***{s.dot} Create Table Statements***

*Note - AUTO\_INCREMENT does not work in Oracle / Sql Plus, only MySql*

*Note - MySql does not recognize NUMBER as a variable type*

**Drop Previous Tables**

*Note – In this order due to foreign key constraints*

DROP TABLE Forum;

DROP TABLE Bugs;

DROP TABLE Tasks;

DROP TABLE ProjectMembers;

DROP TABLE Projects;

DROP TABLE Users;

**Create Users**

CREATE TABLE Users

(

UserID INT NOT NULL AUTO\_INCREMENT,

Username VARCHAR(20) NOT NULL,

FName VARCHAR(20) NOT NULL,

LName VARCHAR(20) NOT NULL,

Password VARCHAR(20) NOT NULL,

Email VARCHAR(30) NOT NULL,

CONSTRAINT UserPK PRIMARY KEY (UserID),

CONSTRAINT UniqueUser UNIQUE (Username)

);

**Create Projects**

CREATE TABLE Projects

(

ProjectID INT NOT NULL AUTO\_INCREMENT,

ProjectName VARCHAR(50) NOT NULL,

ProjectSummary VARCHAR(1000),

CONSTRAINT ProjectPK PRIMARY KEY (ProjectID)

);

**Create ProjectMembers**

CREATE TABLE ProjectMembers

(

PMProjectID INT NOT NULL,

PMUserID INT NOT NULL,

CONSTRAINT ProjectFK FOREIGN KEY (PMProjectID) REFERENCES Projects (ProjectID),

CONSTRAINT UserFK FOREIGN KEY (PMUserID) REFERENCES Users (UserID)

);

**Create Tasks**

CREATE TABLE Tasks

(

TaskID INT NOT NULL AUTO\_INCREMENT,

Priority VARCHAR(6) NOT NULL,

Task VARCHAR(200) NOT NULL,

Complete CHAR(5) NOT NULL,

SCRUM INT NOT NULL,

Remarks VARCHAR(500),

TProjectID INT NOT NULL,

TAssignedID INT NOT NULL,

CONSTRAINT TaskPK PRIMARY KEY (TaskID),

CONSTRAINT TProjectFK FOREIGN KEY (TProjectID) REFERENCES Projects (ProjectID),

CONSTRAINT TUserFK FOREIGN KEY (TAssignedID) REFERENCES Users (UserID),

CONSTRAINT CompleteCheck CHECK(Complete='false' OR Complete='true')

CONSTRAINT PriorityCheck CHECK(Priority='Low' OR Priority='Medium' OR Priority='High')

);

**Create Bugs**

CREATE TABLE Bugs

(

BugID INT NOT NULL AUTO\_INCREMENT,

SourceFile VARCHAR(50) NOT NULL,

Location VARCHAR(250) NOT NULL,

Description VARCHAR(500) NOT NULL,

Status VARCHAR(25) NOT NULL,

BProjectID INT NOT NULL,

BReporterID INT NOT NULL,

BAssignedID INT NOT NULL,

CONSTRAINT BugsPK PRIMARY KEY (BugID),

CONSTRAINT BProjectFK FOREIGN KEY (BProjectID) REFERENCES Projects (ProjectID),

CONSTRAINT BReporterFK FOREIGN KEY (BReporterID) REFERENCES Users (UserID),

CONSTRAINT BAssignedFK FOREIGN KEY (BAssignedID) REFERENCES Users (UserID)

CONSTRAINT StatusCheck CHECK(Status='Unassigned' OR Status='Low Priority' OR Status='Med Priority' OR Status='High Priority')

);

**Create Forum**

CREATE TABLE Forum

(

ForumID INT NOT NULL AUTO\_INCREMENT,

Comment VARCHAR(3000) NOT NULL,

FProjectID INT NOT NULL,

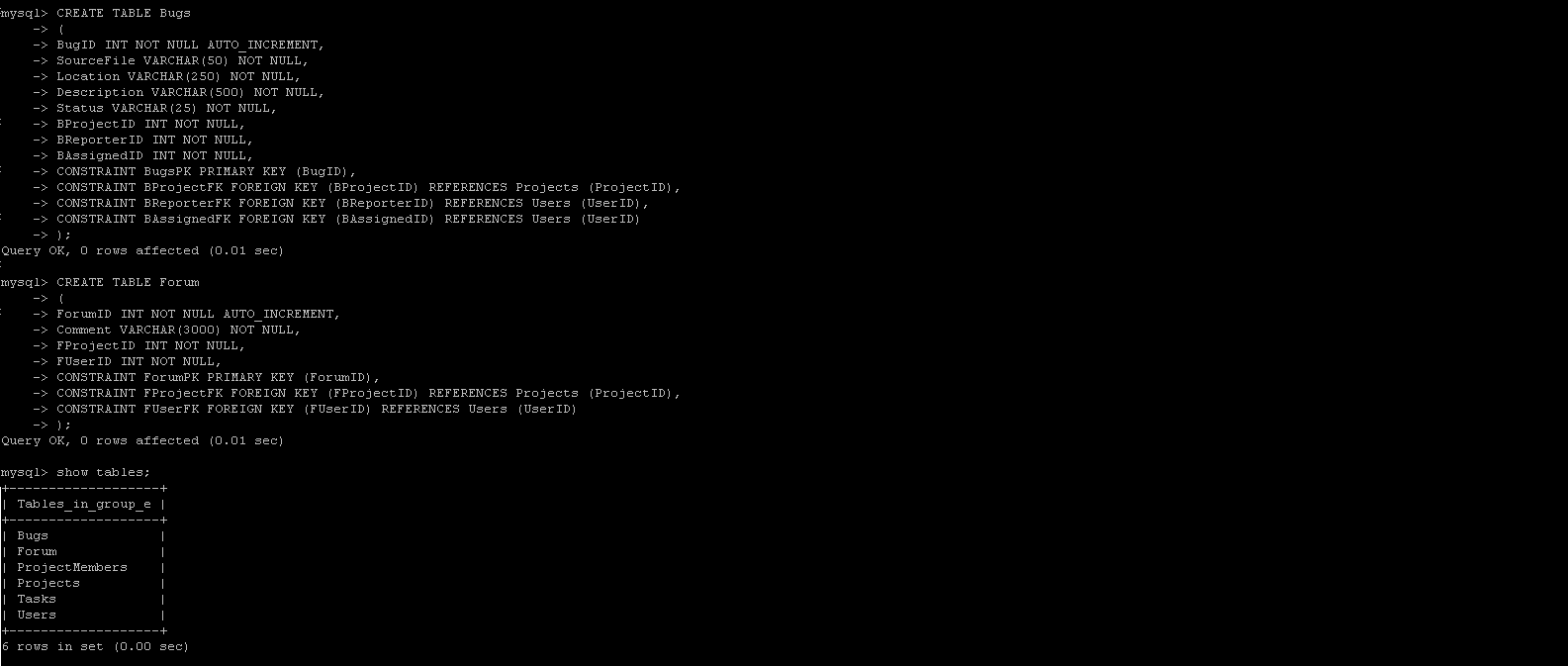
FUserID INT NOT NULL,

CONSTRAINT ForumPK PRIMARY KEY (ForumID),

CONSTRAINT FProjectFK FOREIGN KEY (FProjectID) REFERENCES Projects (ProjectID),

CONSTRAINT FUserFK FOREIGN KEY (FUserID) REFERENCES Users (UserID)

);

***{s.dot} Test Data Load***

**Populate Users**

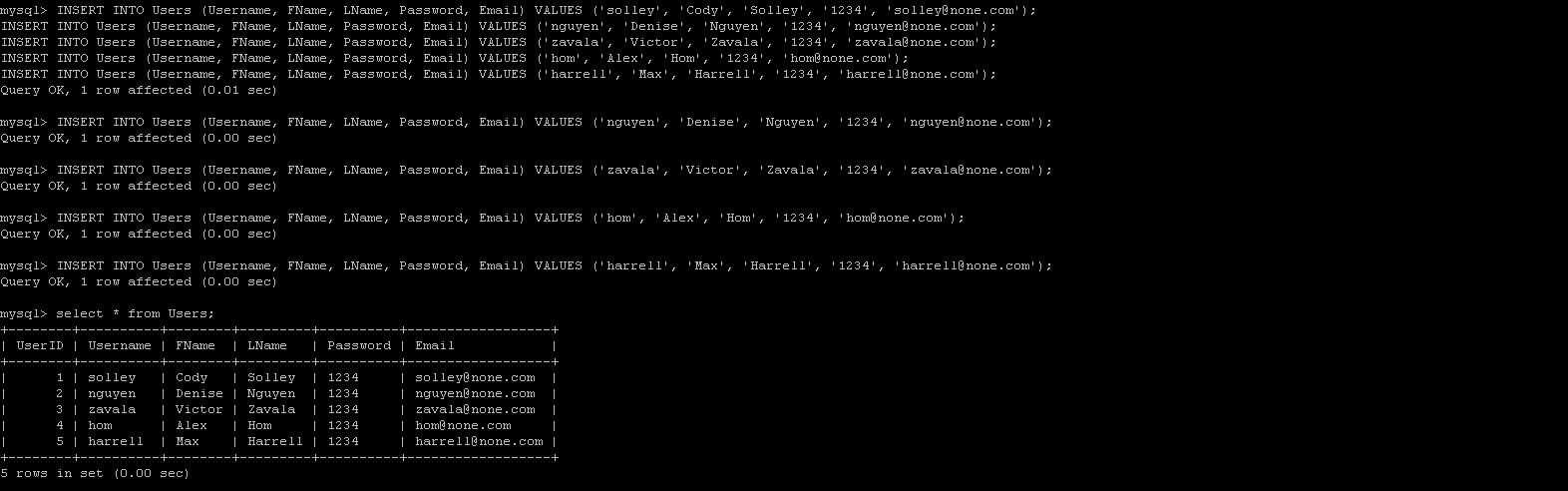
INSERT INTO Users (Username, FName, LName, Password, Email) VALUES ('solley', 'Cody', 'Solley', '1234', 'solley@none.com');

