

Project Management System
Final Project - Part 1
CIS 551

Sohan Akolkar

February 21, 2011

Contents

1	Summary	3
2	Logical Design	4
3	Physical Database Design	6
4	Physical Application Design	8

1 Summary

Handling concurrent projects is a huge hassle in all companies, no matter whether the company is big or small. To convolute the problem further, a single company works with many companies for collaborative projects. Therefore, every company needs a project management system which can take of storing and retrieving relevant information regarding the on-going and completed projects. The Project Management System focuses on implementing such a system but on a very basic level. The mini world will include entities such as company, its projects, users, department, projects tasks and calendar. This system includes company employees called users, who can access the project information like start date, due date, of which company, etc. So, this system emphasizes more on the interaction between the user and the other entities like company, project and department. The projects can be assigned tasks for better scheduling of work among the employees and important events can be viewed on a calendar specific to every project. The maintaining of tasks and events for the company projects is an important and unique feature of the project.

Implementation of the project will be done in simple four steps which are given as follows:

1. **Logical Design:** For any database management system, there has to be a logical schema or representation of the mini world and the relationships of all the entities. Therefore, logical design is the ideal step to start begin visualizing the system and then implementing it.
2. **Physical Database Design:** This will be the actual designing of the database required for the system. This will contain design and creation of the tables in the database.
3. **Designing the front end:** Designing and implementing the user interface of the system which will be done using PHP.
4. **Integrating back-end and front-end:** The final step will be to connect front end web pages to the required tables in the database.

2 Logical Design

As mentioned earlier the logical design gives us the various entities in the mini-world and their relationships with each other. In this system we have six entities and eight relationships between these entities, which are explained as follows:

1. COMPANY : The company entity indicates the company which handles the projects. Users or employees work in this company. A company has different departments where the users' work on the projects.
2. PROJECT: Project is handled by a company and the user works on the project. Many projects can be handled by a single company simultaneously. Similarly, many users can work on the same project at the same time.
3. TASK : Every project has one or more tasks associated with it. Tasks can be related to each other, or in other words, one task can be dependent on other task. But for simplicity this system does not show such kind of dependency.
4. CALENDAR: The calendar is nothing but a collection of events related to the projects of a particular company. The calendar will give the user a picture of what events are going to occur in the current and coming weeks.
5. USER: User is the one who will interact with the system the directly. As an employee, the user works for one or more projects and part of a department. Of course, user also belongs to a company which gives him/her projects to work on.
6. DEPARTMENT: The department is a part of the company. Many departments when considered together make up the whole company. Department controls multiple projects and has one or many users working on the projects.

The above description of the entity and their relations will be more clear by looking at the following ER Diagram (Figure 1).

