



PRESENTATION

MUSIC STORE ANALYSIS

P I C N I C R A U T A R A Y



INTRODUCTION MUSIC STORE ANALYSIS

"A music store is seeking to pinpoint crucial factors that can aid in boosting its business, diagnosing issues within the current business landscape, recognizing high-value customers, and gaining insights into new promotions and initiatives.

We will employ SQL to address these challenges."

**QUERIES ARE DIVIDED
INTO THREE CATEGORIES**

EASY

MODERATE

ADVANCED



EASY QUERIES

USING SIMPLE QUERIES AND JOINS

VISIT PROFILE



QUE NO-1

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

Query Query History

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

Data Output Messages Notifications

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

QUE NO-2

Q2: Which countries have the most Invoices?

```
SELECT count(*) as most_invoice,billing_country  
from invoice  
group by billing_country  
order by most_invoice DESC;|
```

| | most_invoice 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 |

QUE NO-3

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 |

QUE NO-4

Q4: 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;
```

| | billing_city character varying (30) | invoicetotal double precision |
|---|--|----------------------------------|
| 1 | Prague | 273.24000000000007 |
| 2 | Mountain View | 169.29 |
| 3 | London | 166.32 |
| 4 | Berlin | 158.4 |
| 5 | Paris | 151.47 |
| 6 | São Paulo | 129.69 |
| 7 | Dublin | 114.83999999999997 |

QUE NO-5

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,first_name,last_name, sum(total)as spending_money  
from customer  
join invoice ON customer.customer_id = invoice.customer_id  
group by customer.customer_id  
order by spending_money DESC  
limit 1;
```

| customer_id [PK] integer | first_name character | last_name character | spending_mo double precisi |
|-----------------------------|-------------------------|------------------------|-------------------------------|
| 5 | R | Madhav | 144.540000 |

MODERATE QUERIES

USING CTE QAND JOINS

[VISIT PROFILE](#)



QUE NO-1

Q1: 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 |

QUE NO-2

Q2: 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 |

QUE NO-3

Q3: 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;
```

| | 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 |

ADVANCE QUERIES

COMPLEX QUERIES AND JOINS

VISIT PROFILE



QUE NO-1

Q-1 Find how much amount spent by each customer on artists? Write a query to return the customer name, artist name, and total spent

