

Sonic Bloom Music Store Analysis

Introduction

The Sonic Bloom Music Store is a dynamic platform offering a wide range of music tracks, albums, and genres to a global customer base. This analysis aims to provide actionable insights into customer behavior, sales patterns, and popular music trends using the store's transactional and customer data.

Project Background

Sonic Bloom Music Store collects a wealth of data from its operations, including customer details, invoices, and track metadata. This data is crucial for understanding customer preferences, identifying top-performing artists and genres, and optimizing marketing strategies.

Analysis Goals and Objectives

- **Goal 1:** Identify key customer demographics and spending patterns.
- **Goal 2:** Determine the top-performing genres, artists, and tracks.
- **Goal 3:** Provide actionable insights for promotional events and marketing campaigns.

Technology Used

- Database Management System: PostgreSQL
- Programming Language: SQL

Data Source and Collection

The data was sourced from transactional CSV files generated by the Sonic Bloom Music Store. These files include:

- customer.csv: Customer details
- invoice.csv: Invoice data
- track.csv: Track metadata
- genre.csv: Genre information
- artist.csv: Artist details
- invoice_line.csv: Invoice line items

Creating Database and Copying the CSV Data in Database

1. Database Creation:

```
CREATE DATABASE sonic_bloom;
```

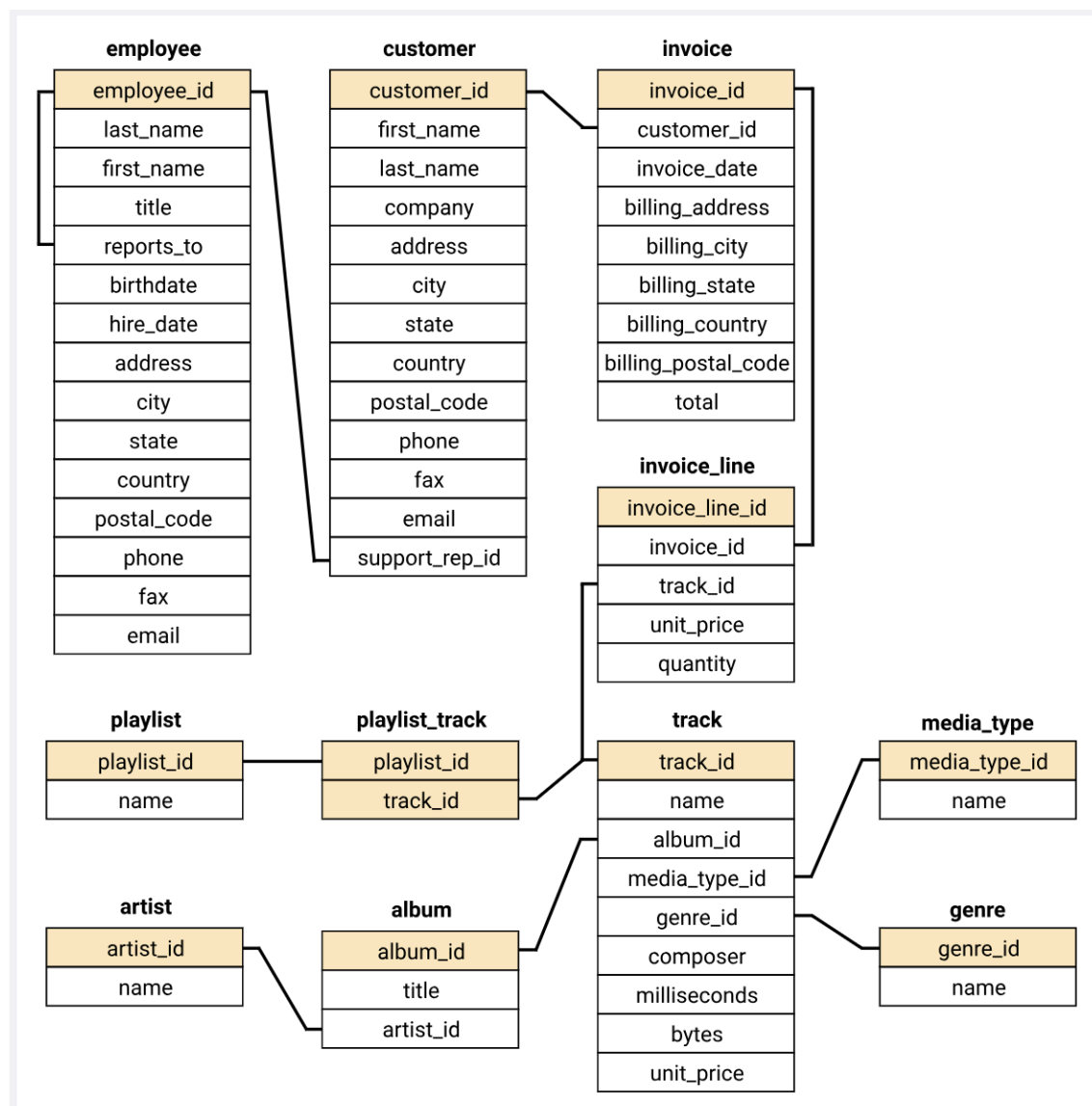
2. Table Creation and Data Import: Tables were created to reflect the structure of the CSV files.

