



MUSIC

MUSIC



## ABOUT US

## ABOUT US

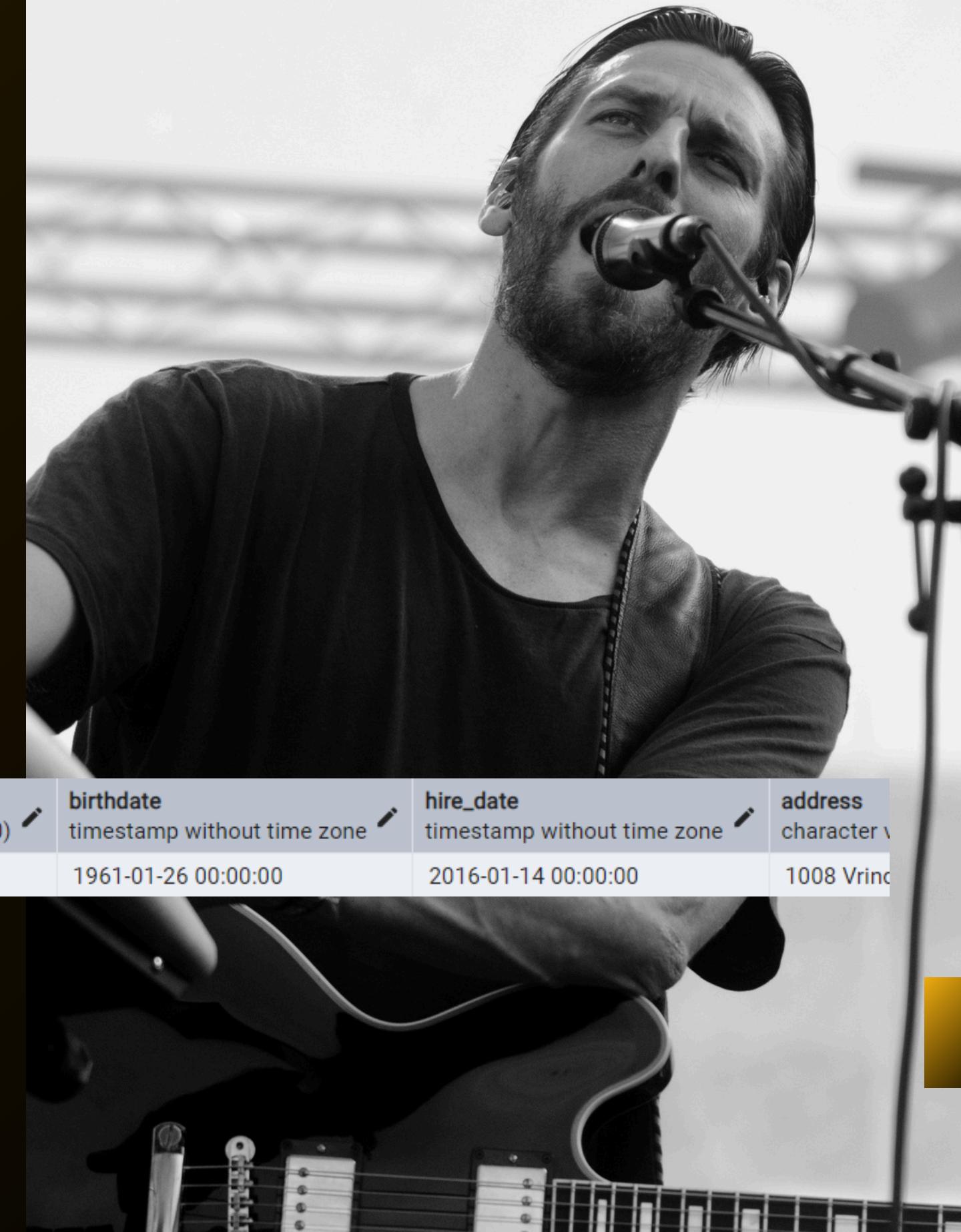
Our project, Music Store Analysis, is a data-driven study conducted using SQL to gain insights into sales trends, artists trend and many more in a music store. We structured our analysis around key predefined questions to extract meaningful insights and improve decision-making.



