

COMS/SE 319: Construction of User Interfaces

Spring 2022

LAB Activity 6 – Setting up Web Server & MySQL on VM and interacting with MySQL Server by Node JS

Download **Fall2021-Lab06-sampleCode.zip**

Download the file **Individual_VMs.pdf** to see your VM Number in the second column.

We have 4 tasks:

1. **Task 1: Web Server Setup and Connecting to Server by SSH**
2. **Task 2: Setting up MySQL on VM**
3. **Task 3: Interact with database by NodeJS at server side.**
4. **Task 4 (Optional): Querying from database by NodeJS at client side.**

Prerequisite:

SSH: to connect to server's terminal.

- Ubuntu: <https://linuxize.com/post/how-to-enable-ssh-on-ubuntu-18-04/>
- Mac: <http://osxdaily.com/2011/09/30/remote-login-ssh-server-mac-os-x/>
- Windows: You can use one of 2 softwares:
 - + <https://www.putty.org>
 - + <https://jcutrer.com/windows/install-openssh-on-windows10>

FileZilla: to view/add/ drop files to server.

- Ubuntu:
<https://www.computernetworkingnotes.com/linux-tutorials/how-to-install-filezilla-ft-p-client-in-ubuntu.html>
- Mac OSX and Windows:
<https://filezilla-project.org/download.php?type=client>

You are recommended to set this password to your MySQL users:

Username	Password
root (default)	See the end of task 2.1
root (changed)	comsVM@319
team1	team1comsVM@319

Contents

- Login/ssh to 319 server
- Start http service
- Host an HTML file to server (using terminal and FileZilla)
- Interact with MySQL by nodejs.

Task 1: Web Server Setup and Connecting to Server by SSH

1.1. Connecting to 319 Server

- We will connect to the server using ssh (Secure Shell is a cryptographic network protocol for operating network services)
- Use the command in terminal (mac and linux). If you are using Windows use GIT Bash terminal.

- Command: **ssh netid@coms-319-###.cs.iastate.edu**

- E.g., **ssh abc@coms-319-###.cs.iastate.edu**

- password will be the net-id password

Note: you can find your VM No. (###) inside the provided Spring2021-Students-VMs List.
If you have number 1, your VM no is 001.

1.2. Apache http server

- Enable httpd:

```
sudo systemctl enable httpd.service
```

- Start the http service by following command

```
sudo service httpd start
```

Optional:

- You can check that the service is running by:
\$ sudo service httpd status
- If you want you can stop the service by:
\$ sudo service httpd stop
- The configuration file for the http server is httpd.conf. All the configurations like port number, Document Root and Access Controls will be in this file.
- Apache Httpd Config directory: /etc/httpd/conf

1.3. Create a Webpage

- Open the file index.html

```
1  <html>
2  <head>
3    <title></title>
4  </head>
5  <body>
6    <h1>Lab 03</h1>
7    <h3>Hello World</h3>
8  </body>
9  </html>
```

1.4. Change mode of '/var/www/html/':

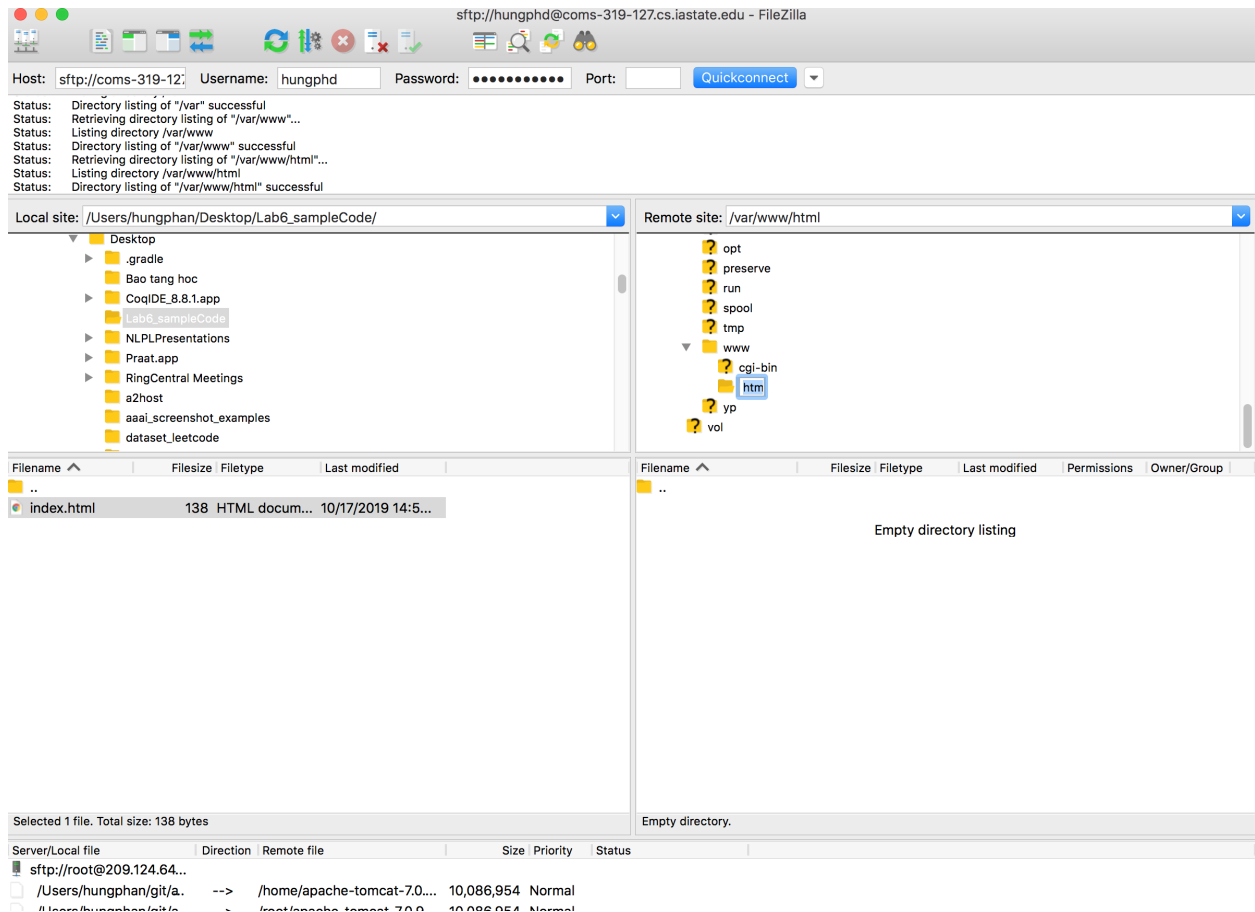
You need to grant permission to copy files to '/var/www/html'

