Music Store Data Analysis

ABSTRACT:

The music retail sector is leveraging data analysis to gain valuable insights for informed decision-making and operational optimization, as music stores now serve as hubs for data generated from various aspects of their operations.

The Music Store Data Analysis project explores a music store's database to uncover patterns, trends, and insights. It includes tables like 'employee,' 'customer,' 'invoice,' 'track,' 'playlist,' 'artist,' 'album,''media_type,' and 'genre.' These tables contain information about employees, customers, sales transactions, music tracks, playlists, artists, albums, and genres.

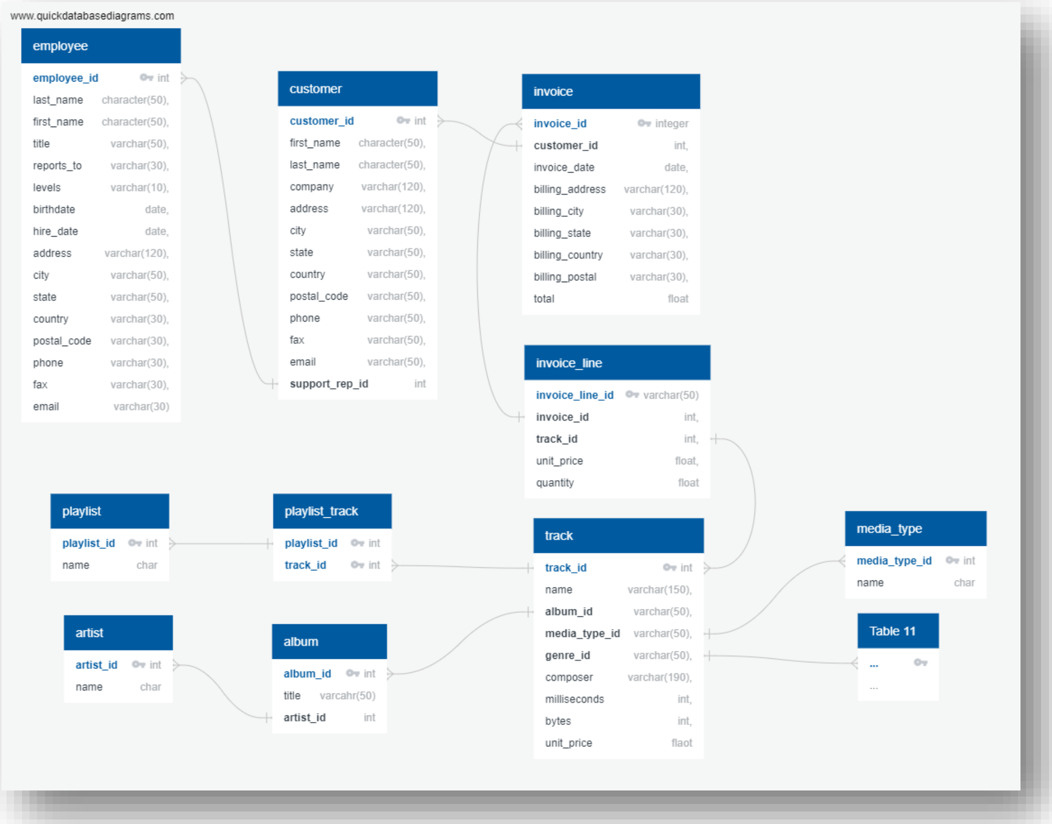
AIM:

This project aims to provide data-driven insights to a music store, enabling strategic decisions and enhancing customer satisfaction and profitability. The study identifies senior employees, top invoice-paying countries, top music genres, and top spending customers by analyzing job titles, customer preferences, and purchase data, while also revealing popular music genres in each country.

Concept Used:

SQL Database Management: The provided schema will be utilized for data extraction, transformation, and analysis using SQL queries.

Database Schema:



employee	
employee_id	🔑 int
last_name	character(50),
first_name	character(50),
title	varchar(50),
reports_to	varchar(30),
levels	varchar(10),
birthdate	date,
hire_date	date,
address	varchar(120),
city	varchar(50),
state	varchar(50),
country	varchar(30),
postal_code	varchar(30),
phone	varchar(30),
fax	varchar(30),
email	varchar(30)

customer	
customer_id	🔑 int
first_name	character(50),
last_name	character(50),
company	varchar(120),
address	varchar(120),
city	varchar(50),
state	varchar(50),
country	varchar(50),
postal_code	varchar(50),
phone	varchar(50),
fax	varchar(50),
email	varchar(50),
support_rep_id	int

invoice	
invoice_id	🔑 integer
customer_id	int,
invoice_date	date,
billing_address	varchar(120),
billing_city	varchar(30),
billing_state	varchar(30),
billing_country	varchar(30),
billing_postal	varchar(30),
total	float

invoice_line	
invoice_line_id	🔑 varchar(50)
invoice_id	int,
track_id	int,
unit_price	float,
quantity	float

playlist	
playlist_id	🔑 int
name	char

playlist_track	
playlist_id	🔑 int
track_id	🔑 int

track	
track_id	🔑 int
name	varchar(150),
album_id	varchar(50),
media_type_id	varchar(50),
genre_id	varchar(50),
composer	varchar(190),
milliseconds	int,
bytes	int,
unit_price	flaot

media_type	
media_type_id	🔑 int
name	char

Table 11	
...	🔑
...	

