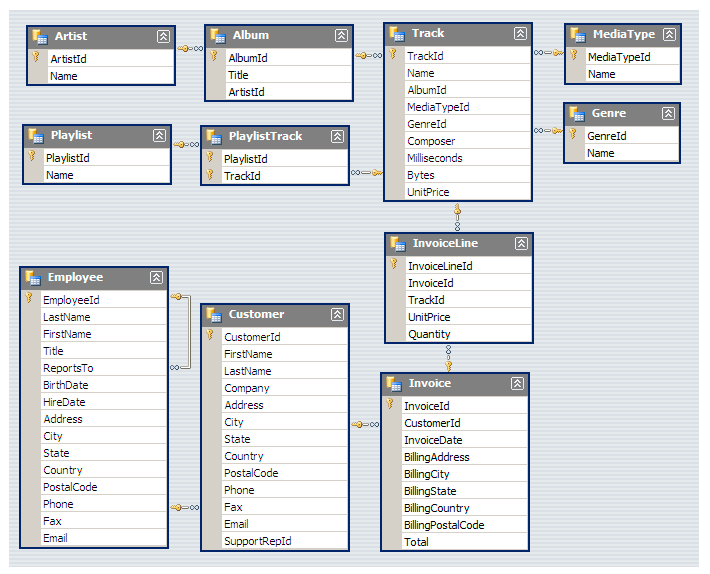
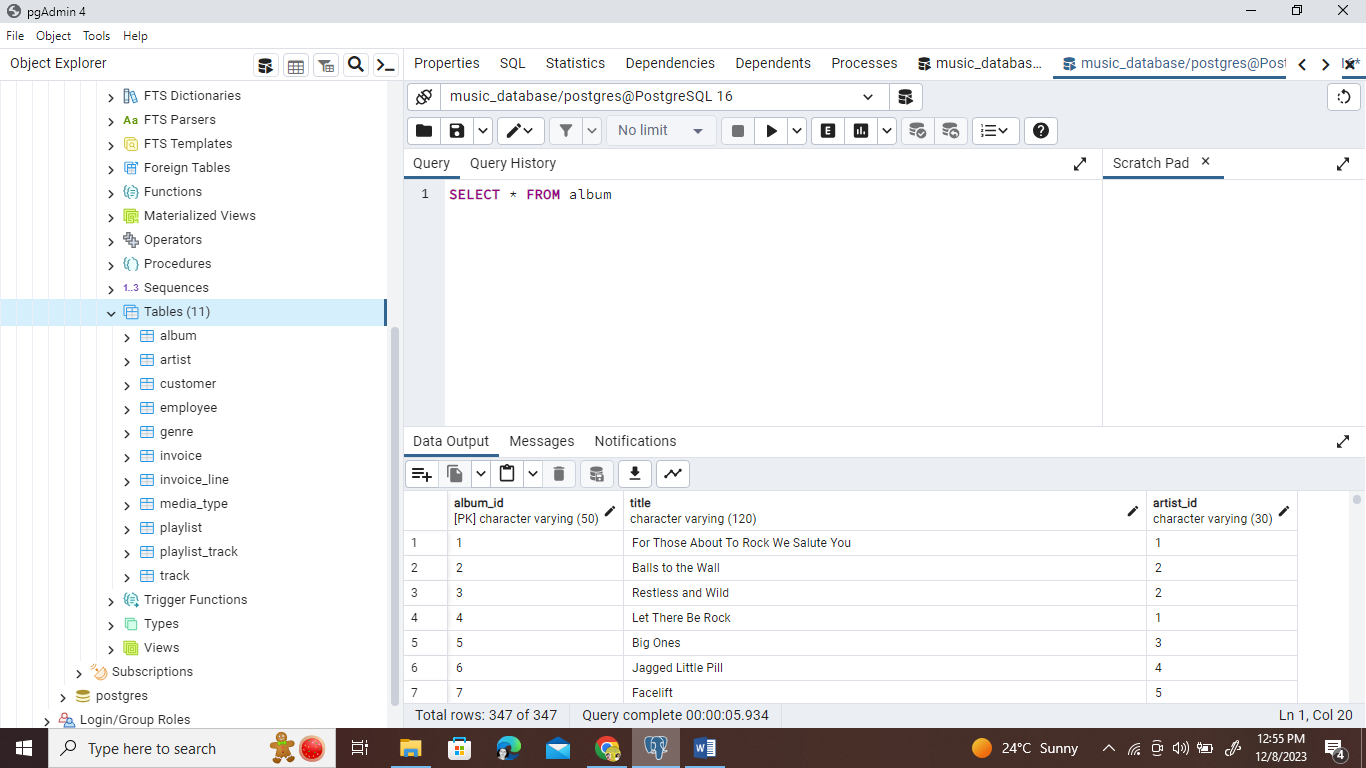
**Digital Music Store Analysis Using SQL**

**Objective-** This project will deal about how to analyze the music playlist database. You can examine the dataset with SQL and help the store understand its business growth by answering some related questions.

**Schema- **

**Import Data-** We have imported data on our Postgre SQL. We followed-

1. Downloaded the required music database from web.
2. Then created a database in Postgre SQL and named it as music\_database
3. And then we have imported the SQL file to our Postgre SQL database.

