Digital Music Store Analysis - SQL Project

Objective

This project is for beginners and will teach you how to analyze the music playlist database. You can examine the dataset with SQL and help the store understand its business growth by answering simple questions.

Question Set 1 -Easy

1) Who is the senior most employee based on job title?

```
SELECT * FROM employee
ORDER BY levels desc
limit 1
```

##RESULT

	employee_id [PK] character varying (50)	last_name character	first_name character	title character varying (50)	reports_to character varying (30)	levels character varying (10)
1	9	Madan	Mohan	Senior General Manager	[null]	L7

2) Which countries have the most Invoices?

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

##RESULT

	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
11	13	Ireland
12	11	Spain
13	11	Finland
14	10	Australia
15	10	Netherlands
16	10	Sweden
17	10	Poland
18	10	Hungary
19	10	Denmark
20	9	Austria
21	9	Norway

3) What are the top 3 values of total invoice?

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

##RESULT

	total double precision
1	23.759999999999998
2	19.8
3	19.8

4) 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 SUM(total) as invoice_total, billing_city
FROM invoice
group by billing_city
order by invoice_total desc
```

##RESULT

	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	114.83999999999997	Dublin
8	111.86999999999999	Delhi
9	108.89999999999998	São José dos Campos
10	106.91999999999999	Brasília
11	102.96000000000001	Lisbon
12	99.99	Bordeaux
13	99.99	Montréal
14	98.01	Madrid
15	98.01	Redmond
16	97.02000000000001	Santiago
17	94.05000000000001	Frankfurt
18	92.07000000000001	Orlando
19	91.08000000000001	Reno
20	91.08	Ottawa
21	86.13000000000002	Fort Worth
22	84.14999999999999	Tucson

Total rows: 53 of 53 Query complete 00:00:00.086

5) 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?

##Query:

```
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
```

##RESULT

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

Question Set 2 - Moderate

1. Write a 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;
```

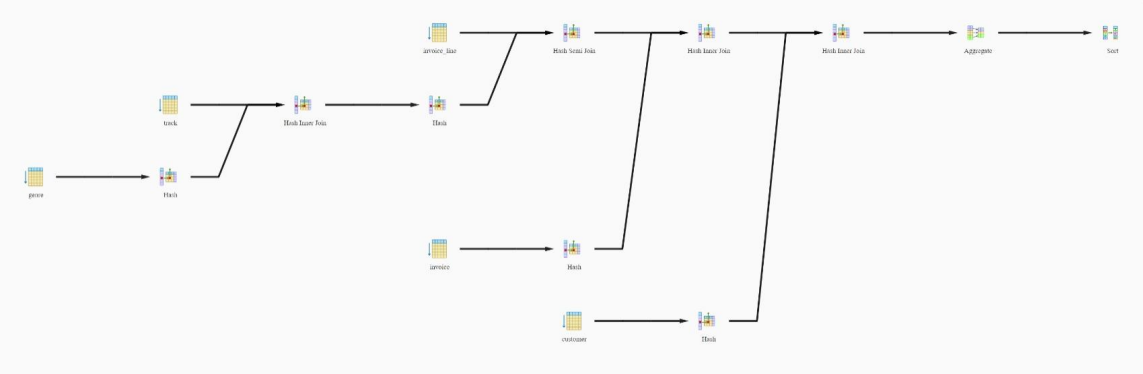
or,

```
SELECT DISTINCT email AS Email,first_name AS FirstName, last_name AS LastName,
genre.name AS Name
FROM customer
JOIN invoice ON invoice.customer_id = customer.customer_id
JOIN invoice_line ON invoice_line.invoice_id = invoice.invoice_id
JOIN track ON track.track_id = invoice_line.track_id
JOIN genre ON genre.genre_id = track.genre_id
WHERE genre.name LIKE 'Rock'
ORDER BY email;
```

