

Ödev-2 Çözümleri

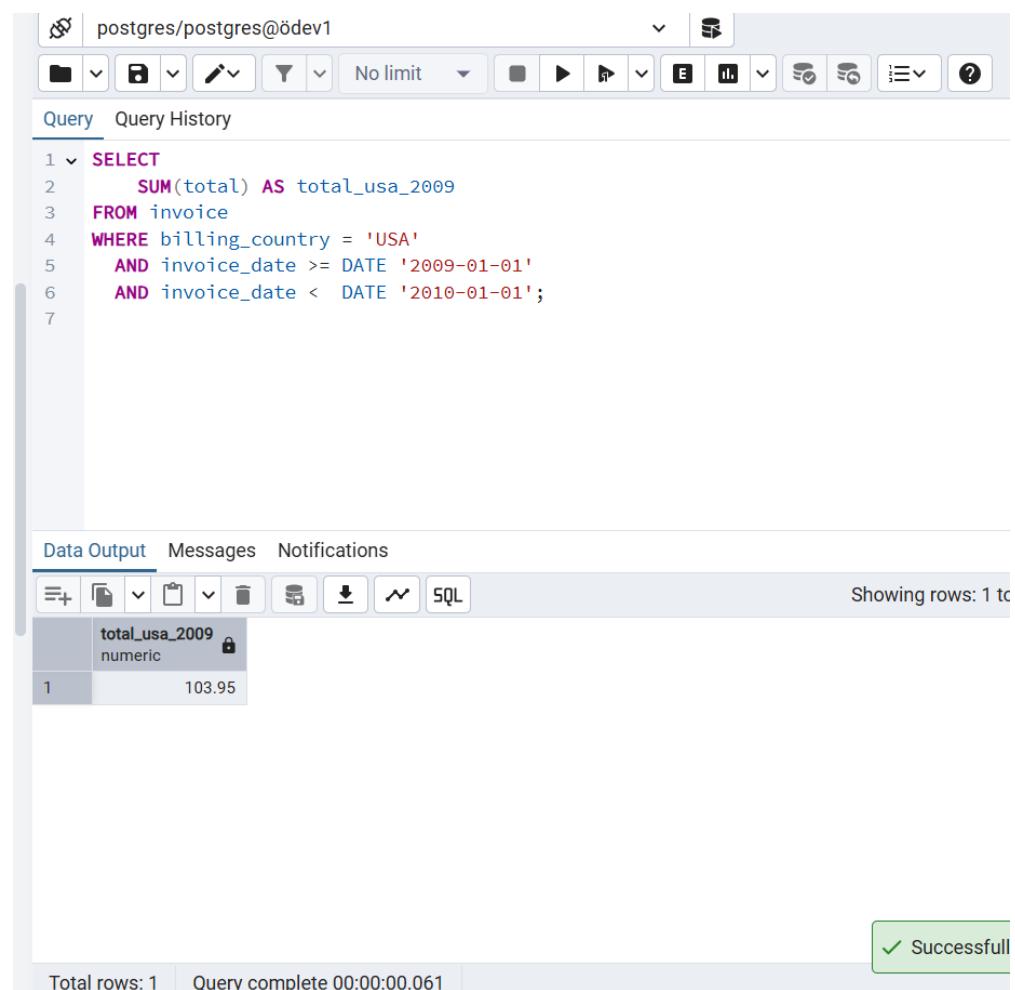
Soru “USA” ülkesine ait, 2009 yılı içerisinde oluşturulmuş tüm faturaların (Invoice) toplamını listeleyen bir sorgu yazınız.

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 kullanmanın sebebi tür dönüşümünde bir sıkıntısı çıkmaması içindi. Tarih aralığının string olarak değil date olarak gelmesini istiyorum.



The screenshot shows the pgAdmin 4 interface with a successful SQL query execution. The query retrieves the total value for invoices from the USA in 2009. The result is a single row with one column named 'total_usa_2009' containing the value '103.95'. The status bar at the bottom right indicates 'Successful'.

```
postgres/postgres@ödev1
No limit
SELECT
    SUM(total) AS total_usa_2009
FROM invoice
WHERE billing_country = 'USA'
    AND invoice_date >= DATE '2009-01-01'
    AND invoice_date < DATE '2010-01-01';
```

total_usa_2009
103.95

Total rows: 1 | Query complete 00:00:00.061 | ✓ Successful

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 yapsaydım 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 the pgAdmin 4 interface with a query editor and a data output viewer.

Query Editor:

```
postgres/postgres@ödev1
No limit
SELECT
    t.*,
    pt.playlist_id,
    p.name AS playlist_name
FROM track AS t
LEFT JOIN playlisttrack AS pt
    ON pt.track_id = t.track_id
LEFT JOIN playlist AS p
    ON p.playlist_id = pt.playlist_id
ORDER BY t.track_id, pt.playlist_id;
```

Data Output:

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 istedigim sütunları seçtim. From ile track tablomun ana tablom olduğunu belirttim ve "t" takma adını verdim. Join kısmında track tablosundaki album_id ile albüm tablsoundaki albüm_id eşleştirildim. 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 eşleştirildim."where" ile istedigim albümün başlığını belirttim. Order by ile parçanın süresine göre azalan bir sıralama yaptım.

The screenshot shows the pgAdmin 4 interface with a query editor and a data output grid.

Query History:

```
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;
```

Data Output:

	track_id	track_name	milliseconds	album_title	artist_name
1	20	Overdose	369319	Let There Be Rock	AC/DC
2	17	Let There Be Rock	366654	Let There Be Rock	AC/DC
3	15	Go Down	331180	Let There Be Rock	AC/DC
4	19	Problem Child	325041	Let There Be Rock	AC/DC
5	22	Whole Lotta Rosie	323761	Let There Be Rock	AC/DC
6	18	Bad Boy Boogie	267728	Let There Be Rock	AC/DC
7	21	Hell Ain't A Bad Place To Be	254380	Let There Be Rock	AC/DC
8	16	Dog Eat Dog	215196	Let There Be Rock	AC/DC

Total rows: 8 Query complete 00:00:00.136