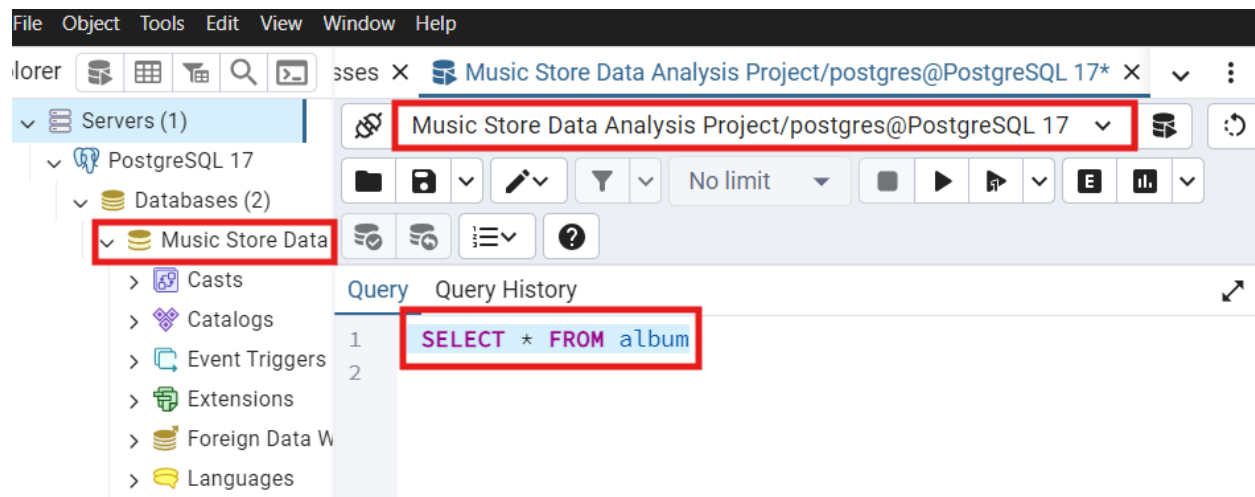


Music Store Data Analysis using SQL Database

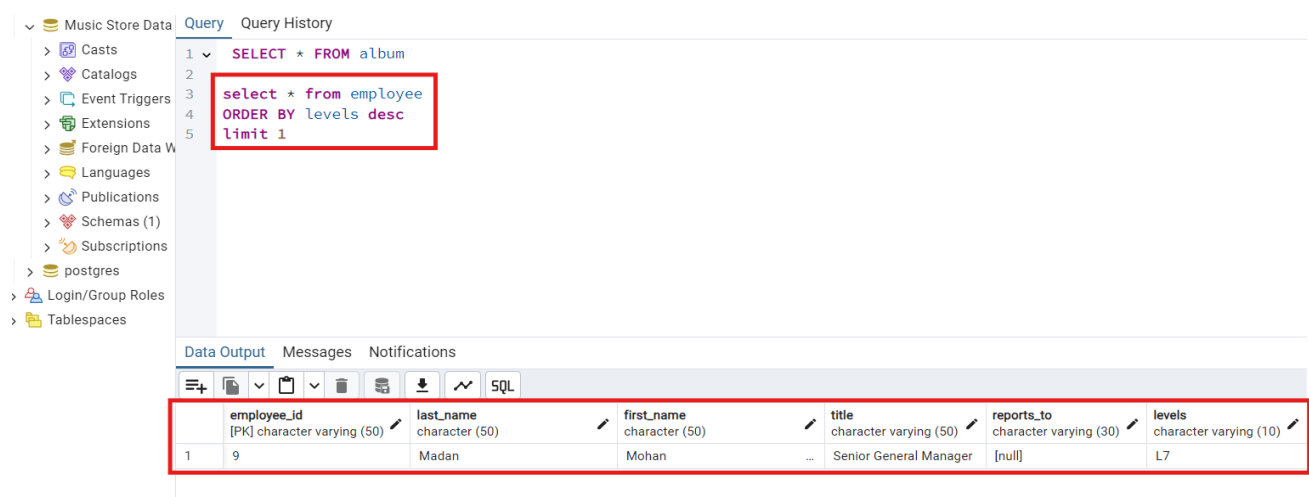
Question Set 1 – Easy Total Questions 5

Q1: Who is the senior most employee based on job role?

Solution: First of all, we will import our database after that we will click the query to solve the questions so to import the database, we will select the database **by SELECT * FROM album**. And will press **F5** to execute the command.



So next is to find **the senior most employee based on job role** so as we know they have given some **rankings to employees** so we will select employees and **will sort the order**. And as we need only one name so we will **limit to 1** as in the image. So it will show us result in one row.



Q2: Which countries have the most Invoices?

Solution: As its about **invoices** and we have the invoice table so we will select that. We will have then **selected count** from **billing countries** to knows the **most billings in countries from invoices**. Then will group by them as we can have multiples invoices from same country. And will sort it descending order. We can also limit them to 1 or more rows on how many we want.

```
5 select COUNT (*) as c , billing_country
6 from invoice
7 group by billing_country
8 order by c desc
```

Data Output Messages Notifications

	c	billing_country
1	131	USA
2	76	Canada

So, in a result we have found the most invoices from a country.

