



Music Store

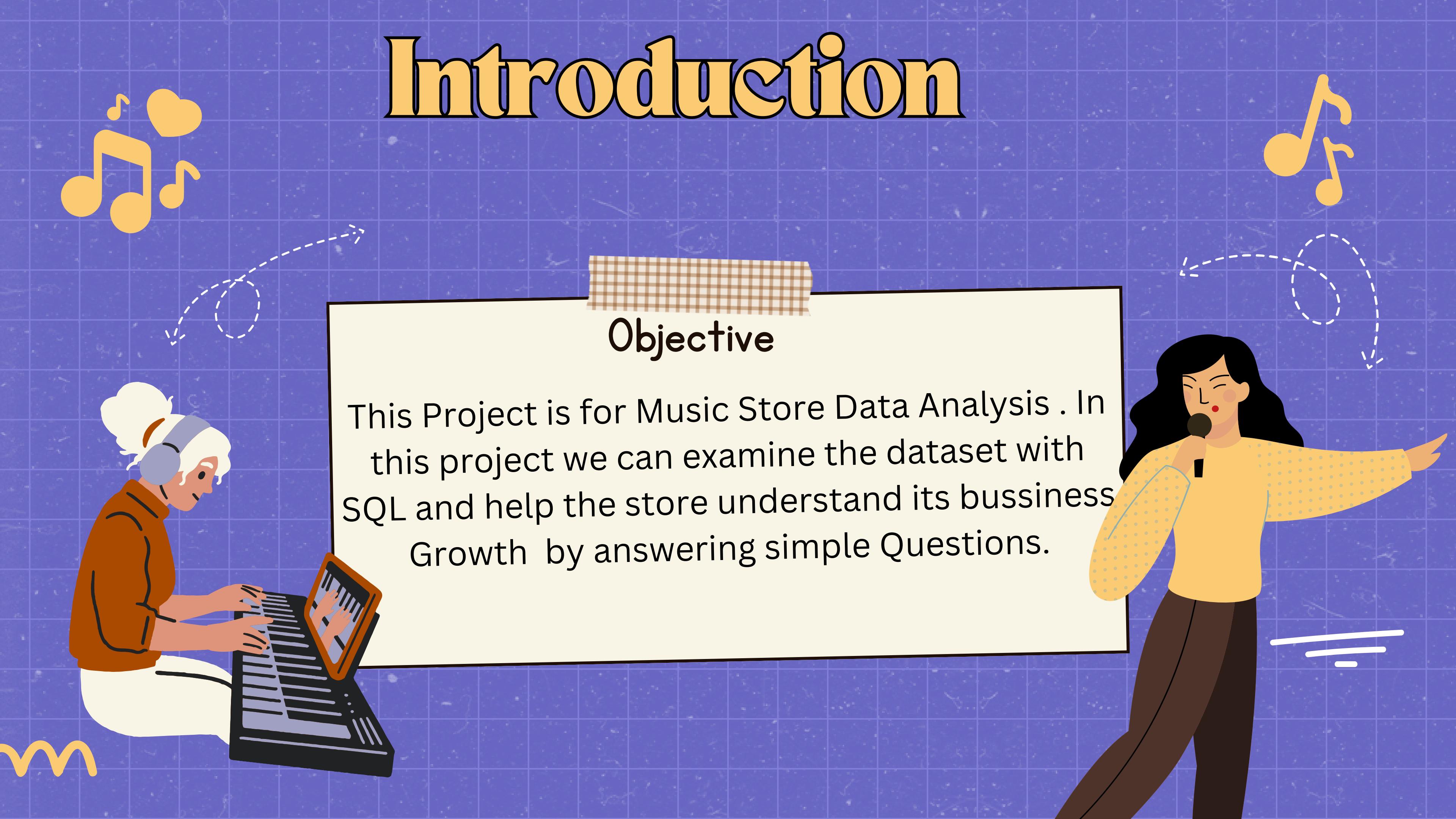
Data Analysis Project Using SQL

By:Pratibha Khade

Introduction

Objective

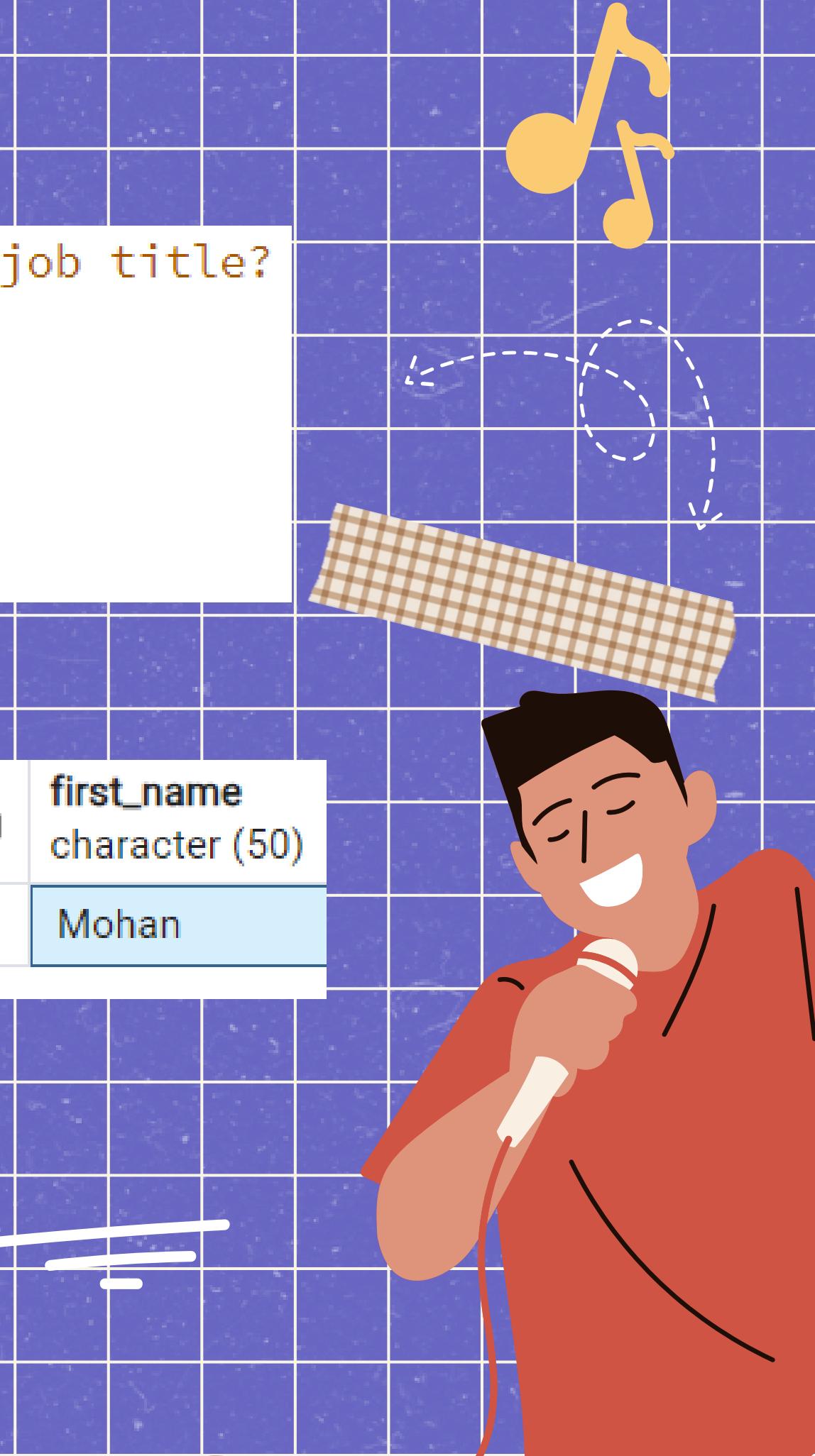
This Project is for Music Store Data Analysis . In this project we can examine the dataset with SQL and help the store understand its bussiness Growth by answering simple Questions.

