

SQL Project

Music Store Data Analysis

Introduction

The Music Store Data Analysis project provides an opportunity to apply SQL concepts to a real-world dataset. By working with a music store database, you will practice creating tables, defining relationships, importing data, and writing complex queries to answer business questions. This project reinforces skills in database design, data manipulation, and analytical query writing.

1 Learning Objectives

- Design and implement a relational database schema with appropriate constraints.
- Import and validate data from CSV files.
- Write SQL queries to extract and analyze data.
- Apply advanced SQL techniques such as joins, aggregations, and subqueries.

Project Setup

To begin, you will create a database for the music store and set up all necessary tables based on the provided schema diagram. Follow these steps to ensure a successful setup:

1.1 Database Creation

- Create a new database named `music_store`.
- Use a SQL database management system.

2 Table Creation

```
-- 1. Genre and MediaType
CREATE TABLE Genre (
    genre_id INT PRIMARY KEY,
    name VARCHAR(120)
);

CREATE TABLE MediaType (
    media_type_id INT PRIMARY KEY,
    name VARCHAR(120)
);

-- 2. Employee
```

```

CREATE TABLE Employee (
    employee_id INT PRIMARY KEY,
    last_name VARCHAR(120),
    first_name VARCHAR(120),
    title VARCHAR(120),
    reports_to INT,
    levels VARCHAR(255),
    birthdate DATE,
    hire_date DATE,
    address VARCHAR(255),
    city VARCHAR(100),
    state VARCHAR(100),
    country VARCHAR(100),
    postal_code VARCHAR(20),
    phone VARCHAR(50),
    fax VARCHAR(50),
    email VARCHAR(100)
);

-- 3. Customer
CREATE TABLE Customer (
    customer_id INT PRIMARY KEY,
    first_name VARCHAR(120),
    last_name VARCHAR(120),
    company VARCHAR(120),
    address VARCHAR(255),
    city VARCHAR(100),
    state VARCHAR(100),
    country VARCHAR(100),
    postal_code VARCHAR(20),
    phone VARCHAR(50),
    fax VARCHAR(50),
    email VARCHAR(100),
    support_rep_id INT,
    FOREIGN KEY (support_rep_id) REFERENCES Employee(employee_id)
);

-- 4. Artist
CREATE TABLE Artist (
    artist_id INT PRIMARY KEY,
    name VARCHAR(120)
);

-- 5. Album
CREATE TABLE Album (
    album_id INT PRIMARY KEY,
    title VARCHAR(160),
    artist_id INT,
    FOREIGN KEY (artist_id) REFERENCES Artist(artist_id)
);

-- 6. Track
CREATE TABLE Track (

```

```

        track_id INT PRIMARY KEY,
        name VARCHAR(200),
        album_id INT,
        media_type_id INT,
        genre_id INT,
        composer VARCHAR(220),
        milliseconds INT,
        bytes INT,
        unit_price DECIMAL(10,2),
        FOREIGN KEY (album_id) REFERENCES Album(album_id),
        FOREIGN KEY (media_type_id) REFERENCES MediaType(media_type_id),
        FOREIGN KEY (genre_id) REFERENCES Genre(genre_id)
    );

-- 7. Invoice
CREATE TABLE Invoice (
    invoice_id INT PRIMARY KEY,
    customer_id INT,
    invoice_date DATE,
    billing_address VARCHAR(255),
    billing_city VARCHAR(100),
    billing_state VARCHAR(100),
    billing_country VARCHAR(100),
    billing_postal_code VARCHAR(20),
    total DECIMAL(10,2),
    FOREIGN KEY (customer_id) REFERENCES Customer(customer_id)
);

-- 8. InvoiceLine
CREATE TABLE InvoiceLine (
    invoice_line_id INT PRIMARY KEY,
    invoice_id INT,
    track_id INT,
    unit_price DECIMAL(10,2),
    quantity INT,
    FOREIGN KEY (invoice_id) REFERENCES Invoice(invoice_id),
    FOREIGN KEY (track_id) REFERENCES Track(track_id)
);

-- 9. Playlist
CREATE TABLE Playlist (
    playlist_id INT PRIMARY KEY,
    name VARCHAR(255)
);

-- 10. PlaylistTrack
CREATE TABLE PlaylistTrack (
    playlist_id INT,
    track_id INT,
    PRIMARY KEY (playlist_id, track_id),
    FOREIGN KEY (playlist_id) REFERENCES Playlist(playlist_id),
    FOREIGN KEY (track_id) REFERENCES Track(track_id)
);

