Phase V – Overview

The goal of this phase is to develop queries that support a potential iPlanner application. To get you started, the solution for Phase IV a new ERD, as well as Database Schema (attributes, Key/FK relationships, and additional data) will be provided. This solution will serve as the starting point for Phase V.

Do not try to merge this starting point for Phase V with your own Phase IV. Simply create a new database called “iplanner_5” and copy and paste the schema provided for this phase. (you will get this schema once phase IV closed in canvas.)

Database Queries:

- **Review**: Select-From-Where, DISTINCT, IN, Join, Left Join, Group by, Having, Order By, Aggregate functions (Add, Count, Avg, Min, Max, Concat, etc.) encryption functions (MD5, SHA1, SHA2, AES_ENCRYPT, AES_DECRYPT, etc.), Sub-string Comparison (Like, %, _ etc.), other functions (ISNULL, etc.) (some of these commands you will find in our text book, others you should look up online or in the reference books I provided in the syllabus http://cs.iusb.edu/~hhakimza/C442/Syllabus_Spring_2018.pdf)

1) **Implement at least 8 SQL queries (see appendix A).**

2) **Review the Queries provided in appendix A.** For each query, develop an **execution plan**, identifying which tables are needed in order to arrive at the solution, and how you will connect those tables.

3) **Make sure that your SQL queries are based on the information given in the “Question”,** not information you manually look up or know from memory. For example; if the query indicates “Computer and Information Sciences” organization, do not manually look up the record, get the key “CSCI”, and then use the key in your query. Instead, use SQL to get the query.

4) Ensure that there is sufficient data in the tables so that when you execute a query, at least one record is displayed. No query should return an empty set. (Some queries in Appendix A will ask for specific number of records > 1).

5) **Debugging your queries**: After executing each query go back and look at the data in the table(s) to see if the correct results has been displayed. To further make sure the query is working properly, add a record to the “appropriate” table or tables to see if the additional information is being picked up by your query. You can also add records that should not be picked up by your query, and then test the query again to make sure that data is not picked up! Be especially careful about spurious results. In other words, make sure that your query has
proper join conditions. Typically if the join condition is not correct, you end-up getting a cartesian product instead of a join and therefore you get a lot more results that are erroneous.)

6) Note that you must be able to explain each of the queries when you present your results. After submitting this phase, each person will schedule an appointment to present and discuss their queries. (so, be ready for that). We will discuss this more in class.

What to hand in?

A binder with the following:

- A copy of the ERD
- A copy of the Decomposition Diagram
- For each of the 8 queries provide the following:
  1. Query description (see appendix A)
  2. SQL Query
  3. Result of the SQL query
  4. Explanation of what records had to be added to the database (the actual records, as well as which tables, and why) in order to satisfy the query.
  5. What did you do to audit the query? To make sure it works for the data you have in the system.
- A dump of the entire schema
Appendix A: Queries

Q1) **Show the set of known projects for the CSCI organization that occur once per semester (more precisely: those have a recurring frequency of "Single Occurrence Per Semester")**. Display the Organization ID, name, the administrator’s ID, the project’s Title, Description, and Priority ID. Also, to make sure you have the right records make sure you display the field that show the recurring frequency (not just ID’s).

**IN CLASS DEMO**

Q2) **What is the status of project 4?** (study stored procedures and make this query a stored procedure!) (also study the IF statement in MySQL) (Make sure to pass the “project number” as a parameter to stored procedure.) Display the project ID, name, description, start date, end date, deadline, project manager’s id, and the tasks that need to be implemented for this project including (Task ID, name, description, start date, end date, deadline, task leader ID, and if the project is past due, and if any of its tasks are past due)

**Note for now we will define a project is past due if:**

\[
\text{IF}(\text{project.Deadline} < \text{project.EndDate}) \text{ then } \\
\quad \quad \text{"Project Past Deadline"} \\
\text{else} \\
\quad \quad \text{"Project On Time"}
\]

**Similarly, we define a task as past due if:**

\[
\text{IF}(\text{project.Deadline} < \text{task.Deadline}) \text{ then } \\
\quad \quad \text{"Task Deadline Past Project Deadline"} \\
\text{else} \\
\quad \quad \text{"On Time"}
\]

**Review the IF statement as well as the Stored procedures in MySQL.**

Q4) **What resources are needed for project named 'CS Search and Screen'?** Display the ProjectID, project title, PMID, TaskID, Title, Description, and the ResourceID, Name, and Description.