artist	
artist_id	🔑 int
name	char

album	
album_id	🔑 int
title	varcahr(50)
artist_id	int

Table Details:

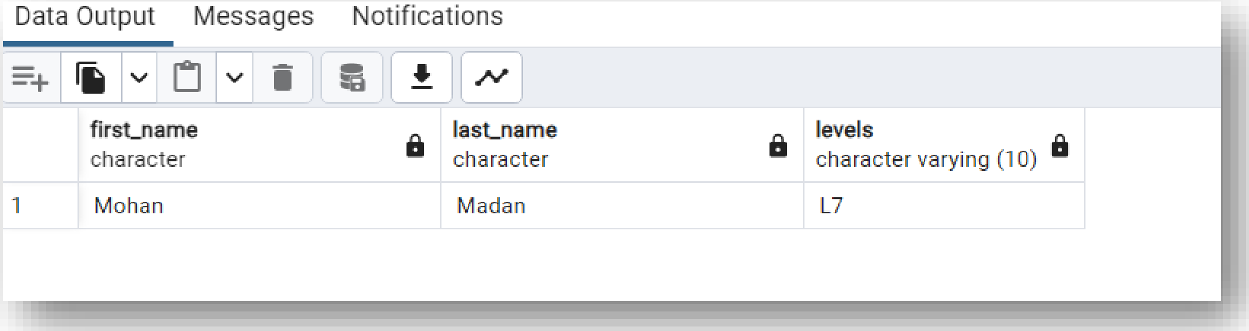
- Employee Table
- Customer Table
- Invoice Table
- Invoice_line Table
- Track Table
- Album Table
- Artist Table
- Playlist Table
- Playlist_track Table
- Media_type Table
- Genre Table

Use the link to access table details.

Data Insights

1. Who is the senior most employee based on job title?

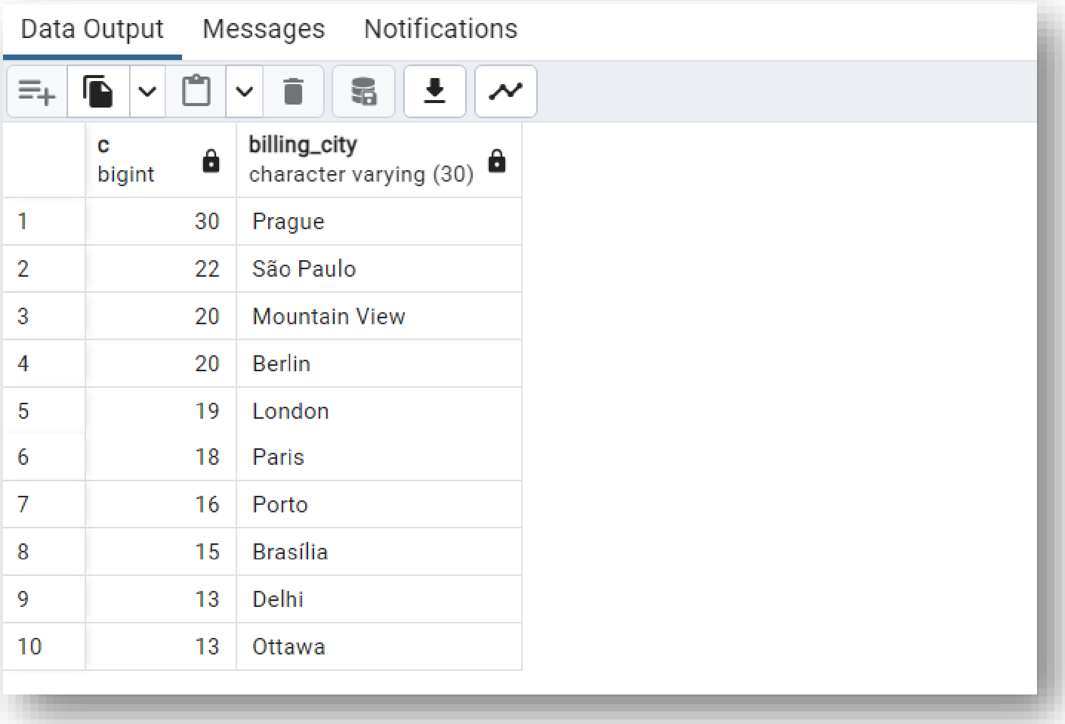
```
select first_name, last_name, levels from employee

Order by levels desc

limit 1
```



