Chapter

13

Introduction to XAMPP, MySQL, PHP & JSON

At the end of this chapter, you will be able to:

- Describe FriendsZone
- Setup XAMPP
- Use phpMyAdmin
- Run PHP codes
- Read JSON

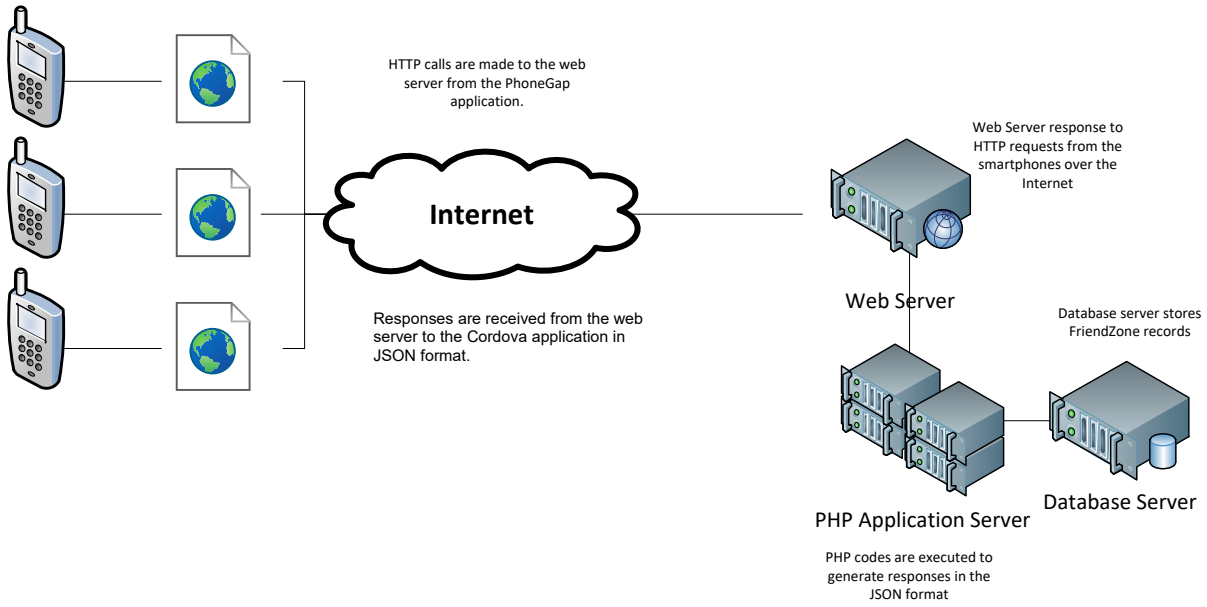
Instruction 1: Describe FriendsZone

1. From chapter 14 onwards, we will be developing FriendsZone. FriendsZone is a Cordova application. FriendsZone is a "cross" between FaceBook and Twitter. A user can create his profile on FriendsZone and search for users to make friends with. Friends can view one another's profiles. A user can also "shout" on his wall. When he shouts, his friends can see it. He can also update his current location for all to see.

2. There are 2 components of FriendsZone. The client component is a Cordova application which we will be developing in the next few chapters.

   The server components of FriendsZone update a database server that contains information such as profiles, wall posts and relationships between FriendsZone users. The server components are written in PHP. The codes have been developed and provided to you. You will be writing codes in Cordova to make calls to these FriendsZone server components. You will learn how to make these calls in chapters 14 to 17.

3. The illustration below describes the architecture of FriendsZone.



Instruction 2: Setup XAMPP

1. To serve requests from our FriendsZone Cordova application, we need the following server applications:

   a. MySQL Database Server: The database server stores our FriendsZone data.

   b. Web Server: The web server will serve requests over a HTTP connection that our Cordova application will use to make calls to the server.

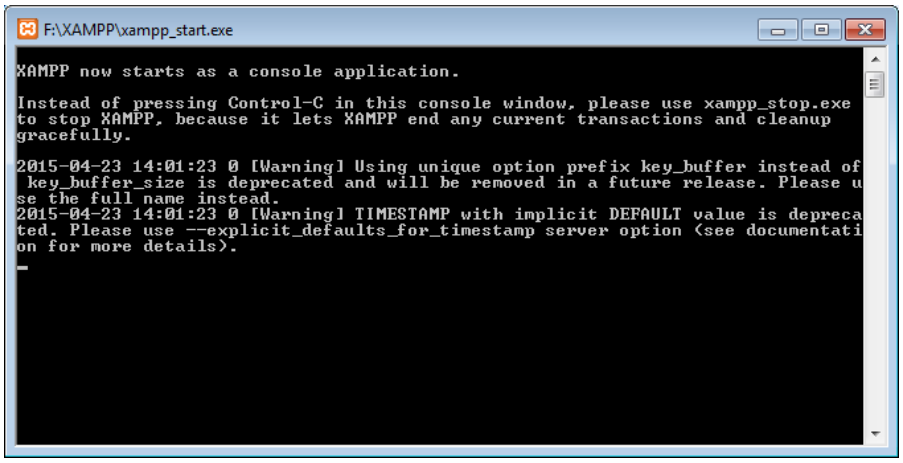
   c. PHP Application Server: The PHP Application Server executes PHP web services that will deliver contents to and from the Cordova application.

