

University of Victoria
CSC 370: Database Systems

ASSIGNMENT 6

Due on: Friday, November 1 at 11:59pm (56 marks)

Introduction

The purpose of this assignment is to practice creating entity-relation diagrams that meet the requirements outlined by a written description and a provided mockup.

Scenario

You were hired to design a database for a new BC-based competitor to Reddit called BC Chat. Reddit is an online discussion forum¹ where users can post and discuss content. Your employer has big plans for BC Chat and wants to differentiate itself from Reddit by including new features. Therefore they do not want to simply copy Reddit's format and design, but instead want to start from scratch with a new design.

To make their new product appear more lively, they scraped Reddit for BC-related discussions and compiled a `.csv` file containing that data. They want to import this into their new database to make their platform look like it has people using it already. This is where you will start with the database design process.

Provided Materials

You are provided with the following materials:

1. A written description of the system.
2. A mock-up showing the desired look and feel of the system.
3. A `.csv` file containing data that they want to migrate to the new database — the “data dump”. (You used this on the last assignment.)

¹https://en.wikipedia.org/wiki/Internet_forum

3.1 Written Description

The discussion forum that your employer envisions is made up primarily of user-provided posts grouped into topics. Each topic contains a hierarchical organization of posts, as users can create new posts that either reference no other post or a previously existing post. Each post belongs to exactly one user, but if that user is deleted (e.g., because they were causing trouble), their posts should remain. Each post belongs to exactly one topic; if that topic is deleted, then so too should the posts belonging to that topic be deleted. Trolls (troublesome users) can also simply be banned, meaning that they are no longer able to make new posts.

Users should be able to browse for discussion topics based on categories. Every topic should have exactly one category. Administrators and moderators should also be able to “pin” topics so that they always show up at the top of the topic listings for any particular category. Categories should have unique names as well as a bit of descriptive text.

Topics can be “tagged”. Unlike categories, where topics could only have one category, topics can have as many tags as desired or none at all. Tags should be defined separately from topic creation and can be re-used across multiple topics. Only administrators should be allowed to define new tags. Tags should have unique names.

Every post can be rated by every user. After each post has been rated, the combined rating should be re-calculated and stored alongside the post’s information. Ratings can be positive (1) or negative (-1). A user can only rate a post one time. If a user mistakenly rates a post, they can re-rate it, or they can clear it, resulting in a rating of 0.

Users should be able to optionally “watch” both topics and posts. When a topic is watched, then any time a new post is added to that topic an email should be sent to the user notifying them of the new post. Watching a post is similar, except that only postings that are direct replies of the watched post should result in email notifications. In both cases, it should be possible for the user to view a listing of new postings relative to the time when they first started to watch the post or topic.

To keep track of whether or not there have been new replies to a topic since a user last visited, a log of the user’s activity should be created. The log should keep track of the user, the topic, and the time that they last viewed it on. If the user views the topic that they have visited previously and there are new posts, those posts should be highlighted in green.

If any topic or post is deleted, then all of the ratings, activity, and watch statuses for that topic or post should also get deleted. Additionally, Categories cannot be deleted if they contain any topics at all — all of the topics in that category should be migrated to another category before deletion.

3.2 Mock-up

Their mock-up shows what a user might see when they are viewing a topic in the online discussion forum.

This is my thread!

Category

tag1, tag2, tag3, tag4

▲ 10 ▼		4 Posts	3 Users	2024-10-10 Last Reply	New Posts	Watch
▲ 43 ▼	▲ 0 ▼	<div>avatar</div> <div>User A</div> <div>Posts: 100 Joined: 2023-01-01</div>	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam rutrum commodo sapien et dictum. Suspendisse potenti. Nulla facilisi. Ut dignissim ante ut lacus lobortis egestas. Sed bibendum nunc quis varius rutrum. Nulla laoreet ante et lacus auctor, auctor condimentum urna tempor. Vivamus quis facilisis metus, ut ultrices nibh. Nunc non fringilla ligula, in malesuada tortor. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos.			<div>Watch</div> <div>Reply</div> 2024-10-08
		<div>avatar</div> <div>User B</div> <div>Posts: 166 Joined: 2023-08-01</div>	Donec justo tellus, feugiat vitae ipsum vitae, convallis volutpat nulla. Maecenas venenatis libero nisl, ut facilisis purus rutrum vel. Suspendisse at felis et dolor sagittis pulvinar. Mauris molestie quis tortor in pretium. Aliquam aliquet ac ipsum eu posuere. Morbi sed augue accumsan nisi lacinia facilisis eu non metus. Pellentesque pellentesque, risus nec accumsan egestas, leo urna elementum erat, ut dapibus mi velit ullamcorper orci. Maecenas tempus egestas arcu sit amet dapibus.			<div>Watch</div> <div>Reply</div> 2024-10-08
	▲ 2 ▼	<div>avatar</div> <div>User C</div> <div>Posts: 30 Joined: 2024-02-12</div>	Cras velit nisi, sollicitudin dignissim pharetra sed, efficitur sit amet ligula. Etiam suscipit quam risus, hendrerit euismod diam auctor in. Aliquam tincidunt a tellus a placerat. Aliquam in mi id ipsum euismod tempus vitae vitae odio. Vivamus faucibus nibh purus, sed varius mi sodales nec.			<div>New</div> <div>Watch</div> <div>Reply</div> 2024-10-10
		<div>avatar</div> <div>User C</div> <div>Posts: 30 Joined: 2024-02-12</div>	Quisque mollis vestibulum magna sed sollicitudin. Morbi scelerisque dui ipsum, vitae luctus urna tempor suscipit. Donec faucibus diam vitae dolor rutrum eleifend. In nec erat id felis eleifend viverra. Curabitur venenatis est magna, nec convallis dui porttitor quis. Phasellus in hendrerit mi. Maecenas in massa vel elit rutrum cursus. In hac habitasse platea dictumst. Sed vitae neque odio. Mauris et scelerisque elit.			<div>Stop Watching</div> <div>Reply</div> 2024-10-09

The Conceptual Model

For the upcoming questions, you will need to reference the written description and the mock-up of the system. This assignment continues where Assignment 5 left off, however, the columns listed in the data dump (data to import) will have names based on the previous system that may be somewhat different than what you find based on analyzing their requirements and their mock-up. By comparing your conceptual model to the normalized database from Assignment 5, you can check your work.

