

MUSIC STORE SQL PROJECT

DATA OVERVIEW

The dataset used in this project is based on a Music Store sales database.

It contains detailed information about customers, employees, invoices, tracks, albums, artists, and genres. The data captures every aspect of the store's operations — from customer purchases to individual song details.

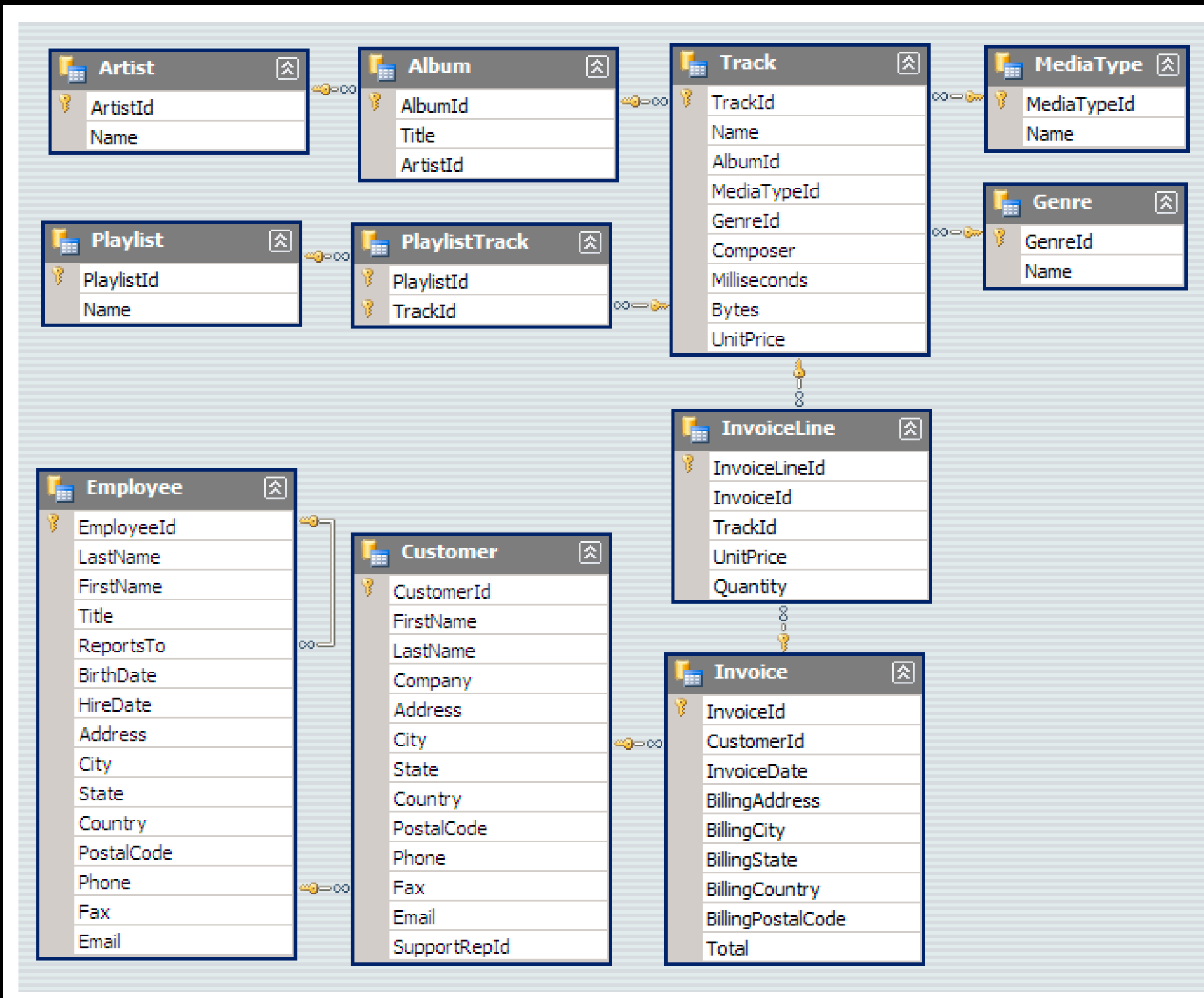
The database is organized across multiple tables such as Customer, Employee, Invoice, InvoiceLine, Track, Album, Artist, and Genre. Each table holds specific information and is connected through key relationships, creating a unified view of the business.

This dataset provides a comprehensive record of music sales transactions, enabling in-depth analysis of:

- Customer purchasing behavior and geographic trends,
- Top-performing tracks, genres, and artists,
- Employee and sales performance, and
- Overall revenue patterns across countries and time periods.

It serves as a strong foundation for data-driven insights into the store's performance, customer preferences, and market opportunities.