2. Fortunately, these servers can be set up in a single step with XAMPP. XAMPP is a popular 4-in-1 package that consists of a webserver, a PHP application server, a Pearl Interpreter and a database

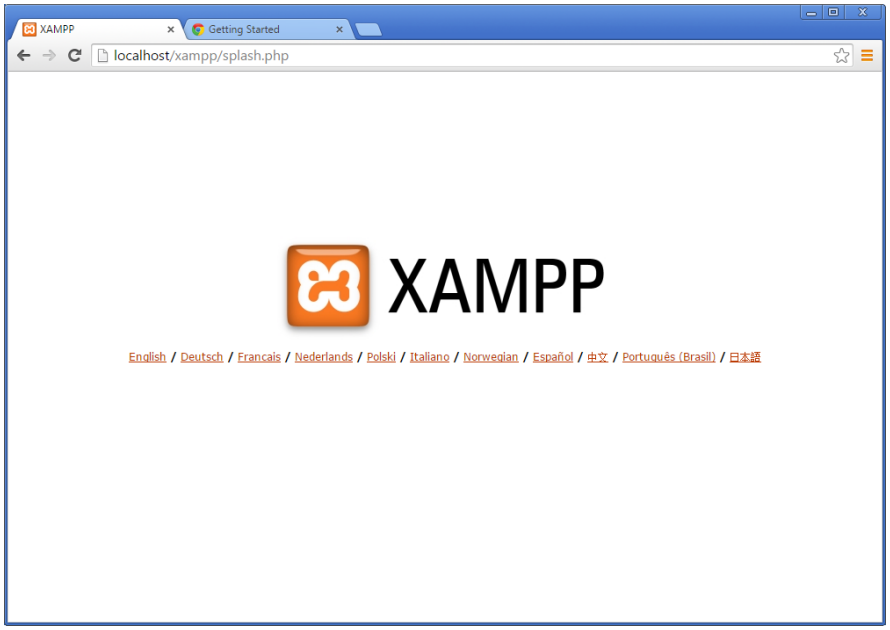
server. Pearl is another popular web programming language that is beyond the scope of this exercise. Everything we need comes as a package called XAMPP that runs on your computer. XAMPP can be downloaded and installed at https://www.apachefriends.org/index.html.

3. Install XAMPP.

4. Run xampp_start.exe. This will start the XAMPP servers that consist of the Apache Web Server, the MySQL Database Server and all other related services. Keep the window open throughout your session.

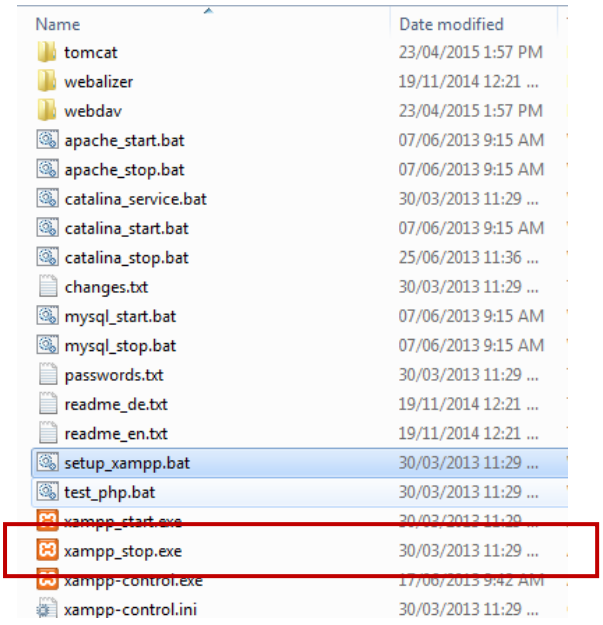


5. Open Google Chrome and visit http://localhost. XAMPP has been successfully started if you see the screen below. Click "English" to go to the main XAMPP screen.



6. Create a new folder "friendszone" in \xampp\htdocs\. Download the PHP codes for friendszone from the GitHub repository of this chapter from the \PHP folder and copy them into the "friendszone" folder. We will be making calls to these PHP pages using the FriendsZone Cordova application.

