CPS510 - A6

By: Isaac Martin, Bowie Chau, Hachi Ndu

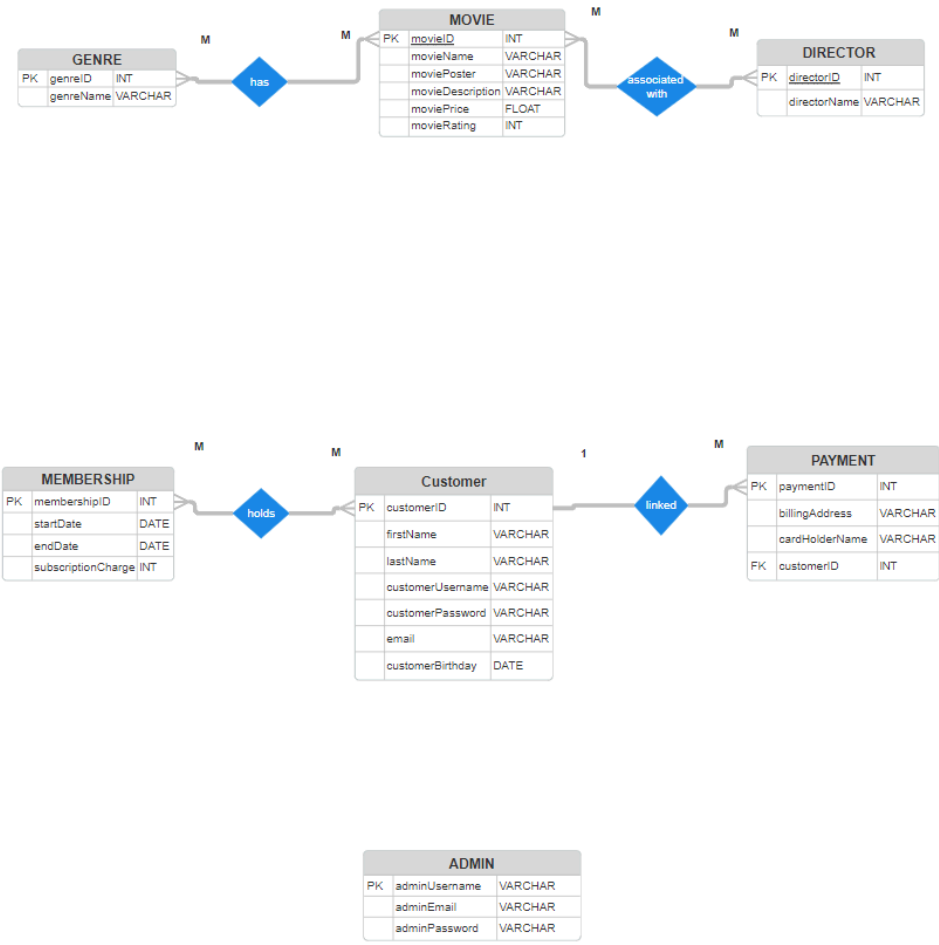
Instructor: Dr. S. B. Tajali

TA: George Lopez

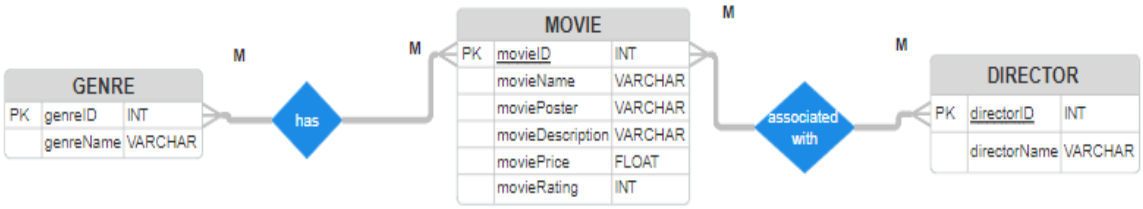
Section #: 06

Lab Number: 6

Oct 30th, 2024



After the completion of our design for our DBMS system, we arrived at the step of normalizing our design. In order to normalize our design, we first outline the functional dependencies in the system. For example, our many-to-many relationship from movie to genre and movie to director.



movieID uniquely identifies each movie and determines all attributes. Each movie can be associated with multiple directors and genres. There the functional dependencies (FD) that would express this relationship would be:

movieID → movieName, moviePoster, movieDescription, moviePrice, movieRating.

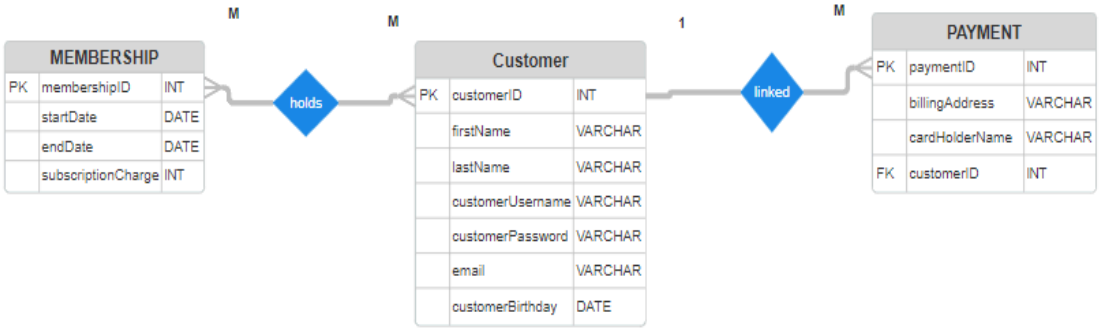
The director table FD is:

directorID → directorName

and is a Many-to-Many relationship with Movie because the directorID uniquely identifies each director and determines directorName. Each director can be linked to multiple movies. The Genres table FD is:

genreID → genreName

and is a Many-to-Many relationship with Movie the genreID uniquely identifies each genre, determining genreName. Each genre can be associated with multiple movies.



The customer's table FD is:

customerID → firstName, lastName, customerUsername, customerPassword, email, customerBirthday

customerUsername → customerID, firstName, lastName, customerPassword, email, customerBirthday

email → customerID, firstName, lastName, customerUsername, customerPassword, customerBirthday.

The relationship is a One-to-Many relationship with Payment thus each customer can have multiple payment months. There is also a Many-to-Many relationship with Membership. The

customerID, CustomerUsername, and email each uniquely identify a customer and determine all attributes. A customer can have multiple payments and memberships.

The FD to payment table is:

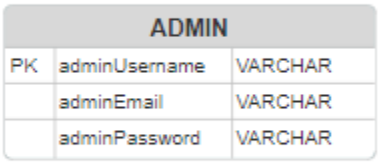
paymentID → billingAddress, cardHolderName,

customerID because payment ID uniquely identifies each payment, determining billingAddress, cardHolderName, and the linked customerID.

The FD to membership is:

membershipID → startDate, endDate, subscriptionCharge,

because membershipID uniquely identifies each membership, determining attributes like startDate, endDate, and subscriptionCharge. A membership can be associated with multiple customers.



ADMIN		
PK	adminUsername	VARCHAR
	adminEmail	VARCHAR
	adminPassword	VARCHAR

The admin table FD is:

adminUsername → adminEmail, adminPassword

adminEmail → adminUsername, adminPassword

The relationship is independent from other table with mean there is no direct relationship with other tables. The adminUsername and adminEmail uniquely identify each admin, determining the other attributes.