SQL PROJECT

MUSIC STORE ANALYSIS

Author: **Viraj N. Bhutada**

INTRODUCTION

Greetings and welcome to our SQL Music Store Data Analysis project—a pragmatic exploration into the heart of our business data. In this initiative, we've employed SQL to dissect and understand the nuances of our dataset. Our goal is simple yet impactful: to extract valuable insights that will guide our decision-making process. This project is all about translating raw data into practical knowledge, shedding light on customer behaviors and industry trends. As we navigate through the project, we aim to deliver actionable intelligence that can contribute to the strategic growth of our music store. Let's delve into the details and uncover the meaningful stories our data holds.

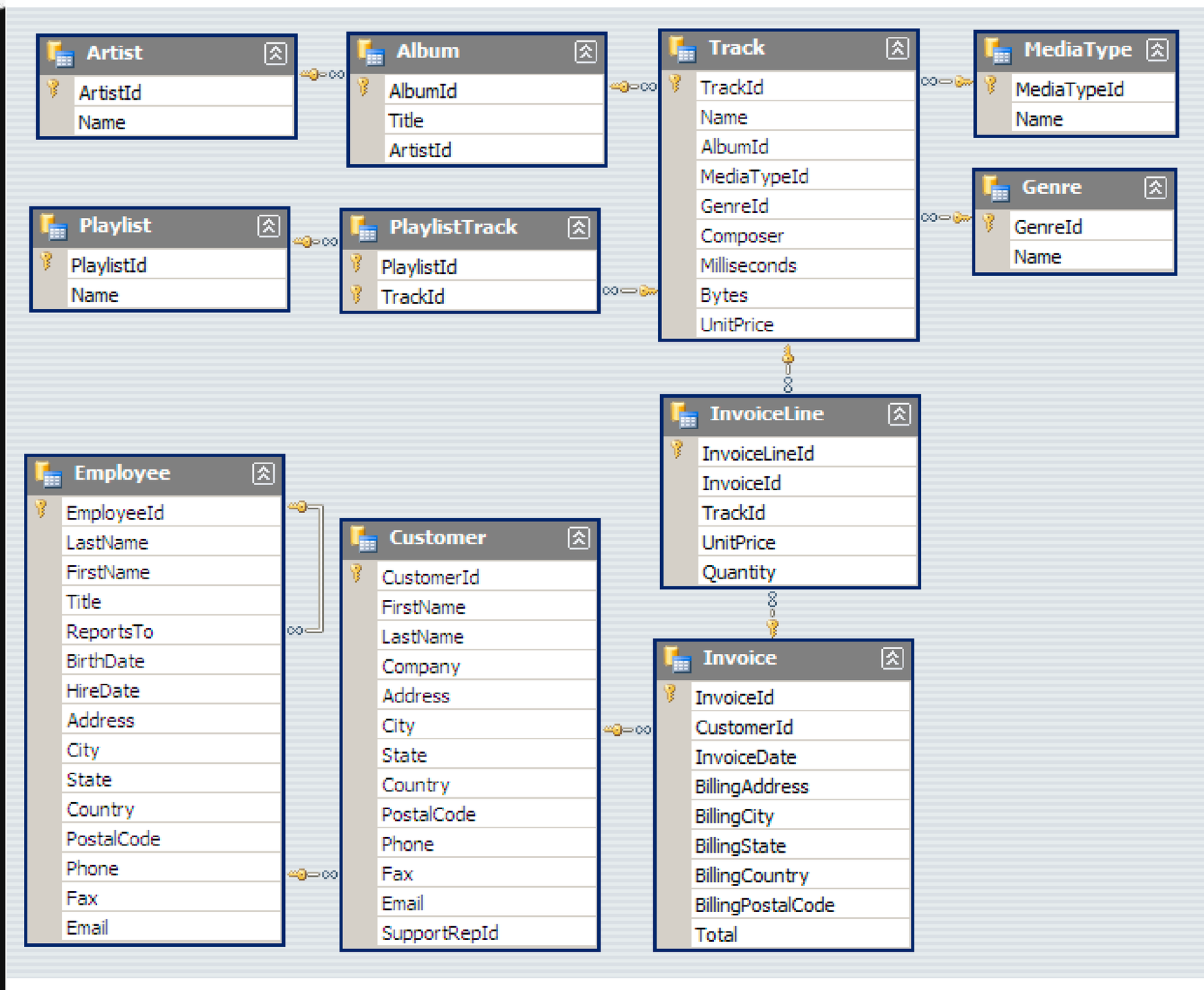
OBJECTIVE

The primary objective of this SQL project is to facilitate sustainable business growth for the music store by addressing existing challenges through a meticulous examination of its dataset. Utilizing SQL, we aim to answer crucial business questions and provide actionable insights. By distilling complex data into meaningful intelligence, our goal is to empower the music store with a strategic understanding of its growth trajectory, enabling informed decision-making and fostering long-term success.

