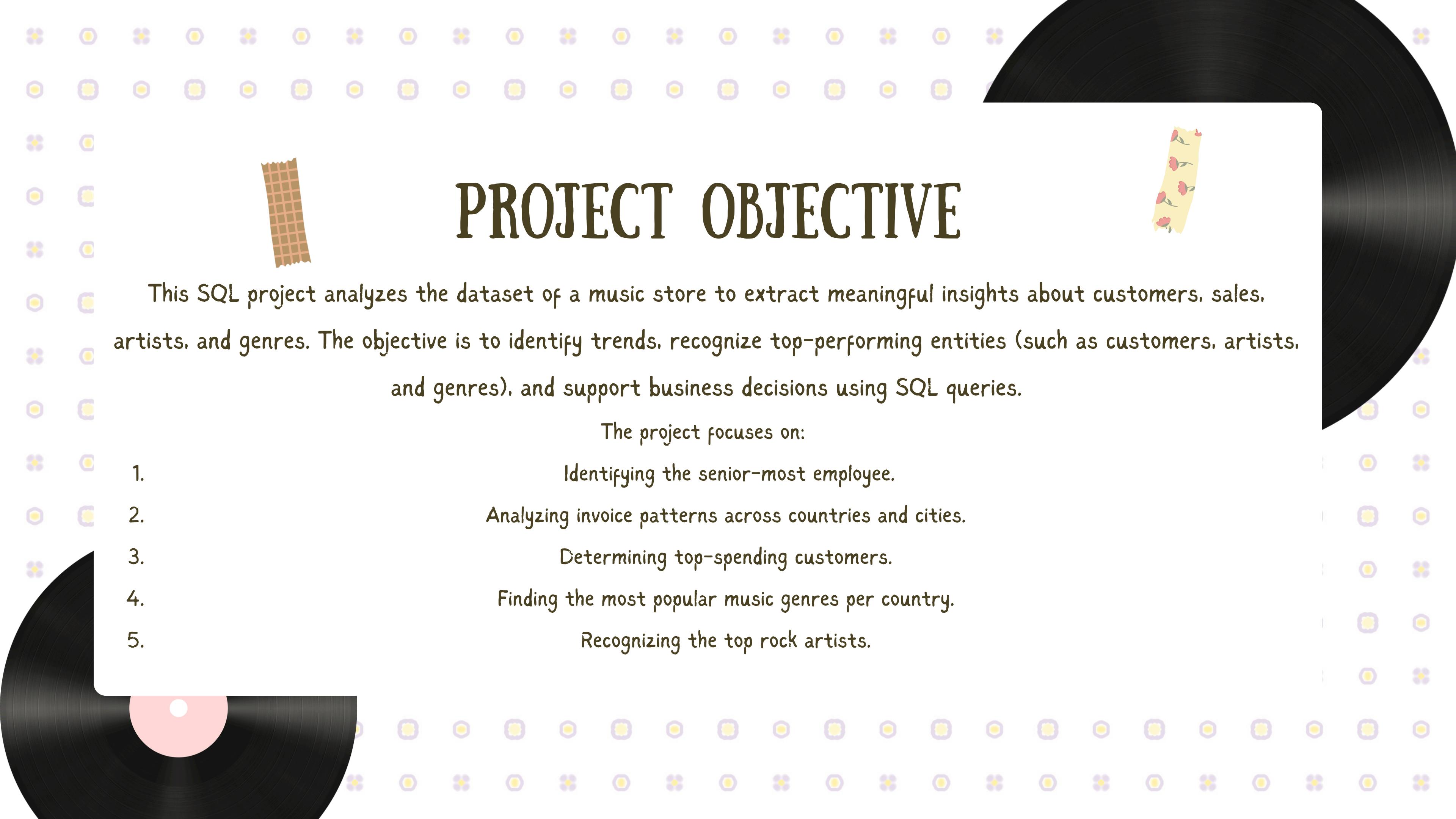




# MUSIC STORE ANALYSIS

## SQL

-harshmeet singh

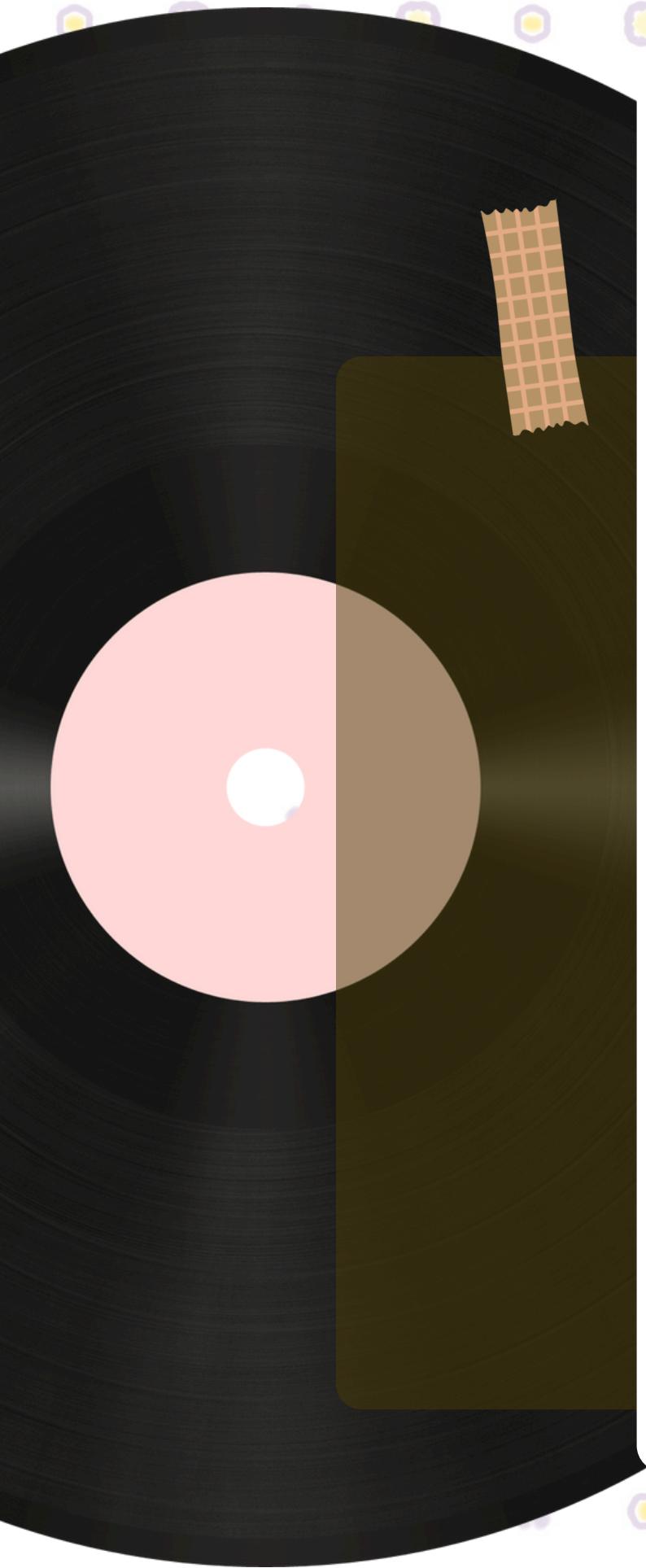


# PROJECT OBJECTIVE

This SQL project analyzes the dataset of a music store to extract meaningful insights about customers, sales, artists, and genres. The objective is to identify trends, recognize top-performing entities (such as customers, artists, and genres), and support business decisions using SQL queries.

The project focuses on:

1. Identifying the senior-most employee.
2. Analyzing invoice patterns across countries and cities.
3. Determining top-spending customers.
4. Finding the most popular music genres per country.
5. Recognizing the top rock artists.



# PROJECT PROCESS:

- Data Exploration – Retrieved tables, examined columns, and understood relationships.
- Data Querying – Used SQL queries with JOIN, GROUP BY, ORDER BY, and window functions.
- Analysis & Insights – Identified top customers, genres, artists, and revenue-generating locations.