Use ssh to run these commands on server:

```
sudo chmod 777 /var/www/html
```

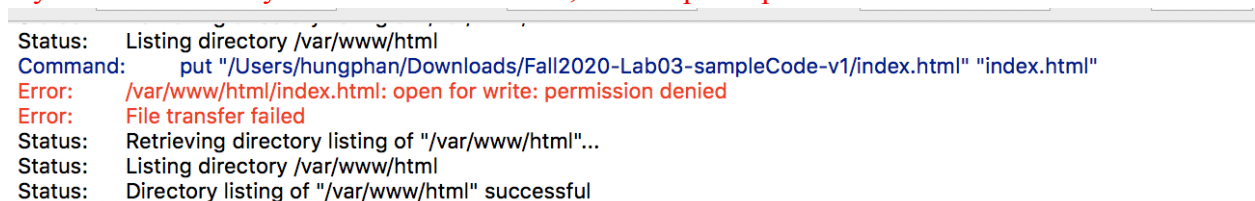
1.5. Transfer files from your computer to server

- You can use Filezilla, FTP/SFTP file transfer GUI tool to copy your webpage to the server.
- Host: sftp://server-address (sftp://**coms-319-###.cs.iastate.edu**)
- Username: Net ID
- Password: Net ID-password
- Port is not needed
- Drag and drop the index.html from your local machine to the "/var/www/html" directory of the server (You can copy "/var/www/html" and paste it in Remote site path).



1.5.1. Optional - Transfer files indirectly.

If you don't have any error with file transfer, let's skip this part.




If the file transfer is not successful due to the file permission (see above picture), you can copy the **index.html** to `/home/{NetID}/` by FileZilla. Then you use ssh to move the file to the `/var/www/html` location using **sudo mv**:

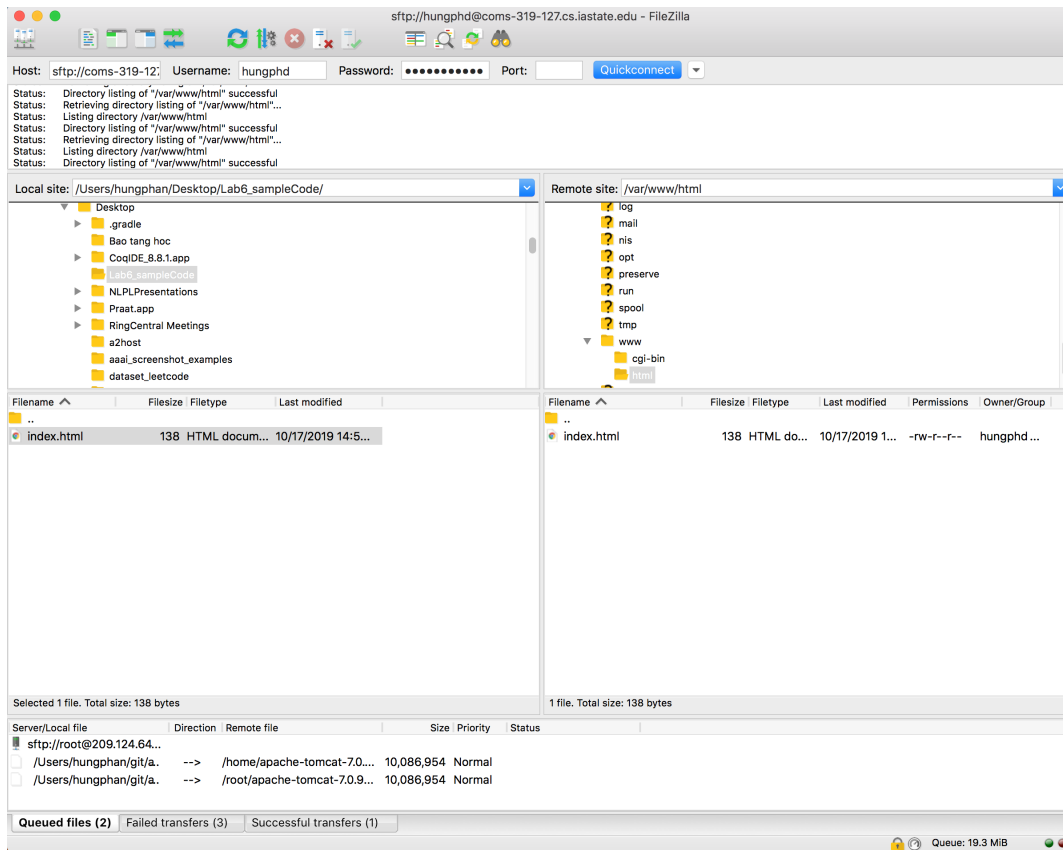
```
sudo mv /home/hungphd/index.html /var/www/html
```

```

[hungphd@coms-319-127 ~]$ sudo mv /home/hungphd/index.html /var/www/html/
[sudo] password for hungphd:
Sorry, try again.
[sudo] password for hungphd:
[hungphd@coms-319-127 ~]$

