Music Database Business Analysis Using Mysql

Project Overview:

In this project, I analysed a fictional music database using SQL to answer business questions and uncover insights related to customers, sales, and music genres. This is an opportunity to apply your SQL skills, including joins, grouping, filtering, sorting, and using CTEs and window functions for advanced queries.

Dataset Structure:

Assume the database has the following tables:

1. **employee:** Contains details of employees, including their job title and seniority levels.

2. **invoice:** Contains all invoices with billing information.

3. **customer:** Contains customer data, such as first and last names and contact information.

4. **invoice_line:** Detailed information on each item within an invoice.

5. **track:** Details of each track, including genre and duration.

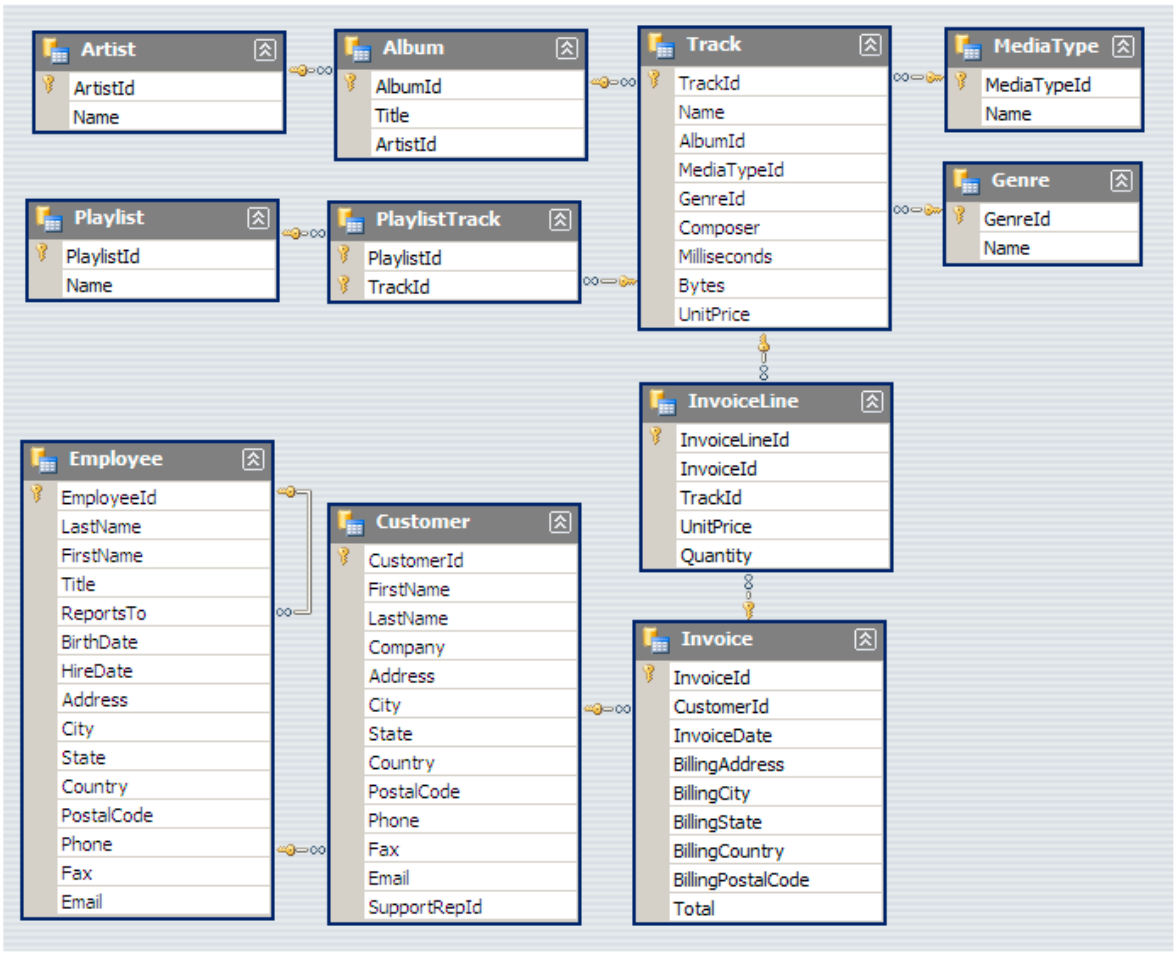
6. **album:** Contains information on music albums.

7. **artist:** Information on artists.

