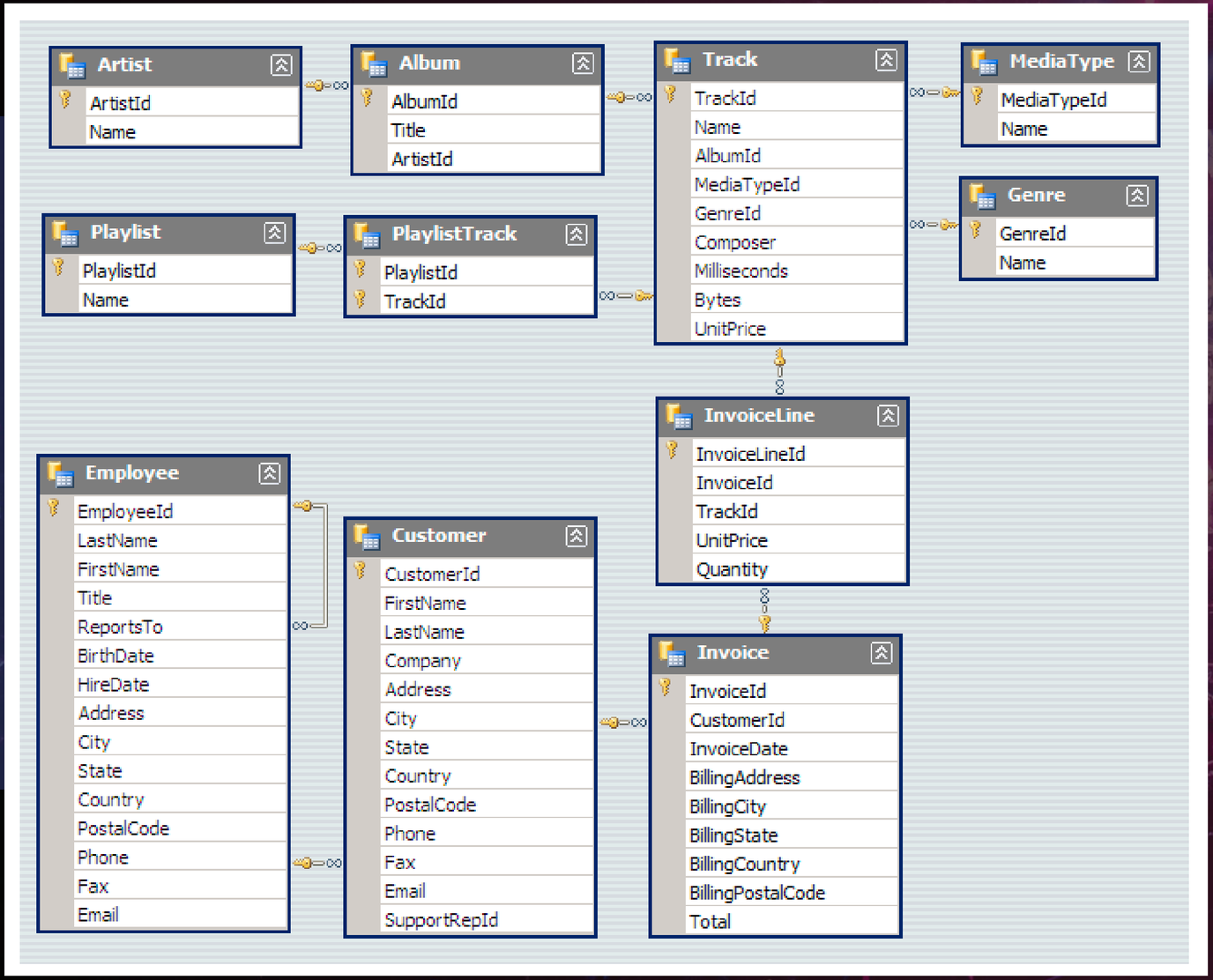

SQL PROJECT
MUSIC STORE ANALYSIS

INTRODUCTION

I am an aspiring data analyst with a strong interest in SQL and database management. This project is the culmination of my learning, where I applied SQL to analyze a real-world dataset for a Music Store Analysis.

The Music Store Analysis project focuses on exploring and extracting meaningful insights from the store's database. The database contains information about various albums, songs by different artists, music genres, employees, customers, and invoice details. This project demonstrates my ability to manage real-world data scenarios, write efficient queries, and derive insights that can inform decision-making.



THE DATABASE STRUCTURE

Steps I worked on:

Created a database in MySQL Workbench and imported eleven CSV files.

Established relationships between the tables for data connectivity.

Executed various SQL queries on the dataset to derive insights

PROBLEM STATEMENT CATEGORY

EASY

- SELECT
- ORDER BY
- LIMIT

MODERATE

- JOINS
- AGGREAGTE
- FUNCTIONS

HARD

- RANKINGS
- FUNCTIONS
- CTE
- JOINS

EASY CATEGORY

Who is the senior most employee based on job title?