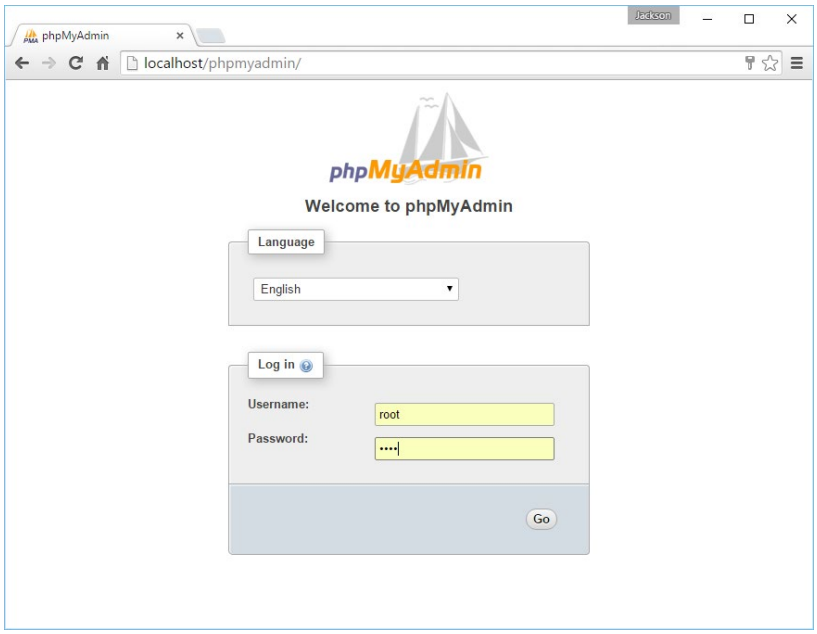
7. When you have completed your work, remember to shut down XAMPP by running xampp_stop.exe. Always ensure that both the windows for xampp_start.exe and xampp_stop.exe have closed.



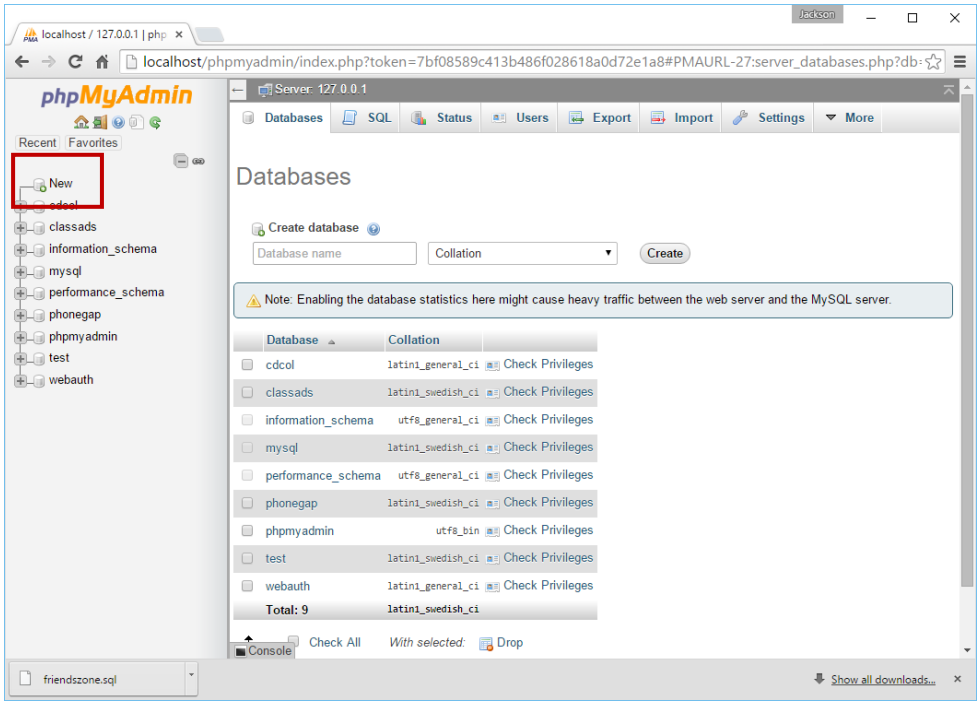
Instruction 3: Use phpMyAdmin

1. phpMyAdmin is a free web-based software to handle MySQL Database administration. Database operations such as adding new databases, creating and changing database table design, updating records and managing MySQL user access rights can be done with phpMyAdmin.

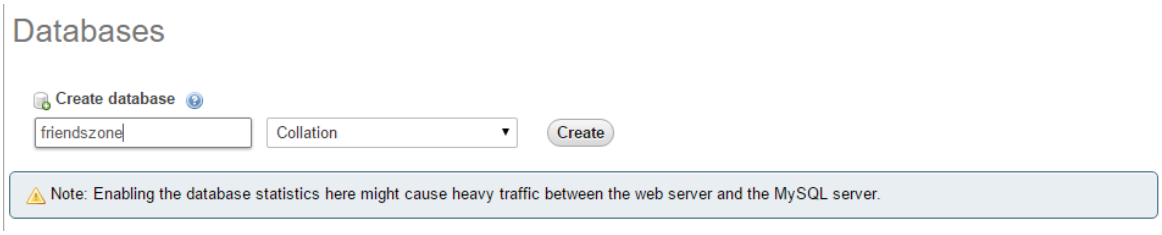
2. phpMyAdmin is a component of the XAMPP server. To access it, visit http://localhost/phpmyadmin. Enter the username and password that you have configured for phpMyAdmin when you installed XAMPP in Instruction2 Step 3 and click [Go] to continue.