INSERT INTO Users (Username, FName, LName, Password, Email) VALUES ('nguyen', 'Denise', 'Nguyen', '1234', 'nguyen@none.com');

INSERT INTO Users (Username, FName, LName, Password, Email) VALUES ('zavala', 'Victor', 'Zavala', '1234', 'zavala@none.com');

INSERT INTO Users (Username, FName, LName, Password, Email) VALUES ('hom', 'Alex', 'Hom', '1234', 'hom@none.com');

INSERT INTO Users (Username, FName, LName, Password, Email) VALUES ('harrell', 'Max', 'Harrell', '1234', 'harrell@none.com');

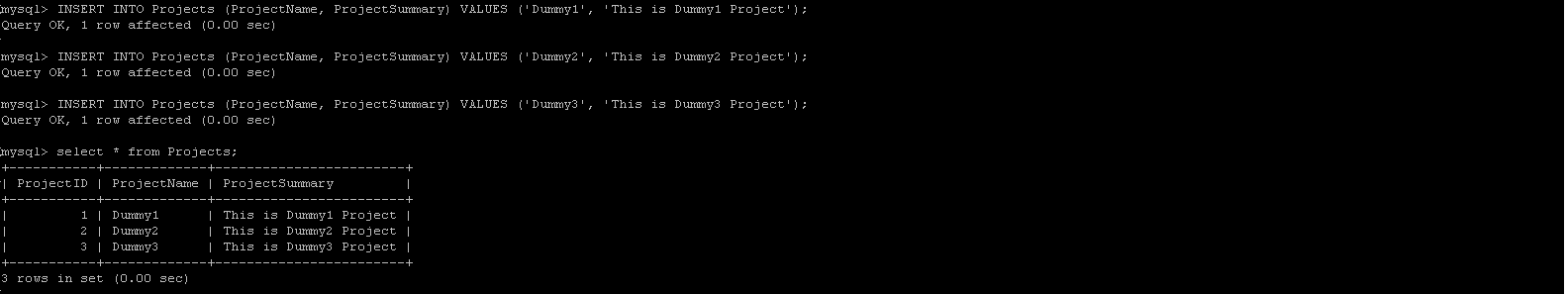


**Populate Projects**

INSERT INTO Projects (ProjectName, ProjectSummary) VALUES ('Dummy1', 'This is Dummy1 Project');

INSERT INTO Projects (ProjectName, ProjectSummary) VALUES ('Dummy2', 'This is Dummy2 Project');

INSERT INTO Projects (ProjectName, ProjectSummary) VALUES ('Dummy3', 'This is Dummy3 Project');



**Populate ProjectMembers**

INSERT INTO ProjectMembers VALUES (1,1);

INSERT INTO ProjectMembers VALUES (1,2);

INSERT INTO ProjectMembers VALUES (1,3);

INSERT INTO ProjectMembers VALUES (1,4);

INSERT INTO ProjectMembers VALUES (1,5);

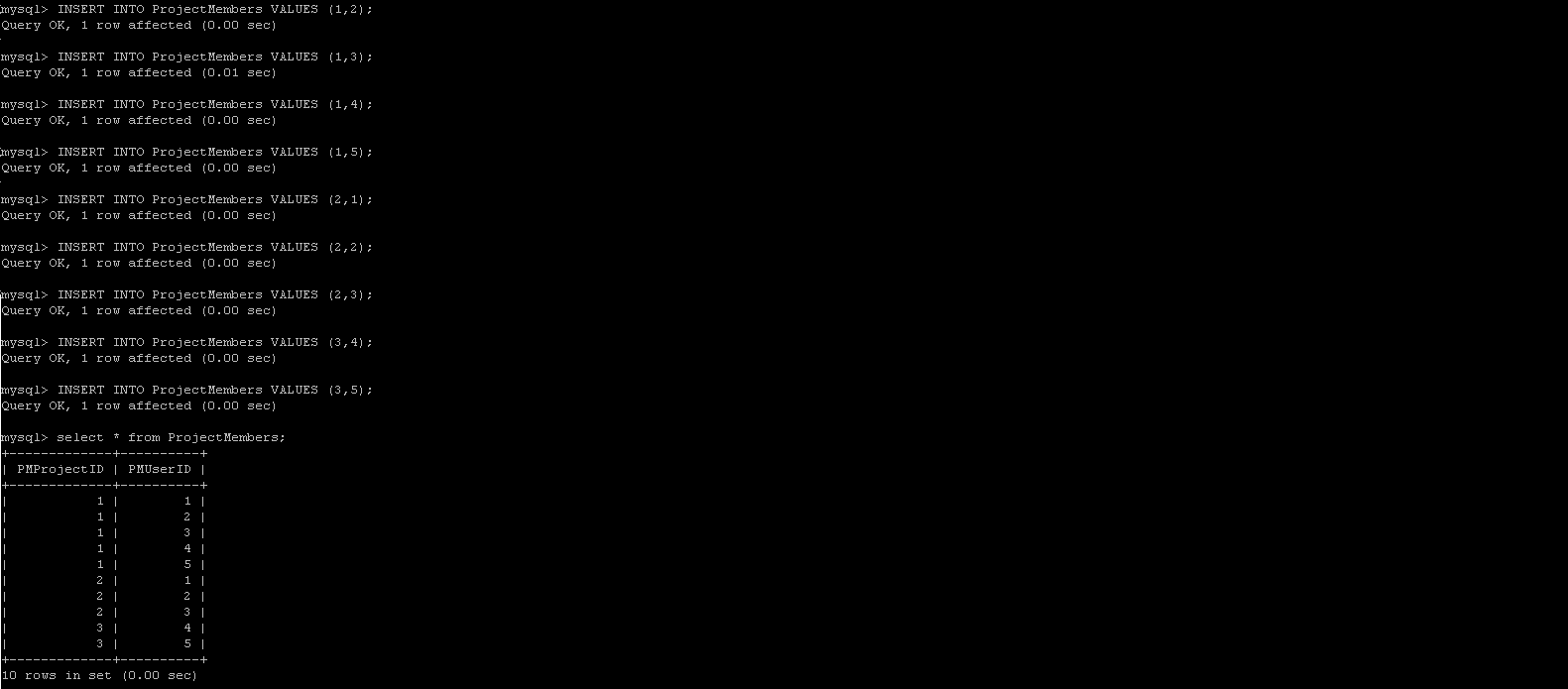
INSERT INTO ProjectMembers VALUES (2,1);

INSERT INTO ProjectMembers VALUES (2,2);

INSERT INTO ProjectMembers VALUES (2,3);

INSERT INTO ProjectMembers VALUES (3,4);

INSERT INTO ProjectMembers VALUES (3,5);



**Populate Tasks**

INSERT INTO Tasks (Priority, Task, Complete, SCRUM, Remarks, TProjectID, TAssignedID) VALUES ('Low', 'DummyTask1', 'false', 1, 'None', 1, 1);

INSERT INTO Tasks (Priority, Task, Complete, SCRUM, Remarks, TProjectID, TAssignedID) VALUES ('Low', 'DummyTask2', 'false', 1, 'None', 1, 2);

INSERT INTO Tasks (Priority, Task, Complete, SCRUM, Remarks, TProjectID, TAssignedID) VALUES ('Medium', 'DummyTask3', 'false', 3, 'None', 2, 3);

INSERT INTO Tasks (Priority, Task, Complete, SCRUM, Remarks, TProjectID, TAssignedID) VALUES ('Medium', 'DummyTask4', 'false', 2, 'None', 2, 2);

INSERT INTO Tasks (Priority, Task, Complete, SCRUM, Remarks, TProjectID, TAssignedID) VALUES ('High', 'DummyTask5', 'true', 5, 'None', 3, 5);

INSERT INTO Tasks (Priority, Task, Complete, SCRUM, Remarks, TProjectID, TAssignedID) VALUES ('Low', 'DummyTask6', 'false', 1, 'None', 1, 2);

INSERT INTO Tasks (Priority, Task, Complete, SCRUM, Remarks, TProjectID, TAssignedID) VALUES ('Low', 'DummyTask7', 'false', 1, 'None', 2, 2);

INSERT INTO Tasks (Priority, Task, Complete, SCRUM, Remarks, TProjectID, TAssignedID) VALUES ('Medium', 'DummyTask8', 'false', 3, 'None', 2, 3);

INSERT INTO Tasks (Priority, Task, Complete, SCRUM, Remarks, TProjectID, TAssignedID) VALUES ('Medium', 'DummyTask9', 'false', 2, 'None', 3, 4);

INSERT INTO Tasks (Priority, Task, Complete, SCRUM, Remarks, TProjectID, TAssignedID) VALUES ('High', 'DummyTask10', 'true', 5, 'None', 3, 5);

INSERT INTO Tasks (Priority, Task, Complete, SCRUM, Remarks, TProjectID, TAssignedID) VALUES ('Low', 'DummyTask11', 'false', 1, 'None', 1, 5);