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

employee_id	last_name	first_name	title	reports_to	levels	birthdate	hire_date	address
1	Adams	Andrew	General Manager	9	L6	18-02-1962 00:00	14-08-2016 00:00	11120 Jasper Av

Which countries have the most Invoices?

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

c	billing_country
131	USA
76	Canada
61	Brazil
50	France
41	Germany
30	Czech Republic
29	Portugal
28	United Kingdom
21	India

What are top 3 values of total invoice?

```
SELECT total
FROM invoice
ORDER BY total DESC
LIMIT 3;
```

total
23.75999999999998
19.8
19.8

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 SUM(total) AS invoice_total, billing_city
FROM invoice
GROUP BY billing_city
ORDER BY invoice_total DESC;
```

invoice_total	billing_city
273.24000000000007	Prague
169.29	Mountain View
166.32	London
158.4	Berlin
151.47	Paris
129.69	SÃ£o Paulo
114.83999999999997	Dublin
111.86999999999999	Delhi
108.89999999999998	SÃ£o JosÃ© dos Campos

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

customer_id	first_name	last_name	total
5	FrantiÅ¡ek	WichterlovÃ¡	144.54000000000002

MODERATE CATEGORY

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

```sql
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;
```

email	first_name	last_name
aaronmitchell@yahoo.ca	Aaron	Mitchell
alero@uol.com.br	Alexandre	Rocha
astrid.gruber@apple.at	Astrid	Gruber
bjorn.hansen@yahoo.no	BjÃ¸rn	Hansen
camille.bernard@yahoo.fr	Camille	Bernard
daan_peeters@apple.be	Daan	Peeters
diego.gutierrez@yahoo.ar	Diego	GutiÃ©rrez
dmiller@comcast.com	Dan	Miller
dominiquelefebvre@gmail.com	Dominique	Lefebvre

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

```sql
SELECT artist.artist_id, artist.name, COUNT(artist.artist_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;
```

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
2	Accept	4

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;
```

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
The Alchemist	509413
Wheels Of Confusion / The Straightener	494524

ADVANCE CATEGORY

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

```sql
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 album2 ON album2.album_id = track.album_id
	JOIN artist ON artist.artist_id = album2.artist_id
	GROUP BY artist.artist_id, artist.name
	ORDER BY total_sales 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 album2 alb ON alb.album_id = t.album_id
JOIN best_selling_artist bsa ON bsa.artist_id = alb.artist_id
GROUP BY c.customer_id, c.first_name, c.last_name, bsa.artist_name
ORDER BY amount_spent DESC;
```

OUTPUT

customer_id	first_name	last_name	artist_name	amount_spent
54	Steve	Murray	AC/DC	17.82
53	Phil	Hughes	AC/DC	10.89
21	Kathy	Chase	AC/DC	10.89
49	StanisÅ‚aw	WÃ³jcik	AC/DC	9.9
1	LuÃ-s	GonÃ§alves	AC/DC	7.920000000000001
24	Frank	Ralston	AC/DC	7.920000000000001
31	Martha	Silk	AC/DC	3.96
16	Frank	Harris	AC/DC	2.9699999999999998
42	Wyatt	Girard	AC/DC	2.9699999999999998
6	Helena	HolÃ½	AC/DC	2.9699999999999998
38	Niklas	SchrÃ¶der	AC/DC	2.9699999999999998

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

```sql
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 customer.country, genre.name, genre.genre_id
)
SELECT *
FROM popular_genre
WHERE RowNo = 1;
```

OUTPUT

purchases	country	name	genre_id	RowNo
1	Argentina	Rock	1	1
18	Australia	Rock	1	1
6	Austria	Rock	1	1
5	Belgium	Rock	1	1
26	Brazil	Rock	1	1
57	Canada	Rock	1	1
7	Chile	Rock	1	1
14	Czech Republic	Rock	1	1
6	Denmark	Rock	1	1
6	Finland	Rock	1	1
26	France	Rock	1	1
28	Germany	Rock	1	1

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

```sql
WITH Customer_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 customer.customer_id, first_name, last_name, billing_country
)
SELECT *
FROM Customer_with_country
WHERE RowNo = 1;
```

OUTPUT

customer_id	first_name	last_name	billing_country	total_spending	RowNo
56	Diego	GutiÃ©rrez	Argentina	39.6	1
55	Mark	Taylor	Australia	81.18	1
7	Astrid	Gruber	Austria	69.3	1
8	Daan	Peeters	Belgium	60.38999999999999	1
1	LuÃ-s	GonÃ§alves	Brazil	108.89999999999998	1
3	FranÃ§ois	Tremblay	Canada	99.99	1
57	Luis	Rojas	Chile	97.02000000000001	1
5	FrantiÅiek	WichterlovÃi	Czech Republic	144.54000000000002	1
9	Kara	Nielsen	Denmark	37.61999999999999	1
44	Terhi	HÃ¤mÃ¤lÃ¤inen	Finland	79.2	1
42	Wyatt	Girard	France	99.99	1

THANK YOU.