DATA SCHEMA



1) WHO IS THE SENIOR MOST EMPLOYEE BASED ON JOB TITLE?

input:

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

output:

	title	full_name
▶	General Manager	AndrewAdams

2) WHICH COUNTRIES HAVE THE MOST INVOICES?

input:

```
select  billing_country, count(invoice_id) as invoice_count from invoice
group by billing_country
order by invoice_count desc;
```

output:

	billing_country	invoice_count
►	USA	131
	Canada	76
	Brazil	61
	France	50
	Germany	41
	Czech Republic	30
	Portugal	29
	United Kingdom	28
	India	21
	Ireland	13
	Chile	13
	Finland	11
	Spain	11

3) WHAT ARE TOP 3 VALUES OF TOTAL INVOICE?

input:

```
select invoice_id , sum(total) as total_amount from invoice
group by invoice_id
order by total_amount desc limit 3;
```

output:

	invoice_id	total_amount
▶	183	23.759999999999998
	92	19.8
	31	19.8

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

input:

```
select billing_city as city, round(sum(total),2) as total_amount from invoice
group by city
order by total_amount desc;
```

output:

	city	total_amount
►	Prague	273.24
	Mountain View	169.29
	London	166.32
	Berlin	158.4
	Paris	151.47
	SÃ£o Paulo	129.69
	Dublin	114.84
	Delhi	111.87

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

input:

```
select customer.customer_id,customer.first_name,customer.last_name,round(sum(invoice.total),2) as amount_spend
from invoice join customer
on invoice.customer_id = customer.customer_id
group by customer.customer_id,customer.first_name,customer.last_name
order by amount_spend desc limit 1;
```

output:

	customer_id	first_name	last_name	amount_spend
►	5	František	Wichterlovský	144.54

6)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

input:

```
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;
```

output:

	email	first_name	last_name
▶	aaronmitchell@yahoo.ca	Aaron	Mitchell
	alero@uol.com.br	Alexandre	Rocha
	astrid.gruber@apple.at	Astrid	Gruber
	bjorn.hansen@yahoo.no	Björn	Hansen
	camille.bernard@yahoo.fr	Camille	Bernard
	daan_peeters@apple.be	Daan	Peeters
	diego.gutierrez@yahoo.ar	Diego	Gutiérrez

7)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

input:

```
SELECT artist.name AS artist_name,COUNT(track.track_id) AS total_rock_tracks
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_rock_tracks DESC
LIMIT 10;
```

output:

	artist_name	total_rock_tracks
►	AC/DC	18
	Aerosmith	15
	Audioslave	14
	Led Zeppelin	14
	Alanis Morissette	13
	Alice In Chains	12
	Frank Zappa & Captain Beefheart	9
	Accept	4

8)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

input:

```
select track.name , milliseconds from track
where milliseconds > (
select avg(milliseconds)
from track )
order by milliseconds desc;
```

output:

	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

9)FIND HOW MUCH AMOUNT SPENT BY EACH CUSTOMER ON ARTISTS? WRITE A QUERY TO RETURN CUSTOMER NAME, ARTIST NAME AND TOTAL SPENT

input:

```
select customer.first_name,customer.last_name,artist.name as artist_name, sum(invoice_line.unit_price * invoice_line.quantity) as total_spend
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 album on track.album_id = album.album_id
join artist on album.artist_id = artist.artist_id
group by customer.first_name,customer.last_name,artist_name
order by total_spend desc;
```

output:

	first_name	last_name	artist_name	total_spend
▶	Steve	Murray	AC/DC	17.82
	Jennifer	Peterson	Aerosmith	14.8500000000000001
	Mark	Taylor	Aerosmith	14.8500000000000001
	Fernanda	Ramos	Antônio Carlos Jobim	13.8600000000000001
	Leonie	Köhler	Audioslave	13.8600000000000001
	Edward	Francis	Alanis Morissette	12.8700000000000001
	Emma	Jones	Alanis Morissette	12.8700000000000001

10)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

input:

```
SELECT billing_country, name AS genre_name, count
FROM (
SELECT billing_country,genre.name,COUNT(genre.name) AS count,
RANK() OVER (PARTITION BY billing_country ORDER BY COUNT(genre.name) DESC) AS rnk
FROM invoice
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
GROUP BY billing_country, genre.name
) ranked
WHERE rnk = 1
ORDER BY billing_country;
```

output:

	billing_country	genre_name	count
▶	Argentina	Rock	1
	Australia	Rock	18
	Austria	Rock	6
	Belgium	Rock	5
	Brazil	Rock	26

11)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

input:

```
SELECT country,first_name,last_name,total_spent
FROM (
    SELECT c.country,c.first_name,c.last_name,SUM(i.total) AS total_spent,
    RANK() OVER (PARTITION BY c.country ORDER BY SUM(i.total) DESC) AS rnk
    FROM customer c JOIN invoice i
    ON c.customer_id = i.customer_id
    GROUP BY c.country, c.first_name, c.last_name
) ranked
WHERE rnk = 1
ORDER BY country;
```

output:

	country	first_name	last_name	total_spent
►	Argentina	Diego	Gutiérrez	39.6
	Australia	Mark	Taylor	81.18
	Austria	Astrid	Gruber	69.3
	Belgium	Daan	Peeters	60.38999999999999
	Brazil	Luís	Gonçalves	108.89999999999998

CONCLUSION

The Music Store dataset offers a rich and comprehensive foundation for exploring real-world business insights through SQL.

By analyzing customer data, sales transactions, and music catalog information, it enables a deep understanding of purchasing behavior, revenue distribution, and musical preferences across regions.

Through this project, key analytical objectives such as identifying top-performing genres, artists, and customers were achieved, along with evaluating employee performance and sales trends.

Overall, the analysis demonstrates how structured SQL queries can transform raw transactional data into actionable insights, supporting data-driven decision-making and showcasing proficiency in database management, querying, and analytical storytelling.

THANKYOU