



SQL PROJECT

MUSIC STORE ANALYSIS

BY SARANSH SAHRMA



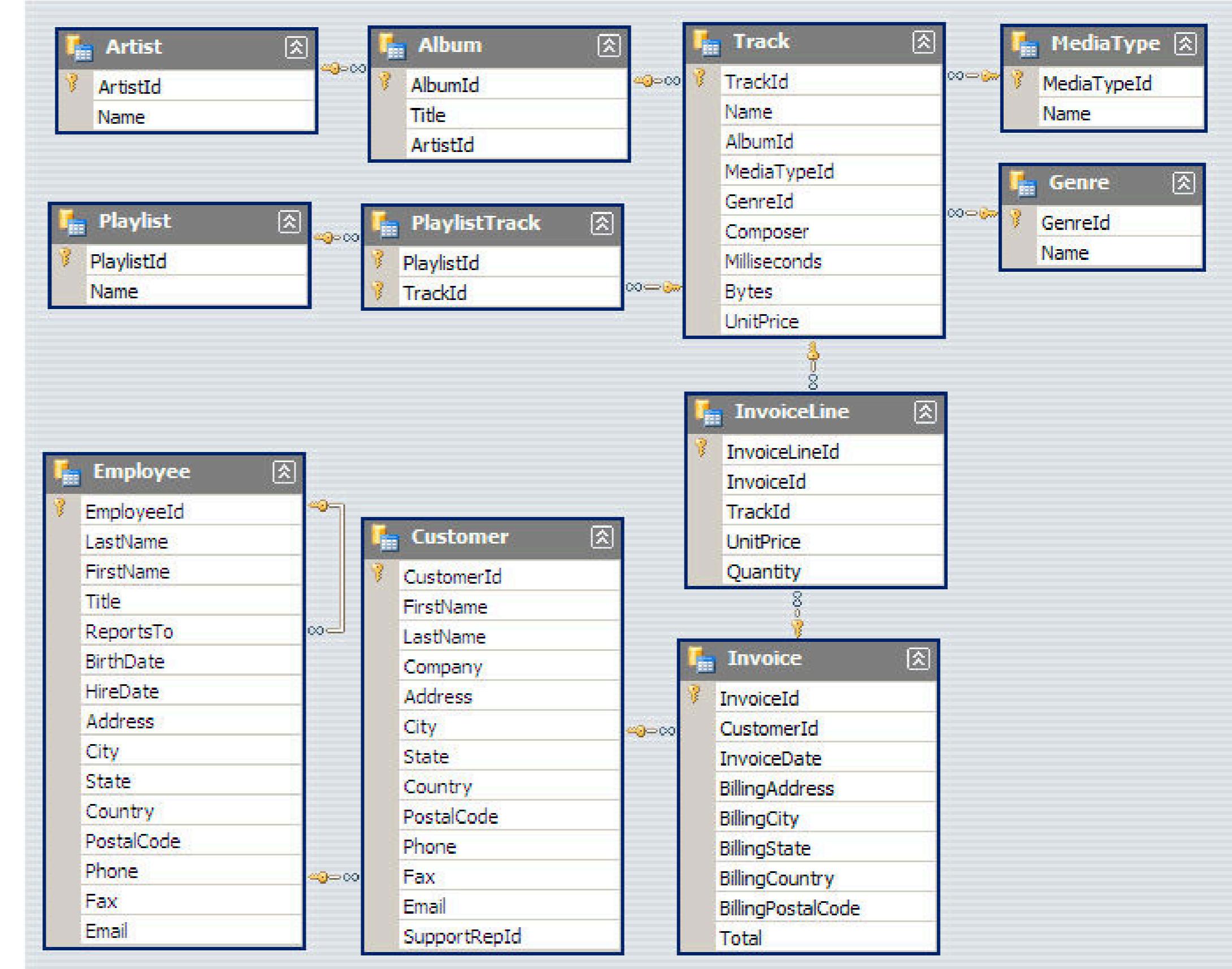


OBJECTIVE

The primary objective of the music store is to achieve sustainable business growth while addressing existing challenges.

We need to examine the dataset with SQL and help the music store understand its business growth by answering simple questions.