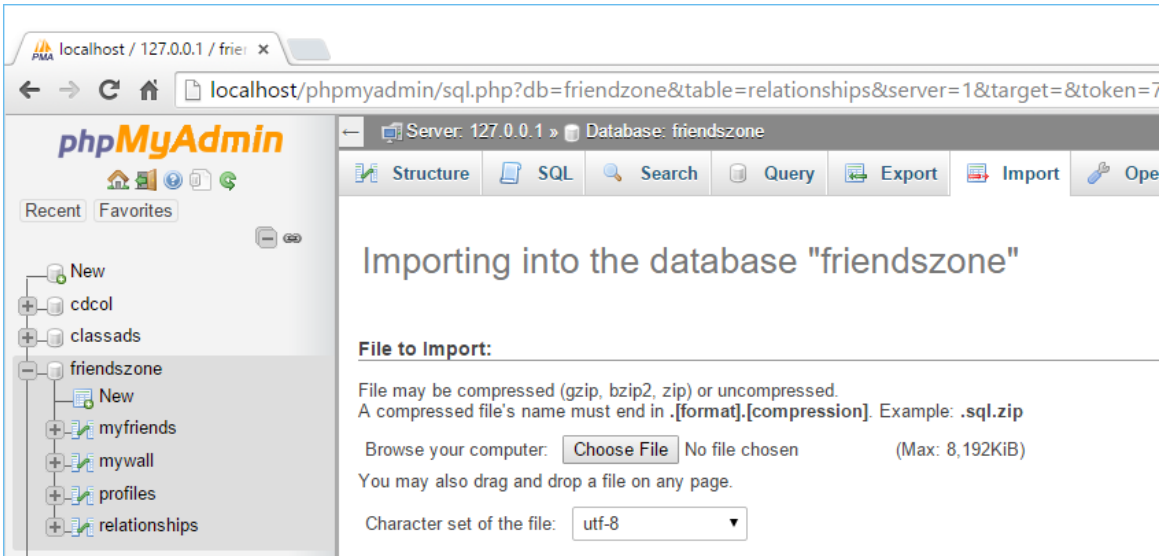
3. Our FriendsZone application's database will be stored in MySQL server. We will now create a database for this application. Click [New].



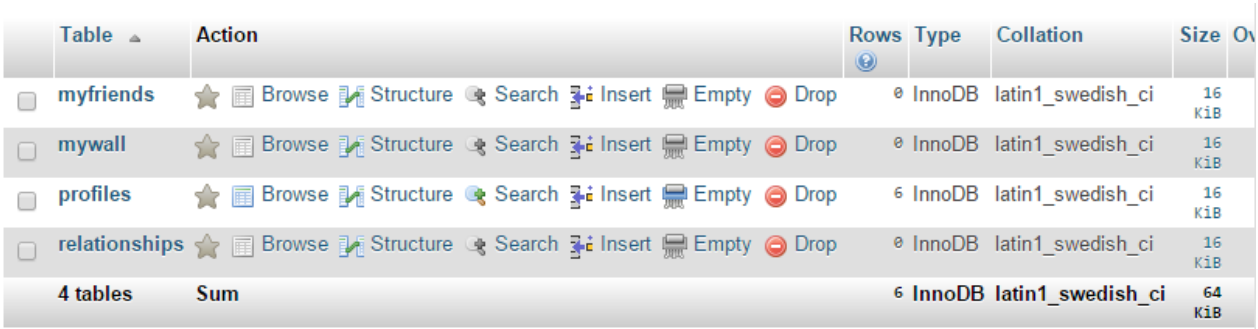
4. Enter "friendszone" as the database name.



5. Click the new friendszone database that you have just created. Notice that there are no tables in this database. A set of SQL statements to create the tables for the friendszone application can be found the Github repository VC13-MySQL-JSON. Click on "Import" in the top menu. Click on the [Choose File] button to choose the file friendszone.sql. Click [Go] to begin the import process.



