

DIGITAL MUSIC STORE

ANALYSIS PROJECT





OBJECTIVE

The goal of this project is to analyze the music store database using SQL to uncover insights that can enhance decision-making and improve business outcomes. Specifically, the objectives include:

1. Employee and Customer Analysis:
2. Sales and Invoice Analysis:
3. Music Preferences:
4. Detailed Spending Insights:

Through this analysis, the project will provide actionable recommendations to improve customer engagement, optimize music offerings, and plan strategic promotions for revenue growth.





PROBLEM STATEMENT

The music industry relies heavily on data to make informed decisions about customer preferences, sales trends, and marketing strategies. A music store's database contains valuable information, such as customer purchase histories, artist contributions, and track characteristics. However, extracting actionable insights from this data remains a challenge due to the large volume and complexity of the information. This project aims to address key business questions, such as identifying top-performing cities, artists, genres, and customers, to help the store optimize its operations and promotional strategies.





QUESTION SET 1:EASY

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

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

title character varying (50)	last_name character (50)	first_name character (50)
Senior General Manager	Madan	Mohan

Q2: Which countries have the most Invoices?

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

bigint	character varying (
131	USA
76	Canada
61	Brazil
50	France
41	Germany
30	Czech Republic
29	Portugal
28	United Kingdom
21	India
13	Chile
13	Ireland
11	Spain
11	Finland
10	Australia
10	Netherlands
10	Sweden
10	Poland
10	Hungary
10	Denmark
9	Austria
9	Norway



Q3: What are top 3 values of total invoice?

```
SELECT total  
FROM invoice  
ORDER BY total DESC
```

	total	double precision	locked
1	23.759999999999998	23.759999999999998	
2		19.8	
3		19.8	

Q4: 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 billing_city,SUM(total) AS InvoiceTotal  
      FROM invoice  
     GROUP BY billing_city  
ORDER BY InvoiceTotal DESC  
      LIMIT 1;
```

billing_city	invoicetotal	locked
Prague	273.2400000000000	

Q5: 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(total) AS  
      total_spending  
     FROM customer  
JOIN invoice ON customer.customer_id = invoice.customer_id  
    GROUP BY customer.customer_id  
ORDER BY total_spending DESC  
      LIMIT 1;
```

customer_id	customer_id	first_name	last_name	total_spending
1	[PK] integer	R	Madhav	144.5400000000002



QUESTION SET 2: INTERMEDIATE

Q1: 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 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
11	The Long Patrol	2925008
12	The Magnificent Warriors	2924716
13	The Living Legend, Pt. 1	2924507
14	The Gun On Ice Planet Zero, Pt. 2	2924341
15	The Hand of God	2924007
16	Experiment In Terra	2923548
17	War of the Gods, Pt. 2	2923381
18	The Living Legend, Pt. 2	2923298
19	War of the Gods, Pt. 1	2922630
20	Lost Planet of the Gods, Pt. 1	2922547
21	Baltar's Escape	2922088
22	The Lost Warrior	2920045



	email character varying (50)	firstname character (50)	lastname character (50)	name character varying (120)
1	aaronmitchell@yahoo.ca	Aaron	Mitchell	Rock
2	alero@uol.com.br	Alexandre	Rocha	Rock
3	astrid.gruber@apple.at	Astrid	Gruber	Rock
4	bjorn.hansen@yahoo.no	Bjørn	Hansen	Rock
5	camille.bernard@yahoo.fr	Camille	Bernard	Rock
6	daan_peeters@apple.be	Daan	Peeters	Rock
7	diego.gutierrez@yahoo.ar	Diego	Gutiérrez	Rock
8	dmiller@comcast.com	Dan	Miller	Rock
9	dominiquelefebvre@gmail.c...	Dominique	Lefebvre	Rock
10	edfrancis@yachoo.ca	Edward	Francis	Rock
11	eduardo@woodstock.com.br	Eduardo	Martins	Rock
12	ellie.sullivan@shaw.ca	Ellie	Sullivan	Rock
13	emma_jones@hotmail.com	Emma	Jones	Rock
14	enrique_munoz@yahoo.es	Enrique	Muñoz	Rock
15	fernadaramos4@uol.com.br	Fernanda	Ramos	Rock
16	fharris@google.com	Frank	Harris	Rock
17	gabriela.santos@gmail.com	Gabriela	Santos	Rock

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 AS Email,first_name AS FirstName, last_name AS LastName, genre.name AS Name
FROM customer
JOIN invoice ON invoice.customer_id = customer.customer_id
JOIN invoiceline ON invoiceline.invoice_id = invoice.invoice_id
JOIN track ON track.track_id = invoiceline.track_id
JOIN genre ON genre.genre_id = track.genre_id
WHERE genre.name LIKE 'Rock'
ORDER BY email;
```

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.

```
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;
```

	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

QUESTION SET 3 : ADVANCE

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 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;
```

	customer_id integer	first_name character (50)	last_name character (50)	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	36	Hannah	Schneider	Queen	1.98
20	11	Alexandre	Rocha	Queen	1.98
21	8	Daan	Peeters	Queen	1.98
22	42	Wivatt	Girard	Queen	1.98





	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
20	108	Portugal	Rock	1	1
21	46	Spain	Rock	1	1
22	60	Sweden	Rock	1	1

Total rows: 24

Query complete 00:00:00.235

Q3: 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. */

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 * FROM Customer_with_country WHERE RowNo <= 1

Q2: We want to find out the most popular music Genre for each country. We determine the most popular genre 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. */

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

	customer_id integer	first_name character (50)	last_name character (50)	billing_country character varying (30)	total_spending double precision	rowno bigint
1	56	Diego	...	Gutiérrez	39.6	1
2	55	Mark	...	Taylor	81.18	1
3	7	Astrid	Gruber	...	Austria	69.3
4	8	Daan	Peeters	...	Belgium	60.38999999999999
5	1	Luís	Gonçalves	...	Brazil	108.8999999999998
6	3	François	Tremblay	...	Canada	99.99
7	57	Luis	Rojas	Chile	97.0200000000001	1
8	5	R	Madhav	Czech Republic	144.5400000000002	1
9	9	Kara	Nielsen	Denmark	37.6199999999999	1
10	44	Terhi	Hämäläinen	Finland	79.2	1
11	42	Wyatt	Girard	France	99.99	1
12	37	Fynn	Zimmermann	Germany	94.0500000000001	1
13	45	Ladislav	Kovács	Hungary	78.21	1
14	58	Manoj	Pareek	India	111.8699999999999	1
15	46	Hugh	O'Reilly	Ireland	114.8399999999997	1
16	47	Lucas	Mancini	Italy	50.49	1
17	48	Johannes	Van der Berg	Netherlands	65.34	1
18	4	Bjørn	Hansen	Norway	72.2700000000001	1
19	49	Stanisław	Wójcik	Poland	76.2299999999999	1
20	34	João	Fernandes	Portugal	102.9600000000001	1
21	50	Enrique	Muñoz	Spain	98.01	1
22	51	Inokim	Johansson	Sweden	75.24	1

Total rows: 24

Query complete 00:00:00.178



THANK YOU