8. **genre:** Information on genres.

Each table has relevant fields like customer_id, invoice_id, track_id, artist_id, etc., for joining tables.

Database Schema:



Section 1:

SQL Query Execution and Output Documentation

1. Easy Level Queries

Q1: Find the most senior employee based on job title.

Code:

```sql
SELECT
    first_name,  -- Employee's first name
    last_name,   -- Employee's last name
    title,       -- Job title of the employee
    levels       -- Seniority level (higher value = more senior)
FROM
    employee     -- Source table containing employee records
ORDER BY
    levels DESC  -- Sort employees from highest to lowest seniority
LIMIT 1;         -- Return only the most senior employee
```

Output:

first_name	last_name	title	levels
Andrew	Adams	General Manager	L6

Q2: Determine which countries have the most invoices.

Code:

```sql
SELECT
    billing_country,          -- Country where the invoice was billed
    COUNT(*) AS Invoice_count -- Total number of invoices per country
FROM
    music.invoice             -- Invoice table in the music schema
GROUP BY
    billing_country           -- Aggregate invoices by country
ORDER BY
    Invoice_count DESC        -- Sort countries by invoice count (highest first)
LIMIT 1;                      -- Return only the country with the most invoices
```

Output:

billing_country	Invoice_count
USA	131

Q3: Identify the top 3 invoice totals.

Code:

```
SELECT *                        -- Retrieve all columns for each invoice
FROM music.invoice              -- Invoice table in the music schema
ORDER BY total DESC             -- Sort invoices by total amount (highest first)
LIMIT 3;                        -- Return the top 3 highest-value invoices
```

Output:

invoice_id	customer_id	invoice_date	billing_address	billing_city	billing_state	billing_country	billing_postal_code	total
183	42	2018-02-09 00:00:00	9, Place Louis Barthou	Bordeaux	None	France	33000	23.759999999999998
92	32	2017-07-02 00:00:00	696 Osborne Street	Winnipeg	MB	Canada	R3L 2B9	19.8
526	5	2020-06-08 00:00:00	Klanova 9/506	Prague	None	Czech Republic	14700	19.8

Q4: Find the city with the highest total invoice amount to determine the best location for a promotional event.

Code:

```
SELECT
    billing_city,
    billing_state,
    billing_country,
    ROUND(SUM(total),2) AS Total_Sales    -- Total sales per location
FROM music.invoice
GROUP BY
    billing_city,
    billing_state,
    billing_country
ORDER BY
    Total_Sales DESC            -- Highest sales first
LIMIT 1;                        -- Top location only
```

Output:

billing_city	billing_state	billing_country	Total_Sales
Prague	None	Czech Republic	273.24

Q5: Identify the customer who has spent the most money.

Code:

```
SELECT
    c.customer_id,
    c.first_name,
    c.last_name,
    ROUND(SUM(i.total), 2) AS total_sales  -- Total amount spent by customer
FROM customer c
JOIN invoice i
    ON c.customer_id = i.customer_id
GROUP BY
    c.customer_id,
    c.first_name,
    c.last_name
ORDER BY
    total_sales desc          -- Highest spender first
LIMIT 1;                      -- Single customer
```

Output:

customer_id	first_name	last_name	total_sales
5	FrantiÅiek	WichterlovÃi	144.54

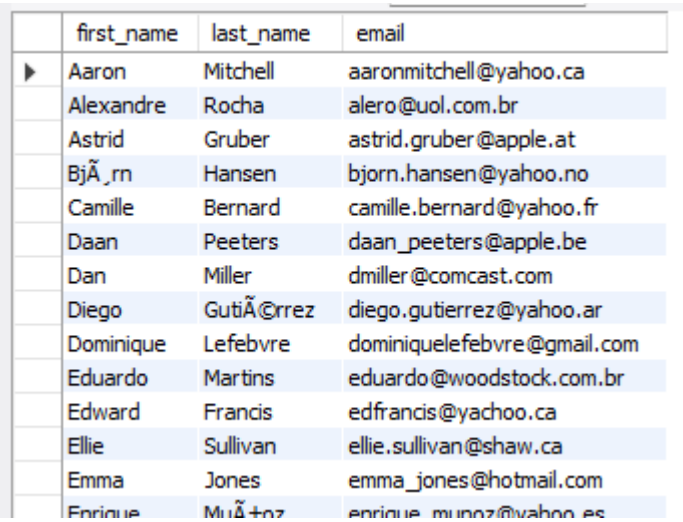
2. Moderate Level Queries

Q1: Find the email, first name, and last name of customers who listen to Rock music.

