Music Store Analysis

"Complete SQL Project"

Prepared By Punit Pal

punitpalofficial@gmail.com

Project Overview

- The Music Store Data Analysis Project is a structured SQL-based analytical project that focuses on extracting meaningful insights from a relational music store database.
- The database simulates a real-world digital music store, containing customer transactions, employee details, music tracks, albums, artists, genres, and invoices.
- Using SQL queries ranging from basic to advanced level, this project answers business-oriented questions related to sales performance, customer behavior, genre popularity, and revenue distribution.

punitpalofficial@gmail.com

Project Objectives

The main objectives of this project are:

- To analyze customer purchasing behavior
- To identify high-revenue customers, cities, and countries
- To find top-performing artists and genres
- To practice SQL queries from beginner to advanced level
- To convert raw database records into actionable business insights

punitpalofficial@gmail.com

Basic: 1

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

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

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

employee_id	last_name	first_name	title	reports_to	levels	birthdate
1	Adams	Andrew	General Manager	9	L6	18-02-1962 00:00

punitpalofficial@gmail.com

Basic: 2

```
7   -- Q2: Which country have the most invoices?
8
9   SELECT billing_country, count(invoice_id)
10  FROM music_store.invoice
11  group by billing_country
12  order by count(invoice_id) desc
13  limit 5;
14
15
16
17
```

Result Grid | Filter Rows: | Export: | Wrap

billing_country	count(invoice_id)
USA	131
Canada	76
Brazil	61
France	50
Germany	41

punitpalofficial@gmail.com

Basic: 3

```
20      -- Q3: What are the top 3 values of total invoice?
21
22  •   select total from music_store.invoice
23      order by total desc
24      limit 3;
25
26
27
28
29
30
```

Result Grid | Filter Rows: | Export: | Wrap Cell Conte

total
23.759999999999998
19.8
19.8

punitpalofficial@gmail.com

Basic: 4

```
31      -- Q4: Which city has the best customers? We woild like to throw a promotional Music Festival in the city
32      -- we made the most money. Write a query that returns one city that has the highest sum of invoice totals.
33      -- Return both the city name & sum of all invoice totals.
34
35      select billing_city, sum(total)
36      from music_store.invoice
37      group by billing_city
38      order by sum(total) desc
39      limit 10;
40
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

billing_city	sum(total)
Prague	273.24000000000007
Mountain View	169.29
London	166.32
Berlin	158.4
Paris	151.47
São Paulo	129.69
Dublin	114.83999999999997
Delhi	111.86999999999999
São José dos Campos	108.89999999999998
Brasília	106.91999999999999

punitpalofficial@gmail.com

Basic: 5

```
43    -- Q5: Who is the best customer? The customer who has spent the most money will be declared the best customer.
44    -- Write a query that returns the person who has spent the most money.
45
46 •  select customer.customer_id, customer.first_name, customer.last_name, sum(invoice.total)
47    from music_store.customer
48    join music_store.invoice
49    on customer.customer_id= invoice.customer_id
50    group by customer.customer_id, customer.first_name, customer.last_name
51    order by sum(invoice.total) desc
52    limit 1;
53
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	first_name	last_name	sum(invoice.total)
5	František	Wichterlová	144.54000000000002

punitpalofficial@gmail.com

Moderate: 1

```sql
-- Q1: Write query to return the email, first name, last name, & Genre of all Rock Music listeners.
-- Return your list ordered alphabatically by email starting with A

select distinct customer.first_name, customer.last_name, customer.email
from music_store.customer
join music_store.invoice
on customer.customer_id = invoice.invoice_id
join music_store.invoice_line
on invoice.invoice_id = invoice_line.invoice_id
where track_id in(
select track_id
from music_store.track
join music_store.genre
on track.genre_id = genre.genre_id
where genre.name like 'Rock')
order by customer.email;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

first_name	last_name	email
Aaron	Mitchell	aaronmitchell@yahoo.ca
Alexandre	Rocha	alero@uol.com.br
Astrid	Gruber	astrid.gruber@apple.at
Bjørn	Hansen	bjorn.hansen@yahoo.no
Camille	Bernard	camille.bernard@yahoo.fr

punitpalofficial@gmail.com

Moderate: 2