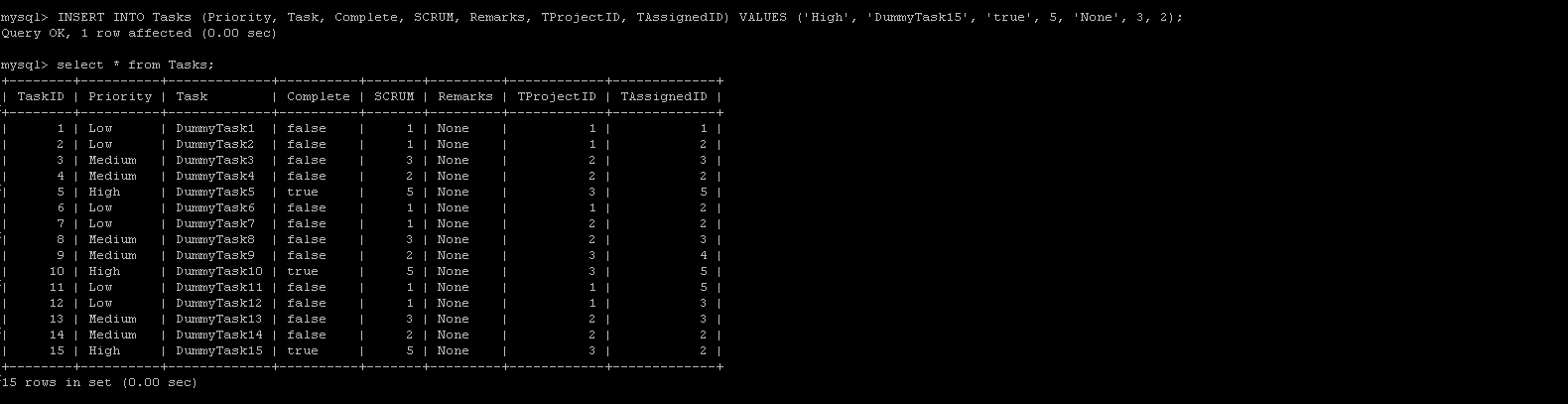
INSERT INTO Tasks (Priority, Task, Complete, SCRUM, Remarks, TProjectID, TAssignedID) VALUES ('Low', 'DummyTask12', 'false', 1, 'None', 1, 3);

INSERT INTO Tasks (Priority, Task, Complete, SCRUM, Remarks, TProjectID, TAssignedID) VALUES ('Medium', 'DummyTask13', 'false', 3, 'None', 2, 3);

INSERT INTO Tasks (Priority, Task, Complete, SCRUM, Remarks, TProjectID, TAssignedID) VALUES ('Medium', 'DummyTask14', 'false', 2, 'None', 2, 2);

INSERT INTO Tasks (Priority, Task, Complete, SCRUM, Remarks, TProjectID, TAssignedID) VALUES ('High', 'DummyTask15', 'true', 5, 'None', 3, 2);





**Populate Bugs**

INSERT INTO Bugs (SourceFile, Location, Description, Status, BProjectID, BReporterID, BAssignedID) VALUES ('Dummy1.exe', 'Line 1 Dummy Code', 'There is no real code', 'Low Priority', 1, 1, 1);

INSERT INTO Bugs (SourceFile, Location, Description, Status, BProjectID, BReporterID, BAssignedID) VALUES ('Dummy2.exe', 'Line 15 Dummy Code', 'There is no real code', 'Low Priority', 2, 1, 2);

INSERT INTO Bugs (SourceFile, Location, Description, Status, BProjectID, BReporterID, BAssignedID) VALUES ('Dummy3.exe', 'Line 11 Dummy Code', 'There is no real code', 'Med Priority', 3, 3, 1);

INSERT INTO Bugs (SourceFile, Location, Description, Status, BProjectID, BReporterID, BAssignedID) VALUES ('Dummy4.exe', 'Line 54 Dummy Code', 'There is no real code', 'High Priority', 2, 1, 2);

INSERT INTO Bugs (SourceFile, Location, Description, Status, BProjectID, BReporterID, BAssignedID) VALUES ('Dummy5.exe', 'Line 12 Dummy Code', 'There is no real code', 'Low Priority', 3, 5, 4);

INSERT INTO Bugs (SourceFile, Location, Description, Status, BProjectID, BReporterID, BAssignedID) VALUES ('Dummy6.exe', 'Line 11 Dummy Code', 'There is no real code', 'Low Priority', 1, 3, 2);

INSERT INTO Bugs (SourceFile, Location, Description, Status, BProjectID, BReporterID, BAssignedID) VALUES ('Dummy7.exe', 'Line 21 Dummy Code', 'There is no real code', 'Low Priority', 3, 3, 1);

INSERT INTO Bugs (SourceFile, Location, Description, Status, BProjectID, BReporterID, BAssignedID) VALUES ('Dummy8.exe', 'Line 54 Dummy Code', 'There is no real code', 'Med Priority', 2, 1, 2);

INSERT INTO Bugs (SourceFile, Location, Description, Status, BProjectID, BReporterID, BAssignedID) VALUES ('Dummy9.exe', 'Line 23 Dummy Code', 'There is no real code', 'Unassigned', 3, 4, 4);

INSERT INTO Bugs (SourceFile, Location, Description, Status, BProjectID, BReporterID, BAssignedID) VALUES ('Dummy10.exe', 'Line 12 Dummy Code', 'There is no real code', 'Low Priority', 3, 5, 4);

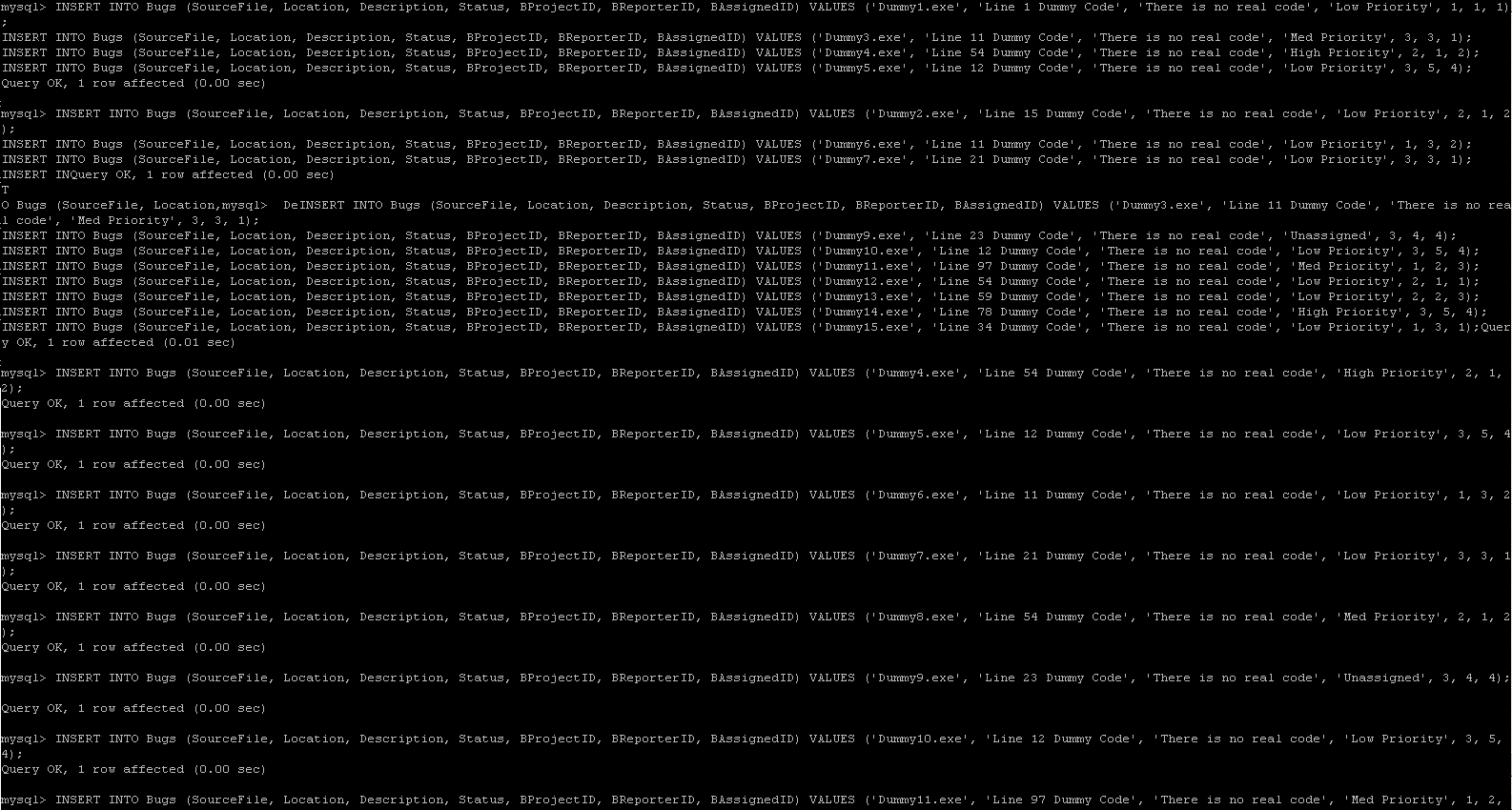
INSERT INTO Bugs (SourceFile, Location, Description, Status, BProjectID, BReporterID, BAssignedID) VALUES ('Dummy11.exe', 'Line 97 Dummy Code', 'There is no real code', 'Med Priority', 1, 2, 3);

