

## Project Assignment 1

(due February 25<sup>th</sup>, 2014, 3:30pm, in class—hard-copy please)

### Reminders:

- a. Out of 100 points. Contains 4 pages.
- b. Rough time-estimates: 1~2 hours.
- c. Please type your answers. Illegible handwriting may get no points, at the discretion of the grader. Only drawings may be hand-drawn, as long as they are neat and legible.
- d. There could be more than one correct answer. We shall accept them all.
- e. Whenever you are making an assumption, please state it clearly.
- f. Unless otherwise mentioned, you may use any SQL operator seen in class.
- g. Feel free to use the linear notation for RA and create intermediate views for SQL.
- h. **Important:**
  - a. For E/R diagrams, use only the style and notation given in the lecture slides.
  - b. A useful tool for creating E/R diagrams: <http://logicnet.dk/DiagramDesigner/>. You may have to manually draw-in some things though (like adding proper constraints etc.). There are other such programs too.
- i. Lead TA for the project: Qianzhou Du.

### Project Description: VTflix

The goal of the class project is to implement a database system application for a movie-ratings-cum-social-network system VTflix (like Flixster.com) where users can explore information about movies, rate movies, and see ratings of their friends. We will work on a subset of data crawled from the web. The project includes the following activities spread over the entire semester:

- Based on the project description, design the ER diagram draft. (this assignment)
- Model the data stored in the database (Identify the entities, roles, relationships, constraints, etc.). (this assignment)
- Design, normalize, and perfect the relational database schema. (Project Assignment 2)
- Write the SQL commands to create the database, find appropriate data, and populate the database. (Project Assignment 2)
- Finally and most importantly, write the software needed to embed the database system in the application. (Project Assignment 3)

The end result should be a functioning application that runs on the web and that uses your database to allow useful functionality.

## Q1. Warm-up [20 points]

Form a team of 2-3 people and decide on a group name. All groups will work on the same overall project, but each group will have its own instance of the project data and will build a separate and unique interface to its instance. Please answer the following questions:

- Q1.1 (5 points) Please write name of the project group, names and pids of the students in the group. (Note: We will use your group name to give you access and create your instance on PostgreSQL server.)
- Q1.2 (5 points) The domain of your database application. (Note: Just write a one-two sentence description of the overall context or situation that your application is a part of).
- Q1.3 (5 points) Write a couple of other "value-added" facilities your system could support? (Note: For example, such a facility could be a "recommender system" that makes movie recommendations for users based on buying trends. In other words, the recommender system is a facility that will be enabled by the presence of a database system).
- Q1.4 (5 points) What is the role of each project member in the project? (Note: You don't have to "commit" to anything now. We want to see if any of the group members brings special talents/experiences to bear upon the project. For example, if one of you has worked in a digital store project, (s)he may help identify design choices from this point of view. If one of you has experience in web-based software development, then that would be a good thing to mention).

## Q2. ER Diagram and SQL DDL statements [80 points]

You will work with video information, user information, ratings of videos, and users' social network.

- Each video has an id, title, release date, producer, color (e.g., black-n-white, or Eastman color), and which country it comes from.
- There are 3 categories of videos: movies, TV episodes and collections.
- Different TV episodes are different videos. A TV episode belongs to a collection and has an episode number.
- Each collection can contain one or more seasons. For each collection, we want to additionally record these season numbers. E.g. we want to model that a collection titled 'Seinfeld: The Early Episodes' contains seasons '1' and '2'.

- Movies don't have different parts e.g. 'Harry Potter I' and 'Harry Potter II' are different movies.
- One or more directors direct each movie or a TV episode. And each director is identified by an id and has name and age.
- Each video could belong to one or more genre (e.g., action, comedy, romantic, horror, thriller, and historical).
- VTFlix needs to keep the track of user ratings for each video (e.g., 1-5 and 1 for 'I hate it!', to 5, for 'I love it!'). A particular user rates a particular video at a particular time. One user can rate the same video multiple times.
- You also need to record the information of performers in the videos. They are identified by id, name, age, and whether they are actors or actresses.
- Each video gets a MPAA certification [Motion Picture Association of America]. Each certification has a unique name and content rating. E.g. "PG-13" rating means that some content may be inappropriate for children under the age of 13. Here the name of the certification would be 'PG-13' and content rating would be 'Inappropriate for children under the age of 13'.
- Each user is identified by id and has name, gender, age, and location.
- There is also the social network between users. Users can send 'friend-requests' to other users. Such requests can be accepted or ignored. Once accepted, the two users are considered 'friends'. If ignore, the user can send the request again. We want to additionally store when the requests were sent (timestamp), the result of the requests and when the request was answered (timestamp).

- Q2.1 (25 points) Draw an ER diagram for this database. Make sure to indicate primary keys, cardinality constraints, weak entities (if any), and participation constraints. There might be extra constraints which cannot be captured by the E/R diagram, make sure you mention them below the diagram. List any assumptions you make in the process. *Hint:* The E/R diagram should contain at least ~6 (may be more) entities; otherwise it is not of sufficient complexity for a CS 4604 project.
- Q2.2 (15 points) For each entity set and relationship, write a short description in plain English of what it represents or models. One or two sentences per entity set and relationship is enough. These descriptions are primarily to help us understand that you are modeling the VTFlix database correctly.
- Q2.3 (40 points) Translate the ER diagram in Q2.1 into relational database tables (i.e. give the SQL DDL statements). Make sure that the translation captures key constraints (primary keys and foreign keys if applicable) and participation constraints in the ER diagram. Identify constraints, if any, that you are not able to capture.

### Common Mistakes to avoid in design:

1. Modeling a database administrator explicitly in your E/R diagram. The DBMS usually has its own internal representation for administrators.
2. Missing arrows or rounded arrows in a many-one and/or a one-one relationship.
3. Missing arrows from a weak set to the set(s) that provide its key attribute(s).
4. Using inheritance when there is no "ISA" connection between two sets.
5. Forgetting that when entity set *B* inherits from entity set *A*, *B* inherits set *A*, *B* inherits **everything** that *A* has. In addition, *B* can define its own attributes of its own. Therefore, there is no need to repeat all the attributes/relationships that *A* has again for *B*.
6. "Cooking up" multi-way relationships, weak entity sets, or inheritance when they are not needed.
7. Forgetting to underline key attributes in the E/R model.
8. Repeating (reusing) names for different entity sets or for different relationships within the same entity set, i.e., using the same name to denote two different things.
9. When converting a multiway relationship to many two-way relationships using a connecting entity set, forgetting to introduce many-one relationships!