Question 3: What are top 3 values of total invoice?

So, we will again select our invoice table and as we need just total from invoice so we won't select all but just select total will sort it to get the data in sorted value and will limit it to 3 values as in the image.

Music Store Data Query

```
1 SE invoice
2
3 select total from invoice
4 order by total desc
5 limit 3
```

Data Output Messages Notifications

	total
1	23.759999999999998
2	19.8
3	19.8

We can see the top 3 values from invoice.

Question4: Which city has the best customers? We would like to throw a promotional Music Festival in the city where 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 totals.

Solution: As we will need to select sum of invoices and will name it as invoice total and then also select billing city as we need the city with most invoices which means Profit from the most city where they want to throw a music festival after that we will sort the data of sum, we get from city in order. As we can see in the image.

```
1 SELECT SUM(total) as invoice_total, billing_city
2 FROM invoice
3 group by billing_city
4 order by invoice_total desc
```

invoice_total	billing_city
273.24000000000007	Prague
169.29	Mountain View
166.32	London

So the answer is Prague city where they can throw the music festival

Q5: 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.

Solution: As we don't have invoice table in **customers** so we'll connect **invoice table** with customer id of customer with the help of schema. As we need which customers has spent the more money.

```
1 select customer.customer_id, customer.first_name, customer.last_name, SUM(invoice.total) as total
2 from customer
3 JOIN invoice ON customer.customer_id = invoice.customer_id
4 GROUP BY customer.customer_id
5 ORDER BY total desc
6 Limit 1
```

customer_id	first_name	last_name	total
5	R	Madhav	144.54000000000002

Questions Set 2 – Moderate- Total Questions 3

Q1. 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

Solution:

The screenshot shows a database query editor with a sidebar on the left containing a tree view of database objects. The main area displays a SQL query. The query is as follows:

```
1 SELECT DISTINCT email,first_name, last_name
2 From customer
3 JOIN invoice on customer.customer_id = invoice.customer_id
4 JOIN invoice_line on invoice.invoice_id = invoice_line.invoice_id
5 where track_id in (
6     select track_id from track
7     join genre on track.genre_id = genre.genre_id
8     where genre.name like 'Rock'
9 )
10 order by email;
```

Below the query editor, the 'Data Output' tab is active, showing the results of the query. The results are displayed in a table with the following columns: email, first_name, and last_name. The table contains 9 rows of data, ordered alphabetically by email.

	email	first_name	last_name
1	aaronmitchell@yahoo.ca	Aaron	Mitchell
2	alero@uol.com.br	Alexandre	Rocha
3	astrid.gruber@apple.at	Astrid	Gruber
4	bjorn.hansen@yahoo.no	Bjorn	Hansen
5	camille.bernard@yahoo.fr	Camille	Bernard
6	daan.peeters@apple.be	Daan	Peeters
7	diego.gutierrez@yahoo.ar	Diego	Gutiérrez
8	dmiller@comcast.com	Dan	Miller
9	dominiquelefebvre@gmail.c...	Dominique	Lefebvre

Q2: 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.

Solution: using scheme to connect and then execute as shown in image count 10 bands.

The screenshot shows a database query editor with a sidebar on the left containing a tree view of database objects. The main area displays a SQL query. The query is as follows:

```
1 SELECT artist.artist_id, artist.name,count (artist.artist_id) as number_of_songs
2
3 From track
4 JOIN album on album.album_id = track.album_id
5 JOIN artist on artist.artist_id = album.artist_id
6 JOIN genre on genre.genre_id = track.genre_id
7 where genre.name like 'Rock'
8 group by artist.artist_id
9 order by number_of_songs desc
10 limit 10;
```

Below the query editor, the 'Data Output' tab is active, showing the results of the query. The results are displayed in a table with the following columns: artist_id, name, and number_of_songs. The table contains 10 rows of data, ordered by the number of songs in descending order.

	artist_id	name	number_of_songs
2	150	U2	112
3	58	Deep Purple	92
4	90	Iron Maiden	81
5	118	Pearl Jam	54
6	152	Van Halen	52
7	51	Queen	45
8	142	The Rolling Stones	41
9	76	Creedence Clearwater Revival	40
10	52	Kiss	35

Q3. 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: IN this we have two steps first of all we will have to create average then use filter in it.

The screenshot shows a database interface with a query editor and a results table. The query editor contains the following SQL code:

```
1 select name,milliseconds
2 from track
3 where milliseconds > (
4 select avg (milliseconds) as avg_track_length
5 from track)
6 order by milliseconds desc ;
```

A red arrow points from the query editor to the results table. The results table has two columns: 'name' (character varying (150)) and 'milliseconds' (integer). The table contains 9 rows of data, ordered by milliseconds in descending order.

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