SQL Movie Project

Creating a Movie Database with SQL

Presented by

- Sameena Mujawar
- Laylaa Tariq
- Sai Srivatsa Thangallapelly

Agenda

- Introduction
- Database Structure
- ER Diagram
- Queries
- Conclusion

Introduction

Introduction

In today's presentation, we will be discussing the creation of a movie database using SQL. The database will store information about movies such as their titles, years of release, durations, genres, and production companies. Additionally, the database will include data on the people involved in making these movies, including actors, directors, and producers.

To create this database, we will be using the SQL programming language, which allows us to manage large amounts of data efficiently. We will walk through the code step by step and explain the purpose and functionality of each command. We will also introduce several views that allow us to extract useful information from the database.

Database Structure

Database Structure

In this Project we used the database from the below source:
https://www.kaggle.com/datasets/blitzapurv/rsvp-case-study?resource=download

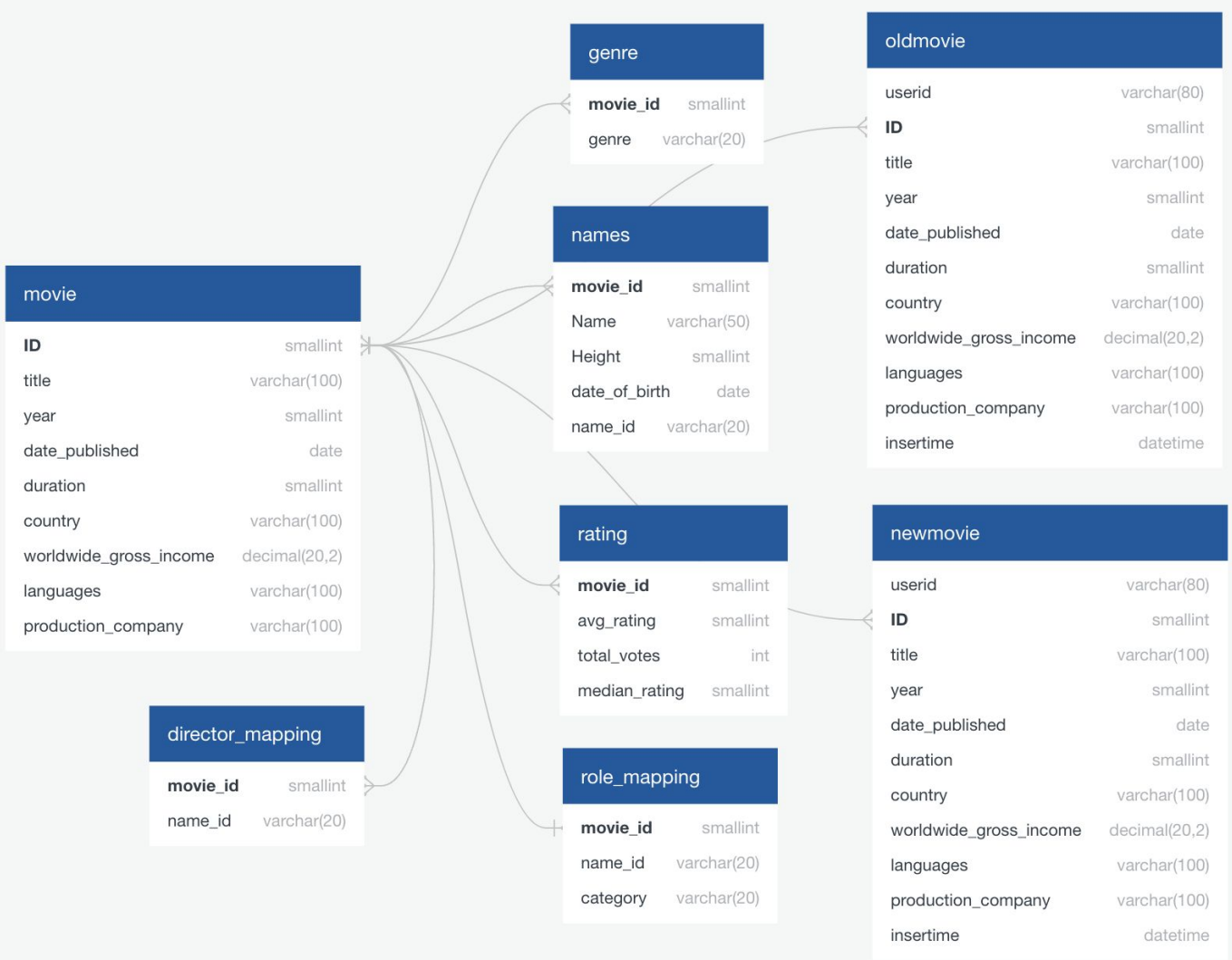
This Database consists of six tables, **Movie, Genre, Names, Directory_Mapping, Role_Mapping, Rating** with minimum of 3 columns and 5 rows.