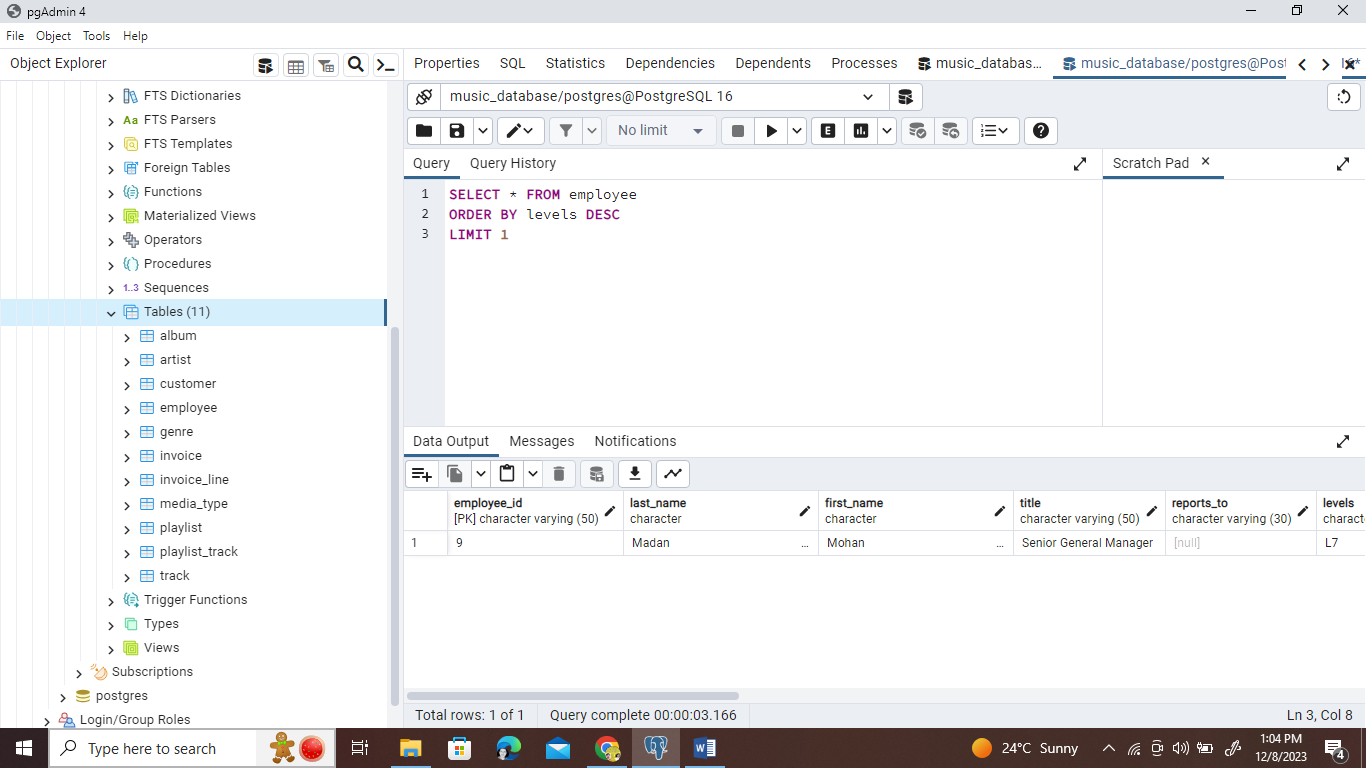


**Questions Set 1-[Easy]-**

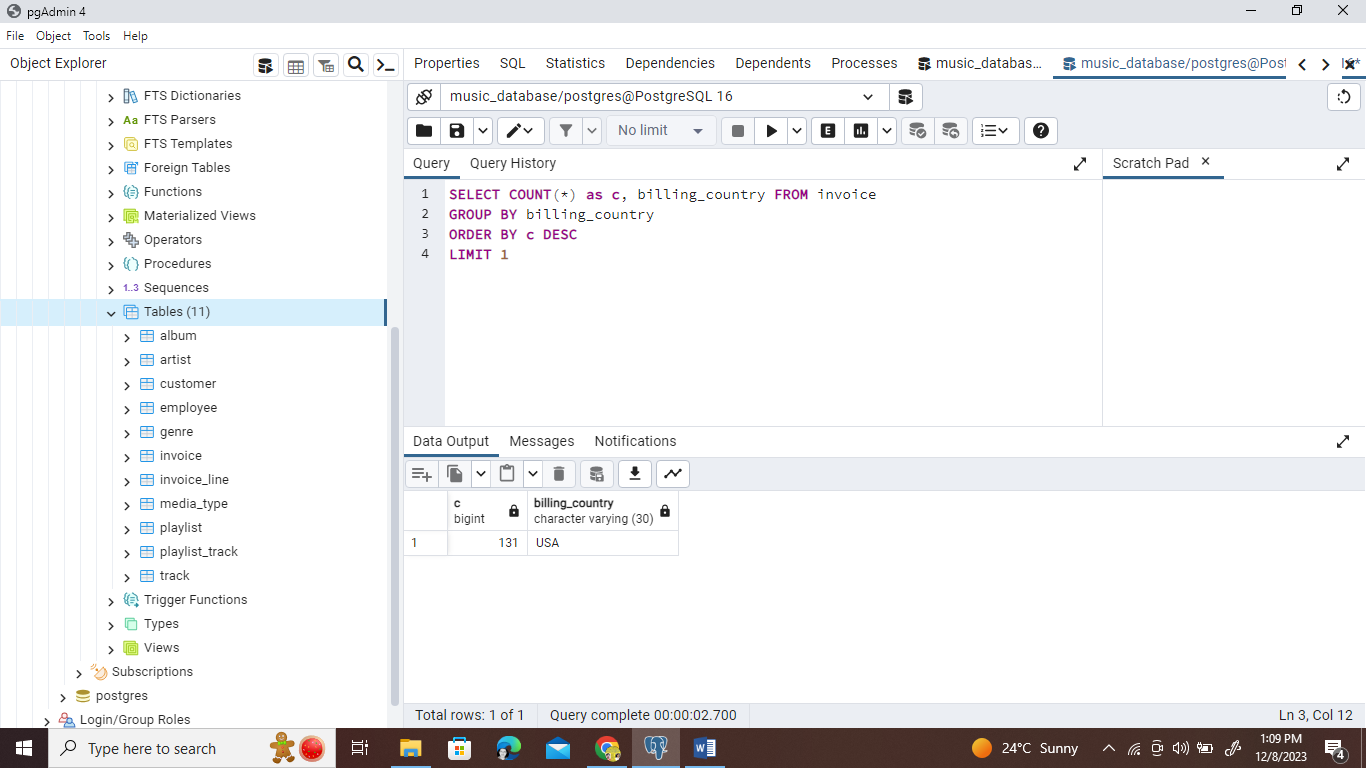
1. Who is the senior most employee based on job title?
2. Which countries have the most Invoices?
3. What are top 3 values of total invoice?
4. Which city has the best customers? We would like to throw a promotional Music Festival in the city we made the most money. Write a query that returns one city that has the highest sum of invoice totals. Return both the city name & sum of all invoice total.
5. Who is the best customer? The customer who has spent the most money will be declared the best customer. Write a query that returns the person who has spent the most money.

