Phase IV - Overview

Following our initial design of the iPlanner system (Solution ERD and EAT provided as part of assignment 3), our goal in phase 4 is to transform the latest version of our ERD (see appendix A) into a relational database system (using MySQL database). Once again you will be provided with a starting point for this phase, and you will be asked to use MySQL’s DDL capabilities to create the iPlanner schema. In this phase you are also provided with some data to populate the initial database.

Once you create the schema, you should review each table in MySQL and ensure the primary key is correctly defined. If all the primary keys (including those with composite primary keys) are ok, you can move to the next step. Make sure that all tables use the INNODB engine. (google this term and see why we want to use this engine). If you find any tables that use MyISAM or other engines, switch it back to InnoDB (See the “Operations” tab under phpmyadmin).

Next, take the data that has been provided and populate the database. These should load without too much trouble. Load the data in steps (one table at a time). This allows you to identify and resolve any issues more accurately. After every table is loaded use the phpmyadmin GUI interface or the SQL (SELECT statement) to display the content of the table and verify that it has been correctly populated.

The next step is to establish the foreign keys. This is a very important step. Do this systematically so you do not miss any tables. Make sure all the FK’s are properly established using the ALTER table command. Note that you may have to establish an index on your FK fields before MySQL allows you to make the FK connections. Also, note that when you try to establish the FK’s, you may encounter an error. These errors typically have to do with referential integrity issues. At this point you need to put on your debugging hat and trace each of the FK’s to their tables to see what the source of the error might be, and then resolve the problem.

Once the database schema is created, and the PK’s, indexes, and FK’s are established, and the initial data is populated, you are ready to do some of your own schema additions and modifications. So, now that you understand the general overview, let us discuss the details.

Step - 1) Database Creation: (DDL)

- Review: Create Schema, Create Table, Drop Schema, Drop Table, Alter Table, Create Domain
• Using SQL, create the iPlanner schema (provided as starting point in Appendix B). Review and understand how entities and their relationships in an ERD are transformed into the relational database (for example how PK keys are established).
• Note how certain attributes are set to be unique (alternate key), or NOT NULL (to establish entity integrity or to ensure data is provided for the field),
• Note how certain attributes could be set to have default values,
• Note how and why certain attributes have an index established on them. (This is to allow them to connect to other tables. FK \rightarrow PK connections).
• Note how certain relationships (1-1 and 1-M) end up becoming FK’s, and others (M-M) become associative entities (a table of their own, borrowing portion of their key from each contributing table).
• Make sure to create each table separately and validate the schema of the table by visually reviewing it after each table is created.
• If you wish to change the schema (perhaps to accommodate some new feature, etc.) print the new ER and clearly point out the proposed modifications. However, your modifications should be proposed to me to get approval.

Step - 2) Initial Database Population:

• **Review:** Insert INTO, Delete FROM, Update-Set-Where

  1) Using SQL insert statements (provided as part of the initial starting point, Appendix C) populate the tables. Note some tables do not have any data at this point, however, later in STEP – 4 you will need to make sure all tables have stufliest amount of data to be able answer the desired queries. (you can shoot for 3 or more records per table)
  2) Load the data for each table separately. Then validate that the data has be correctly loaded. Use the phpmyadmin GUI interface or the SQL (SELECT statement) to display the content of the table. Don't attempt to load all the data at the same time. This allows you to encounter and fix any errors early on. (Note: this approach is equivalent to compiling your code after every small modification vs. waiting to compile your code after typing 100 lines of code, then trying to compile and run your code and hope that it works)
  3) **Take a screen shot of each table structure** after the data for each table is populated. (Yes, you will need a binder for this phase) (This should be about 38 or so tables)

Step - 3) Establish the FK’s:

• **Review:** Alter table, Drop Table, command

  1) Using SQL Alter Table statement and later using the phpmyadmin’s GUI interface establish the FK connections between the tables. Do this systematically (either by following the ER and checking off entities as you go, or alphabetically (based on table names), click on each table and then make sure it’s FK’s are established properly. (you will need a copy of your ERD handy to refer to it.)
  2) If you encounter any errors when you are trying to establish the FK’s, you need to make sure track the problem to its source. In other words, finding out why the database is rejecting your attempt to establish the FK relationship. In most cases, this is because there is some inconsistency in the database.
  • After all the FK’s are successfully established (there should be about 40+ FK relationships), Using the Export Capability of MySQL:
    a. Print the structure of all the tables. This will be similar to what I have provided you in appendix B and C, followed by all the ALTER TABLE commands that establishes the FK’s.

• Ask me if you are not sure how do present your work for this phase.(This above should go to your Binder as well.

