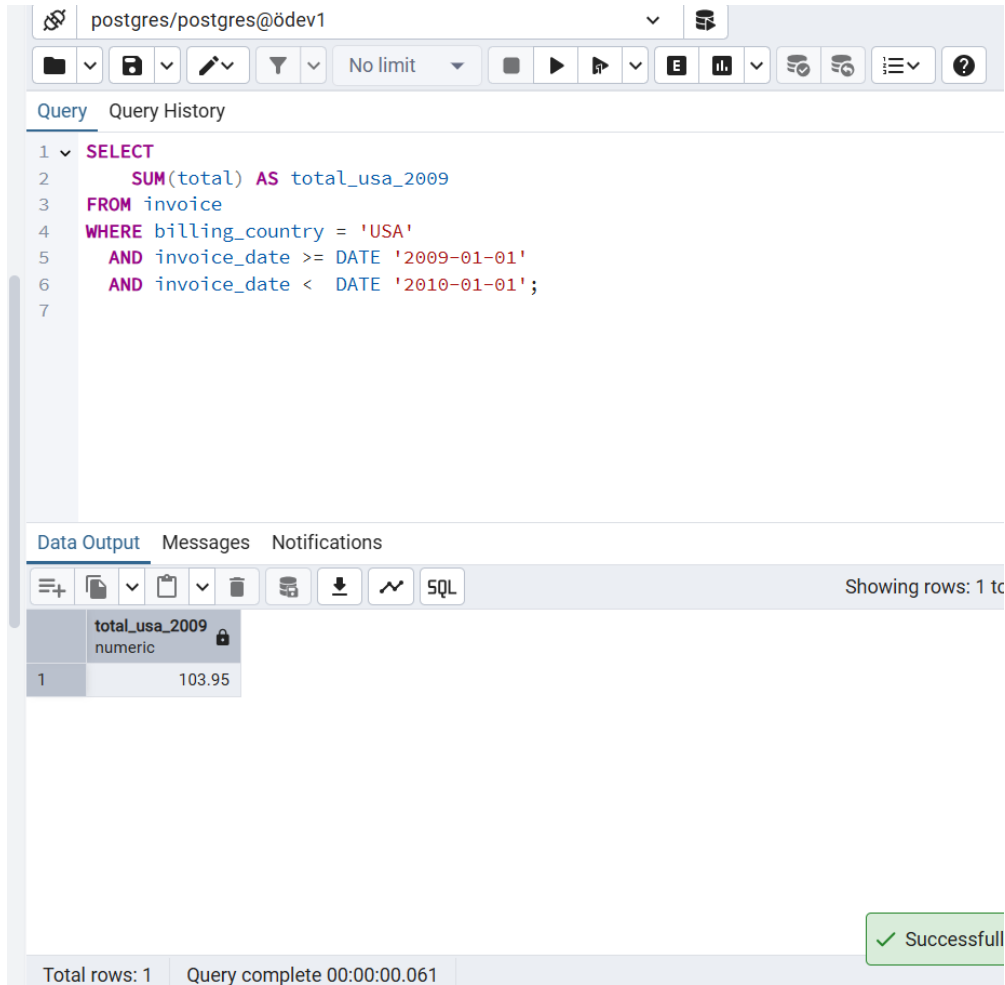
Soru “USA” ülkesine ait, 2009 yılı içerisinde oluşturulmuş tüm faturaların (Invoice) toplamını listeleyen bir sorgu yazınız.

Çözüm

invoice tablosunda oluşturulmuş faturaların toplamını “sum” fonksiyonu ile topladım. “where” ile ülkeyi filtreleyen bir koşul yazdım.

“and” operatörleri ile istediğim tarih aralığını filtreledim.

burada date kullanmamın sebebi tür dönüşümünde bir sıkıntı çıkmaması içindi. Tarih aralığının string olarak değil date olarak gelmesini istiyorum.



The screenshot shows a PostgreSQL query editor interface. The query is as follows:

```
1 SELECT
2     SUM(total) AS total_usa_2009
3 FROM invoice
4 WHERE billing_country = 'USA'
5     AND invoice_date >= DATE '2009-01-01'
6     AND invoice_date < DATE '2010-01-01';
7
```

The results are displayed in a table with the following structure:

	total_usa_2009
1	103.95

The interface also shows a status bar at the bottom indicating "Total rows: 1" and "Query complete 00:00:00.061". A green success message "✓ Successfull" is visible in the bottom right corner.

