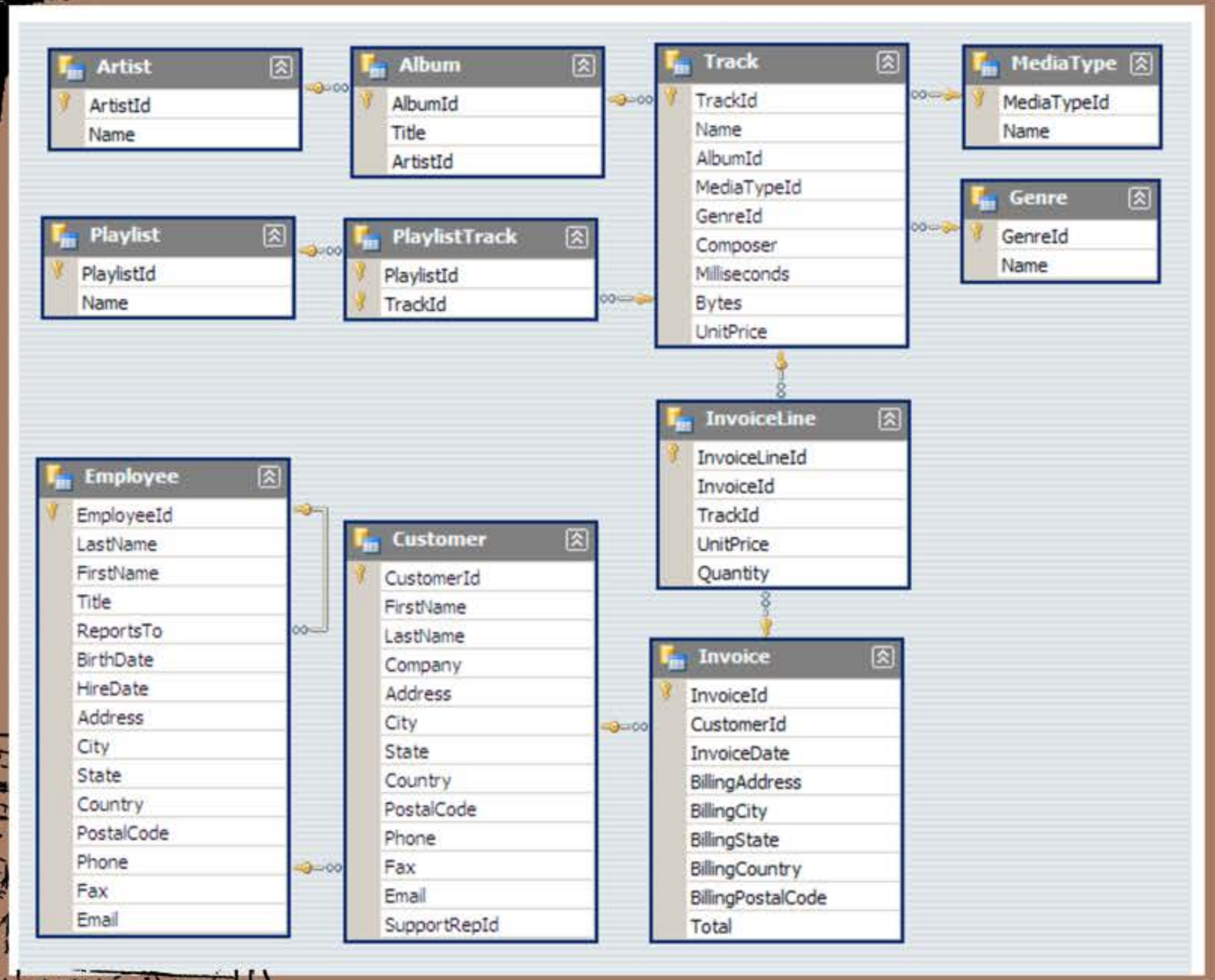
SQL Project On Music Store Data Analysis

HELLO

My name is Vaidehi Patel and this is my Sql project on Music Store Data Analysis. I have utilized Sql queries to solve questions related to Music Store Data Taken from kaggle .



