

Node.js and MySQL

ผศ. ดร. เก็จแก้ว ธเนศวร

kejkaew.tha@mail.kmutt.ac.th

ตารางเรียน

สัปดาห์	วันที่	หัวข้อ/รายละเอียด	การบ้าน (ผู้ตรวจ)	ส่ง Assignment
1	8/8/2566	Introduction to Web programming, Basic HTML 1	Assignment 1 (KT,KN)	1
2	15/8/2566	Basic HTML 2 + CSS 1	Assignment 2 (KT,KN)	1,2
3	22/8/2566	CSS 2	Assignment 3 (KT)	1,2,3
4	29/8/2566	Responsive Web design	Assignment 4 (KT,KN)	2,3,4
5	5/9/2566	JavaScript 1 + DOM	Assignment 5 (KT,KN)	3,4,5
6	12/9/2566	JavaScript 2 + GitHub	Assignment 6 (KT)	4,5,6
7	19/9/2566	JavaScript 3	Assignment 7 (KT,KN)	5,6,7
8	26/9/2566	AJAX + JSON	Assignment 8 (KT,KN)	6,7,8
9	3/10/2566	Node.js 1	Assignment 9 (KT)	6,7,8,9
	10/10/2566	No class	ส่ง portfolio web page (KT)	
10	17/10/2566	Node.js 2	Assignment 10 (KT,KN)	8,9,10
11	24/10/2566			8,9,10
12	31/10/2566	Node.js 3	Assignment 11 (KT,KN)	9,10,11
13	7/11/2566	Node.js 4 + MySQL	Assignment 12 (KT,KN)	10,11,12
14	14/11/2566		online	10,11,12
15	21/11/2566	ส่งความก้าวหน้าโปรเจค+ Quiz	online	12
16	28/11/2566	ส่งโปรเจค (9:00-16:00)	KT (ห้องเรียน)	
		สอบปลายภาค ในตาราง	KT	

สอบ Quiz วันที่ 21 พ.ย. 2566 (60 นาที)

- 10 คะแนน
- มีสอบ 1 รอบ คือ เวลาช่วงเย็น ประมาณ 17.00 น. ทุกคนจะได้ว่างพร้อมกัน เป็นสอบออนไลน์
- เรื่อง NodeJS
- ไม่มีเรียน แต่มีส่ง Assignment

Project Progress

- ส่ง progress ของ project แต่ละกลุ่ม (5 คะแนน) ในวันที่ 21 พ.ย. 2566
- อาจารย์จะมีแบบฟอร์มให้แต่ละกลุ่มกรอก
- Project ต้องส่งภายใน วันที่ 27 พ.ย. 2566 เวลา 23.55 น. (10 คะแนน)
- นำเสนอแต่ละกลุ่ม วันที่ 28 พ.ย. 2566 เวลา 9.00 – 15.00 น. (20 คะแนน) ไม่
ต้องทำ presentation

Web Programming Project: Web application

- Web application ที่เป็น platform เช่น social media, Amazon, Lazada etc.
 - รองรับทั้งผู้พัฒนาหรือผู้ขาย และ ผู้ใช้หรือผู้ซื้อ
- คะแนน 35 คะแนน
- กลุ่มละไม่เกิน 3 คน (จับกลุ่มกับเพื่อนต่าง section ได้)

-
- เลือกทำ 1 อย่างจากหัวข้อต่อไปนี้
 - Freelancing platform
 - Online learning platform
 - Browser game platform
 - กรณาดูรายละเอียดใน slide หน้าถัดไปด้วย
 - สามารถใช้ CSS, JavaScript, Backend framework ได้
 - ไม่ให้ใช้ web builder หรือ CMS ต่าง ๆ เช่น word press
 - ส่งรายชื่อ พร้อมเขียน project plan ว่าเราจะทำอะไร พร้อมแผนคร่าว ๆ ส่งวันที่ 3 พ.ย. 2566