Code:

```
SELECT first_name, last_name, email
FROM customer c
JOIN invoice i
ON c.customer_id = i.customer_id
JOIN invoice_line il
ON i.invoice_id = il.invoice_id
JOIN Track t
ON il.track_id = t.track_id
JOIN genre g
ON t.genre_id = g.genre_id
where g.name = 'Rock'                -- Only Rock genre purchases
GROUP BY  c.first_name, c.last_name, c.email
ORDER BY c.first_name, c.last_name, c.email;
```

Output:



first_name	last_name	email
Aaron	Mitchell	aaronmitchell@yahoo.ca
Alexandre	Rocha	alero@uol.com.br
Astrid	Gruber	astrid.gruber@apple.at
BjÃ¸rn	Hansen	bjorn.hansen@yahoo.no
Camille	Bernard	camille.bernard@yahoo.fr
Daan	Peeters	daan_peeters@apple.be
Dan	Miller	dmiller@comcast.com
Diego	GutiÃ©rrez	diego.gutierrez@yahoo.ar
Dominique	Lefebvre	dominiquelefebvre@gmail.com
Eduardo	Martins	eduardo@woodstock.com.br
Edward	Francis	edfrancis@yachoo.ca
Ellie	Sullivan	ellie.sullivan@shaw.ca
Emma	Jones	emma_jones@hotmail.com
Enrique	MuÃ±oz	enrique_munoz@yahoo.es

Q2: Identify the top 10 rock artists based on track count.

Code:

```sql
SELECT
    a.name AS artist_name,
    COUNT(t.track_id) AS total_rock_tracks
FROM artist AS a
JOIN album2 AS al
    ON a.artist_id = al.artist_id
JOIN track AS t
    ON al.album_id = t.album_id
JOIN genre AS g
    ON t.genre_id = g.genre_id
WHERE
    g.name = 'Rock'              -- Only Rock tracks
GROUP BY
    a.name
ORDER BY
    total_rock_tracks DESC        -- Most Rock tracks first
LIMIT 10;                         -- Top 10 artists
```

Output:

artist_name	total_rock_tracks
AC/DC	18
Aerosmith	15
Audioslave	14
Led Zeppelin	14
Alanis Morissette	13
Alice In Chains	12
Frank Zappa & Captain Beefheart	9
Accept	4

Q3: Find all track names that are longer than the average track length.

Code:

```
SELECT
    name,
    milliseconds
FROM
    track
WHERE
    milliseconds > (              -- Only tracks longer than average
        SELECT AVG(milliseconds)
        FROM track
    )
ORDER BY
    milliseconds DESC;            -- Longest tracks first
```

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
The Alchemist	509413
Wheels Of Confusion / The Straightener	494524
Book Of Thel	494393
You Oughta Know (Alternate)	491885
Terra	482429

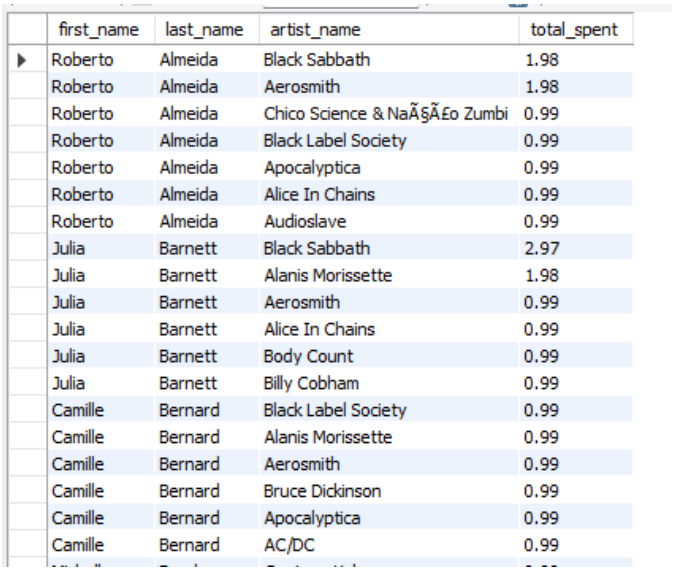
3. Advanced Level Queries (Using CTEs and Window Functions)

Q1: Calculate how much each customer has spent on each artist.