Step - 4) Adding More Data:

• Now we want to add some more data in the database.
• When entering your own data do not use too much abbreviation since it makes it difficult to evaluate the result of your queries.) (Note: I realize that it takes longer to do this with SQL than it does using the GUI provided by “phpMyAdmin”. So, insert one or two of the records using SQL and the rest using the interface. Make sure to show (provide screen shots) the SQL statements which were used to populate the tables. (at least for 2 distinct insert statements in at least 2 different tables.) (Use a highlighter to show the inserted records)
• Add a new employee named Murli Nair to the employee table. Dr. Nair works for the BIOL organization. His job title is Bioinformatics Specialist.
• Look up Dr. Adaikkalava’s schedule for teaching and office hours this semester and insert those hours as “Unavailable” in the employee schedule table.
• Look up the same information for Dr. Vrajitoru and insert her “unavailable” schedule to the employee schedule. (also note that she is not currently an employee)
• Add your own availability and unavailability schedule to the employee schedule. Note that you should already be in the employee table. If you are not, fix that first.
• Project manager “hhakimza” has selected you and 3 other employee of your choice to be part of a project-workgroup, that will work on the high priority project called “CS Fall 2018 Tutor Hiring”. Make sure the database reflects this information.
• Insert the following tasks for project 3:
  a. “Initial Announcement”, “Make announcements to the CS and INFO students”, april 1, 2018, april 30, 2018. The deadline should be the same as end date, and the task leader is of course “YOU”.
  b. “Review”, “Review applications”, May 1, 2018, May 7, 2018 and Leader is you or one of the other members working on this project (your choice).
  c. “Hire”, “Notify Employee of hire”, may 8, 2018, may 14, 2018, leader is YOU.

The hiring of tutors typically happens once per semester.

Step – 5) Update operations:
• Update employees with the UserID of “hhakimza” and YOUR own UserID and change their organization to “INFO”. (Note that if this operation fails, you need to find out why and fix it.)
• Update YOUR own address in the database (User_profile table)

Step – 6) Delete Operations:
• Delete an employee of your choice.

Step – 7) Drop Table
• Drop the zz_project_type table.

Step – 8) Create Table.
• Create a table called activity_log with the following attributes:
  User-id, role, timedate, IP_address, note
• Make sure the data type for the attributes are compatible with the data that will be placed in the table.
• The key for the table is composite UserID and TimeDate

Phase V – What to Hand In
• A binder as well as a PDF document including the following:
  • Cover page (title, name, course # and name(s), assignment #, date)
  • Screen shot, printouts, before and after screen shots (follow the steps above) also include tabs in the binder so it is easy to navigate it.
  • Include a copy of the ER or the schema produced. (Original or one modified by you)
  • Printed Structure of all relations in SQL format
Appendix – A (Entity Relationship Diagram)

