



# SQL PROJECT- MUSIC STORE DATA ANALYSIS

---

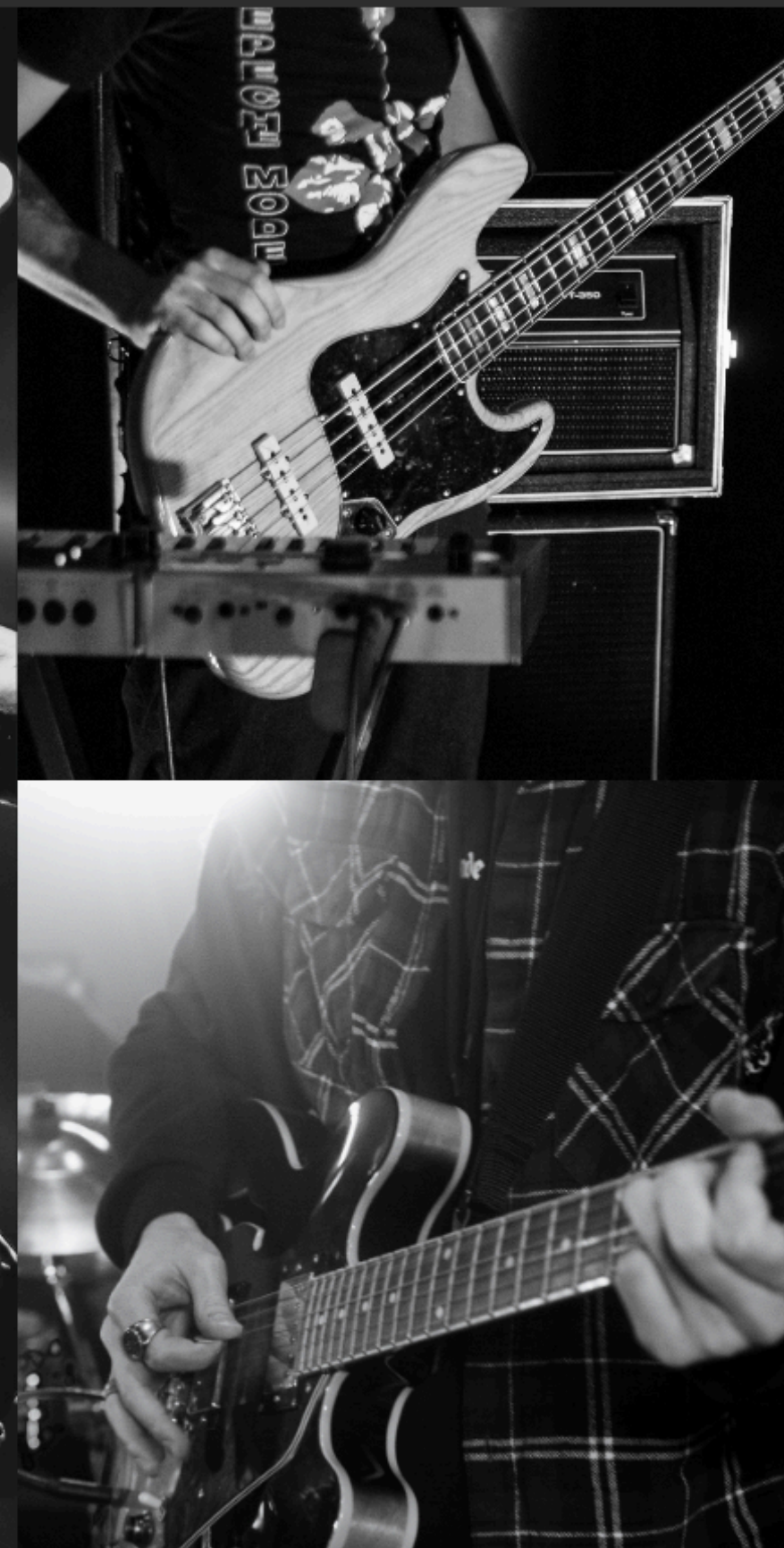


# ABOUT PROJECT

I am a computer science student passionate about data analytics. After earning my SQL certification, I created my first project: SQL Music Database Analysis using PostgreSQL and pgAdmin. The project is divided into three sections—Easy, Intermediate, and Advanced—covering basic queries, joins, aggregate functions, and advanced techniques like window functions and CTEs. The goal is to demonstrate SQL proficiency and derive insights from a music database, with the full code available on GitHub for collaboration

GITHUB LINK :

