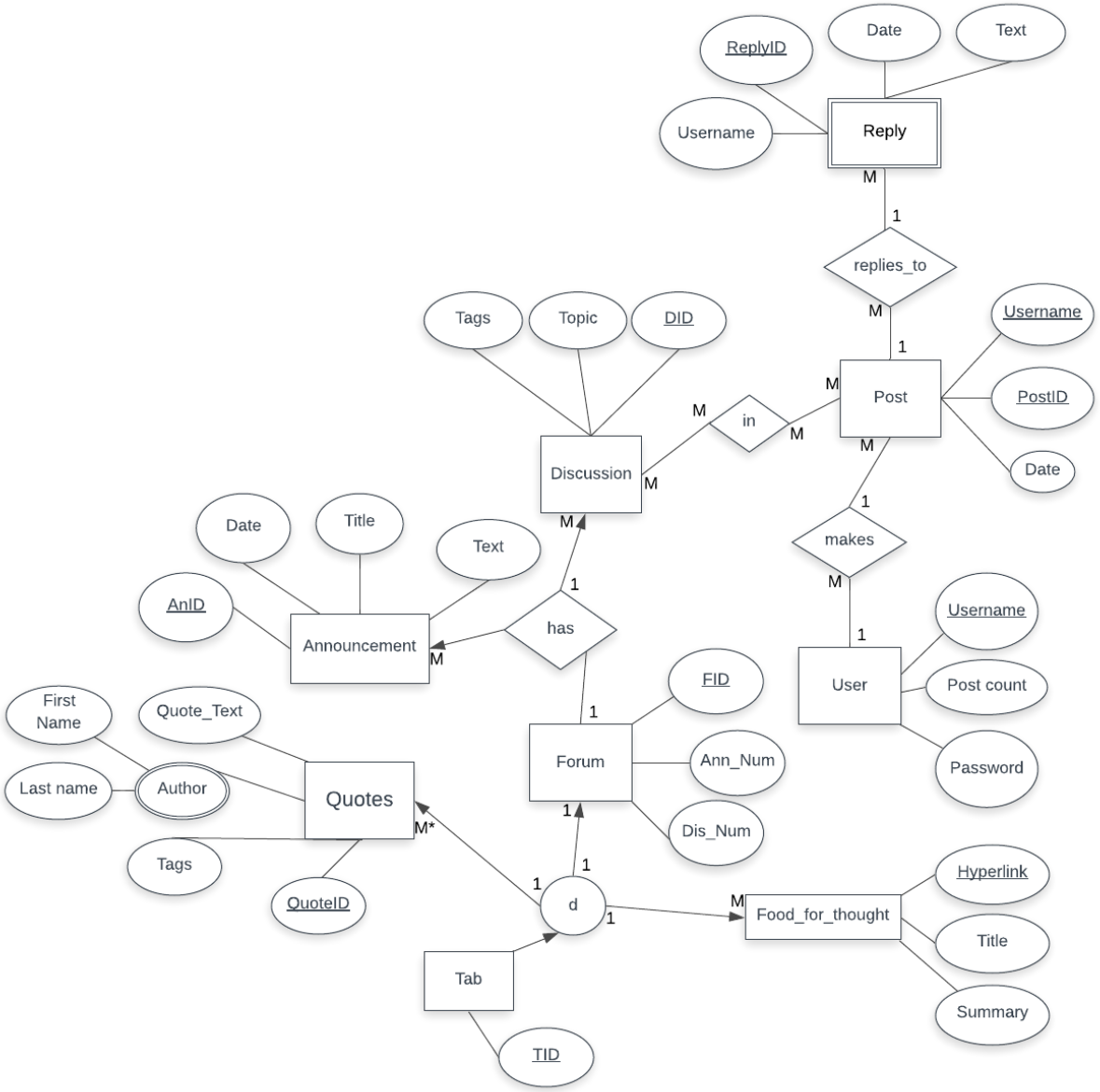
Group 2
Collin Pounds
Adrian Atanasov
Andrew Huhman