```

You can check on FileZilla and use the Refresh button () to see the result. The correct result should be like this:



1.6. Navigate to the Address

- Type the address (http://server-address) in browser and navigate to the webpage.

Lab 03

Hello World

1.6 Open Apache port (Optional)

If you don't have any errors when doing step 1.5, please skip this part.

- If the above method does not work, you might need to open the server port. Use the following two commands for opening the port.
- Port 80 is open to be accessible from web browser. You can do that by:
`$ sudo firewall-cmd --add-port 80/tcp --permanent`
- You need to reload the setting by using:
`$ sudo firewall-cmd --reload`

```
[[hungphd@coms-319-127 ~]$  
[[hungphd@coms-319-127 ~]$  
[[hungphd@coms-319-127 ~]$ sudo firewall-cmd --add-port 80/tcp --permanent  
Warning: ALREADY_ENABLED: 80:tcp  
success  
[[hungphd@coms-319-127 ~]$ sudo firewall-cmd --reload  
success  
[[hungphd@coms-319-127 ~]$ █
```

Task 2: Setting up MySQL on VM

Contents

- How to create SSH into your VM?
- How to initiate SQL instance?
- How to create new database?
- How to create new users for any host?
- How to grant permissions for a user?
- How to log into the sql instance using the new user.

2.1. SSH into your VM

- The host name for your VM is provided. you can find your VM No. (###) inside the provided list
- Open terminal.
- Command `ssh netid@coms-319-###.cs.iastate.edu`
- Eg. `ssh netid@coms-319-001.cs.iastate.edu`
- And enter your NetId Password.

A successful login should show on console like this:

```
[hungphd@coms-319-000.cs.iastate.edu's password:
```

NOTES:

These systems are designed to automatically update and restart on Monday mornings between 1:00 am and 6:00 am.

If this is your first time logging in, please read the README.txt file in your home folder (less README.txt).

```
Last login: Thu Feb 13 13:23:00 2020 from 10.64.76.93
```

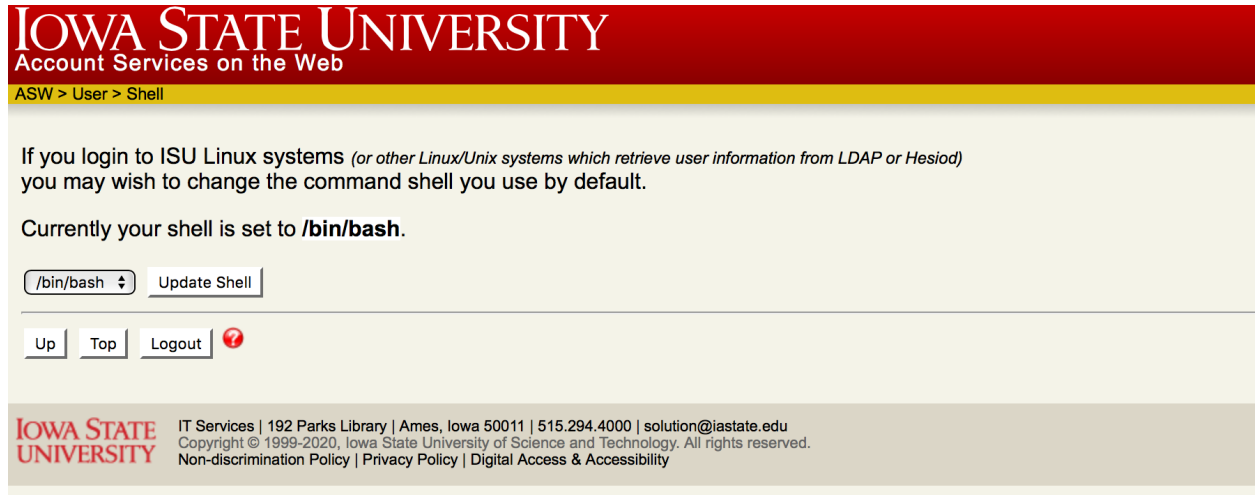
```
[hungphd@coms-319-000 ~]$ █
```

2.1.1. Change your current shell (Optional).

If you are able to SSH, skip this part.

If you cannot ssh to your VMs, you might need to update Git Bash shell configuration by AWS. Go to this link and update the current shell to “Bin/Bash” option. Then click on the ‘Update Shell’ button.

<https://asw.iastate.edu/cgi-bin/acropolis/user/shell>




IOWA STATE UNIVERSITY
Account Services on the Web

ASW > User > Shell

If you login to ISU Linux systems (or other Linux/Unix systems which retrieve user information from LDAP or Hesiod) you may wish to change the command shell you use by default.

