

Lab-4a: connect-db. Probably after watching the demo-app online, to help clarify the db connection issues

Connecting to a MySQL database.

VIDEO: lab-connect-db

Be sure that you have installed

```
mysql-connector-python      (if you didn't install this, you may be okay)
mysql-connector
mysqlclient
Flask-MySQLdb
Flask-SQLAlchemy
```

These were probably installed while you were watching the FlaskDemoIntro ppt slides.

The slides and the Setup lab detail some issues that may arise, and how you might address them.

You can always check by typing `pip list` in your Anaconda environment.

In order to complete this lab, you need to have a "user" table in your company database on localhost.

If you completed DemoApps (text) app #4, then you probably have one.

If you do not have a User table in your company database, then please create one with the following structure:

(Or, if you simply want to start "clean", you can "drop" your existing user table, and then create this one:)

```
CREATE TABLE user (id INT NOT NULL AUTO_INCREMENT, username
VARCHAR(20) NOT NULL , email VARCHAR(120) NOT NULL , image_file
VARCHAR(20) NOT NULL DEFAULT "default.jpg" , password varchar(60),
PRIMARY KEY (id));
```

Note the auto_increment key!!
Populate with a few records

In PHPMyAdmin, Click on the company database (not on one specific table). In the SQL window, insert the above SQL.

If your user table already exists, you should "drop" the user table first.

If the database won't let you drop your user table because it has foreign key references, the first rename your existing user table to something else, then create this new user table.

Alternatively, you can drop the posts that reference existing users.

Important for Mac/MAMP users: (and optional "can't hurt" Windows users:)

In your connections parameters, you will (most likely) be using a host of '127.0.0.1' instead of 'localhost'















This is due to the way that the Mac handles socket mapping with respect to localhost vs. 127.0.0.1

Therefore, you have to make sure that your username has privileges on your database for 127.0.0.1

Recall that during Orientation, you were instructed to setup a user in PHPMyAdmin with full privileges.

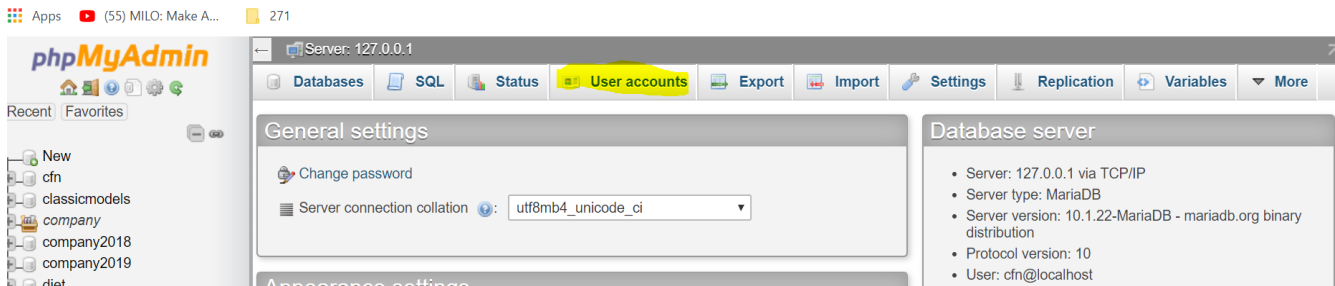
You set that up for localhost. Notice that I have set up cfn with access via localhost.

But notice that root also has access through 127.0.0.1 (and on the Mac, I don't think that's automatic.

	User name	Host name	Password	Global privileges	User group	Grant	Action
<input type="checkbox"/>	Any	%	No	USAGE		No	 Edit privileges  Export
<input type="checkbox"/>	Any	localhost	No	USAGE		No	 Edit privileges  Export
<input type="checkbox"/>	cfn	%	Yes	ALL PRIVILEGES		Yes	 Edit privileges  Export
<input type="checkbox"/>	cfn	localhost	Yes	ALL PRIVILEGES		Yes	 Edit privileges  Export
<input type="checkbox"/>	pma	localhost	No	USAGE		No	 Edit privileges  Export
<input type="checkbox"/>	root	127.0.0.1	No	ALL PRIVILEGES		Yes	 Edit privileges  Export
<input type="checkbox"/>	root	::1	No	ALL PRIVILEGES		Yes	 Edit privileges  Export

root localhost Yes ALL PRIVILEGES Yes [Edit privileges](#) [Export](#)

You want to set up your user (in my case cfn) to have all privileges access via 127.0.0.1 also.
To do this, make sure that you have clicked on User Accounts:

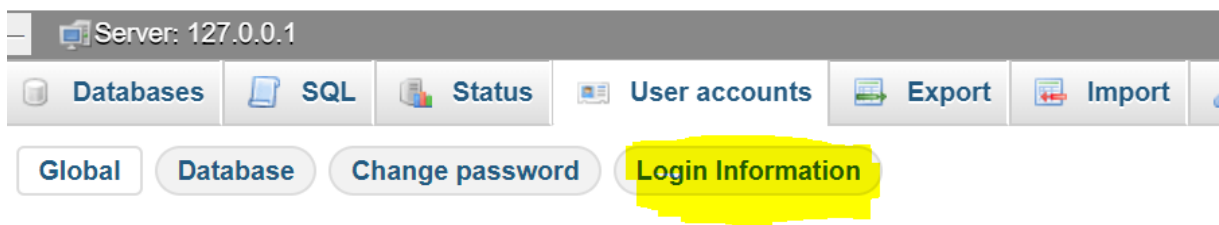


Then, click on Edit privileges for the user you want to modify (in my case cfn)

	User name	Host name	Password	Global privileges	User group	Grant	Action
<input type="checkbox"/>	Any	%	No	USAGE		No	Edit privileges Export
<input type="checkbox"/>	Any	localhost	No	USAGE		No	Edit privileges Export
<input type="checkbox"/>	cfn	%	Yes	ALL PRIVILEGES		Yes	Edit privileges Export
<input checked="" type="checkbox"/>	cfn	localhost	Yes	ALL PRIVILEGES		Yes	Edit privileges Export
<input type="checkbox"/>	pma	localhost	No	USAGE		No	Edit privileges Export
<input type="checkbox"/>	root	127.0.0.1	No	ALL PRIVILEGES		Yes	Edit privileges Export
<input type="checkbox"/>	root	:::1	No	ALL PRIVILEGES		Yes	Edit privileges Export
<input type="checkbox"/>	root	localhost	Yes	ALL PRIVILEGES		Yes	Edit privileges Export

[Check all](#) [With selected:](#) [Export](#)

Click on Login Information



Edit privileges: User account 'cfn'@'localhost'

Note: You are attempting to edit privileges of the user with which you are currently logged in.

Global privileges ☒ [Check all](#)

Change localhost to 127.0.0.1. But notice that you are actually adding a new user so you are not losing the ability of your user to access through localhost too!

Be sure to click "Go" at the bottom of the page. (not shown here, but it's there)

Assignment: Before coming to class, run through these 7 short examples (files included), and submit a screen shot of each one of the examples, showing your own access to your own database or, in the case of a remote DB, simply show a screenshot that you have run the example.

Connect to a local db (on localhost, e.g., through XAMPP or other localhost installation)

No flask, no web connection, just showing the connection:

- 01 `python-mysql-connect.py` (edit, don't open) **substitute your username and pwd**
Make sure your localhost (e.g. XAMPP/MAMP) is running **screenshot connection to db at 11:50 in video**
Mac/ MAMP: In the `mysql.connector.connect`, where it specifies the parameters, add `port='8889'`

Connect to a remote db (e.g., on cnaiman.com)

No flask, no web interface, just showing the connection:

- 02 `python-mysql-connect-remote.py`

Connect to remote db, select and print our results

No flask, no web GUI, no python graphics, just connect, query and display results

- 03 `python-mysql-connect-remote-select.py` **screenshot (discussion starting at 15:47)**
you will have to make changes for the screen shot

Connect to a remote db, select and print results, using flask and mysql.connector

using correct folder structure, `env.bat`, `run.bat`; not using special flask-mysqldb

test-flask-db-remote (folder)

screenshot (discussion 28:50 - 29:52)

04 **test-folder-DB.py (file)**

change SQL query, output row and single field

run from within Anaconda (if using), flask run, or, if you have a run.bat file setup, just "run".

Connect to a local db, select and print results, using flask and mysql.connector

using correct folder structure, env.bat, run.bat; not using special flask-mysqldb

test-flask-db-local (folder)

05 **test-folder-DB.py (file)**

run from within Anaconda (if using), flask run, or, if you have a run.bat file setup, just "run".

You must also have up and running XAMPP, or whatever localhost you are using.

you may have to pip install PyYAML and pip install ruamel-yaml

But do a pip list first--they may already be there!

change your connection string to your credentials

pip install flask-mysqldb

(no screenshot needed)

You may not have Users and Posts in your company database. Those tables were not there originally.

My database does have Users and Posts, because I added them during the lectures.

So you may want to add those tables (with Autonumber IDs) if you want to follow along. (see above)

06 **Simple-CRUD-Flask**

add DictCursor to app.config

update the yaml file!

change template to extract dictionary entries

use the template users-dict.html

submit screenshots of that

be sure to uncomment app.config (line 13)

07 **Simple-CRUD-Flask-Department**

(Note that these are using regular SQL INSERT clauses. Normally, we'd want to use SQLAlchemy for this.)

We want to use regular SQL queries for the SELECT queries. **update the yaml file!**

<https://stackoverflow.com/questions/48730683/connect-python-to-flask-mysqldb>