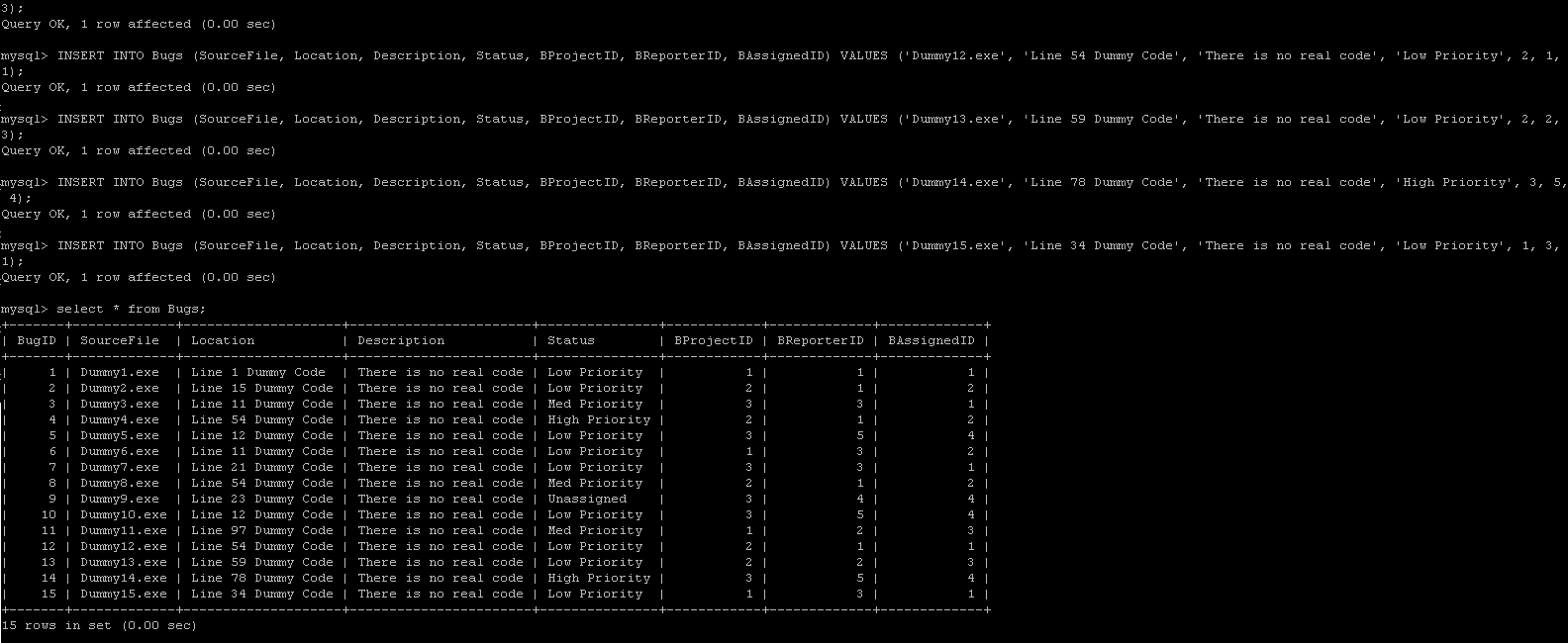
INSERT INTO Bugs (SourceFile, Location, Description, Status, BProjectID, BReporterID, BAssignedID) VALUES ('Dummy12.exe', 'Line 54 Dummy Code', 'There is no real code', 'Low Priority', 2, 1, 1);

INSERT INTO Bugs (SourceFile, Location, Description, Status, BProjectID, BReporterID, BAssignedID) VALUES ('Dummy13.exe', 'Line 59 Dummy Code', 'There is no real code', 'Low Priority', 2, 2, 3);

INSERT INTO Bugs (SourceFile, Location, Description, Status, BProjectID, BReporterID, BAssignedID) VALUES ('Dummy14.exe', 'Line 78 Dummy Code', 'There is no real code', 'High Priority', 3, 5, 4);

INSERT INTO Bugs (SourceFile, Location, Description, Status, BProjectID, BReporterID, BAssignedID) VALUES ('Dummy15.exe', 'Line 34 Dummy Code', 'There is no real code', 'Low Priority', 1, 3, 1);





**Populate Forum**

INSERT INTO Forum (Comment, FProjectID, FUserID) VALUES ('Hi this is a comment', 1, 1);

INSERT INTO Forum (Comment, FProjectID, FUserID) VALUES ('Hi this is a comment reply', 1, 2);

INSERT INTO Forum (Comment, FProjectID, FUserID) VALUES ('I can comment too', 1, 3);

INSERT INTO Forum (Comment, FProjectID, FUserID) VALUES ('I can comment 4th', 1, 4);

INSERT INTO Forum (Comment, FProjectID, FUserID) VALUES ('This forum sucks', 1, 5);

INSERT INTO Forum (Comment, FProjectID, FUserID) VALUES ('Hi this is a comment', 2, 1);

INSERT INTO Forum (Comment, FProjectID, FUserID) VALUES ('Hi this is a comment reply', 2, 2);

INSERT INTO Forum (Comment, FProjectID, FUserID) VALUES ('Hi this is the third comment', 2, 3);

INSERT INTO Forum (Comment, FProjectID, FUserID) VALUES ('Ill take over now', 2, 2);

INSERT INTO Forum (Comment, FProjectID, FUserID) VALUES ('No you wont', 2, 1);

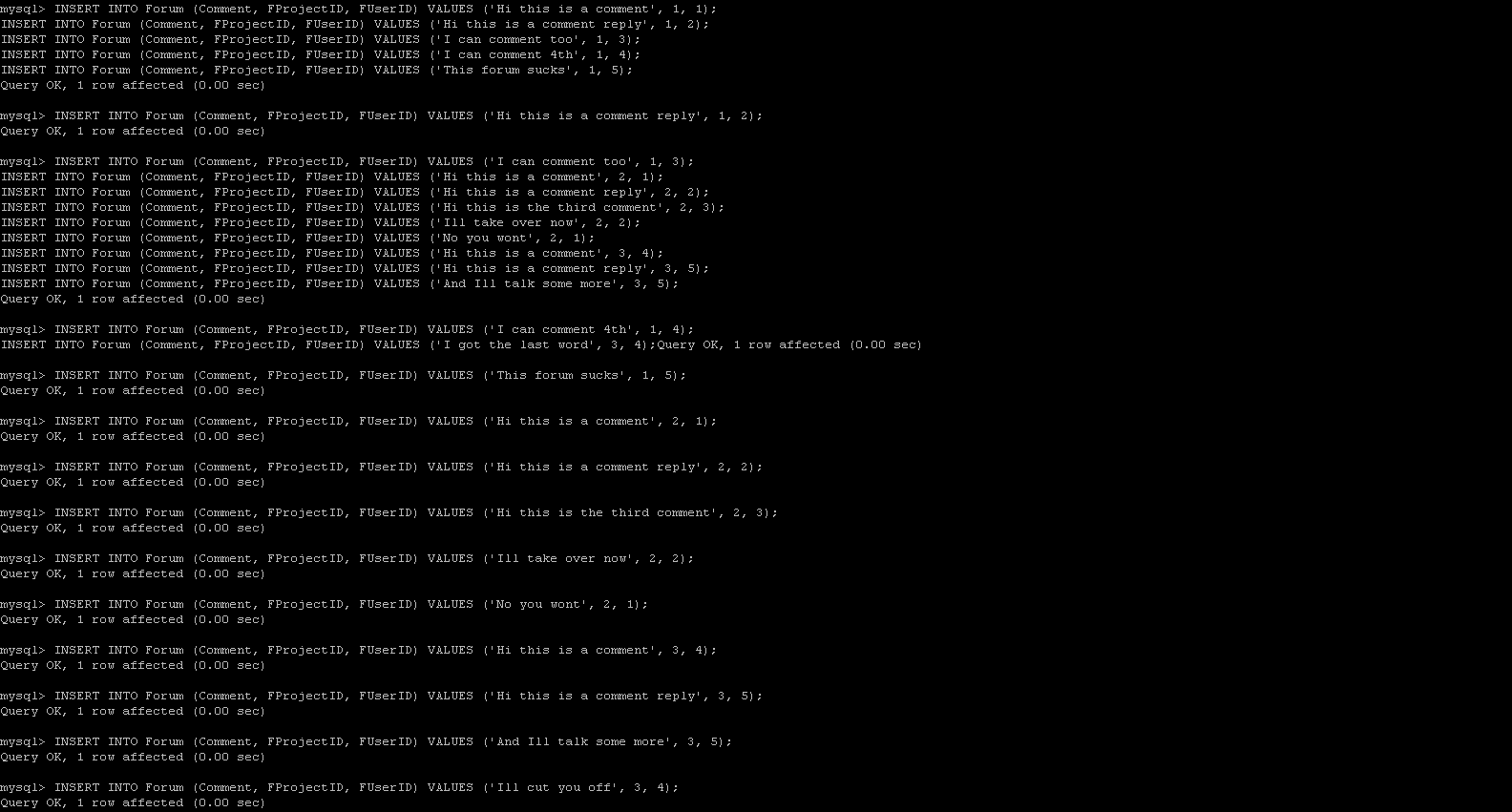
INSERT INTO Forum (Comment, FProjectID, FUserID) VALUES ('Hi this is a comment', 3, 4);

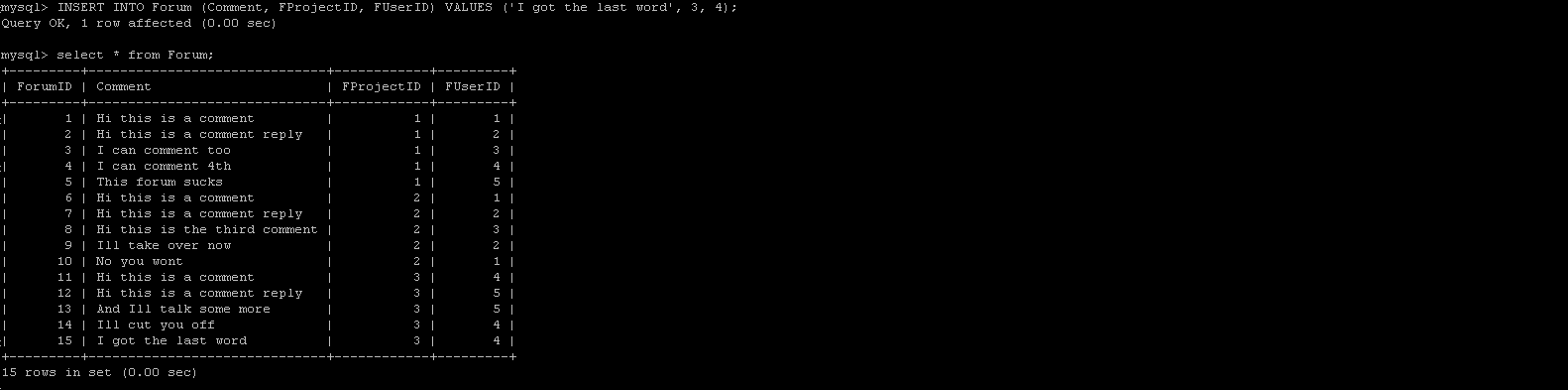
INSERT INTO Forum (Comment, FProjectID, FUserID) VALUES ('Hi this is a comment reply', 3, 5);