## Q1. Who is the senior most employee based on job title?

```
select * from employee  
ORDER BY levels desc  
limit 1
```

	employee_id [PK] character varying (50)	last_name character (50)	first_name character (50)	title character varying (50)	reports_to character varying (30)	levels character varying (10)	birthdate timestamp without time zone	hire_date timestamp without time zone	address character v
1	9	Madan	Mohan	Senior General Manager	[null]	L7	1961-01-26 00:00:00	2016-01-14 00:00:00	1008 Vrind



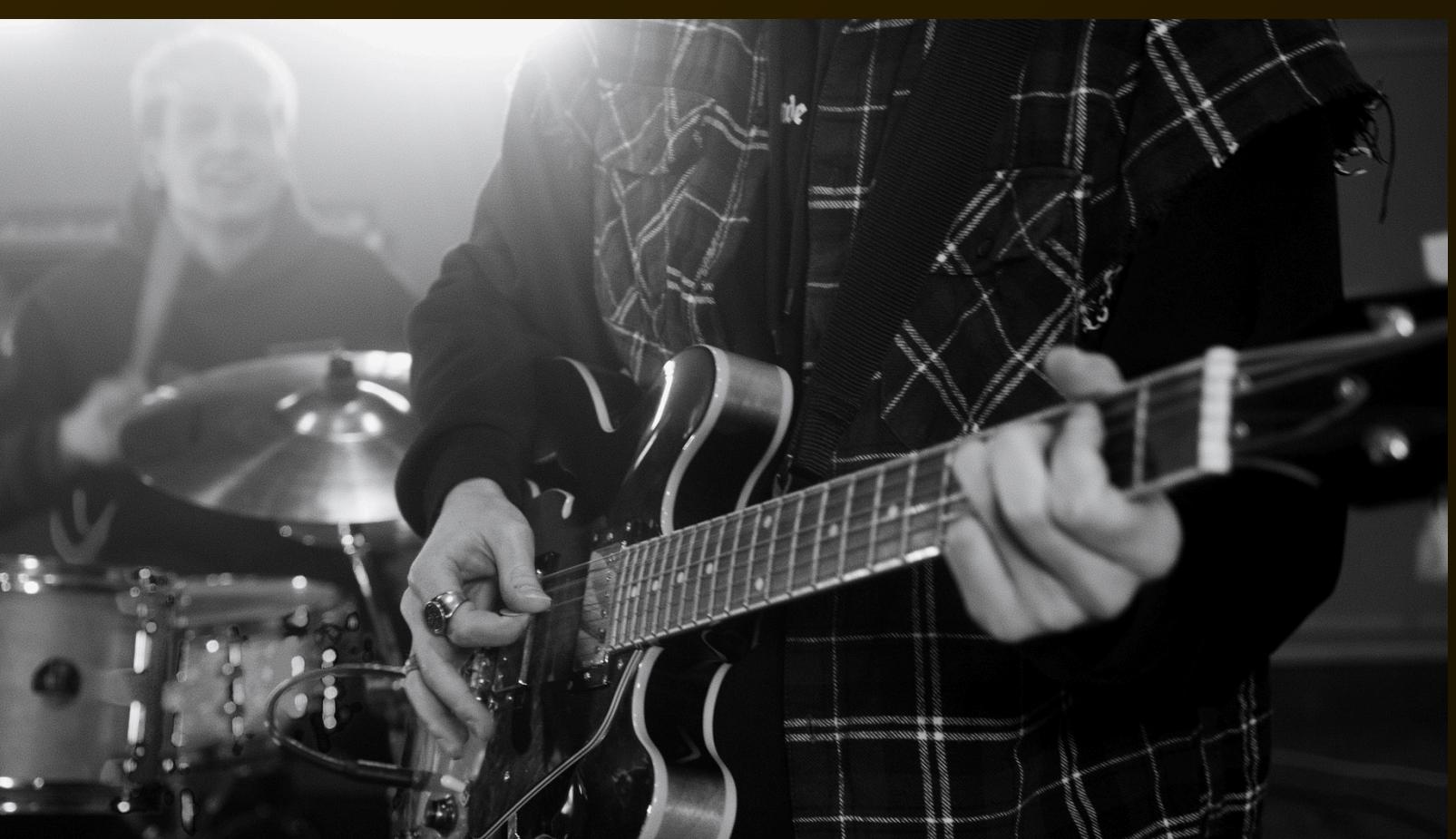


## Q2. Which countries have the most Invoices?

```
select COUNT(*) as c, billing_country
from invoice
group by billing_country
order by c desc
limit 10;
```

