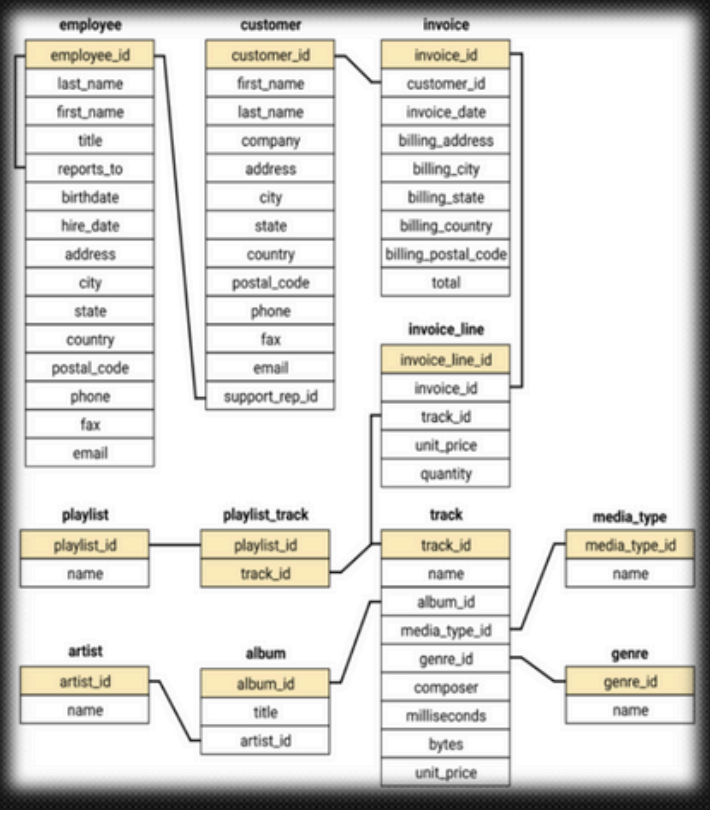
MUSIC STORE DATA ANALYSIS IN SQL

- You are hired as a data analyst at a music store, let's call it "Spotify". You're given access to a database with information about customers, singers, and more.
- As you explore connections between the data, your manager shares exciting news: the company plans to launch a new product! They need insights from the data to guide their decision-making.
- Your task is to analyse the database and provide answers to your manager's problem statements

Problem Statements

1. We want to assign an senior employee to lead that project. Who is the senior most employee based on job title?
2. Which county has most number of Invoices?
3. What is value of top 3 invoices ?
4. We would like to throw a promotional Music Festival in the city we made the most money. Which city has the best customers?
5. Who is the best customer? The customer who has spent the most money will be declared the best customer.
6. Details of customers who listens Rock music.
7. Let's invite the artists who have written the most rock music in our dataset.
8. We want to find out the most popular music Genre for each country.
9. We want to give gifts to top customer so Determine which customer has spent the most on music for each country.

Database Schema:



Solution 1 :

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

Solution 2 :

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

Solution 3 :

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

Solution 4 :

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

Solution 5 :

```
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
ORDER BY total_spending DESC
LIMIT 1;
```

Solution 6 :

```
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;
```

Solution 7 :

```
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;
```

Solution 8 :

```
WITH popular_genre AS
(
SELECT COUNT(invoice_line.quantity) AS purchases,
customer.country, genre.name as 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
```

Solution 9 :

```
 WITH Customter_with_country AS (
                        SELECT
customer.customer_id,first_name,last_name,billing_country,SU
M(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
order by total_spending desc
```