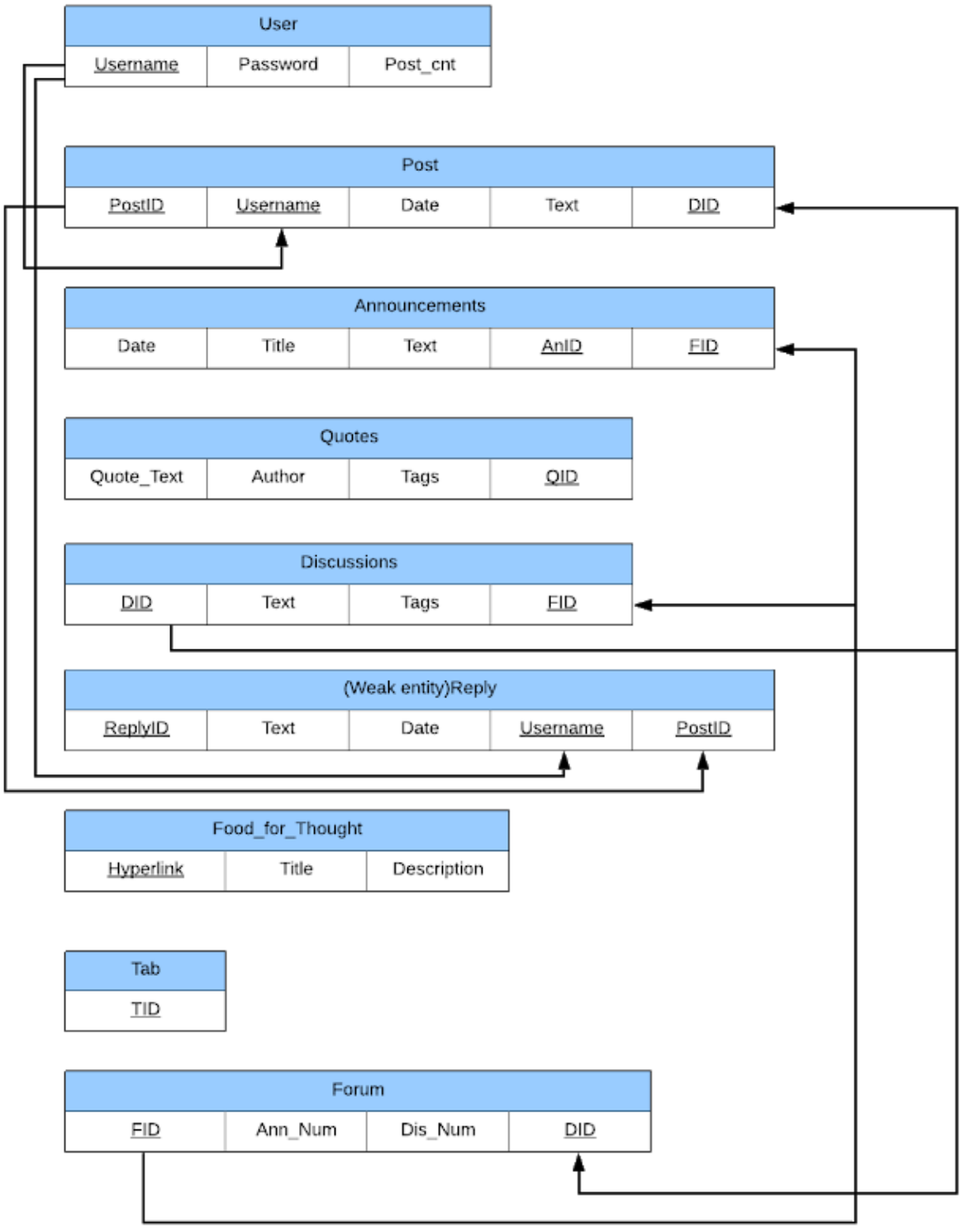
Project Report

Problem Statement: A website that offers conscious building educational material and a discussion based forum to facilitate the growth of humanity as a whole. A database system is essential to store all of the data that is provided on the site when navigating through it, and to query and store all the data from the forum. The database will also be used to store usernames and passwords of the users, allowing them to post on the site. However, users may navigate the site anonymously meaning a username and password are not needed. This system will store, manage, retrieve and manipulate all of the data on the site input by users. The website will provide the interface to see or change the data allowing the users to navigate through the database.

Conceptual Database Design:



Logical Database Design:



Discussions		
Attributes	**Type**	**Description**
DID	integer	Unique ID for discussion thread
Text	variable length string	Content of oroginal thread
Tags	String array	tags related to post for search query
FID	Integer	Forum ID

(weak entity)Reply		
Attributes	**Type**	**Description**
ReplyID	integer	starting at 0 going by order of date posted
Text	variable length string	Content of reply
Date	Date	Date posted
Username	string	Username of poster
PostID	integer	unique ID of post reply is to

Food_for_Thought		
Attributes	**Type**	**Description**
Hyperlink	variable length string	hyperlink of video embedded
Title	variable length string	Title of video or general idea of video
Description	variable length string	Description of what topic video will cover

Tab		
Attributes	**Type**	**Description**
TID	Integer	Unique ID of tab

Forum		
Attributes	**Type**	**Description**
FID	Integer	Unique ID of Forum
Ann_Num	Integer	Number of announcements in forum
Dis_Num	Integer	Number of discussions in forum

User		
Attributes	**Type**	**Description**
Username	variable length string	Unique id for each user
Password	variable length string	Users password
Post_cnt	integer	Number of posts made by user

Make_Post		
Attributes	**Type**	**Description**
PostID	integer	Unique id for each post
Username	variable length alphanumeric	The username of each user
Date	Date Field	The date of when the post was made
Text	variable length string	the text contained within the post

Announcements		
Attributes	**Type**	**Description**
Date	Date type	Date announcement posted
Title	variable length string	Title of announcement
Text	variable length string	Body of announcement
AnID	Integer	Unique ID of announcement post

Quotes		
Attributes	**Type**	**Description**
Quote_text	variable length string	Actual qoute
Author	variable length string	First and last name of quote originator
Tags	String array	tags that relate to quote
QID	Integer	Unique ID of quote

Application Program Design:

User Table:
CREATE TABLE `wbsitecs3380`.`Users` (`Username` VARCHAR(15) NOT NULL , `Password` VARCHAR(15) NOT NULL , `Post_cnt` INT NOT NULL , PRIMARY KEY (`Username`(15))) ENGINE = InnoDB;

Posts Table:
CREATE TABLE `wbsitecs3380`.`Posts` (`PostID` INT NOT NULL AUTO_INCREMENT ,`Username` VARCHAR(15) NOT NULL , `Date` DATE NOT NULL , `Text` TEXTNOT NULL , PRIMARY KEY (`PostID`)) ENGINE = InnoDB;

Set Foreign Key between User and Posts:
ALTER TABLE `Posts` ADD CONSTRAINT `Username` FOREIGN KEY (`Username`) REFERENCES `Users`(`Username`) ON DELETE NO ACTION ON UPDATE RESTRICT;

Annoucements:
CREATE TABLE `wbsitecs3380`.`Annoucements` (`Date` DATE NOT NULL ,`Title` VARCHAR(100) NOT NULL , `Body` TEXT NOT NULL , `AnID` INT NOTNULL AUTO_INCREMENT , PRIMARY KEY (`AnID`)) ENGINE = InnoDB;

Quotes:
CREATE TABLE `wbsitecs3380`.`Quotes` (`QID` INT NOT NULL AUTO_INCREMENT , `Author` VARCHAR(30) NOT NULL , `Quote_text` TEXT NOT NULL , PRIMARY KEY (`QID`)) ENGINE = InnoDB;

Discussions:
CREATE TABLE `wbsitecs3380`.`Discussions` (`DID` INT NOT NULL AUTO_INCREMENT , `Text` TEXT NOT NULL , `FID` INT NOT NULL , PRIMARY KEY (`DID`)) ENGINE = InnoDB;

Replies:
CREATE TABLE `wbsitecs3380`.`Replies` (`ReplyID` INT NOT NULLAUTO_INCREMENT , `Text` TEXT NOT NULL , `Date` DATE NOT NULL ,`Username` VARCHAR(15) NOT NULL , `PostID` INT NOT NULL , PRIMARY KEY(`ReplyID`)) ENGINE = InnoDB;

Link foreign keys Username and PostID to the reply Table:
ALTER TABLE `Reply` ADD CONSTRAINT `Link user` FOREIGN KEY (`Username`) REFERENCES `Users`(`Username`) ON DELETE NO ACTION ON UPDATE RESTRICT;

ALTER TABLE `Reply` ADD CONSTRAINT `PostID` FOREIGN KEY (`PostID`) REFERENCES `Posts`(`PostID`) ON DELETE NO ACTION ON UPDATE RESTRICT;

Food_for_Thought:
CREATE TABLE `wbsitecs3380`.`Food_for_Thought` (`HyperLink`VARCHAR(100) NOT NULL , `Title` VARCHAR(100) NOT NULL , `Description`TEXT NOT NULL , PRIMARY KEY (`HyperLink`(100))) ENGINE = InnoDB;

