

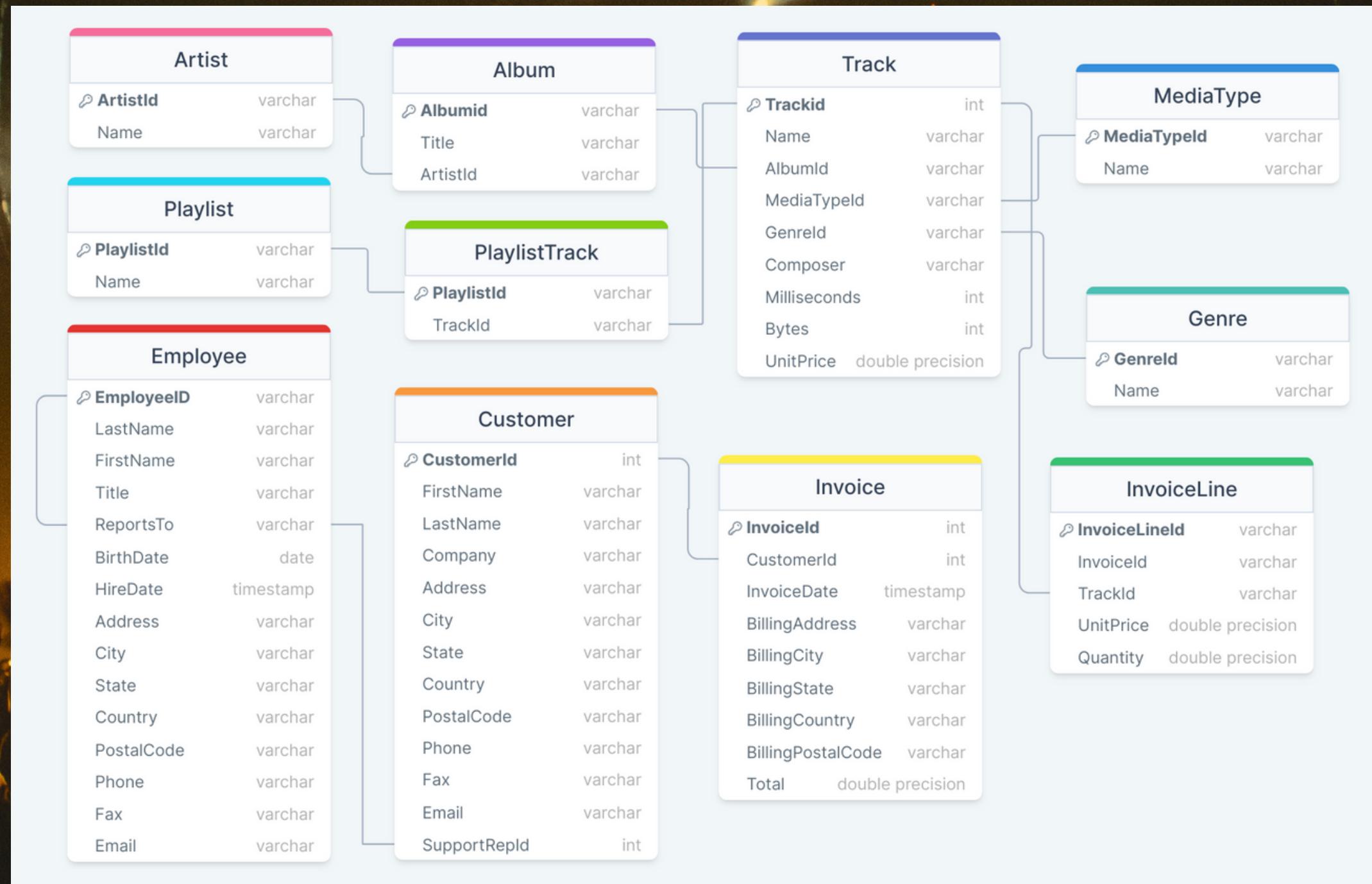


Music Store Analysis

FT. POSTGRESQL

Music Store Analysis

SCHEMA



Introduction

In this project, I conducted a comprehensive analysis of a Music Store Database, employing PostgreSQL on the PgAdmin4 tool. The database utilized a star schema, connecting multiple data tables to facilitate a holistic exploration of data relationships. This project demanded a deep understanding of SQL concepts to derive meaningful insights.