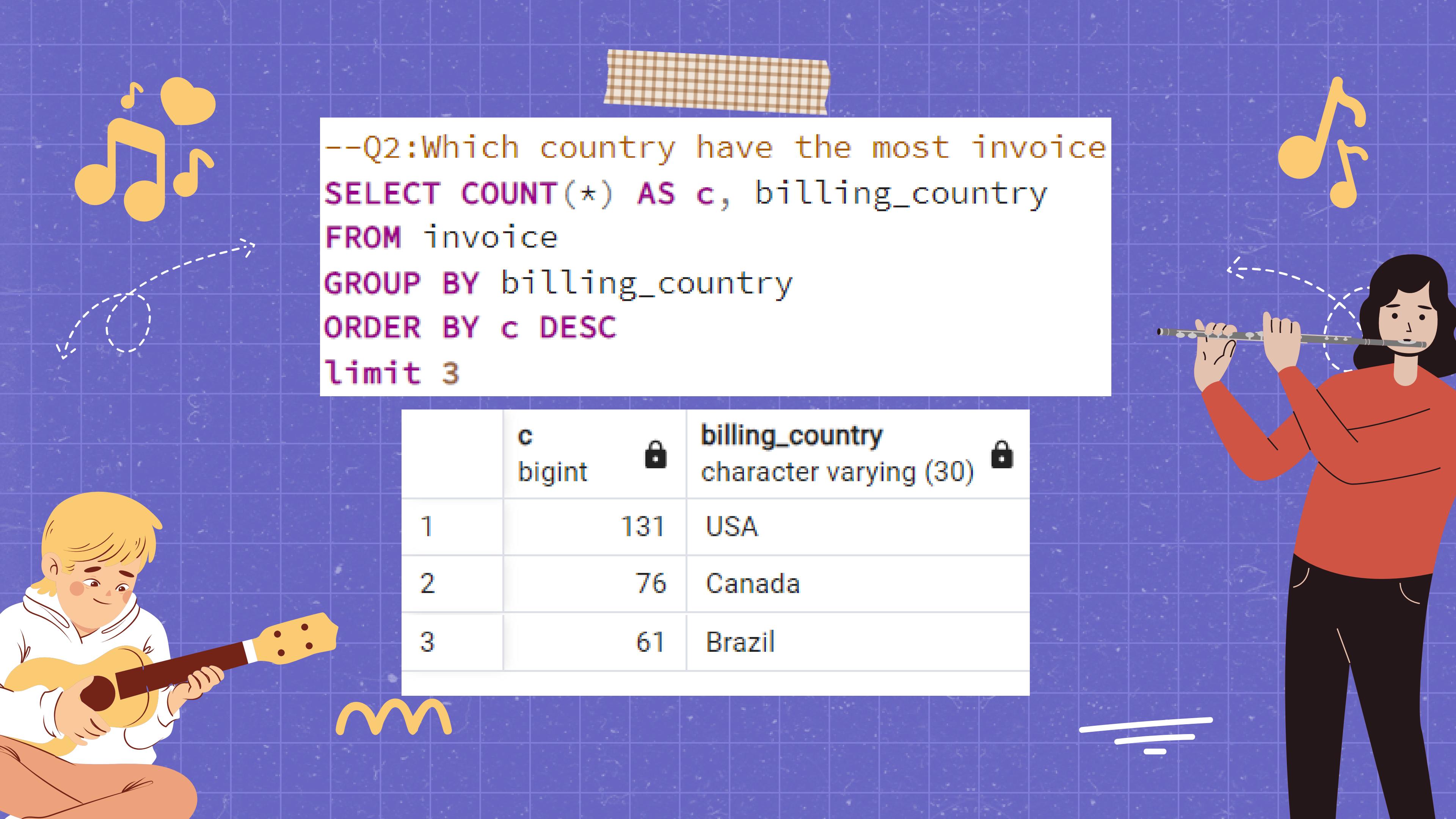


--Q1: Who is the senior most employee based on job title?

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

	title character varying (50)	last_name character (50)	first_name character (50)
1	Senior General Manager	Madan	Mohan





--Q2: Which country have the most invoice

```
SELECT COUNT(*) AS c, billing_country
FROM invoice
GROUP BY billing_country
ORDER BY c DESC
LIMIT 3
```

	c bigint	billing_country character varying (30)
1	131	USA
2	76	Canada
3	61	Brazil

--Q3:What are top 3 values of total Invoice

```
SELECT total  
FROM invoice  
order by total desc  
limit 3
```