The relational database management system (RDBMS) employed for query execution is **POSTGRESQL**.

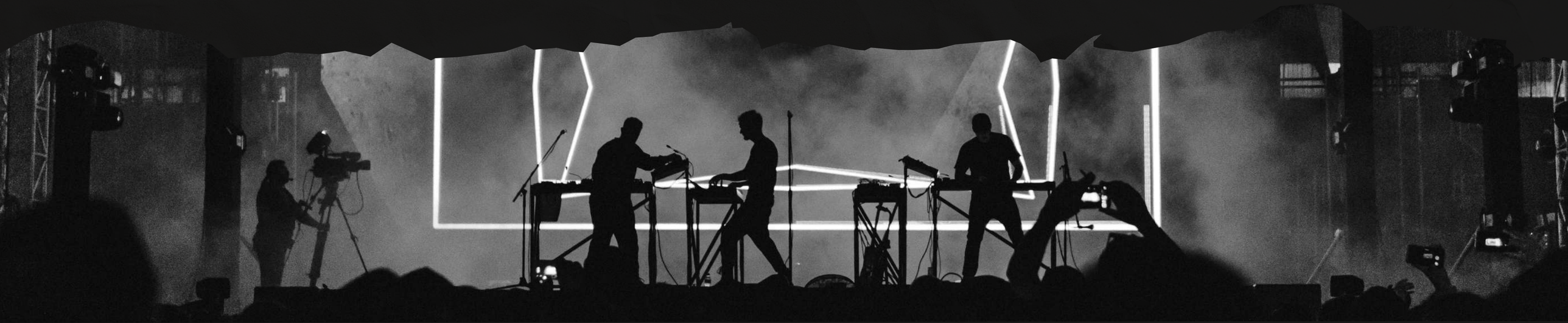
MUSICAL PLAYLIST DATABASE SCHEMA

Please note that the 'album' table is referenced as 'album2' in the database.



TO STREAMLINE THE ASSESSMENT PROCESS, QUESTIONS HAVE BEEN METICULOUSLY CATEGORIZED INTO THREE TIERS OF COMPLEXITY:

- EASY
- MODERATE
- ADVANCED



QUESTION SET 1 - EASY

1. Identify the most senior employee based on their job title.

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

employee_id [PK] character varying (50)	last_name character	first_name character	title character varying (50)	reports_to character var
9	Madan	Mohan ...	Senior General Manager	[null]

2. DETERMINE THE COUNTRIES WITH THE HIGHEST NUMBER OF INVOICES AND PROVIDE A COUNT FOR EACH.

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

	invoice_count bigint	billing_country character varying (30)
1	131	USA
2	76	Canada
3	61	Brazil
4	50	France
5	41	Germany
6	30	Czech Republic
7	29	Portugal
8	28	United Kingdom
9	21	India
10	13	Chile
11	13	Ireland
12	11	Spain
13	11	Finland
14	10	Australia
15	10	Netherlands
16	10	Sweden
17	10	Poland
18	10	Hungary
19	10	Denmark
20	9	Austria

3. Retrieve the top three values of total invoices.

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

	total double precision
1	23.759999999999998
2	19.8
3	19.8

4. Identify the city with the highest sum of invoice totals, as we plan a promotional Music Festival.

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

	billing_city character varying (30)	total_invoice_sum double precision
1	Prague	273.24000000000007

5. Find the customer who has spent the most money.

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

	customer_id [PK] integer	first_name character	last_name character	total_spent double precision
1	5	R ...	Madhav	144.54000000000002