INSERT INTO Forum (Comment, FProjectID, FUserID) VALUES ('And Ill talk some more', 3, 5);

INSERT INTO Forum (Comment, FProjectID, FUserID) VALUES ('Ill cut you off', 3, 4);

INSERT INTO Forum (Comment, FProjectID, FUserID) VALUES ('I got the last word', 3, 4);





***{s.dot} Page Interactions with SQL and Data Requirements***

*Notes - [brackets] indicate a variable or user fill-in from the page in the example query*

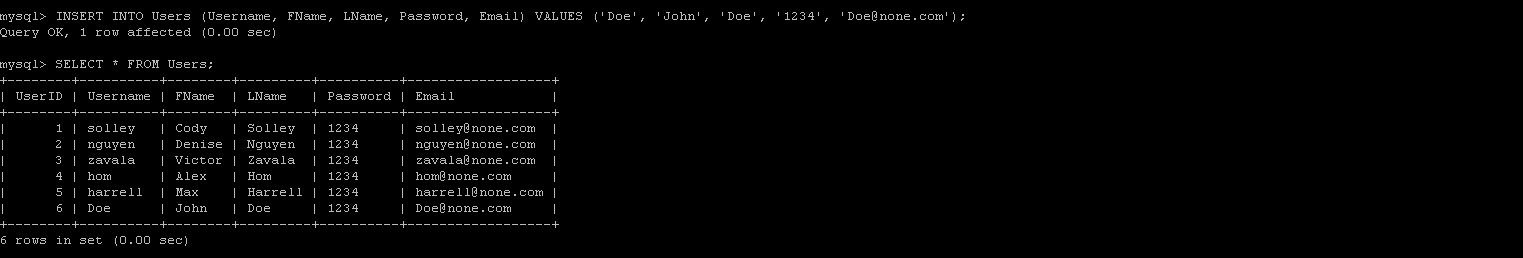
**New User**

User fills out page, upon clicking Submit run query

**INSERT INTO Users (Username, FName, LName, Password, Email) VALUES ('[Username\_Field]', '[FirstName\_Field]', '[LastName\_Field]', '[Password\_Field]', '[Email\_Field]');**

then return to the login page.

*Example: Doe, John, Doe, 1234, doe@none.com*



**Login**

User fills in username and password, upon clicking Submit run query

**SELECT Password FROM Users WHERE Username = '[Username\_Field]';**

and compare return to [Password\_Field].

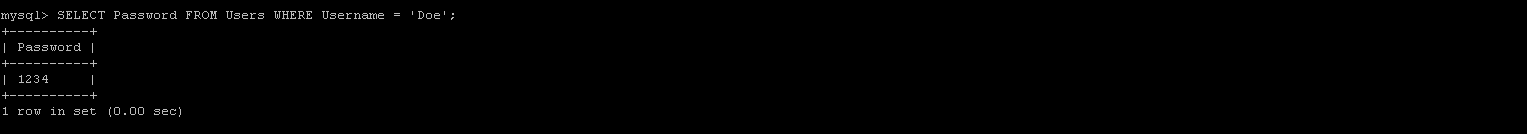
If equal then run query

**SELECT UserID FROM Users WHERE Username = '[Username\_Field]';**

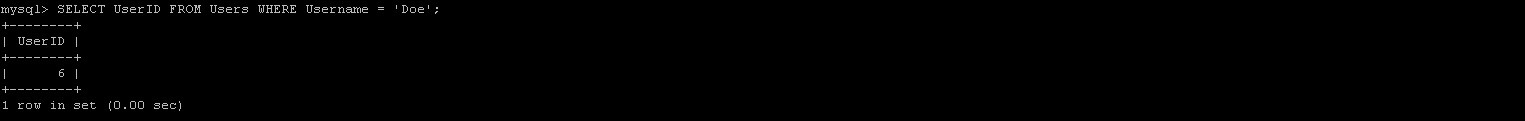
and save the UserID to a permanent variable that can be transferred from page to page.

Else reject login and display an error.

*Example: Getting the password for username Doe*

**

*Example: Getting the UserID for Doe once password is confirmed*



**Choose Project**

To dynamically fill in the page, run the following query

**SELECT ProjectName FROM Projects, ProjectMembers WHERE (PMUserID = [UserID\_Variable]) AND (ProjectID = PMProjectID);**

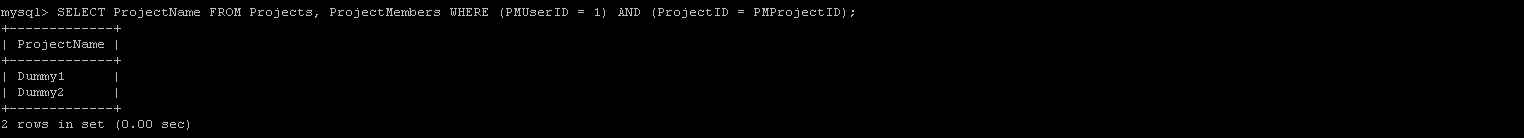
then set the project names on the page as choose-able links.

Once a link is chosen, run the following query

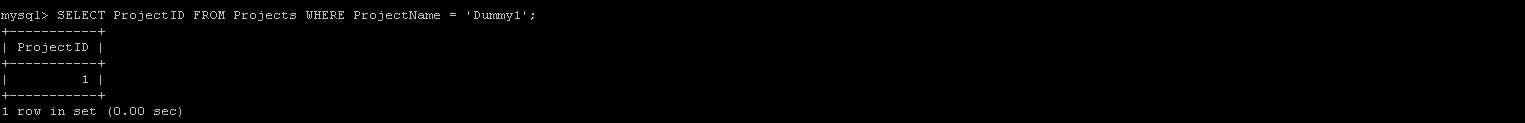
**SELECT ProjectID FROM Projects WHERE ProjectName = '[ProjectName\_Chosen]';**

And save as a permanent variable that can be taken page to page.

*Example: Show all projects for UserID 1, who is logged in*

**

*Example: User Selects Dummy1, retrieving the ProjectID*



**Create Project**

User fills out the form, upon Submit run the query

**INSERT INTO Projects (ProjectName, ProjectSummary) VALUES ('[ProjectName\_Field]','[ProjectSummary\_Field]');**

Then we must add the creating user to the new project for access, save the new ProjectID to a temp variable

**SELECT ProjectID FROM Projects WHERE ProjectName = ‘[ProjectName\_Field]’;**

and insert it into the ProjectMembers table

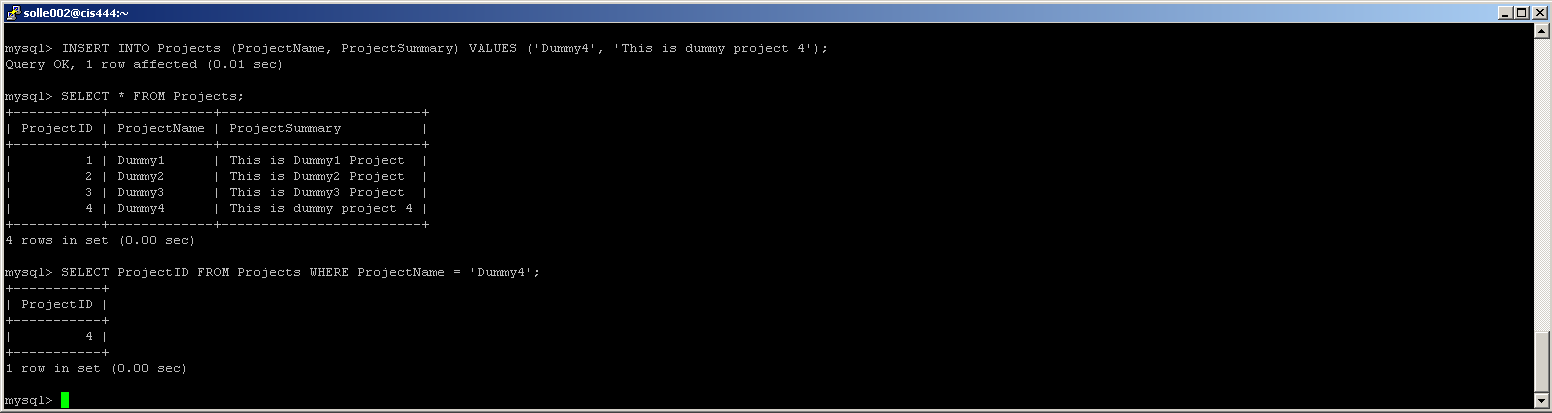
**INSERT INTO ProjectMembers VALUES ('[ProjectID\_Temp\_Variable]', '[UserID\_Variable]');**

then take back to the Choose Project page.

