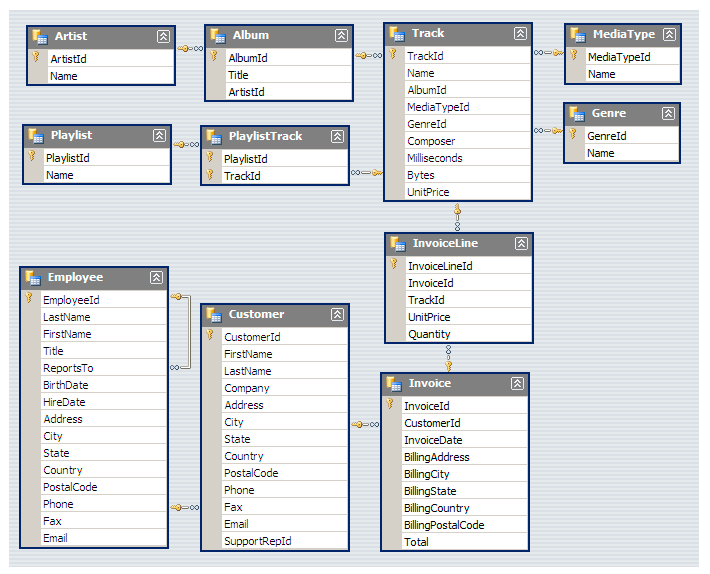
**MUSIC STORE SQL PROJECT REPORT**

**In this Project all the tables are connected in following manner:**

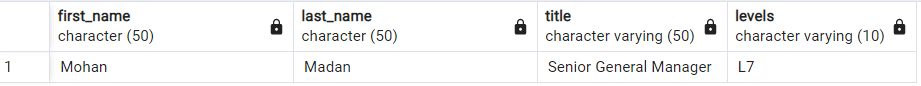
****

**Q.1. Who is the senior most employee based on job title?**

**Ans:**

Select first\_name, last\_name, title, levels from employee order by levels desc limit 1;

***Output:***

****

**Q.2. Who is the senior most employee based on Age?**

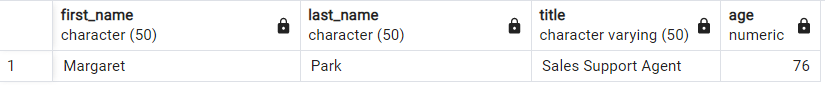
**Ans:**

Select first\_name, last\_name, title, (extract(year from current\_date)-(extract(year from birthdate)))

as Age from employee

order by Age desc limit 1;

***Output:***

****

**Q.3. Who is the most senior in terms of working years?**

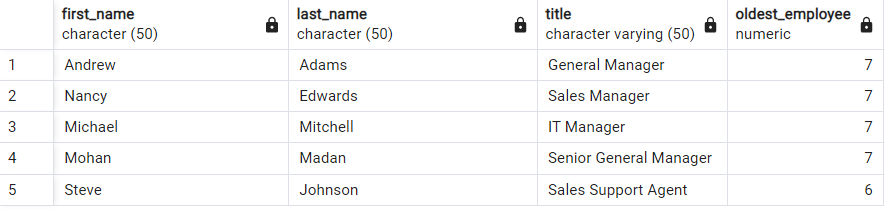
**Ans:**

Select first\_name, last\_name, title, (extract(year from current\_date)-(extract(year from hire\_date)))As Oldest\_employee

from employee

order by Oldest\_employee desc limit 5;

***Output:***



**Q.4.Which Top 5 countries have the most Invoices?**

**Ans**:

Select count(\*) as count,

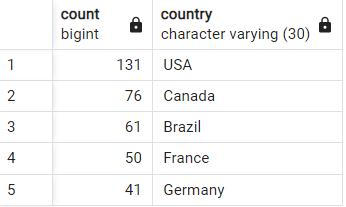
billing\_country as country

from invoice

group by country

order by count desc limit 5;

***Output:***

****

**Q.5. What are top 3 values of total invoice?**

**Ans:**

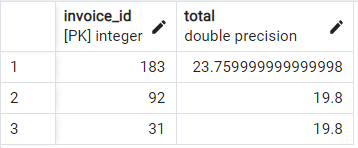
Select invoice\_id, total

from invoice

order by total desc

limit 3;