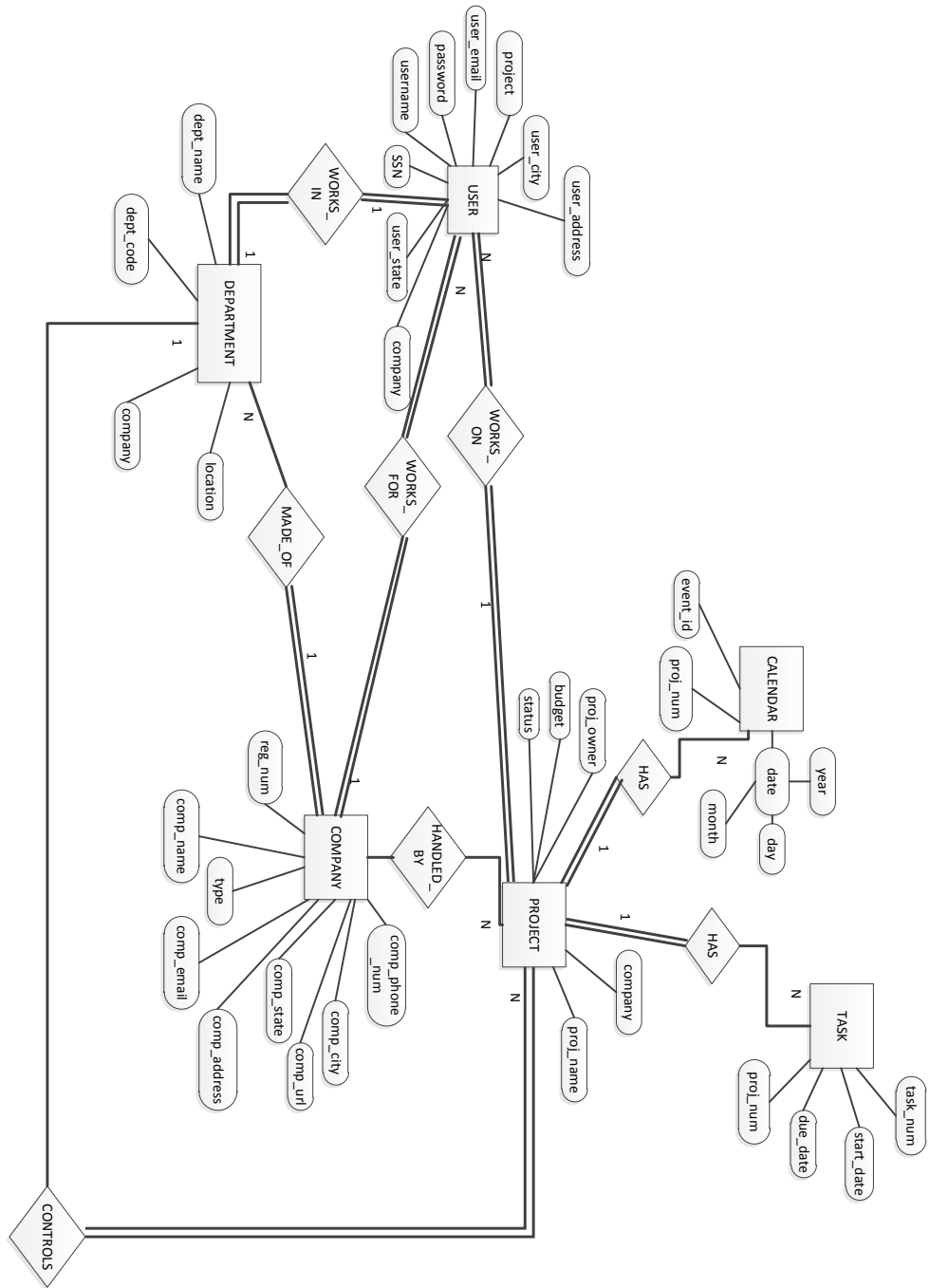


Figure 1: ER Diagram of the Project Management System

3 Physical Database Design

Based upon the logical design and ER Diagram, the relational schema was designed. The relationship between attributes of the each table are given below in relational schema.

1. COMPANY: {reg_num, comp_name, comp_address, comp_phone, comp_email, comp_type, comp_url}
Primary Key: reg_num
Foreign Key: reg_num is foreign key to PROJECT (reg_num), DEPARTMENT(reg_num) and USER(proj_num)
2. PROJECT: {proj_num, reg_num, proj_name, budget, status, proj_url, task_num, event_id, proj }
Primary Key: proj_num
Foreign Key: proj_num is foreign key to USER(proj_num), TASK(proj_num) and CALENDAR(proj_num)
3. CALENDAR: {event_id, proj_num, date, month, day, year}
Primary Key: event_id, proj
4. DEPARTMENT: {dept_code, ssn , dept_name, reg_num}
Primary Key: dept_code
Foreign Key: dept_code is foreign key to USER(dept_code)
5. USER: {username, password, ssn, proj_num, reg_num, dept_code, user_address, user_phone_num, user_email }
Primary Key: ssn and password
Foreign Key: ssn is foreign key to DEPARTMENT(ssn)
6. TASK :{task_num, proj_num, start_date, due_date, description}
Primary Key: task_num

Normalization issues were not very intricate as the tables and the relationships were straightforward. Couple of general issues include deleting ADMIN entity because it was not feasible to make one more entity when there was a generic USER entity. Therefore, for the simplicity of the project, only one generic entity named USER would suffice. Similarly, the MAINTAINS relation between ADMIN with TASK and CALENDAR was deleted.

