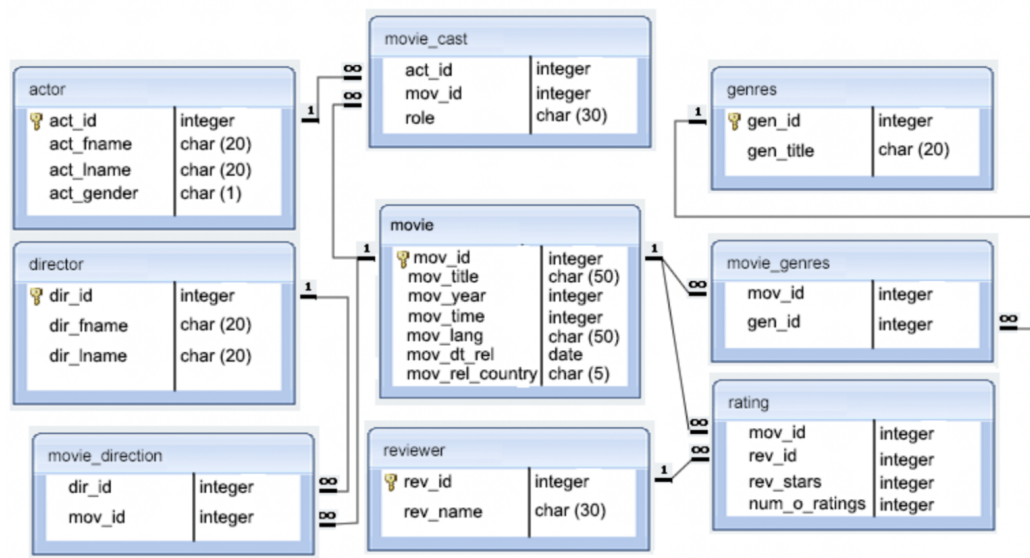


# Databases and information systems laboratory

## CS313

IIT Dharwad

Handout 6  
27 – 09 – 2023



Open two tabs in the terminal and access `moviedb` from both the terminals. These act as the two clients who are concurrently accessing the database.

1. Begin transaction in terminal 1 using `begin;` command
2. Insert a row into actor table from terminal 1
3. Execute `select *` on actor table from terminal 2
4. Commit the transaction in terminal 1 using `commit;` command

5. Now again execute `select *` on actor table from terminal 2
6. Repeat all the previous questions, except the 4<sup>th</sup> step where execute `rollback;` command
7. Begin a new transaction in terminal 1 and write a query that has syntax error. Commit the transaction and see what happens

For each of the question below, write a python code that connects to the `moviedb` and does the following:

1. Take the `act_id`, first name, last name and gender as input from the console, check if the id is already present in the database. If yes then then display appropriate message; if not then insert the row into the actor table. <sup>1</sup>. To execute commit / rollback you can do it in two ways (one by executing the corresponding command via the cursor object, or by calling the corresponding function from the `psycopg2` object itself)

Sample 1:

```
Enter actor id : 101
Enter First name : Rajkumar
Enter Last name : Dr.
Enter Gender : M
```

Output: *Actor ID already exists*

Sample 2:

```
Enter actor id : 131
Enter First name : Rajkumar
Enter Last name : Dr.
Enter Gender : M
```

Output: *Actor details inserted into the actor table successfully*

2. Use the following command on `psql` console to declare (`mov_id`, `act_id`) as the primary key for the `movie_cast` table:

```
ALTER TABLE movie_cast ADD PRIMARY KEY (mov_id,);
```

---

<sup>1</sup>To know how to pass arguments to SQL queries, refer <https://www.psycopg.org/psycopg3/docs/basic/params.html>

3. For the newly added actor in the earlier question, take as input the movie ids and roles that (s)he has acted in. Begin a transaction and start adding each movie information to the movie\_cast table. If at some point you encounter a movie id that is not present in the movie table, rollback the transaction. If the entire list has been inserted then commit the transaction. Display appropriate messages in each case.

Sample 1:

```
Enter the number of movies: 3
Enter movie id of movie number 1 : 923
Enter role of the actor in movie number 1: Babruvahana
Enter movie id of movie number 2 : 999
Enter role of the actor in movie number 2: Kalidasa
Enter movie id of movie number 3 : 921
Enter role of the actor in movie number 3: Mayura Varma
```

Output : *Movie number 2 is not present in the database. Database is not updated*

Sample 2:

```
Enter the number of movies: 3
Enter movie id of movie number 1 : 923
Enter role of the actor in movie number 1: Babruvahana
Enter movie id of movie number 2 : 999
Enter role of the actor in movie number 2: Kalidasa
Enter movie id of movie number 3 : 921
Enter role of the actor in movie number 3: Mayura Varma
```

Output : *Database update successfull*

4. Execute the previous query from two different terminals and alternatively give inputs such that the values interact. See how the database behaves for various cases (your submission for this question should have all input cases that you have tried and the corresponding observations on the output. Full points will be given only if all possible test cases are covered)