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

2. Which countries have the most Invoices?

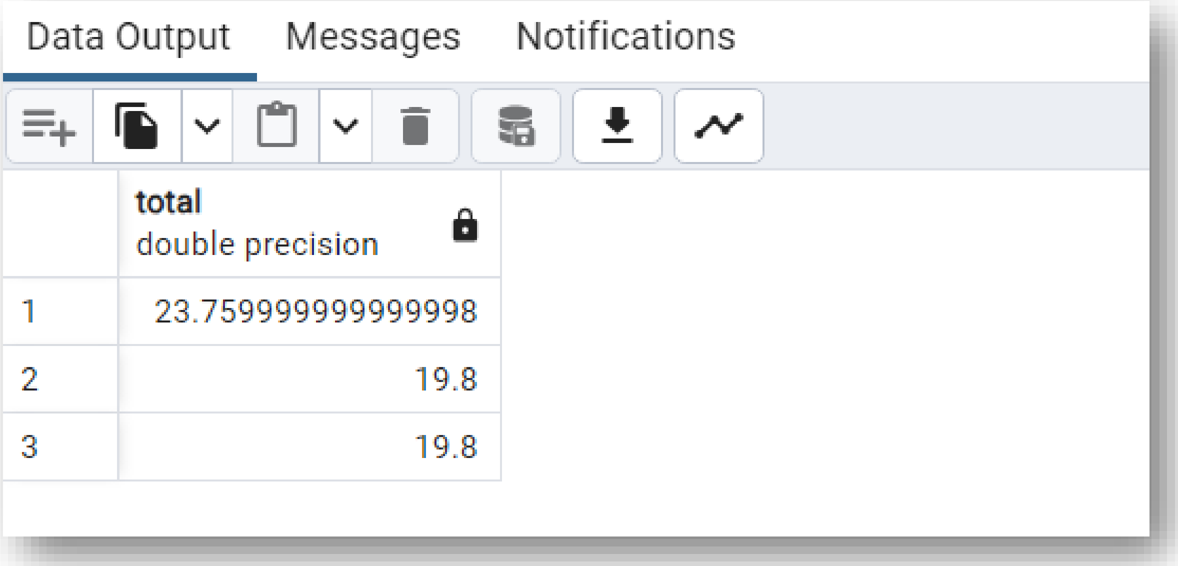
```
select count(*) as c, billing_city from invoice

Group by billing_city

order by c desc

limit 10
```



	c bigint	billing_city character varying (30)
1	30	Prague
2	22	São Paulo
3	20	Mountain View
4	20	Berlin
5	19	London
6	18	Paris
7	16	Porto
8	15	Brasília
9	13	Delhi
10	13	Ottawa

3. **What are the top 3 values of the total invoice?**

```
select total from invoice

order by total desc

limit 3
```

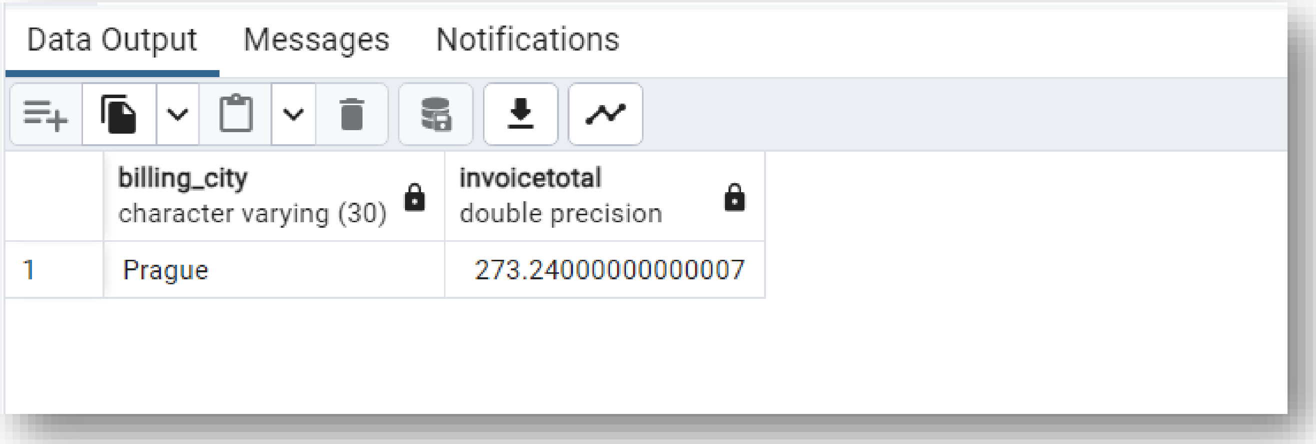


	total double precision
1	23.759999999999998
2	19.8
3	19.8

4. **Which city has the best customers? We would like to throw a promotional Music Festival in the city where 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
```

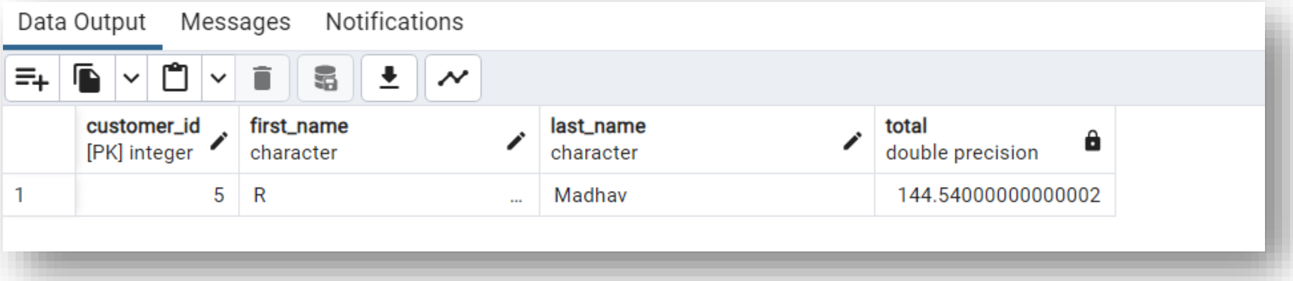
```
from invoice