***Output:***



**Q.6. 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 totals**

**Ans:**

Select billing\_city as city ,sum(total) as total\_invoice

from invoice

group by city

order by total\_invoice desc limit 1;

***Output:***



**Q.7. 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**

**Ans:**

Select customer.first\_name, last\_name,

customer.customer\_id,

sum(invoice.total) as Total

from customer join invoice

on customer.customer\_id=invoice.customer\_id

group by customer.customer\_id

order by Total desc

limit 1

***Output:***



**Q.8. 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**

**Ans:**

SELECT distinct email, first\_name, last\_name

from customer

JOIN invoice on customer.customer\_id=invoice.customer\_id

JOIN invoice\_line on invoice.invoice\_id=invoice\_line.invoice\_id

WHERE track\_id IN(

select track\_id from track

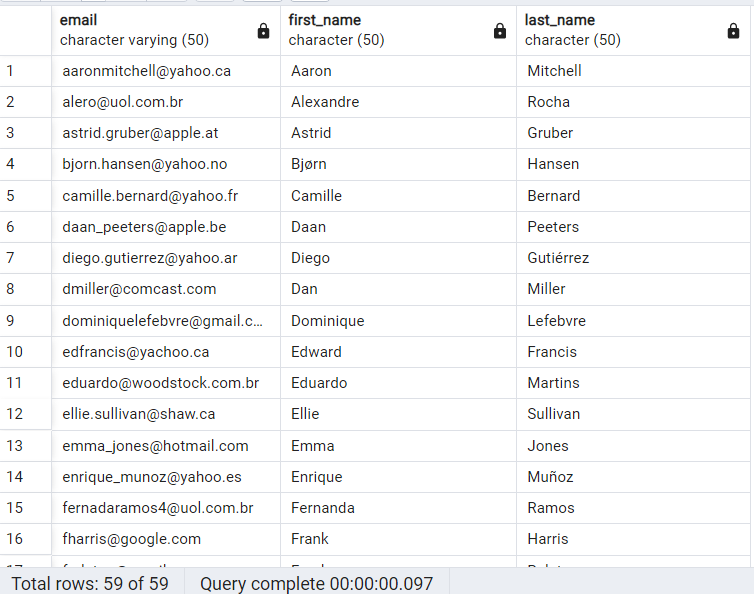
join genre on track.genre\_id=genre.genre\_id

where genre.name like 'Rock'

)

order by email;

***Output:***



**Q.9. Let's invite the artists who have written the most rock music in our dataset. Write a qry that returns the Artist name and total track count of the top 10 rock ban**

**Ans:**

Select artist.name, artist.artist\_id,

count(artist.artist\_id) as Number\_of\_songs

from artist

join album on artist.artist\_id=album.artist\_id

join track on album.album\_id=track.album\_id

where track\_id In(

select track\_id from track

join genre on track.genre\_id=genre.genre\_id

where genre.name LIKE 'Rock'

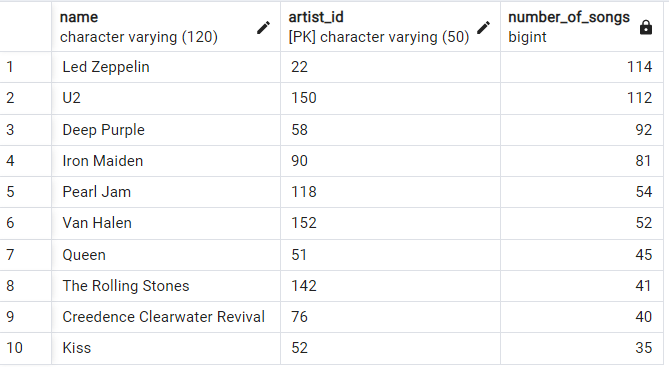
)

group by artist.artist\_id

order by Number\_of\_songs desc

limit 10;

***Output:***



**Q.10. 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 long songs listed first**

**Ans:**

Select name, milliseconds from track

where milliseconds>

(

select avg(milliseconds)

from track

)

order by milliseconds desc

***Output:***