After designing the logical and relational schema, we can map the two in a mapping diagram as shown in Figure 2:

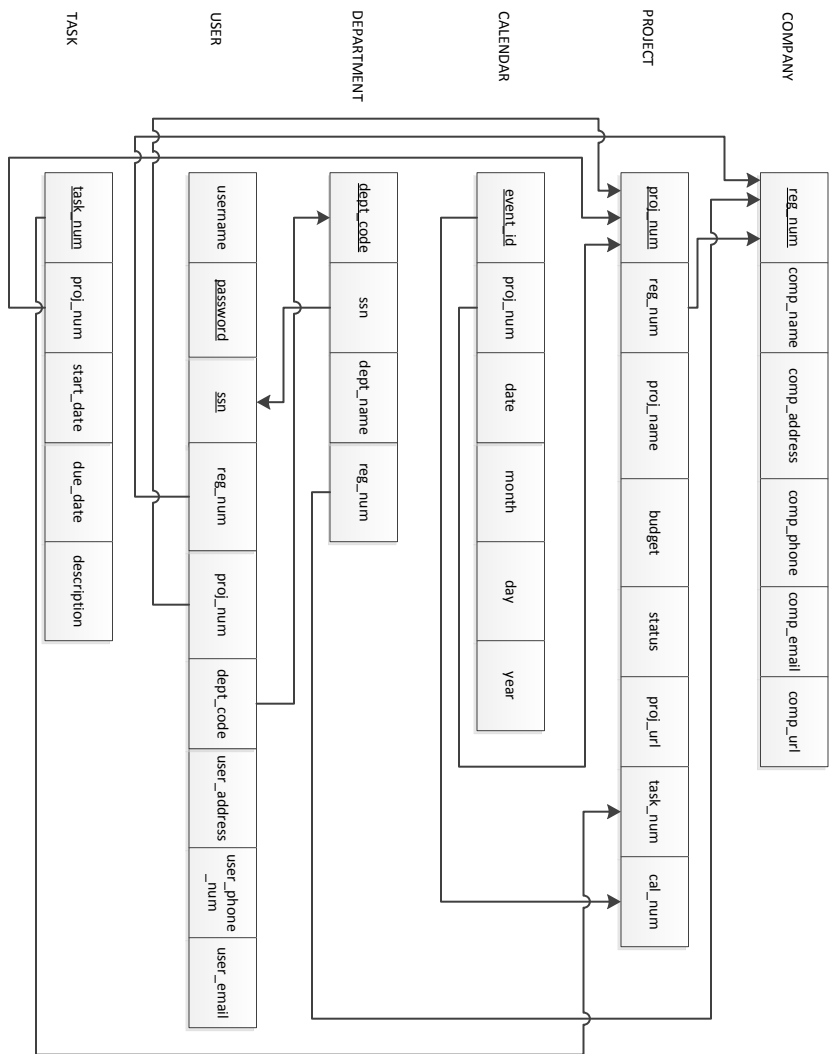


Figure 2: Mapping Diagram of the Project Management System

4 Physical Application Design

One thing worth mentioning is that, the user¹ can only retrieve the information using the web interface. Henceforth, affected tables will mean the tables accessed by the user. The desired applications and the corresponding tables affected for the Project Management System are as below:

1. **Authentication:** The authentication will be as same as that of any other web interface. The user (here it is an employee) will login using unique username and password. The password will have restriction of 10 characters long. For the sake of better testing of the system the username will be hard coded to *root* and the password will be *root*. The table affected will be the USER table, to access the username and password.
2. **Displaying names of users working on projects:** The system will have an application to query the database and display the names and their information of the users working on one or more projects. Tables affected are USER and PROJECT.
3. **Displaying names of users working in company:** Information of the users working in their respective companies will be queried and populated on the web interface. Tables affected are USER and COMPANY.
4. **Displaying names of users working for departments:** The system will have an application to query the database and display the names and their information of the users working on one or more projects. Tables affected are USER and DEPARTMENT.
5. **Displaying projects handled by a company:** The user will want to find out the projects currently on-going and completed in a company. Tables affected are COMPANY and PROJECT.
6. **Displaying departments in a company:** The system will also display the composition of the company in form of how many departments are included in a company. Tables affected are COMPANY and DEPARTMENT.
7. **Displaying Calendar for project:** Every project has a calendar representing the events scheduled for the projects. The calendar will be simple grid displaying the month, day, year and the events. Tables affected are PROJECT and CALENDAR.
8. **Displaying Task for project:** Like calendar, task is also associated with every project and describes the up-coming tasks. The *description* attribute gives the brief definition of the task.

¹The user will be given privileges only to access the tables using the *guest* account