Group by billing_city

order by InvoiceTotal Desc

Limit 1
```



	billing_city character varying (30)	invoicetotal double precision
1	Prague	273.24000000000007

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.**

```
select c.customer_id,c.first_name,c.last_name, sum(total) as Total

from customer as c

join invoice as inv

on c.customer_id=inv.customer_id

group by c.customer_id

order by Total desc

limit 1
```

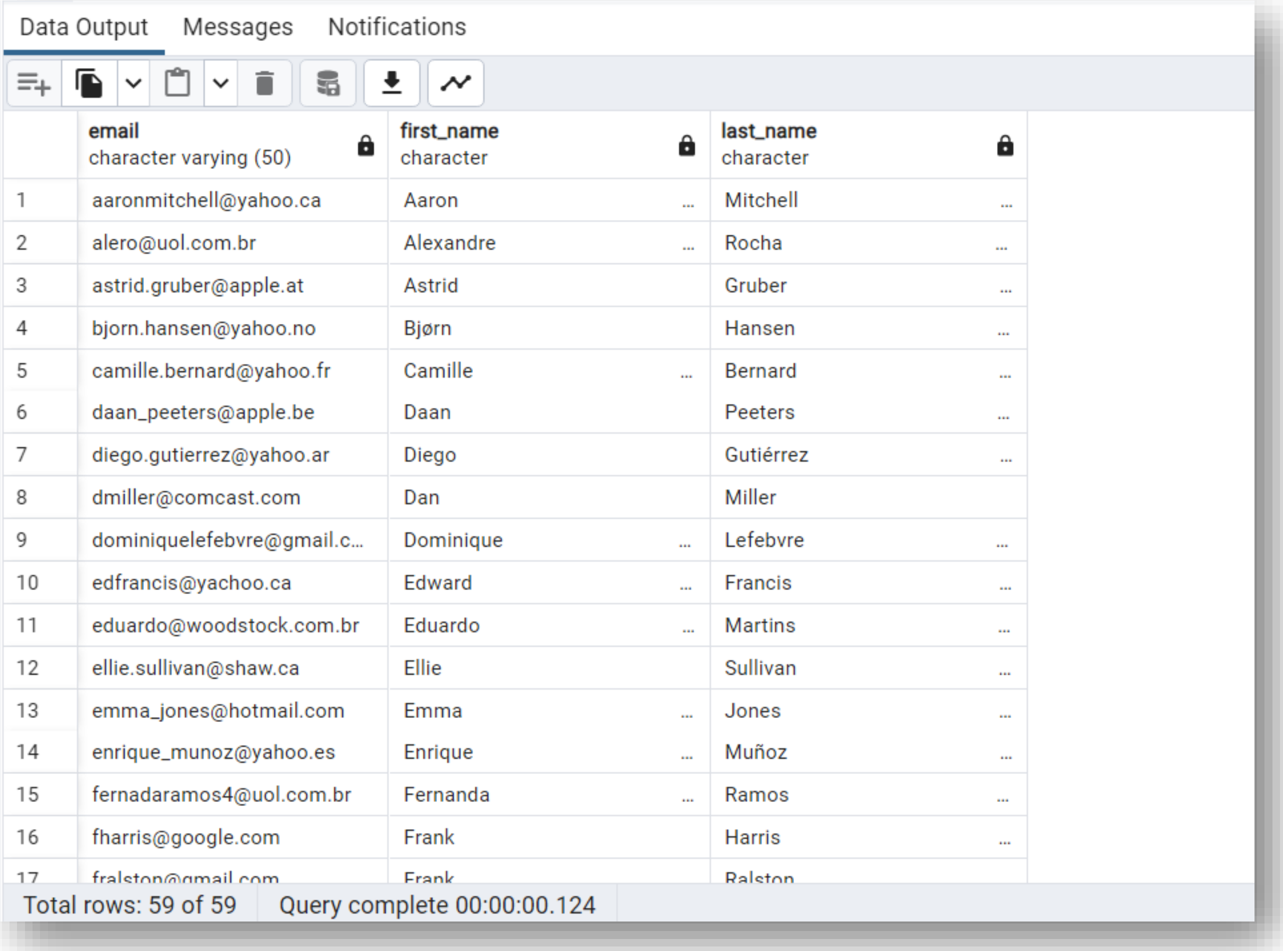


	customer_id [PK] integer	first_name character	last_name character	total double precision
1	5	R ...	Madhav	144.5400000000002

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.**