All the tables are related to each other using foreign keys, which ensure data integrity and consistency. The movie table is the main table, and all other tables are related to it.

In addition to the six tables, the database also includes 2 triggers that keeps a record of newly inserted movies in the **newmovie** table, and old movies in the **oldmovie** table along with the user who inserted them and the time of insertion.

ER Diagram

SCHEMA

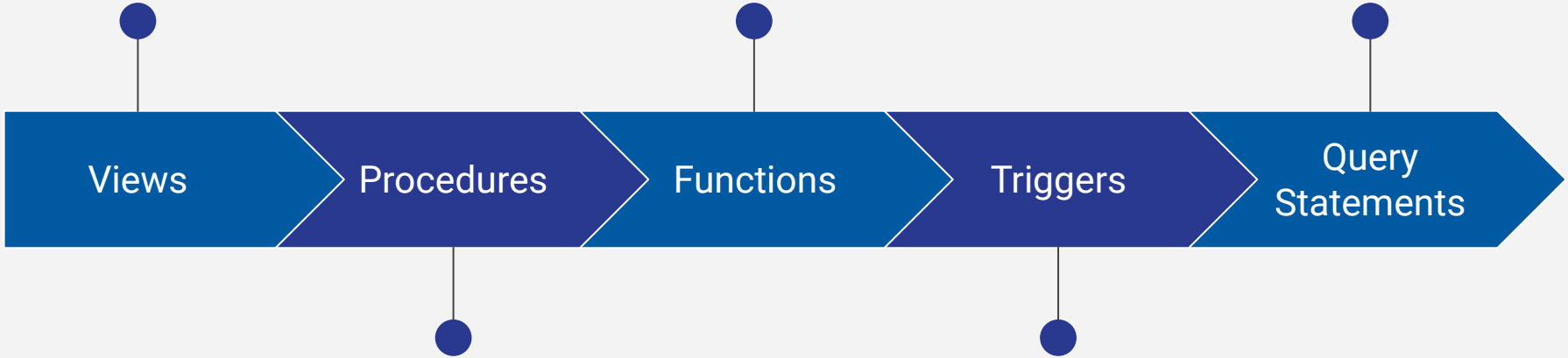

genre
movie_id smallint
genre varchar(20)
oldmovie
userid varchar(80)
ID smallint
title varchar(100)
year smallint
date_published date
duration smallint
country varchar(100)
worldwide_gross_income decimal(20,2)
languages varchar(100)
production_company varchar(100)
insertime datetime
names
movie_id smallint
Name varchar(50)
Height smallint
date_of_birth date
name_id varchar(20)
movie
ID smallint
title varchar(100)
year smallint
date_published date
duration smallint
country varchar(100)
worldwide_gross_income decimal(20,2)
languages varchar(100)
production_company varchar(100)
rating
movie_id smallint
avg_rating smallint
total_votes int
median_rating smallint
newmovie
userid varchar(80)
ID smallint
title varchar(100)
year smallint
date_published date
duration smallint
country varchar(100)
worldwide_gross_income decimal(20,2)
languages varchar(100)
production_company varchar(100)
insertime datetime
director_mapping
movie_id smallint
name_id varchar(20)
role_mapping
movie_id smallint
name_id varchar(20)
category varchar(20)

Queries

Views are virtual tables that are created by querying one or more tables.

Functions are reusable code blocks that perform a specific task.

Query statements are used to retrieve and manipulate data in a database.



Procedures are similar to functions but can perform multiple tasks and do not return values.

Triggers are special types of stored procedures that are automatically executed in response to certain database events.

Conclusion

Conclusion

In this presentation, we have shown the code for creating and populating several tables in a MySQL database named "dbmovies". We have created tables to store information about movies, genres, names of actors and directors, and ratings of movies.

Furthermore, we have demonstrated the use of views to query the database. Specifically, we have created a view called "movieview" to find the total number of movies done by each actor, and used it to identify the actor who has done the highest number of movies.

Overall, this code provides a framework for building a database of movies and associated information, and demonstrates the use of views to extract meaningful insights from the data.

Any Questions

Adios!