

Soru Her albüm (album) için parçaların (track) ortalama süresini (milliseconds) hesaplayınız.

avg fonksiyonu albümlerdeki parçaların sürelerinin ortalamasını alır. Burada ana tablom albüm tablosu bunu from ile belirttim. Join ile track ve albüm tablolarındaki alanları bağladım. Group by ile albüm tablosunun alanları ile gruplama yaptım. order by ile ortalama süreleri azalan şekilde sıraladım.

postgres/postgres@ödev1

Query Query History

```

1 SELECT
2     al.title AS album_title,
3     AVG(t.milliseconds) AS avg_duration
4 FROM album al
5 JOIN track t ON t.album_id = al.album_id
6 GROUP BY al.album_id, al.title
7 ORDER BY avg_duration DESC;
8

```

Data Output Messages Notifications

Showing rows: 1 to 347 Page No: 1 of 1

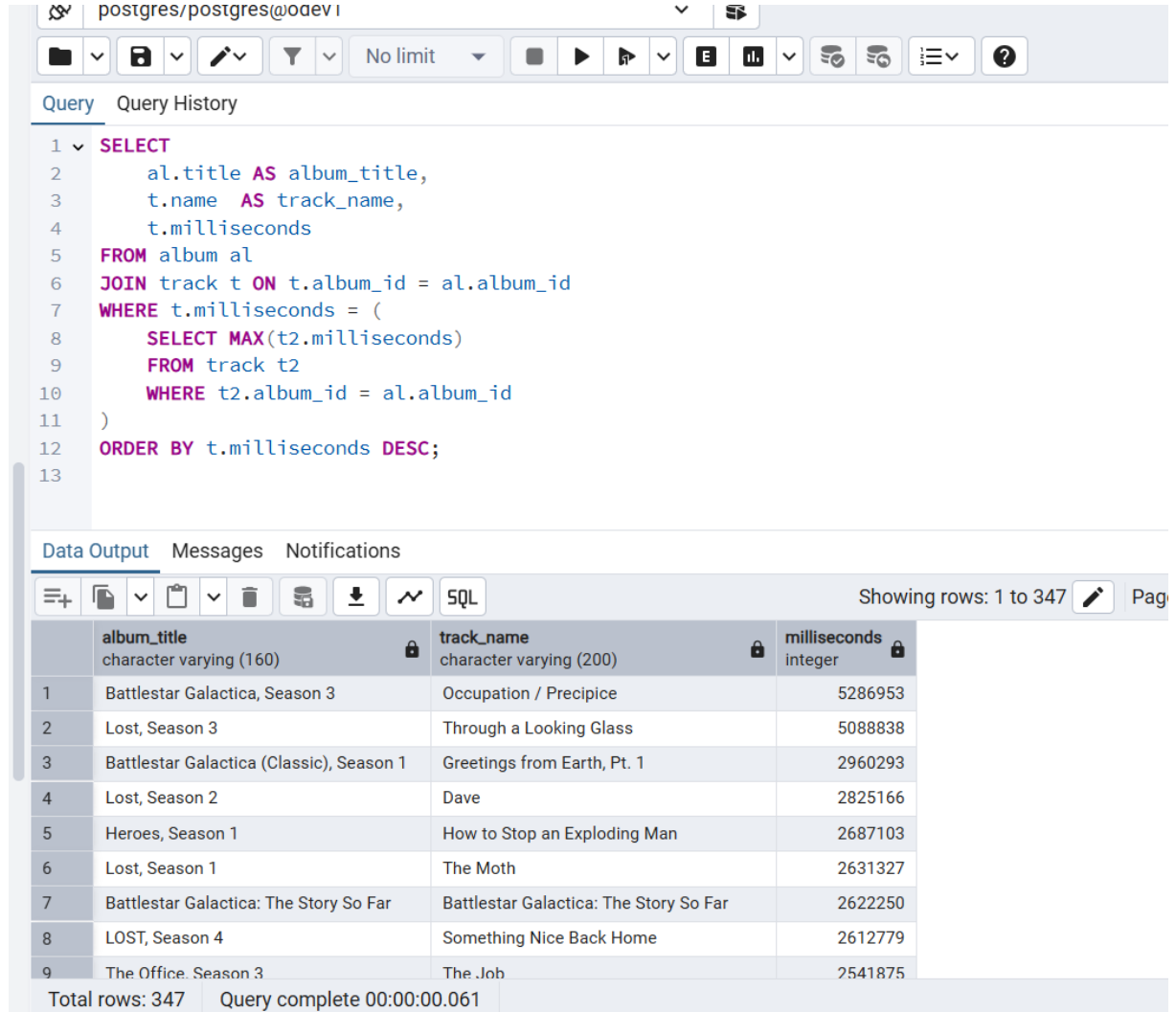
	album_title character varying (160)	avg_duration numeric
1	Battlestar Galactica (Classic), Season 1	2925574.3333333333
2	Battlestar Galactica, Season 3	2778265.315789473684
3	Lost, Season 3	2717907.0000000000
4	Lost, Season 2	2637067.9583333333
5	Battlestar Galactica: The Story So Far	2622250.0000000000
6	Heroes, Season 1	2599142.086956521739
7	Lost, Season 1	2594197.4400000000
8	Aquaman	2484567.0000000000
9	LOST: Season 4	2321672.529411764706

