**Lab 02 - Relational Algebra**

**(Selection, Projection, Join)**

Sungsoo Cho

121182190

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?

Select the director who has ID 100

* 1. **Πtitle, year(**)

Select title and year of director who has ID 100

* 1. **Πtitle, year,name,lastname(**

Select title and year and lastname of directors who are included movies and directors relations

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

**Πid,name,lastname (**)

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

**Πtitle,year (MOVIE)**

1. Find all movies produced in 2010.

1. List all actors in the Avatar movie. (Assume that Avatar *movieid* is 223344.)

**Πid,name,lastnameactors.id=castings.actorid(castings.movieid=movies.id(**)))

1. Find the director name for movies produced in 2019.

**Πname** ( **directors.id=movies.director**(

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 (DIRECTORSdirectors.id=movies.director MOVIESmovies.id=castings.movieid (Πmovieid** ( ))

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

**Π id,name,lastname (ACTORS actor.id=castings.actorid CASTINGScastings.movieid=movies.id** (**Πid** ()))