```
73    -- Q2: Let's invite the artist who have written the most rock music in our dataset.
74    -- Write a query that returns the Aritst name and total track count of the top 10 rock brands
75
76 •  select artist.name, artist.artist_id, count(artist.artist_id) as number_of_songs
77    from music_store.track
78    join music_store.album2
79    on track.album_id = album2.album_id
80    join music_store.artist
81    on album2.artist_id = artist.artist_id
82    join music_store.genre
83    on track.genre_id = genre.genre_id
84    where genre.name like 'Rock'
85    group by artist.artist_id, artist.name
86    order by number_of_songs desc
87    limit 10;
88
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	artist_id	number_of_songs
Led Zeppelin	22	114
U2	150	112
Deep Purple	58	92
Iron Maiden	90	81
Pearl Jam	118	54

punitpalofficial@gmail.com

Moderate: 3

```
92    -- Q3: Return all the track names that have a song length longer than the average song length.
93    -- Retrun the name and Milliseconds for each track.
94    -- Order by the song length with the longest songs listed first.
95
96    select track.name, track.milliseconds
97    from music_store.track
98    where milliseconds >
99    (select avg(track.milliseconds)
100   from music_store.track)
101   order by track.milliseconds desc;
102
103
104
105
106
107
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	milliseconds
Occupation / Precipice	5286953
Through a Looking Glass	5088838
Greetings from Earth, Pt. 1	2960293
The Man With Nine Lives	2956998
Battlestar Galactica, Pt. 2	2956081

punitpalofficial@gmail.com

Advance: 1(a)

```sql
-- 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 album2 ON album2.album_id = track.album_id
    JOIN artist ON artist.artist_id = album2.artist_id
    GROUP BY 1,2
    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
```

punitpalofficial@gmail.com

Advance: 1(b)

```
126     SUM(il.unit_price*il.quantity) AS amount_spent
127     FROM invoice i
128     JOIN customer c ON c.customer_id = i.customer_id
129     JOIN invoice_line il ON il.invoice_id = i.invoice_id
130     JOIN track t ON t.track_id = il.track_id
131     JOIN album2 alb ON alb.album_id = t.album_id
132     JOIN best_selling_artist bsa ON bsa.artist_id = alb.artist_id
133     GROUP BY 1,2,3,4
134     ORDER BY 5 DESC;
135
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	first_name	last_name	artist_name	amount_spent
46	Hugh	O'Reilly	Queen	27.719999999999985
38	Niklas	Schröder	Queen	18.81
3	François	Tremblay	Queen	17.82
34	João	Fernandes	Queen	16.830000000000002
53	Phil	Hughes	Queen	11.88
41	Marc	Dubois	Queen	11.88
47	Lucas	Mancini	Queen	10.89
33	Ellie	Sullivan	Queen	10.89
20	Dan	Miller	Queen	3.96
5	František	Wichterlová	Queen	3.96

Result 17

punitpalofficial@gmail.com

Advance: 2(a)

```sql
-- Q2: 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 purchase.
-- 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.

select g.genre_id, g.name, inv.billing_country, (inv.total) as total
from music_store.genre as g
join music_store.track as t
on g.genre_id = t.genre_id
join music_store.invoice_line as inl
on inl.track_id = t.track_id
join music_store.invoice as inv
on inv.invoice_id = inl.invoice_id
join music_store.customer as c
on c.customer_id = inv.customer_id;
```

punitpalofficial@gmail.com

Advance: 2(b)

```sql
-- Method 1:

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 music_store.invoice_line
join music_store.invoice on invoice.invoice_id = invoice_line.invoice_id
join music_store.customer on customer.customer_id = invoice.customer_id
join music_store.track on track.track_id = invoice_line.track_id
join music_store.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;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

purchases	country	name	genre_id	RowNo
17	Argentina	Alternative & Punk	4	1
34	Australia	Rock	1	1
40	Austria	Rock	1	1
26	Belgium	Rock	1	1
205	Brazil	Rock	1	1

punitpalofficial@gmail.com

Advance: 2(c)

```sql
-- Method 2:

with recursive
sales_per_country as(
select count(*) as purchases_per_genre, customer.country, genre.name, genre.genre_id
from music_store.invoice_line
join music_store.invoice on invoice.invoice_id = invoice_line.invoice_id
join music_store.customer on customer.customer_id = invoice.customer_id
join music_store.track on track.track_id = invoice_line.track_id
join music_store.genre on genre.genre_id = track.genre_id
group by 2,3,4
order by 2
),
max_genre_per_country as (select Max(purchases_per_genre) as max_genre_number, country
from sales_per_country
group by 2
order by 2)

select sales_per_country.*
from sales_per_country
join max_genre_per_country on sales_per_country.country = max_genre_per_country.country
where sales_per_country.purchases_per_genre = max_genre_per_country.max_genre_number;
```

punitpalofficial@gmail.com

Advance: 2(d)

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

purchases_per_genre	country	name	genre_id
17	Argentina	Alternative & Punk	4
34	Australia	Rock	1
40	Austria	Rock	1
26	Belgium	Rock	1
205	Brazil	Rock	1

Result 21

punitpalofficial@gmail.com

Advance: 3(a)

```sql
-- 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

-- Method 1:

with recursive
customer_with_country as (
select customer.customer_id, first_name, last_name, billing_country, sum(total) as total_spending
from music_store.invoice
join music_store.customer on customer.customer_id = invoice.customer_id
group by 1,2,3,4
order by 2,3 desc),

country_max_spending as(
select billing_country, max(total_spending) as max_spending
from customer_with_country
group by billing_country)
select cc.billing_country, cc.total_spending, cc.first_name, cc.last_name, cc.customer_id
from customer_with_country cc
join country_max_spending ms
on cc.billing_country = ms.billing_country
where cc.total_spending = ms.max_spending
order by 1;
```

punitpalofficial@gmail.com

Advance: 3(b)

```
217     -- Method 2:
218
219     with customer_with_country as (
220     select customer.customer_id, first_name, last_name, billing_country, sum(total) as total_spending,
221     row_number() over(partition by billing_country order by sum(total) desc) as Rowno
222     from music_store.invoice
223     join music_store.customer on customer.customer_id = invoice.customer_id
224     group by 1,2,3,4
225     order by 4 asc, 5 desc)
226     select * from customer_with_country where RowNo <=1
227
228
229
230
231
232
233
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	first_name	last_name	billing_country	total_spending	Rowno
56	Diego	Gutiérrez	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	Luís	Gonçalves	Brazil	108.89999999999998	1

Result 22

punitpalofficial@gmail.com

Results & Findings

- Identified top customers contributing maximum revenue
- Found Rock as the most popular genre across multiple countries
- Determined high-revenue cities ideal for promotional events
- Highlighted top-selling artists
- Observed regional variation in music preferences
- A small percentage of customers generate a large portion of revenue
- Music preference varies significantly by country
- Certain genres dominate global sales
- Artist popularity directly impacts overall revenue

punitpalofficial@gmail.com

Recommendations

- Target marketing campaigns in high-revenue cities
- Promote country-specific popular genres
- Introduce loyalty programs for top customers
- Collaborate with top-performing artists
- Personalize recommendations based on customer behavior

punitpalofficial@gmail.com

👨‍💻 About Me:

Punit Pal

Aspiring Data Analyst

- Skills: MySQL, Excel, Power BI, Python, Data Cleaning, Data Visualization, Canva, Google Sheets, MS Office, Google Suite.

📫 Contact

- GitHub: https://github.com/punitpalofficial
- LinkedIn: https://www.linkedin.com/in/punit-pal/
- Email: punitpalofficial@gmail.com
- Phone: 7983555927

Thank you!

Prepared By Punit Pal

punitpalofficial@gmail.com