University of Ottawa
School of Electrical Engineering and Computer Science
CSI2132-2013
Database Project Specification

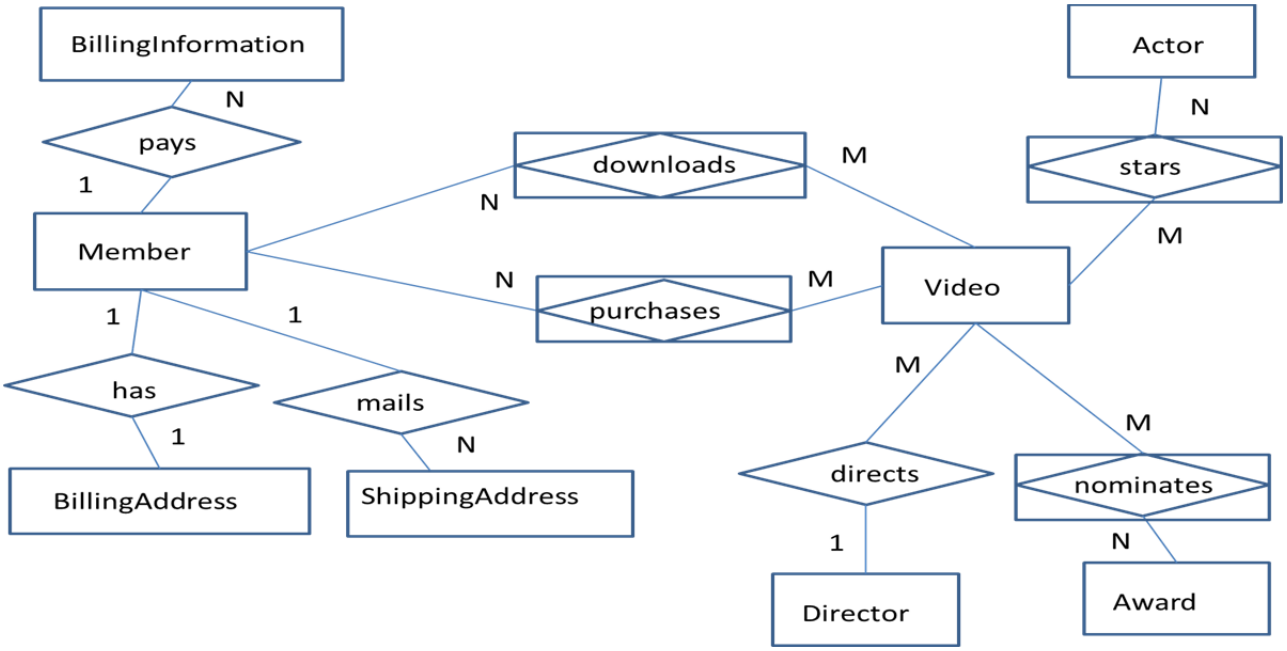
This document contains the requirements as created in class. Note that I made *some minor simplifications* to the design, in order to limit the scope of this project.

Instructions
1. Complete this project in a group of two (2) to three (3) students.
2. Demonstrate the project on **Monday April 8, 2013** in a 15 minute timeslot, as allocated by the TA.
3. Use PostgreSQL to complete this project, together with a language such as Java and JSP, or PhP, to create your Web-based front-end.

Deliverables:
Submit all your source code via the Virtual Campus**, before Sunday April 7, 2012 at 23h55**. Note that ***all*** students should submit the source code, not just one per group. All group members should attend the project demonstration.

Your task:
Consider an Online Video Store. This store allows members to purchase and/or download videos. (In this description, I focus on movies, but you may also include TV Series if you are so inclined.) Members have the option to either buy DVD or Blu-ray disks, which is then shipped to their home. (See e.g. www.amazon.ca for a description of this type of option.) Alternatively, they may choose to download streaming content, by e.g. buying a monthly purchasing plan from a site such as Netflix (https://signup.netflix.com/). A number of streamed videos may also be downloaded for free.

Here is a high level summary of the EER diagram, which only shows the entities and relationships.



Here is a list of assumptions that we made.

1. Online customers have to be *registered members* to purchase or download videos. They provide details such as their email addresses, their billing and shipping addresses, as well as their billing information (credit cards, Paypal, etc.). (Note that we may have some repetition here, since the billing and shipping addresses may be the same. This may be dangerous, and we will need to ensure that address changes are propagated to both.)
2. In this design, for shipment, we assume that shipping costs are pre-calculated and already added to the database.
3. A member is only allowed to return a video if the plastic cover has not been removed and the seal has not been broken. A customer has to pay the postage costs of any return herself and returns are handled as negative sales.
4. For each video, we keep the video's unique identifier, together with the sales price and the video genre, such as "horror" or "comedy". Note that the sales price for free downloads are set to $0.00.
5. In this database, a specific video is associates with only one genre. We also keep information such as the rating (e.g. Suitable for all ages or PG), whether it has been nominated for (or won) Academy Awards, and so on.
6. We maintain information about the main actors that star in a video. This information includes awards, and other personal information that we may choose to harvest from websites such as http://www.imdb.com/ .
7. Similar, we keep the information about the directors of the videos.

