Example Assignment:
The Relational Data Model and SQL

COMPSCI 2DB3: Databases

Jelle Hellings Holly Koponen

Department of Computing and Software
McMaster University

Description

Our local cinema chain recently introduced a *subscription service* that gives regular visitors the option to watch a film at their cinema whenever they want for a fixed price per month. To promote this new service, an online community consultant proposed to expand the cinema website with elements that encourage *engagement*. To do so, the consultant proposed to build a social online environment into the cinema ordering website such that subscribers can review and discuss films and make friendships with other subscribers. The consultant already put in some legwork by providing an ER-model for this social online environment and a relational schema to store basic film information.

The consultant provided a two-part design. The first part describes the social media features and the second part describes a high-level sketch of the information maintained per film (which the consultant figured would be of use for other parts of the website as well).

Part one: The social media features

The social media features center around its users (the subscribers). Users have usernames, which are not necessary unique. To distinguish users with the same username, the consultant proposed to assign each such user a different number. Users will log in using their unique email address and a *password*. According to the consultant, the password should not be stored in plain text, but as a pair of a 512-bit (64-byte) hash and a 64-byte salt value. Together, the hash and salt are sufficient to determine whether a login attempt provided the right password, this without providing easy access to the stored password (even if the system gets compromised).

The social media features break down in three types:

- Users can *befriend* other users. If a user X indicates that Y is a friend, then this indicates that X follows the activities of Y, but this does not imply that Y is also a friend of X.
- Users can *review* films. Each review assigns a score to the film and reviews can be revised with more information. Each review can optionally have a *video* component (e.g., as in a vlog) and optionally have a *text* component (e.g., as in a blog).
- Users can place *reactions* on reviews and on other reactions (a reaction is either on a single review or on a single reaction, but not on both).

The ER-Diagram for the social media feature can be found in Figure 1.

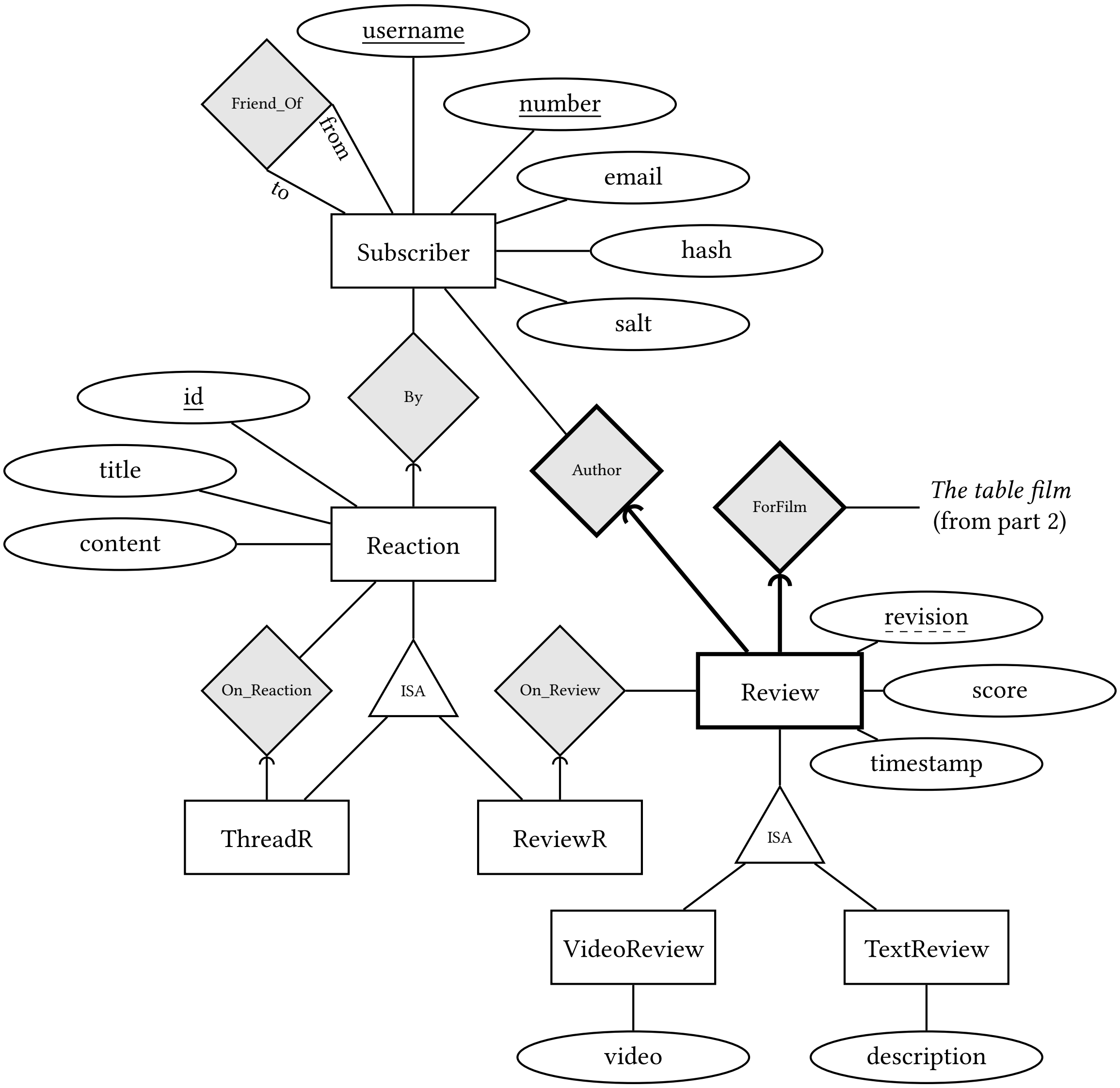

Figure 1: ER-Diagram for the social media features.

Part two: The film information

The consultant sketched the following layout of the database maintaining film information, but indicated that some details still need to be figured out:

- A *table* **person**(id, name, birthdate$_{\text{optional}}$).
  This table provides basic information on people that work on films.
- A *table* **film**(title, year, creator, duration, budget).
  This table represent films. The consultant figured out that films can have the same title and be released in the same year (e.g., in 2005 there was a film *Chaos* directed by Tony Giglio and another film *Chaos* directed by David DeFalco—The consultant didn't see either of them, however). To distinguish between films with the same title and made in the same year, the consultant proposed to use the main director (*creator*, which refers to the id column in the **person** table) of the film.
- A *view* **film_info** to easily retrieve a film (columns title, year, duration, and budget) together with the name of its creator (column creator_name).
- The consultant wants a table **role** that stores, per film, all people that worked on the film and on the role these people had (e.g., actor, writer, director, producer, costume, casting, editor, or makeup). The consultant had difficulties coming up with a proper design for this table, but noted that people can have several roles in the same film.
- A *constraint* that all film creators (mentioned in the **film** table) have the role of director for that film (as recorded in the **role** table).

Assignment

The goal of the assignment is to translate the above description into proper SQL statements that create tables and set up all required constraints. To do so, you proceed in two steps. First, you write a report in which you step-by-step translate the ER diagram of part one (Figure 1) into a relational schema and complete the details of the relational schema of part two. Next, you provide the translation of your relational schema into a sequence of SQL statements that will set up the database.