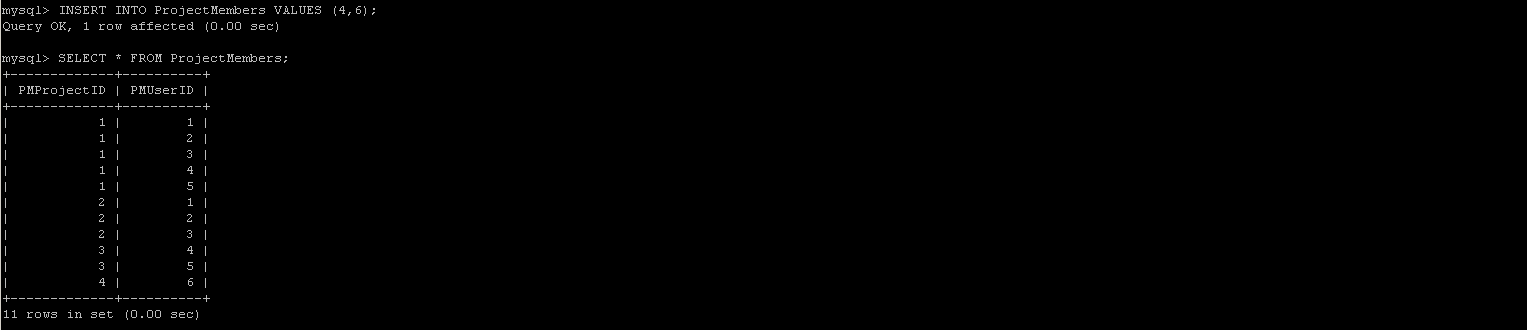
*Example: Creating new project Dummy4*

**

*Example: Getting new project Dummy4 ID*

**

*Example: Adding the creator, Doe, to the new project Dummy4*

**

**User Preferences**

To dynamically fill in the page, run the query

**SELECT Username, FName, LName, Password, Email FROM Users WHERE UserID = [UserID\_Variable];**

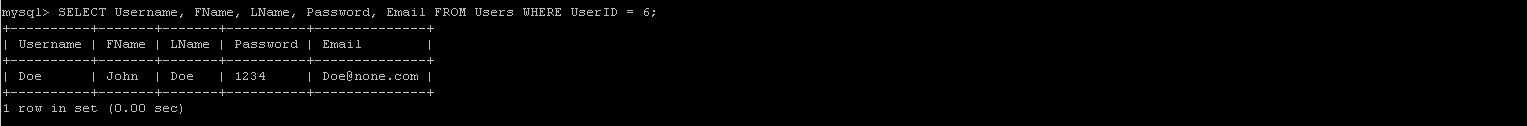
then fill the data into the appropriate fields on the webpage form.

User can change everything but UserID, upon clicking Submit run update query

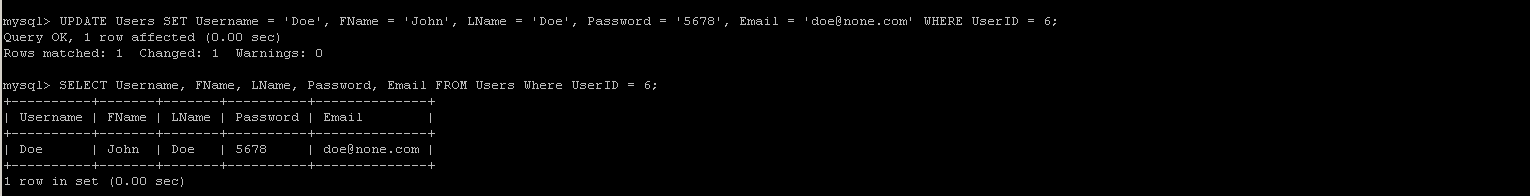
**UPDATE Users SET Username = [Username\_Field], FName = [FName\_Field], LName = [LName\_Field], Password = [Password\_Field], Email = [Email\_Field] WHERE UserID = [UserID\_Variable];**

else cancel takes you back to Choose Project.

*Example: Getting the user data for UserID 6, who is logged in*



*Example: Updating Doe’s password to 5678*



**Project Preferences**

To dynamically fill in project info, run the query

**SELECT ProjectName, ProjectSummary FROM Projects WHERE ProjectID = [ProjectID\_Variable];**

then fill in the appropriate form fields on the webpage form.

To fill in to User Access area of who has access to this project, run the query

**SELECT Username FROM Users, ProjectMembers WHERE (PMProjectID = [ProjectID\_Variable]) AND (UserID = PMUserID);**

To show all usernames for an add member to project function, run the query

**SELECT Username FROM Users;**

User can change everything but ProjectID, upon clicking submit run update query

**UPDATE Projects SET ProjectName = [ProjectName\_Field], ProjectSummary = [ProjectSummary\_Field] WHERE ProjectID = [ProjectID\_Variable];**

To add a user for access, choose a username and get their UserID by running the query

**SELECT UserID FROM Users WHERE Username = '[Username\_Choosen]';**

and save to a temp variable then run the query

**INSERT INTO ProjectMembers VALUES ([ProjectID\_Variable], [UserID\_Temp]);**

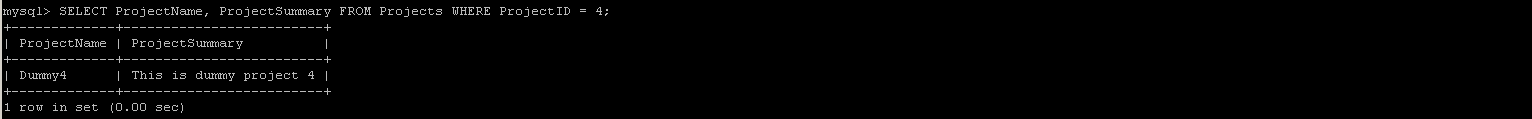
To remove a user from access, choose a username and get their UserID by running the query

**SELECT UserID FROM Users WHERE Username = '[Username\_Chosen]';**

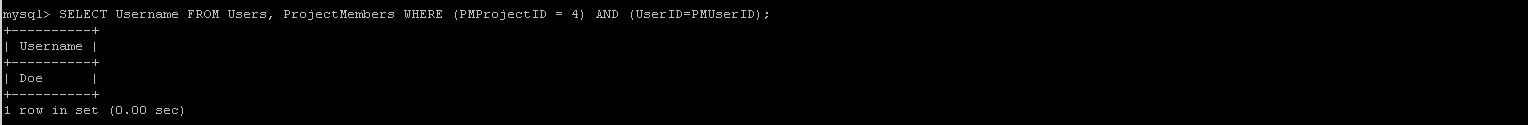
and save to a temp variable then run the query

**DELETE FROM ProjectMembers WHERE (PMProjectID = [ProjectID Variable]) AND (PMUserID = [UserID\_Temp]);**

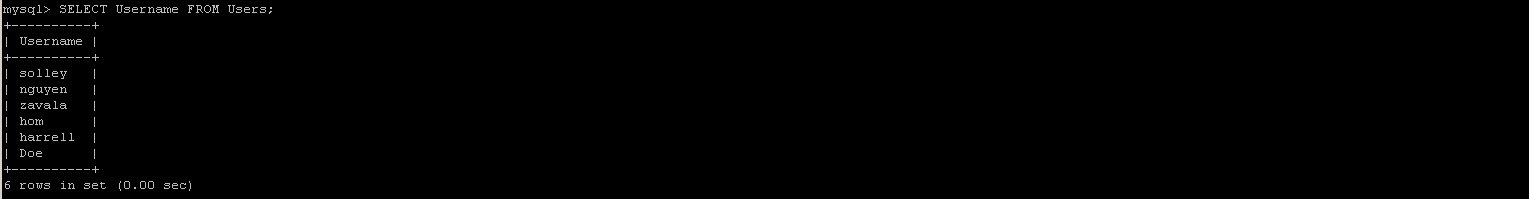
*Example: Getting the project data for Dummy4*

**

*Example: Getting the data for who has access to Dummy4*

**

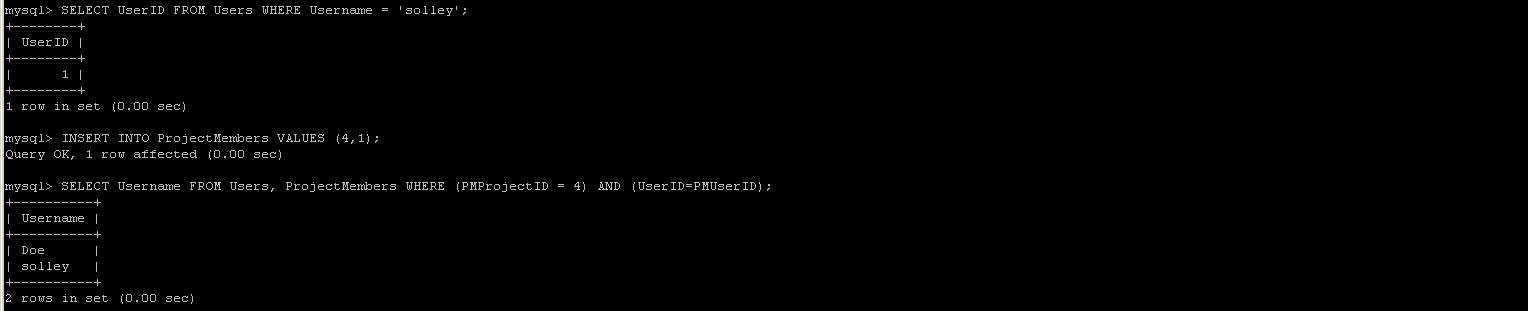
*Example: Getting all users on the site*

