

# MUSIC DATA ANALYSIS

*By  
Megha*

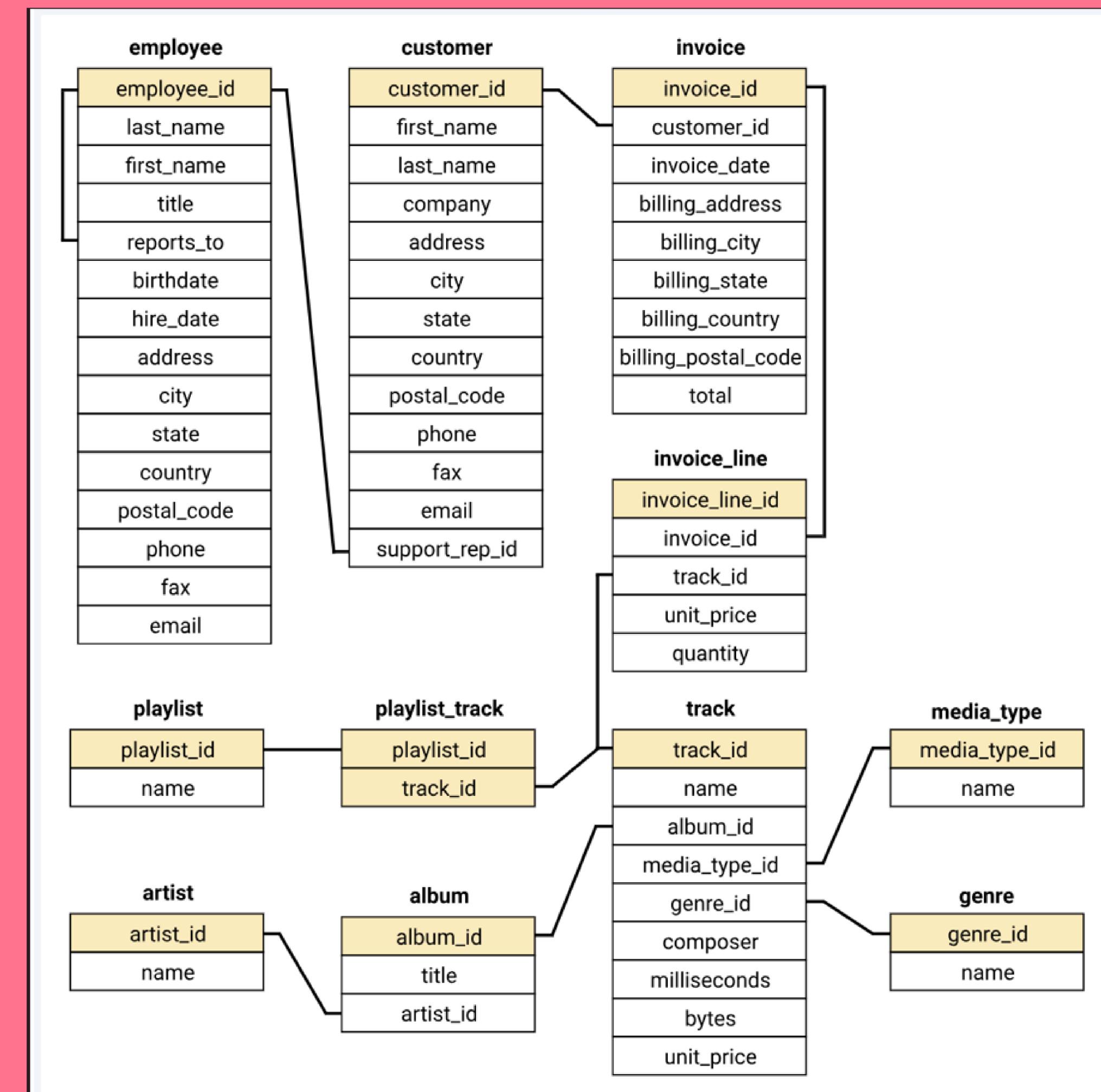


# OBJECTIVES

The primary objective of the music store is to achieve sustainable business growth while addressing existing challenges.

We need to examine dataset with SQL and help- the music store understand its business growth by answering simple questions

# SCHEMA



# QUESTIONS



## Question 1.

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

# INPUT:

```
SELECT * from employee  
ORDER BY levels DESC  
LIMIT 1;
```

# OUTPUT:

employee_id	last_name	first_name	title	reports_to	levels	birthdate	hire_date	address	city	state	country	postal_code	phone	fax	email
1	Adams	Andrew	General Manager	9	L6	18-02-1962 00:00	14-08-2016 00:00	11120 Jasper Ave NW	Edmonton	AB	Canada	T5K 2N1	+1 (780) 428-9482	+1 (780) 428-3457	andrew.adams@company.com