```
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;
```

OUTPUT

Q-1 Find how much amount spent by each customer on artists? Write a query to return the customer name, artist name, and total spent

| | 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 |

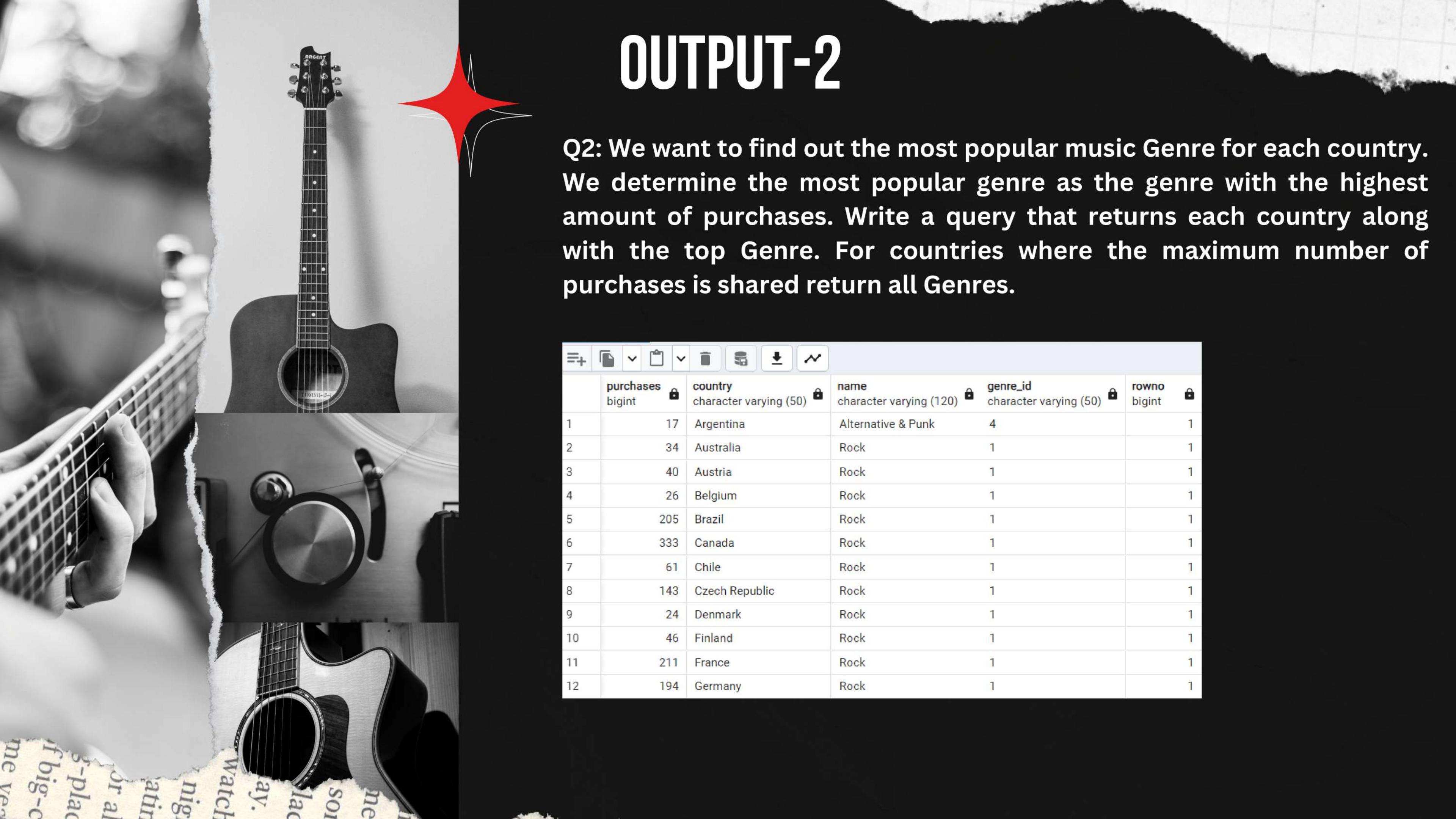
QUE NO-2

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
```

OUTPUT-2

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.

A collage of various guitars and musical instruments, including an acoustic guitar, a electric guitar, and a keyboard, set against a dark background.

| | 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 |
| 6 | 333 | Canada | Rock | 1 | 1 |
| 7 | 61 | Chile | Rock | 1 | 1 |
| 8 | 143 | Czech Republic | Rock | 1 | 1 |
| 9 | 24 | Denmark | Rock | 1 | 1 |
| 10 | 46 | Finland | Rock | 1 | 1 |
| 11 | 211 | France | Rock | 1 | 1 |
| 12 | 194 | Germany | Rock | 1 | 1 |

