**Lab 02 - Relational Algebra**

**(Selection, Projection, Join)**

Consider the following relations:

**MOVIES** (id:int, title:varchar(35), year:int, director:int) Id is the key

**ACTORS** (id:int, name:varchar(20), lastname:varchar(30)) Id is the key.

**CASTINGS** (movieid:int, actorid:int) (movieid and actorid) is the key

**DIRECTORS**(id:int, name:varchar(20), lastname:varchar(30)) id is the key

1. What is the result of the following queries?

Display **DIRECTORS** relation with 3 attributes: id, name and lastname and display tuple match with id component is 100

* 1. **Πtitle, year(**)

Display **MOVIES** relation with 2 attributes: title and year and display tuple match with director component is 100

* 1. **Πtitle, year,name,lastname(**

Combine tuple of **MOVIES** relation and **DIRECTORS** relation when director component in **MOVIES** relation are equal to id component in **DIRECTORS** relation and display match result into one relation with 4 attributes: title, year, name and lastname

1. Using the same schema as above, write each of the following queries as a relational algebra expression:
2. List all actors.

**Πname,lastname**

1. List the name and the year of all movies.

**Πtitle, year**

1. Find all movies produced in 2010.
2. List all actors in the Avatar movie.

**Πname,lastname( )**

1. Find the movie title, year, and the director name for movies produced in 2019.

**Πtitle,year,name ()**

1. Find movie title and the movie director’s name and last name for all movies that the actor with ID = 200 plays a character in them.

**Πtitle,name,lastname()**

1. Find all actors played in movies produced after 2010 and before 2018.

**Πname,lastname ()**