Forum:
CREATE TABLE `wbsitecs3380`. (`FID` INT NOT NULL AUTO_INCREMENT , `Ann_Num` INT NOT NULL , `Dis_Num` INT NOT NULL , PRIMARY KEY (`FID`)) ENGINE = InnoDB;

Functional SQL Statements

Inserting Users:
INSERT INTO `wbsitecs3380`.`Users` (`Username`, `Password`, `Post_cnt`) VALUES ('ahuhman', 'Cs3380!', 0);

Removing Users:
DELETE FROM `wbsitecs3380`.`Users` WHERE `Username` LIKE 'testuser' ESCAPE '#';

Searching for Users:
SELECT t.* FROM wbsitecs3380.Users t WHERE Username LIKE '%ctpx%'LIMIT 1;

Adding a Post:
INSERT INTO `wbsitecs3380`.`Posts` (`Username`, `Date`, `Text`) VALUES ('asa368', '2019-11-30', 'This is a test post that should be very short and simple.');

Adding a reply to a post:
INSERT INTO `wbsitecs3380`.`Replies` (`Text`, `Date`, `Username`, `PostID`) VALUES ('This is a test reply to the original post and I will be keeping it short and sweet', '2019-11-30', 'ctpx', 1)

Searching the database for keyphrase;
SELECT t.* FROM wbsitecs3380.Annoucements t WHERE Title LIKE '%test%' OR Body LIKE '%test%' LIMIT 1;

SELECT t.* FROM wbsitecs3380.Discussions t WHERE Text LIKE '%test%' LIMIT 1;

SELECT t.* FROM wbsitecs3380.Food_for_Thought t WHERE HyperLink LIKE '%test%' OR Title LIKE '%test%' OR Description LIKE '%test%' LIMIT 1;

SELECT t.* FROM wbsitecs3380.Posts t WHERE Username LIKE '%test%' OR Text LIKE '%test%' LIMIT 1;

SELECT t.* FROM wbsitecs3380.Quotes t WHERE Author LIKE '%test%' OR Quote_text LIKE '%test%' LIMIT 1;

SELECT t.* FROM wbsitecs3380.Replies t WHERE Text LIKE '%test%' OR Username LIKE '%test%' LIMIT 1;

SELECT t.* FROM wbsitecs3380.Users t WHERE Username LIKE '%test%' OR Password LIKE '%test%' LIMIT 1;

Adding a quote:
INSERT INTO `wbsitecs3380`.`Quotes` (`Author`, `Quote_text`) VALUES ('Adrian A.', '"How you do anything is how you do everything!"');

Deleting a quote:
DELETE FROM `wbsitecs3380`.`Quotes` WHERE `QID` = 2;

Creating a Forum:
INSERT INTO `wbsitecs3380`.`Forums` (`Ann_Num`, `Dis_Num`) VALUES (0, 0);

Deleting a Forum:
DELETE FROM `wbsitecs3380`.`Forums` WHERE `FID` = 2

Creating a Discussion:
INSERT INTO `wbsitecs3380`.`Discussions` (`Text`, `FID`) VALUES ('"Water"', 1);

Deleting a Discussion:
DELETE FROM `wbsitecs3380`.`Discussions` WHERE `DID` = 2

Adding Food_For_Thought:
INSERT INTO `wbsitecs3380`.`Food_for_Thought` (`HyperLink`, `Title`, `Description`) VALUES ('https://www.youtube.com/watch?v=jkLRith2wcc', 'Water Sounds', 'Testing Stuff');

Deleting Food_For_ Thought:
DELETE FROM `wbsitecs3380`.`Food_for_Thought` WHERE `HyperLink` LIKE 'https://www.youtube.com/watch?v=jkLRith2wcc' ESCAPE '#';

Creating an Annoucement:
INSERT INTO `wbsitecs3380`.`Annoucements` (`Date`, `Title`, `Body`) VALUES ('2019-11-30', 'Tests', '"Hello Crazy People!!!!!!!"');

Deleting an Annoucement:
DELETE FROM `wbsitecs3380`.`Annoucements` WHERE `AnID` = 1

Aggregate Function(Average Post):
SELECT AVG(Post_cnt) FROM Users

Gets post count per User
SELECT u.Username, COUNT(p.Username)
FROM Users AS u
LEFT JOIN Posts AS p ON u.Username = p.Username
GROUP BY u.Username;

Number of users
SELECT COUNT(Users.Username) FROM Users;

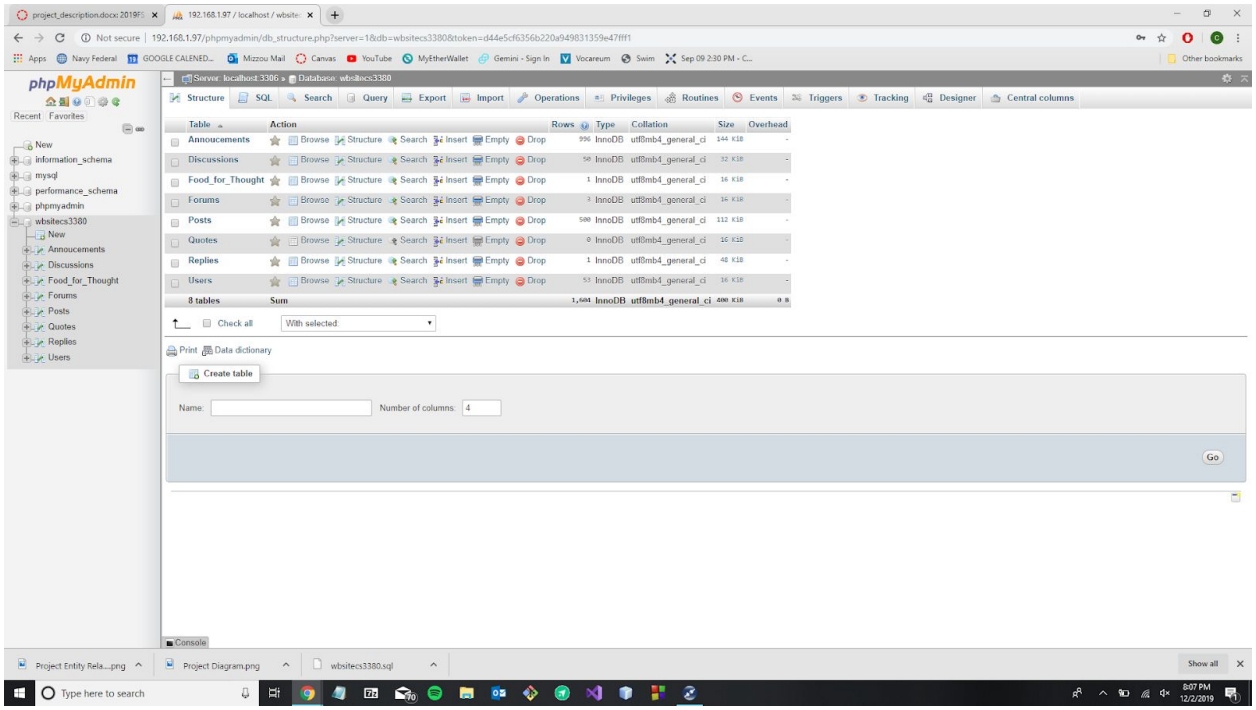
Gets number of posts per discussion
SELECT d.DID, COUNT(f.DID)
FROM Discussions AS d
LEFT JOIN Posts as f ON d.DID = f.DID
GROUP BY d.DID

