Cairo University
Faculty of Engineering
Computer Engineering Department
CMP 202& CMP N202

بسم الله الرحمن الرحيم

"نرفع درجات من نشاء وفوق كل ذي علم عليم"

Database Systems

Database Project Requirements

Project Description

Objectives

This project should make the student able to:

- ❑ Go through the phases of creating a relational-database-based system
    a. Analysis phase
    b. Design phase
    c. Implementation phase
- ❑ Use database design tools (e.g. SQL server, Oracle, MySQL…….)
- ❑ Create database application
- ❑ Work in a team and learn how to communicate and organize work with others

Project Phases

Project Phase	*Deliverables*	*%*
Analysis Phase	**Project Proposal**	**5%**
Design Phase	**ER Diagram**	**20%**
	Database Schema	**10%**
Implementation Phase	**Media Delivery**	**65%**
	Project Presentation	

Notes:

Feedback for each delivered report is **one week after** delivery.
Project assigned total grade will be announced in the lecture.
One day late makes you lose 1/4 of the grade.
Two days late makes you lose 1/2 of the grade.
Three days late makes you lose 3/4 of the grade.

Project Requirements

It is required to create a relational-database-based system. Your system should be based on a ***real-world*** model. You will go through the following phases:

1- Team Formation Phase

Number of students per team is 3 to 4 students.

2- Analysis Phase

- ❑ Your team should select a ***real-world*** application that uses database for its operation.
- ❑ Identify the requirements for your database. You may:
  - o Meet people who use the real-world model of your system,
  - o Get a list of requirements from a documented system, or
  - o Identify reasonable and acceptable requirements by yourself.
- ❑ Applications are not supposed to be neither *too complex nor trivial*.
- ❑ Innovative ideas are rewarded. Think of non-conventional applications.
- ❑ **Prepare and deliver a *project proposal* report.**

3- Design Phase

In this phase, you will identify the structure of the database that will be used in your system.

- ❑ Identify the database entities and the relationships between these entities. This is done by thinking about the real-world model of your system.
- ❑ Draw the "Entity-Relationship" (ER) diagram for your database or the “Extended Entity-Relationship” (EER) if applicable.
- ❑ A good, clear and a well-defined ER diagram will make it easier for you when creating your tables and relationships.
- ❑ **Prepare and deliver the *ER diagram* report.**

Your ER report will be evaluated by instructors to correct any design errors
Then:

- ❑ Starting from the corrected ER report
  - o Follow the algorithm described in the text book to map the diagram into database relations
  - o Show primary and foreign keys
- ❑ **Prepare and deliver the *database schema* report.**

4- Implementation Phase

In this phase you will convert your design into a working system. This phase implies both the ***creation of the database*** itself and the development of the ***user interface*** for the system (windows application/web application/mobile application using **C#**). Each team should divide the workload among its members to achieve these two activities in ***parallel***.

- ❑ Identify the tools you are going to use for implementation.

- □ Create the database using a database engine (e.g. SQL Server, Oracle, MySQL etc.)
- □ Develop friendly GUI using tools like VC#.net or ASP.NET
- □ Don't assume that the system users know anything about the database systems or database queries. Your interface should hide such technical details from the user.
- □ **Prepare a *presentation* and a *demo* to show your work.**

Project Deliverables

Cover Page

All project deliverables should have the following cover page

Cairo University
Faculty of Engineering
Computer Engineering Department
CMP 202 (or CMP N202 for CHS)

<Write here SEM or CHS
followed by team number>
example1: **CHS - team 7**
example2: **SEM - team 9**

Introduction to Database Systems
<Project Name>
<Report Title>
Team Number: #

Team Members:
<Name> <Sec> <B.N> (or Name and ID for CHS)

Contact info:
Write one email to be able to contact your team.

<**Date**>

Note: ***SEM**: for semester, **CHS**: for credit hours system.*

Analysis Phase Deliverables

Project Proposal Report

Report should contain

1- Cover page. (described above)
2- Proposed project description: one or two paragraphs to describe the project (the system) you intend to implement.
3- List of system users: who can use your system? (2-4 types of users)*
4- For each user, write functionalities that your system provides (5-10 functionalities per user type)*
5- List of real-world entities that should be present in your system. (10-15 entities)*

Notes:

❑ It is recommended to write three different project ideas ordered from the most idea you prefer.

❑ Storing data is NOT functionality; rather functionality is what users will do with stored data.

❑ Reports should be submitted on the elearning course page: http://www.elearn.eng.cu.edu.eg/course/view.php?id=29, links to every submission will be announced.

❑ If any problem happens with your submission on the website, you can exceptionally send to the following email: dina.tantawy@eng.cu.edu.eg

* *The above numbers are for guidance and may be somehow tolerated.*

Design Phase Deliverables

1- ER Diagram Report

Report should contain

1- Cover page. (described above)
2- Problem definition.
3- List of system users and privileges of each user.
4- List of entities and a brief description for each entity. This doesn't mean to list the attributes of the entity. It means to describe what this entity represents in the database.
5- List of relationships and a brief description for each relationship.
6- ER Diagram.
    a. The first page should show the entities and the relationships between all the entities (with no attributes on entities. Relationships attributes should be shown here).
    b. The rest pages should show the attributes of each entity.