- Business Recommendations – Provided insights for marketing, promotions, and sales strategies.



# QUESTIONS

- Q1 WHO IS THE SENIOR MOST EMPLOYEE BASED ON THE JOB TITILE?

```
SELECT * FROM EMPLOYEE ORDER BY LEVELS DESC LIMIT 1
```

- Q2 WHICH COUNTRIES HAVE THE MOST INVOICE

```
SELECT * FROM INVOICE
```

```
SELECT COUNT (*) AS C, BILLING_COUNTRY FROM INVOICE  
GROUP BY BILLING_COUNTRY ORDER BY C DESC
```

- Q3 WHAT ARE THE TOP THREE VALUES OF THE TOTAL INVOICE

```
SELECT TOTAL FROM INVOICE ORDER BY TOTAL DESC LIMIT 3
```



Q4 WHICH CITY HAS THE BEST CUSTOMER? WE WOULD LIKE TO THROW A PROMOTIONAL MUSIC FEST IN THE CITY WE MADE THE MOST MONEY. WRITE A QUERY THAT RETURN ONE CITY THAT HAS THE HIGHEST SUM OF INVOICE TOTALS. RETURN BOTH THE CITY NAME AND SUM OF ALL INVOICE TOTAL

```
SELECT * FROM INVOICE
```

```
SELECT SUM (TOTAL)AS INVOICE_TOTAL,BILLING_CITY FROM INVOICE GROUP BY  
BILLING_CITY ORDER BY INVOICE_TOTAL DESC
```