6. Click on the "friendszone" database again. Notice that 4 tables have been created.



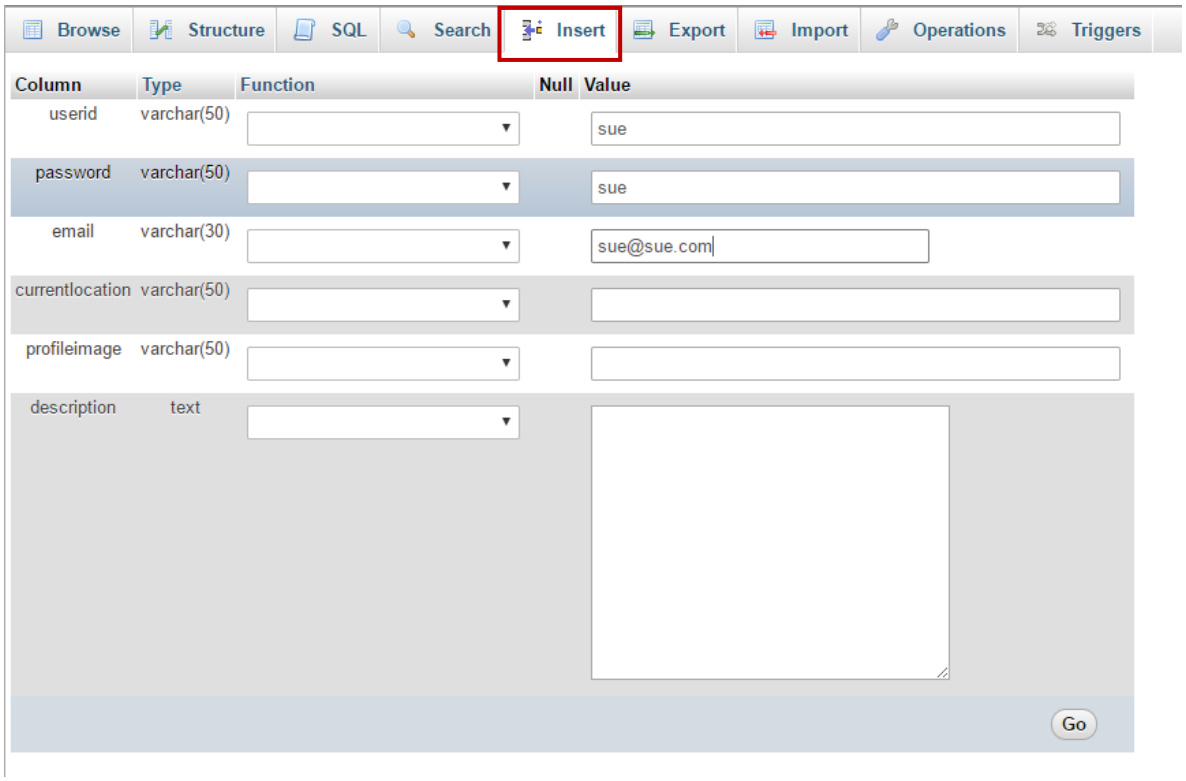
7. These are the descriptions of the FriendsZone tables:

Table	Description
myfriends	Records of the friends of each profile found in the FriendsZone application.
mywall	Records of every wall post belonging to each profile.
profiles	Username, password, description and picture of each FriendsZone profile.

relationships	Records of the types of relationships between FriendsZone profiles for example, acquaintance, family and friends.

8. The tables that you have just imported are empty.

9. To insert a new record in the profiles table, click [Insert] in the top menu. Enter the values of the record that you wish to insert and click [Go] to confirm the insertion.



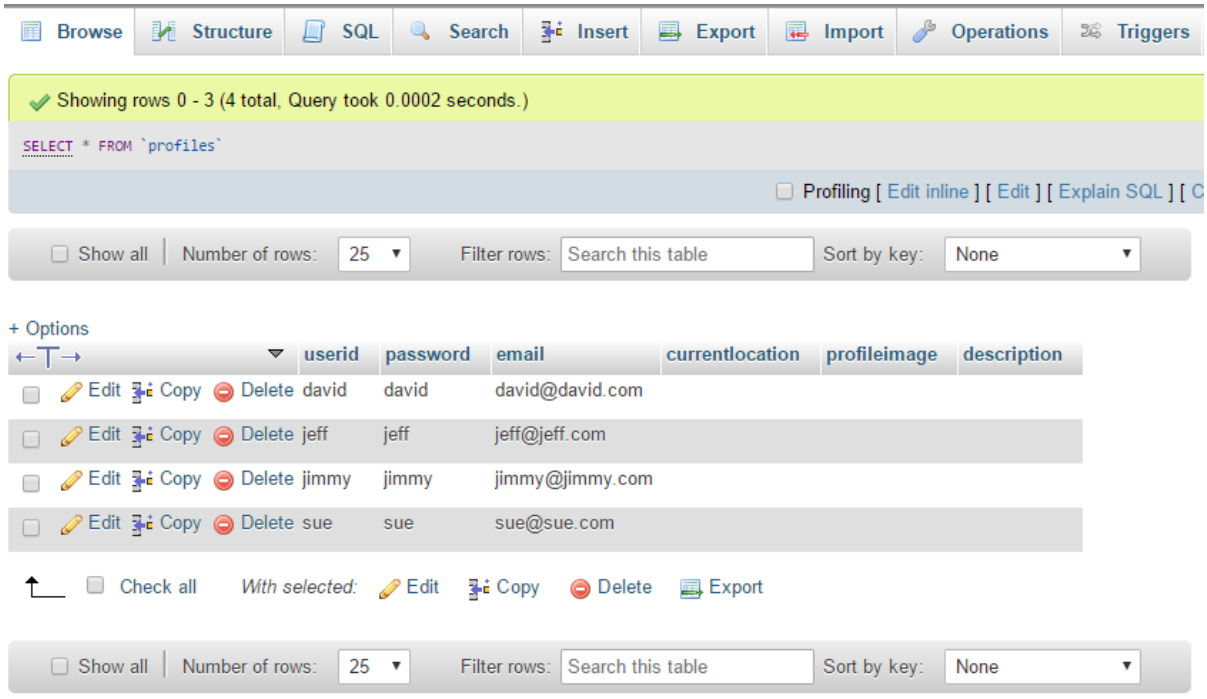
10. Add more records to the profiles table:

    a. userid: david, password: david, email: david@david.com

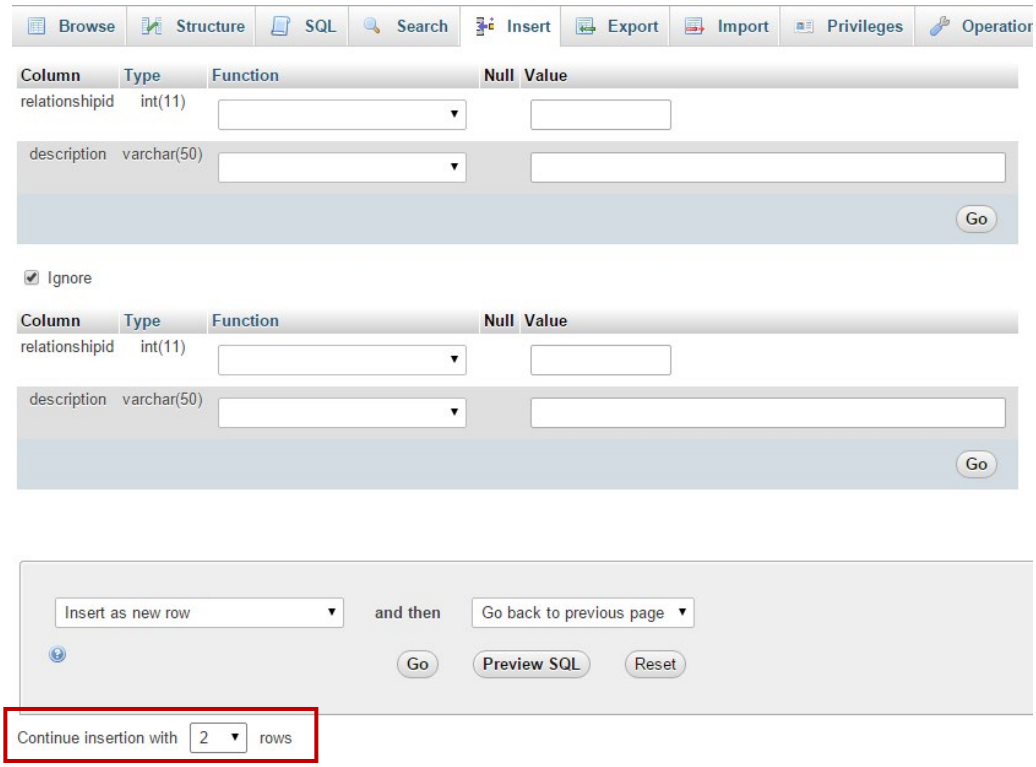
    b. userid: jeff, password: jeff, email: jeff@jeff.com

    c. userid: jimmy, password: jimmy, email: jimmy@jimmy.com

11. Click profiles to display the records in this table. You may click [Edit], [Copy] and [Delete] on each record to perform the corresponding actions on them. The check boxes on each record and the "Check All" button allows you to edit 2 or more records collectively.



12. We wish to insert records to the relationships table. Click the relationships table, followed by the [Insert] button in the top menu. Change the "Continue insertion with [2] rows" drop down box to 5. Notice that 5 sets of empty forms have been created for your entry.

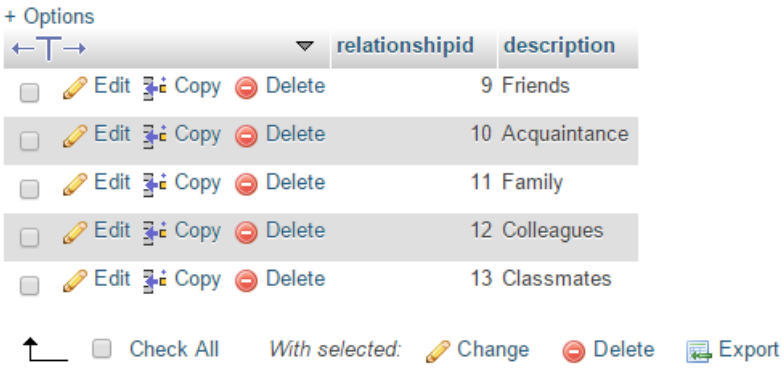


13. You only need to enter the description field. Leave relationshipid blank as it will be entered automatically because the relationshipid field has been set to auto-increment as new records are entered. Enter the following descriptions. Click [Go] on the last entry to confirm the insertion.