**Solutions Set 1-[Easy]-**

1. Madan Mohan is the senior most employee.



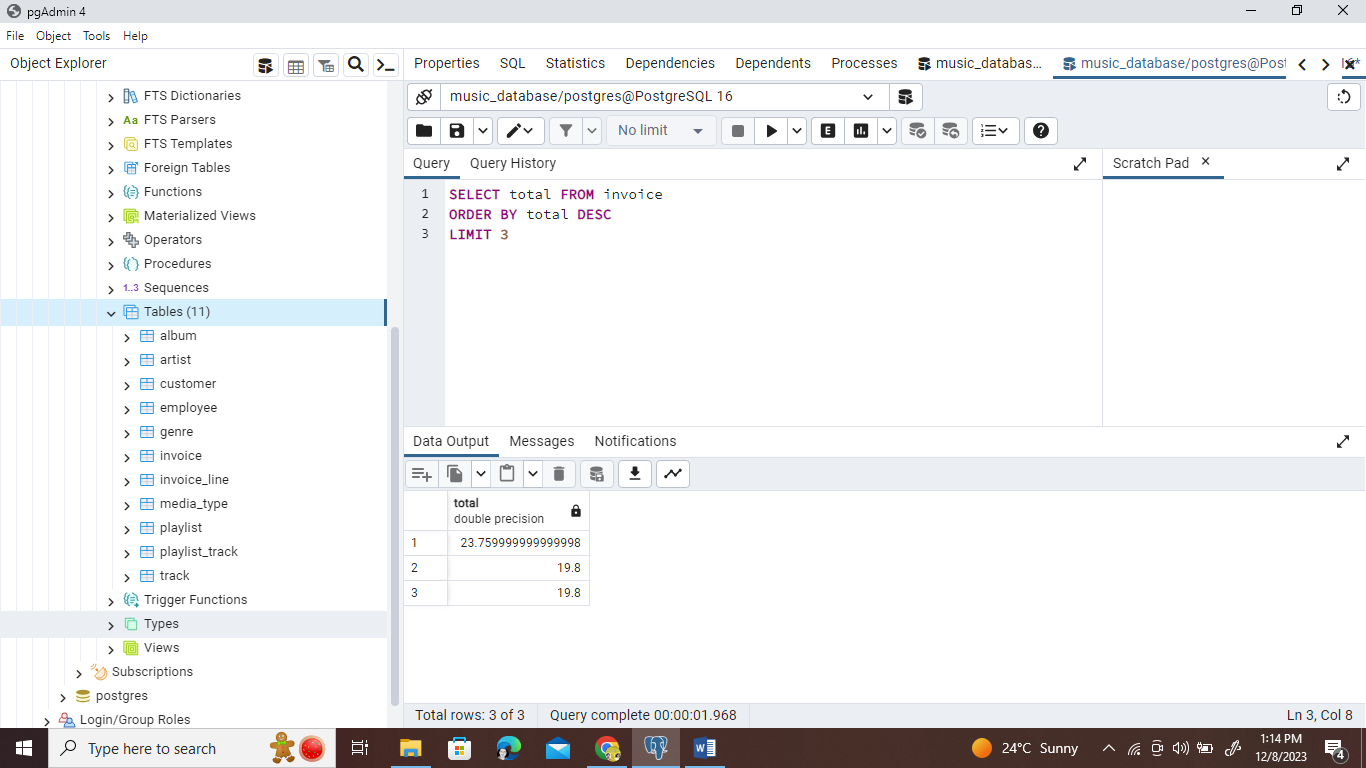
1. USA have the most Invoices.