##RESULT

	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
8	dmiller@comcast.com	Dan	Miller
9	dominiquelefebvre@gmail.c...	Dominique	Lefebvre
10	edfrancis@yachoo.ca	Edward	Francis

##Explain Plan



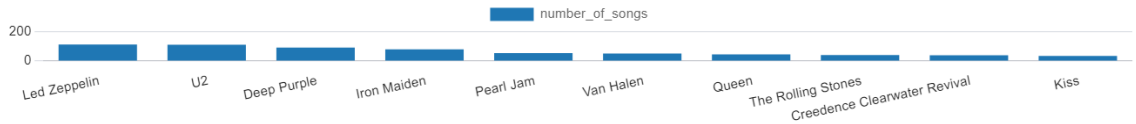
2. Let's invite the artist who has written the most rock music in our dataset. Write a query that returns the Artist name and total track count of the top 10 rock bands
##QUERY

```
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;
```

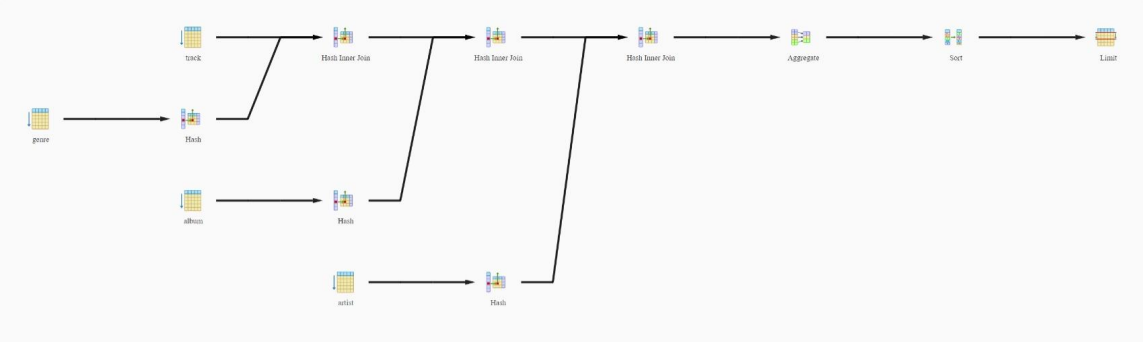
##RESULT

	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

##GRAPH CHART



##EXPLAIN PLAN



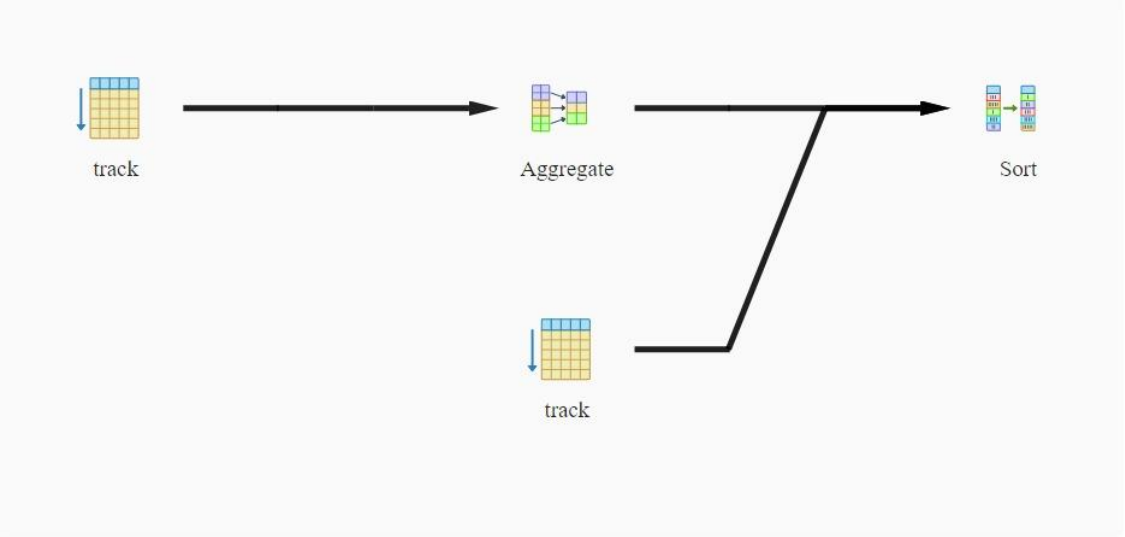
3. Return all the track names that have a song length longer than the average song length. Return the Name and Milliseconds for each track. Order by the song length with the longest songs listed first.

```
SELECT name,milliseconds
FROM track
WHERE milliseconds > (
        SELECT AVG(milliseconds) AS avg_track_length
        FROM track
)
ORDER BY milliseconds DESC;
```