รายละเอียด

- Freelancing platform: สิ่งที่ต้องมีเป็นอย่างน้อย
 - Login page
 - Register page
 - Profile หรือ portfolio page (ให้กรอกหรือ upload ก็ได้)
 - Job search page
 - ผู้ใช้สามารถ post ข้อความ ใน job ที่ต้องการ และกด save งานที่ต้องการทำได้
 - 1 database

-
- Online learning platform : สิ่งที่ต้องมีเป็นอย่างน้อย
 - Login page
 - Register page
 - Course page และมีตัวอย่าง video ให้ดู
 - Quiz page ของแต่ละ course หลังจากสมัครเรียนแล้ว
 - ผู้ใช้สามารถสมัครเรียน ในแต่ละ course ได้
 - 1 database

-
- Browser game: สิ่งที่ต้องมีเป็นอย่างน้อย
 - Login page
 - Register page
 - Game page (หาเกมที่มีอยู่แล้วได้)
 - Leader board page
 - ต้องกด Like/Love และ comment ใน leader board ได้
 - 1 database

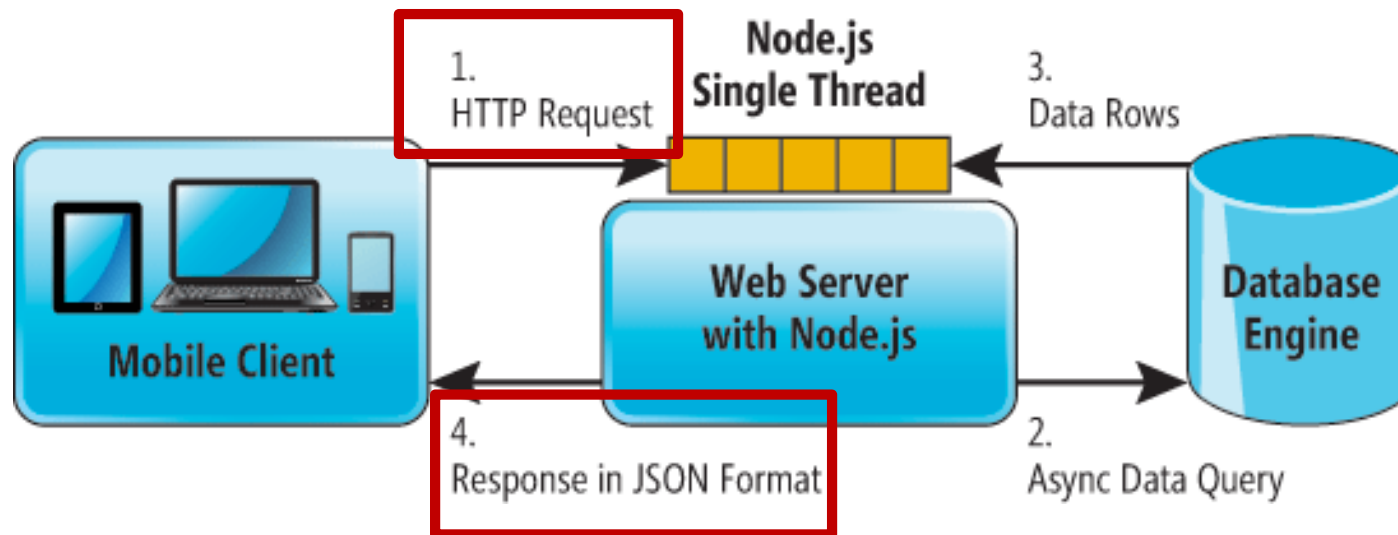
เรื่องที่เรียนไปสัปดาห์ที่แล้ว

- HTML Form and Express.js
- Cookies
- Upload Image File
- Assignment 11

วันนี้เราจะเรียนเรื่อง

- MySQL
- Create table
- Insert
- Update
- Delete
- Select
- Assignment 12: Register and Login (Database)

ทบทวน



- Client – Server relationship

MySQL

Database Management System (DBMS)