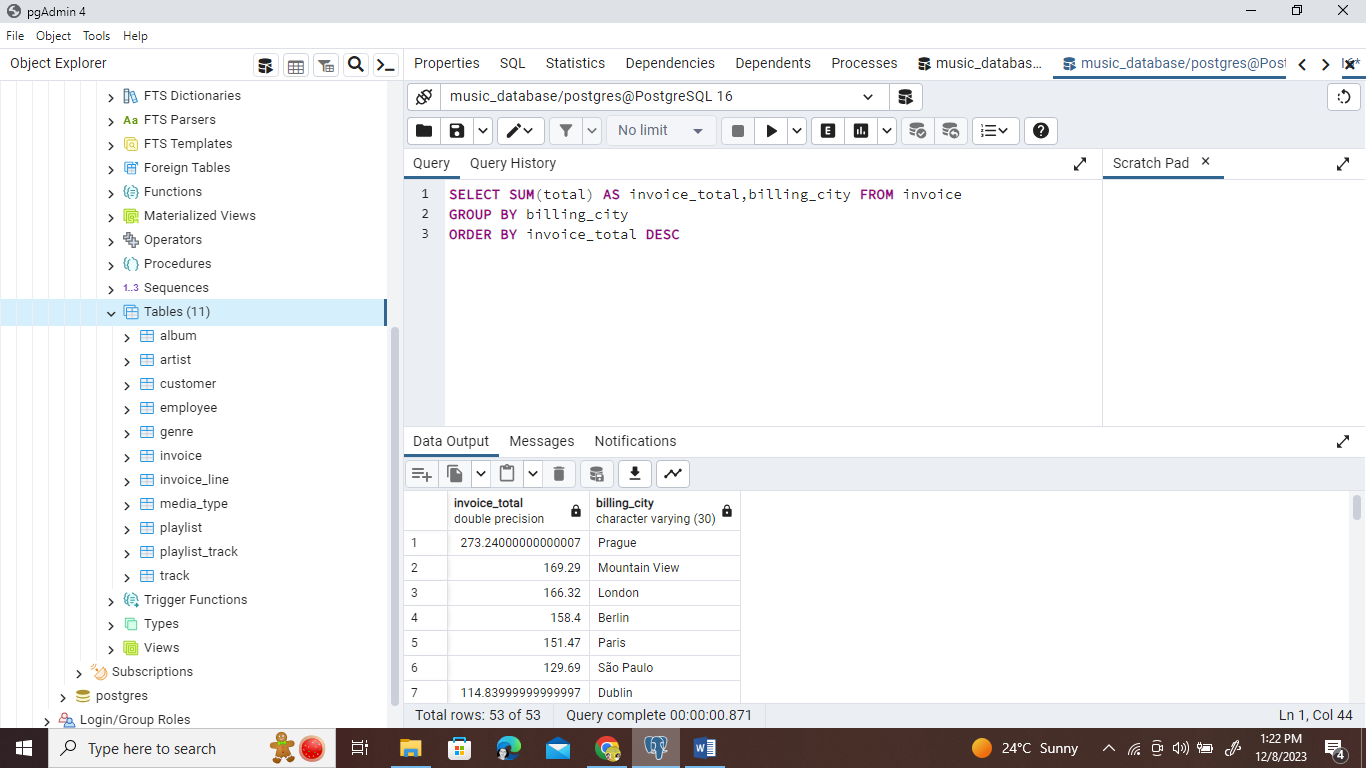
1. Top 3 values of total invoices are-

23.7599999, 19.8 and 19.8

See below-

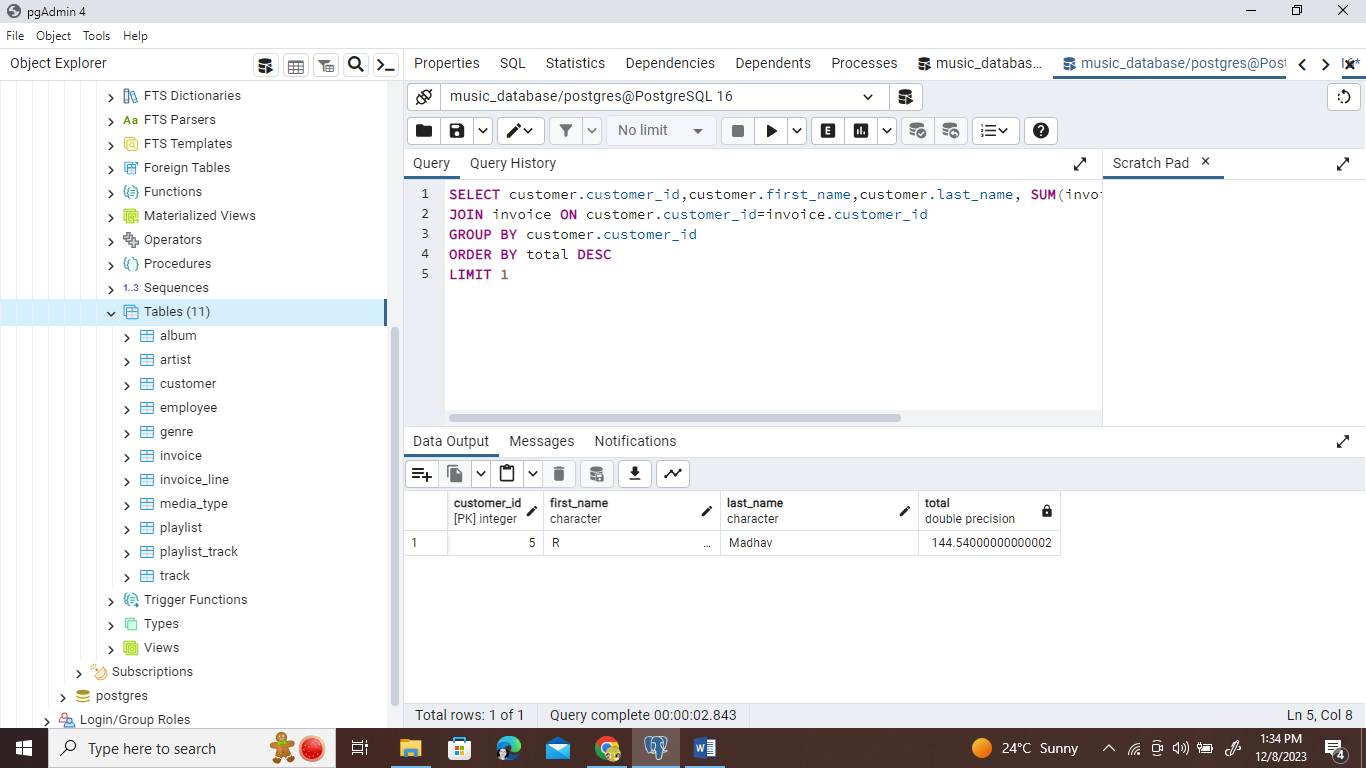


1. Prague city has best customers. And the required table as shown below-



1. To answer this question first we have to connect the customer table with another table. And we have to take help from schema to check which table can connect with the customer table to solve our problem. Why we are connecting the table? Because in the customer table we have no data of invoices etc.

And checking from schema we get to know we can connect customer table to invoice table using customer\_id to solve our problem.



So, by this we have ‘R Madhav is Best Customer’.