Intelligent Planning - iPlanner

Include a detailed description of the Entity Relationship Diagram (ERD) depicted on the page, focusing on the relationships and entities involved.
-- Database: `iPlanner - Schema`
--
-- -----------------------------------------------
--
-- Table structure for table `admin`
--
CREATE TABLE IF NOT EXISTS `admin` (  
`AdminID` varchar(64) NOT NULL COMMENT 'PK and FK to UserID',  
`OrgID` varchar(64) NOT NULL COMMENT 'FK to Organization',  
`SupervisedBy` varchar(64) NOT NULL COMMENT 'FK to AdminID',  
`ManagesOrg` int(11) NOT NULL COMMENT 'FK to OrgID',  
`ApprovedBy` varchar(64) NOT NULL COMMENT 'FK to SU',  
PRIMARY KEY (`AdminID`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
-- -----------------------------------------------
--
-- Table structure for table `availability_status`
--
CREATE TABLE IF NOT EXISTS `availability_status` (  
`AvailabilityStatusID` varchar(32) NOT NULL,  
`Description` varchar(128) NOT NULL,  
PRIMARY KEY (`AvailabilityStatusID`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
-- -----------------------------------------------
--
-- Table structure for table `contract`
--
CREATE TABLE IF NOT EXISTS `contract` (  
`ContractID` int(11) NOT NULL AUTO_INCREMENT,  
`Title` varchar(256) NOT NULL,  
`Description` text NOT NULL,  
`DateIssued` date NOT NULL,  
`DateExpires` date NOT NULL,  
`ContractorID` int(11) NOT NULL COMMENT 'FK to ExternalContractor',  
`ProjectID` int(11) NOT NULL COMMENT 'FK Project.ProjectID',  
PRIMARY KEY (`ContractID`),  
KEY `ProjectID` (`ProjectID`),  
KEY `ContractorID` (`ContractorID`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=3;
-- -----------------------------------------------
--
-- Table structure for table `employee`
```sql
-- CREATE TABLE IF NOT EXISTS `employee` (  
  `EmployeeID` varchar(64) NOT NULL COMMENT 'FK to User',  
  `Title` varchar(128) NOT NULL,  
  `WorksForOrgID` varchar(32) NOT NULL COMMENT 'FK to Organization',  
  PRIMARY KEY (`EmployeeID`),  
  KEY `WorksForOrgID` (`WorksForOrgID`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -- Table structure for table `employee_schedule`

-- CREATE TABLE IF NOT EXISTS `employee_schedule` (  
  `EmployeeID` varchar(64) NOT NULL COMMENT 'PK (Composite)',  
  `DayOfWeek` varchar(16) NOT NULL COMMENT 'PK (Composite), FK to Weekdays',  
  `FromTime` time NOT NULL,  
  `ToTime` time NOT NULL,  
  `AvailabilityStatusID` varchar(32) NOT NULL DEFAULT 'UNAVAILABLE' COMMENT 'AVAILABLE / UNAVAILABLE',  
  PRIMARY KEY (`EmployeeID`, `DayOfWeek`, `FromTime`),  
  KEY `AvailabilityStatusID` (`AvailabilityStatusID`),  
  KEY `DayOfWeek` (`DayOfWeek`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -- Table structure for table `employee_specialties`

-- CREATE TABLE IF NOT EXISTS `employee_specialties` (  
  `EmployeeID` varchar(64) NOT NULL COMMENT 'PK (Composite), FK to Employee',  
  `SkillID` int(11) NOT NULL COMMENT 'PK (Composite), FK Skills',  
  `DateSkillLearned` date NOT NULL,  
  PRIMARY KEY (`EmployeeID`, `SkillID`),  
  KEY `SkillID` (`SkillID`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -- Table structure for table `external_contractor`

-- CREATE TABLE IF NOT EXISTS `external_contractor` (  
  `ContractorID` int(11) NOT NULL AUTO_INCREMENT COMMENT 'PK',  
  `CompanyName` varchar(64) NOT NULL,  
  `Address1` varchar(128) DEFAULT NULL,  
  `Address2` varchar(128) DEFAULT NULL,  
  `City` varchar(128) DEFAULT NULL,  
  `State` varchar(32) DEFAULT NULL,  
  `Zip` varchar(32) DEFAULT NULL,  
  `Phone` varchar(32) DEFAULT NULL,  
  PRIMARY KEY (`ContractorID`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```
CREATE TABLE IF NOT EXISTS `organization` (  `OrgID` varchar(512) NOT NULL COMMENT 'PK',  `Name` varchar(64) NOT NULL,  `Address1` varchar(128) NOT NULL,  `Address2` varchar(128) NOT NULL,  `City` varchar(128) NOT NULL,  `State` varchar(32) NOT NULL,  `Zip` varchar(32) NOT NULL,  `Phone` varchar(32) NOT NULL,  `Email` varchar(128) NOT NULL,  `WebSite` varchar(128) NOT NULL,  PRIMARY KEY (`OrgID`) ) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `priority` (  `PriorityID` varchar(32) NOT NULL,  `Title` varchar(256) NOT NULL,  `Description` text NOT NULL,  PRIMARY KEY (`PriorityID`) ) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `project` (  `ProjectID` int(11) NOT NULL AUTO_INCREMENT,  `Title` varchar(128) NOT NULL,  `Description` text NOT NULL,  `StartDate` date NOT NULL,  PRIMARY KEY (`ProjectID`) ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
`EndDate` date NOT NULL COMMENT 'Projected End Date',
`Deadline` date NOT NULL COMMENT 'Hard Deadline (if any)',
`PMID` varchar(64) NOT NULL COMMENT 'FK to Project Manager"s ID',
`FrequencyID` int(11) NOT NULL COMMENT 'FK to Recuring Frequency',
`PriorityID` varchar(32) NOT NULL COMMENT 'FK to Priority',
`ProjectStatus` varchar(32) NOT NULL DEFAULT 'CHECK PROJECT DELIVERABLE STATUS' COMMENT 'Project Status can be derived from Project Deliverable Status',
PRIMARY KEY (`ProjectID`),
KEY `PMID` (`PMID`),
KEY `FrequencyID` (`FrequencyID`),
KEY `PriorityID` (`PriorityID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=9 ;

-- Table structure for table `project_checklist`

CREATE TABLE IF NOT EXISTS `project_checklist` (  
`ProjectID` int(11) NOT NULL COMMENT 'PK composite, FK to Project',
`OrderOfAppearance` int(11) NOT NULL COMMENT 'PK composite',
`Title` varchar(128) NOT NULL,
`Description` text NOT NULL,
PRIMARY KEY (`ProjectID`,`OrderOfAppearance`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Table structure for table `project_contract`

CREATE TABLE IF NOT EXISTS `project_contract` (  
`ProjectID` int(11) NOT NULL COMMENT 'FK to Project',
`ContractID` int(11) NOT NULL COMMENT 'FK to Contract',
`ContractorID` int(11) NOT NULL COMMENT 'FK to external_contractor',
`QuoteID` int(11) NOT NULL COMMENT 'FK to Quote',
`RequisitionID` int(11) NOT NULL COMMENT 'FK to Requisition',
`DateSigned` date NOT NULL,
`DateDue` date NOT NULL,
PRIMARY KEY (`ProjectID`,`ContractID`,`ContractorID`,`QuoteID`,`RequisitionID`),
KEY `ContractID` (`ContractID`),
KEY `ContractorID` (`ContractorID`),
KEY `QuoteID` (`QuoteID`),
KEY `RequisitionID` (`RequisitionID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Table structure for table `project_deliverables`
CREATE TABLE IF NOT EXISTS `project_deliverables` (
`ProjectID` int(11) NOT NULL COMMENT 'PK composite, FK to Project',
`OrderOfAppearance` int(11) NOT NULL COMMENT 'PK composite',
`Title` varchar(128) NOT NULL,
`Description` text NOT NULL,
`Recipients` varchar(64) DEFAULT NULL,
`ProjectStatusID` int(11) NOT NULL COMMENT 'FK to Status',
PRIMARY KEY (`ProjectID`, `OrderOfAppearance`),
KEY `ProjectStatusID` (`ProjectStatusID`) ) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -----------------------------

-- Table structure for table `project_goal`