Here is a partial definition of the relations you will need to create. You may want to add additional information/attributes to personalise the project.

Information about members and their purchases or downloads:

- Member(MemberNumber, Lastname, Firstname, email, password, …)
- BillingInformation(MemberNumber, CardNumber, Type, …), where type is e.g. Visa, None (for free content), PayPal, Subscription, etc.
- BillingAddress(BAddresID, address1, address2, city, phone, postalcode,…, MemberNumber)
- ShippingAddress(SAddresID, address1, address2, city, phone, postalcode,…, MemberNumber)
- Purchase(InvoiceNumber, date-ordered, date-shipped, shipping cost, speed, carrier, …, MemberNumber, VideoID)
- Download(MemberNumber, VideoID, Date, Time, Fee, ….)

Here, the fee is set to be either equal to $0.00, a "pay per view" fee, or it may be a fixed monthly subscription fee that is charged at the end of each month.

Information about the Videos:

- Video(VideoID, Name, Year, Salesprice, Genre, Rating, Duration, InStock (y/n), …, DirectorID)
- Director(DirectorID, Lastname, Firstname, Date-of-birth,….)
- Award(AwardID, Year, Description, Category, …)
- VideoAwards(VideoID, AwardID, Year, Won(y/n), …) (This corresponds to the *nominates* bridge entity.)
- Actor(ActorID, Lastname, Firstname, Date-of-birth, IMdB link, …)
- VideoStar(VideoID, ActorID, RoleName) (This corresponds to the *stars* bridge entity.)

Requirements and Mark Allocation

You are required to complete the following tasks: (**Total 100 marks**)

1. **(10 marks)** Transform the description into a relational model and create all the tables in PostgreSQL. Add all other relevant attributes and remember to enforce entity and referential integrity.

2. **(10 marks)** Populate the tables with your own data, using the videos of your own choice. It follows that your data, and the attribute values you choose, should be sufficient in order to implement and test the queries specified below. Your database should include at least 30 different videos, in 6 categories. The number of customers should be at least 30 and a total of 25% of the customers only download free content. There should be at least 50 invoices in your database, with an average "basket size" of 2 videos. About 5% of the videos are returned. On average, the number of shipping addresses and the number of billing information are around 2 per customer. For each video, we store information about the 3 main actors. A video has only one director. We also store information about the Oscars (Academy Awards; see http://oscar.go.com/nominees) that movies have won, or have been nominated for.

3. **(10 marks)** Provide the user with the ability to add data to, and delete data from, the following tables in your database: VideoStar, BillingAddress and Actor.

4. **(40 marks)** Create a number of SQL queries to explore this data. The following is a list of "typical" queries that should be implemented.
   a. Display all the information about a user-specified video. That is, the user should select the name of the video from a list, and the information is then displayed on the screen.
   b. For each user-specified genre, list all the video names together with the last date that it was purchased or downloaded by a member. The user should be able to select the genre from a list.
   c. Find the billing information and shipping address for a user-specified member. The user should be able to select the account from a list.
   d. Given a user-specified genre, find the name(s) and email(s) of the members(s) who downloaded it the most often. *For example, John Smith is the member who has downloaded the most horror movies.*
   e. Find the total number of videos that are in our inventory, grouped by director.
   f. For each genre, find the name(s) and price(s) of the video(s) that are returned most often. *For example, in the Romantic Comedy section, the Bounty Hunter is returned most often.*
   g. Find the name(s) and emails of all member(s) that have a total purchase that is higher than the average purchase.
   h. Find the name(s) of the video(s) that are our best sellers, in terms of the total quantity sold and the total number of times it has been downloaded.
   i. Find the name(s) and price(s) of the video(s) where no-one ever bought the DVD or Blu-ray, but these videos were downloaded more than 4 times.
   j. Find the name(s) and date(s) of birth of the actor(s) that star in the video(s) that are downloaded most frequently.
   k. Find the name(s) of the director(s) who directed the video(s) that have the highest number of Oscar nominations. Also display the number of nominations these persons had. (*For example, Steven Spielberg and Kathleen Kennedy have directed 8 movies that have been nominated for Oscars.)*
   l. Find the name(s) of the director(s) of the movie(s) that have been most often nominated for the Oscars, but has never actually won an Oscar.
   m. Find the name(s) of the movie(s) that won the most awards. Display this information together with the name of the director, as well as the main actors. (*Three films have won 11 Academy Awards. They are Ben-Hur (1959), Titanic (1997) and The Lord of the Rings: The Return of the King (2003*)).

5. **(30 marks)** Create a web-based front-end GUI, for the user to directly query the database.

6. Additional effort, such as creating a superb front-end or including a multimedia component, may earn you up to **20 bonus** marks.