Total rows: 347 Query complete 00:00:00.072

Soru Her albüm (album) için en uzun süreye (milliseconds) sahip parçayı (track) bulunuz.

Çözüm

albüm tablosu ana tablom olacak bunu from ile belirttim. Track ve albüm tablolarındaki alanları eşleştirdim bu sayede eşleşen parçaları ve albümleri görebileceğim. Where koşulu ile her albümdeki en uzun süreye sahip parçayı bulan koşulu yazdım. Burada track tablosunu seçtim ve tekrar bir where koşulu yazarak track ile albüm tablolarındaki alanları eşleştirdim ve max fonksiyonu kullandım bu sayede her albüm için en uzun süreye sahip parçam gelecek. Order by ile bunları süreleri azalan şekilde sıraladım.



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

```
1 SELECT
2     al.title AS album_title,
3     t.name AS track_name,
4     t.milliseconds
5 FROM album al
6 JOIN track t ON t.album_id = al.album_id
7 WHERE t.milliseconds = (
8     SELECT MAX(t2.milliseconds)
9     FROM track t2
10    WHERE t2.album_id = al.album_id
11 )
12 ORDER BY t.milliseconds DESC;
```

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

	album_title	track_name	milliseconds
1	Battlestar Galactica, Season 3	Occupation / Precipice	5286953
2	Lost, Season 3	Through a Looking Glass	5088838
3	Battlestar Galactica (Classic), Season 1	Greetings from Earth, Pt. 1	2960293
4	Lost, Season 2	Dave	2825166
5	Heroes, Season 1	How to Stop an Exploding Man	2687103
6	Lost, Season 1	The Moth	2631327
7	Battlestar Galactica: The Story So Far	Battlestar Galactica: The Story So Far	2622250
8	LOST, Season 4	Something Nice Back Home	2612779
9	The Office. Season 3	The Job	2541875

Total rows: 347 Query complete 00:00:00.061

Soru Fiyatı (unit_price) en yüksek 5 parçayı (track), albüm (album) ve sanatçı (artist) bilgisi ile birlikte listeleyiniz.

Çözüm

select ile track tablosunda hangi sütunları istediğimi seçtim ve as ile bunları isimlerini verdim. hangi parçanın hangi albümde olduğunu öğrenmek için bir join sorgusu yazdım track ve albüm tablolarındaki id'leri eşleştirdim.

albümden sanatçıyı almak için tekrar bir join sorgusu yazdım ve albüm ile artist tablolarındaki id'leri birleştirdim. Bu şekilde ekrana gelen sorguda her parçanın albüm ve sanatçı bilgisini görebileceğim.

order by ile unit_price değerini büyükten küçüğe sıraladım. Limit ifadesi ile de en pahalı parçayı döndürdüm.

postgres/postgres@ödev1

Query Query History

```

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

```

Data Output Messages Notifications

Showing rows: 1 to 5 Page No: 1

	track_name character varying (200)	unitprice numeric (10,2)	album_title character varying (160)	artist_name character varying (120)
1	"?"	1.99	Lost, Season 2	Lost
2	...And Found	1.99	Lost, Season 2	Lost
3	...In Translation	1.99	Lost, Season 1	Lost
4	.07%	1.99	Heroes, Season 1	Heroes
5	A Benihana Christmas, Pts. 1 & 2	1.99	The Office, Season 3	The Office

Total rows: 5 Query complete 00:00:00.054