**Complete-ish overview of my SQL queries/commands. (Some commands weren’t included since they were just SELECT \* FROM tablename)**

DROP TABLE Movie;

DROP TABLE Actor;

I did this because I wanted to start fresh for the coursework.

CREATE TABLE Actor( actID INT NOT NULL,

actName VARCHAR(255) NOT NULL,

constraint pk\_actor PRIMARY KEY (actID),

constraint ck\_actor UNiQUE (actName));

CREATE TABLE Movie(mvID INT NOT NULL,

mvTitle VARCHAR(255) NOT NULL,

actID INT NOT NULL,

mvPrice INT NOT NULL,

mvYear INT NOT NULL,

mvGenre VARCHAR(255),

CONSTRAINT pk\_mv PRIMARY KEY (mvID),

CONSTRAINT fk\_mv\_act FOREIGN KEY (actID) REFERENCES Actor (actID)

);

# I made the tables with the relevant parameters needed. The foreign key chosen is actID since this is the most unique identifer for the actors in the Actor table. All of the attributes of the tables are going to be NOT NULL since I

INSERT INTO Actor(actID,actName)

VALUES (0,"Wax Milson");

SELECT \* FROM Actor; # I used this to see that the table was being populated and that the data was valid

INSERT INTO Actor(actID,actName)

VALUES (1,"Beven Stagley");

INSERT INTO Actor(actID,actName)

VALUES (2,"Pichael Mound");

SELECT \* FROM Actor; # checking on data

INSERT INTO Actor(actID,actName) # Funny dummy data so I can check that removing data from a table is possible

VALUES(33,"MichealSoft Binbows")

DELETE FROM Actor WHERE actID = 33;

SELECT \* FROM Actor; # Making sure the data was removed

DESCRIBE Movie; # Used so I can remember what the Movie table requires

INSERT INTO Movie(mvID,mvTitle,actID,mvPrice,mvYear,mvGenre)

VALUES(0,"G.I. GoLang",0,1000,1992,"Action");

SELECT \* FROM Movie, Actor WHERE Actor.actID = 0; This shows the current movie that Wax Milson is in

INSERT INTO Movie(mvID,mvTitle,actID,mvPrice,mvYear,mvGenre)

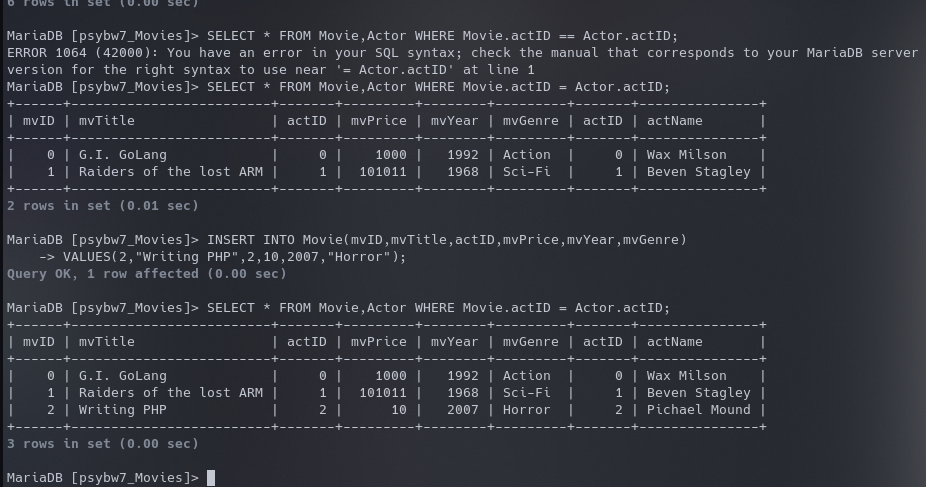
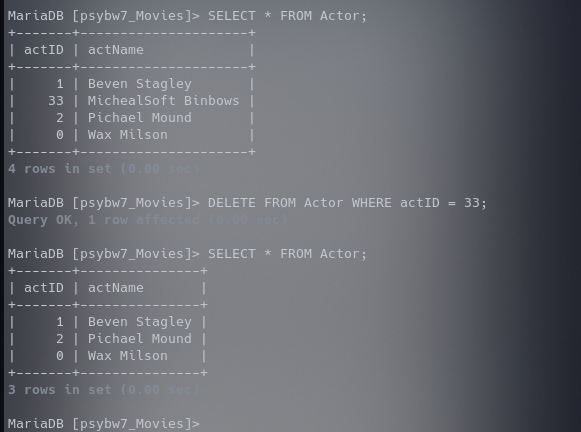
VALUES(1,"Raiders of the lost ARM",1,101011,1968,"Sci-Fi");

SELECT \* FROM Movie,Actor WHERE Movie.actID = Actor.actID; # Show the movies with the corresponding actor

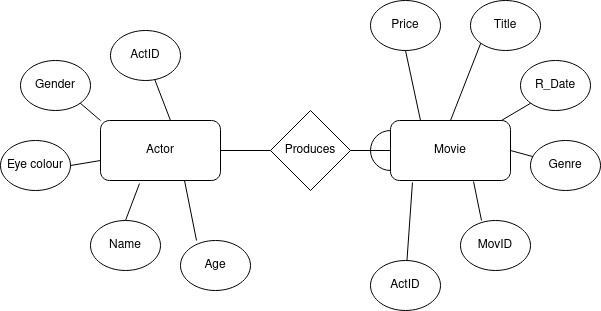
INSERT INTO Movie(mvID,mvTitle,actID,mvPrice,mvYear,mvGenre)

VALUES(2,"Writing PHP",2,10,2007,"Horror");

**Screenshots of some of the commands in practice**



**ER Diagram that I based my tables off of**

As you can see, some of the names are different but the data still remains just under differing aliases. However some attributes are missing from actor and one from the movie. This is simply because its data that I didn’t find to necessary to the actual database. They could produce some semi-unique identifiers for primary keys but in the end I chose not to include them