Currently your shell is set to **/bin/bash**.



IOWA STATE UNIVERSITY IT Services | 192 Parks Library | Ames, Iowa 50011 | 515.294.4000 | solution@iastate.edu
Copyright © 1999-2020, Iowa State University of Science and Technology. All rights reserved.
Non-discrimination Policy | Privacy Policy | Digital Access & Accessibility

If you still have errors when SSH, please ask the TAs for further help.

2.2. Initializing MySQL on VMs

Run this command on SSH:

```
sudo /usr/local/bin/isu_mysql_initialize
```

Enter ‘y’ to agree that you want to wipe out the current database.

You will see a report of the initialization. In the end of the report, you should see the password like the screenshot below (**the password can be different in your server**). Save this password.


```
Status: Directory listing of "/var/www/html" successful
hungphan — hungphd@coms-319-002:/var/www/html — ssh hungphd@coms...
...h hungphd@coms-319-002.cs.iastate.edu ...319-TA/Fall-2020/Lab3/after — -bash

+ mysql@2.18.1
added 11 packages from 15 contributors and audited 11 packages in 0.618s
found 0 vulnerabilities

[[hungphd@coms-319-002 html]$ node createCompanies_2.js
Connected to server!
Prepare to insert to table Companies:
3 records inserted
Close connection!
[[hungphd@coms-319-002 html]$ node viewCompanies_2.js
[ RowDataPacket { name: 'Amazon', foundedYear: 1994 },
  RowDataPacket { name: 'Microsoft', foundedYear: 1975 },
  RowDataPacket { name: 'Pixar', foundedYear: 1986 } ]
Close connection!
[[hungphd@coms-319-002 html]$ sudo /usr/local/bin/isu_mysql_initialize

Would you like to initialize the MySQL Server?
Any existing database data will be wiped out.

Initialize MySQL Server? [y/N] y

[[hungphd@coms-319-002 html]$ node viewCompanies_2.js
[ RowDataPacket { name: 'Amazon', foundedYear: 1994 },
  RowDataPacket { name: 'Microsoft', foundedYear: 1975 },
  RowDataPacket { name: 'Pixar', foundedYear: 1986 } ]
Close connection!
[[hungphd@coms-319-002 html]$ sudo /usr/local/bin/isu_mysql_initialize

Would you like to initialize the MySQL Server?
Any existing database data will be wiped out.

Initialize MySQL Server? [y/N] y
Stopping MySQL Server...
Purging existing database structure...
Truncating logs...
Initializing core MySQL structures...
Reapplying permissions...
Enabling MySQL on boot...
Starting MySQL Server...
Setting MySQL root password...
MySQL root user password is 96a10cac93ef7670.
[[hungphd@coms-319-002 html]$
```

In the screenshot above, the password of the server is **96a10ccacc93ef7670** (no '.' letter)

2.3. Accessing MySQL on VMs

MySQL is already installed on the VMs.

- To enable MySQL, run two commands:

```
sudo systemctl enable mysqld.service  
sudo systemctl start mysqld.service
```

The expected output should look like this:

```
Last login: Thu Feb 13 13:23:00 2020 from 10.64.76.93  
[hungphd@coms-319-000 ~]$ sudo systemctl enable mysqld.service  
[hungphd@coms-319-000 ~]$ sudo systemctl start mysqld.service  
[hungphd@coms-319-000 ~]$ █
```

- **Default password for root**

You can use this command to access to mysql:

```
mysql -u root -p
```

- Password (no space) is the password you got from previous step.
- **Note** : This is the **temporary** password for your VMs DB. You are required to change the password for **root** which is easier to remember for you in the next part.

- **Change the default password for 'root'.**

```
ALTER USER 'root'@'localhost' IDENTIFIED BY 'comsVM@319';
```

- See details from <https://stackoverflow.com/questions/33467337/reset-mysql-root-password-using-alter-user-statement-after-install-on-mac>

The successful output should look like this:

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'comsVM@319';
Query OK, 0 rows affected (0.03 sec)

mysql> █
```

2.4. Accessing MySQL on VMs

Useful DB Commands

- **Show Databases;** (This will show the list of DBs available and granted for that user. Note: There will be already some default DBs created by Mysql. Please create new one for your project.)
- **Create DATABASE {Db-name};** (This will create a database with the given name.)
- **Use {db-name};** (This will switch the database to the given Db-name. And you can start accessing the Tables under that db-name database.)

2.5. Create Database:

- A database can be thought of as a collection of related tables.
- One can create many databases in the SQL instance.
- Create a Database using the command **CREATE DATABASE {your-dbname};**
- e.g. **CREATE DATABASE MyProject;**

```
CREATE DATABASE MyProject;
```

Then show your database by this command:

```
show databases;
```

```
hungphan — hungphd@coms-319-000:~ — ssh hungphd@coms-319-000.cs.i...
+-----+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+-----+
4 rows in set (0.01 sec)

[mysql> CREATE DATABASE MyProject;
Query OK, 1 row affected (0.04 sec)