	c bigint	billing_country character varying (30)
1	131	USA
2	76	Canada
3	61	Brazil
4	50	France
5	41	Germany
6	30	Czech Republic
7	29	Portugal
8	28	United Kingdom
9	21	India
10	13	Chile





## Q3. What are top 3 values of total invoice?

```
select total from invoice  
order by total desc  
limit 3
```



	total double precision	lock
1	23.759999999999998	
2		19.8
3		19.8



**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?**

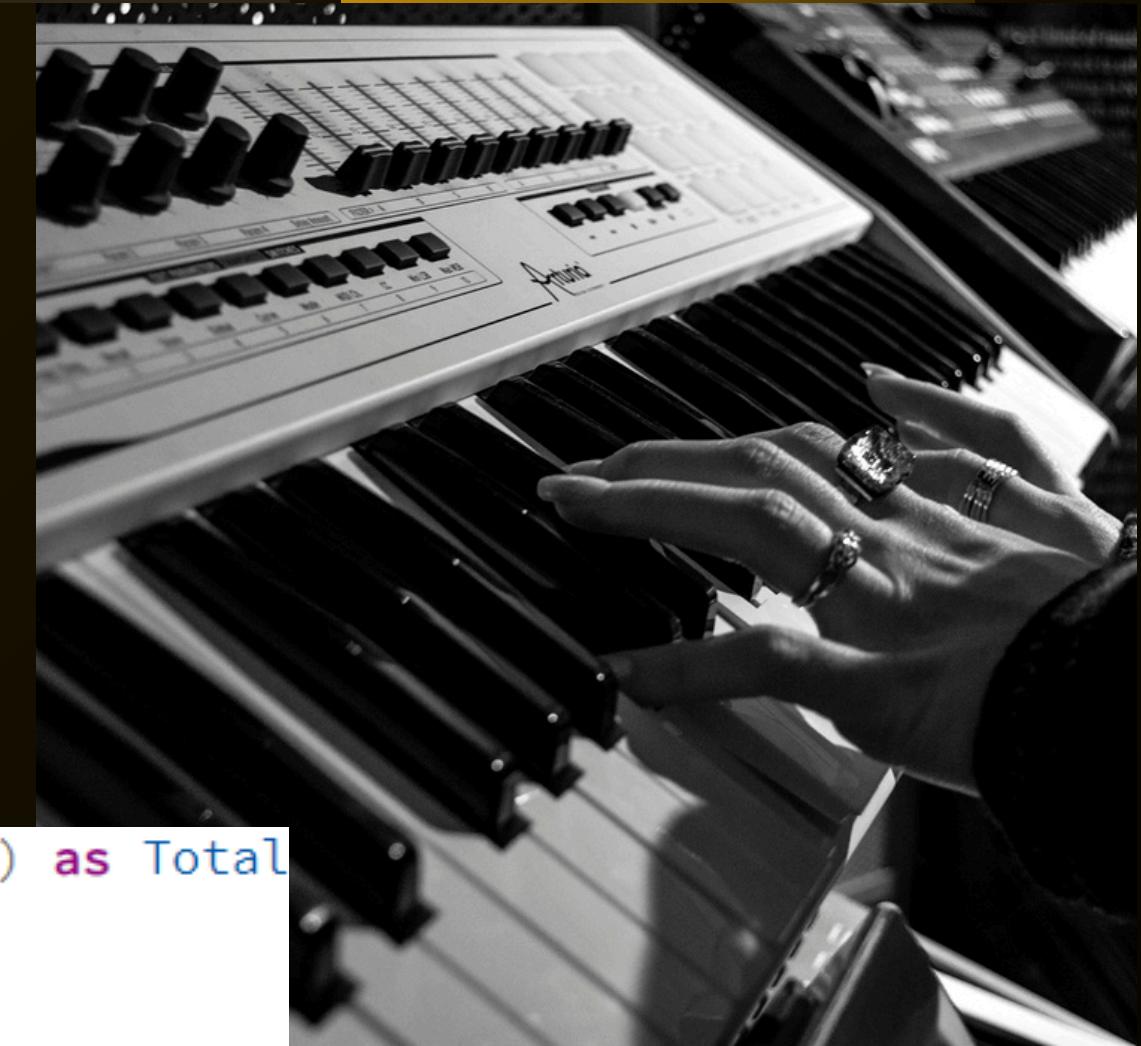
```
SELECT SUM(total) as invoice_total, billing_city
from invoice
group by billing_city
order by invoice_total desc
LIMIT 10;
```