	total	double precision
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
```

	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



--Write query to return the email, first name, last name,
--& Genre of all Rock Music listeners.
--Return your list ordered alphabetically by email starting w

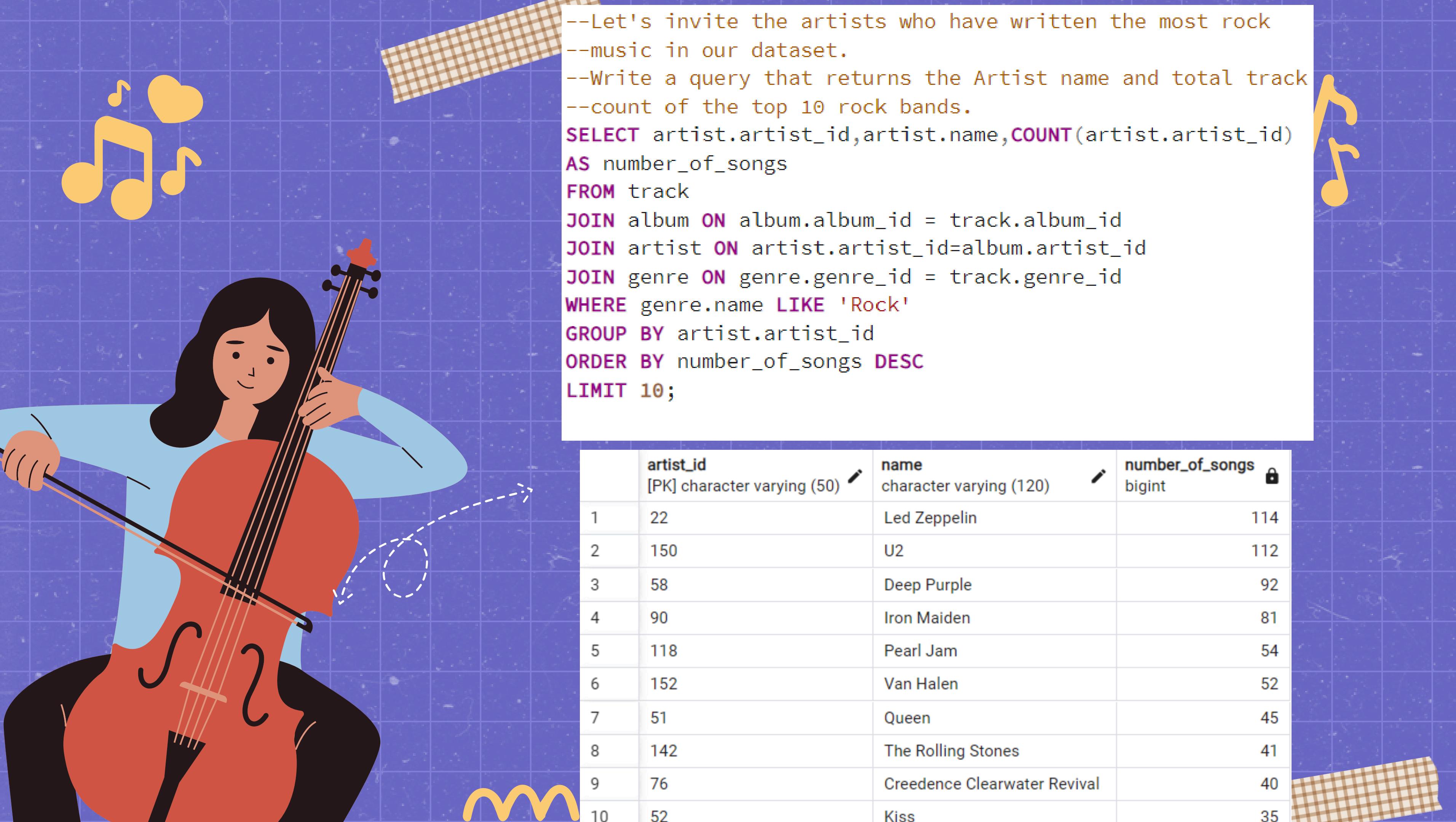
```
SELECT DISTINCT email,first_name,last_name
FROM customer
JOIN invoice ON customer.customer_id=invoice.customer_id
JOIN invoice_line ON invoice.invoice_id=invoice_line.invoice_
WHERE track_id IN(  

    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 (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



--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

--Return all the track names that have a song length longer than the average
--Return the Name and Milliseconds for each track. Order by
--the song length with the longest songs

```
SELECT name,milliseconds
FROM track
WHERE milliseconds >(
    SELECT AVG(milliseconds)AS avg_track_length
    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





--Q1: Find how much amount spent by each customer on artists?
--Write a query to return customer name, artist name and
--totalspent

```
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;
```

	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
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