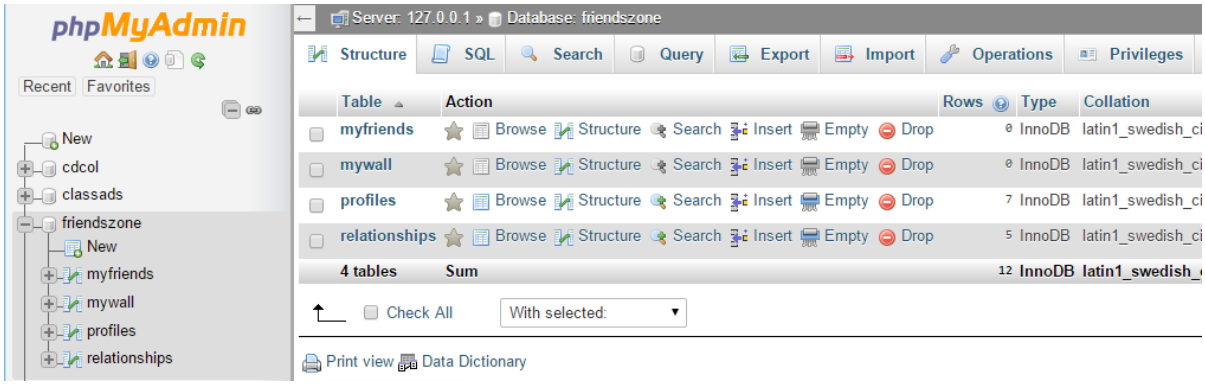
    - Friend

- Acquaintance
- Family
- Colleagues
- Classmates

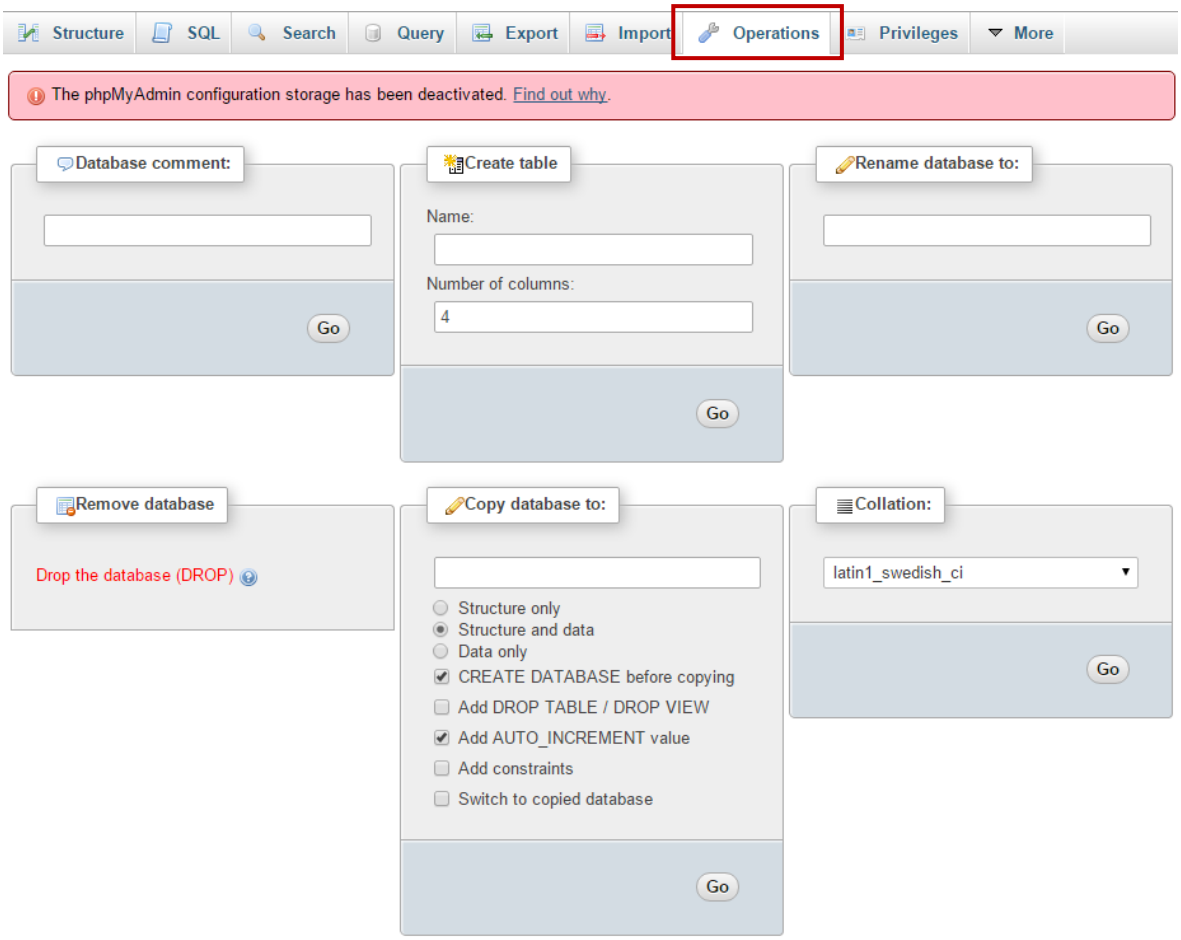
14. Click on the relationships table again. Notice that the records have been inserted and the relationshipid fields of each record has been filled automatically.