Get number of discussions per Forum
SELECT h.FID, COUNT(g.FID)
FROM Forums h
LEFT JOIN Discussions as g ON h.FID = g.FID
GROUP BY h.FID

User Manual:

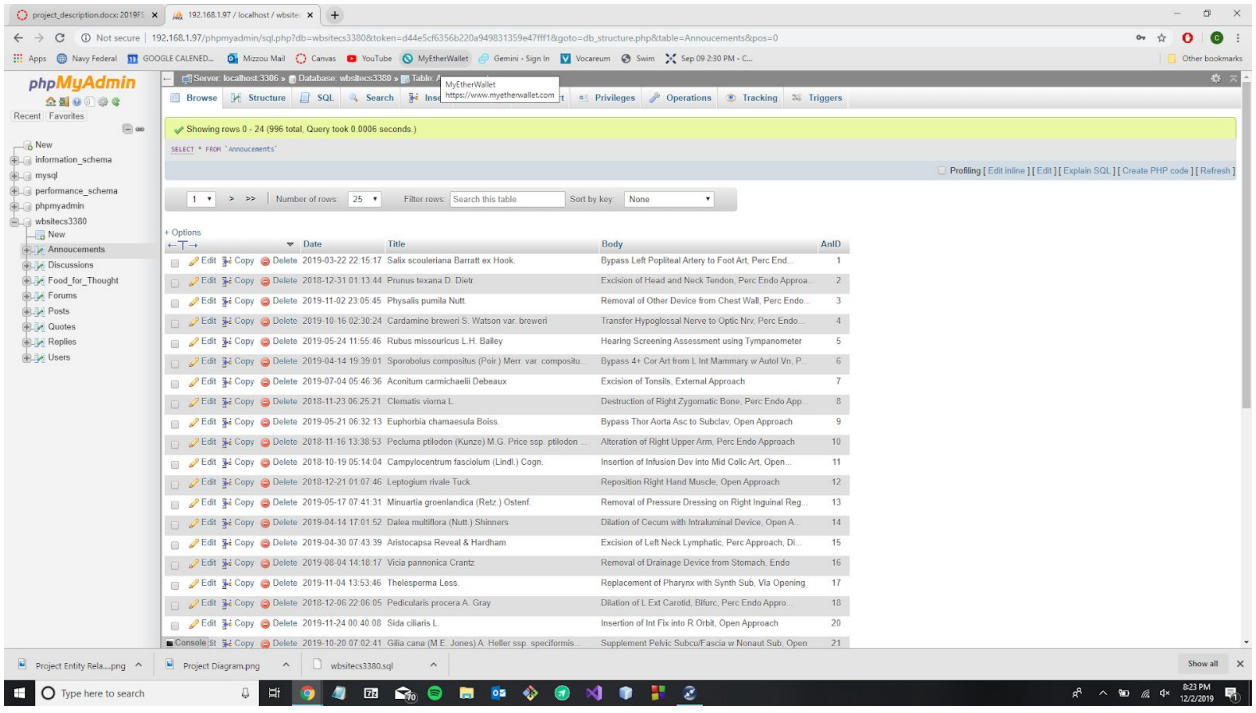
The first step to using this database is to import the zipped source .sql file into your database management client. Once you have imported the zipped file, you will be able to access the data stored and organized into 8 different tables. The 8 tables are named Announcements, Discussions, Food_For_Thought, Forums, Posts, Quotes, Replies, Users. Below will describe the process to change the properties of the whole relation. On the left side of your window, you should see database “wbsite3380” which you should be able to click the plus button next to it.

This shows you a dropdown of the 8 relations. If you click on the “wbsite3380” you will see:

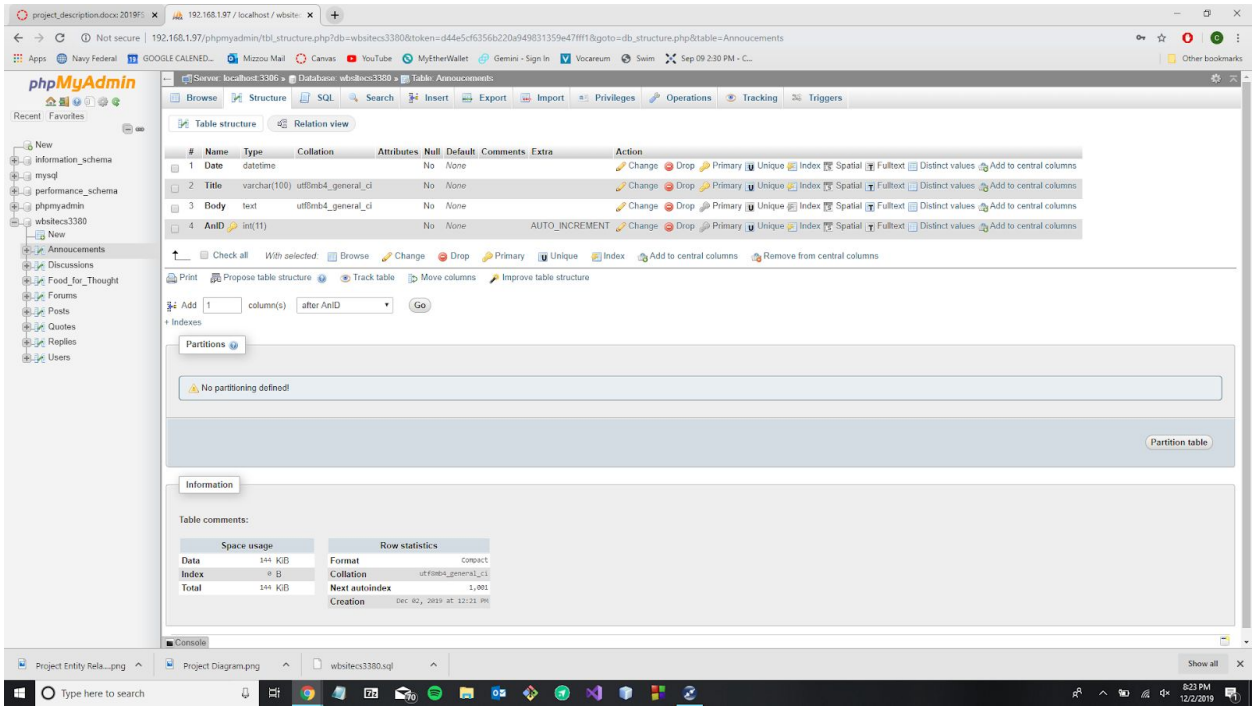


This shows you a list of each relation and several options for each one. Options include: Browse, Structure, Search, Insert, Empty, and Drop.

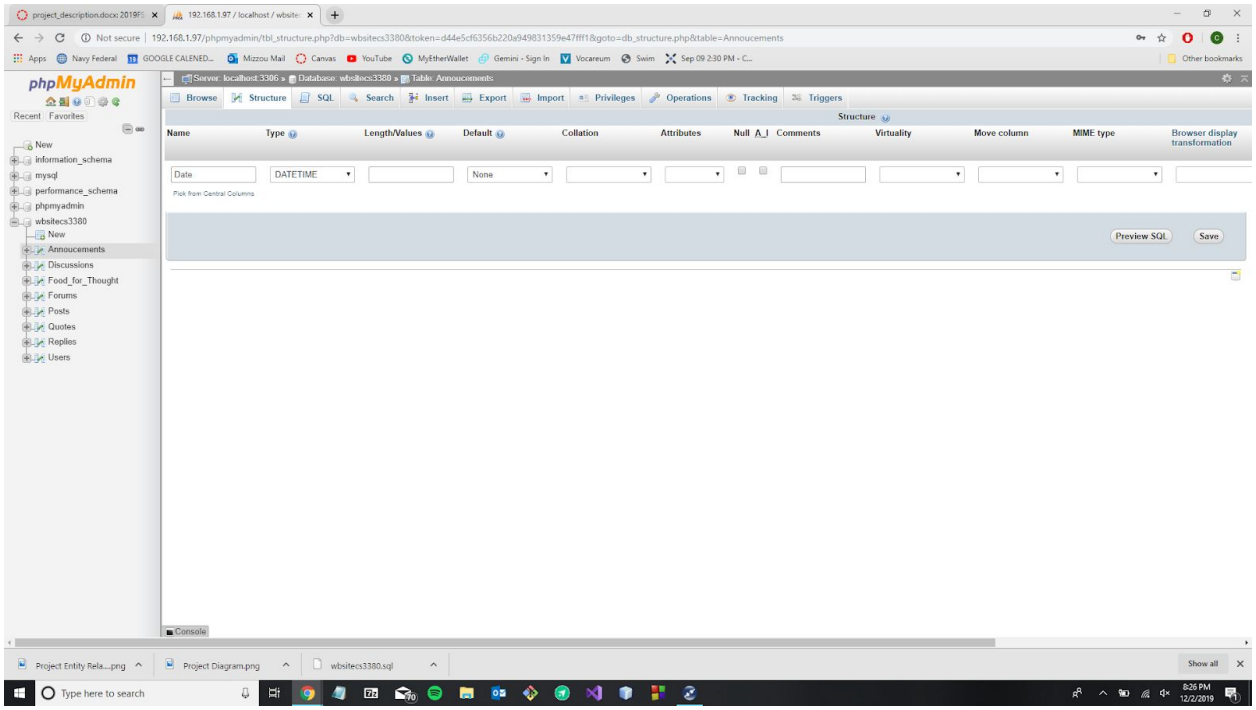
If you select a relation/click any of the options next to the relation names, it will change the tabs above to reflect which one you selected:
Browse:



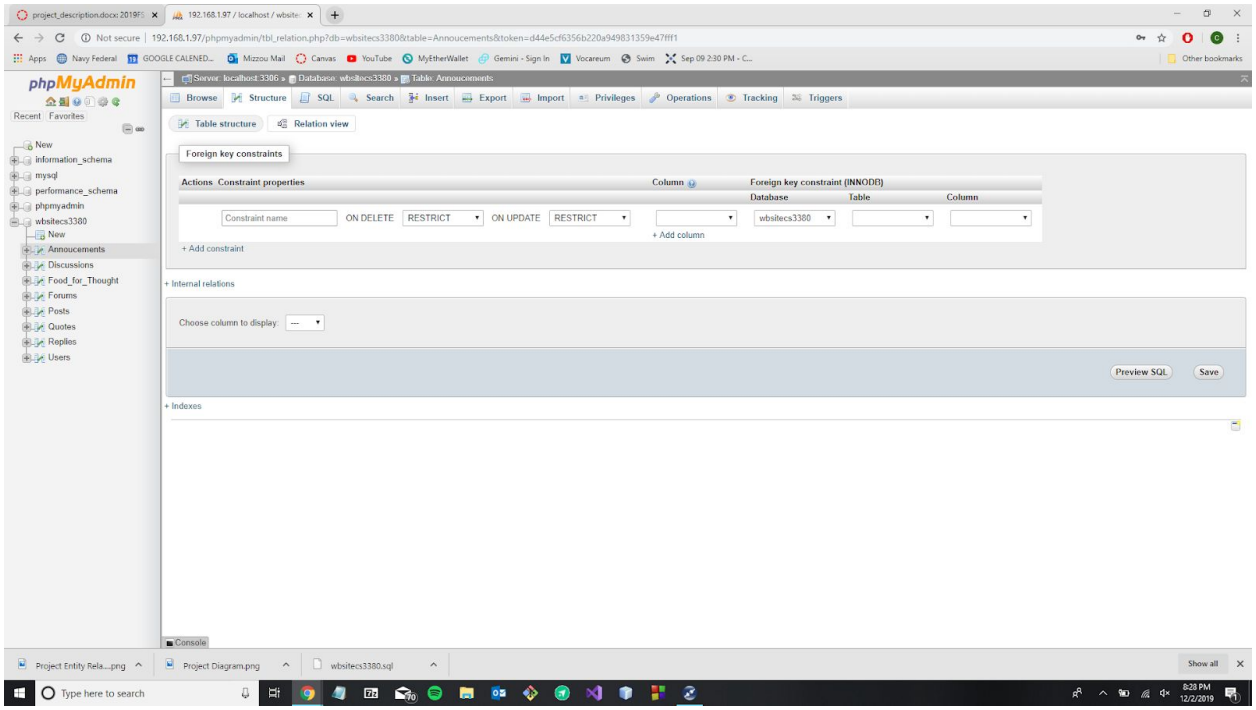
Structure:



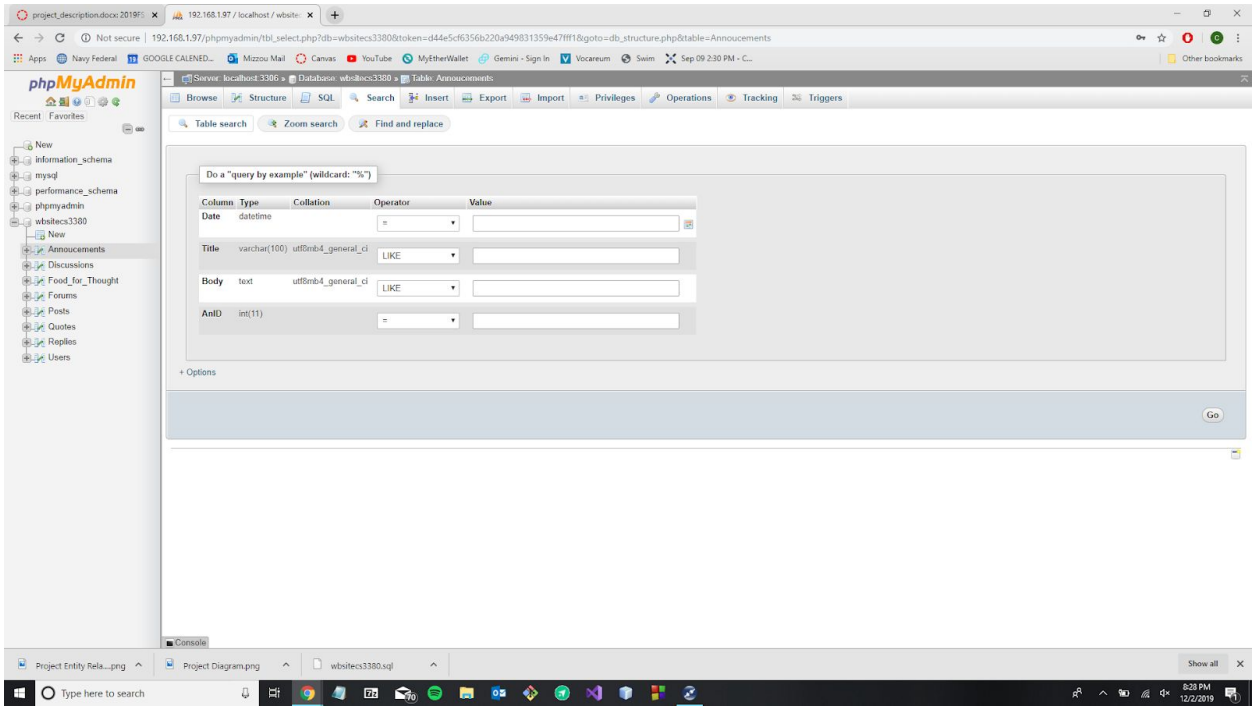
In this tab you can make many changes to the relation that will change the properties of the entire relation by selecting the checkbox next to one of the column names, then clicking “change” below.



You can set a column as the primary key, along with many other things like clicking “Relation view” and adding information about the tables foreign key constraints.

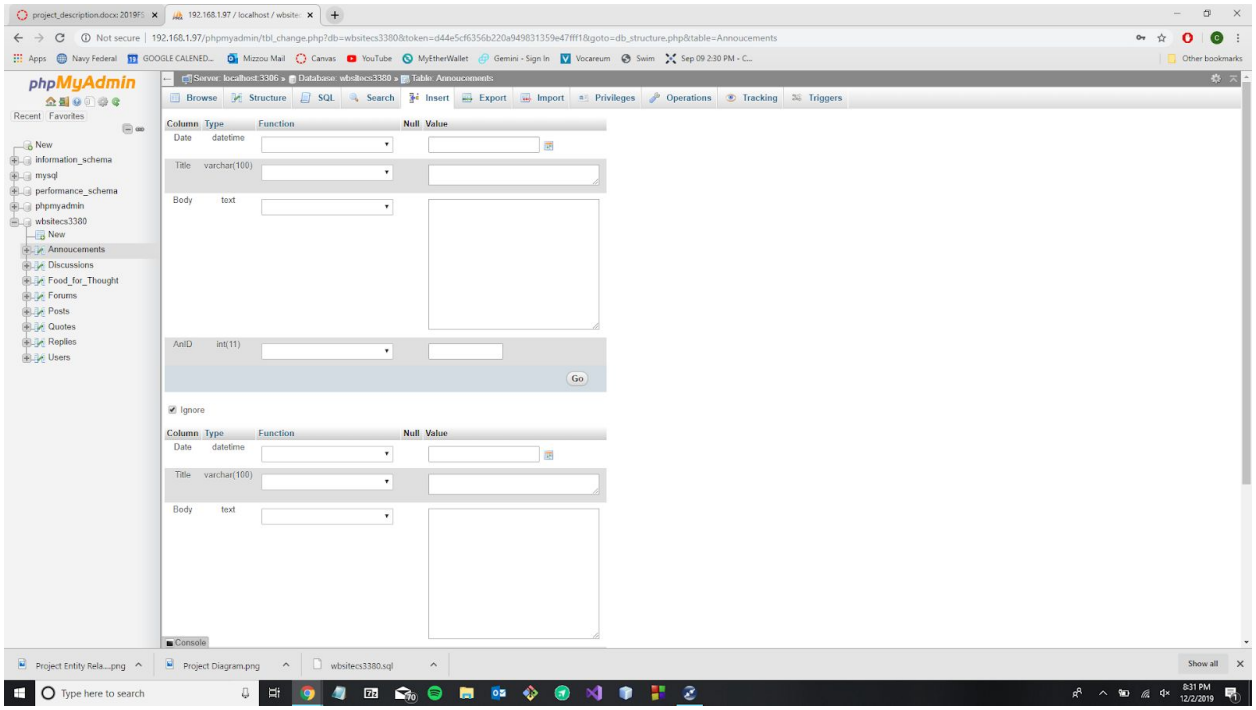


Search:



Makes it possible for you to search for specific values of the different attributes of the relation to look for a specific tuple, without needing to write a query.