--

CREATE TABLE IF NOT EXISTS `project_goal` (
`ProjectID` int(11) NOT NULL COMMENT 'PK composite, FK to Project',
`OrderOfAppearance` int(11) NOT NULL COMMENT 'PK composite',
`Title` varchar(128) NOT NULL,
`Description` text NOT NULL,
PRIMARY KEY (`ProjectID`, `OrderOfAppearance`) ) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -----------------------------

-- Table structure for table `project_manager`
--

CREATE TABLE IF NOT EXISTS `project_manager` (
`PMID` varchar(64) NOT NULL COMMENT 'PK, FK to UserID',
`DesignatedBy` varchar(64) NOT NULL COMMENT 'FK to user_admin',
`DateDesignated` date NOT NULL,
PRIMARY KEY (`PMID`),
KEY `DesignatedBy` (`DesignatedBy`) ) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -----------------------------

-- Table structure for table `project_membership_requirement`
--
CREATE TABLE IF NOT EXISTS `project_membership_requirement` (  `ProjectMembershipReqID` int(11) NOT NULL AUTO_INCREMENT COMMENT 'PK',  `ProjectID` int(11) NOT NULL COMMENT 'FK to Project',  `RequirementDescription` varchar(512) NOT NULL,  PRIMARY KEY (`ProjectMembershipReqID`),  KEY `ProjectID` (`ProjectID`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;

-- -----------------------------------------------------------------------------

--

-- Table structure for table `project_ownership`

--

CREATE TABLE IF NOT EXISTS `project_ownership` (  `AdminID` varchar(64) NOT NULL COMMENT 'Composite PK, FK to Admin',  `ProjectID` int(11) NOT NULL COMMENT 'Composite PK, FK to Project',  `DateInitiated` date NOT NULL,  PRIMARY KEY (`AdminID`, `ProjectID`),  KEY `ProjectID` (`ProjectID`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -----------------------------------------------------------------------------

--

-- Table structure for table `project_prereq`

--

CREATE TABLE IF NOT EXISTS `project_prereq` (  `ProjectPrereqID` int(11) NOT NULL AUTO_INCREMENT COMMENT 'PK (Composite',  `ProjectID` int(11) NOT NULL COMMENT 'PK (Composite)',  `Title` varchar(256) NOT NULL,  `Description` text NOT NULL,  PRIMARY KEY (`ProjectPrereqID`),  KEY `ProjectID` (`ProjectID`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=6 ;

-- -----------------------------------------------------------------------------

--

-- Table structure for table `project_workgroup`

--

CREATE TABLE IF NOT EXISTS `project_workgroup` (  `EmployeeID` varchar(64) NOT NULL COMMENT 'PK (Composite), FK to Employee',  `ProjectID` int(11) NOT NULL COMMENT 'PK (Composite), FK Project',  `PMID` varchar(64) NOT NULL COMMENT 'FK to Project_Manage',


`EmployeeID`, `ProjectID`),
KEY `PMID` (`PMID`),
KEY `ProjectID` (`ProjectID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Table structure for table `quote`

CREATE TABLE IF NOT EXISTS `quote` (  
`QuoteID` int(11) NOT NULL AUTO_INCREMENT COMMENT 'PK (Internal Quote ID)',  
`VendorQuoteNo` varchar(128) NOT NULL COMMENT 'Vendor Quote #',  
`Title` varchar(256) NOT NULL,  
`Description` text NOT NULL,  
`DateIssued` date NOT NULL,  
`DateExpired` date NOT NULL,  
`ExternalContractorID` int(11) NOT NULL COMMENT 'FK to ExternalContractor',  
PRIMARY KEY (`QuoteID`),  
KEY `ExternalContractorID` (`ExternalContractorID`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;

-- Table structure for table `recurring_frequency`

CREATE TABLE IF NOT EXISTS `recurring_frequency` (  
`FrequencyID` int(11) NOT NULL AUTO_INCREMENT,  
`Title` varchar(256) NOT NULL,  
`Description` text NOT NULL,  
PRIMARY KEY (`FrequencyID`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=7 ;

-- Table structure for table `requsition`

CREATE TABLE IF NOT EXISTS `requsition` (  
`RequsitionID` int(11) NOT NULL COMMENT 'PK',  
`Title` varchar(256) NOT NULL,  
`Description` text NOT NULL,  
`DateIssued` date NOT NULL,
CREATE TABLE IF NOT EXISTS `resource` (  `ResourceID` int(11) NOT NULL AUTO_INCREMENT COMMENT 'PK',  `Name` varchar(128) NOT NULL,  `Description` text NOT NULL,  `ResourceTypeID` int(11) NOT NULL COMMENT 'FK to Resource_Type',  PRIMARY KEY (`ResourceID`),  KEY `ResourceTypeID` (`ResourceTypeID`) ) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=13 ;

CREATE TABLE IF NOT EXISTS `resources_required` (  `TaskID` int(11) NOT NULL COMMENT 'FK to Taks',  `ResourceID` int(11) NOT NULL COMMENT 'FK to Resource',  PRIMARY KEY (`TaskID`),  KEY `TaskID` (`TaskID`),  KEY `ResourceID` (`ResourceID`) ) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `resource_type` (  `ResourceTypeID` int(11) NOT NULL AUTO_INCREMENT,  `Name` varchar(128) NOT NULL,  PRIMARY KEY (`ResourceTypeID`) ) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=7 ;
CREATE TABLE IF NOT EXISTS `skills` (  `SkillID` int(11) NOT NULL AUTO_INCREMENT,  `Name` varchar(128) NOT NULL,  `Description` text NOT NULL,  `SkillTypeID` int(11) NOT NULL COMMENT 'FK to Skill_type',  PRIMARY KEY (`SkillID`),  KEY `SkillTypeID` (`SkillTypeID`) ) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=21 ;

CREATE TABLE IF NOT EXISTS `skills_needed` (  `TaskID` int(11) NOT NULL COMMENT 'PK (Composite), FK to Task',  `SkillID` int(11) NOT NULL COMMENT 'PK (Composite), FK to Skill',  `MinYears` int(11) NOT NULL COMMENT 'Minimum number of years experience in this skill',  PRIMARY KEY (`TaskID`, `SkillID`),  KEY `SkillID` (`SkillID`) ) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `skill_type` (  `SkillTypeID` int(11) NOT NULL AUTO_INCREMENT,  `Title` varchar(256) NOT NULL,  `Description` text NOT NULL,  PRIMARY KEY (`SkillTypeID`) ) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=7 ;

CREATE TABLE IF NOT EXISTS `status` (  `StatusID` int(11) NOT NULL AUTO_INCREMENT,  PRIMARY KEY (`StatusID`),  KEY `StatusID` (`StatusID`) ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
CREATE TABLE IF NOT EXISTS `su` (  
`SUID` varchar(64) NOT NULL,  
`StartDate` date NOT NULL,  
PRIMARY KEY (`SUID`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `task` (  
`TaskID` int(11) NOT NULL AUTO_INCREMENT COMMENT 'PK',  
`Title` varchar(256) NOT NULL,  
`Description` text NOT NULL,  
`StartDate` date NOT NULL,  
`EndDate` date NOT NULL,  
`Deadline` date NOT NULL,  
`LeaderID` varchar(64) NOT NULL COMMENT 'FK Employee',  
`ProjectID` int(11) NOT NULL COMMENT 'FK Project',  
PRIMARY KEY (`TaskID`),  
KEY `LeaderID` (`LeaderID`),  
KEY `ProjectID` (`ProjectID`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;

CREATE TABLE IF NOT EXISTS `task_deliverables` (  
`TaskID` int(11) NOT NULL COMMENT 'PK composite, FK Task',  
`OrderOfAppearance` int(11) NOT NULL COMMENT 'PK composite',  
`Title` varchar(128) NOT NULL,  
`Description` text NOT NULL,  
`Recipients` varchar(64) DEFAULT NULL,  
`StatusID` int(11) NOT NULL COMMENT 'FK to Status.StatusID',  
PRIMARY KEY (`TaskID`, `OrderOfAppearance`),  
KEY `StatusID` (`StatusID`)
```sql
-- Table structure for table `task_workgroup`

