SQL-Based Analytical Study of Music Store Sales, Customers & Genres

---

Project Overview

This project analyzes the Music Store database using SQL to derive meaningful business insights. It covers customer behavior, revenue trends, artist performance, and genre popularity. The goal is to apply SQL skills ranging from basic SELECT queries to advanced concepts like joins, subqueries, CTEs, and window functions to solve realistic analytics problems.

Key Points:

- Analyzed Music Store dataset using SQL to generate business insights.
- Identified top customers, best-performing cities, and highest-selling genres.
- Ranked artists based on sales and track production.
- Used joins, subqueries, window functions, and CTEs for advanced analysis.
- Extracted revenue patterns and customer behavior insights.

Easy Level

🟦 Query 1: Senior-Most Employee

Find the employee with the highest job level (senior-most).

```sql
SELECT title, last_name, first_name
FROM employee
ORDER BY levels DESC
LIMIT 1;
```

Output:

	title	last_name	first_name
▶	General Manager	Adams	Andrew

🟩 Query 2: Countries With the Most Invoices

Which countries generate the most orders?

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

Output:

c	billing_country
131	USA
76	Canada
61	Brazil
50	France
41	Germany
30	Czech Republic
29	Portugal
28	United Kingdom
21	India
13	Ireland
13	Chile
11	Finland
11	Spain
10	Poland
10	Denmark
10	Australia
10	Hungary
10	Sweden
10	Netherlands
9	Norway
9	Italy
9	Austria
7	Belgium
5	Argentina

🟧 Query 3: City That Generates the Most Revenue

Where should we plan our next Music Festival?

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

Output:

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
106.91999999999999	BrasÃ-lia
102.96000000000001	Lisbon
99.99	Bordeaux
99.99	MontrÃ©al
98.01	Madrid
98.01	Redmond
97.02000000000001	Santiago
94.05000000000001	Frankfurt
76.23	Madison
76.22999999999999	Warsaw
75.24000000000001	Yellowknife
75.24	Stockholm
73.25999999999999	Dijon
72.27000000000001	Oslo
72.27	Salt Lake City
71.28	Chicago
71.28	Bangalore
70.28999999999999	Winnipeg
69.3	Vienne
66.33	Vancouver
66.33	Boston
65.34	Amsterdam
64.35	Lyon
62.370000000000005	Halifax
60.38999999999999	Brussels
54.449999999999996	Cupertino
50.49	Rome
40.59	Toronto
39.6	Buenos Aires
37.61999999999999	Copenhagen
29.699999999999996	Edmonton

Query 4: What are top 3 values of total invoice?

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

Output:

	total
▶	23.759999999999998
	19.8
	19.8

🟥 Query 5: Best Customer (Highest Spender)

```
SELECT c.customer_id, c.first_name, c.last_name,
       SUM(i.total) AS Total
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 DESC
LIMIT 1;
```

Output:

	customer_id	first_name	last_name	Total
▶	5	FrantiÅiek	WichterlovÃi	144.54000000000002

Moderate Level

Q1: 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

```
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;
```

Output:

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
edfrancis@yachoo.ca	Edward	Francis
eduardo@woodstock.com.br	Eduardo	Martins
ellie.sullivan@shaw.ca	Ellie	Sullivan
emma_jones@hotmail.com	Emma	Jones
enrique_munoz@yahoo.es	Enrique	MuÃ±oz
fernadaramos4@uol.com.br	Fernanda	Ramos
fharris@google.com	Frank	Harris
fralston@gmail.com	Frank	Ralston
frantisekw@jetbrains.com	FrantiÅ¡ek	WichterlovÃ¡
ftremblay@gmail.com	FranÃ§ois	Tremblay
fzimmermann@yahoo.de	Fynn	Zimmermann
hannah.schneider@yahoo.de	Hannah	Schneider
hholy@gmail.com	Helena	HolÃ½
hleacock@gmail.com	Heather	Leacock
hughoreilly@apple.ie	Hugh	O'Reilly
isabelle_mercier@apple.fr	Isabelle	Mercier
jacksmith@microsoft.com	Jack	Smith
jenniferp@rogers.ca	Jennifer	Peterson
jfernandes@yahoo.pt	JoÃ£o	Fernandes
joakim.johansson@yahoo.se	Joakim	Johansson
johavanderberg@yahoo.nl	Johannes	Van der Berg
johngordon22@yahoo.com	John	Gordon
jubarnett@gmail.com	Julia	Barnett
kachase@hotmail.com	Kathy	Chase
kara.nielsen@jubii.dk	Kara	Nielsen
ladislav_kovacs@apple.hu	Ladislav	KovÃ¡cs
leonekohler@surfeu.de	Leonie	KÃ¶hler
lucas.mancini@yahoo.it	Lucas	Mancini
luisg@embraer.com.br	LuÃs	GonÃ§alves
luisrojas@yahoo.cl	Luis	Rojas
manoj.pareek@rediff.com	Manoj	Pareek
marc.dubois@hotmail.com	Marc	Dubois
mark.taylor@yahoo.au	Mark	Taylor
marthasilk@gmail.com	Martha	Silk
masampaio@sapo.pt	Madalena	Sampaio
mphilips12@shaw.ca	Mark	Philips
nschroder@surfeu.de	Niklas	SchrÃ¶der
patrick.gray@aol.com	Patrick	Gray
phil.hughes@gmail.com	Phil	Hughes
ricunningham@hotmail.com	Richard	Cunningham
rishabh_mishra@yahoo.in	Rishabh	Mishra
robbrown@shaw.ca	Robert	Brown
roberto.almeida@riotur.gov.br	Roberto	Almeida
stanisÅ‚aw.wÃ³jcik@wp.pl	StanisÅ‚aw	WÃ³jcik
steve.murray@yahoo.uk	Steve	Murray
terhi.hamalainen@apple.fi	Terhi	HÃ¤mÃ¤lÃ...
tgoyer@apple.com	Tim	Goyer
vstevens@yahoo.com	Victor	Stevens
wyatt.girard@yahoo.fr	Wyatt	Girard

Q2: 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;
```

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
2	Accept	4

Q3: 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;
```

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
Snoopy's search-Red baron	456071
Sozinho (Hitmakers Classic Mix)	436636
Master Of Puppets	436453

Some more Rows…….

Advanced

Q1: Find how much amount spent by each customer on artists? Write a query to return customer name, artist name and total spent.

```
WITH best_selling_artist AS (
    SELECT ar.artist_id, ar.name AS artist_name,
         SUM(il.unit_price * il.quantity) AS total_sales
    FROM invoice_line il
    JOIN track t ON t.track_id = il.track_id
    JOIN album2 a ON a.album_id = t.album_id
    JOIN artist ar ON ar.artist_id = a.artist_id
    GROUP BY ar.artist_id, ar.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 a ON a.album_id = t.album_id
JOIN best_selling_artist bsa ON bsa.artist_id = a.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
35	Madalena	Sampaio	AC/DC	2.9699999999999998
44	Terhi	HÃ¤mÃ¤lÃ...	AC/DC	2.9699999999999998
9	Kara	Nielsen	AC/DC	1.98
34	JoÃ£o	Fernandes	AC/DC	1.98
57	Luis	Rojas	AC/DC	1.98
27	Patrick	Gray	AC/DC	1.98
20	Dan	Miller	AC/DC	1.98
30	Edward	Francis	AC/DC	1.98
5	FrantiÅ¡ek	WichterlovÃ¡	AC/DC	1.98
47	Lucas	Mancini	AC/DC	0.99
43	Isabelle	Mercier	AC/DC	0.99
19	Tim	Goyer	AC/DC	0.99
39	Camille	Bernard	AC/DC	0.99

	8	Daan	Peeters	AC/DC	0.99
	15	Jennifer	Peterson	AC/DC	0.99
	58	Manoj	Pareek	AC/DC	0.99
	46	Hugh	O'Reilly	AC/DC	0.99
	32	Aaron	Mitchell	AC/DC	0.99
	45	Ladislav	KovÃics	AC/DC	0.99
	29	Robert	Brown	AC/DC	0.99
	26	Richard	Cunningham	AC/DC	0.99
	17	Jack	Smith	AC/DC	0.99
	14	Mark	Philips	AC/DC	0.99
	2	Leonie	KÃ¶hler	AC/DC	0.99
	56	Diego	GutiÃ©rrez	AC/DC	0.99
	48	Johannes	Van der Berg	AC/DC	0.99
	13	Fernanda	Ramos	AC/DC	0.99
	55	Mark	Taylor	AC/DC	0.99
	7	Astrid	Gruber	AC/DC	0.99
	59	Rishabh	Mishra	AC/DC	0.99
	10	Eduardo	Martins	AC/DC	0.99