## Question 2

**Which countries have the most Invoices?**

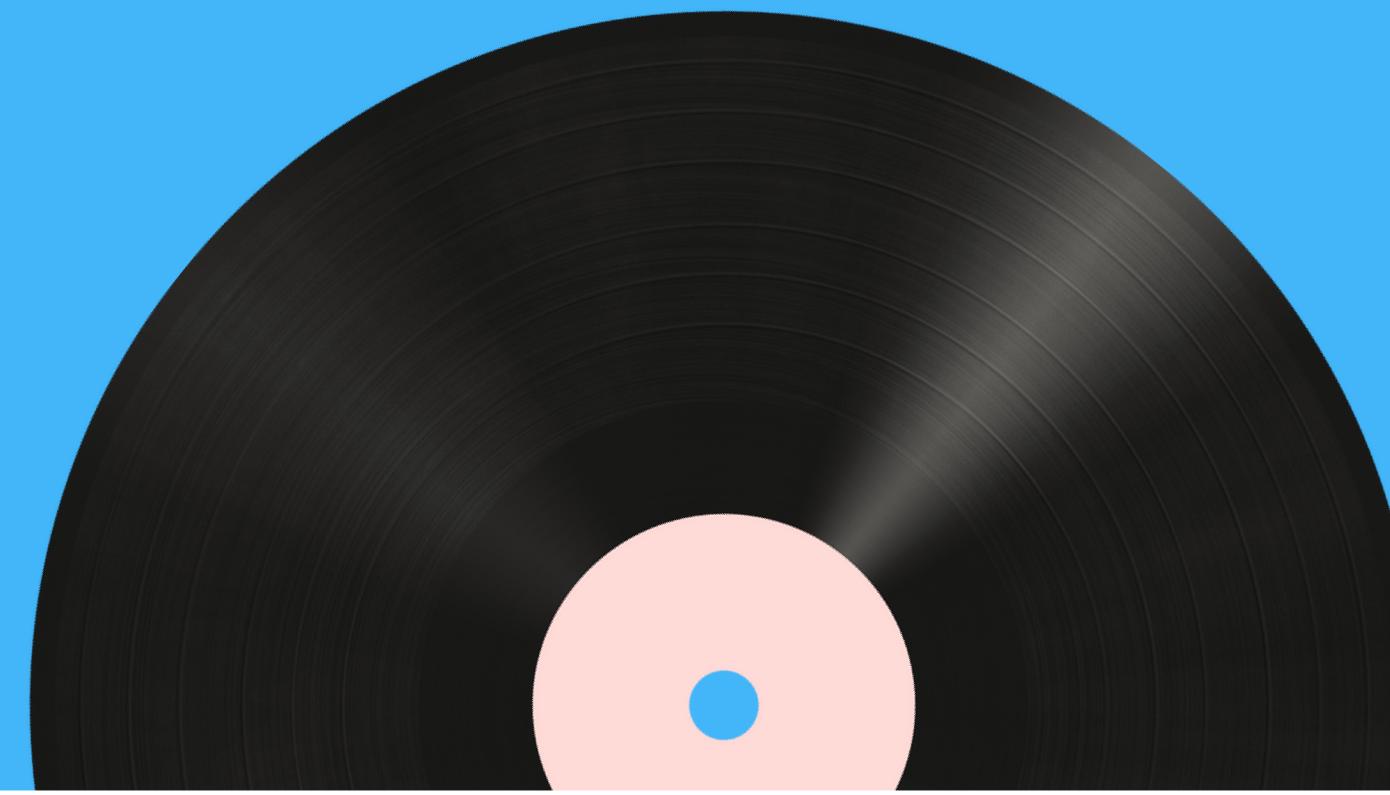
## INPUT:

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

## OUTPUT:

	Count_of_Invoices	Billing_country
▶	131	USA
	76	Canada
	61	Brazil
	50	France
	41	Germany
	30	Czech Republic
	29	Portugal





### Question 3

**What are the top 3 values of Total Invoice**

# INPUT:

```
SELECT * FROM Invoice  
ORDER BY Total DESC  
LIMIT 3;
```

# OUTPUT:

	total
▶	23.759999999999998
	19.8
	19.8





## Question 4

**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.**

## INPUT:

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

## OUTPUT:

	billing_city	InvoiceTotal
▶	Prague	273.24000000000007





## Question 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.**

## INPUT:

```
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, first_name, last_name
ORDER BY total_spending DESC
LIMIT 1;
```

## OUTPUT:

	customer_id	first_name	last_name	total_spending
▶	5	František	Wichterlová	144.5400000000002





## Question 6

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.

## INPUT:

```
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_line_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:

	email	first_name	last_name
▶	aaronmitchell@yahoo.ca	Aaron	Mitchell
	alero@uol.com.br	Alexandre	Rocha
	astrid.gruber@apple.at	Astrid	Gruber
	camille.bernard@yahoo.fr	Camille	Bernard
	daan_peeters@apple.be	Daan	Peeters
	diego.gutierrez@yahoo.ar	Diego	GutiÃ©rrez
	dmiller@comcast.com	Dan	Miller



## Question 7

**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.**

# INPUT:

```
SELECT artist.artist_id, artist.name, COUNT(track.track_id) AS number_of_songs  
FROM track  
JOIN album2 ON album2.album_id = track.album_id  
JOIN artist ON artist.artist_id = album2.artist_id  
JOIN genre ON genre.genre_id = track.genre_id  
WHERE genre.name LIKE 'Rock'  
GROUP BY artist.artist_id, artist.name  
ORDER BY number_of_songs DESC  
LIMIT 10;
```

# OUTPUT:

	artist_id	name	number_of_songs
▶	1	AC/DC	18
	3	Aerosmith	15
	8	Audioslave	14
	22	Led Zeppelin	14
	4	Alanis Morissette	13
	5	Alice In Chains	12
	23	Frank Zappa & Captain Beefheart	9





## Question 8

**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.**

# INPUT:

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

# OUTPUT:

	name	milliseconds
▶	How Many More Times	711836
	Advance Romance	677694
	Sleeping Village	644571
	You Shook Me(2)	619467
	Talkin' 'Bout Women Obviously	589531
	Stratus	582086
	No More Tears	555075

