

Project Description

In this project, you will design and implement a database for keeping track of information for an online “**SOCIAL NETWORK**” system (e.g. a simplified version of Facebook!). You will first design an **EER schema diagram** for this database application. Then, you will map the **EER** schema into a relational database schema and implement it on ORACLE or MySQL or some other relational DBMS. Finally, you will load some data into your database (via user Interface), and create some queries and update transactions.

Part 1

Assume that the following requirements were collected for this application:

1. The database keeps track of **PROFILES**, **PAGES**, **POSTs**(Text, Audio, Video, Photo), **COMMENTs**, **LIKEs**, **MESSAGEs**, and etc.
2. Each **PROFILE** (Account) has a unique **PROFILE_ID** (assume this is a unique integer generated by the system for each new account, such as 1, 2, 3, ...), a **First Name** (string MAX 100 characters), a **Last Name** (string MAX 100 characters), a **Mobile No.** (a string of 12 characters such as “817-333-6000”), an **Email** (a UNIQUE string of MAX 100 characters), a **User Name** (a UNIQUE string of between 8-12 characters), a **Password** (a string of between 8-12 characters), **Created Date** (Datetime), and etc.
3. Each **PAGE** has a **PAGE_ID** (a unique number for each page – assume it is a number 1001, 1002, 1003, ...), Page Name, Description, Category (a word or two to best describe each page), Members (registered profiles e.g. likes), Views, Admin(s), Image(e.g. a header/logo for each page).
4. The database will keep track of the current (active) **PAGES** as well as active **PROFILES**. For each **PAGE**, the information will include the specific **POSTs** and **PROFILES** as well as the **COMMENTs** for each **POST**, Created Date, and likes, etc.
5. The database also will keep track of the **MESSAGEs** between different **PROFILES** (Accounts). For each Message, the information will include the Sender, Receiver, Message, CreatedDate, etc.

You will first design an EER schema diagram based upon the SOCIAL NETWORK database requirements specified above, and create an EER schema diagram and documentation report describing your design choices. As part of this assignment, you should identify any missing or incomplete requirements, and explicitly state them in your documentation. You should also explicitly state any assumptions you made that were not part of the requirements listed above.

Part 2

The second part of the assignment will be to map the EER schema design to a relational database schema, and create the tables corresponding to the relational schema using the ORACLE DBMS (or MySQL). You will add to your report a listing of the CREATE TABLE statements. Specify as many constraints (key, referential integrity) as you can in the relational schema. You should state the choices you made during the EER-to-relational mapping, and the reasons for your choices.

Part 3

The third part of the project is to load some data into the database, and apply certain update transactions and retrieval queries. You will create your own data. Include at least 20 accounts, and (at least) 10 pages, and (at least) 10 posts in each page.

The followings are the tasks for the third part of the project:

1. **Load some initial data** (as discussed above) into the database tables that you created in Part 2 of the assignment. You can either write a loading program, or use SQL/PLUS (insert command), or use SQL/FORMS or any other method you are familiar with. Your data should be kept in files so that it can easily be reloaded during debugging. The data format should be designed by you. (Note: You can also use the transactions created by you in item 3 below to load some of the data).
2. **Write queries** to retrieve and print all the data you entered. Try to print the data so that it is easy to understand (for example, print appropriate headings, such as: Accounts, Pages, Posts, Messages, etc.).
3. Write the following database **update transactions** using either PRO*C or JAVA/JDBC or PHP or some other programming language or scripting language.
 - 3.1 The first transaction is to add information about a new PROFILE (ACCOUNT).
 - 3.2 The second transaction is to add all the information about a new PAGE.
 - 3.3 The third transaction is to create a new POST (this must find an available account and page).
 - 3.4 The fourth transaction is to show the details of a page (by PAGE_ID).
 - 3.5 The fifth transaction is to return a list of posts by specific account (AccountID) in different pages.
 - 3.6 The sixth transaction is to return the total posts/comments in of each page by all accounts(users) on specific date (e.g. PostDate).
 - 3.7 The seventh transaction is to update information of pages (e.g. update the Page name(s) or description of page(s)).
 - 3.8 The eighth transaction is to delete a PROFILE (What if a Profile is the Admin of at least one page?!)

4. Each transaction should have a user-friendly interface to enter the information needed by the transaction. This can either be a Web-based interface (recommended), a command line interface, or a forms interface.
5. Test your transactions by adding a few new Accounts, Pages, and Posts.

Due Dates:

1. **Parts 1 and 2 Due Date: Tuesday, October 30.** This should include for Part 1 the EER diagrams for your designs as well as documentation describing any assumptions you made, and the reasons for your design choices. Draw the EER diagrams using the notation in the textbook. You can also use UML class diagrams notation. You can use any drawing tool for drawing your diagrams. For part 2, what you turn in should include your relational schema diagram design and your CREATE table statements, and documentation describing your EER-to-relational mapping choices.
2. **Part 3 Due Date: Tuesday, November, 27.** This will include a demo demonstrating that your implementation works, as well as demonstrating your transactions to the GTA. Source code of all your transactions should be submitted, as well as the data files. A demo schedule will be determined before the due date. The files that you turn in should be submitted by midnight of each due date using blackboard. In addition, hardcopies of your EER diagrams and documentation (Part 1), and relational schema diagram with EER-to-relational mapping documentation (Part 2) should also be scanned and submitted. For part 3, you should submit a copy of the printout of the data as loaded in the database as discussed in item 2 of Part 3 of the assignment.

Important Notes:

- 1) Late penalty: -5% per day late.
- 2) Make a zip file with all the files together and submit it on Blackboard.
- 3) Each project can be done in a team of (MAX) 3 persons, or individually.
- 4) If two or three persons do the project, they will receive the same grade.
- 5) Copying from other students or teams (or any other resources) is not permitted and will result in a grade of **ZERO** for the entire project.