6. Determine the average total for all invoices.

```sql
SELECT AVG(total) AS average_invoice_total
 FROM invoice;
```

	average_invoice_total double precision
1	7.670081433224746

7. Which genre has the highest number of tracks in the database?

```sql
SELECT genre.name, COUNT(track.track_id) AS track_count
FROM genre
JOIN track ON genre.genre_id = track.genre_id
GROUP BY genre.name
ORDER BY track_count DESC
LIMIT 1;
```

	name character varying (120)	track_count bigint
1	Rock	1297

QUESTION SET 2 – MODERATE

1. Retrieve the email, first name, and last name of all Rock Music listeners, ordered alphabetically by email.

```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.invoic
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 character varying (50)	first_name character	last_name character
1	aaronmitchell@yahoo.ca	Aaron	Mitchell
2	alero@uol.com.br	Alexandre	Rocha
3	astrid.gruber@apple.at	Astrid	Gruber
4	bjorn.hansen@yahoo.no	Bjørn	Hansen
5	camille.bernard@yahoo.fr	Camille	Bernard
6	daan_peeters@apple.be	Daan	Peeters
7	diego.gutierrez@yahoo.ar	Diego	Gutiérrez
8	dmiller@comcast.com	Dan	Miller
9	dominiquelefebvre@gmail.c...	Dominique	Lefebvre
10	edfrancis@yachoo.ca	Edward	Francis
11	eduardo@woodstock.com.br	Eduardo	Martins
12	ellie.sullivan@shaw.ca	Ellie	Sullivan
13	emma_jones@hotmail.com	Emma	Jones
14	enrique_munoz@yahoo.es	Enrique	Muñoz
15	fernadaramos4@uol.com.br	Fernanda	Ramos
16	fharris@google.com	Frank	Harris
17	fralston@gmail.com	Frank	Ralston
18	ftremblay@gmail.com	François	Tremblay

2. Identify the top 10 rock bands based on the total number of tracks they have written.

```sql
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;
```

OUTPUT

	artist_id [PK] character varying (50)	name character varying (120)	number_of_songs bigint
1	22	Led Zeppelin	114
2	150	U2	112
3	58	Deep Purple	92
4	90	Iron Maiden	81
5	118	Pearl Jam	54
6	152	Van Halen	52
7	51	Queen	45
8	142	The Rolling Stones	41
9	76	Creedence Clearwater Revival	40
10	52	Kiss	35

3. Retrieve track names and milliseconds for tracks longer than the average song length.

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

OUTPUT

	name character varying (150)	milliseconds 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
6	Battlestar Galactica, Pt. 1	2952702
7	Murder On the Rising Star	2935894
8	Battlestar Galactica, Pt. 3	2927802
9	Take the Celestra	2927677
10	Fire In Space	2926593

4. List all artists with albums containing more than 20 tracks.

```sql
SELECT artist.name, COUNT(track.track_id) AS track_count
FROM artist
JOIN album ON artist.artist_id = album.artist_id
JOIN track ON album.album_id = track.album_id
GROUP BY artist.name
HAVING COUNT(track.track_id) > 20
ORDER BY track_count DESC;
```

OUTPUT

	name character varying (120)	track_count bigint
1	Iron Maiden	213
2	U2	135
3	Led Zeppelin	114
4	Metallica	112
5	Deep Purple	92
6	Lost	92
7	Pearl Jam	67
8	Lenny Kravitz	57
9	Various Artists	56
10	The Office	53
11	Faith No More	52

5. Identify customers who have made purchases in every genre available. Return customer details such as customer_id, first_name, and last_name.

```sql
WITH GenreCount AS (
    SELECT
        c.customer_id,
        COUNT(DISTINCT g.genre_id) AS total_genres
    FROM
        customer c
    CROSS JOIN
        genre g
    LEFT JOIN
        track t ON g.genre_id = t.genre_id
    LEFT JOIN
        invoice_line il ON t.track_id = il.track_id
    GROUP BY
        c.customer_id
)

SELECT
    c.customer_id,
    c.first_name,
    c.last_name
FROM
    customer c
JOIN
    GenreCount gc ON c.customer_id = gc.customer_id
WHERE
    gc.total_genres = (SELECT COUNT(*) FROM genre);
```

OUTPUT

	customer_id [PK] integer	first_name character	last_name character
1	1	Luís	Gonçalves
2	2	Leonie	Köhler
3	3	François	Tremblay
4	4	Bjørn	Hansen
5	5	R	Madhav
6	6	Helena	Holý
7	7	Astrid	Gruber
8	8	Daan	Peeters
9	9	Kara	Nielsen
10	10	Eduardo	Martins
11	11	Alexandre	Rocha
12	12	Roberto	Almeida
13	13	Fernanda	Ramos
14	14	Mark	Philips
15	15	Jennifer	Peterson
16	16	Frank	Harris
17	17	Jack	Smith
18	18	Michelle	Brooks
19	19	Tim	Goyer
20	20	Dan	Miller
21	21	Kathy	Chase

QUESTION SET 3 — ADVANCED

1. Determine the amount spent by each customer on artists, returning 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 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;
```