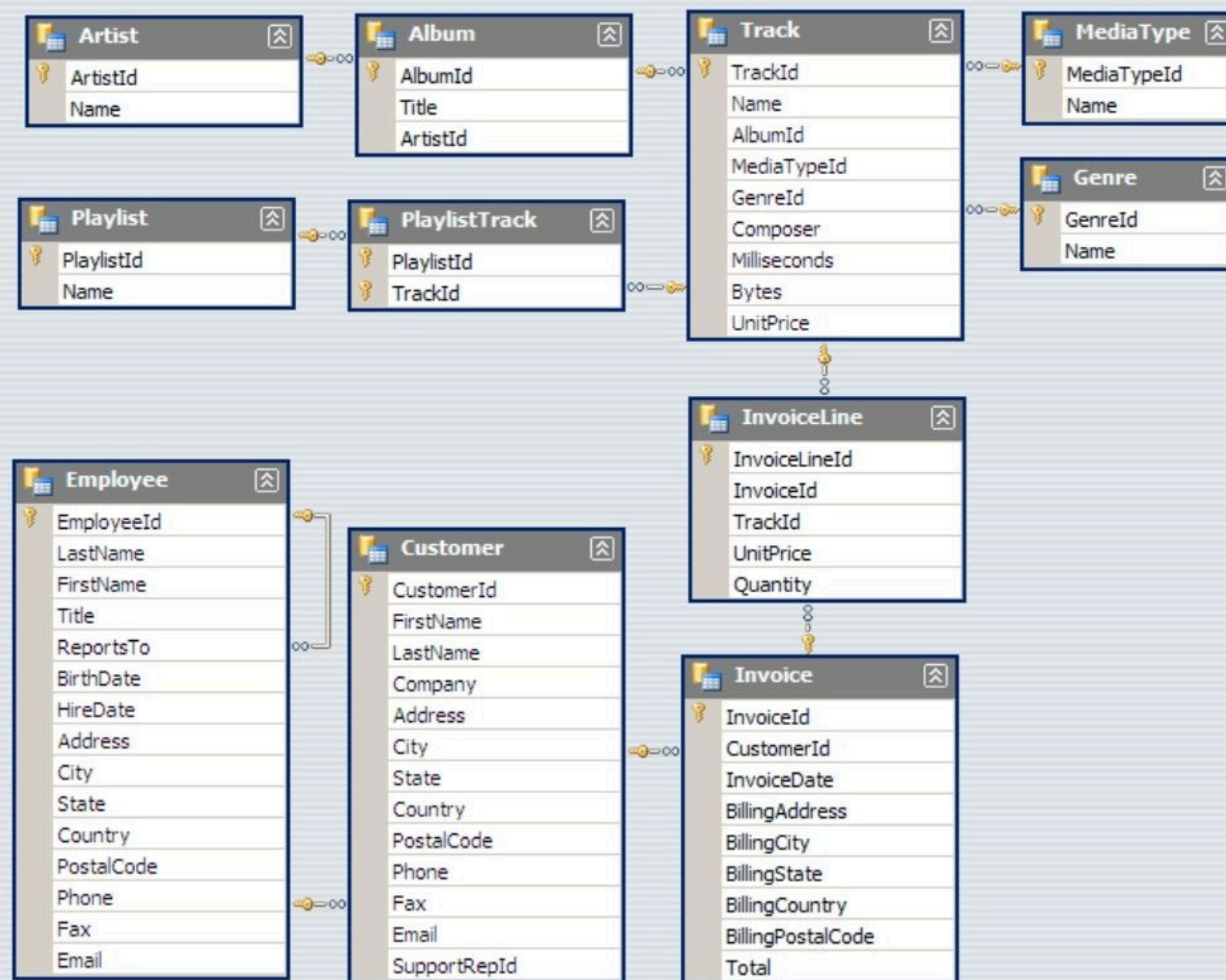
<https://github.com/Akash-Sherkar?tab=repositories>







# MUSIC DATABASE SCHEMA



The schema is designed for normalization, referential integrity, and scalability, supporting efficient queries and comprehensive music data analysis.

# Problems

## MUSIC DATABASE ANALYSIS

### 1.Question 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

# Problems

## MUSIC DATABASE ANALYSIS

### Question 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 totals
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

# Problems

## MUSIC DATABASE ANALYSIS

### Question Set 3 – Advance

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 Genres
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

Question set 1 - Easy



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

```
SELECT title, last_name, first_name  
FROM employee  
ORDER BY levels DESC  
LIMIT 1
```



**Q2: Which countries have the most Invoices?**

```
SELECT COUNT(*) AS c, billing_country  
FROM invoice  
GROUP BY billing_country  
ORDER BY c DESC
```