MUSIC DATABASE SCHEME





DIVISION OF QUESTIONS

EASY

Queries Include:
**SELECT, GROUP BY,
ORDER BY , LIMIT,
DESC**

MODERATE

Queries Include:
**JOINS, GROUP BY,
ORDER BY ,
LIMIT, DESC**

HARD

Queries Include:
**CTE (COMMON TABLE
EXPRESSIONS),
WINDOWS
FUNCTIONS**

EASY QUESTIONS

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

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

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

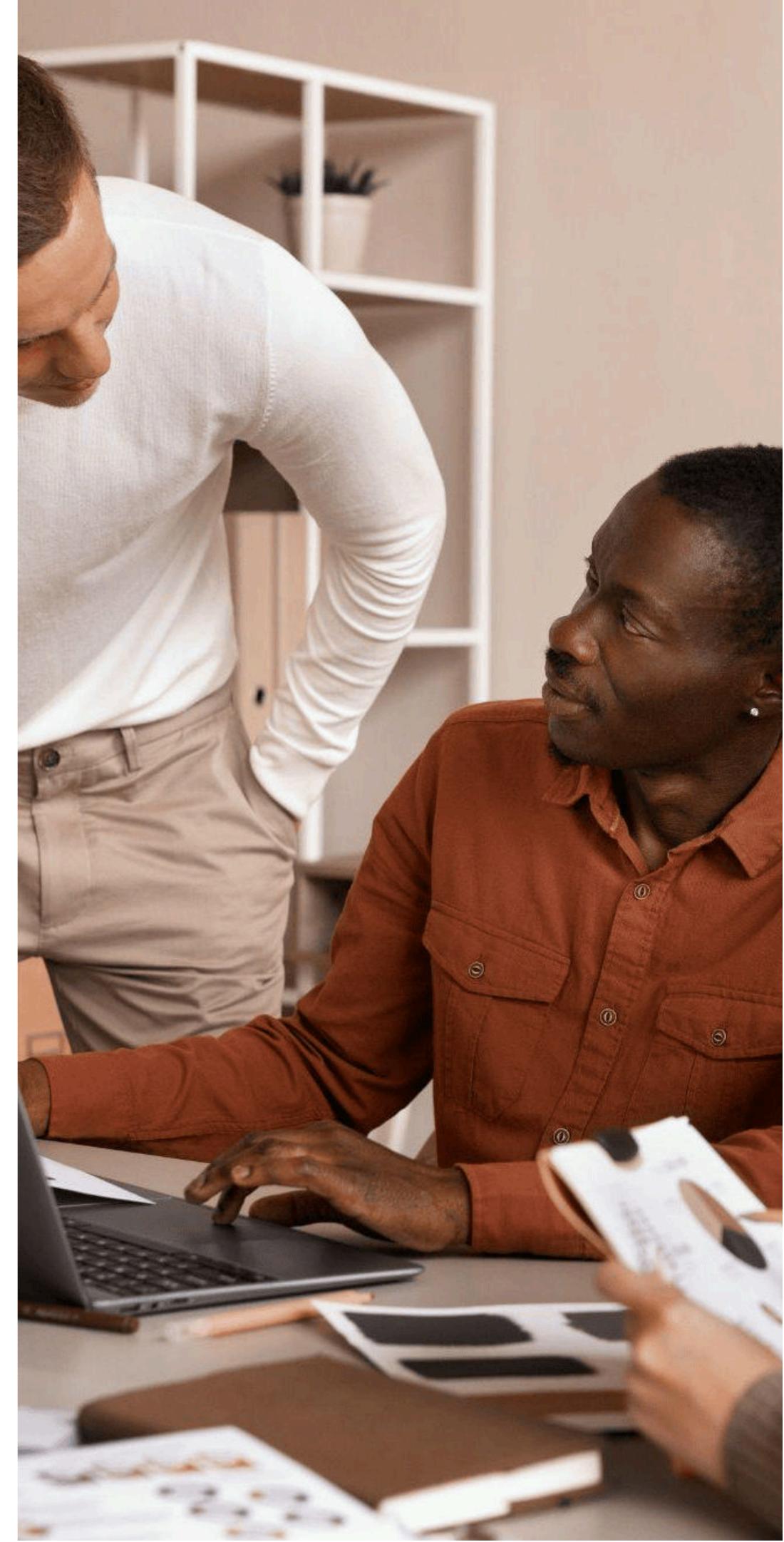
EASY QUESTIONS

Q2. Which countries have the most Invoices?

```
SELECT COUNT(*) AS cnt, billing_country
FROM invoice
GROUP BY billing_country
ORDER BY cnt DESC;
```

	cnt 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

Total rows: 24 of 24 Query complete 0



EASY QUESTIONS

Q3. What are top 3 values of total invoice?

```
SELECT total  
FROM invoice  
LIMIT 3;
```

	total double precision	🔒
1	15.84	
2	9.9	
3	1.98	

EASY QUESTIONS

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 billing_city, SUM(total) as invoice_total  
FROM invoice  
GROUP BY billing_city  
ORDER BY invoice_total DESC  
LIMIT 1;
```

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

EASY QUESTIONS

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

	customer_id [PK] integer	first_name character	last_name character	invoice_total double precision
1	5	R	Madhav	144.54000000000002

MODERATE QUESTIONS

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 as c
JOIN invoice as i ON c.customer_id = i.customer_id
JOIN invoice_line as il ON i.invoice_id = il.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	dieno.gutierrez@yahoo.ar	Dieno	Gutiérrez
Total rows: 59 of 59		Query complete 00:00:00.199	

MODERATE QUESTIONS

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

Total rows: 10 of 10 Query complete 00:00:00.196

MODERATE QUESTIONS

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
Total rows: 494 of 494		Query complete 00:00:00.110

HARD QUESTIONS

Q1. Find how much amount spent by each customer on artists? Write a query to return 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.qu
    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 customer AS c
JOIN invoice AS i ON i.customer_id = c.customer_id
JOIN invoice_line AS il ON il.invoice_id = i.invoice_id
JOIN track AS t ON t.track_id = il.track_id
JOIN album AS alb ON alb.album_id = t.album_id
JOIN best_selling_artist AS bsa ON bsa.artist_id = alb.artist_id
GROUP BY 1,2,3,4
ORDER BY 5 DESC;
```

	customer_id	first_name	last_name	artist_name	amount_spent
	integer	character	character	character varying (120)	double precision
1	46	Hugh	O'Reilly	Queen	27.71999999999985
2	38	Niklas	Schröder	Queen	18.81
3	3	François	Tremblay	Queen	17.82
4	34	João	Fernandes	Queen	16.83000000000002
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
Total rows: 43 of 43 Query complete 00:00:00.683					

HARD QUESTIONS

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 amounts 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

Total rows: 24 of 24

Query complete 00:00:00.095

HARD QUESTIONS

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 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

Total rows: 24 of 24 Query complete 00:00:00.095

Thanks!