```
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
```

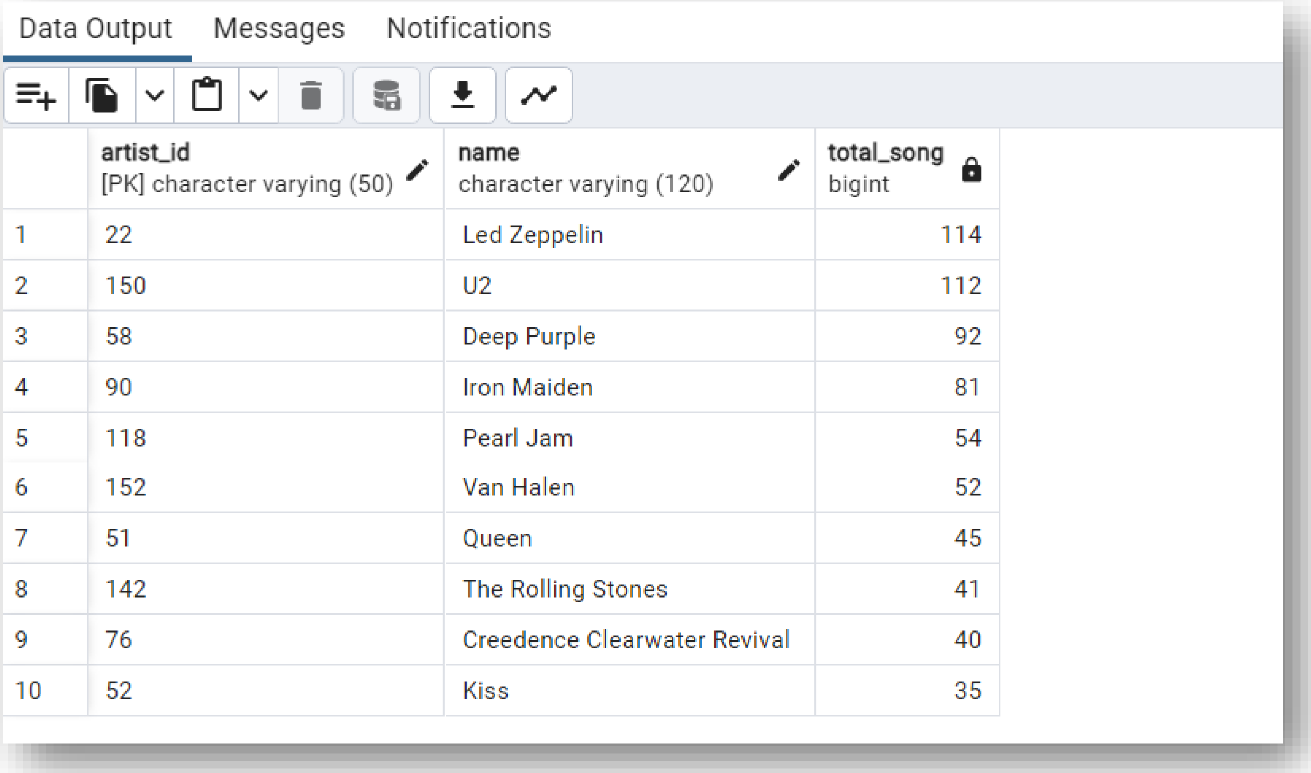


	email character varying (50)	first_name character	last_name character
1	aaronmitchell@yahoo.ca	Aaron	Mitchell
2	alero@uol.com.br	Alexandre	Rocha
3	astrid.gruber@apple.at	Astrid	Gruber
4	bjorn.hansen@yahoo.no	Bjørn	Hansen
5	camille.bernard@yahoo.fr	Camille	Bernard
6	daan_peeters@apple.be	Daan	Peeters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

Total rows: 59 of 59 Query complete 00:00:00.124

7. **Let us 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.**

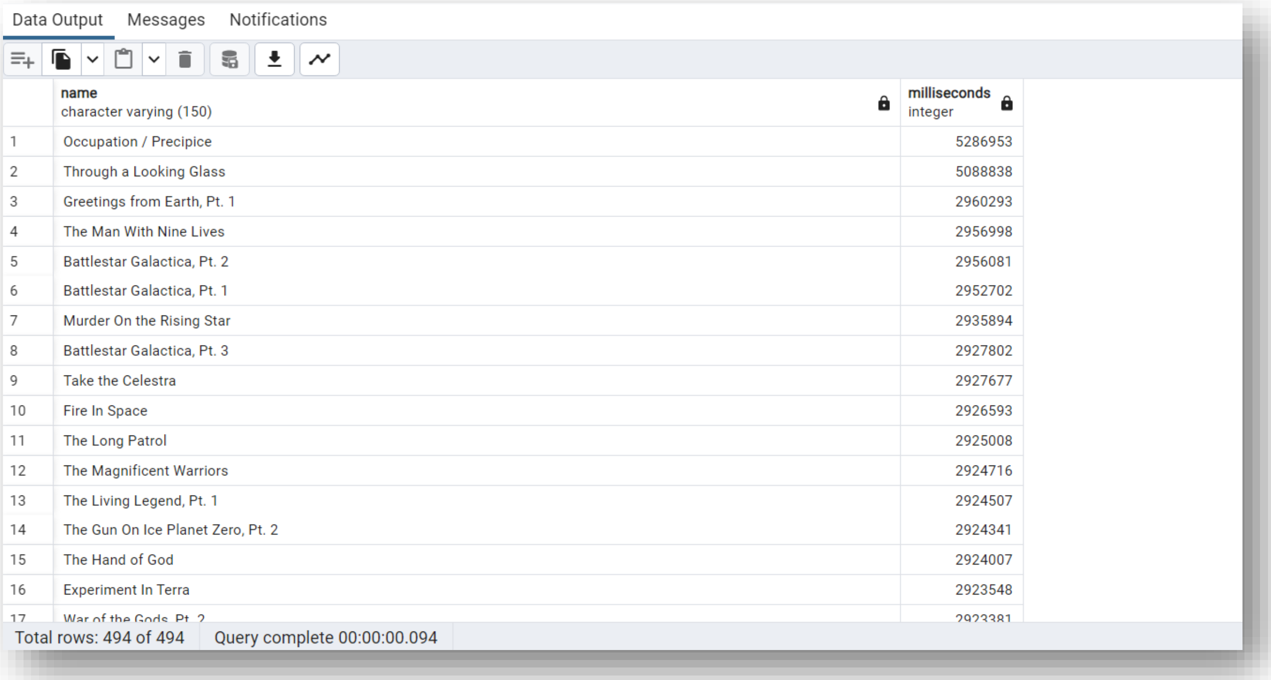
```
select artist.artist_id, artist.name, count(artist.artist_id) as total_song