Code:

```
WITH ArtistSales AS (
    -- Step 1: Compute total spent on each invoice line and link it to the artist
    SELECT
        il.invoice_id AS iid,                   -- Invoice ID
        al.artist_id AS aid,                    -- Artist ID
        il.unit_price * il.quantity AS lt       -- Line total (cost)
    FROM invoice_line AS il
    JOIN track AS tr
        ON il.track_id = tr.track_id            -- Link invoice line to track
    JOIN album2 AS al
        ON tr.album_id = al.album_id            -- Link track to album/artist
)
SELECT
    c.first_name AS fn,                         -- Customer first name
    c.last_name AS ln,                          -- Customer last name
    ar.name AS an,                              -- Artist name
    ROUND(SUM(asales.lt), 2) AS total_spent     -- Total spent by customer on this artist
FROM ArtistSales AS asales
JOIN invoice AS i
    ON asales.iid = i.invoice_id                -- Link back to invoice for customer info
JOIN customer AS c
    ON i.customer_id = c.customer_id           -- Get customer details
JOIN artist AS ar
    ON asales.aid = ar.artist_id               -- Get artist details
GROUP BY
    c.customer_id, c.first_name, c.last_name, ar.name  -- Aggregate per customer and artist
ORDER BY
    c.last_name, total_spent DESC;             -- Sort by customer last name, then spending
```

Output:



first_name	last_name	artist_name	total_spent
Roberto	Almeida	Black Sabbath	1.98
Roberto	Almeida	Aerosmith	1.98
Roberto	Almeida	Chico Science & NaÃ§Ã£o Zumbi	0.99
Roberto	Almeida	Black Label Society	0.99
Roberto	Almeida	Apocalyptica	0.99
Roberto	Almeida	Alice In Chains	0.99
Roberto	Almeida	Audioslave	0.99
Julia	Barnett	Black Sabbath	2.97
Julia	Barnett	Alanis Morissette	1.98
Julia	Barnett	Aerosmith	0.99
Julia	Barnett	Alice In Chains	0.99
Julia	Barnett	Body Count	0.99
Julia	Barnett	Billy Cobham	0.99
Camille	Bernard	Black Label Society	0.99
Camille	Bernard	Alanis Morissette	0.99
Camille	Bernard	Aerosmith	0.99
Camille	Bernard	Bruce Dickinson	0.99
Camille	Bernard	Apocalyptica	0.99
Camille	Bernard	AC/DC	0.99

Q2: Determine the most popular music genre for each country based on purchases.

Code:

```sql
WITH CountryGenrePurchases AS (
    -- 1. Count sales for each genre in each country and rank them
    SELECT
        i.billing_country AS bc,          -- Country
        g.name AS gn,                     -- Genre name
        COUNT(il.invoice_line_id) AS pc, -- Purchase count
        ROW_NUMBER() OVER (
            PARTITION BY i.billing_country
            ORDER BY COUNT(il.invoice_line_id) DESC
        ) AS gr                           -- Rank genres within each country
    FROM invoice AS i
    JOIN invoice_line AS il
        ON i.invoice_id = il.invoice_id  -- Link invoice to invoice line
    JOIN track AS t
        ON il.track_id = t.track_id      -- Link invoice line to track
    JOIN genre AS g
        ON t.genre_id = g.genre_id       -- Link track to genre
    GROUP BY i.billing_country, g.name
)
SELECT
    bc AS billing_country,
    gn AS genre_name,
    pc AS purchase_count
FROM CountryGenrePurchases
WHERE gr = 1       -- Keep only the top genre per country
ORDER BY pc desc;       -- Sort by country
```

Output:

billing_country	genre_name	purchase_count
USA	Rock	70
Canada	Rock	57
United Kingdom	Rock	47
Germany	Rock	28
Brazil	Rock	26
France	Rock	26
Portugal	Rock	23
Australia	Rock	18
Czech Republic	Rock	14
Poland	Rock	14
India	Rock	13
Chile	Rock	7
Austria	Rock	6
Denmark	Rock	6
Finland	Rock	6
Netherlands	Rock	6
Belgium	Rock	5

Q3: Identify the top-spending customer for each country.