- กลุ่มโปรแกรมที่ทำหน้าที่เป็นตัวกลางในระบบติดต่อระหว่างผู้ใช้กับฐานข้อมูล เพื่อจัดการและควบคุมความถูกต้อง ความซ้ำซ้อน และความสัมพันธ์ระหว่างข้อมูลต่างๆ ภายในฐานข้อมูล
- DBMS มีหลายประเภท เช่น Hierarchical databases, Network databases, Relational databases, Object-oriented databases, Graph databases, ER model databases, Document databases และ NoSQL databases

DBMS ที่เป็น Relational databases

Relational database คือ database ที่แสดงความสัมพันธ์ระหว่าง data หรือ table โดยมีการเก็บข้อมูลในรูปแบบตาราง ที่ประกอบด้วย columns และ rows

ตัวอย่างของ Relational databases

- Oracle
- MySQL (free software)
- Microsoft SQL Server
- PostgreSQL
- IBM Db2
- SQLite (free software)
- Microsoft Access
- MariaDB (free software)
- Hive (free software; specialized for data warehouses).[23]
- Microsoft Azure SQL Database

MySQL

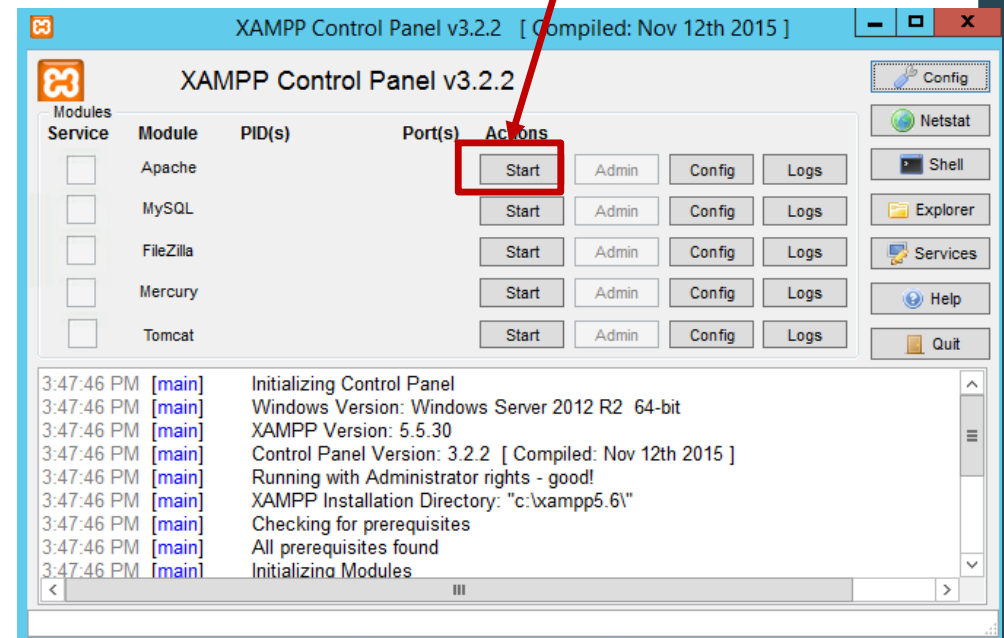
- MySQL is the most popular relational database system.
- What is MySQL?
 - Used on the web
 - Runs on a server
 - For both small and large applications
 - Very fast, reliable, and easy to use
 - Uses standard SQL (structured query language)
 - Compiles on a number of platforms
 - Free to download and use
 - Developed, distributed, and supported by oracle corporation