	invoice_total double precision	billing_city character varying (30)
1	273.24000000000007	Prague
2	169.29	Mountain View
3	166.32	London
4	158.4	Berlin
5	151.47	Paris
6	129.69	São Paulo
7	114.8399999999997	Dublin
8	111.8699999999999	Delhi
9	108.8999999999998	São José dos Campos
10	106.9199999999999	Brasília



**Q5. 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 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
```



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



# Q6. 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
LIMIT 10;
```

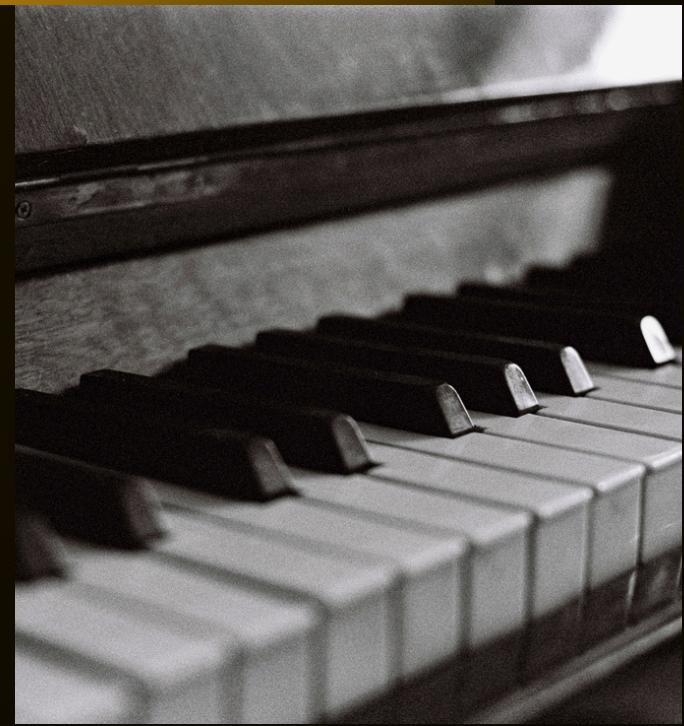
	email character varying (50)	first_name character (50)	last_name character (50)
1	aaronmitchell@yahoo.ca	Aaron	Mitchell
2	alero@uol.com.br	Alexandre	Rocha
3	astrid.gruber@apple.at	Astrid	Gruber
4	bjorn.hansen@yahoo.no	Bjørn	Hansen
5	camille.bernard@yahoo.fr	Camille	Bernard
6	daan_peeters@apple.be	Daan	Peeters
7	diego.gutierrez@yahoo.ar	Diego	Gutiérrez
8	dmiller@comcast.com	Dan	Miller
9	dominiquelefrevre@gmail.com	Dominique	Lefebvre
10	edfrancis@yachoo.ca	Edward	Francis



**Q7. 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.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
```

	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
7	51	Queen	45
8	142	The Rolling Stones	41
9	76	Creedence Clearwater Revival	40
10	52	Kiss	35





Q8. 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_track_length
    FROM track)
ORDER BY milliseconds DESC
LIMIT 10;
```

	name character varying (150)	milliseconds integer
1	Occupation / Precipice	5286953
2	Through a Looking Glass	5088838
3	Greetings from Earth, Pt...	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



# THANK

# YOU

For more such projects visit my  
portfolio

<https://peerlist.io/abhijeetranjan>

Presented By:  
Abhijeet Ranjan