##RESULT

	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

##EXPLAIN PLAN



Question Set 3 - Advance

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

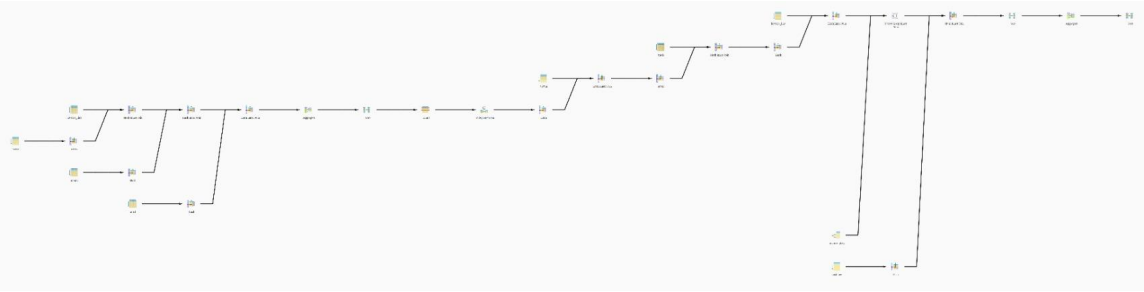
```
WITH best_selling_artist AS (
        SELECT artist.artist_id AS artist_id, artist.name AS artist_name,
        SUM(invoice_line.unit_price*invoice_line.quantity) AS total_sales
        FROM invoice_line
        JOIN track ON track.track_id = invoice_line.track_id
        JOIN album ON album.album_id = 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;
```

##RESULT

	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

##EXPLAIN PLAN



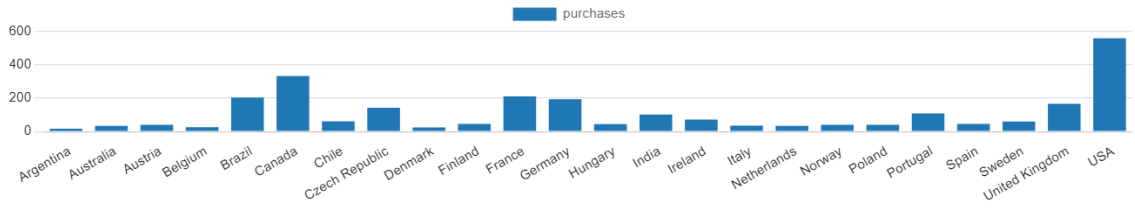
2. We want to find out the most popular music Genre for each country. We determine the most popular genre as the genre with the highest amount of purchases. Write a query that returns each country along with the top Genre. For countries where the maximum number of purchases is shared, return all Genres.

```
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
```

