Contents

[Project Documentation for Digital Music Store Analysis (SQL) 1](#_Toc151721634)

[1.0 Objective 2](#_Toc151721635)

[2.0 Database Schema 2](#_Toc151721636)

[3.0 Import data to Postgres (.sql file to dbms) 2](#_Toc151721637)

[4.0 Data Analysis 3](#_Toc151721638)

[4.0.1 Set 1 – Easy Questions 3](#_Toc151721639)

[4.0.2 Set 2 – Medium difficult questions 5](#_Toc151721640)

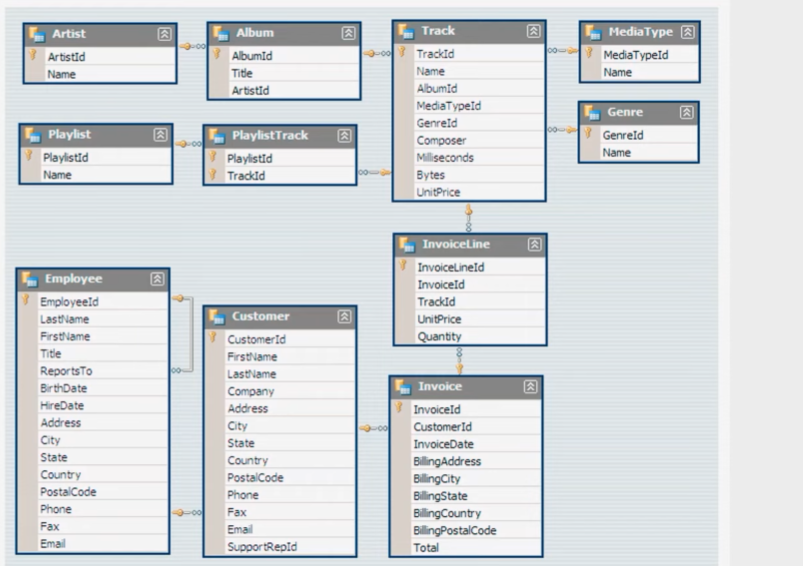
[4.0.3 Advanced set 7](#_Toc151721641)

# Project Documentation for Digital Music Store Analysis (SQL)

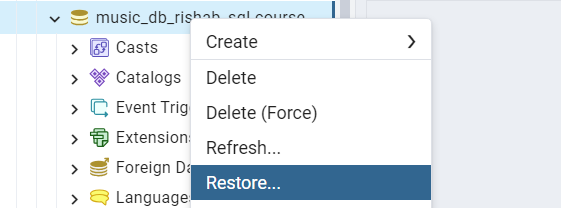
## Objective

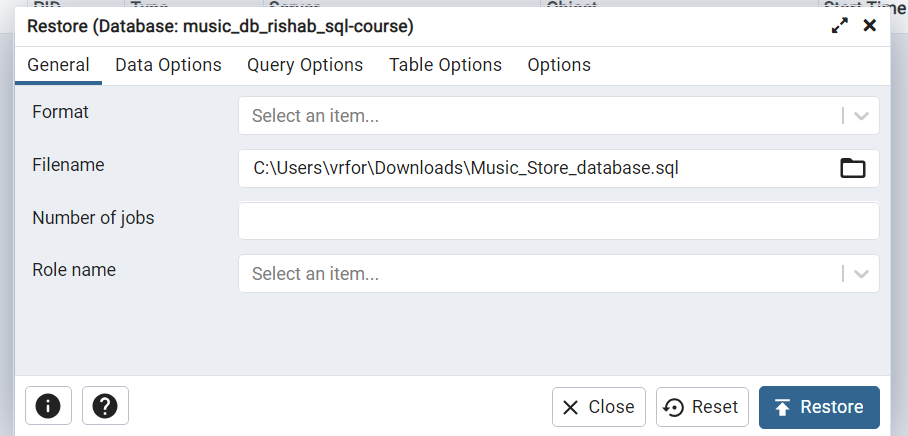
Examine the dataset with SQL and help the store understand its business growth by answering simple questions.

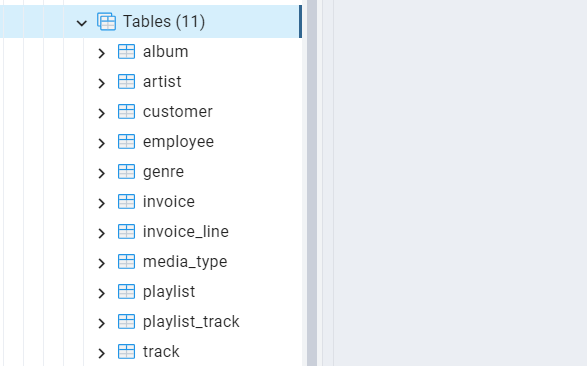
## 2.0 Database Schema



## 3.0 Import data to Postgres (.sql file to dbms)







We now have all tables in database for further analysis.

## 4.0 Data Analysis

### 4.0.1 Set 1 – Easy Questions

**-- Q1 Who is the senior most employee based on job title?**

*select \* from employee*

*ORDER BY levels desc*

*limit 1*

**-- Q2 Which countries have the most Invoices?**

*select billing\_country, count(\*) as total\_invoices*

*from invoice*

*group by billing\_country*

*order by total\_invoices desc*

*limit 1*

**-- Q3 What are the top 3 values of total invoice**

*select total from invoice*

*order by total desc*

*limit 3*

**-- Q4 Which city has the best customers? We would like to throw a prmotional Music Festival in the city we made the most money. Write a query that returns on city that has the highest sum of invoice totals. Return both the city name and sum of all invoice totals.**

*select billing\_city, sum(total) sumz*

*from invoice*

*group by billing\_city*

*order by sumz desc*

**-- Q5 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?**

*select c.customer\_id, c.first\_name, c.last\_name, round(sum(i.total)) as total\_money\_spent from customer c*

*full outer join invoice As i*

*on c.customer\_id = i.customer\_id*

*group by c.customer\_id*

*order by total\_money\_spent desc*

*limit 1*

### 4.0.2 Set 2 – Medium difficult questions

-**- Q1. Write a query to return the email, first\_name, last\_name and Genre of all Rock Music listeners Return your list list ordered aplhabetically by email starting with A**

looking into schema we cant join genre and customer directly.. we need to follow path join multiple table like track, invoiceline, invoice, customer. so ..

*select DISTINCT email, first\_name, last\_name*

*FROM customer*

*JOIN invoice as i ON customer.customer\_id = i.customer\_id*

*JOIN invoice\_line as il ON il.invoice\_id = i.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*

**-- Q2. Let's 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 brands.**

-- this query is not optimized because of lots of joins need to make subqueries

*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;*

**-- Q3. Return all the track names that have a song length longer than the average song length**

**Return the Name and Milllisecons 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;*

### 4.0.3 Advanced set

**-- Qn1. Find how much amount spend by each cusomer on artists? Write a query to return customer name,artist name and total spent**

*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 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 1,2,3,4*

*ORDER BY 5 DESC;*

**-- Qn2. 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 top Genre. For countries where the maximum number of purchases is shared return all Genres.**

/\* Using CTE \*/

*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*

**--Q3: 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.**

*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*