[mysql> show databases;
+-----+
| Database           |
+-----+
| MyProject           |
| information_schema  |
| mysql               |
| performance_schema  |
| sys                 |
+-----+
5 rows in set (0.01 sec)

mysql> █
```

2.6. Create user for your DataBase

You can create ONE USER per team for your group project.

- The Passwords must have “a-z,A-z,0-9” and some special characters.
- e.g. Password: **team1comsVM@319**.
- Command: **CREATE USER team1'@'%' IDENTIFIED BY 'Yourpasswordhere';**
- Use ‘%’ to connect from any host. (RECOMMENDED).
- e.g. **CREATE USER 'team1'@'%' IDENTIFIED BY 'team1comsVM@319';**
- Note :You should inform your other members of the group “the user credentials to access your DB while doing the group project.

To create user “**team1**”, run this command:

```
CREATE USER 'team1'@'%' IDENTIFIED BY 'team1comsVM@319';
```

```
hungphan — hungphd@coms-319-000:~ — ssh hungphd@coms-319-000.cs.i...
[mysql>
[mysql>
[mysql>
[mysql>
[mysql>
[mysql>
[mysql>
[mysql>
[mysql>
[mysql>
[mysql>
[mysql>
[mysql>
[mysql>
[mysql>
[mysql>
[mysql>
[mysql>
[mysql>
[mysql>
[mysql> CREATE USER 'team1'@'%' IDENTIFIED BY 'team1comsVM@319';
Query OK, 0 rows affected (0.04 sec)

mysql> █
```

2.7. Grant Permissions to the user

Make sure you Give all the permissions that you need.

- **GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP ON MyProject.* TO 'team1'@'%';**
- Please Refer to “<https://dev.mysql.com/doc/mysql-securityexcerpt/5.7/en/adding-users.html>” on setting the Grants.

To grant the permissions to ‘team1’, run this command:

```
GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP ON MyProject.* TO  
'team1'@'%';
```

```
hungphan — hungphd@coms-319-000:~ — ssh hungphd@coms-319-000.cs.i...
[mysql>
[mysql>
[mysql>
[mysql>
[mysql>
[mysql>
[mysql>
[mysql>
[mysql>
[mysql>
[mysql>
[mysql>
[mysql>
[mysql>
[mysql>
[mysql>
[mysql>
[mysql>
[mysql> GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP ON MyProject.* TO 'team1'@'
%';
Query OK, 0 rows affected (0.03 sec)

mysql>
```

2.8. Access Mysql using the new User from any host.

- **Logout of your current MySQL console**

If you don't know how to get out of the mysql terminal, use this command to return to normal terminal:

```
ctrl+'z'
```

```
mysql>
[1]+  Stopped                  mysql -u root -p
[hungphd@coms-319-127 ~]$
```

- **Login to MySQL instance using the new user (eg: team1) by this command.**

```
mysql -u team1 -p
```

- Use “**show databases;**” To see the list of databases.

```
show databases;
```

- “use [DB-Name];” by this command:

```
use MyProject;
```

The expected output after 2..7:

```
mysql> show databases;
+-----+
| Database |
+-----+
| MyProject |
| information_schema |
+-----+
2 rows in set (0.01 sec)

mysql> use MyProject;
Database changed
mysql> █
```

2.9. To Access your Mysql instance using MySQL Workbench

For encryption purpose, please alter the User that you created.

- Login into MYSql on your VM using the **root** user.
(Please refer to page “Accessing MySQL on VM”)
- ALTER USER 'your_user'@'localhost' IDENTIFIED WITH mysql_native_password BY 'your_password';
- e.g. ALTER USER 'team1'@'%' IDENTIFIED WITH mysql_native_password BY 'team1comsVM@319';

- Switch back to root user by these commands:

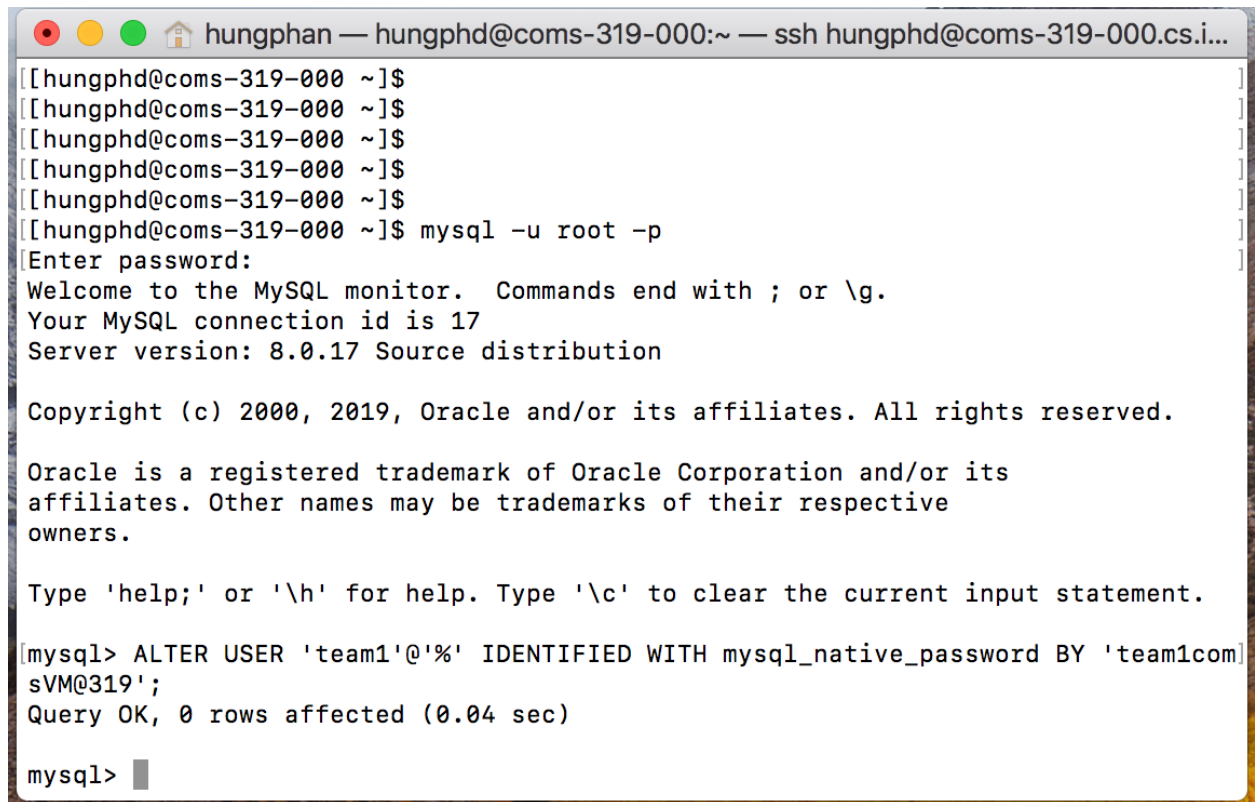
```
ctrl+z
```