Insert:

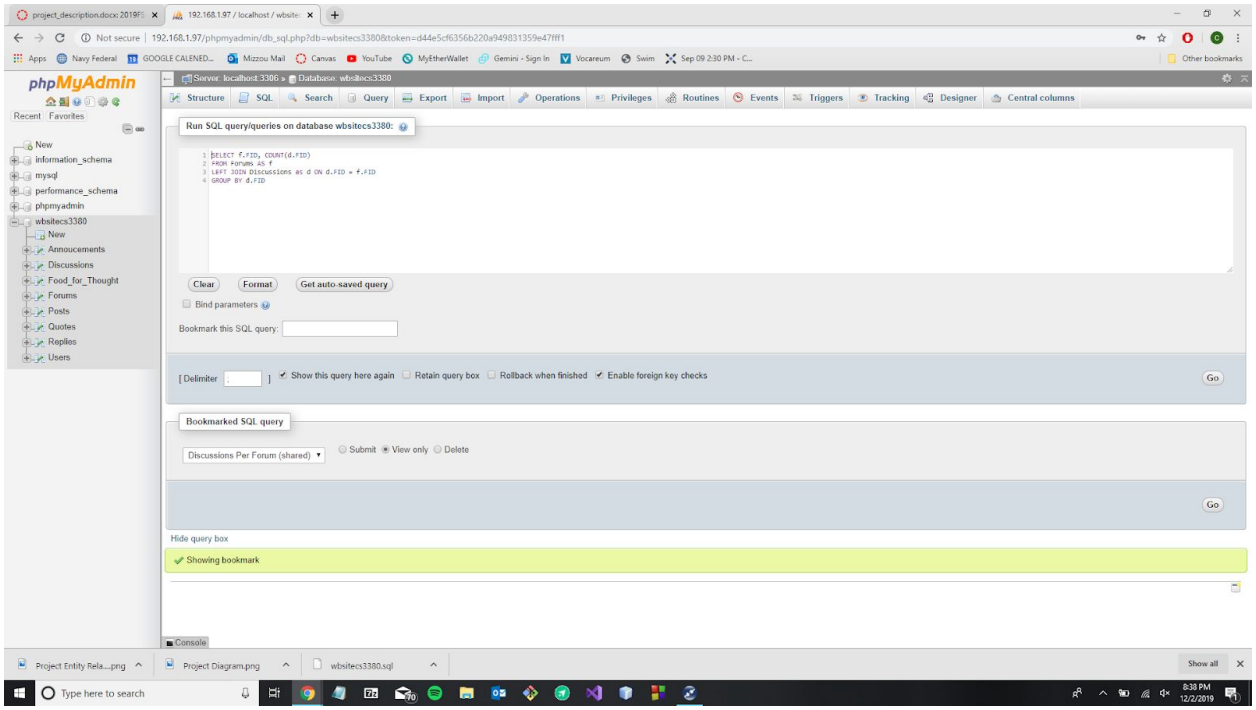


The insert button will bring you to this screen, where you can add tuples to the table, and make them follow a certain function using a dropdown menu that will retrieve a value or modify the value to be encrypted, for example.

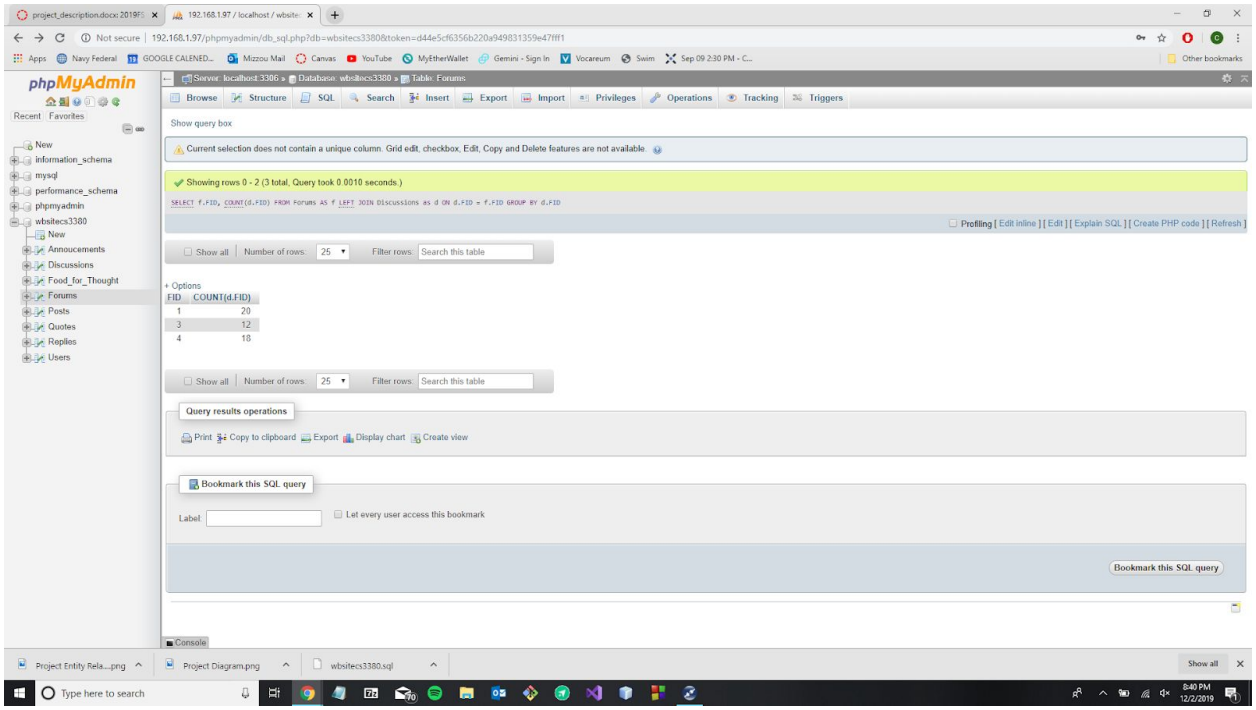
Empty: This selection will result in a Truncate of the entire table you select. It will empty the entire table, but keep the table structure in your database.

Drop: This will completely delete the table from your database, along with its data that was stored and its constraints.