Note: *State **explicitly** any reasonable **assumptions or restrictions** you have.*

2- Database Schema Report

Report should contain

1- Cover page. (described above)
2- The new ER diagram after correcting any errors in ER report according the ER feedback.
3- Database schema diagram showing
   a. Database relations (tables) showing primary keys.
   b. Foreign keys showing the referenced relations. (Can be shown as arrows from referencing relation to referenced relation)
4- Any database **constraints**.

Implementation Phase Deliverables

Deliverables

1- A Soft copy of the Project.

Deliver a CD that contains
(1) ID.txt file. (Information about the team and its members)
(2) The project files.

Your tables should contain ready sample data - At least 20 tuples in major tables.

Delivery Schedule: To be announced.

Evaluation Criteria

- ❑ **User support (15%)**
  - Different types of users supported by your application.
  - No hardcoded passwords. Users' data should be stored in a DB table.
  - Each individual user should have a username and a password.
  - Application should enable users to sign up, login, logout and change password.
  - Only first admin user can be created manually then he should be able to change his password.
  - Admin should be able to create other admins through the application.
  - Encrypted password is recommended.
- ❑ **System Functionality (50%)**
  - How comprehensive is the functionality supported by your application for different types of end users.
  - All access to the database must be done through the application. Direct access to database table is not allowed.

- ❑ **Reporting Facility (20%)**
  - ▪ Statistical reports depending on your application
    - ▪ Detailed statistical reports for specific parts of the database
    - ▪ Managerial level reports to see some overall statistics the whole application
- ❑ **GUI (15%)**
  - ▪ Do not expect users to be database programmers.
  - ▪ Do not expect users to memorize IDs. Use names instead of IDs.
  - ▪ Interface should be operation-oriented not table oriented. i.e. categorize you end user interface items with respect to functionalities to be supported rather than tables to be accessed.
- ❑ **Individual's role**: Grade for each individual according to his/her role.

Important Note

- Each team member should identify EXACTLY his/her role in the project; as he/she will be evaluated accordingly, not necessarily that all team members got the same mark.

Appendix A

Super entity and sub entity (Inheritance)

This appendix explains how to handle **inheritance** (**super-sub** entity) relationship during project design phase.

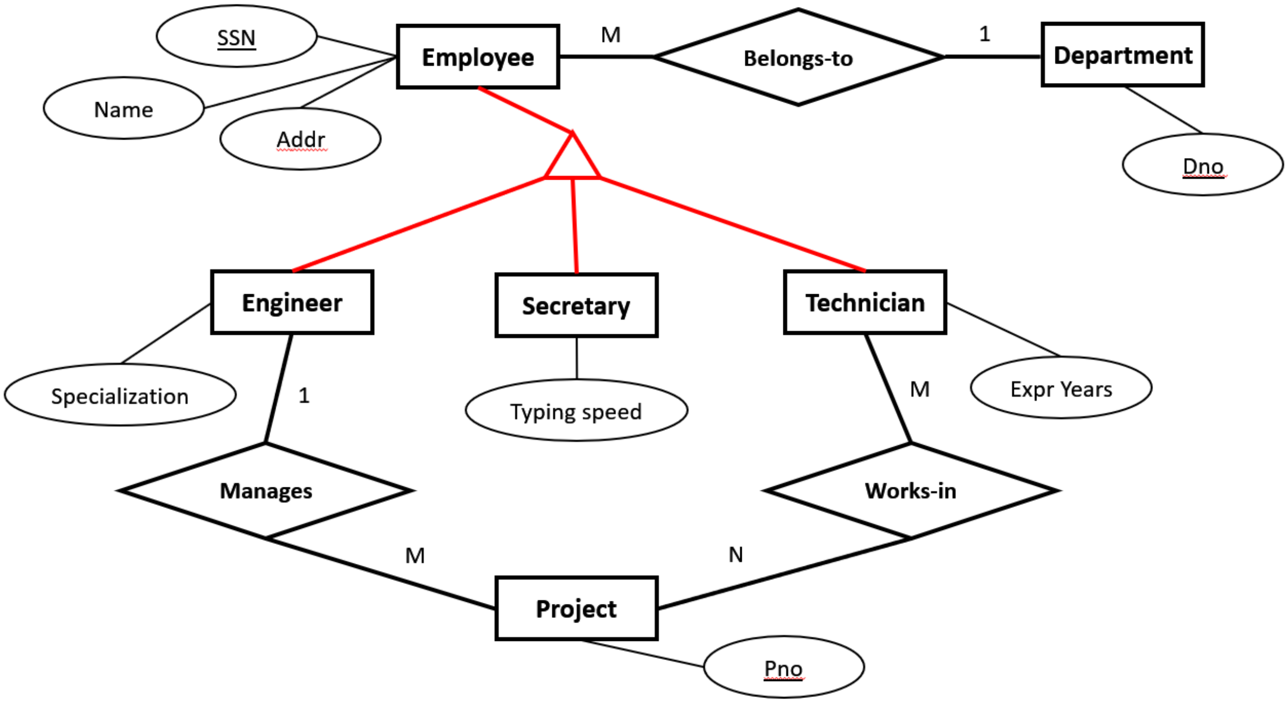
Example:

Assume you are designing a database for a company with the following partial requirements:

- ❑ ALL employees has ssn, name, addr,…..etc. and belongs to a department
- ❑ An engineer is an employee with a specialization and manages company projects
- ❑ Other types of employees are secretary with a typing speed and technician who works in projects

Here we have three types of employees with some common attributes and relationships but each type also has its own different attributes and/or relationships

To represent such a case in ER diagram, make **employee a super entity** and **derive three sub entities from it** as shown below:



Here entity **employee** is a **super entity** as it has attributes and relationships common to Engineer, Secretary, and Technician entities. Each sub entity **inherits** both the attributes and relationships from its super entity. In addition, it has the **same key** (as will be shown in the schema below)

The super entity is connected to its sub entities through a **triangle** as shown in the figure. Please use the same notation in your project ER.

Mapping super entity and sub entity into schema

Super-sub entity relation may be considered as a special type of one-to-one relationship as each sub-entity is related to exactly one super

So, the above ER mapping to a schema is:

Department	Dno	*…rest of Dept table*			
Project	Pno	*…rest of Project table*	**Mngr_ssn**		
Employee	SSN	Name	Addr	**Dno**	*….rest of Emp table*
Engineer	**SSN**	Specialization	*….rest of Eng table*		
Secretary	**SSN**	Typing Speed	*….rest of Sec. table*		

Technician

SSN	Expr Years	*….rest of Tech. table*

Works_IN

Tssn	**Pno**

Foreign keys: (shown in bold)

- Employee (Dno) references Department (Dno)
- Project (Mngr_ssn) references **Engineer** (SSN)
- Works_IN (Tssn) references **Technician** (SSN)
- Works_IN (Pno) references Project (Pno)

- Engineer (SSN) references Empolyee (SSN)
- Secretary (SSN) references Empolyee (SSN)
- Technician (SSN) references Empolyee (SSN)

Note:

The last three keys are foreign keys from the super-table and at the same time the primary keys in the sub-tables. This is one possible representation of one-to-one relationships. Other possible representations are applicable too.

Appendix B

Some Ideas Needed in Our Department

In this appendix, we stated some ideas that we need in our computer department. You can choose from them or get any other innovative idea. In this section, we will give you a brief overview of the idea and you should go to the involved users and talk with them to take their requirements and design your application accordingly. Then, you should write the project proposal that contains the details and the scope you will handle in your project.

Idea 1: Laboratory Tracking System

We have many laboratories in our department. Each laboratory contains a number of PCs. Each PC has some installed software programs and hardware components. We need a tracking system that tracks the hardware and software status of each PC in each laboratory.

Your application should enable the user to view the software and hardware status of each laboratory with a useful statistical summary. The user should also be able to report a PC problem to the lab operators with all the needed details. Then, the lab operators should respond to the request with a status like: pending, resolved or stuck, …etc. or escalate the problem to the department administrator.

Concerning software tracking, each PC has software programs already installed on it and some software programs needed to be installed. In addition, some PCs have problems like: no operating system, low disk space, or being so slow, …etc. The user should be able to request a software to be installed and the lab operators should respond accordingly.

Concerning the hardware tracking, each PC should have a working screen, mouse, keyboard, an internet cable, working sockets like the USB socket, enough power sockets, …etc. If you will include the logic labs to your scope, you should also track the available kits, number of components, …etc.

Idea 2: Projects Tracking System

As you know, computer department is a practical department. Almost every course in the department has some assignments and a project the students should make to pass the course. It is very useful if we can store all the student projects, assignments and any kind of student work for every course. This can be useful in many ways:

- To serve as a portfolio for each student, so the student can view his progress in the department and make, with the help of the teaching staff, a development plan for himself with the lead and guidance of professors and TAs.
- For the TAs, to choose some samples to view to other students even the demos without the complete source codes.
- For the TAs, to be able to check against plagiarism.

Each student work item (assignment, project, …etc.) should be accessible for the author (student or team) in addition to all the TAs and professors. However, it must be inaccessible for other students to avoid plagiarism. Some TAs may want to publish a sample projects themselves or change the accessibility of some projects to public to be project samples for new students. The information of the authors of the project should be saved in addition to the feedback of the TAs and professors on these projects and the rating of the project. Some projects are multi-phase projects. Your application may handle this too. Your application may also include the graduation projects.