```
mysql -u root -p
```

- Run this command:

```
ALTER USER 'team1'@'%' IDENTIFIED WITH mysql_native_password BY
'team1comsVM@319';
```

Expected output

A terminal window titled 'hungphan — hungphd@coms-319-000:~ — ssh hungphd@coms-319-000.cs.i...' displays a series of commands and their outputs. The user enters 'mysql -u root -p', followed by the MySQL prompt. The user then enters the SQL command 'ALTER USER 'team1'@'%' IDENTIFIED WITH mysql_native_password BY 'team1com]sVM@319';'. The output shows 'Query OK, 0 rows affected (0.04 sec)' and the prompt returns to 'mysql>'.

```
[hungphd@coms-319-000 ~]$  
[hungphd@coms-319-000 ~]$  
[hungphd@coms-319-000 ~]$  
[hungphd@coms-319-000 ~]$  
[hungphd@coms-319-000 ~]$  
[hungphd@coms-319-000 ~]$ mysql -u root -p  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 17  
Server version: 8.0.17 Source distribution  
  
Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql> ALTER USER 'team1'@'%' IDENTIFIED WITH mysql_native_password BY 'team1com]  
sVM@319';  
Query OK, 0 rows affected (0.04 sec)  
  
mysql> █
```

You need to do this step to be able to access MySQL from other servers such as nodejs.

Task 3: Interact with database by NodeJS at server side.

In this task, you will create a Companies table to your mysql host server on VM by using nodejs and view the content of Companies table using html and nodejs.

- **Make sure that you did all steps from task 1 and task 2.**

Step 1: Add content to the createCompanies.js file as follow:

- Add create table query and insert query as follows :

```
var sqlCreate='create table if not exists Companies(name VARCHAR(255),foundedYear int,PRIMARY KEY (name))'
```

```
var sqlInsert='Insert into Companies (name,foundedYear) values ?'
```

The expected code should be:

Before:

```

1  var mysql = require('mysql');
2
3  var con = mysql.createConnection({
4    host:'localhost',
5    user:'team1',
6    password:'team1comsVM@319',
7    database:'MyProject'
8  });
9  con.connect(function(err) {
10    if (err) throw err;
11    console.log("Connected to server!");
12    // add code here
13    con.query(sqlCreate, function (err, result) {
14      if (err) {
15        console.log(err);
16        throw err;
17      } else{
18        console.log("Prepare to insert to table Companies: ");
19        var values= [
20          ["Pixar", 1986],
21          ["Amazon",1994],
22          ["Microsoft",1975]
23        ];
24        // add code here
25        con.query(sqlInsert,[values], function (err, result) {
26          if (err) {
27            // console.log(err);
28            console.log("3 records exists!");
29          } else{
30            console.log("3 records inserted");
31          }
32        });
33      });
34
35      con.end(function(err) {
36        if (err) {
37          return console.log(err.message);
38        } else{
39          console.log("Close connection!");
40        }
41      });
42    }
43
44  });
45

```

After