Tables Present in Music Store Data Analysis Database

BASIC

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

SQL QUERY:

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

RESULT:

Result Grid | Filter Rows:

	title	last_name	first_name
▶	General Manager	Adams	Andrew

- **Which countries have the most Invoices?**

SQL QUERY:

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

RESULT:

Result Grid | Filter Rows:

	c	billing_country
▶	131	USA
	76	Canada
	61	Brazil
	50	France
	41	Germany
	30	Czech Republic
	29	Portugal
	28	United Kingdom
	21	India
	13	Ireland
	13	Chile
	11	Finland
	11	Spain
	10	Poland
	10	Denmark
	10	Australia
	10	Hungary

- **What are top 3 values of total invoice?**

SQL QUERY:

```
SELECT
    total
FROM
    invoice
ORDER BY total DESC
LIMIT 3
;
```

RESULT:

Result Grid | Filter

	total
▶	23.759999999999998
	19.8
	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**

SQL QUERY:

```
SELECT
    billing_city, SUM(total) AS InvoiceTotal
FROM
    invoice
GROUP BY billing_city
ORDER BY InvoiceTotal DESC
LIMIT 1;
```

RESULT:

Result Grid | Filter Rows:

billing_city	InvoiceTotal
Prague	273.24000000000007

- **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.**

SQL QUERY:

```
SELECT customer.customer_id, first_name, last_name,
 SUM(total) AS total_spending
FROM customer
JOIN invoice ON customer.customer_id = invoice.customer_id
GROUP BY customer.customer_id, first_name, last_name
ORDER BY total_spending DESC
LIMIT 1;
```

RESULT:

Result Grid | Filter Rows: | Export:

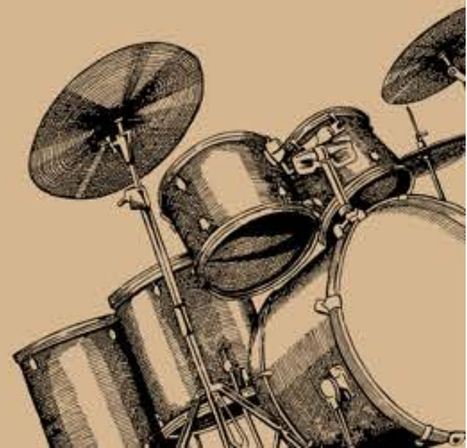
customer_id	first_name	last_name	total_spending
5	FrantiÅiek	WichterlovÃ¡	144.54000000000002

INTERMEDIATE

- **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.**

SQL QUERY:

```
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;
```



- **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.**

SQL QUERY:

```
SELECT artist.artist_id, artist.name, COUNT(track.track_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, artist.name
ORDER BY number_of_songs DESC
LIMIT 10;
```

RESULT:

Result Grid | Filter Rows: | Export:

artist_id	name	number_of_songs
3	Aerosmith	15
8	Audioslave	14
22	Led Zeppelin	14
5	Alice In Chains	12
1	AC/DC	10
23	Frank Zappa & Captain Beefheart	9
2	Accept	1

- **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.**

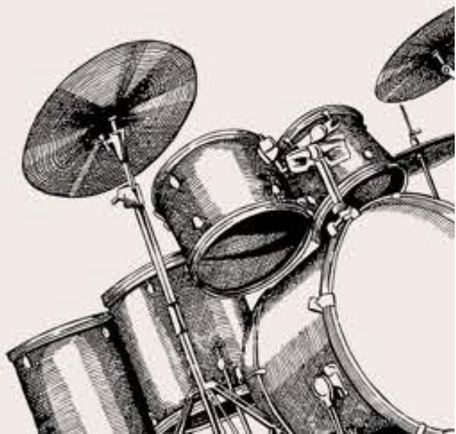
SQL QUERY:

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

RESULT:

Result Grid | Filter Rows:

name	milliseconds
How Many More Times	711836
Advance Romance	677694
Sleeping Village	644571
You Shook Me(2)	619467
Talkin' 'Bout Women Obviously	589531
Stratus	582086
No More Tears	555075
The Alchemist	509413
Wheels Of Confusion / The Straightener	494524
Book Of Thel	494393
You Oughta Know (Alternate)	491885
Terra	482429
Snoopy's search-Red baron	456071
Sozinho (Hitmakers Classic Mix)	436636
Master Of Puppets	436453
Stone Crazy	433397
Snowblind	420022



ADVANCED

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

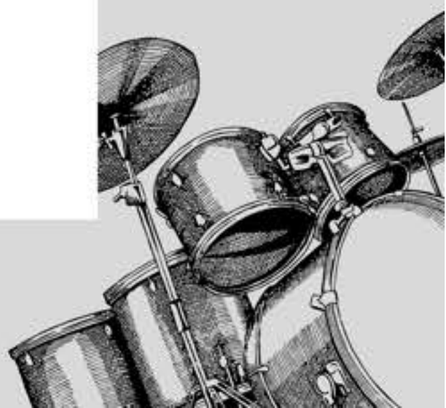
SQL QUERY:

```sql
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 artist.artist_id, artist.name
  ORDER BY total_sales 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 c.customer_id, c.first_name, c.last_name, bsa.artist_name
ORDER BY amount_spent DESC;
```

- **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.**

SQL QUERY:

```
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;
```



- **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.**

SQL QUERY:

```
WITH Customter_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 Customter_with_country WHERE RowNo <= 1;
```

- **Software used : MySql**
- **Language:Sql**