15. To drop a table in a database, click on the database's name in the list of databases on the left. Click [Drop] at the table that you wish to drop.



16. To perform operations such as dropping an entire database, adding a new table or renaming an existing database, click on [Operations] on the top menu.



Instruction 4: Run PHP codes

1. The FriendsZone Cordova application will be making calls to the FriendsZone web services to retrieve, update and insert records. Each of these php file perform a specific task. Their functionalities are described below. For more details of how these web services work, refer to Appendix 5: Business Logic in PHP.

PHP Filename	Description
viewwall.php	Views the wall post of a profile in FriendsZone.
getrelationships.php	Retrieve a list of relationships.
newfriend.php	Create a new record that states that one profile is a friend of another profile.
searchfriends.php	Search profiles based on user ID or user description.
savenewwallpost.php	Adds a new wall post.
savenewdescription.php	Update a profile with a new profile description.
savenewpassword.php	Change a profile's password.

savenewimage.php	Change a profile's profile image.
getprofile.php	Retrieve a profile's details.
savenewlocation.php	Update a profile with its current location.
getcurentllocation.sphp	Retrieve a profile's current location.
newuser.php	Create a new user record.
login.php	Checks the authenticity of a profile with his user ID and password.
deleteimg.php	Delete a profile's existing profile image.
upload.php	Upload a profile's image into the friendszone folder.
global.php	Contains constants and a function that is used across all the other php functions.

2. Run login.php by entering the following URL into your web browser:

```
http://localhost/friendszone/login.php?userid=jimmy&password=jimmy
```

The parameters in the URL, userid=jimmy&password=jimmy tells login.php to attempt to login with an account where user ID is jimmy and its password is jimmy. The following result is returned.

```
[{"result":"1"}]
```

The result of 1 tells us that the user ID and password is correct. Now enter the following URL into your web browser:

```
http://localhost/friendszone/login.php?userid=jimmy&password=peter
```

The following results tells us that there is a problem with the login. Either the user ID or password is incorrect.

```
[{"result":"0"}]
```

Each of the PHP web service in the FriendsZone folder returns a set of values to the caller. In the next few chapters, we will be making these calls using a Cordova application. We will also be describing how to call each of these PHP web services.

You may also refer to Appendix 5 for detailed line-by-line explanation of each PHP web service.

3. View the profiles table using phpMyAdmin. Note the correct userid and password field and how running login.php in the browser returns either a result:1 or result:0 depending on the parameters you passed into login.php.

Instruction 5: Read JSON

1. JSON is short form for JavaScript Object Notation. It is a format used to exchange data between software applications. We mentioned earlier that our Cordova FriendsZone application will be calling the PHP web services in XAMPP to read and write from the database. The data format that is used between Cordova and PHP is JSON. Apart from being a suitable format for data interchange between applications, JSON is also easy for human to read and write. Consider the following records:

relationshipid	description
9	Friends
10	Acquaintance
11	Family
12	Colleagues
13	Classmates

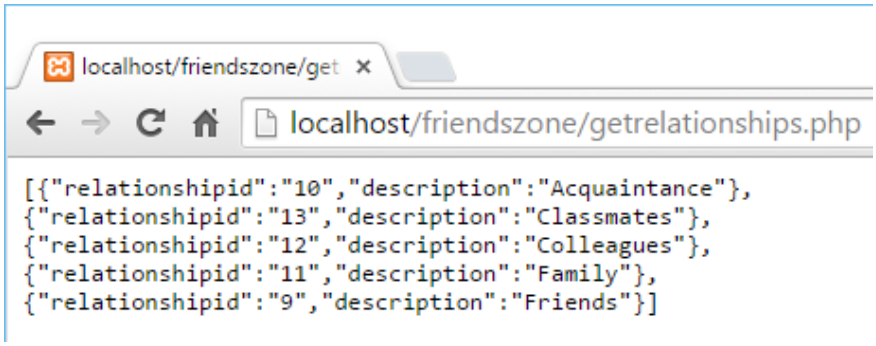
2. These records can represented in JSON as:

```
[
        {"relationshipid": "9"
          "description": "Friends"
        }
        {"relationshipid": "10"
          "description": "Acquaintance"
        }
        {"relationshipid": "11"
          "description": "Family"
        }
        {"relationshipid": "12"
          "description": "Colleagues"
        }
        {"relationshipid": "13"
          "description": "Classmates"
        }
]
```

The square brackets "[" and "]" represent the starting and ending notation of a JSON record-set while the round brackets "{" and "}" represent each record. Each field in the record is represent with a "<field name>" : "<content>" notation for example "relationshipid":"13".

3. Enter the following URL in your web browser and note the JSON output generated by getrelationships.php.

```
http://localhost/friendszone/getrelationships.php
```



Exercise

There are no exercises for this chapter.

Codes

Codes for this chapter can be found by searching VC13-MySQL-JSON on Github.com.