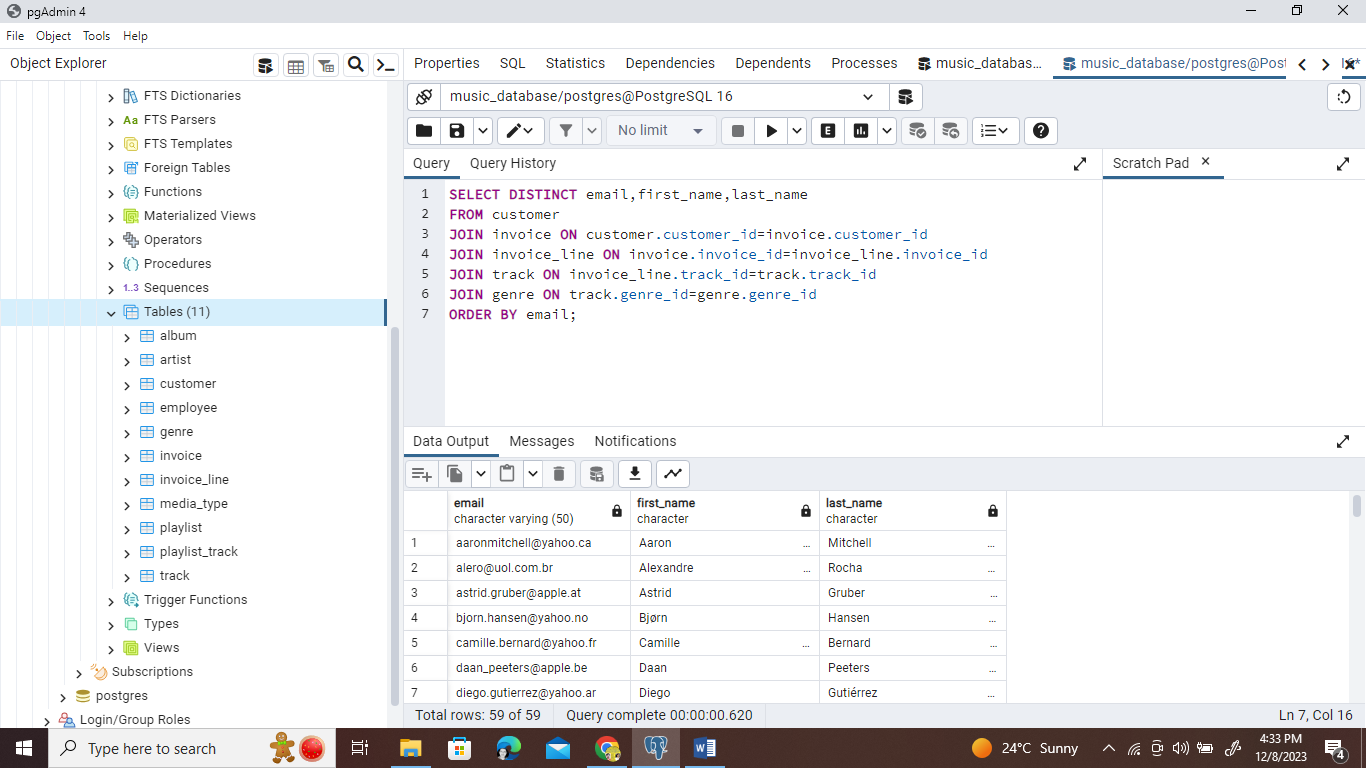
Thus, we have answered all our Easy Questions using Postgre SQL.

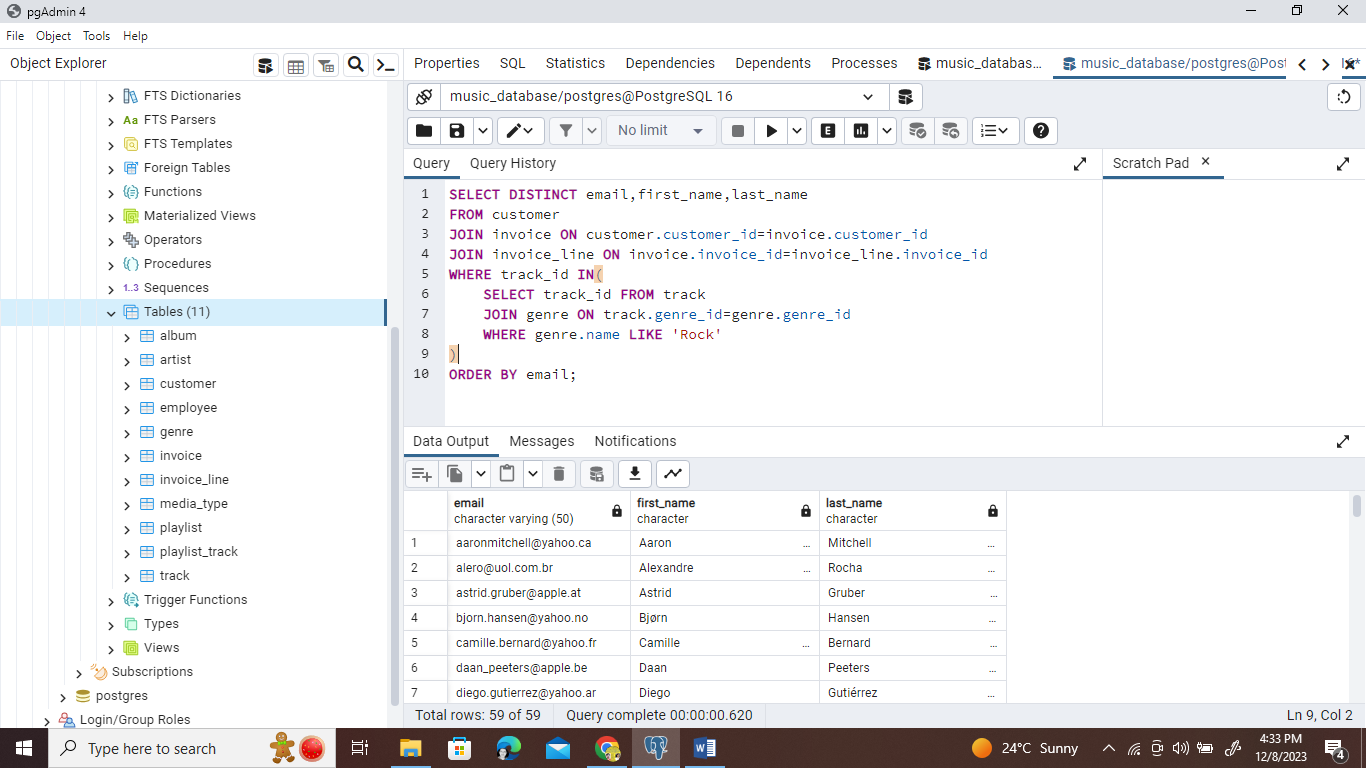
**Questions Set-2[Moderate]-**

1. Write query to return the email, first name, last name, & Genre of all Rock Music listeners. Return your list ordered alphabetically by email starting with A
2. Let's invite the artists who have written the most rock music in our dataset. Write a query that returns the Artist name and total track count of the top 10 rock bands
3. Return all the track names that have a song length longer than the average song length. Return the Name and Milliseconds for each track. Order by the song length with the longest songs listed first.

