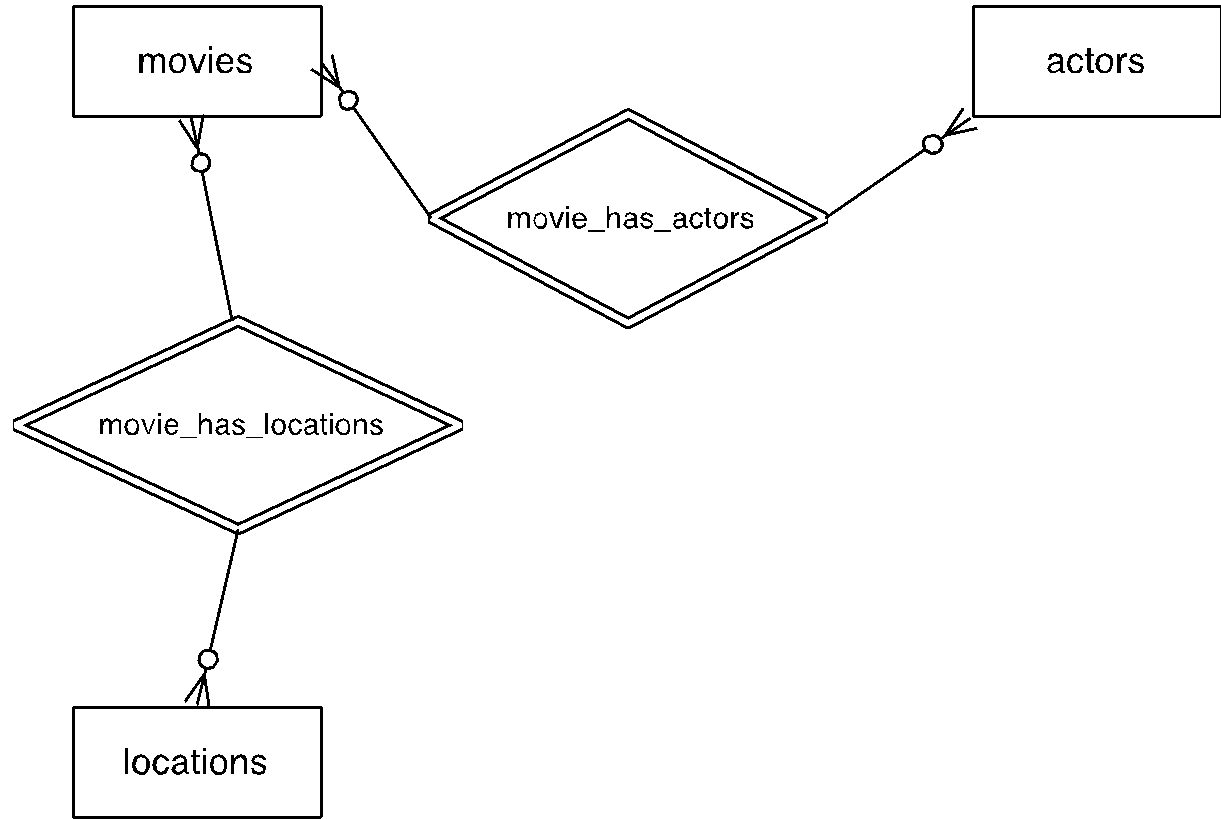
# INFSCI 1022 – Database Systems – Summer 2016

## Homework 2 – Introduction to SQL

1. Write SQL queries for each of the tasks using MySQL workbench.
2. You cannot use Workbench’s GUI to create database schemas and tables – you must write your own SQL statements.
3. Follow the underscore convention when naming your database schema, entities and attributes.
4. Save your work as an SQL script – name your script **yourPittID\_assignment2.sql**.
5. Submit your work via CourseWeb.

Consider the relational database schema diagram below. It consists of 3 entities – movies, actors, and locations.



Because movies and actors have a many-to-many relationship and movies and locations have a many-to-many relationship, your schema would also have 2 junction tables – movies\_actors and movies\_locations. Each table’s logical structure is described below:

**movies**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Primary Key** | **Data type** | **Is null?** | **Is auto-increment?** |
| movie\_id | yes | INT | no | yes |
| title | no | VARCHAR(200) | no | no |
| release\_date | no | DATETIME | no | no |
| plot\_description | no | VARCHAR(4000) | no | no |

**actors**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Primary Key** | **Data type** | **Is null?** | **Is auto-increment?** |
| actor\_id | yes | INT | no | yes |
| first\_name | no | VARCHAR(100) | no | no |
| last\_name | no | VARCHAR(100) | no | no |
| birth\_date | no | DATETIME | no | no |
| biography | no | VARCHAR(1000) | no | no |

**locations**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Primary Key** | **Data type** | **Is null?** | **Is auto-increment?** |
| location\_id | yes | INT | no | yes |
| location\_name | no | VARCHAR(100) | no | no |
| street\_address | no | VARCHAR(150) | no | no |
| city | no | VARCHAR(100) | no | no |
| state | no | CHAR(2) | no | no |
| zip | no | VARCHAR(5) | no | no |

**movies\_actors**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Primary Key** | **Data type** | **Is null?** | **Is auto-increment?** |
| movie\_id | no | INT | no | no |
| actor\_id | no | INT | no | no |

**movies\_locations**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Primary Key** | **Data type** | **Is null?** | **Is auto-increment?** |
| movie\_id | no | INT | no | no |
| location\_id | no | INT | no | no |

**Task 1 (5 points)**: Create a database called **movie\_tracker** using CREATE DATABASE statement.

**Task 2 (25 points):** In database **movie\_tracker**, create the following entity tables:

1. movies
2. actors
3. locations

Each tables logical structure should correspond to the descriptions provided in this assignment. Use CREATE TABLE statement.

**Task 3 (15 points)**: In database movie\_tracker, create the following junction tables:

1. movies\_actors
2. movies\_locations

Use CREATE TABLE statement to create junction tables. Make sure to create appropriate foreign keys – each table will have two foreign keys. Use ALTER TABLE statement to create foreign keys.

**Task 4 (20 points):** For each entity table, insert at least 3 rows using INSERT statement:

1. At least 3 movies in the **movies** table
2. At least 3 actors in the **actors** table
3. At least 3 locations in the **locations** table

You can make up your own data for the INSERT statements.

**Task 5 (10 points):** For each junction table, create at least 2 relationships (insert at least two rows of appropriate IDs).

**Task 6 (5 points):** Write a SELECT statement to display top 2 actors sorted by actor’s last name.

**Task 7 (5 points):** Write a SELECT statement to display location name, street address and city sorted by location name is descending order.

**Task 8 (5 points):** Write a SELECT statement to display movies released between two dates of your choice.

**Task 9 (5 points):** Write an UPDATE statement to update zip code for all locations to 15217

**Task 10 (5 points):** DELETE one row of your choice from **actors** table. Be careful – make sure to write a correct WHERE clause.

**Due on Wednesday, June 8 by midnight**