```

3 Data Import

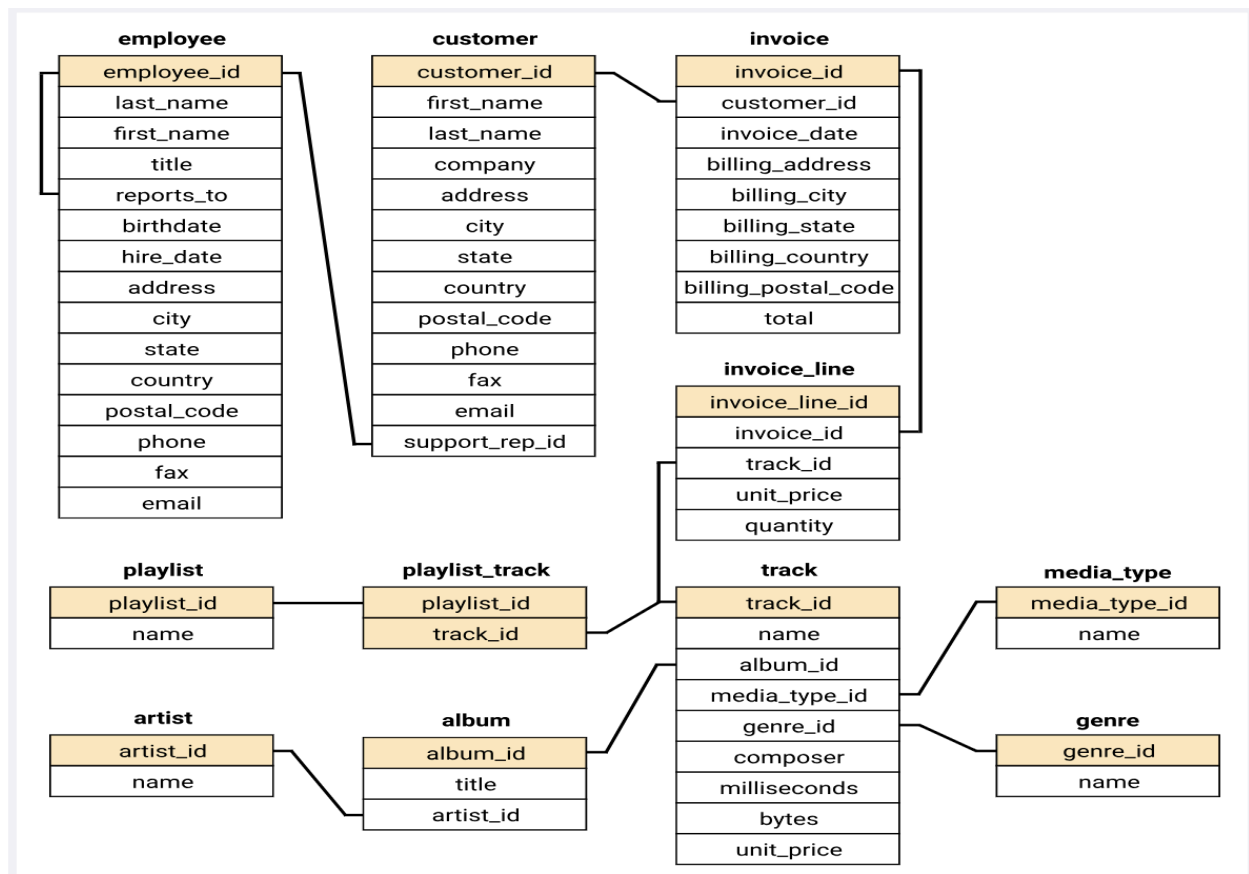
- Download the dataset from the provided link [Click Here](#)
- Challenges Note: Importing track.csv (Track Table) [Click Here](#)

Please be patient while importing the data especially the track.csv, invoiceline.csv and playlisttrack.csv file —as it contains a large volume of records and may take some time to load completely.

Schema Diagram

The schema diagram, included in the project folder, illustrates the relationships between tables such as customers, invoices, tracks, genres, artists, and others. Use this diagram to:

- Identify primary and foreign keys.
- Understand table relationships (e.g., one-to-many between customers and invoices).
- Ensure all constraints are correctly implemented.



Note: If you cannot access the schema diagram, consult your instructor for clarification.

Key things to consider while creating a database:

- Add auto increment constraint in all the table's primary keys.
- Make sure you are using cascade and while creating foreign keys in all the tables.
- While importing the data carefully check whether the column names are matching to csv file column names or not.
- If the data is too large, follow the steps in the document [Click Here](#)

Challenges You Might Face

- Understanding table relationships and applying correct joins.
- Ensuring data consistency with foreign key constraints.
- Handling aggregation across joined tables.
- Extracting time-based trends from date data (especially if in VARCHAR format)

Task Questions

Below are the tasks you will complete using SQL queries. Each question requires you to analyze the music store database and provide accurate results. Write your queries in a clear, efficient manner, and include comments to explain your logic.

1. Who is the senior most employee based on job title?
2. Which countries have the most Invoices?
3. What are the top 3 values of total invoice?
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
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
6. Write a 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
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
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
9. Find how much amount is spent by each customer on artists? Write a query to return customer name, artist name and total spent
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
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

Project Presentation Template

As part of this project, you are required to create and present the analysis findings. Use the following PowerPoint template to structure your presentation:

👉 Click here to [find the PPT Template for the Project Presentation](#)

Submission

After completion of the project, zip the **.sql query file** and **PPT**, upload the zip file with your name and batch number In LMS.