```
1  var mysql = require('mysql');
2
3  var con = mysql.createConnection({
4    host: 'localhost',
5    user: 'team1',
6    password: 'team1comsVM@319',
7    database: 'MyProject'
8  });
9  con.connect(function(err) {
10   if (err) throw err;
11   console.log("Connected to server!");
12   // add code here
13   var sqlCreate='create table if not exists Companies(name VARCHAR(255),foundedYear int, PRIMARY KEY (name))'
14   con.query(sqlCreate, function (err, result) {
15     if (err) {
16       console.log(err);
17       throw err;
18     } else{
19       console.log("Prepare to insert to table Companies: ");
20       var values= [
21         ["Pixar", 1986],
22         ["Amazon",1994],
23         ["Microsoft",1975]
24       ];
25       // add code here
26       var sqlInsert='Insert into Companies (name,foundedYear) values ?'
27       con.query(sqlInsert,[values], function (err, result) {
28         if (err) {
29           // console.log(err);
30           console.log("3 records exists!");
31         } else{
32           console.log("3 records inserted");
33         }
34       });
35     }
36   });
37   con.end(function(err) {
38     if (err) {
39       return console.log(err.message);
40     } else{
41       console.log("Close connection!");
42     }
43   });
44 }
45
46 });
47
48
```

Step 2: Add content to **viewCompanies.js** .

Add the following queries in 4 locations (line 14, 21, 27, 34) in following order:

SELECT * FROM Companies
SELECT * FROM Companies where FOUNDEDYear<1980
SELECT * FROM Companies where Name like 'Ama%'
SELECT * FROM Companies where foundedYear>=1980

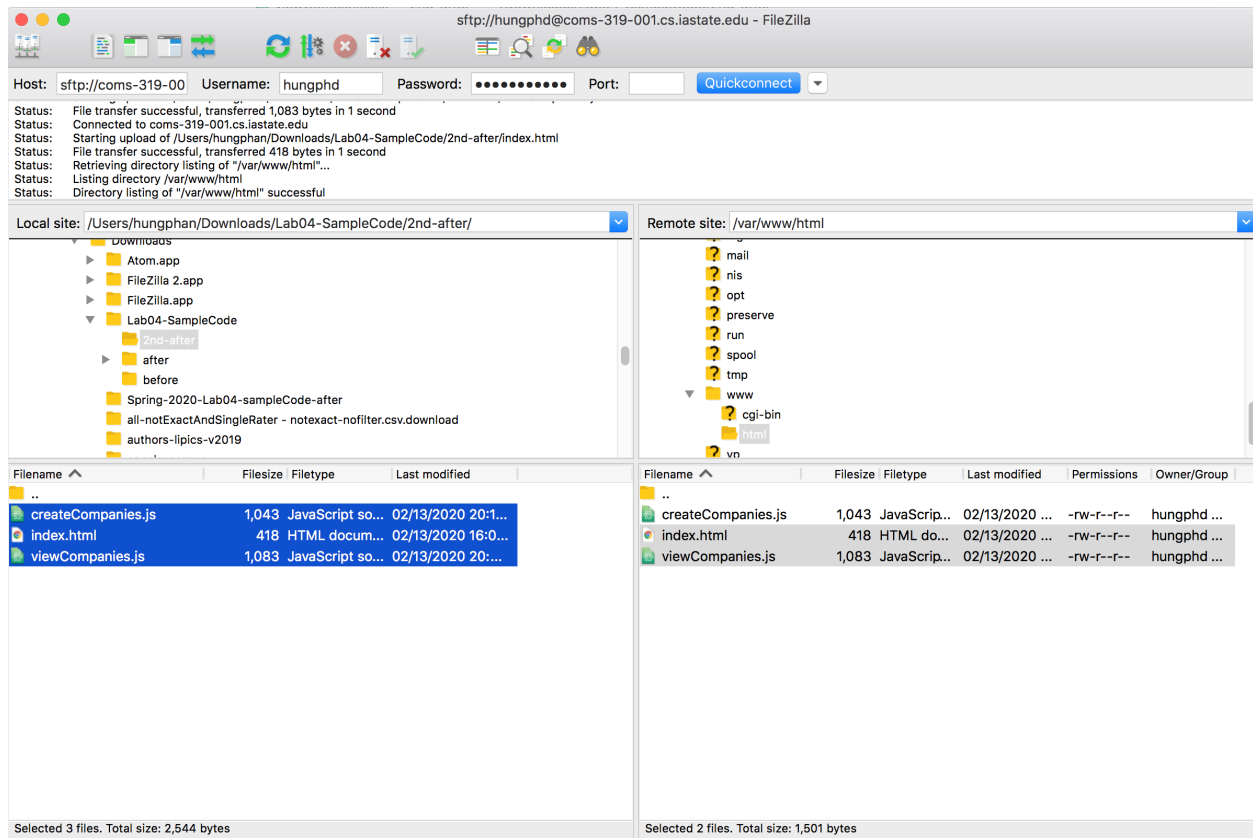
Before