CREATE TABLE IF NOT EXISTS `task_workgroup` (  
`EmployeeID` varchar(64) NOT NULL COMMENT 'PK (Composite), FK to Employee',  
`TaskID` int(11) NOT NULL COMMENT 'PK (Composite), FK Skills',  
PRIMARY KEY (`EmployeeID`, `TaskID`),  
KEY `TaskID` (`TaskID`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Table structure for table `user`

CREATE TABLE IF NOT EXISTS `user` (  
`UserID` varchar(64) NOT NULL,  
`password` varchar(512) NOT NULL,  
PRIMARY KEY (`UserID`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Table structure for table `user_profile`

CREATE TABLE IF NOT EXISTS `user_profile` (  
`UserID` varchar(64) NOT NULL COMMENT 'PK, and FK to User.UserID',  
`OrgID` varchar(512) NOT NULL COMMENT 'FK to Organization.OrgID',  
`FName` varchar(64) NOT NULL,  
`LName` varchar(64) NOT NULL,  
`Address1` varchar(128) NOT NULL,  
`Address2` varchar(128) NOT NULL,  
`City` varchar(128) NOT NULL,  
`State` varchar(32) NOT NULL,  
`Zip` varchar(32) NOT NULL,  
`Phone` varchar(32) NOT NULL,  
`Email` varchar(128) NOT NULL,  
PRIMARY KEY (`UserID`),  
KEY `OrgID` (`OrgID`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Table structure for table `week_days`

CREATE TABLE IF NOT EXISTS `week_days` (  
`DayOfWeek` varchar(16) NOT NULL COMMENT 'PK (MON, TUE, WED, THU, FRI, SAT, SUN)'  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```
CREATE TABLE IF NOT EXISTS `zz_project_type` (  `ProjectTypeID` int(11) NOT NULL AUTO_INCREMENT,  `Title` varchar(256) NOT NULL,  `Description` text NOT NULL,  PRIMARY KEY (`ProjectTypeID`) ) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

CREATE TABLE IF NOT EXISTS `z_queries` (  `QueryID` int(25) NOT NULL AUTO_INCREMENT,  `Recipient` varchar(25) DEFAULT NULL,  `QueryDescription` longtext,  `SQLStatement` longtext,  `HowToAudit` longtext COMMENT 'Describe how this query can be audited by the SQL programmer.',  `Audited` varchar(16) DEFAULT 'NO',  `CampusID` varchar(32) NOT NULL DEFAULT 'SB',  PRIMARY KEY (`QueryID`) ) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=22 ;
-- Database: `iPlanner`

--
-- Dumping data for table `admin`
--
INSERT INTO `admin` (`AdminID`, `OrgID`, `SupervisedBy`, `ManagesOrg`, `ApprovedBy`) VALUES
('hhakimza', 'CSCI', 'radaikka', 0, 'hhakimza');

-- Dumping data for table `availability_status`
--
INSERT INTO `availability_status` (`AvailabilityStatusID`, `Description`) VALUES
('AVAILABLE', 'Definitely Available'),
('UNAVAILABLE', 'Definitely Unavailable');