- Q5 WHO IS THE CUSTOMER? THE CUSTOMER WHO SPEND THE MOST MONEY WILL BE DECLARED THE BEST CUSTOMER. WRITE A QUERY THAT RETURNS THE PERSON WHO HAS SPEND THE MOST MONEY

```
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
```



- Q6 WRITE A QUERY TO RETURN THE EMAIL, FIRST NAME, LAST NAME, GENRE OF ALL ROCK MUSIC LISTNERS. RETURN YOUR LIST ORDERED ALPHABETICALLY BY EMAIL STARTING WITH A

```
SELECT DISTINCT  
    CUSTOMER.EMAIL,  
    CUSTOMER.FIRST_NAME,  
    CUSTOMER.LAST_NAME,  
    GENRE.NAME AS GENRE  
FROM CUSTOMER  
JOIN INVOICE ON CUSTOMER.CUSTOMER_ID = INVOICE.CUSTOMER_ID  
JOIN INVOICE_LINE ON INVOICE.INVOICE_ID = INVOICE_LINE.INVOICE_ID  
JOIN TRACK ON INVOICE_LINE.TRACK_ID = TRACK.TRACK_ID  
JOIN GENRE ON TRACK.GENRE_ID = GENRE.GENRE_ID  
WHERE GENRE.NAME = 'Rock'  
ORDER BY CUSTOMER.EMAIL
```



- Q7 LETS INVITE THE ARTIST 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
    ARTIST.ARTIST_ID, ARTIST.NAME,
    COUNT(TRACK.TRACK_ID) AS TOTAL_TRACK_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 = 'Rock'
GROUP BY ARTIST.ARTIST_ID, ARTIST.NAME
ORDER BY TOTAL_TRACK_COUNT DESC
LIMIT 10
```



- Q8 RETURN ALL THE TRACK NAME THAT HAVE THE SONG LENGTH LONGER THAN THE AVG SONG LENGTH. RETURN THE NAME AND MILLISECOND FOR EACH TRACK. ORDER BY THE SONG LENGTH WITH THE LONGEST SONG LISTED FIRST.

```
SELECT NAME, milliseconds  
FROM TRACK  
WHERE milliseconds > (  
    SELECT AVG(milliseconds) AS AVG_TRACK_LENGTH  
    FROM TRACK  
)  
ORDER BY milliseconds DESC
```



- Q9 FIND HOW MUCH AMOUNT SPEND BY EACH CUSTOMER ON ARTISTS? WRITE A QUERY TO RETURN CUSTOMER NAME, ARTIST NAME AND TOTAL SPEND



```
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 customer.customer_id, customer.first_name,
       customer.last_name,best_selling_artist.artist_name,
       sum (invoice_line.unit_price * invoice_line.quantity) as amount_spend
  from invoice
 join customer on customer.customer_id= invoice.customer_id
 join invoice_line on invoice_line.invoice_id= invoice.invoice_id
 join track on track.track_id= invoice_line.track_id
 join album on album.album_id = track.album_id
 join best_selling_artist on best_selling_artist.artist_id= album.artist_id
 group by 1,2,3,4
 order by 5 desc;
```

- Q10 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 THE EACH COUNTRY ALONG WITH THE TOP GENRE. FOR COUNTRY WHERE THE MAX NO. OF PURCHASES IS SHARED RETURN ALL GERNES.

```
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 row_no
  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 invoice_line.track_id= track.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 row_no <=1
```



- Q11 write a query that determines the customer that has spent most on the music for each country.write a query that returns country along with top customers and how much they spent. for country where top amount is spent shared. provide all customers who spend this amount

```
with customer_with_country as (
    select
        customer.customer_id, first_name, last_name, billing_country,
        sum(invoice.total) as total_spend,
        row_number () over(partition by billing_country order by sum
            (invoice.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
```



# PROJECT OUTCOME & INSIGHTS:

- The highest-ranking employee in the company was identified.
- The country with the most invoices was found, helping target key markets.
- The top city in total sales was identified for potential promotional events.
- The most valuable customer was determined based on total spending.
- The most listened-to music genre in each country was discovered, helping tailor marketing efforts.
- The top rock artists were identified, useful for organizing music events.
- The longest songs were found, highlighting track duration trends.
- The customer spending on top-selling artists was analyzed for sales strategies.
- The most popular genre in each country was determined based on the highest purchases.

This project demonstrates how SQL can be leveraged to gain business insights from raw data, aiding in strategic decisions for a music store.

# THANK YOU!