```
1  var mysql = require('mysql');
2
3  var con = mysql.createConnection({
4    host: 'localhost',
5    user: 'team1',
6    password: 'team1comsVM@319',
7    database: 'MyProject'
8  });
9
10
11 con.connect(function(err) {
12   if (err) throw err;
13
14   con.query("?", function (err, result, fields) {
15     if (err) throw err;
16     console.log("Query 1 result:");
17     console.log(result);
18
19   });
20
21   con.query("?", function (err, result, fields) {
22     if (err) throw err;
23     console.log("Query 2 result:");
24     console.log(result);
25   });
26
27   con.query("?", function (err, result, fields) {
28     if (err) throw err;
29     console.log("Query 3 result:");
30     console.log(result);
31
32   });
33
34   con.query("?", function (err, result, fields) {
35     if (err) throw err;
36     console.log("Query 4 result:");
37     console.log(result);
38
39   });
40   con.end(function(err) {
41     if (err) {
42       return console.log(err.message);
43     } else {
```

After

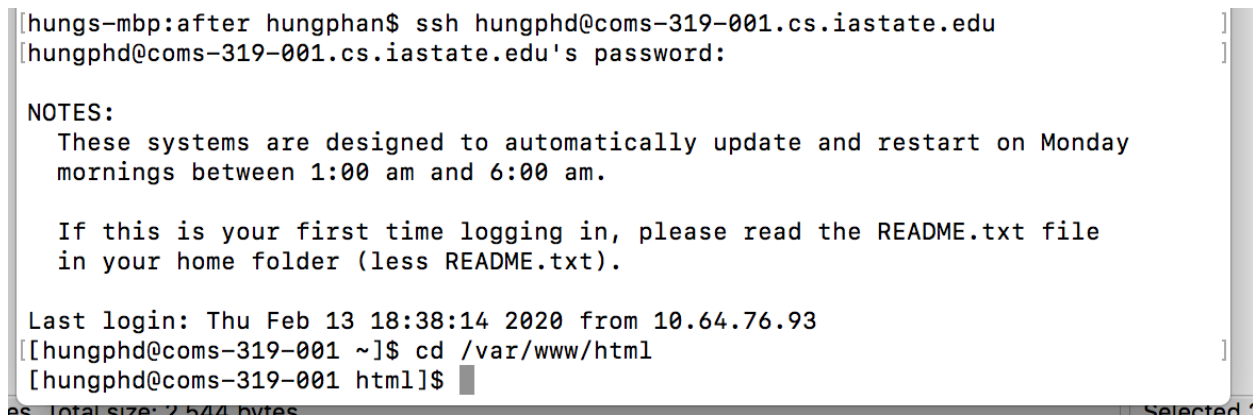
```
1  var mysql = require('mysql');
2
3  var con = mysql.createConnection({
4    host: 'localhost',
5    user: 'team1',
6    password: 'team1comsVM@319',
7    database: 'MyProject'
8  });
9
10
11 con.connect(function(err) {
12   if (err) throw err;
13
14   con.query("SELECT * FROM Companies", function (err, result, fields) {
15     if (err) throw err;
16     console.log("Query 1 result:");
17     console.log(result);
18   });
19
20
21   con.query("SELECT * FROM Companies where FOUNDEDYear<1980", function (err, result, fields) {
22     if (err) throw err;
23     console.log("Query 2 result:");
24     console.log(result);
25   });
26
27   con.query("SELECT * FROM Companies where Name like 'Ama%'", function (err, result, fields) {
28     if (err) throw err;
29     console.log("Query 3 result:");
30     console.log(result);
31   });
32
33
34   con.query("SELECT * FROM Companies where foundedYear>=1980", function (err, result, fields) {
35     if (err) throw err;
36     console.log("Query 4 result:");
37     console.log(result);
38   });
39
40   con.end(function(err) {
41     if (err) {
42       return console.log(err.message);
43     } else {
44       console.log("Close connection!");
45     }
46   });
47 });
```

Step 3: Copy viewCompanies.js and createCompanies.js to /var/www/html by FileZilla:



Step 4: Run the following commands (on server) :

- SSH to server
- go to /var/www/html



Step 5: Install mysql package on NodeJS:

```
npm init  
npm install mysql
```

Step 6: create company table and insert 3 rows.

```
node createCompanies
```

```
[hungs-mbp:after hungphan$ node createCompanies.js  
Connected to server!  
Prepare to insert to table Companies:  
3 records inserted  
Close connection!  
hungs-mbp:after hungphan$ █
```

- Run **node createCompanies.js** for the second time. Now the output change:

```
[hungs-mbp:after hungphan$ node createCompanies.js  
Connected to server!  
Prepare to insert to table Companies:  
3 records exists!  
Close connection!  
█
```

- The insertion of 3 records has failed since you already inserted successfully 3 records in previous steps. MySQL will check the validity of input by using **primary key**.

Step 7: view company:

```
node viewCompanies
```

The output should be:


```
[[hungphd@coms-319-001 html]$ node viewCompanies.js
Query 1 result:
[ RowDataPacket { name: 'Amazon', foundedYear: 1994 },
  RowDataPacket { name: 'Microsoft', foundedYear: 1975 },
  RowDataPacket { name: 'Pixar', foundedYear: 1986 } ]
Query 2 result:
[ RowDataPacket { name: 'Microsoft', foundedYear: 1975 } ]
Query 3 result:
[ RowDataPacket { name: 'Amazon', foundedYear: 1994 } ]
Query 4 result:
[ RowDataPacket { name: 'Amazon', foundedYear: 1994 },
  RowDataPacket { name: 'Pixar', foundedYear: 1986 } ]
Close connection!]
```

References

- <https://dev.mysql.com/doc/refman/8.0/en/creating-database.html>
- <https://dev.mysql.com/doc/mysql-security-excerpt/5.7/en/addingusers.html>

Task 4 (Optional): Querying from database by NodeJS at client side.

If you have any errors with this task, please contact the TAs in/after the class. This task is optional and we don't use this task as material for the quiz.

4.1. Install mysql on client

Go to the sample code folder on your local computer, then

- Install mysql package on NodeJS:

```
npm init
npm install mysql
```

4.2. Open viewCompanies_client.js, change the localhost to your assigned VM:

```

1  var mysql = require('mysql');
2
3  var con = mysql.createConnection({
4      host: 'coms-319-002.cs.iastate.edu',
5      user: 'team1',
6      password: 'team1comsVM@319',
7      database: 'MyProject'
8  });
9
10
11 con.connect(function(err) {
12     if (err) throw err;
13     con.query("SELECT * FROM Companies", function (err, result, fields) {
14         if (err) throw err;
15         console.log(result);
16         con.end(function(err) {
17             if (err) {
18                 return console.log(err.message);
19             } else{
20                 console.log("Close connection!");
21             }
22         });
23     });
24 });

```

4.3. Run viewCompanies_client.js on your computer

- You should see the same output with task 3 step 7.

Please submit only the Lab Activity Quiz question on Canvas. No need to submit other questions mentioned in this pdf.