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 total_song desc
limit 10
```



	artist_id [PK] character varying (50)	name character varying (120)	total_song 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

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.**

```
select name,milliseconds from track
where milliseconds > (
    select avg(milliseconds)from track
)
order by milliseconds desc
```

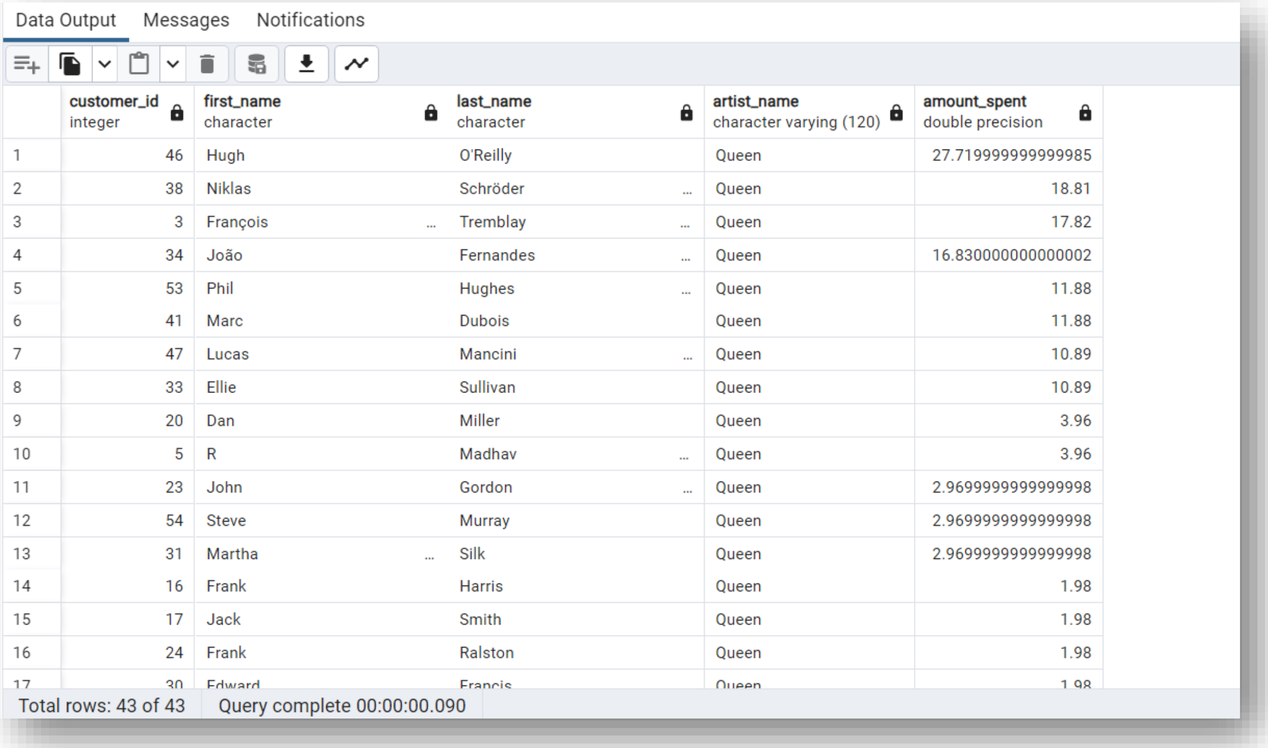


	name character varying (150)	milliseconds integer
1	Occupation / Precipice	5286953
2	Through a Looking Glass	5088838
3	Greetings from Earth, Pt. 1	2960293
4	The Man With Nine Lives	2956998
5	Battlestar Galactica, Pt. 2	2956081
6	Battlestar Galactica, Pt. 1	2952702
7	Murder On the Rising Star	2935894
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

Total rows: 494 of 494 Query complete 00:00:00.094

9. Find how much amount spent by each customer on artists. Write a query to return the customer's name, artist name, and total spent