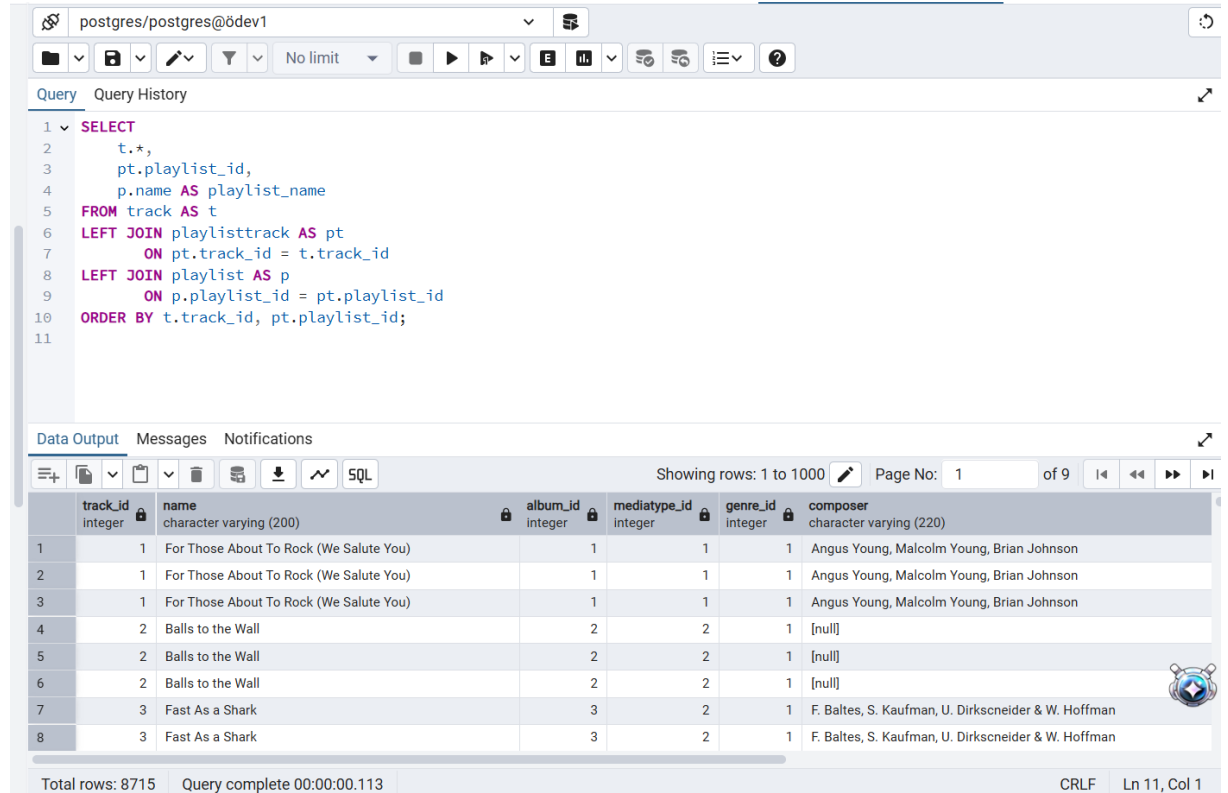
Soru Tüm parça (track) bilgilerini, bu parçaların ait olduğu playlisttrack ve playlist tablolarıyla birleştirerek (JOIN) listeleyen bir sorgu yazınız.

Çözüm

tüm track parçalarının gelmesini istediğim için left join yaptım eğer inner join yazsaydım playlistte olmayan trackler gelmezdi.

burada ana tablom track tablosu ve buna “t” adında takma ad verdim.