-- Dumping data for table `employee`
--
INSERT INTO `employee` (`EmployeeID`, `Title`, `WorksForOrgID`) VALUES
('ahabraha', 'Software Engineer', 'CSCI'),
('briacarl', 'Software Engineer', 'CSCI'),
('csassano', 'Software Engineer', 'CSCI'),
('dcampau', 'Software Engineer', 'CSCI'),
('hhakimza', 'Programmer', 'CSCI'),
('hmeck', 'Software Engineer', 'CSCI'),
('jbulger', 'Database Designer', 'CSCI'),
('jjkotva', 'Database Designer', 'CSCI'),
('joecart', 'Web Developer', 'CSCI'),
('mrbarney', 'Software Engineer', 'CSCI'),
('nefranks', 'Software Engineer', 'CSCI'),
('pegarcia', 'Web Developer', 'CSCI'),
('shwhitfi', 'Software Engineer', 'CSCI'),
('szimmerl', 'Software Engineer', 'CSCI'),
('tarwiley', 'Database Designer', 'CSCI'),
('zacricha', 'Web Developer', 'CSCI');

-- Dumping data for table `employee_schedule`
--
INSERT INTO `employee_schedule` (`EmployeeID`, `DayOfWeek`, `FromTime`, `ToTime`, `AvailabilityStatusID`) VALUES
('hhakimza', 'MON', '10:00:00', '14:00:00', 'UNAVAILABLE'),
('hhakimza', 'THU', '10:00:00', '17:00:00', 'AVAILABLE'),

Appendix – C (Initial Data)
('hhakimza', 'TUE', '10:00:00', '17:00:00', 'AVAILABLE'),
('hhakimza', 'WED', '10:00:00', '20:10:00', 'UNAVAILABLE');

-- Dumping data for table `employee_specialties`

INSERT INTO `employee_specialties` (`EmployeeID`, `SkillID`, `DateSkillLearned`) VALUES
('hhakimza', 1, '2016-09-01'),
('hhakimza', 2, '1984-03-01');

-- Dumping data for table `external_contractor`

INSERT INTO `external_contractor` (`ContractorID`, `CompanyName`, `Address1`, `Address2`, `City`, `State`, `Zip`, `Phone`, `Email`, `RepName`) VALUES
(1, 'ACME Contractors', 'Indianapolis', NULL, 'Indianapolis', 'IN', '46666', '555.555.5555', 'shoddy@acme_con.com', 'Shoddy D. Job'),
(2, 'Circle of Friends Ministries', '1700 Mishawaka Ave.', NULL, NULL, 'IN', NULL, NULL, 'jj@cofm.com', 'Joseph'),
(3, 'Art Strategies LLC', NULL, NULL, 'Bloomington', 'IN', NULL, 'art@asllc.com', 'Art');

-- Dumping data for table `organization`

INSERT INTO `organization` (`OrgID`, `Name`, `Address1`, `Address2`, `City`, `State`, `Zip`, `Phone`, `Email`, `WebSite`) VALUES
('BIOL', 'Biological Sciences', '1700 Mishawaka Ave.', '', 'South Bend', 'IN', '46615', '574-520-0011', 'biol@iusb.edu', 'www.iusb.edu/biology'),
('CLAS', 'College of Liberal Arts and Sciences', '1700 Mishawaka Ave.', '', 'South Bend', 'IN', '46615', '574-520-0000', 'clas@iusb.edu', 'www.iusb.edu/clas'),
('CSCI', 'Computer and Information Sciences', '1700 Mishawaka Ave.', '', 'South Bend', 'IN', '46615', '574-520-0000', 'info@cs.iusb.edu', 'www.cs.iusb.edu'),
('DEV', 'Office of Development', '1700 Mishawaka Ave.', '', 'South Bend', 'IN', '46615', '574-520-0000', 'Dev@iusb.edu', 'www.iusb.edu/dev'),
('ENG', 'English', '1700 Mishawaka Ave.', '', 'South Bend', 'IN', '46615', '574-520-0000', 'english@iusb.edu', 'www.iusb.edu/english'),
('FAC', 'IU South Bend Facilities', '1700 Mishawaka Ave.', '', 'South Bend', 'IN', '46615', '574-520-0000', 'workorder@iusb.edu', 'www.iusb.edu/facilities');

-- Dumping data for table `priority`

INSERT INTO `priority` (`PriorityID`, `Title`, `Description`) VALUES
('HIGH', 'High Priority', 'May preempt other project with normal or low priority'),
('LOW', 'Low Priority', 'Can be paused if other higher priority items/tasks require more resources'),
('NORMAL', 'Normal Priority', 'No special resources needed. Will not yield to low priority, but can yield to high priority project.');
20 | Page

(1, 11, 'Completion of the Process', 'A. The Search and Screen Committee Chairperson must assure that all candidates who were not hired are sent letters indicating that the position has been filled. 

B. The Search and Screen Committee Chairperson must complete a Search and Screen Committee Final Report and forward it to the AAO. 

