

Assignment 1 (worth 15%)

[Please read this entire document. There are important notes at the end.]

Aims

This assignment aims to give you practice in

- cooperating in a virtual environment
- analysing/refining problem requirements
- designing ER data models based on requirements
- mapping ER data models to SQL schema definitions

The ultimate goal is to build an SQL schema to represent a photo-sharing website called myPhotos.net similar to [Flickr](#), [Photobucket](#), [Photo.net](#), and [so on](#).

Assignment Structure

This assignment will run in two stages: (1) design, (2) implementation.

Stage 1 is a design exercise (discussion), carried out during Week 4 on the WebCMS Forums under "Assignments/Assignment 1/Assignment 1 (Stage 1)" section as well as Lecture/Lab and is non-assessable. However, preparation of your ER model beforehand and participation in the discussion are highly recommended. By the end of Week 4, you will have discussed the data requirements of the application and developed a complete ER model based on the requirements. We will then release a sample ER design at the beginning of Week 5. This ER model will be provided to you for Stage 2. You may discuss with your Lab Tutor during Week 5 the discrepancies between your design and the provided sample ER model.

Stage 2 is an individual implementation exercise. We will post a "standard" ER design that best captures all aspects of the application requirements. You should then individually develop a PostgreSQL schema to accurately implement the provided standard ER model. The PostgreSQL schema is the only thing that needs to be formally submitted (details on how to do this will be provided later along with the Sample/Standard ER Design).

Timeline

Week 4 (Monday 13 to Sunday 19 August 2018)	Discussion of the requirements/design on WebCMS Forums and during the lecture/lab
Monday 20 August 2018 (Week 5)	Sample/Standard Stage 1 ER model released and assignment submission instructions provided
Sunday 2 September 2018 @ 23.59 (Week 6)	Deadline for submission of PostgreSQL schema

Assessment

This assignment is worth a total of **15 marks**.

The submitted schema (in a file called) will be primarily auto-marked and manually checked to determine:

- whether it is syntactically correct; and
- how accurately it captures the sample ER model.

More details on what you need to submit and how the marking is done will be provided when the sample/standard ER model is released.

The Problem Domain

Photo-sharing sites have proliferated on the Web over the last few years, and have proved remarkably popular (Flickr, for example, has around 90 million users). The sites allow users to register an account, and then use that account to upload and manage a collection of their photos. There is usually a limit on the storage allowed for each user; this can generally be extended via a paid "premium" subscription. Management of photos is typically done by grouping them in various ways. And, of course, to flatter users' inner [Ansel Adams](#), most sites allow other users to rate your photos and comment on how wonderful they are. Photo-sharing sites are typically backed by massive storage for the photos themselves, along with a database to hold all of the data about photos, users, ratings, etc.

The aim of this assignment is to develop a database design that can support the core functionality of a site like Flickr. Since a site like Flickr offers too much functionality to be feasibly modelled in an assignment of this size, we'll restrict ourselves to considering a subset of its functionality. Note that we are dealing with data representation only; we are not actually going to implement any of the functionality. However, whatever data structures you design must be rich enough to enable the functionality below to be realised.

Let's call our website [Flickr](#). The site has to deal primarily with photos, people, collections of photos, and groups of people. The public front page of [Flickr](#) displays a collection of random photos, drawn from all of the public and safe photos on the site. When a user logs in to [Flickr](#), the first thing they see is a list of their photo collections, with each collection appearing as a thumbnail of the "key photo" for the collection. There will be a menu that allows them to upload photos, create and organise photo collections, create groups, post comments, and all of the other things that people like to do in a typical social networking environment.

To get you started with your design, here are some details on the kinds of data items that might need to be stored in the back-end of the [Flickr](#) web site:

Photos

- photos can be uploaded onto the site and must be in JPEG format
- each photo is stored in the file system and a record about the photo is created
- the photo record contains a title for the picture, descriptive text and the file size in KB
- the photo record also contains the date when the photo was taken and when it was uploaded
- there is also scope for an optional description of the technical details of the photo (e.g. what kind of camera, lens, exposure settings, etc.)
- the system often displays thumbnails of photos (e.g. in photo lists); each photo has a thumbnail, also stored on the filesystem, created when the photo is uploaded
- users can make comments on photos (see Comments below)
- users can also rate photos (on a 1-5 star scale; 1=ordinary, 5=exceptional)
- each photo must be assigned a "safety-level" when it is uploaded; possible settings are: Safe (suitable for children), Moderate (not suitable for children), Restricted (R-rated)
- the owner of a photo can also set its visibility; possible settings are Private, Friends, Family, Friends+Family, Public

Photo Collections

- a photo collection is simply a group of photos
- each collection has an owner (a user), a title and some descriptive text
- a collection may include photos from any user, as long as they are public
- photos within a collection appear in an order defined by the owner
- one photo is defined to be a "key photo" for the collection and is displayed whenever the collection appears in a list of collections (along with the title)

People

- for every person associated with the site, we need to know at least their email address and their name
- we also store separate family and given names, for sorting lists of people
- everyone must have at least one given name, but some people may have no family name (e.g. "Prince")

Users

- users are people who can log in to the site and upload pictures

- each user needs to be registered with the system and must provide an email address and a password
- their email and password are used for authentication when they log in to the system
- we record the date when each user joins the site (registration date)
- logging in takes them to their home-page, which displays a list of their photo collections
- once logged in, they can modify information related to themselves via their "Account" page
- users may provide additional information about themselves, including their personal website, their gender and their birthday
- they can also upload a small image of themselves (JPEG, smaller than 64KB)
- users can also create contact lists for people they might want to inform about their photos

Contacts

- a contact is a person outside the system who is associated with a user
- all that we really need to know about a contact is their name and their email address

Groups

- users can form groups (e.g. special interest groups) consisting of a set of users
- each group is created and managed by some user (its owner)
- a group can be set up in a number of modes: private, by-invitation, by-request
 - private groups are managed by a user and might be "Friends", "Family", etc.
 - by-invitation groups allow the owner to invite people to join
 - by-request groups allows users to ask to be added to a group
- private and by-invitation groups are not visible when users search/browse groups
- each group has a title, a set of photo collections, and a list of discussion threads (a discussion thread is simply a title and a list of messages)

Tags

- photos can be tagged by short phrases like "landscape", "street life", etc.
- tags are used as a mechanism for finding groups of similar photos (they could also be viewed as defining informal collections)
- when a user is tagging a photo, auto-completion will be used to suggest tags, to try to ensure consistent usage

Comments

- a comment is a small piece of text written by a user
- comments can be threaded (by one comment referring to an earlier comment)
- for each comment, we need to record the author and the time it was posted

The above should provide sufficient information to get started on a design, but you may want more details. Note that not all of the things mentioned will end up in the data model; some are interface functionality, which does not enter into the data model.