Introduction

Each row represents a relationship between a playlist and a track.
playlist_id identifies the playlist.
track_id identifies the track included in that playlist.
A single playlist can include multiple tracks.
A single track can appear in multiple playlists.
The dataset supports analysis of playlist structure, track distribution, and music recommendation patterns.

Solving SQL Queries With Result

who is the senior most employee based on job title.

```
select employee_id, concat(first_name,
    last_name)as full_name, title
        from employee
        order by levels desc
        limit 1;
```

	employee_id [PK] character varying (50)	full_name text	title character varying (50)
1	9	Mohan Madan...	Senior General Manag...

Retrieves the most senior employee from the employee table
Seniority is determined by the highest levels value

Solving SQL Queries With Result

which countries have the most invoices.

```
select count(*)as total_count,
billing_country
from invoice
group by billing_country
order by total_count desc;
```

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

Counts the total number of invoices for each country

Solving SQL Queries With Result

what are top 3 values of total invoice.

```
select*
from invoice
order by total desc
limit 3;
```

	invoice_id [PK] integer	customer_id integer	invoice_date timestamp without ti	billing_address character varying (120)	billing_city character varying	billing_state character vary	billing_country character varyir	billing_postal character varyi	total double precision
1	183	42	2018-02-09 00:00...	9, Place Louis Barthou	Bordeaux	None	France	33000	23.75999999999999
2	92	32	2017-07-02 00:00...	696 Osborne Street	Winnipeg	MB	Canada	R3L 2B9	19.
3	31	3	2017-02-21 00:00...	1498 rue Bélanger	Montréal	QC	Canada	H2G 1A7	19.

Retrieves data from the invoice table

Solving SQL Queries With Result

who is the best customer. the customer who has spend the most money will be declared the best customer.

write the query that returns the person who has spend the most money

```
select c.customer_id,concat(first_name,'', last_name)as full_name,
sum(i.total)as total
from customer as c join invoice as i
on c.customer_id = i.customer_id
group by c.customer_id
order by total desc
limit 1;
```

	customer_id [PK] integer	full_name text	total double precision
1	5	R Madhav ...	144.54000000000002

Joins customer and invoice tables using customer_id

Solving SQL Queries With Result

write query to return the email, firstname, lastname, & genere of all rock music listeners.

return your list ordered alphabetically by email starting with A.

```
SELECT DISTINCT c.email,CONCAT(c.first_name, ' ', c.last_name) AS full_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 il.track_id IN (SELECT t.track_id FROM track t JOIN genre g
    ON t.genre_id = g.genre_id
    WHERE g.name = 'Rock')
ORDER BY c.email;
```

	email character varying (50)	full_name text
1	aaronmitchell@yahoo.ca	Aaron Mitche...
2	alero@uol.com.br	Alexandre Roch...
3	astrid.gruber@apple.at	Astrid Gruber ...
4	bjorn.hansen@yahoo.no	Bjørn Hansen...
5	camille.bernard@yahoo.fr	Camille Bernar...

Retrieves unique customers using DISTINCT

Solving SQL Queries With Result

lets 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 ar.artist_id, ar.name AS artist_name,
    COUNT(t.track_id) AS number_of_songs
FROM track t JOIN album a ON t.album_id = a.album_id
JOIN artist ar ON a.artist_id = ar.artist_id
JOIN genre g ON t.genre_id = g.genre_id
WHERE g.name = 'Rock'
GROUP BY ar.artist_id, ar.name
ORDER BY number_of_songs DESC
LIMIT 10;
```

	artist_id [PK] character v	artist_name character varying	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

Retrieves artist ID and artist name

Solving SQL Queries With Result

return all the track names that have a song length langer 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

Selects track name and duration (milliseconds)

Solving SQL Queries With Result

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  ar.artist_id,ar.name AS artist_name,
 SUM(il.unit_price * il.quantity) AS total_sales
 FROM invoice_line il JOIN track t ON il.track_id = t.track_id
 JOIN album a ON t.album_id = a.album_id
 JOIN artist ar ON a.artist_id = ar.artist_id
 GROUP BY ar.artist_id, ar.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 i.customer_id = c.customer_id
JOIN invoice_line il ON i.invoice_id = il.invoice_id
JOIN track t ON il.track_id = t.track_id
JOIN album a ON t.album_id = a.album_id
JOIN best_selling_artist bsa ON a.artist_id = bsa.artist_id
GROUP BY c.customer_id, c.first_name,c.last_name,bsa.artist_name
ORDER BY amount_spent DESC;
```

	customer_id integer	first_name character (50)	last_name character (50)	artist_name character var	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

Uses a CTE (best_selling_artist) to find the top-selling artist by total revenue

Solving SQL Queries With Result

we wont 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.

```
with popular_genre as(
select count(il.quantity)as purchases, c.country, g.name, g.genre_id,
row_number()over(partition by c.country order by count(il.quantity)desc)as
rowno
from invoice_line as il join invoice as i
on il.invoice_id = i.invoice_id
join customer as c on i.customer_id = c.customer_id
join track as t on il.track_id = t.track_id
join genre as gon t.genre_id = g.genre_id
group by c.country, g.genre_id, g.name
order by c.country asc ,purchases desc)
select*
from popular_genre
where rowno <=1;
```

	purchases bigint	country character varying	name character varying (12	genre_id character va	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

Uses a CTE (popular_genre) to find the most popular genre per country

Solving SQL Queries With Result

write a query that determines the customer that has spent the most on music for each country.

write a query that returns the country alang 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 customer_with_country as (
    select c.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 as i join customer as c
 on i.customer_id = c.customer_id
 group by c.customer_id, first_name, last_name, billing_country
 order by billing_country asc, total_spending desc)
select*
from customer_with_country
where rowno <=1 ;
```

	customer_id integer	first_name character (50)	last_name character (50)	billing_country character varyir	total_spending double precision	rowno bigint
1	56	Diego	Gutiérrez	Argentina	39.6	1
2	55	Mark	Taylor	Australia	81.18	1
3	7	Astrid	Gruber	Austria	69.3	1
4	8	Daan	Peeters	Belgium	60.38999999999999	1
5	1	Luís	Gonçalves	Brazil	108.89999999999998	1

Uses a CTE to calculate customer spending per country

Conclusion

This project analyzes a digital music store dataset using SQL to uncover insights about customers, artists, genres, and sales performance. It connects playlists, tracks, invoices, and customer data to study music purchasing and listening patterns. Using joins, subqueries, window functions, and CTEs, the project identifies the best customers, top-selling artists, most popular genres by country, and countries with the highest sales. It also finds Rock music listeners and tracks longer than average. The analysis supports business decisions such as targeted marketing, customer segmentation, regional trend identification, and revenue optimization, demonstrating the practical use of SQL for real-world data analytics.

THANK YOU