C. All documentation concerning the search must be kept on file in the department for three (3) years. This includes: meetings, procedural decisions, votes, forms, advertising information, budget, selection criteria, dossiers, correspondence, phone conversations with candidates and references, candidate rosters and rankings, and interview questions and schedules. ');

--

-- Dumping data for table `project_deliverables`

--

INSERT INTO `project_deliverables` ('ProjectID', 'OrderOfAppearance', 'Title', 'Description', 'Recipients', 'ProjectStatusID') VALUES
(1, 1, 'Search Report 1', '', 'Dean''s office', 5),
(1, 2, 'Search Report 2', '', 'Vice Chancellor for Academic Affairs', 1);

--

-- Dumping data for table `project_goal`

--

INSERT INTO `project_goal` ('ProjectID', 'OrderOfAppearance', 'Title', 'Description') VALUES
(1, 1, 'Obtain funding for the position', ''),
(1, 2, 'Successful Hiring', '');

--

-- Dumping data for table `project_manager`

--

INSERT INTO `project_manager` ('PMID', 'DesignatedBy', 'DateDesignated') VALUES
('hhakimza', 'hhakimza', '2017-11-01'),
('radaikka', 'hhakimza', '2017-10-04');

--

-- Dumping data for table `project_membership_requirement`

--

INSERT INTO `project_membership_requirement` ('ProjectMembershipReqID', 'ProjectID', 'RequirementDescription') VALUES
(1, 7, 'All voting members should be tenured faculty'),
(2, 7, 'One untenured faculty can participate as a non-voting member. Not to include the candidate for tenure'),
(3, 3, 'Appointed by the Project Manager (typically the Chair)');

--

-- Dumping data for table `project_ownership`

--

INSERT INTO `project_ownership` ('AdminID', 'ProjectID', 'DateInitiated') VALUES
('hhakimza', 1, '2018-03-01'),
('hhakimza', 2, '2018-03-01'),
('hhakimza', 3, '2018-03-25');

--

-- Dumping data for table `project_prereq`
INSERT INTO `project_prereq` (ProjectPrereqID, ProjectID, Title, Description) VALUES
(1, 1, 'Position Request', 'A position request must be submitted to the dean''s office by Sept 15. Indicate if the position is a replacement or new.'),
(2, 1, 'Equal Opportunity Training', 'If approval is obtained, contact HR and the office of Diversity for training for the search and screen committee.'),
(3, 3, 'Get Appointment Letter', 'Get appointment letter or email notification from project manager.'),
(4, 3, 'Ensure No Conflict', 'Members of committee must not have any conflict of interest when it comes to evaluating candidates. The chair/Project manager must be notified of any perceived conflicts.'),
(5, 3, 'Position Description and Skills', 'Position description and required skills must be well defined before the hiring begins.');

-- Dumping data for table `quote`

--

INSERT INTO `quote` (QuoteID, VendorQuoteNo, Title, Description, DateIssued, DateExpired, ExternalContractorID) VALUES
(1, 'X19227', 'Replacement windows for 2nd floor', 'Nice windows for the second floor', '2018-03-01', '2018-07-08', 1);

-- Dumping data for table `recurring_frequency`

--

INSERT INTO `recurring_frequency` (FrequencyID, Title, Description) VALUES
(1, 'Single Occurrence per Semester', ''),
(2, 'Once per year', ''),
(3, 'Every 3 years', ''),
(4, 'Every 7 years', ''),
(5, 'First monday of every month', ''),
(6, 'Ad-hoc recurrence', '');

-- Dumping data for table `resource`

--

INSERT INTO `resource` (ResourceID, Name, Description, ResourceTypeID) VALUES
(1, 'CS Laptop 1', 'DELL laptop', 1),
(2, 'CS Laptop 2', 'DELL laptop', 1),
(3, 'CS Projection 1', 'Portable Projection', 2),
(4, 'CS Projection 2', 'Portable Projection', 2),
(5, 'CS Projection 3', 'Portable Projection', 2),
(6, 'NS-164 (Mac Lab)', 'Mac Lab / Classroom', 3),
(7, 'NS-166 (Research Lab)', 'Research Lab', 3),
(8, 'NS-166 (Conference Room)', 'Conference Room', 6),
(9, 'NS-376 (Conference Room)', 'Conference Room', 6),
(10, 'NS-300 (Conference Room)', 'Conference Room', 6),
(11, 'NS-207 (CS LAB)', 'CS Lab', 5),
(12, 'NS-209 (CS LAB)', 'CS Lab', 5);
-- Dumping data for table `resources_required`

--

INSERT INTO `resources_required` (`TaskID`, `ResourceID`) VALUES
(1, 1),
(1, 3),
(1, 9),
(2, 7);

-- Dumping data for table `resource_type`

--

INSERT INTO `resource_type` (`ResourceTypeID`, `Name`) VALUES
(1, 'Laptop'),
(2, 'Projection'),
(3, 'iMac Computer Lab'),
(4, 'PC Computer Lab'),
(5, 'Unix Lab'),
(6, 'Conference Room');