- a. Creating tables using "CREATE TABLE" command and add the same column names as in the csv file.
- b. Importing data from csv files using the command "COPY table_name(columns_name) FROM (file_location) DELIMITER ',' CSV HEADER;

Creating the Schema Diagram for the Database

The schema diagram includes the following key tables and their relationships:

- **Customer:** Linked to Employee and Invoice
- **Invoice:** Linked to Invoice_Line
- **Invoice_Line:** Linked to Track
- **Track:** Linked to Playlist Track, Album, Media Type and Genre
- **Playlist Track:** Linked to Playlist
- **Album:** Linked to Artist



Questions and SQL Queries

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

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

This query identifies the senior-most employee by ordering employees based on their levels column in descending order.

Output:

	first_name character (50) 🔒	last_name character (50) 🔒	levels character varying (10) 🔒
1	Mohan	Madan	L7

Q2: Which countries have the most invoices?

```
SELECT  
  
    billing_country,  
  
    COUNT(*) as total_invoice  
  
FROM invoice  
  
GROUP BY billing_country  
  
ORDER BY total_invoice DESC;
```

This query aggregates invoices by country and orders them to identify the top countries.

Output:


	billing_country character varying (30) 🔒	total_invoice bigint 🔒
1	USA	131
2	Canada	76
3	Brazil	61
4	France	50
5	Germany	41
6	Czech Republic	30
Total rows: 24		Query complete 00:00:00.388

Q3: What are the top 3 values of total invoice?

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

This query returns the top three invoice totals by sorting the total column in descending order.

Output:



	total double precision 
1	23.759999999999998
2	19.8
3	19.8

Q4: Which city has the best customers?

```
SELECT
    billing_city,
    SUM(total) as invoice_total
FROM invoice
GROUP BY billing_city
ORDER BY invoice_total DESC
LIMIT 1;
```

The city with the highest total invoices is identified, providing insights for promotional events.

Output:

	billing_city character varying (30) 	invoice_total double precision 
1	Prague	273.240000000000007

Q5: Who is the best customer?

```
SELECT

    customer.customer_id,

    customer.first_name,

    customer.last_name,

    SUM(invoice.total) as total

FROM customer

JOIN invoice ON customer.customer_id =
invoice.customer_id

GROUP BY customer.customer_id

ORDER BY total DESC

LIMIT 1;
```

The customer who has spent the most is identified as the best customer.

Output:

	customer_id [PK] integer	first_name character (50)	last_name character (50)	total double precision
1	5	R	Madhav	144.54000000000002

Q6: List email, first name, last name & genre of all Rock Music listeners ordered alphabetically by email.

```
SELECT DISTINCT
    first_name,
    last_name,
    email
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;
```

This query lists all rock music listeners sorted by email.

Output:





	first_name character (50) 🔒		last_name character (50) 🔒		email character varying (50) 🔒
1	Aaron	...	Mitchell	...	aaronmitchell@yahoo.ca
2	Alexandre	...	Rocha	...	alero@uol.com.br
3	Astrid		Gruber	...	astrid.gruber@apple.at
4	Bjørn		Hansen	...	bjorn.hansen@yahoo.no
5	Camille	...	Bernard	...	camille.bernard@yahoo.fr
6	Daan		Peeters	...	daan_peeters@apple.be
Total rows: 59		Query complete 00:00:00.357			

Q7: Top 10 artists with the most rock music tracks.

```
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;
```

Top 10 artists who created the most rock music tracks are identified.

Output:

Showing rows: 1 to 10  Page No: <input type="text" value="1"/> of 1			
	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
Total rows: 10 Query complete 00:00:00.395			

Q8: Tracks longer than the average song length.

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

This query returns tracks longer than the average song length.

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
Total rows: 494 Query complete 00:00:00.369		CRLF

Q9: Amount spent by each customer on artists.

```
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;
```

This query identifies the amount spent by each customer on artists.

Output:

	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
Total rows: 43		Query complete 00:00:00.308			

Q10: Most popular music Genre for each country.

```
WITH most_selling_genre AS(  
  
    SELECT  
  
        i.billing_country AS country_name,  
  
        g.name AS genre_name,  
  
        COUNT(il.quantity) AS purchase,  
  
        g.genre_id,  
  
        ROW_NUMBER() OVER(PARTITION BY  
i.billing_country ORDER BY COUNT(il.quantity) DESC) AS  
rn  
  
    FROM invoice i  
  
    JOIN invoice_line il ON il.invoice_id =  
i.invoice_id  
  
    JOIN track t ON t.track_id = il.track_id  
  
    JOIN genre g ON g.genre_id = t.genre_id
```

```

GROUP BY 1, 2, 4
)

SELECT

    msg.country_name,

    msg.genre_id,

    msg.genre_name,

    msg.purchase

FROM most_selling_genre msg

WHERE msg.rn=1;

-- I can also solve the question without using WITH
clause just by using "SELECT * from"(...(query)
)"any_alias" "WHERE any_alias.column_name condition;"

```

This query identifies the most popular genre in each country.

Output:

	country_name character varying (30) 🔒	genre_id character varying (50) 🔒	genre_name character varying (120) 🔒	purchase bigint 🔒
1	Argentina	4	Alternative & Punk	17
2	Australia	1	Rock	34
3	Austria	1	Rock	40
4	Belgium	1	Rock	26
5	Brazil	1	Rock	205
6	Canada	1	Rock	333
Total rows: 24		Query complete 00:00:00.238		

Q11: Top customer for each country based on spending.

```
WITH most_spent_customer AS(

    SELECT

        i.billing_country,

        c.customer_id,

        c.first_name,

        c.last_name,

        SUM(i.total),

        ROW_NUMBER() OVER (PARTITION BY
i.billing_country ORDER BY SUM(i.total) DESC) AS rn

    FROM customer c

    JOIN invoice i ON i.customer_id = c.customer_id

    GROUP BY 1,2,3,4

)

SELECT *

FROM most_spent_customer msc

WHERE rn=1;
```

This query identifies the top customer for each country based on spending.

Output:

<div><div><div>≡+</div><div>📄</div><div>▼</div><div>📋</div><div>▼</div><div>🗑</div><div>📦</div><div>⬇</div><div>📶</div><div>SQL</div></div><div>Showing rows: 1 to 24</div><div>Page No: 1 of 1</div></div>							
	billing_country character varying (30)	customer_id integer	first_name character (50)		last_name character (50)	sum double precision	m bigint
1	Argentina	56	Diego	...	Gutiérrez	39.6	1
2	Australia	55	Mark	...	Taylor	81.18	1
3	Austria	7	Astrid	...	Gruber	69.3	1
4	Belgium	8	Daan	...	Peeters	60.38999999999999	1
5	Brazil	1	Luís	...	Gonçalves	108.89999999999998	1
6	Canada	3	Francois	...	Tremblay	99.99	1
Total rows: 24		Query complete 00:00:00.353					CRLF

Summary and Insights

This analysis provides several actionable insights:

- The senior-most employee and best customer were identified, enabling recognition and potential rewards.
- Countries and cities with the highest invoice totals were determined, aiding in targeted marketing campaigns.
- Popular genres and artists were highlighted, offering guidance for promotional events and inventory focus.
- The track-length analysis and customer spending patterns provide additional context for product offerings and personalized marketing.

These insights will empower the Sonic Bloom Music Store to enhance customer engagement, optimize inventory management, and increase overall revenue.