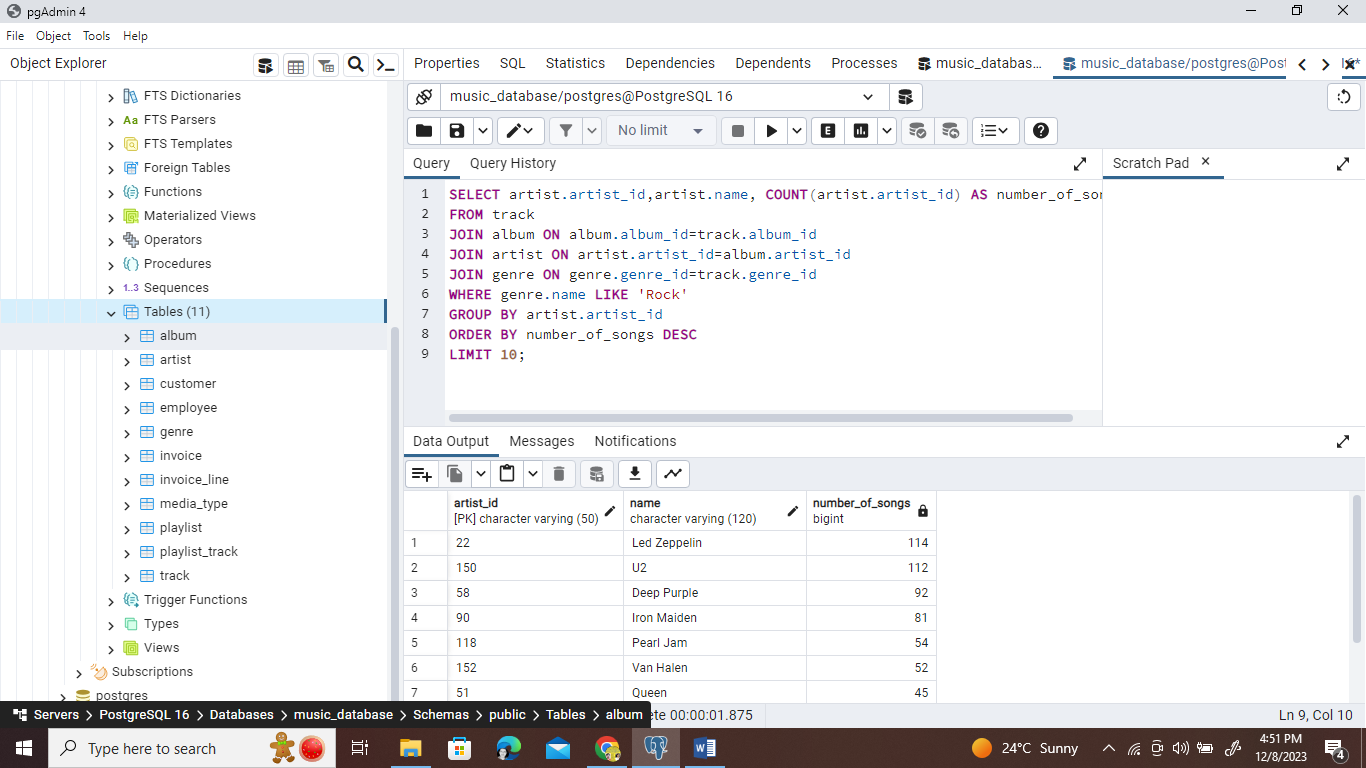
**Solution Set-2[Moderate]-**

1. We first use the schema and check what are the table we need to use. Then we have two ways to write the queries-

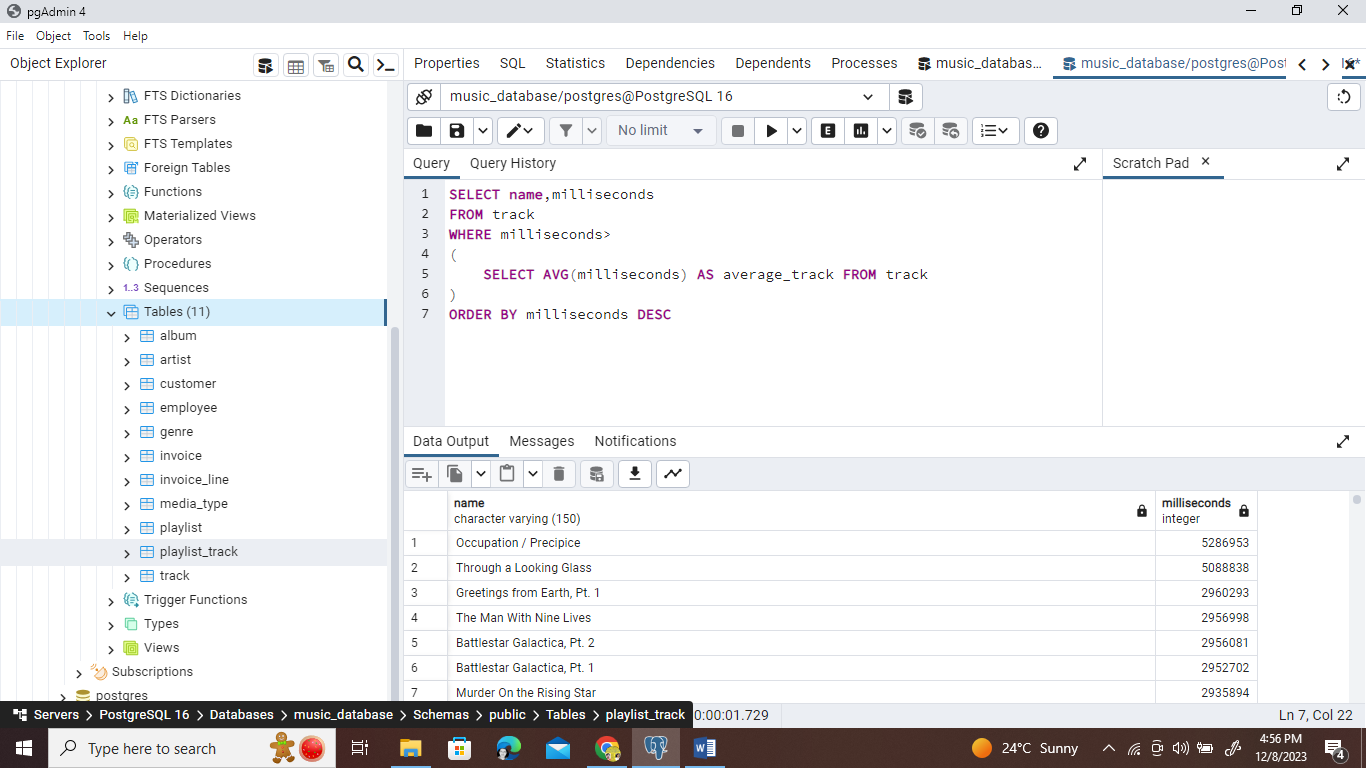




1. Using the help of scheme we have JOIN four table as shown to get the answer.



1. All the track listed are-

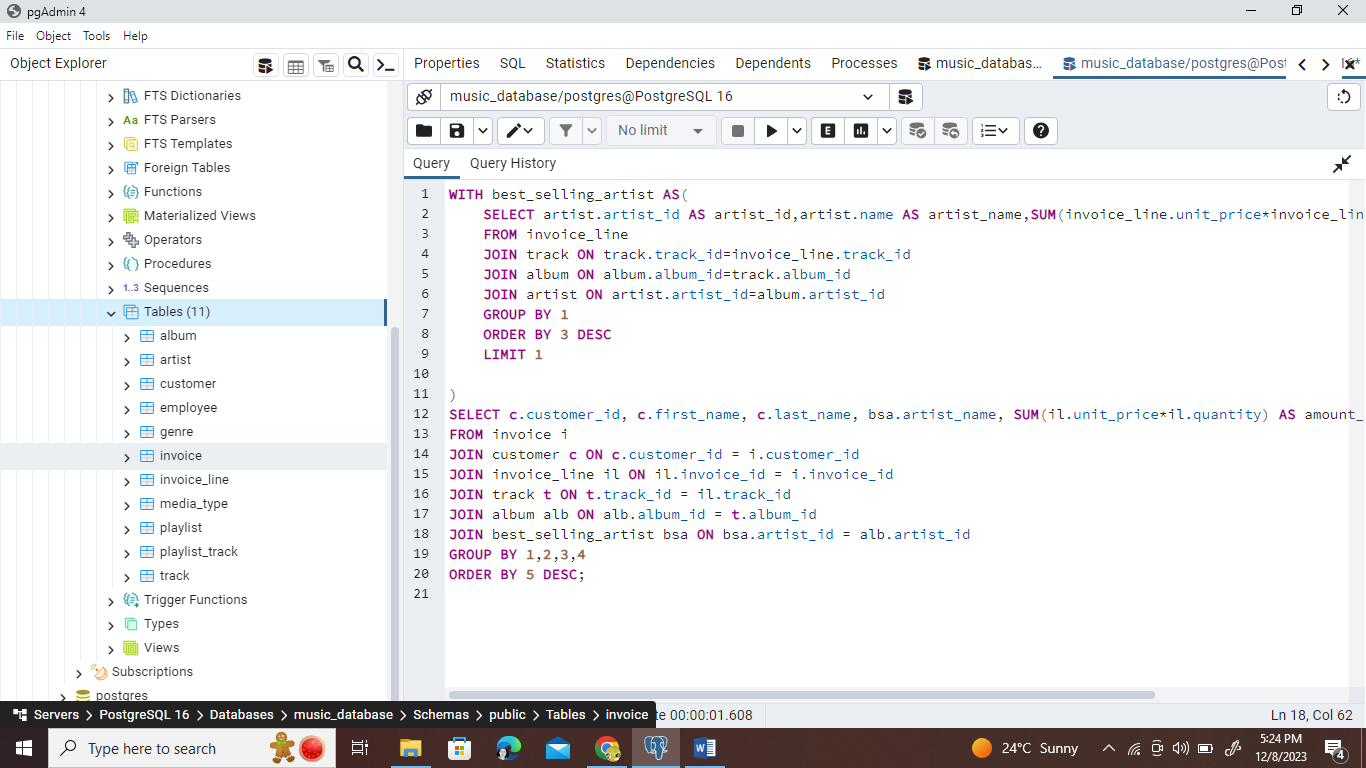


**Questions Set-3[Advanced]-**

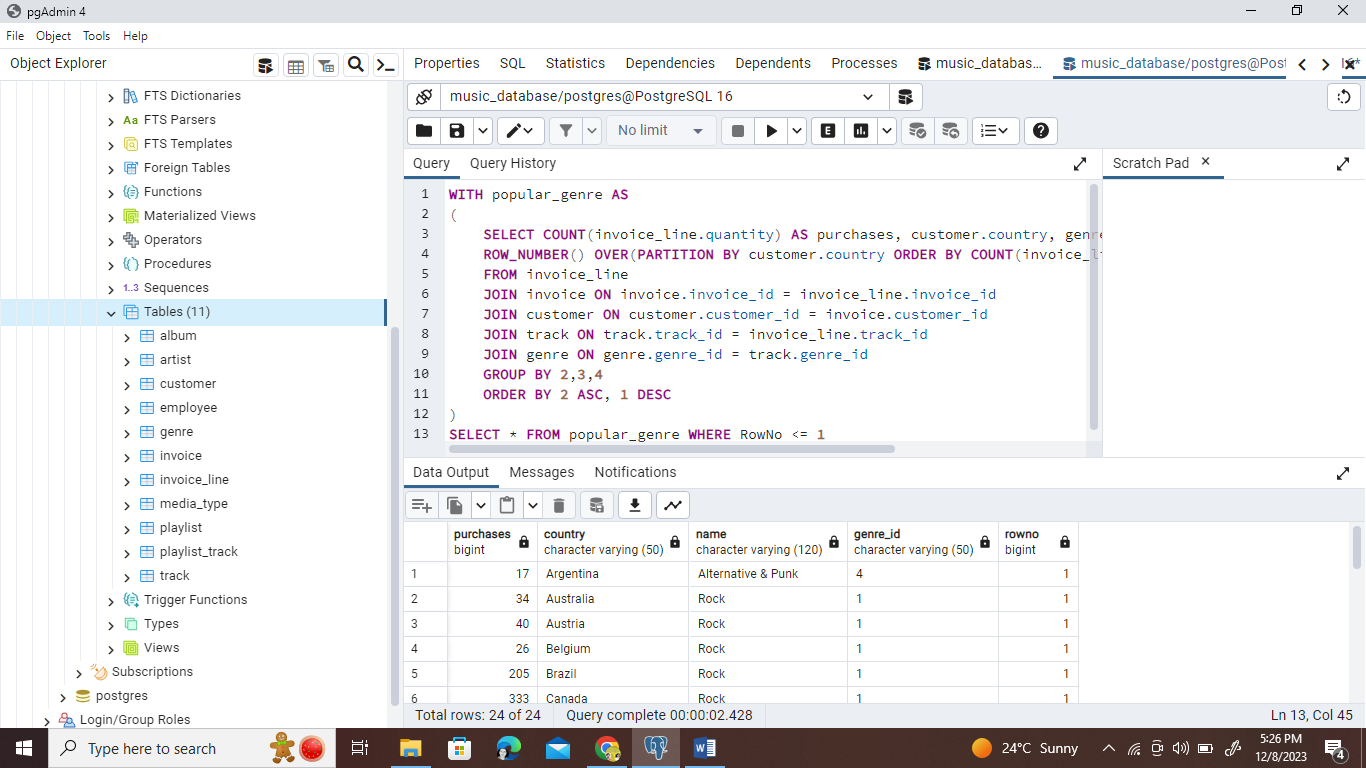