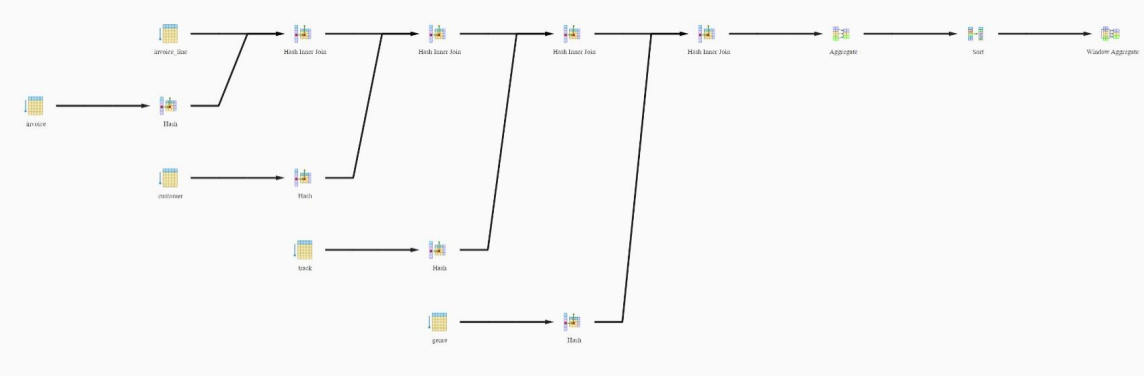
```
        JOIN invoice ON invoice.invoice_id = invoice_line.invoice_id
        JOIN customer ON customer.customer_id = invoice.customer_id
        JOIN track ON track.track_id = invoice_line.track_id
        JOIN genre ON genre.genre_id = track.genre_id
        GROUP BY 2,3,4
        ORDER BY 2 ASC, 1 DESC
)
SELECT * FROM popular_genre WHERE RowNo <= 1

##METHOD - 2
WITH RECURSIVE
   sales_per_country AS(
                SELECT COUNT(*) AS purchases_per_genre, customer.country,
genre.name, genre.genre_id
                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
        ),
        max_genre_per_country AS (SELECT MAX(purchases_per_genre) AS
max_genre_number, country
                                                FROM sales_per_country
                                                GROUP BY 2
                                                ORDER BY 2)

SELECT sales_per_country.*
FROM sales_per_country
JOIN max_genre_per_country ON sales_per_country.country =
max_genre_per_country.country
WHERE sales_per_country.purchases_per_genre =
max_genre_per_country.max_genre_number



##RESULT
```

	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
13	44	Hungary	Rock	1	1
14	102	India	Rock	1	1
15	72	Ireland	Rock	1	1
16	35	Italy	Rock	1	1
17	33	Netherlands	Rock	1	1
18	40	Norway	Rock	1	1
19	40	Poland	Rock	1	1
20	108	Portugal	Rock	1	1
21	46	Spain	Rock	1	1

##GRAPH



##EXPLAIN PLAIN



3. 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_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.billing_country = ms.billing_country
WHERE cc.total_spending = ms.max_spending
ORDER BY 1;

##METHOD - 2: CTE METHOD

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
```

##RESULT

	billing_country character varying (30)	total_spending double precision	first_name character	last_name character	customer_id integer
1	Australia	1.98	Richard	Cunningham	26
2	Brazil	8.91	Frank	Harris	16
3	Canada	19.8	Martha	Silk	31
4	Chile	5.9399999999999995	Lucas	Mancini	47
5	Denmark	8.91	Kathy	Chase	21
6	Finland	3.96	Jennifer	Peterson	15
7	Finland	3.96	Johannes	Van der Berg	48
8	France	9.9	Dan	Miller	20
9	Germany	10.89	Alexandre	Rocha	11
10	Germany	10.89	Puja	Srivastava	59
11	Hungary	5.9399999999999995	Robert	Brown	29
12	India	9.9	Fynn	Zimmermann	37
13	India	9.9	Daan	Peeters	8
14	Ireland	10.89	Edward	Francis	30
15	Norway	11.879999999999999	Frank	Ralston	24
16	Poland	10.89	Astrid	Gruber	7
17	Portugal	16.83	Diego	Gutiérrez	56
18	Spain	7.92	Marc	Dubois	41
19	Sweden	6.93	Hannah	Schneider	36
20	United Kingdom	9.9	John	Gordon	23
21	USA	17.82	Steve	Murray	54

##EXPLAIN PLAN