



Search



Home



Library

# Digital Music Store Analysis

- Presented by Soumya Das



Links:

- Data
- PostgreSQL Queries

5th July, 2024

# Introduction

## Project Overview:

---

This project analyzes music store data to answer a series of questions using PostgreSQL. The analysis covers various aspects of the store's operations, including employee hierarchy, sales distribution, customer behavior, and music preferences.

## Objectives:

---

### Employee Analysis:

- Identify the senior-most employee based on job title.

### Sales Distribution:

- Determine which countries have the most invoices.
- Identify the top three invoice values.
- Find the city with the highest sum of invoice totals to plan a promotional event.

### Customer Insights:

- Recognize the best customer by the total money spent.

### Music Preferences:

- List Rock music listeners with their email, first name, last name, and genre.
- Invite the top 10 rock bands by track count.
- List all tracks longer than the average song length, ordered by length.

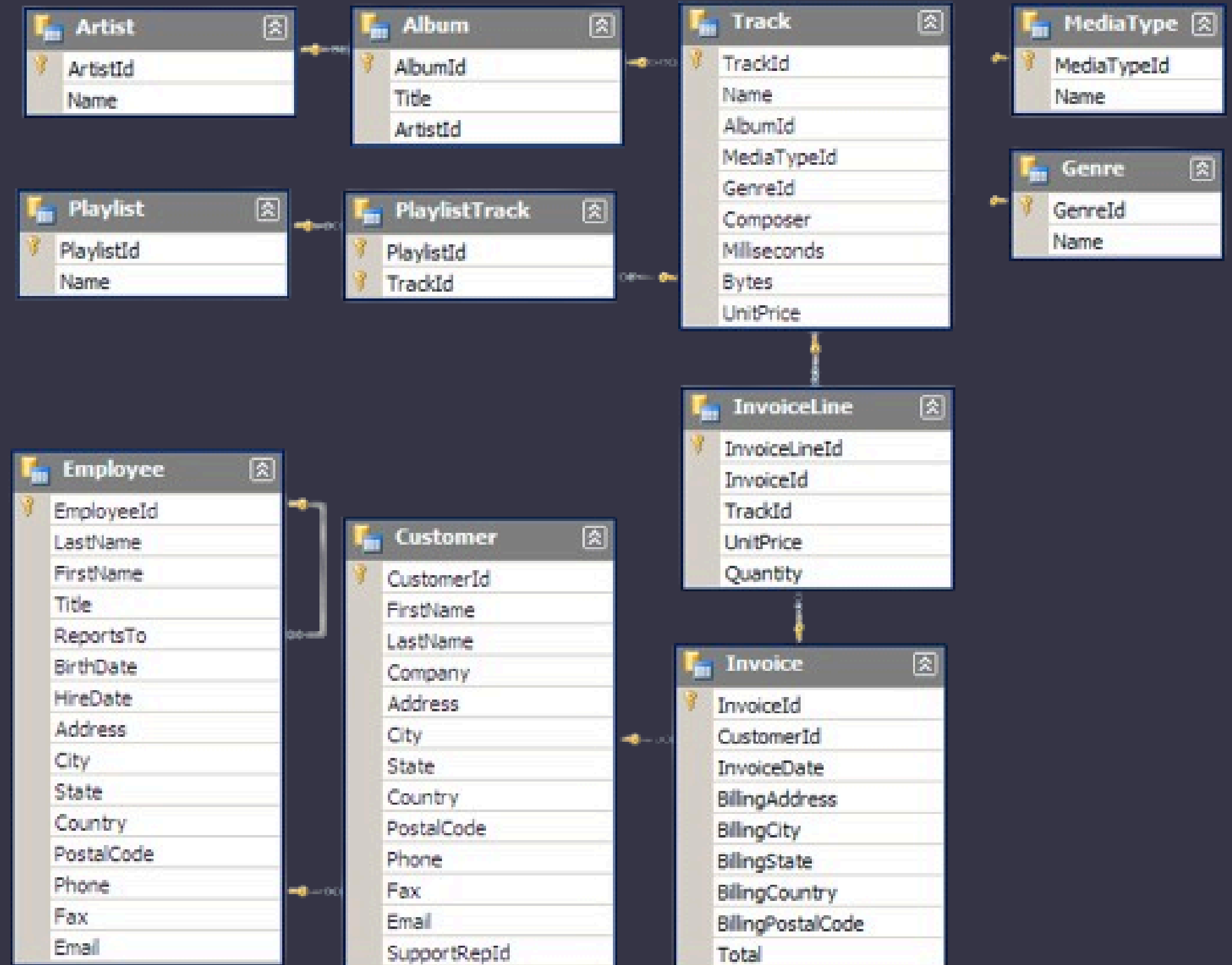
### Advanced Analysis:

- Calculate the amount spent by each customer on different artists.
- Determine the most popular music genre in each country based on purchases.
- Identify the top spending customer in each country.



# Schema

- Album
- Artist
- Customer
- Employee
- Genre
- Invoice
- Invoice\_Line
- Media\_Type
- Playlist
- Playlist\_Track



# MOST SENIOR EMPLOYEE

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

```
select * from employee
ORDER BY levels desc
limit 1
```

01

| employee_id<br>[PK] character varying (50) | last_name<br>character (50) | first_name<br>character (50) | title<br>character varying (50) | reports_to<br>character varying (30) | levels<br>character varying (50) |
|--|-----------------------------|------------------------------|---------------------------------|--------------------------------------|----------------------------------|
| 9  | Madan                       | Mohan                        | Senior General Manag...         | [null]                               | L7                               |

Madan Mohan

# COUNTRY WITH MOST INVOICE

Q2: Which countries have the most Invoices?


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



## 02



# USA

|   | c<br>bigint |  | billing_country<br>character varying (30) |
|---|-------------|---|---|
| 1 | 131         |   | USA                                       |
| 2 | 76          |   | Canada                                    |
| 3 | 61          |   | Brazil                                    |
| 4 | 50          |   | France                                    |
| 5 | 41          |   | Germany                                   |
| 6 | 30          |   | Czech Republic                            |

# TOP 3 VALUES OF TOTAL INVOICE


Q3: What are top 3 values of total invoice

```
SELECT total FROM invoice
order by total desc
limit 3
```

03



23.7, 19.8, 19.8


|   | total<br>double precision  |
|---|---|
| 1 | 23.75999999999999998  |
| 2 | 19.8  |
| 3 | 19.8  |

# CITY THAT HAVE THE BEST CUSTOMERS

```
select SUM(total) as invoice_total, billing_city
from invoice
group by billing_city
order by invoice_total desc
```

04

Prague

| invoice_total<br>double precision  | billing_city<br>character varying (30) |
|---|--|
| 273.24000000000000  | Prague                                 |
| 169.29  | Mountain View                          |
| 166.32  | London                                 |
| 158.4   | Berlin                                 |



# WHO IS THE BEST CUSTOMER

```
select customer.customer_id, customer.first_name, customer.last_name, 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
```

05

| customer_id<br>[PK] integer | first_name<br>character (50) | last_name<br>character (50) | total<br>double precision |
|-----------------------------|------------------------------|-----------------------------|---------------------------|
| 5                           | R                            | Madhav                      | 144.54000000000000        |

R Madhav





# EMAIL, FIRST NAME, LAST NAME OF ALL ROCK GENRE MUSIC LISTNER

```
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;
```

06

aaron\*\*@ya\*\* Aaron Mitchell  
(format)

| email                  | first_name     | last_name      |
|------------------------|----------------|----------------|
| character varying (50) | character (50) | character (50) |
| aaronmitchell@yahoo... | Aaron          | Mitchell       |
| alero@uol.com.br       | Alexandre      | Rocha          |
| astrid.gruber@apple.at | Astrid         | Gruber         |
| bjorn.hansen@yahoo...  | Bjørn          | Hansen         |
| camille.bernard@yah... | Camille        | Bernard        |

(top 5 results are shown)