1. Find how much amount spent by each customer on artists? Write a query to return customer name, artist name and total spent.
2. We want to find out the most popular music Genre for each country. We determine the most popular genre as the genre with the highest amount of purchases. Write a query that returns each country along with the top Genre. For countries where the maximum number of purchases is shared return all Genre.
3. Write a query that determines the customer that has spent the most on music for each country. Write a query that returns the country along with the top customer and how much they spent. For countries where the top amount spent is shared, provide all customers who spent this amount.

**Solutions Set-3[Advanced]-**

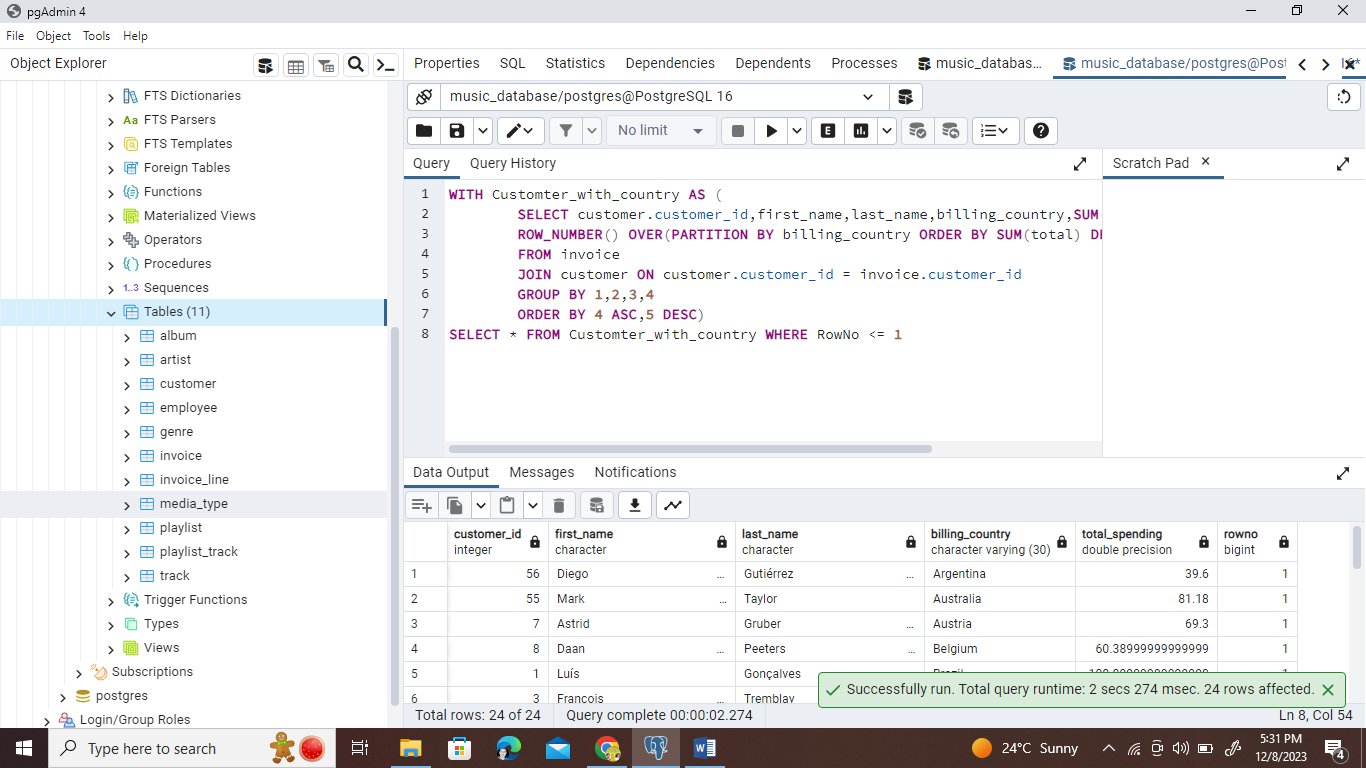
1. We take the help of schema to check what are the table needed to join. Also here we also see an example of creating CTE.



1. Here solution is –



1. Here Solution is-



**Result-** We have answered all the questions and analyzed all the situations using Postgre SQL.