Code:

```sql
WITH CustomerCountrySpending AS (
    -- 1. Calculate total spending per customer per country and rank them
    SELECT
        c.customer_id AS cid,                  -- Customer ID
        c.first_name AS fn,                    -- First name
        c.last_name AS ln,                     -- Last name
        c.country AS cn,                       -- Customer country
        ROUND(SUM(i.total), 2) AS ts,          -- Total spent
        ROW_NUMBER() OVER (
            PARTITION BY c.country
            ORDER BY SUM(i.total) DESC
        ) AS sr                                 -- Rank customers within country
    FROM customer AS c
    JOIN invoice AS i
        ON c.customer_id = i.customer_id       -- Link customer to invoices
    GROUP BY c.customer_id, c.first_name, c.last_name, c.country
)
SELECT
    cn AS country,
    fn AS first_name,
    ln AS last_name,
    ts AS total_spent
FROM CustomerCountrySpending
WHERE sr = 1        -- Keep only the top spender per country
ORDER BY cn;        -- Sort by country
```

Output:

country	first_name	last_name	total_spent
Argentina	Diego	GutiÃ©rrez	39.6
Australia	Mark	Taylor	81.18
Austria	Astrid	Gruber	69.3
Belgium	Daan	Peeters	60.39
Brazil	LuÃ-s	GonÃ§alves	108.9
Canada	FranÃ§ois	Tremblay	99.99
Chile	Luis	Rojas	97.02
Czech Republic	FrantiÅiek	WichterlovÃi	144.54
Denmark	Kara	Nielsen	37.62
Finland	Terhi	HÃ¤mÃ¤lÃ¤inen	79.2
France	Wyatt	Girard	99.99
Germany	Fynn	Zimmermann	94.05
Hungary	Ladislav	KovÃics	78.21
India	Manoj	Pareek	111.87
Ireland	Hugh	O'Reilly	114.84
Italy	Lucas	Mancini	50.49
Netherlands	Johannes	Van der Berg	65.34
Norway	BjÃ¸rn	Hansen	72.27
Poland	StanisÅ‚aw	WÃ³jcik	76.23
Portugal	JoÃ£o	Fernandes	102.96
Spain	Enrique	MuÃ±oz	98.01
Sweden	Joakim	Johansson	75.24
United Kingdom	Phil	Hughes	98.01
USA	Jack	Smith	98.01

Section 2: Final Presentation and Summary of Findings

Executive Summary:

The SQL analysis revealed critical insights into customer value, regional market performance, and music content trends. The findings provide clear data to support strategic decisions in marketing, events, and inventory management.

I. Sales and Market Strategy (Best Locations)

1. **Insight:** Best Event City
   Key Finding: The city with the highest cumulative invoice total is Prague
   Actionable Strategy: Commit marketing and logistics resources to this location for the next major promotional event to target the highest density of high-value consumers.
2. **Insight:** Local Genre Demand
   Key Finding: The most popular music genre varies significantly by country like Canada prefers Rock, while Norway prefers Latin.
   Actionable Strategy: Localize digital advertising and inventory purchasing based on these confirmed regional genre preferences to maximize conversion rates.

II. Top Customer Insights (High-Value Individuals)

1. **Insight:** Overall Top Spender
   Key Finding: The customer, is FrantiÅ¡ek WichterlovÃ¡ from Czech Republic has the highest lifetime spend of 144.54.
   Actionable Strategy: Immediately enroll this customer into a premium, exclusive loyalty tier to ensure maximum retention and satisfaction.
2. **Insight:** Top Spender Per Country
   Key Finding: The analysis isolated the highest-spending customer in every operational country.
   Actionable Strategy: Launch a local ambassador or influence program, recognizing these individuals to encourage word-of-mouth promotion and loyalty within their respective regions.

3. **Insight:** Rock Enthusiasts
   Key Finding: A segmented list of customers who specifically purchase

Rock music was extracted.
Actionable Strategy: Use this pre-qualified list for highly targeted email/social media campaigns focused on new Rock album releases, ensuring efficient marketing spend.

III. Content and Inventory Insights

1. **Insight:** Top Rock Artists
   Key Finding: The top 10 most purchased Rock artists were identified.
   Recommendation: Prioritize these artists for prominent digital placement, front-of-store product displays, and inventory restocking to capitalize on proven demand.

Conclusion

In conclusion, this project successfully utilized SQL to transform raw database records into actionable business intelligence. By executing complex queries involving Joins, CTEs, and Window Functions, we were able to isolate high-value customer segments and identify key geographical markets. The insights generated from this analysis provide a clear, data-driven roadmap for the company to optimize its marketing budget, personalize customer retention efforts, and maximize revenue growth in the coming quarter.