```
--Q2: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
```

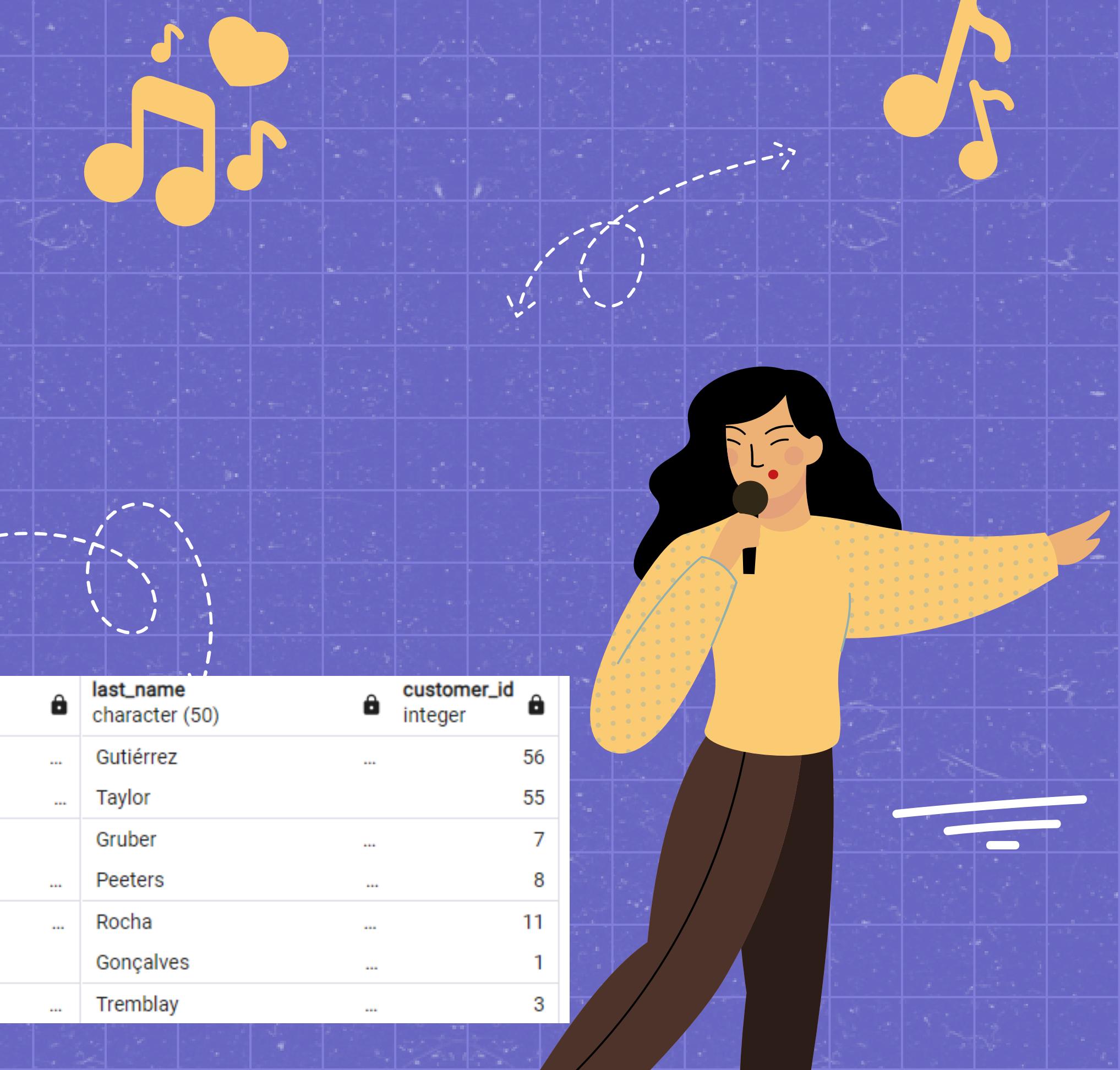
	purchases bigint	country character varying (50)	name character varying (120)	genre_id character varying (50)	rowno bigint
1	17	Argentina	Alternative & Punk	4	1
2	34	Australia	Rock	1	1
3	40	Austria	Rock	1	1
4	26	Belgium	Rock	1	1
5	205	Brazil	Rock	1	1

```
--Q3: 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

```
customer_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 customer_with_country
  GROUP BY billing_country )
SELECT cc.billing_country,cc.total_spending,cc.first_name,
cc.last_name,cc.customer_id
FROM customer_with_country cc
JOIN country_max_spending ms
ON cc.total_spending=ms.max_spending
ORDER BY 1;
```

	billing_country character varying (30)	total_spending double precision	first_name character (50)	last_name character (50)	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	69.3	Alexandre	Rocha	11
6	Brazil	108.89999999999998	Luís	Gonçalves	1
7	Canada	99.99	François	Tremblay	3





Thank You

pratibhakhade2004@gmail.com