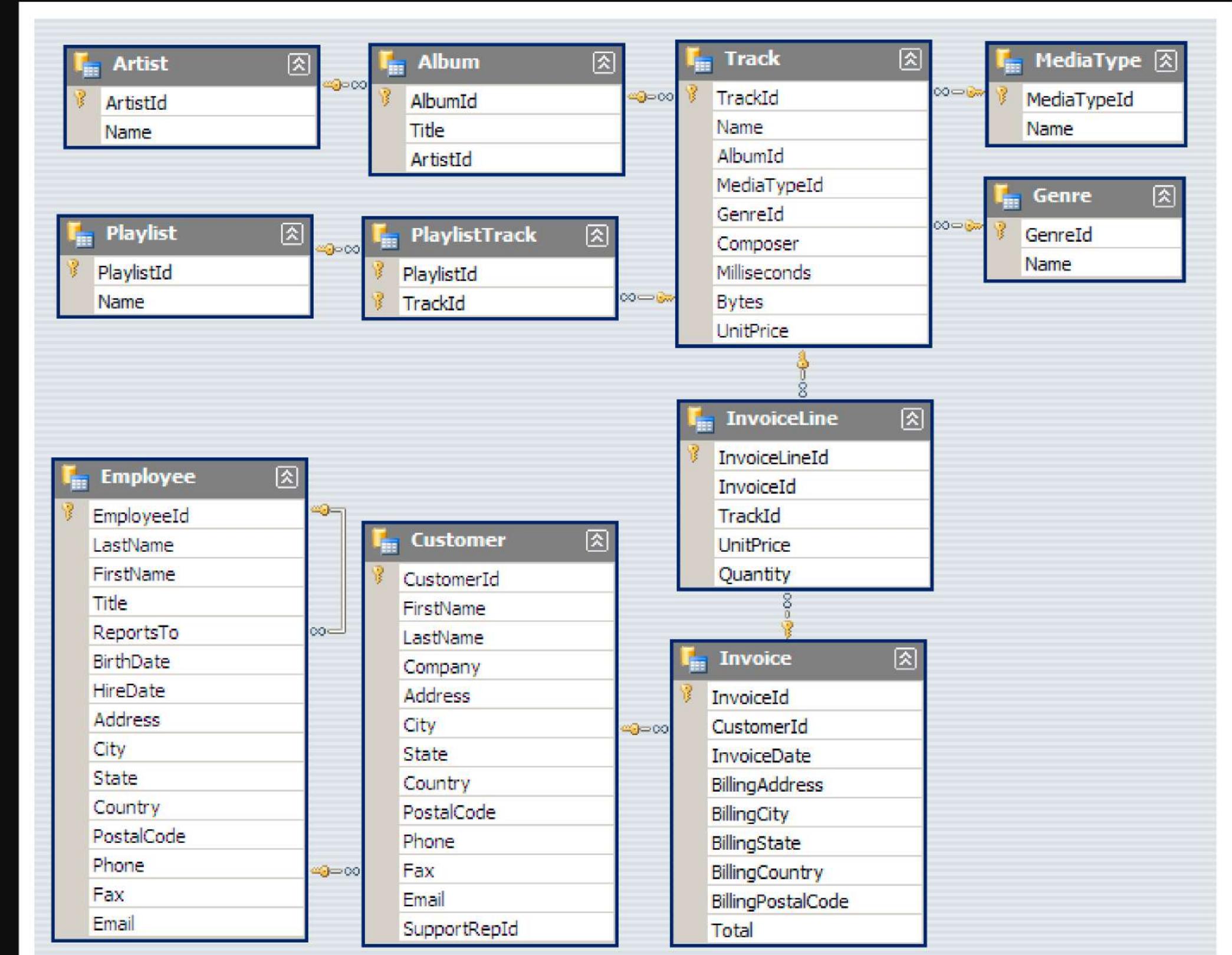
QUE NO-3

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 Customer_with_country AS (
    SELECT customer.customer_id, first_name, last_name, billing_country, SUM(total) AS total_spending,
    ROW_NUMBER() OVER(PARTITION BY billing_country ORDER BY SUM(total) DESC) AS RowNo
    FROM invoice
    JOIN customer ON customer.customer_id = invoice.customer_id
    GROUP BY 1,2,3,4
    ORDER BY 4 ASC,5 DESC)
SELECT * FROM Customer_with_country WHERE RowNo <= 1
```

| | customer_id integer | first_name character | last_name character | billing_country character varying (30) | total_spending double precision | rowno 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 | 60.38999999999999 | 1 |
| 5 | 1 | Luís | Gonçalves | Brazil | 108.8999999999998 | 1 |
| 6 | 3 | François | Tremblay | Canada | 99.99 | 1 |
| 7 | 57 | Luis | Rojas | Chile | 97.0200000000001 | 1 |
| 8 | 5 | R | Madhav | Czech Republic | 144.5400000000002 | 1 |
| 9 | 9 | Kara | Nielsen | Denmark | 37.6199999999999 | 1 |
| 10 | 44 | Terhi | Hämäläinen | Finland | 79.2 | 1 |
| 11 | 42 | Wyatt | Girard | France | 99.99 | 1 |
| 12 | 37 | Fynn | Zimmermann | Germany | 94.0500000000001 | 1 |

SCHEMA





THANK YOU