--

-- Dumping data for table `skills`

--

INSERT INTO `skills` (`SkillID`, `Name`, `Description`, `SkillTypeID`) VALUES
(1, 'Report Writing', 'Familiarity with ISO 9004 report structure', 2),
(2, 'Presentation', 'Ability and experience presenting in front of small and large groups. Ability to think on their feet and answer unexpected questions.', 6),
(3, 'C++ Programming - Level 1', 'Equiv to one semester of programming', 6),
(4, 'C++ Programming - Level 2', 'Equiv to year and half of programming C++, plus some data structures (Micro level problem solving ability)', 6),
(5, 'Analysis and Design', 'Macro level problem solving ability', 6),
(6, 'Java Programming - Level 1', '', 6),
(7, 'Project Management - Level 1', 'Small projects, small teams (1 to 5 team members) (less than 10,000 lines of code.)', 5),
(8, 'Project Management - Level 2', 'Medium Projects, medium teams (6 to 10) (less than 100,000 lines of code.)', 5),
(9, 'Project Management - Master', 'Large Projects, Large teams (larger than 10, more than 100,000 lines of code)', 5),
(10, 'Data Modelling', 'The ability to create an ERD from description of an organization or problem domain.', 6),
(11, 'Database design and Implementation (Relational)', 'Conversion of ERD to a relational database. Understanding of relational constraints, Normalization, etc.', 6),
(12, 'SQL - Level 1', 'Ability to create basic queries', 6),
(13, 'SQL - Level 2', 'Ability to create advanced queries', 6),
(14, 'SQL - Master', 'Ability to create advanced queries, views, transactions, triggers, stored procedures, etc.', 6),
(15, 'Web Design', 'Knowledge of HTML and CSS', 6),
(16, 'Web Programming - Front End - Level - 1', 'Knowledge of HTML, CSS, Java script', 6),
(17, 'Web Programming - Front End - Level - 2', 'Knowledge of HTML, CSS, Java script, JQuery. Knowledge of one or more front-end frameworks such as Bootstrap, Angular, etc.', 6),
(18, 'Web Programming - Back End - Level - 1', 'Knowledge of one or more back-end scripting/programming language, such as PHP, NodeJS, etc.', 6),
(19, 'Web Programming - Back End - Level - 2', 'Knowledge of sessions, security issues, database connectivity, data validations, encryption, etc.', 6),
(20, 'Web Programming - Back End - Master', 'Knowledge of back-end frameworks', 6);

--
-- Dumping data for table `skills_needed`
--
INSERT INTO `skills_needed` ('TaskID', 'SkillID', 'MinYears') VALUES
(1, 1, 3),
(1, 4, 1);

--
-- Dumping data for table `skill_type`
--
INSERT INTO `skill_type` ('SkillTypeID', 'Title', 'Description') VALUES
(1, 'Verbal Communication', ''),
(2, 'Written Communication', ''),
(3, 'Interpersonal Communication', ''),
(4, 'Analytical Skills', ''),
(5, 'Leadership Skills', ''),
(6, 'Technical Skills', '');

--
-- Dumping data for table `status`
--
INSERT INTO `status` ('StatusID', 'Title', 'Description') VALUES
(1, 'Not Started', 'The project has not started yet'),
(2, 'In Progress', 'Project has started and is in progress.'),
(3, 'Terminated', 'Project has started but has abnormally terminated. (Not completed)'),
(4, 'Suspended', 'Project has started, but has been suspended until further notice.'),
(5, 'Completed', 'Project has successfully completed');

--
-- Dumping data for table `su`
--
INSERT INTO `su` ('SUID', 'StartDate') VALUES
('hhakimza', '2017-11-01'),
('radaikka', '2016-03-01');

--
-- Dumping data for table `task`
--
INSERT INTO `task` ('TaskID', 'Title', 'Description', 'StartDate', 'EndDate', 'Deadline', 'LeaderID', 'ProjectID') VALUES
(1, 'Task 1', 'Things that have to be done first...', '2018-03-01', '2018-03-08', '2018-03-08', 'ahabraha', 1),
(2, 'Task 2', 'The other things that need to be done second...', '2018-03-01', '2018-03-15', '2018-03-15', 'briacarl', 1);
-- Dumping data for table `task_deliverables`
--

INSERT INTO `task_deliverables` (`TaskID`, `OrderOfAppearance`, `Title`, `Description`, `Recipients`, `StatusID`) VALUES
(1, 1, 'Survey Results', '', 'Chair and faculty', 1),
(1, 2, 'Report 1', '', 'Chair', 1);
--
-- Dumping data for table `task_workgroup`
--

INSERT INTO `task_workgroup` (`EmployeeID`, `TaskID`) VALUES
('ahabraha', 1),
('briacarl', 1),
('ahabraha', 2),
('briacarl', 2),
('csassano', 2),
('dcampau', 2);
--
-- Dumping data for table `user`
--

INSERT INTO `user` (`UserID`, `password`) VALUES
('ahabraha', 'Pass1'),
('briacarl', 'Pass5'),
('csassano', 'Pass12'),
('dcampau', 'Pass4'),
('hhakimza', 'mypassword'),
('hmeck', 'Pass10'),
('jbulger', 'Pass3'),
('jjkotva', 'Pass9'),
('joecart', 'Pass6'),
('mrbarney', 'Pass2'),
('nefranks', 'Pass7'),
('pegarcia', 'Pass8'),
('radaikka', 'password'),
('shwhitfi', 'Pass13'),
('szimmerl', 'Pass15'),
('tarwiley', 'Pass14'),
('zacricha', 'Pass11');
--
-- Dumping data for table `user_profile`
--

INSERT INTO `user_profile` (`UserID`, `OrgID`, `FName`, `LName`, `Address1`, `Address2`, `City`, `State`, `Zip`, `Phone`, `Email`) VALUES
('ahabraha', 'CSCI', 'Abraham', 'Akiva', '123 Mishawaka Ave.', '', 'South Bend', 'IN', '46615', '574.555.1001', 'ahabraha@iu.edu'),
insert into `week_days` (`DayOfWeek`, `Description`) values
('FRI', 'Friday'),
('MON', 'Monday'),
('SAT', 'Saturday'),
('SUN', 'Sunday'),
('THU', 'Thursday'),
('TUE', 'Tuesday'),
('WED', 'Wednesday');