Once you are back to the main database screen, you can see a tab labeled “SQL” that is the second tab. This is a terminal where you can write SQL queries and they will be executed on the database.
An example shows:

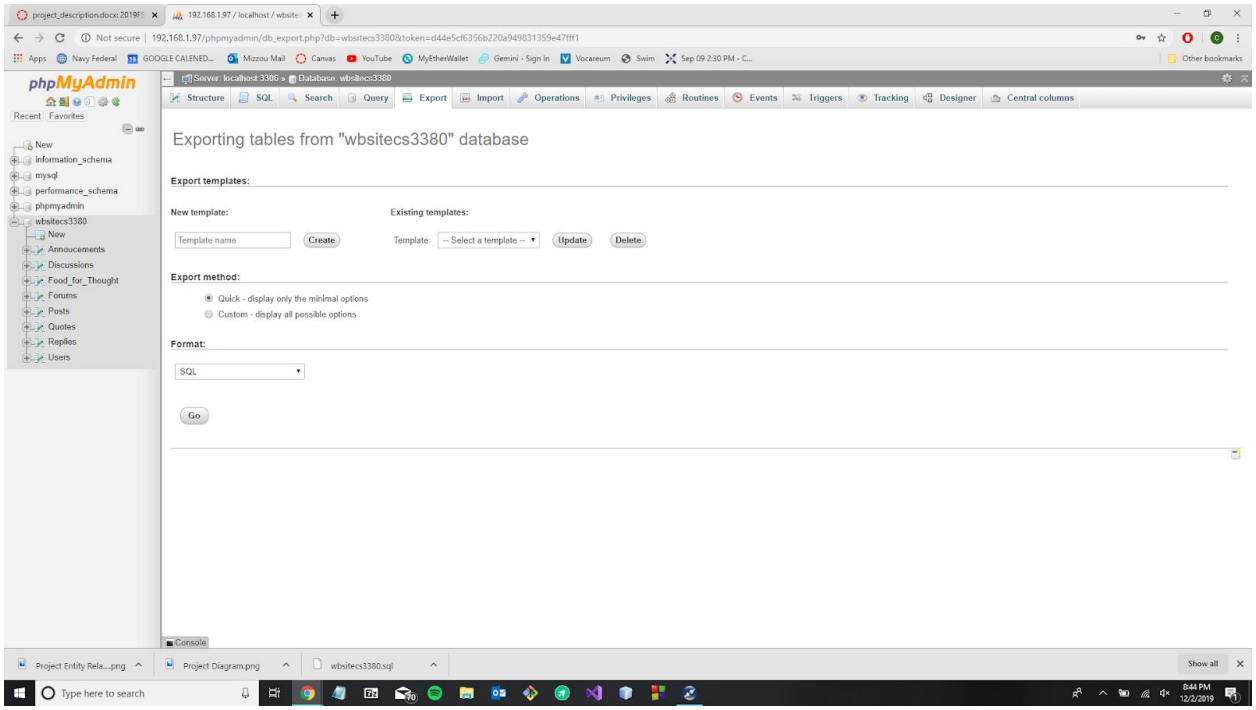


Where an SQL query finds how many discussions each forum contains. You can run this query simply by clicking the associated “Go” button in the bottom right.
When this query is ran, you can see the displayed information below:



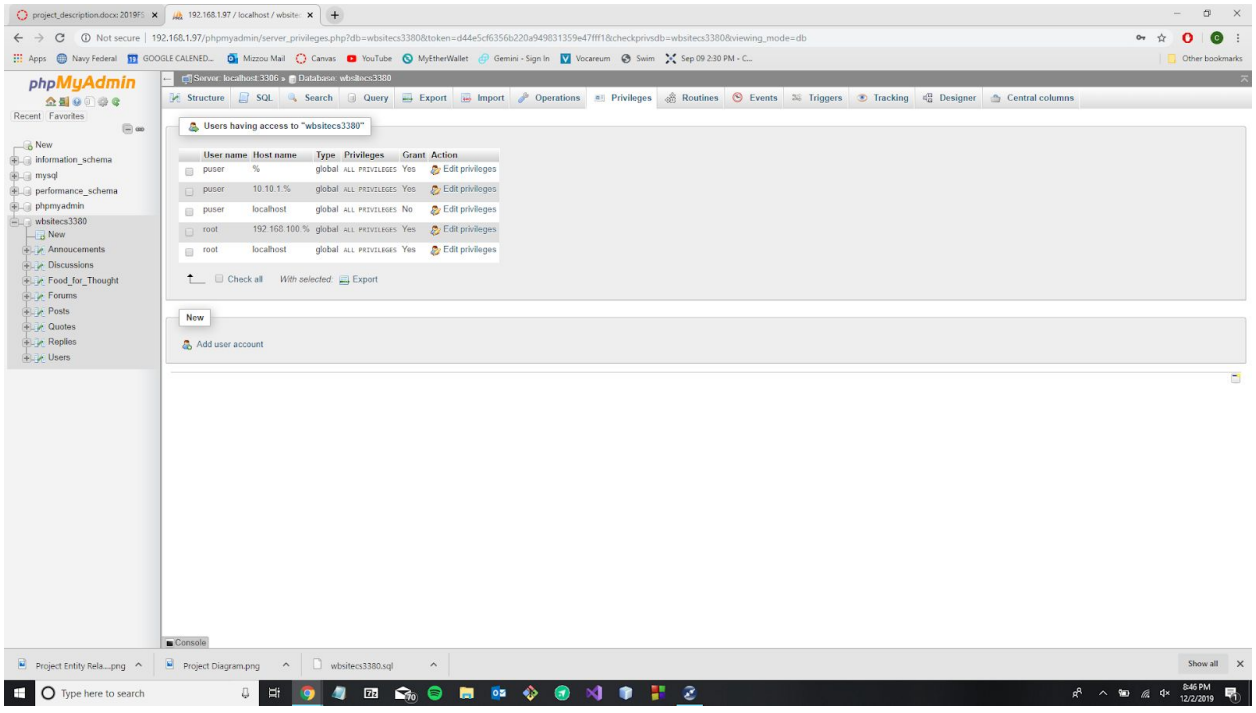
Here you can see each forum ID, and its associated count of discussions within it. If your query has more than the specified number of rows to display, you can change that with the drop down right above the query information.

There is an Export tab where you can export a relation or the entire database to SQL source code. This is useful to transfer large relations between databases.



Along with export, there is also an “Import” tab. This makes it possible to import zip files containing SQL or other database languages into your database.

There are several other tabs as you can see that offer other tools to modify your database such as changing database admin privileges:



All these functions are used to modify the database data and how things are structured. You can see that knowing the GUI of the database software is all you need to know to start adding, deleting, and modifying tuples and relations of the database.