```
WITH best_selling_artist AS (
   SELECT artist.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
```

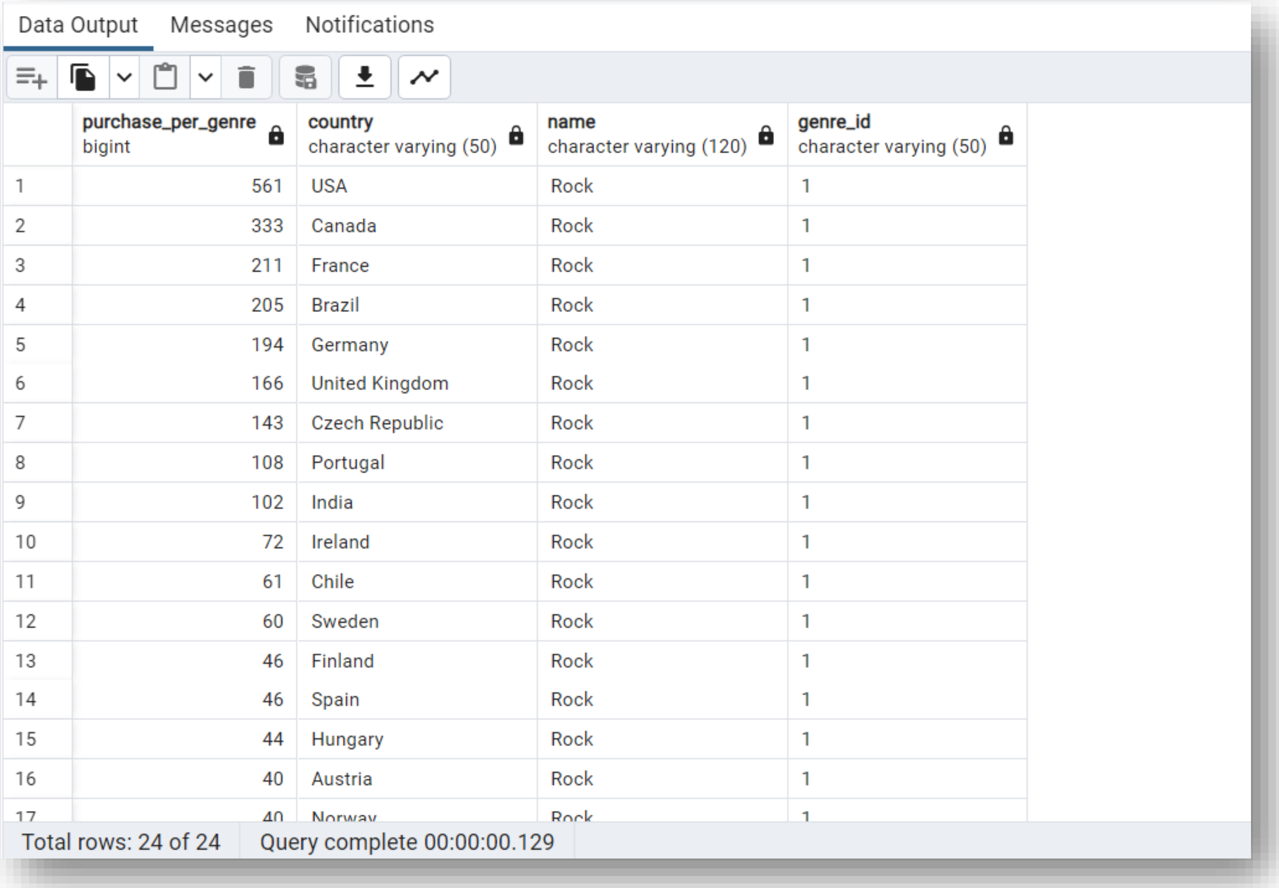


	customer_id integer	first_name character	last_name character	artist_name character varying (120)	amount_spent double precision
1	46	Hugh	O'Reilly	Queen	27.719999999999985
2	38	Niklas	Schröder ...	Queen	18.81
3	3	François ...	Tremblay ...	Queen	17.82
4	34	João	Fernandes ...	Queen	16.830000000000002
5	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

Total rows: 43 of 43   Query complete 00:00:00.090

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 number 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 recursive

sales_per_country as (

    select count(*) as purchase_per_genre, customer.country, genre.name, genre.genre_id

    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

),
max_genre_per_country as (
    select max(purchase_per_genre) as max_genre , 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.purchase_per_genre = max_genre_per_country.max_genre
order by sales_per_country.purchase_per_genre desc
```