Level : Easy

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

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

OUTPUT

first_name	last_name	title
character	character	character varying (50)

Mohan Madan Senior General Manager

Level : Easy

Q. Which countries have the most Invoices?

```
SELECT COUNT(*) as total_invoices, billing_country  
FROM invoice  
GROUP BY billing_country  
ORDER BY total_invoices desc;
```

OUTPUT

	total_invoices bigint	billing_country character varying (30)
1	131	USA
2	76	Canada
3	61	Brazil
4	50	France
5	41	Germany

Level : Easy

Q. What are top 3 values of total invoice

```
SELECT total  
FROM invoice  
ORDER BY total desc limit 3;
```

OUTPUT

	total	double precision
1	23.759999999999998	
2		19.8
3		19.8

Level : Easy

Q. 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 sum_invoices  
FROM invoice  
GROUP BY billing_city  
ORDER BY SUM(total) DESC LIMIT 1;
```

OUTPUT

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

Level : Easy

Q. 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 i.customer_id, c.first_name, c.last_name,  
SUM(i.total) as money_spent  
FROM invoice i  
INNER JOIN customer c  
on i.customer_id = c.customer_id  
GROUP BY i.customer_id, c.first_name, c.last_name  
ORDER BY SUM(i.total) DESC LIMIT 1;
```

OUTPUT

customer_id	first_name	last_name	money_spent
5	R	Madhav	144.54000000000002

Level : Medium

Q. 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 c.email, c.first_name, c.last_name, g.name as genre_name
FROM customer c
INNER JOIN invoice i on c.customer_id = i.customer_id
INNER JOIN invoice_line il on i.invoice_id = il.invoice_id
INNER JOIN track t on il.track_id = t.track_id
INNER JOIN genre g on t.genre_id = g.genre_id
WHERE g.name = 'Rock'
GROUP BY c.email, c.first_name, c.last_name, g.name
ORDER BY c.email ASC;
```

OUTPUT

	email character varying (50)	first_name character	last_name character	genre_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

Level : Medium

Q. 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 ar.name as artist_nameORband,(COUNT(ar.artist_id)) as  
total_rock_music  
FROM artist ar  
INNER JOIN album al on ar.artist_id = al.album_id  
INNER JOIN track t on al.album_id = t.album_id  
INNER JOIN genre g on t.genre_id = g.genre_id  
WHERE g.name LIKE 'Rock'  
GROUP BY ar.name  
ORDER BY (COUNT(ar.artist_id)) DESC  
LIMIT 10;
```

OUTPUT

	artist_nameorband	total_rock_music
1	The Police	30
2	Green Day	20
3	David Coverdale	20
4	Ed Motta	20
5	Sir Georg Solti & Wiener Philharmoniker	20
6	Scholars Baroque Ensemble	18
7	Charlie Brown Jr.	17
8	Nicolaus Esterhazy Sinfonia	17
9	Berliner Philharmoniker & Hans Rosba...	17
10	Pink Floyd	17

Level : Medium

Q. 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_song_length  
    FROM track )  
ORDER BY milliseconds DESC;
```

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

Level : Hard

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

```
WITH top_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, ta.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 top_selling_artist ta ON ta.artist_id = alb.artist_id
GROUP BY 1,2,3,4
ORDER BY 5 DESC;
```

OUTPUT

	customer_id	first_name	last_name	artist_name	amount_spent
	integer	character	character	character varying (120)	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

Level : Hard

Q. 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 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

OUTPUT

	purchases bigint	country character varying (50)	name character varying (120)	genre_id character varying	row_no bigint
1	32	Argentina	Latin	7	1
2	46	Australia	Rock	1	1
3	21	Austria	Jazz	2	1
4	17	Belgium	Reggae	8	1
5	99	Brazil	Latin	7	1

Level : Hard

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 c.customer_id, c.first_name, c.last_name,
i.billing_country, SUM(i.total) as total_spent,
ROW_NUMBER()
OVER(Partition by billing_country ORDER BY SUM(total) DESC) as row_no
```

FROM customer c

JOIN invoice i on c.customer_id = i.customer_id

GROUP BY 1,2,3,4

ORDER BY 4 ASC, 5 DESC

)

```
SELECT * FROM customer_with_country
WHERE row_no <= 1;
```

OUTPUT

	customer_id	first_name	last_name	billing_country	total_spent	row_no
	integer	character	character	character varying (3)	double precision	bigint
1	56	Diego	Gutiérrez	Argentina	39.6	1
2	55	Mark	Taylor	Australia	81.18	1
3	7	Astrid	Gruber	Austria	69.3	1
4	8	Daan	Peeters	Belgium	60.38999999	1
5	1	Luís	Gonçalves	Brazil	108.89999999	1

Conclusion

The Music Store Analysis project has been a dynamic journey, highlighting the pivotal role of joins in unraveling complex queries within star schemas. Navigating through diverse data tables with PostgreSQL on PgAdmin4 not only sharpened my technical prowess but also instilled a strategic problem-solving mindset. The challenges encountered served as stepping stones, fostering a deeper understanding of intricate database structures.

I encourage every data aspirant to embrace such challenges, as they provide a unique opportunity for growth and skill enhancement in the fascinating realm of data analysis.