**Q3: What are top 3 values of total invoice?**

```
SELECT total  
FROM invoice  
ORDER BY total DESC
```

# Solutions

Question set 1 - Easy



**Q4: 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**

```
SELECT billing_city,SUM(total) AS InvoiceTotal
FROM invoice
GROUP BY billing_city
ORDER BY InvoiceTotal DESC
LIMIT 1;
```



# Solutions

Question set 1 - Easy



**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**

```
Select customer.customer_id, first_name, last_name, SUM(total) AS total_spending
FROM customer
JOIN invoice ON customer.customer_id = invoice.customer_id
GROUP BY customer.customer_id
ORDER BY total_spending DESC
LIMIT 1;
```

# Solutions

Question set 2 - Moderate



**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.**

```
SELECT DISTINCT email AS Email,first_name AS FirstName, last_name AS LastName,  
genre.name AS Name  
FROM customer  
JOIN invoice ON invoice.customer_id = customer.customer_id  
JOIN invoiceline ON invoiceline.invoice_id = invoice.invoice_id  
JOIN track ON track.track_id = invoiceline.track_id  
JOIN genre ON genre.genre_id = track.genre_id  
WHERE genre.name LIKE 'Rock'  
ORDER BY email;
```

# Solutions

Question set 2 - Moderate



**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**

```
SELECT artist.artist_id, artist.name, COUNT(artist.artist_id) AS number_of_songs
FROM track
JOIN album ON album.album_id = track.album_id
JOIN artist ON artist.artist_id = album.artist_id
JOIN genre ON genre.genre_id = track.genre_id
WHERE genre.name LIKE 'Rock'
GROUP BY artist.artist_id
ORDER BY number_of_songs DESC
LIMIT 10;
```

# Solutions

Question set 2 - Moderate



**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**

```
SELECT name,milliseconds
FROM track
WHERE milliseconds > (
  SELECT AVG(milliseconds) AS avg_track_length
  FROM track )
ORDER BY milliseconds DESC;
```



# Solutions

Question set 3 - Advance



**Q1: Find how much amount spent by each customer on artists?  
Write a query to return customer name, artist name and total spent**

```
WITH best_selling_artist AS (  
  SELECT artist.artist_id AS artist_id, artist.name AS artist_name,  
  SUM(invoice_line.unit_price*invoice_line.quantity) AS total_sales  
  FROM invoice_line  
  JOIN track ON track.track_id = invoice_line.track_id  
  JOIN album ON album.album_id = track.album_id  
  JOIN artist ON artist.artist_id = album.artist_id  
  GROUP BY 1  
  ORDER BY 3 DESC  
  LIMIT 1)  
SELECT c.customer_id, c.first_name, c.last_name, bsa.artist_name,  
SUM(il.unit_price*il.quantity) AS amount_spent  
FROM invoice i  
JOIN customer c ON c.customer_id = i.customer_id  
JOIN invoice_line il ON il.invoice_id = i.invoice_id  
JOIN track t ON t.track_id = il.track_id  
JOIN album alb ON alb.album_id = t.album_id  
JOIN best_selling_artist bsa ON bsa.artist_id = alb.artist_id  
GROUP BY 1,2,3,4  
ORDER By 5 Desc;
```

# Solutions

## Question set 3 - Advance



**Q2: 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 Genres**

```
WITH popular_genre AS
(
    SELECT COUNT(invoice_line.quantity) AS purchases, customer.country, genre.name, genre.genre_id,
    ROW_NUMBER() OVER(PARTITION BY customer.country ORDER BY COUNT(invoice_line.quantity) DESC) AS RowNo
    FROM invoice_line
    JOIN invoice ON invoice.invoice_id = invoice_line.invoice_id
    JOIN customer ON customer.customer_id = invoice.customer_id
    JOIN track ON track.track_id = invoice_line.track_id
    JOIN genre ON genre.genre_id = track.genre_id
    GROUP BY 2,3,4
    ORDER BY 2 ASC, 1 DESC
)
SELECT * FROM popular_genre WHERE RowNo <= 1
```

# Solutions

## Question set 3 - Advance



**Q3: 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**

```
WITH Customter_with_country AS (  
  SELECT customer.customer_id,first_name,last_name,billing_country,SUM(total) AS total_spending,  
         ROW_NUMBER() OVER(PARTITION BY billing_country ORDER BY SUM(total) DESC) AS RowNo  
  FROM invoice  
  JOIN customer ON customer.customer_id = invoice.customer_id  
  GROUP BY 1,2,3,4  
  ORDER BY 4 ASC,5 DESC)  
SELECT * FROM Customter_with_country WHERE RowNo <= 1
```

# Learning Resources

For learning SQL, try "SQL for Data Analysis" by Cathy Tanimura, and Coursera's "SQL for Data Science." Practice on SQLZoo and LeetCode for interactive exercises.







# THANK YOU

---

