Ramin Mousivand\_CMPE226\_Lab1

# Project Description

In this project a database system similar to Netflix is designed. Users can create accounts, pay their monthly bills and enjoy watching movies.

# How to use

Customers open an account by giving their info, and then can sign into their accounts. They can build up to 5 accounts. Account 1 is always for the customer himself or herself. Each account has its own number and encrypted password. Accounts will have a watch history and only one playlist. Movies can be searched through stored procedures for enhanced security by title or actors, or look for movies ratings more than a given value. Also each account can generate multiple report in case of a problem.

# Tables

Table Queries and ER diagrams are accessible on the github link. All the tables are normalized up to 3NF.  
There is no multi-valued properties (1NF), No partial dependency (2NF) and no Transitional dependency (3NF). All the datatypes, constraints, triggers, indexes and many other details can be found in the file named. “ Entire Code\_Forward Engineering.docx”. Here just very briefly the structure of the tables are discussed.

Tables include: User, Account, Bill, Watches, Adds\_to\_Playlist, Report and last but not least, Movie.

User: First Name, Last Name, Username, and credit card

PK: Username

First\_Name Not NULL

Last\_Name Not NULL

Credit Card Not NULL, Unique

Account (weak entity): Account Number, Username, Password

Account Number is partial primary key  
primary key = (Account Number, Username)

Foreign Key (Username) refers to User table (Username)

It is On Delete and Update Cascade

Constraint -> Check (Account\_Number<=5)   
Passwrod Not NULL, there must be a password

Bill: ID, start date, monthly\_fee, Username (who pays)

This bill is monthly, so it does not have end\_date to avoid transitional dependency.

Primary Key (ID)

Foreign Key (Username) refers to Account table (Username)

Monthly\_Fee Not NULL

Start\_Date Not NULL

This table is to keep track of watch history

Watches: Date, Account\_Number, Username, Movie\_ID

Primary Key (Date, Account\_Number, Username, Movie\_ID)

Foreign Key (Username) refers to Account table (Username)

Foreign Key (Account\_Number) refers to Account table (Account\_Number)

Foreign Key (Movie\_Id) refers to Movie table (ID)

This table is the playlist

Adds\_to\_Playlist: Account\_Number, Username, Movie\_ID

Primary Key ( Account\_Number, Username, Movie\_ID)

Foreign Key (Username) refers to Account table (Username)

Foreign Key (Account\_Number) refers to Account table (Account\_Number)

Foreign Key (Movie\_Id) refers to Movie table (ID)

Report table is for reports in case of a problem, it can be sent by any account. Accounts are not required to generate reports and it is optional.

Report: Report\_ID, Username, Description, Account\_Number

Primary Key ( Report\_ID)

Foreign Key (Username) refers to Account table (Username)

Description Not NULL, there must be a description

Table Movie holds movie information, Table Deleted Movie has this same structure, used for logging.

Primary Key (ID)

Title Not NULL,

Check Rating<=10

Table Deleted\_Credit\_Card is used for logging and keeping track of credit cards used for the system in the past

Primary Key (Credit\_Card)

First\_Name Not NULL

Last\_Name Not NULL

No foreign key

# Stored Procedures

5 Stored Procedures are implemented. They are explained with codes and screenshots

# Triggers and Logging

5 triggers are created. They can be found with great detail in a file named trigger.

The first 2 Triggers are used for logging. They move deleted movies into deleted\_movie table.

Also they move credit card history of deleted users or updated credit cards.

# SQL Codes

11 SQL codes are provided in a file with the same name. They include select, alter, adding constrants, transactions, creating view, creating index, copying table structure and many other things.

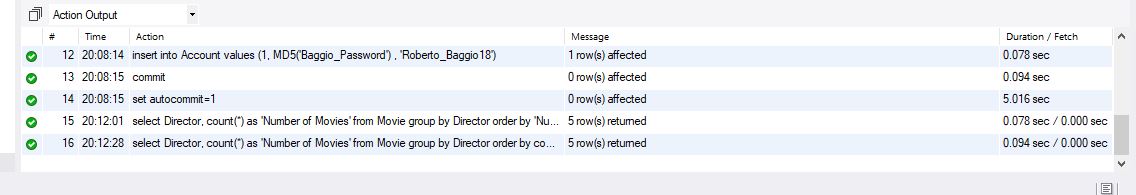
# DB Connection

The DB was connected to python though mysql.connector and performed a select query that lists movies played by Brad Pitt. Python code and Connection screenshot are included.

# Performance Measurement

Regarding Performance, screen shot of runtime for different queries is presented bellow.

The average time is around 1.07 s.



At the end I’d like to thank our TA for helping a lot with this assignment.