**

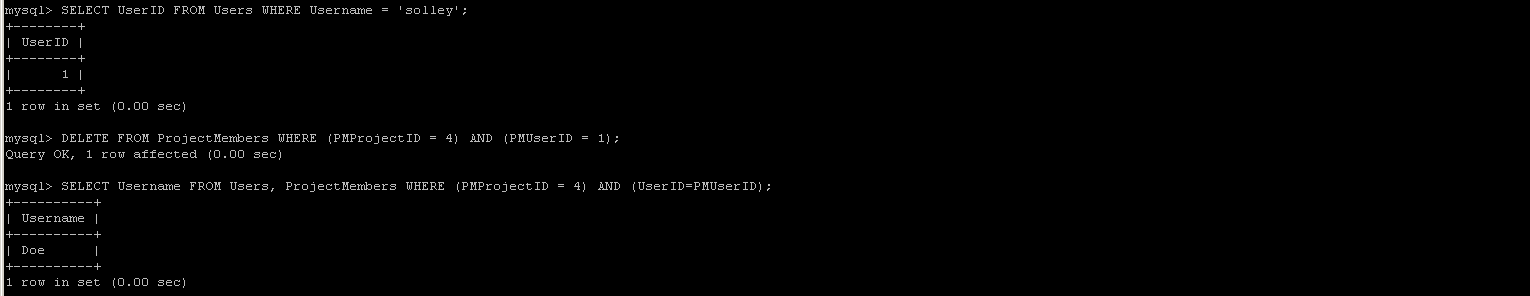
*Example: Updating Dummy4 Description*

**

*Example: Adding solley to Dummy4*

**

*Example: Removing solley from Dummy4*

**

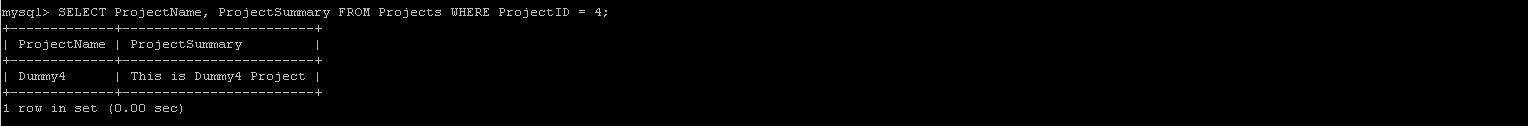
**Project Hub**

Just displays links to the three main pages (Tasks, Bugs and Forum) however we can display project info at the top by running the query

**SELECT ProjectName, ProjectSummary FROM Projects WHERE ProjectID = [ProjectID\_Variable];**

and parsing the data to the form.

*Example: Display the data for Dummy4*

**

**Tasks**

To dynamically fill in the table on the page (sorted by Task ID), run the query

**SELECT TaskID, Priority, Task, Complete, Username, SCRUM, Remarks FROM Tasks, Users WHERE (TProjectID = [ProjectID\_Variable]) AND (UserID = TAssignedID);**

then fill in the appropriate form fields on the webpage form.

To sort by User the query would be

**SELECT TaskID, Priority, Task, Complete, Username, SCRUM, Remarks FROM Tasks, Users WHERE (TProjectID = [ProjectID\_Variable]) AND (UserID = TAssignedID) ORDER BY Username ASC;**

To sort by SCRUM the query would be

**SELECT TaskID, Priority, Task, Complete, Username, SCRUM, Remarks FROM Tasks, Users WHERE (TProjectID = [ProjectID\_Variable]) AND (UserID = TAssignedID) ORDER BY SCRUM ASC;**

To sort by Priority the query would be

**SELECT TaskID, Priority, Task, Complete, Username, SCRUM, Remarks FROM Tasks, Users WHERE (TProjectID = [ProjectID\_Variable]) AND (UserID = TAssignedID) ORDER BY FIELD (Priority, 'High', 'Medium', 'Low');**

To sort by Complete the query would be

**SELECT TaskID, Priority, Task, Complete, Username, SCRUM, Remarks FROM Tasks, Users WHERE (TProjectID = [ProjectID\_Variable]) AND (UserID = TAssignedID) ORDER BY Complete ASC;**

To populate the assigned user dropdown, run the query

**SELECT Username FROM Users, ProjectMembers WHERE (PMProjectID = [ProjectID\_Variable]) AND (UserID = PMUserID);**

To get the chosen usernames UserID:

**SELECT UserID FROM Users WHERE Username = '[Username\_Chosen]';**

And save to a UsedID temp variable.

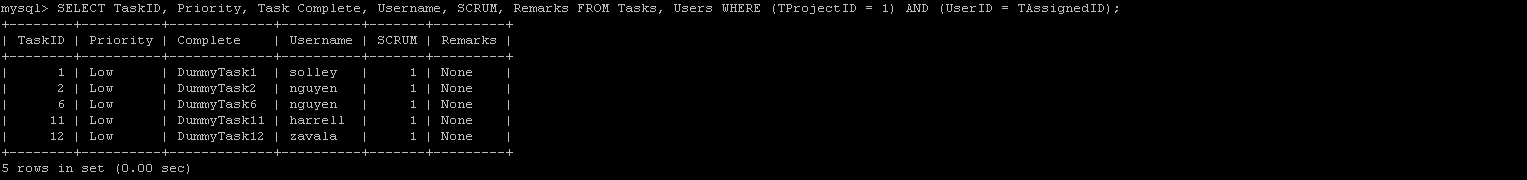
To add new task run the query

**INSERT INTO Tasks (Priority, Task, Complete, SCRUM, Remarks, TProjectID, TAssignedID) VALUES ('[Priority\_Field]','[Task\_Field]','[Complete\_Field]', [SCRUM\_Field], '[Remarks\_Field]', [ProjectID\_Variable], [UserID\_Temp\_Variable]);**

User can change all fields except Task ID, upon clicking Submit run update query

**UPDATE Tasks SET Priority = [Priority\_Field], Task = [Tasks\_Field], Complete [Complete\_Field], SCRUM = [SCRUM\_Field], Remarks = [Remarks\_Field], TProjectID = [ProjectID\_Variable], TAssignedID = [UserID\_Temp\_Variable] WHERE TaskID = [ValueOf\_TaskID\_Field];**

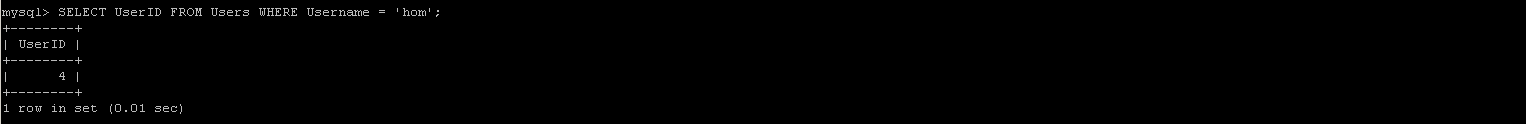
*Example: Get the task data for current project Dummy1, ProjectID 1*



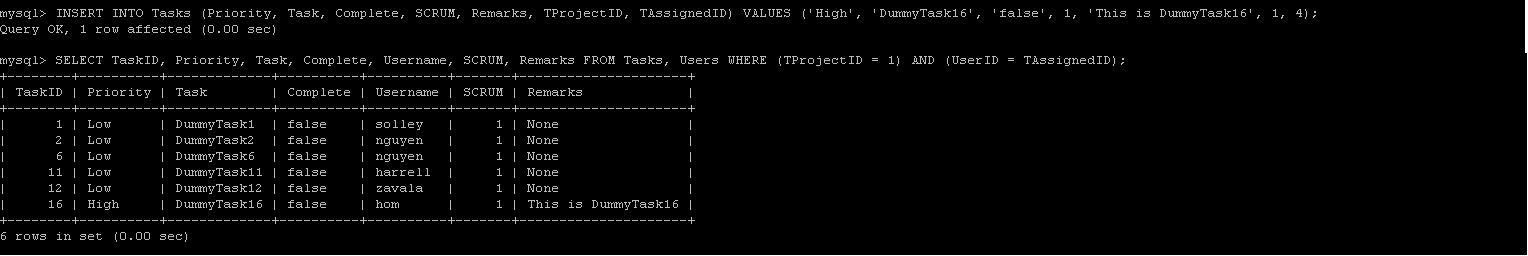
*Example: Populate the assigned users dropdown for ProjectID 1*

**

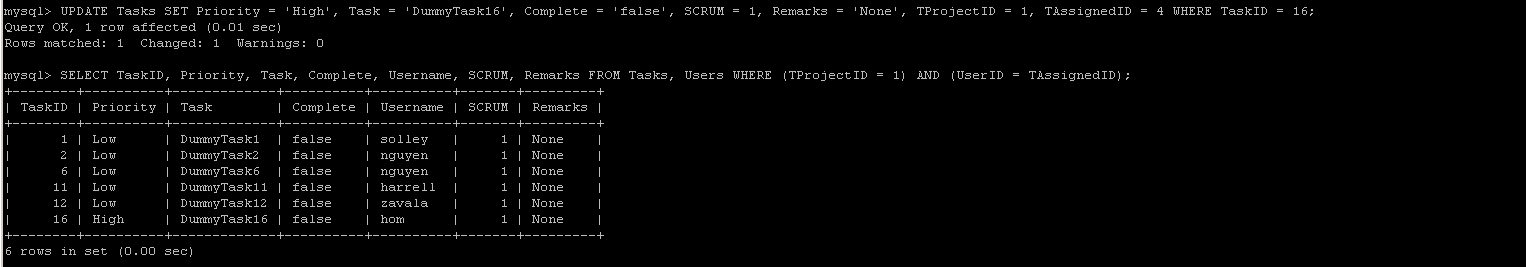
*Example: Get hom UserID*

**

*Example: Add DummyTask16 to ProjectID 1 and assign to hom*

**

*Example: Update DummyTask16 Remarks to None*

**

**Bugs**

To dynamically fill in the table on the page (sorted by BugID), run the query

**SELECT BugID, Users1.Username, Users2.Username, SourceFile, Location, Description, Status FROM Bugs, Users Users1, Users Users2 WHERE (BProjectID = [ProjectID\_Variable]) AND (Users1.UserID = BReporterID) AND (Users2.UserID = BAssignedID);**

then fill in the appropriate form fields on the webpage form.