“left join playlisttrack as pt on pt.track_id=t.track_id” burada left join yapıyorum. t ‘deki track_id ile pt’deki track_id’leri eşleştiriyorum. Hiç eşleşme olmayan satırlar olursa yani ptde olmayan track olursa pt’nin satırı null döner fakat t olduğu gibi gelir kaybolmaz çünkü left join. Eğer bir parça birden çok playlistte varsa her eşleşme için satır çoğalır. İkinci left join ile playlist tablosunu pt ile sol birleştirme yaptım. eğer ilk aşamada pt.playlist_id NULL ise yani parça hiçbir playlistte kayıtlı değilse herhangi bir eşleşme olmayacağı için p’den gelen sütunlar null döner. Select t.* ifadesi track tablosundaki tüm sütunları seçmemi sağlar. Order by kısmında her şeyi t.track_id’e göre sıraya koyar ve pt.playlist_id’e göre artan sıraya dizer.



The screenshot shows a PostgreSQL query editor interface. The query is as follows:

```
1 SELECT
2   t.*,
3   pt.playlist_id,
4   p.name AS playlist_name
5 FROM track AS t
6 LEFT JOIN playlisttrack AS pt
7   ON pt.track_id = t.track_id
8 LEFT JOIN playlist AS p
9   ON p.playlist_id = pt.playlist_id
10 ORDER BY t.track_id, pt.playlist_id;
11
```

The results are displayed in a table with the following columns: track_id, name, album_id, mediatype_id, genre_id, and composer. The table shows 8 rows of data.

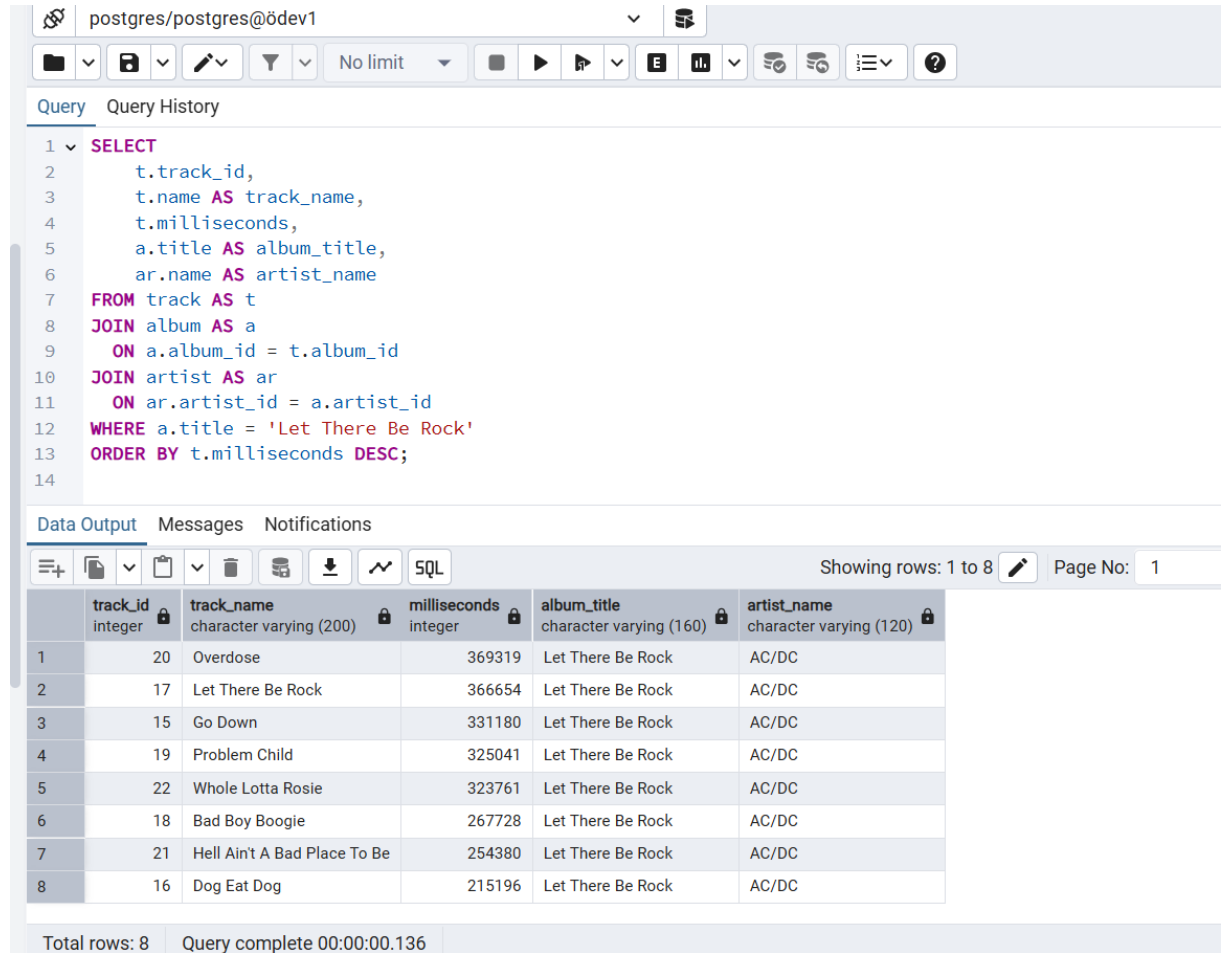
track_id	name	album_id	mediatype_id	genre_id	composer
1	For Those About To Rock (We Salute You)	1	1	1	Angus Young, Malcolm Young, Brian Johnson
2	For Those About To Rock (We Salute You)	1	1	1	Angus Young, Malcolm Young, Brian Johnson
3	For Those About To Rock (We Salute You)	1	1	1	Angus Young, Malcolm Young, Brian Johnson
4	Balls to the Wall	2	2	1	[null]
5	Balls to the Wall	2	2	1	[null]
6	Balls to the Wall	2	2	1	[null]
7	Fast As a Shark	3	2	1	F. Baltes, S. Kaufman, U. Dirksneider & W. Hoffman
8	Fast As a Shark	3	2	1	F. Baltes, S. Kaufman, U. Dirksneider & W. Hoffman