# ARTISTS WHO HAVE WRITTEN THE MOST ROCK MUSIC IN OUR DATASET

```
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;
```

(returns  
top 10)

07

Led Zeppelin

|   | artist_id<br>[PK] character varying (50) | name<br>character varying (120) |
|---|--|---------------------------------|
| 1 | 22                                       | Led Zeppelin                    |
| 2 | 150                                      | U2                              |
| 3 | 58                                       | Deep Purple                     |
| 4 | 90                                       | Iron Maiden                     |
| 5 | 118                                      | Pearl Jam                       |

(top 5 results are shown)

# ALL THE TRACK NAME THAT HAVE A SONG LENGTH LONGER THAN THE AVERAGE SONG LENGTH

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

08

Led Zeppelin

|   | name<br>character varying (150) | milliseconds<br>integer |
|---|---------------------------------|-------------------------|
| 1 | 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                 |

(top 5 results are shown)

# AMOUNT SPENT BY EACH CUSTOMER ON BEST SELLING ARTIST



# 09

```
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;
```



Top-selling Artist: Queen  
Top-spending Customer: Hugh O'Reilly

|   | customer_id | first_name | last_name | artist_name | amount_spent       |
|---|-------------|------------|-----------|-------------|--------------------|
| 1 | 46          | Hugh       | O'Reilly  | Queen       | 27.719999999999999 |
| 2 | 38          | Niklas     | Schröder  | Queen       | 18.81              |
| 3 | 3           | François   | Tremblay  | Queen       | 17.82              |

(top 3 results are shown)

# MOST POPULAR MUSIC GENRE IN EACH COUNTRY BASED ON PURCHASES



```
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
```

# 10

|   | purchases<br>bigint | country<br>character varying (50) | name<br>character varying (120) | genre_id<br>character varying (50) | rowno<br>bigint |
|---|---------------------|-----------------------------------|---------------------------------|------------------------------------|-----------------|
| 1 | 17                  | Argentina                         | Alternative & Punk              | 4                                  | 1               |
| 2 | 34                  | Australia                         | Rock                            | 1                                  | 1               |
| 3 | 40                  | Austria                           | Rock                            | 1                                  | 1               |

(top 3 alphabetically)

## Argentina: Alternative & Punk



# MOST POPULAR MUSIC GENRE IN EACH COUNTRY BASED ON PURCHASES



# 11

```
WITH RECURSIVE
  customter_with_country AS (
    SELECT customer.customer_id, first_name, last_name, billing_country, SUM(total) AS total_spending
    FROM invoice
    JOIN customer ON customer.customer_id = invoice.customer_id
    GROUP BY 1, 2, 3, 4
    ORDER BY 2, 3 DESC),

  country_max_spending AS(
    SELECT billing_country, MAX(total_spending) AS max_spending
    FROM customter_with_country
    GROUP BY billing_country)

SELECT cc.billing_country, cc.total_spending, cc.first_name, cc.last_name, cc.customer_id
FROM customter_with_country cc
JOIN country_max_spending ms
ON cc.billing_country = ms.billing_country
WHERE cc.total_spending = ms.max_spending
ORDER BY 1;
```

|   | billing_country<br>character varying (30) | total_spending<br>double precision | first_name<br>character (50) | last_name<br>character (50) | customer_id<br>integer |
|---|---|------------------------------------|------------------------------|-----------------------------|------------------------|
| 1 | Argentina                                 | 39.6                               | Diego                        | Gutiérrez                   | 56                     |
| 2 | Australia                                 | 81.18                              | Mark                         | Taylor                      | 55                     |
| 3 | Austria                                   | 69.3                               | Astrid                       | Gruber                      | 7                      |
| 4 | Belgium                                   | 60.38999999999999                  | Daan                         | Peeters                     | 8                      |
| 5 | Brazil                                    | 108.89999999999998                 | Luís                         | Gonçalves                   | 1                      |

Country: Argentina  
Top-spending Customer: Diego Gutierrez

(top 5 results are shown)



# Thank you



Checkout my works on:

