

Table of Content

S. no	Particular		Page no.
1	Introduction		2
2	Background		2
3	Data Analysis Process		3
	3.1	Ask	3
	3.2	Prepare	4
	3.3	Process	5
	3.4	Analyze, Share & Act	6

Introduction

In this case study, I will perform many real-world tasks of a junior data analyst at a fictional Music Store Retailer, Beat Music Store. In order to answer the key business questions, I will follow the steps of the data process: Ask, Prepare, Process, Analyse, Share, and Act.

Background

Beat Music Store:

- ☐ Beat Music Store is a music retailer company. Beat Music Store sells its product globally.
- ☐ Beat Music store wants to understand their customers and grow more sales in 2023.

Scenario:

I am assuming to be a junior data analyst working in the marketing analyst team at Beat Music Store, a music retailer company. The director of marketing believes the company's future success depends on their customers and grow more sales in 2023. Therefore, my team wants to understand how different variables affected sales and growth of company in past.

From these insights, my team will design a new marketing strategy to understand customer base and increase sales in 2023. But first, Beat Music Store executives must approve our recommendations, so they must be backed up with compelling data insights.

Data Analysis Process

Ask

Business Task

Beat Music store wants to understand their customer base and grow more sales in 2023.

Analysis Questions and Queries.

1. who is the senior most employee base on job title?
2. which countries have the most invoices?
3. what are top 3 values of total invoice?
4. 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
5. 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
6. 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
7. 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.
8. 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
9. Find how much amount spent by each customer on bestselling artist? Write a query to return customers name, artist name and total spent
10. 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 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

11. 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

Prepare

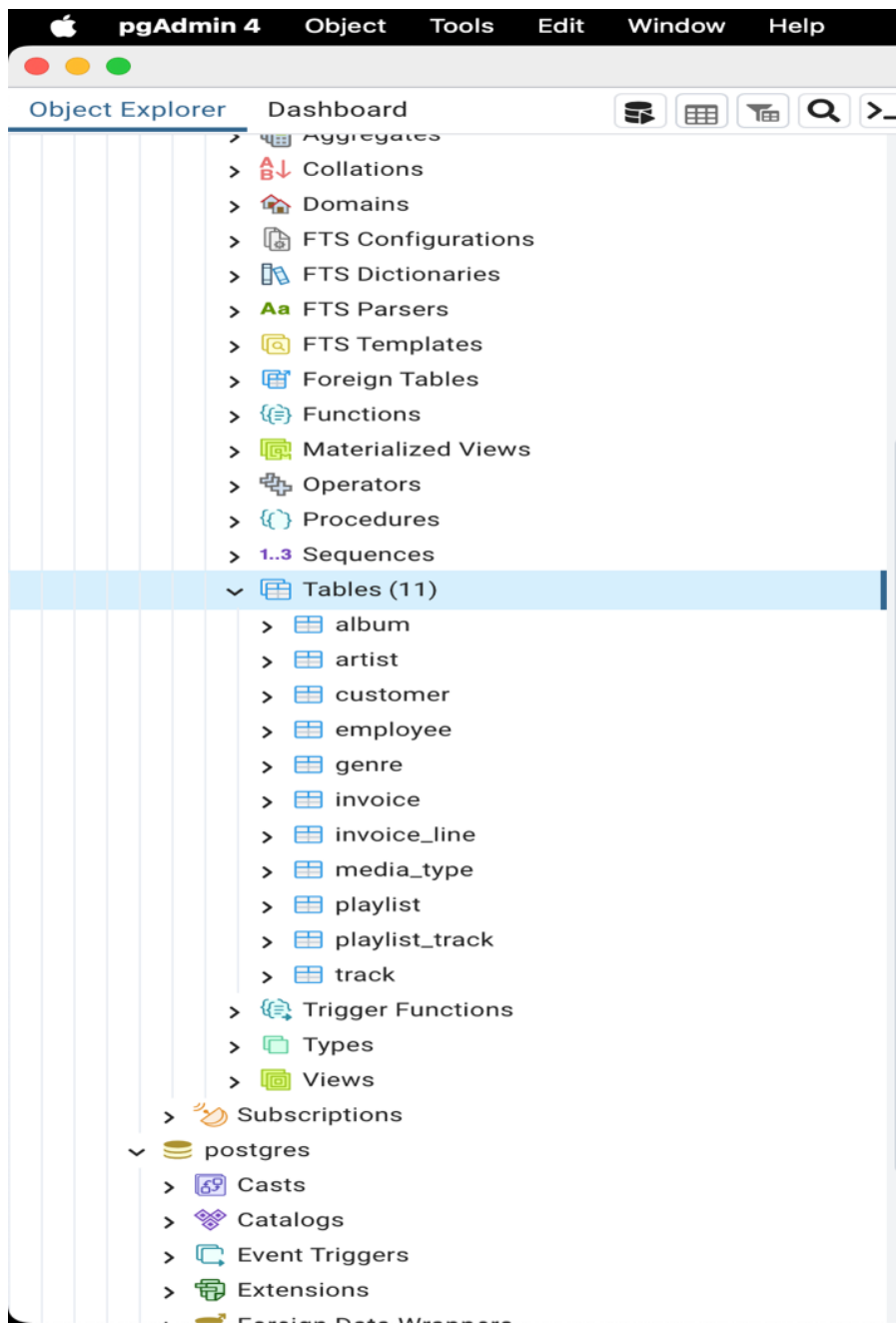
Data source

I will use Beat Music Store sales data to analyze and identify trends from Jan 2022 to Dec 2022 which can be downloaded from GitHub website.

This is public data that can be used to explore how different customer types buying different products of Beat Music Store from all over India.

Data organization

The Beat Music Store database contain 11 tables: album, artist, customer, employee, genre, invoice, invoice_line, media_type, playlist, playlist_track, track. These tables contains data of Beat Music Store.



Process

Data exploration

pgAdmin4 is used for data exploration and to get familiarize with the data

Data Cleaning

Data cleaning and modification is done with the help of SQL queries in pgAdmin4

Analyse , Share & Act

After cleaning the data, it is ready for analysis. pgAdmin4 is used for Data Analysis.

In data analysis we will find the answers to the Data Analysis Questions and Queries.

1. who is the senior most employee base on job title?

The screenshot shows the pgAdmin 4 web interface. On the left, the 'Object Explorer' pane displays a tree structure of database objects. The 'employee' table is selected, and its columns are visible: employee_id, first_name, last_name, title, levels. The main pane shows a SQL query editor with the following query:

```
-- Q1 who is the senior most employee basen on job title?
SELECT employee_id, first_name, last_name, title, levels
FROM employee
ORDER BY levels DESC
LIMIT 1
```

Below the query editor, the 'Data Output' pane displays the results of the query in a table format:

employee_id	first_name	last_name	title	levels
9	Mohan	Madan	Senior General Manager	L7

The status bar at the bottom indicates 'Total rows: 1 of 1' and 'Query complete 00:00:00.048'.

2. which countries have the most invoices?

The screenshot shows the pgAdmin 4 interface. On the left, the 'Object Explorer' pane displays the database structure, with the 'invoice' table selected under the 'customer' schema. The main pane shows a SQL query in the 'Query' tab:

```
-- Q2 which countries have the most invoices?  
SELECT DISTINCT billing_country, COUNT(*) as number_of_invoice  
FROM invoice  
GROUP BY billing_country  
ORDER BY number_of_invoice DESC
```

Below the query, the 'Data Output' tab displays the results of the query in a table format:

	billing_country	number_of_invoice
1	USA	131
2	Canada	76
3	Brazil	61
4	France	50
5	Germany	41
6	Czech Republic	30
7	Portugal	29
8	United Kingdom	28
9	India	21
10	Chile	13
11	Ireland	13
12	Finland	11
13	Spain	11
14	Australia	10
15	Denmark	10
16	Hungary	10
17	Netherlands	10
18	Poland	10

The status bar at the bottom indicates 'Total rows: 24 of 24' and 'Query complete 00:00:00.098'. The current position is 'Ln 8, Col 1'.

3. what are top 3 values of total invoice?

The screenshot shows the pgAdmin 4 interface. On the left, the 'Object Explorer' pane is open, showing the database structure. The 'invoice' table is selected under the 'customer' schema. The 'Columns' pane for 'invoice' is expanded, showing columns: invoice_id, customer_id, invoice_date, billing_address, billing_city, billing_state, billing_country, billing_postal, and total. The 'Query' pane in the center contains the following SQL query:

```
-- Q3 what are top 3 values of total invoice?  
SELECT invoice_id, total  
FROM invoice  
ORDER BY total DESC  
LIMIT 3
```

The 'Data Output' pane at the bottom displays the results of the query in a table format:

invoice_id	total
1	23.759999999999998
2	19.8
3	19.8

The status bar at the bottom indicates 'Total rows: 3 of 3' and 'Query complete 00:00:00.084'.

4. 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

The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer shows the database structure, with the 'invoice' table selected under the 'customer' schema. The main pane displays a SQL query in the Query editor:

```
/* 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 DISTINCT billing_city, SUM(total) AS invoice_total
FROM invoice
GROUP BY billing_city
ORDER BY invoice_total DESC
LIMIT 1
```

Below the query editor, the Data Output pane shows the results of the query:

billing_city	invoice_total
Prague	273.24000000000007

The status bar at the bottom indicates 'Total rows: 1 of 1' and 'Query complete 00:00:00.073'.

5. 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

The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer shows the database structure, with the 'invoice' table selected under the 'customer' schema. The main pane displays a SQL query in the 'Query' tab. The query is as follows:

```
29 /* Q5 Who is the best customer? The customer who has spent the most money will be
30 declared the best customer. Write a query that returns the person who has spent the
31 most money */
32 SELECT invoice.customer_id, first_name, last_name, SUM(total) as total_spend
33 FROM customer JOIN invoice
34 ON customer.customer_id = invoice.customer_id
35 GROUP BY invoice.customer_id, first_name, last_name
36 ORDER BY total_spend DESC
37 LIMIT 1
```

Below the query editor, the 'Data Output' tab shows the results of the query. The results are displayed in a table with the following columns: customer_id, first_name, last_name, and total_spend. The table contains one row of data:

customer_id	first_name	last_name	total_spend
1	S R	Madhav	144.54000000000002

At the bottom of the interface, a status bar indicates 'Total rows: 1 of 1' and 'Query complete 00:00:00.078'. A green notification box at the bottom right states: '✓ Successfully run. Total query runtime: 78 msec. 1 rows affected. ✕'.

6. 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

The screenshot shows the pgAdmin 4 interface with a SQL query executed in the 'Query' tab. The query is as follows:

```

1  -- 1. Write query to return the email, first name, last name, & Genre of all Rock Music
2  -- listeners. Return your list ordered alphabetically by email starting with A
3  SELECT DISTINCT email, first_name, last_name, genre.name AS genre_name
4  FROM genre JOIN track ON genre.genre_id=track.genre_id
5       JOIN invoice_line ON track.track_id=invoice_line.track_id
6       JOIN invoice ON invoice_line.invoice_id=invoice.invoice_id
7       JOIN customer ON invoice.customer_id=customer.customer_id
8  WHERE genre.name = 'Rock'
9  ORDER BY email
10

```

The results are displayed in the 'Data Output' tab, showing a list of 15 Rock Music listeners ordered by email. The columns are email, first_name, last_name, and genre_name.

	email	first_name	last_name	genre_name
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	Bjorn	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@yahoo.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	fernadamos4@uol.com.br	Fernanda	Ramos	Rock

Total rows: 59 of 59 Query complete 00:00:00.087 Ln 9, Col 15

7. 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.

The screenshot shows the pgAdmin 4 interface. On the left, the 'Object Explorer' pane shows the database structure, with 'invoice' selected under the 'genre' table. The main pane displays a SQL query in the 'Query' tab. The query is as follows:

```
--2. 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.name, COUNT(track_id) AS total_track_count
FROM genre JOIN track ON genre.genre_id=track.genre_id
      JOIN album ON track.album_id=album.album_id
      JOIN artist ON album.artist_id=artist.artist_id
WHERE genre.name = 'Rock'
GROUP BY artist.name
ORDER BY total_track_count DESC
LIMIT 10
```

The 'Data Output' pane shows the results of the query, displaying a table with 10 rows and 2 columns: 'name' and 'total_track_count'.

	name	total_track_count
1	Led Zeppelin	114
2	U2	112
3	Deep Purple	92
4	Iron Maiden	81
5	Pearl Jam	54
6	Van Halen	52
7	Queen	45
8	The Rolling Stones	41
9	Creedence Clearwater Revival	40
10	Kiss	35

The status bar at the bottom indicates 'Total rows: 10 of 10' and 'Query complete 00:00:00.070'.

8. 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

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, with the 'invoice' table selected under the 'customer' schema. The main pane shows a SQL query in the 'Query' tab. The query is as follows:

```
/* 3. 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 AS track_name, milliseconds AS song_length
FROM track
WHERE milliseconds > (SELECT AVG(milliseconds) FROM track)
ORDER BY milliseconds DESC
```

The 'Data Output' tab shows the results of the query, displaying a table with two columns: 'track_name' (character varying (150)) and 'song_length' (integer). The results are ordered by song length in descending order, showing 17 tracks.

track_name	song_length
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

Total rows: 494 of 494 Query complete 00:00:00.091 Ln 27, Col 50

9. Find how much amount spent by each customer on bestselling artist? Write a query to return customers name, artist name and total spent

The screenshot shows the pgAdmin 4 interface with a SQL query executed. The query is as follows:

```

1 /* 1. Find how much amount spent by each customer on best selling artist? Write a query to return
2 customers name, artist name and total spent */
3 WITH bsa AS
4 (SELECT artist.artist_id, artist.name AS artist_name, SUM(quantity * invoice_line.unit_price) AS total_sp
5 FROM invoice_line
6 JOIN track ON invoice_line.track_id=track.track_id
7 JOIN album ON track.album_id=album.album_id
8 JOIN artist ON album.artist_id=artist.artist_id
9 GROUP BY artist.artist_id, artist.name
10 ORDER BY total_sp DESC
11 LIMIT 1
12 )
13 SELECT artist_name, first_name, last_name, SUM(quantity * invoice_line.unit_price) AS total_sp
14 FROM customer
15 JOIN invoice ON customer.customer_id=invoice.customer_id
16 JOIN invoice_line ON invoice.invoice_id=invoice_line.invoice_id
17 JOIN track ON invoice_line.track_id=track.track_id
18 JOIN album ON track.album_id=album.album_id
19 JOIN bsa ON album.artist_id=bsa.artist_id
20 GROUP BY artist_name, first_name, last_name
21 ORDER BY total_sp DESC
22

```

The results are displayed in the Data Output tab, showing the top 6 customers by total spent:

	artist_name	first_name	last_name	total_sp
1	Queen	Hugh	O'Reilly	27.7199999999999985
2	Queen	Niklas	Schröder	18.81
3	Queen	François	Tremblay	17.82
4	Queen	João	Fernandes	16.8300000000000002
5	Queen	Marc	Dubois	11.88
6	Queen	Phil	Huohes	11.88

Total rows: 43 of 43 Query complete 00:00:00.033

10. 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 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

The screenshot shows the pgAdmin 4 interface with a SQL query executed in the 'Query' tab. The query is designed to find the most popular music genre for each country based on the total purchase amount. It uses a CTE named 'country_best_genre' to calculate the total purchase for each country and genre, then ranks them to find the top genre for each country. The results are displayed in the 'Data Output' tab, showing a list of countries and their top genres along with the total purchase amount.

Query:

```

/* 2. 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 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 country_best_genre AS
(
SELECT billing_country, genre.name AS genre_n, SUM(invoice_line.unit_price * invoice_line.quantity) AS total_purchase
FROM invoice
JOIN invoice_line ON invoice.invoice_id=invoice_line.invoice_id
JOIN track ON invoice_line.track_id=track.track_id
JOIN genre ON track.genre_id=genre.genre_id
GROUP BY billing_country, genre.name
ORDER BY billing_country ASC, total_purchase DESC)
SELECT billing_country AS country_n, genre_n, total_purchase
FROM country_best_genre
WHERE r_no = 1

```

Data Output:

country_n	genre_n	total_purchase
1 Argentina	Alternative & Punk	16.830000000000002
2 Australia	Rock	33.659999999999998
3 Austria	Rock	39.599999999999994
4 Belgium	Rock	25.739999999999998
5 Brazil	Rock	202.95000000000004
6 Canada	Rock	329.670000000000155
7 Chile	Rock	60.3900000000000036
8 Czech Republic	Rock	141.569999999999982
9 Denmark	Rock	23.759999999999999

Total rows: 24 of 24 Query complete 00:00:00.067

11. 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

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, with the 'invoice' table selected under the 'customer' schema. The main pane shows a SQL query that identifies the top customer by total purchase amount for each country. The query uses a CTE named 'country_best_customer' to find the top customer for each country, then selects the country, customer details, and total purchase amount, ordered by country and then by total purchase amount. The results pane shows the output of the query, displaying 9 rows of data.

```

/* 3. 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 country_best_customer AS(
  SELECT billing_country, first_name, last_name, SUM(invoice_line.unit_price * invoice_line.quantity) /
  ROW_NUMBER() OVER(PARTITION BY billing_country ORDER BY SUM(invoice_line.unit_price * invoice_line
FROM customer
  JOIN invoice ON customer.customer_id=invoice.customer_id
  JOIN invoice_line ON invoice.invoice_id=invoice_line.invoice_id
  GROUP BY billing_country, first_name, last_name
  ORDER BY billing_country ASC)
SELECT billing_country AS country_n, first_name, last_name, total_purchase
FROM country_best_customer
WHERE r_no = 1

```

	country_n	first_name	last_name	total_purchase
1	Argentina	Diego	Gutiérrez	39.599999999999994
2	Australia	Mark	Taylor	81.179999999999995
3	Austria	Astrid	Gruber	69.300000000000001
4	Belgium	Daan	Peeters	60.3900000000000036
5	Brazil	Luis	Gonçalves	108.89999999999998
6	Canada	François	Tremblay	99.989999999999985
7	Chile	Luis	Rojas	97.019999999999987
8	Czech Republic	R	Madhav	144.539999999999985
9	Denmark	Kara	Nielsen	37.619999999999999

Total rows: 24 of 24 Query complete 00:00:00.034

Ln 52, Col 28

✓ File saved successfully. X