Total rows: 8715 Query complete 00:00:00.113 CRLF Ln 11, Col 1

Soru "Let There Be Rock" adlı albüme ait tüm parçaları (Track) listeleyen, sanatçı (Artist) bilgisini de içeren bir sorgu yazınız. Ayrıca, sonuçları parça süresi (milliseconds) büyükten küçüğe sıralayınız.

Çözüm

select ile istediğim sütunları seçtim. From ile track tablomun ana tablom olduğunu belirttim ve "t" takma adını verdim. Join kısmında track tablosundaki albüm_id ile albüm tablosundaki albüm_id eşleştirdim. Sadece bir parçanın album_id değeri album.album_id içinde bulunuyorsa o parça sonuçta yer alacak. Yani t.album_id null ise veya albüm tablosunda o albüm_id yoksa o parça sonuçta gösterilmeyecek. İkinci join ile albüm tablosundaki satırlar ile artist_id'leri eşleştirdim."where" ile istediğim albümün başlığını belirttim. Order by ile parçanın süresine göre azalan bir sıralama yaptım.



The screenshot shows a PostgreSQL query editor interface. The query is as follows:

```
1 SELECT
2     t.track_id,
3     t.name AS track_name,
4     t.milliseconds,
5     a.title AS album_title,
6     ar.name AS artist_name
7 FROM track AS t
8 JOIN album AS a
9     ON a.album_id = t.album_id
10 JOIN artist AS ar
11     ON ar.artist_id = a.artist_id
12 WHERE a.title = 'Let There Be Rock'
13 ORDER BY t.milliseconds DESC;
14
```

The results are displayed in a table with the following columns: track_id, track_name, milliseconds, album_title, and artist_name. The results are sorted by milliseconds in descending order.

track_id	track_name	milliseconds	album_title	artist_name
20	Overdose	369319	Let There Be Rock	AC/DC
17	Let There Be Rock	366654	Let There Be Rock	AC/DC
15	Go Down	331180	Let There Be Rock	AC/DC
19	Problem Child	325041	Let There Be Rock	AC/DC
22	Whole Lotta Rosie	323761	Let There Be Rock	AC/DC
18	Bad Boy Boogie	267728	Let There Be Rock	AC/DC
21	Hell Ain't A Bad Place To Be	254380	Let There Be Rock	AC/DC
16	Dog Eat Dog	215196	Let There Be Rock	AC/DC

Total rows: 8 Query complete 00:00:00.136