Q5) **What are project deliverables for projects that are managed by Project Manager with the last name of 'Hakimzadeh'?** List the project manager's ID, Name, ProjectID and title, as well as the project deliverable title, description and the due date status for each deliverable. Make sure there are at least 3 records in the output.

Q6) **List of employees (last name, first name, email), who have a 'Database Design - Level 2' as well as 'Web Design' skills**. Show their skills. (Ensure there are at least one employee)

Q7) **Who oversees the project titled "2018 Renovations"?** (the administrative staff for the project, including the owner, PM, task leaders). List their UserID, OrgID, Last name, first name, Phone, Email, and their Role (Admin, PM, Task Leader) Note that there is no "specific" field in the database which talks about peoples “roles”. This information must be implicitly derived from your queries (and where you obtain the data for each group of employees). This is not very difficult. (HINT: Try the following SQL statement:
SELECT *, "Team Member" as Role
FROM employee

Q9) **What is the availability schedule (when they are Definitely Available) for employees who work for project PTR for faculty XXX?** List the last name, first name, of the employee, project name (title, etc) and the availability schedule (days, start and end time) Make sure there are at least 3 records displayed.

Q10) **Which projects cross organizational boundaries?** (projects that have multiple owners from different organizations, or have team-members belonging to multiple organizations)

Q11) **Which employees are currently “not” assigned to a project?** List the employeeID, Last Name, First Name, Job Title, Organization Name. HINT: Look up the ISNULL() function in MySQL

Q12) **List the projects, their tasks, and the resources needed for each task.** Show the project ID, title, description, as well as the task id and name and description, followed by the task leader’s id and name, finally include the resource id, name and description.

Q13) **Which Projects require external contractors?** (show the projects, their requisitions, quotes, contracts, and the company name and rep name of the contractor)

Q18) **Which organizations have the largest number of projects with “High Priority” priority?** Display the OrgID, Name, and the number of "High Priority" projects. Show the following 3 columns: OrgID, name, and the number of (count of) high priority projects they have in the system.

Q19) **Display the checklist for the "CS Search and Screen" Project?** Show the ProjectID, name, description, sequence number (Ordering which should be followed), and checklist title and description. In addition, display the name and email of the project manager and the project owner. Make sure the checklist is sorted on the proper order.

Q30) **What resources are needed for any high priority projects?** Show the Project ID, priority, title, description, deadline, as well task title, description, deadline, task Leader’s ID, last name and email, followed by the resource name, description and resource type name.

Q31) **What needed skills are “missing” for completing the project “Online Informatics Degree Proposal”.** (hint: think task and the skill they need, then think project workgroups and the skills those employees have! Then think LEFT JOIN). The result of the query should simply display the Skills Needed, and the `Skills Possessed`.

Q32) **Show all Project Managers (name and email) and their “High Priority” projects (id, title, and due date) that have either past their due date, or they are within 30 days of their due date.** (make sure there is sufficient data in the database to return at least one project in each category.

Q40) **Your own query that uses more than 4 tables (one of which you have not used in any other query) and requires sorting and aggregating.** Provide a description of what you are trying to do, then show the query, and its results.
Q41) **Research Temporary tables in MySQL.** Describe your understanding of Temporary tables in MySQL and then use one of the above queries to create a temporary table. Display the schema of the temporary table, then display the data in that table.

Q42) **Research Views in MySQL.** Describe the purpose and your understanding of Views and how they are different than tables in MySQL and then use one of the above queries to create a view (give a name to the view such as View12, which means the view that captures query 12. Display the schema of the view, then display the data in that view. Now change one of the base tables associated with the View and reprint the data in the view to see if the change in the base table is reflected in the view.
Appendix – D (iPlanner Schema for Phase V)

Check our web site. The Schema will be provided online

http://cs.iusb.edu/~hhakimza/C442/