1. (14 points) Based on the written description of the system and the mock-up, identify all of the entities, attributes, and identifiers (if any) you need to make up the database.

Provide your answer in the form of a list of entities, with each entity in the form of:

- Entity (Identifier1, Attribute1, Attribute2, ...)

Starting Point

Some of the entities have already been identified for you

- User (DisplayName, Email, Password, IsBanned, IsModerator, IsAdministrator, RegisteredOn, Avatar)
- Topic (TopicName, IsPinned, CreatedOn)
- Category (CategoryName, CategoryDescription)

Note: At this stage you do *not* need to consider implementation details or anything related specifically to the relational model (e.g., primary keys or foreign keys). You will do that later (in Q3). You may find that there are no attributes associated with an entity. You may not be able to spot identifiers. Your only goal is to parse the provided description and mock-up to pull out information that could be used to make an ER diagram. There may be details in the description that cannot be represented via this diagram and there may be missing details that you will need to make assumptions about later on.

2. (24 points) Based on the written description and the mock-up, construct a conceptual ER diagram using Crow's Foot notation that demonstrates the relationships between the entities, as well as the minimum and maximum cardinalities.

Note: Since you will have listed the attributes and identifiers in question 1, you do not need to include them in your diagram, although you are welcome to, and you may find it helpful when it is time to implement the database. You may also label the relationships if you feel it is helpful, but this is not a requirement. You can draw the diagram on paper or use a tool such as draw.io.

Your diagram must...

- include the relationships between entities.
- indicate maximum cardinalities.
- indicate minimum cardinalities.

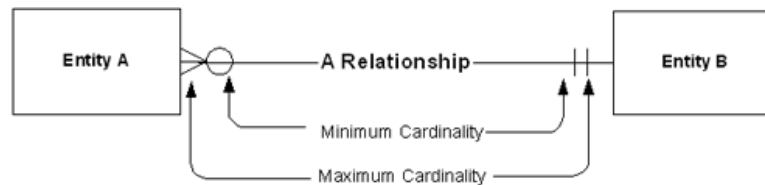
If you think that any of these are not clear from the provided materials, then state an assumption in words like in the example below.

Example Assumptions

Considering the relation between a car and its wheels, here are some possible assumptions:

- Teachers might not teach any classes (regarding minimum cardinality).
- Students can enrol in many classes (regarding maximum cardinality).

Crow's Foot Notation



3. (18 points) Based on the ER diagram you completed in question 2 and the process of normalization you did in the last assignment, you must now implement the database using PostgreSQL. This is the same database as in the prior assignment, so you should extend your solution to Assignment 5's question 5² to meet the additional requirements that were identified from an analysis of the case study.

Provide the DDL **CREATE TABLE** syntax to implement each table in your database.

Notes:

- You will need to translate all entities into tables.
- Identify scenarios where junction tables are needed to form many-to-many relationships.
- Use foreign keys with referential integrity constraints.
- If you were unable to identify any natural keys, use surrogate keys.
- You may find that you are unable to implement certain minimum cardinality constraints.
- The order in which you create tables matters. E.g., you can't create the "Topic" table before the "User" table.

²The solution to Assignment 5 will be posted on Monday, October 28.

The following tables have already been designed for you (you can leave these tables out of your submission):

```
CREATE TABLE "Category" (  
    "CategoryID" SERIAL NOT NULL PRIMARY KEY,  
    "CategoryName" TEXT NOT NULL UNIQUE,  
    "CategoryDescription" TEXT NOT NULL DEFAULT ('')  
);  
  
CREATE TABLE "User" (  
    "UserID" SERIAL NOT NULL PRIMARY KEY,  
    "DisplayName" TEXT NOT NULL,  
    "Email" TEXT NOT NULL UNIQUE,  
    "Password" TEXT NOT NULL,  
    "IsBanned" BOOLEAN NOT NULL DEFAULT FALSE,  
    "IsModerator" BOOLEAN NOT NULL DEFAULT FALSE,  
    "IsAdministrator" BOOLEAN NOT NULL DEFAULT FALSE,  
    "RegisteredOn" TIMESTAMP NOT NULL DEFAULT NOW(),  
    "AvatarUrl" TEXT  
);  
  
CREATE TABLE "Topic" (  
    "TopicID" SERIAL PRIMARY KEY NOT NULL,  
    "TopicName" TEXT NOT NULL,  
    "AuthorUserID" INTEGER REFERENCES "User" ("UserID") ON DELETE SET NULL,  
    "CategoryID" INTEGER NOT NULL REFERENCES "Category" ("CategoryID") ON DELETE RESTRICT,  
    "IsPinned" BOOLEAN NOT NULL DEFAULT FALSE,  
    "CreatedOn" TIMESTAMP NOT NULL DEFAULT NOW()  
);
```