SQL PROJECT- MUSIC STORE DATA ANALYSIS

Question Set 1

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

	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 z
1	9	Madan ...	Mohan ...	Senior General Manager	[null]	L7	1961-01-26 00:00:00	2016-01-14 00:00:00

```
select * from employee
order by levels desc
limit 1
```

2. Which countries have the most Invoices?

	c bigint	billing_country character varying (30)
1	131	USA
2	76	Canada
3	61	Brazil
4	50	France
5	41	Germany

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

3. What are top 3 values of total invoice?

	billing_country character varying (30)	billing_state character varying (30)	billing_city character varying (30)	total double precision
1	France	None	Bordeaux	23.759999999999998
2	Canada	MB	Winnipeg	19.8
3	Canada	QC	Montréal	19.8

```
select billing_country,billing_state,billing_city,total from invoice
order by total desc
limit 3
```

4. 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

	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

```
select SUM(total) AS invoice_total, billing_city
from invoice
group by billing_city
order by invoice_total desc
```

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

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

```
select c.customer_id, c.first_name,c.last_name, sum(i.total) as total
from customer c
JOIN invoice i
ON c.customer_id = i.customer_id
group by c.customer_id
order by total desc
limit 1
```

Question Set 2 — Moderate

1. 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

	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	dominiquelefebvre@gmail.c...	Dominique	Lefebvre
10	edfrancis@yachoo.ca	Edward	Francis
11	eduardo@woodstock.com.br	Eduardo	Martins
12	ellie.sullivan@shaw.ca	Ellie	Sullivan

```sql
SELECT DISTINCT email,first_name,last_name
from customer c
JOIN invoice i ON c.customer_id = i.customer_id
JOIN invoice_line 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;
```

Customer and genre table are not directly linked so I used join multiple times(3) to make a link between them customer → customer_id ← invoice → invoice_id ←invoice_line → track_id ← track → genre_id ←genre.

2. 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

	artist_id [PK] character varying (50)	name character varying (120)	song_count 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

```
SELECT artist.artist_id, artist.name, Count(artist.artist_id) AS song_count
from artist
JOIN album ON artist.artist_id = album.artist_id
JOIN track ON album.album_id = track.album_id
JOIN genre ON track.genre_id = genre.genre_id
WHERE genre.name LIKE 'Rock'
Group by artist.artist_id
order by song_count desc
LIMIT 10
```

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

	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
11	The Long Patrol	2925008

```
SELECT name,milliseconds
from track
WHERE milliseconds > (
    SELECT AVG(milliseconds) as avg_song_length
    from track
)
order by milliseconds DESC
```

Question Set 3 — Advance

1. 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.quantity*invoice_line.unit_price) 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
)

SELECT c.customer_id, c.first_name, c.last_name,bsa.artist_name,
SUM(il.quantity * il.unit_price) 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 a ON a.album_id = t.album_id
JOIN best_selling_artist bsa ON bsa.artist_id = a.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	42	Wyatt ...	Girard ...	Frank Sinatra	23.75999999999999
3	3	François ...	Tremblay ...	The Who	19.799999999999997
4	6	Helena ...	Holý ...	Red Hot Chili Peppers	19.799999999999997
5	5	R ...	Madhav ...	Kiss	19.799999999999997
6	29	Robert ...	Brown ...	Creedence Clearwater Revival	19.799999999999997
7	32	Aaron ...	Mitchell ...	James Brown	19.799999999999997
8	22	Heather ...	Leacock ...	House Of Pain	18.81
9	46	Hugh ...	O'Reilly ...	Nirvana	18.81
10	38	Niklas ...	Schröder ...	Queen	18.81

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

1.

```
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
```

2.

```
WITH RECURSIVE
    sales_per_country AS(
        SELECT COUNT(*) purchase_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(purchase_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 max_genre_per_country.country = sales_per_country.country
WHERE sales_per_country.purchase_per_genre = max_genre_per_country.max_genre_number
```

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

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

1.

```
WITH RECURSIVE
    customer_with_country AS(
        SELECT c.customer_id,first_name,last_name,billing_country,SUM(total) AS total_spending
        FROM invoice
        JOIN customer c ON c.customer_id = invoice.customer_id
        GROUP BY 1,2,3,4
        ORDER BY 1,5 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 cm
ON cm.billing_country = cc.billing_country
WHERE cm.max_spending = cc.total_spending
ORDER BY 1
```

2.

```
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
```

	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	108.89999999999998	Luís ...	Gonçalves ...	1
6	Canada	99.99	François ...	Tremblay ...	3
7	Chile	97.02000000000001	Luis ...	Rojas ...	57
8	Czech Republic	144.54000000000002	R ...	Madhav ...	5
9	Denmark	37.61999999999999	Kara ...	Nielsen ...	9
10	Finland	79.2	Terhi ...	Hämäläinen ...	44