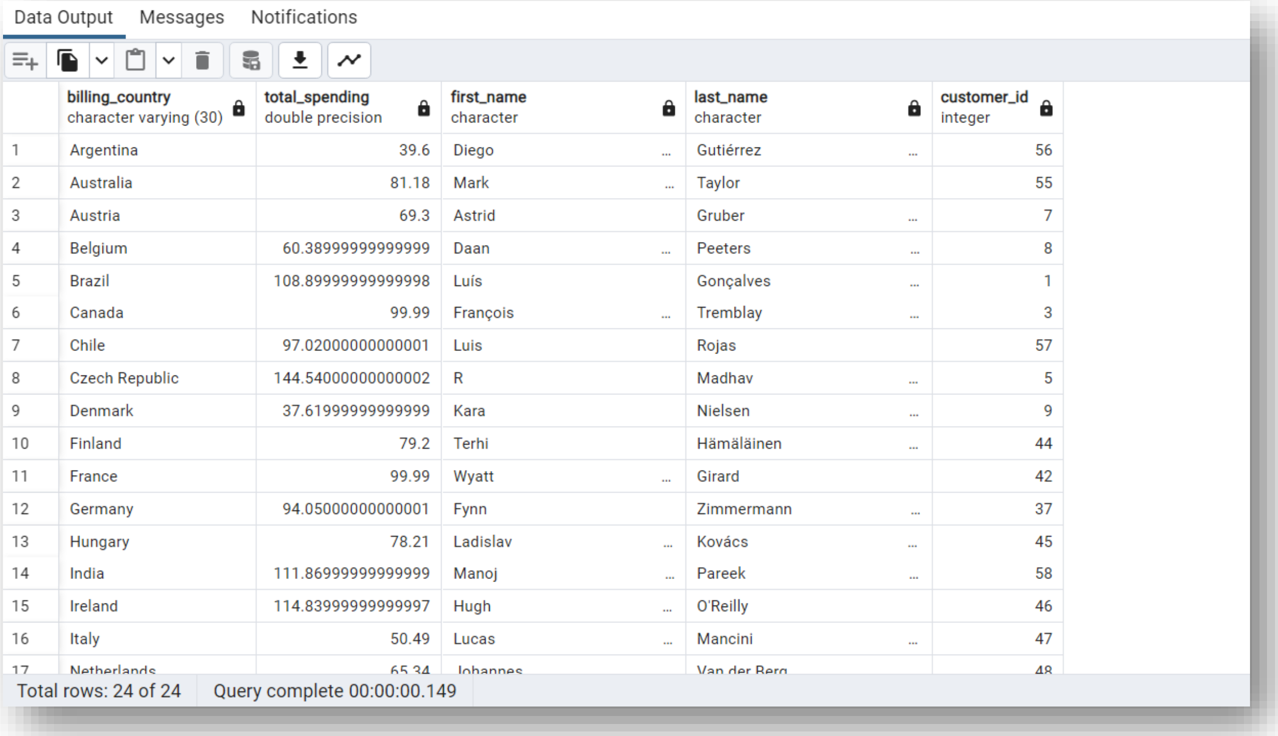
	purchase_per_genre bigint	country character varying (50)	name character varying (120)	genre_id character varying (50)
1	561	USA	Rock	1
2	333	Canada	Rock	1
3	211	France	Rock	1
4	205	Brazil	Rock	1
5	194	Germany	Rock	1
6	166	United Kingdom	Rock	1
7	143	Czech Republic	Rock	1
8	108	Portugal	Rock	1
9	102	India	Rock	1
10	72	Ireland	Rock	1
11	61	Chile	Rock	1
12	60	Sweden	Rock	1
13	46	Finland	Rock	1
14	46	Spain	Rock	1
15	44	Hungary	Rock	1
16	40	Austria	Rock	1
17	40	Norway	Rock	1

```
Total rows: 24 of 24    Query complete 00:00:00.129
```

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.

```
WITH RECURSIVE
```

```
    customter_with_country AS (
            SELECT customer.customer_id,first_name,last_name,billing_country,SUM(total) AS total_spending
            FROM invoice
            JOIN 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 customter_with_country
            GROUP BY billing_country)

SELECT cc.billing_country, cc.total_spending, cc.first_name, cc.last_name, cc.customer_id
FROM customter_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;
```



	billing_country character varying (30)	total_spending double precision	first_name character	last_name character	customer_id integer
1	Argentina	39.6	Diego …	Gutiérrez …	56
2	Australia	81.18	Mark …	Taylor	55
3	Austria	69.3	Astrid	Gruber …	7
4	Belgium	60.38999999999999	Daan …	Peeters …	8
5	Brazil	108.89999999999998	Luís	Gonçalves …	1
6	Canada	99.99	François …	Tremblay …	3
7	Chile	97.02000000000001	Luis	Rojas	57
8	Czech Republic	144.54000000000002	R	Madhav …	5
9	Denmark	37.61999999999999	Kara	Nielsen …	9
10	Finland	79.2	Terhi	Hämäläinen …	44
11	France	99.99	Wyatt …	Girard	42
12	Germany	94.05000000000001	Fynn	Zimmermann …	37
13	Hungary	78.21	Ladislav …	Kovács …	45
14	India	111.86999999999999	Manoj …	Pareek …	58
15	Ireland	114.83999999999997	Hugh …	O'Reilly	46
16	Italy	50.49	Lucas …	Mancini …	47
17	Netherlands	65.34	Johannes	Van der Berg	48

Total rows: 24 of 24 Query complete 00:00:00.149

Conclusion

The Music Store Data Analysis project demonstrates the importance of data-driven decision-making in the music retail industry. It analyzes employee hierarchy, customer behavior, music preferences, and genre popularity to guide strategic decisions. SQL queries provide actionable insights for optimizing operations, enhancing customer engagement, and maximizing revenue, ultimately leading to a more successful music store.