To sort by Assigned Username the query would be

**SELECT BugID, Users1.Username, Users2.Username, SourceFile, Location, Description, Status FROM Bugs, Users Users1, Users Users2 WHERE (BProjectID = [ProjectID\_Variable]) AND (Users1.UserID = BReporterID) AND (Users2.UserID = BAssignedID) ORDER BY Users2.Username ASC;**

To sort by Reporting Username the query would be

**SELECT BugID, Users1.Username, Users2.Username, SourceFile, Location, Description, Status FROM Bugs, Users Users1, Users Users2 WHERE (BProjectID = [ProjectID\_Variable]) AND (Users1.UserID = BReporterID) AND (Users2.UserID = BAssignedID) ORDER BY Users1.Username ASC;**

To sort by SourceFile the query would be

**SELECT BugID, Users1.Username, Users2.Username, SourceFile, Location, Description, Status FROM Bugs, Users Users1, Users Users2 WHERE (BProjectID = [ProjectID\_Variable]) AND (Users1.UserID = BReporterID) AND (Users2.UserID = BAssignedID) ORDER BY SourceFile ASC;**

To sort by Status the query would be

**SELECT BugID, Users1.Username, Users2.Username, SourceFile, Location, Description, Status FROM Bugs, Users Users1, Users Users2 WHERE (BProjectID = [ProjectID\_Variable]) AND (Users1.UserID = BReporterID) AND (Users2.UserID = BAssignedID) ORDER BY FIELD (Status, 'Unassigned', 'High Priority', 'Med Priority', 'Low Priority');**

To populate the assigned and reporter user dropdown options, run the query

**SELECT Username FROM Users, ProjectMembers WHERE (PMProject = [ProjectID\_Variable]) AND (UserID = PMUserID);**

To get the chosen usernames UserID, run the query

**SELECT UserID FROM Users WHERE Username = '[Username\_Chosen]';**

and add to a temp variable for either Assigned or Reporter.

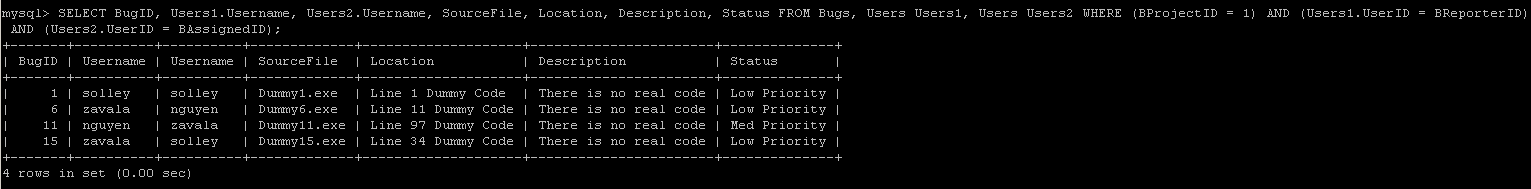
To add new bug, run the query

**INSERT INTO Bugs (SourceFile, Location, Description, Status, BProjectID, BReporterID, BAssignedID) VALUES ('[SourceFile\_Field]', '[Location\_Field]', '[Description\_Field]', '[Status\_Field]', [ProjectID\_Variable], [ReporterID\_Temp\_Variable], [AssignedID\_Temp\_Variable]);**

User can change all fields but BugID, upon clicking Submit run update query

**UPDATE Bugs SET SourceFile = (SourceFile\_Field), Location = (Location\_Field), Description = (Description\_Field), Status = (Status\_Field), BProjectID = [ProjectID\_Variable], BReporterID = [ReporterID\_Temp\_Variable], BAssignedID = [AssignedID\_Temp\_Variable] WHERE BugID = [ValueOf\_BugID\_Field];**

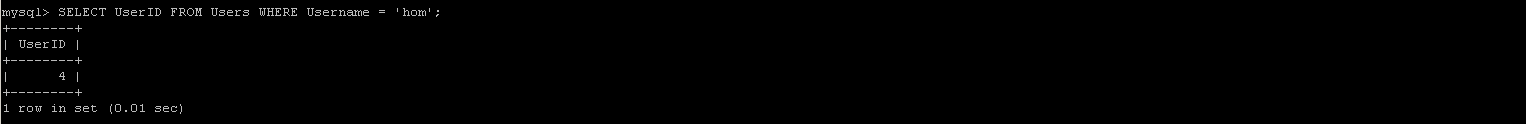
*Example: Get bug data for current project Dummy1, ProjectID 1*

**

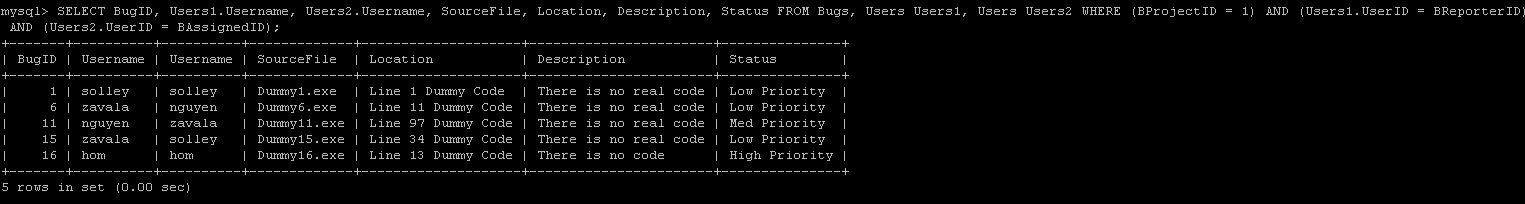
*Example: Populate the assigned users dropdown for ProjectID 1*

**

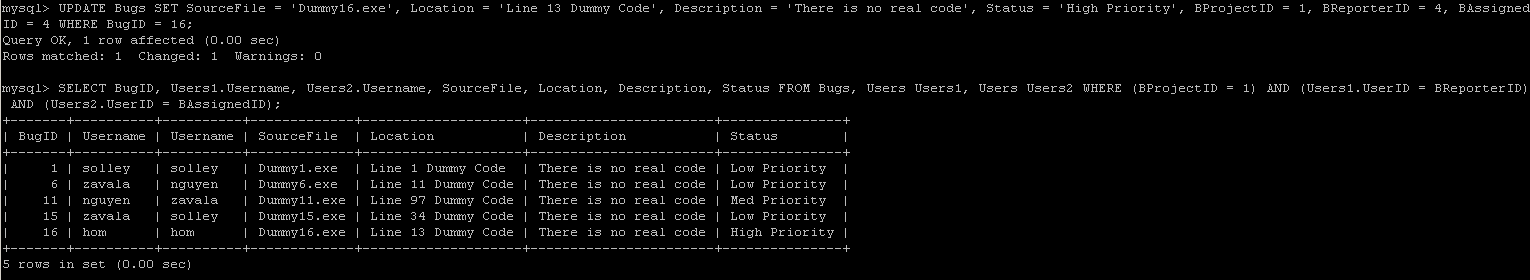
*Example: Get hom UserID*

**

*Example: Add Bug16 to Project1 both reported by and assigned to hom*

**

*Example: Update Bug16 description to match the previous entries*

**

**Forum**

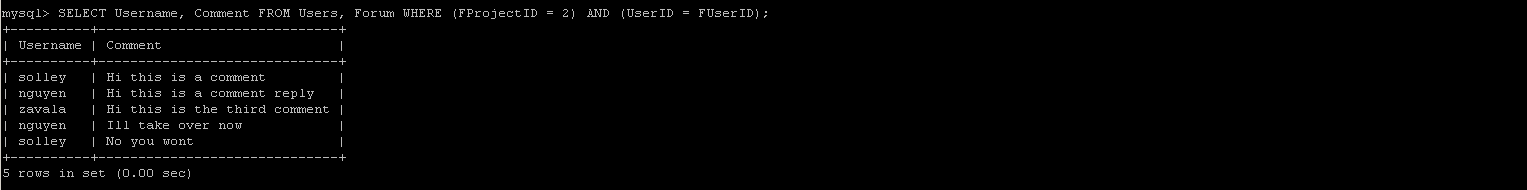
To dynamically fill in the table on the page (Oldest to Newest), run the query

**SELECT Username, Comment FROM Users, Forum WHERE (FProjectID = [ProjectID\_Variable]) AND (UserID = FUserID);**

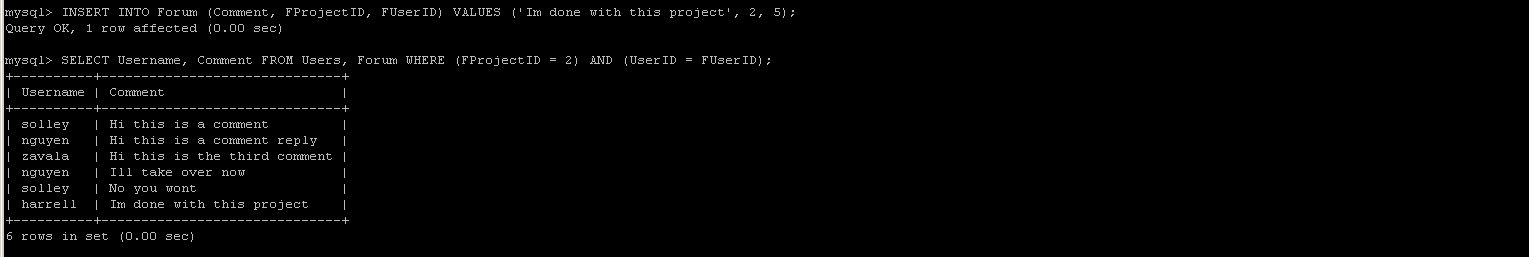
To add a comment, run the query

**INSERT INTO Forum (Comment, FProjectID, FUserID) VALUES ('[Comment\_Field]', [ProjectID\_Variable], [UserID\_Variable]);**

*Example: Get the forum data for current project Dummy 2, ProjectID 2*

**

*Example: Add a comment for current user harrell, UserID 5*



**Logout**

No SQL Calls, Delete Cookies