OUTPUT

	customer_id integer	first_name character	last_name character	artist_name character varying (120)	amount_spent double precision
1	46	Hugh	O'Reilly	Queen	27.719999999999985
2	38	Niklas	Schröder	Queen	18.81
3	3	François	Tremblay	Queen	17.82
4	34	João	Fernandes	Queen	16.830000000000002
5	53	Phil	Hughes	Queen	11.88
6	41	Marc	Dubois	Queen	11.88
7	47	Lucas	Mancini	Queen	10.89
8	33	Ellie	Sullivan	Queen	10.89
9	20	Dan	Miller	Queen	3.96
10	5	R	Madhav	Queen	3.96
11	23	John	Gordon	Queen	2.9699999999999998
12	54	Steve	Murray	Queen	2.9699999999999998
13	31	Martha	Silk	Queen	2.9699999999999998
14	16	Frank	Harris	Queen	1.98
15	17	Jack	Smith	Queen	1.98
16	24	Frank	Ralston	Queen	1.98
17	30	Edward	Francis	Queen	1.98
18	35	Madalena	Sampaio	Queen	1.98
19	26	Hannah	Schneider	Queen	1.98

2. Determine the most popular music Genre for each country, considering the genre with the highest number of purchases.

```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
2, 3, 4
ORDER BY
2 ASC, 1 DESC
)
SELECT *
FROM
popular_genre
WHERE
RowNo <= 1;
```

OUTPUT

	purchases bigint	country character varying (50)	name character varying (120)	genre_id character varying (50)	rowno bigint
1	17	Argentina	Alternative & Punk	4	1
2	34	Australia	Rock	1	1
3	40	Austria	Rock	1	1
4	26	Belgium	Rock	1	1
5	205	Brazil	Rock	1	1
6	333	Canada	Rock	1	1
7	61	Chile	Rock	1	1
8	143	Czech Republic	Rock	1	1
9	24	Denmark	Rock	1	1
10	46	Finland	Rock	1	1
11	211	France	Rock	1	1
12	194	Germany	Rock	1	1
13	44	Hungary	Rock	1	1
14	102	India	Rock	1	1
15	72	Ireland	Rock	1	1
16	35	Italy	Rock	1	1
17	33	Netherlands	Rock	1	1
18	40	Norway	Rock	1	1
19	40	Poland	Rock	1	1

3. Identify the top-spending customer for each country and their total expenditure.

```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
1, 2, 3, 4
ORDER BY
4 ASC, 5 DESC
)
SELECT
customer_id,
first_name,
last_name,
billing_country,
total_spending
FROM
Customer_with_country
WHERE
RowNo <= 1;
```

OUTPUT

	customer_id integer	first_name character	last_name character	billing_country character varying (30)	total_spending double precision
1	56	Diego	Gutiérrez	Argentina	39.6
2	55	Mark	Taylor	Australia	81.18
3	7	Astrid	Gruber	Austria	69.3
4	8	Daan	Peeters	Belgium	60.38999999999999
5	1	Luís	Gonçalves	Brazil	108.89999999999998
6	3	François	Tremblay	Canada	99.99
7	57	Luis	Rojas	Chile	97.02000000000001
8	5	R	Madhav	Czech Republic	144.54000000000002
9	9	Kara	Nielsen	Denmark	37.61999999999999
10	44	Terhi	Hämäläinen	Finland	79.2
11	42	Wyatt	Girard	France	99.99
12	37	Fynn	Zimmermann	Germany	94.05000000000001
13	45	Ladislav	Kovács	Hungary	78.21
14	58	Manoj	Pareek	India	111.86999999999999
15	46	Hugh	O'Reilly	Ireland	114.83999999999997
16	47	Lucas	Mancini	Italy	50.49
17	48	Johannes	Van der Berg	Netherlands	65.34
18	4	Bjørn	Hansen	Norway	72.27000000000001
19	49	Stanisław	Wójcik	Poland	76.22999999999999
20	34	João	Fernandes	Portugal	102.96000000000001
21	50	Enrique	Muñoz	Spain	98.01

