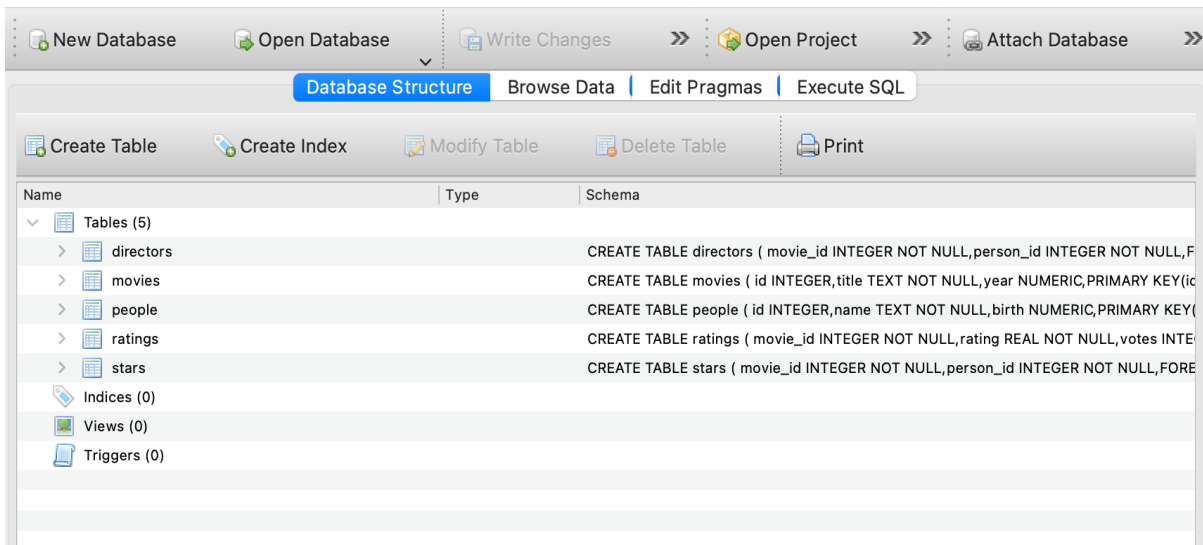# </talentlabs>

Lab Assignment 7 - Table Relationships

Instructions

1. Answer the below question in the boxes.
2. Please submit the assignment through TalentLabs Learning System.

Open the Movies database

Follow the step illustrated in Chapter 3 to open the Movies database using DB Browser for SQLite. You should see 5 tables in the database.

Name	Type	Schema
Tables (5)		
directors		CREATE TABLE directors (movie_id INTEGER NOT NULL,person_id INTEGER NOT NULL,F
movies		CREATE TABLE movies (id INTEGER,title TEXT NOT NULL,year NUMERIC,PRIMARY KEY(id
people		CREATE TABLE people (id INTEGER,name TEXT NOT NULL,birth NUMERIC,PRIMARY KEY(
ratings		CREATE TABLE ratings (movie_id INTEGER NOT NULL,rating REAL NOT NULL,votes INTE
stars		CREATE TABLE stars (movie_id INTEGER NOT NULL,person_id INTEGER NOT NULL,FORE
Indices (0)		
Views (0)		
Triggers (0)		

New Database   Open Database   Write Changes   Open Project   Attach Database

Database Structure | Browse Data | Edit Pragmas | Execute SQL

Create Table   Create Index   Modify Table   Delete Table   Print

Database Study Exercises

Please try to answer the following questions by studying the movie databases. To be specific, please open up each of the tables and try to understand the meaning of each table and column.

If you need some more references or information, you can also read the following description:

> There are 5 tables in the database, including movies, ratings, directors, stars, and people.
>
> The core of the database is "movies", which stores all the movie entries. There is an id column in the movies table which serves as the unique identifier of the movies.
>
> The ratings table stores the average ratings of each movies, and could be linked back to the

movies table with the movie_id column.

Another important table is the people table. It stores the details of the people(actors and directors) who are related to at least one of the movies. Each person in the table got assigned with a unique key.

However, there is no information about who is acting or directing which movies from the people table. The relationships between "acting" and "directing" are stored in the "directors" and "stars" table.

Questions

1. What is the relationship between movies and ratings? Is it a one-to-one relationship, one-to-many relationship or many-to-many relationship?

    - One-to-One relationship

2. What is the relationship between movies - directors - people? What is the function of having the directors table?

    - Many-to-Many relationship
    - Directors table have a movie_id column that links to the movies table's id column and a person_id column that can link to the people table's id column

3. What is the primary key in the movies table?

    - id

4. What is the primary key in the people table?

    - id

5. What is the foreign key in the ratings table? What is the purpose of it?

    - movie_id. To establish a relationship between the movies table and ratings table such as identifying which movie the ratings are for by linking from the ratings table's movie_id column to the movies table's id column

6. List <mark>another many-to-many relationship</mark> in the movies database that was not mentioned above yet. What are the tables involved?

> - The tables involved in this relationship are:
> - **movies table**
> - **people table**
> - **stars table**: This is the junction table that links the **movies** and **people** tables, indicating which actors (people) are in which movies.
>
> The **stars** table connects **movies** and **people**, allowing a single movie to have multiple actors and a single actor to appear in multiple movies, thus establishing a many-to-many relationship.

- End of Assignment -