4. List the top 5 customers who have made the highest total purchases.

```sql
SELECT
customer.customer_id,
customer.first_name,
customer.last_name,
SUM(invoice.total) AS total_purchases
FROM
customer
JOIN
invoice ON customer.customer_id = invoice.customer_id
GROUP BY
customer.customer_id,
customer.first_name,
customer.last_name
ORDER BY
total_purchases DESC
LIMIT 5;
```

OUTPUT

	customer_id [PK] integer	first_name character	last_name character	total_purchases double precision
1	5	R	Madhav	144.54000000000002
2	6	Helena	Holý	128.7
3	46	Hugh	O'Reilly	114.83999999999997
4	58	Manoj	Pareek	111.86999999999999
5	1	Luís	Gonçalves	108.89999999999998

5. Identify the top 3 countries that have the highest average invoice total. Return the country name and the average invoice total.

```sql
SELECT
billing_country,
AVG(total) AS average_invoice_total
FROM
invoice
GROUP BY
billing_country
ORDER BY
average_invoice_total DESC
LIMIT 3;
```

OUTPUT

	billing_country character varying (30)	average_invoice_total double precision
1	Czech Republic	9.108000000000002
2	Spain	8.91
3	Ireland	8.833846153846151

6. Find the artists who have tracks in multiple genres. Return the artist name and the count of distinct genres.

```sql
SELECT
artist.name AS artist_name,
COUNT(DISTINCT genre.genre_id) AS distinct_genre_count
FROM
artist
JOIN
album ON artist.artist_id = album.artist_id
JOIN
track ON album.album_id = track.album_id
JOIN
genre ON track.genre_id = genre.genre_id
GROUP BY
artist_name
HAVING
COUNT(DISTINCT genre.genre_id) > 1
ORDER BY
distinct_genre_count DESC;
```

OUTPUT

	artist_name character varying (120)	distinct_genre_count bigint
1	Iron Maiden	4
2	Audioslave	3
3	Battlestar Galactica	3
4	Various Artists	3
5	Gilberto Gil	3
6	Jamiroquai	3
7	Lenny Kravitz	3
8	The Office	2
9	Guns N' Roses	2
10	Heroes	2
11	Antônio Carlos Jobim	2
12	U2	2
13	Amy Winehouse	2
14	Lost	2
15	Ozzy Osbourne	2
16	Pearl Jam	2
17	R.E.M.	2
18	Red Hot Chili Peppers	2
19	Eric Clapton	2
20	Faith No More	2
21	Foo Fighters	2

KEY CONTRIBUTIONS

1. Operational Efficiency: The project focuses on enhancing the store's operational efficiency by identifying areas for improvement in inventory management, customer engagement, and marketing strategies.

2. Customer Insights: Through in-depth customer analysis, the project provides insights into customer behaviors, preferences, and spending patterns. This information is crucial for tailoring marketing campaigns and enhancing customer satisfaction.

3. Genre Performance: The analysis of genre-wise performance helps the store optimize its music catalog, ensuring that the most popular genres are well-represented. This strategic move positively impacts sales and customer engagement.

4. International Market Exploration: By examining purchasing trends across different countries, the project enables the music store to explore and capitalize on opportunities in various international markets.

5. Revenue Maximization: The identification of top-spending customers and genres allows the store to focus on high-value segments, maximizing revenue potential.

PRACTICAL IMPLICATIONS

This project transcends theoretical analysis and directly translates insights into practical strategies for the music store. The application of SQL ensures a robust and scalable solution that aligns with real-world business needs.

FUTURE PROSPECTS

As the music store continues to evolve, the insights gained from this data analysis project will serve as a foundation for ongoing improvements and strategic decision-making. The adoption of data-driven practices positions the store at the forefront of the industry, ready to adapt to changing market dynamics.

CONCLUSION

In conclusion, this SQL Music Store Data Analysis project is not just an academic exercise but a practical and impactful endeavor. By leveraging the power of SQL, it goes beyond mere data querying and transforms raw information into actionable intelligence. The result is a comprehensive understanding of the business landscape, paving the way for a successful and sustainable future for the music store.

THANK YOU