การติดตั้ง database

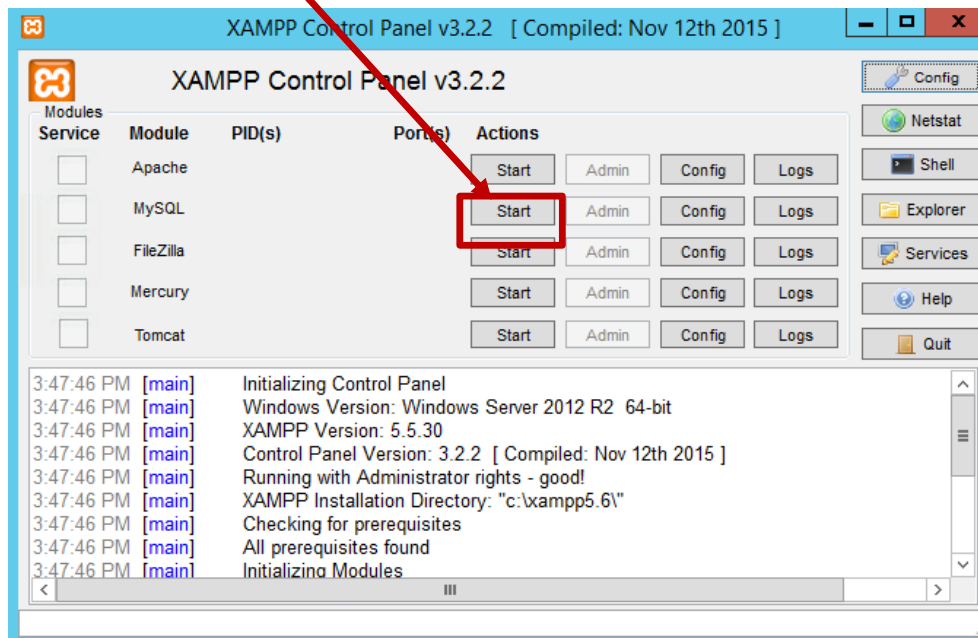
เลือกใช้ phpMyAdmin หรือ MySQL Workbench ก็ได้

ขั้นตอน local server and MySQL setup

- Download XAMPP
(<https://www.apachefriends.org/index.html>)
- Install XAMPP ลงบนเครื่องคอมพิวเตอร์
- ทำการ start Apache ในโปรแกรม XAMPP

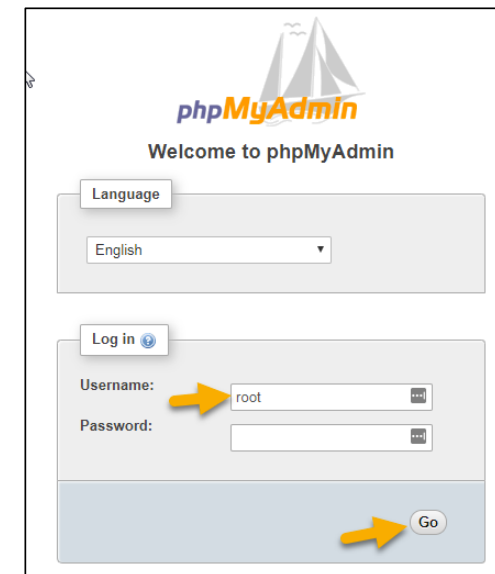
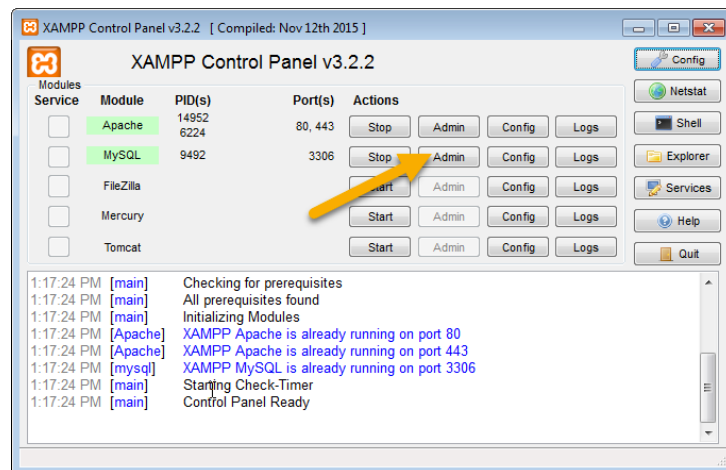


■ 4. ทำการ start MySQL ในโปรแกรม XAMPP



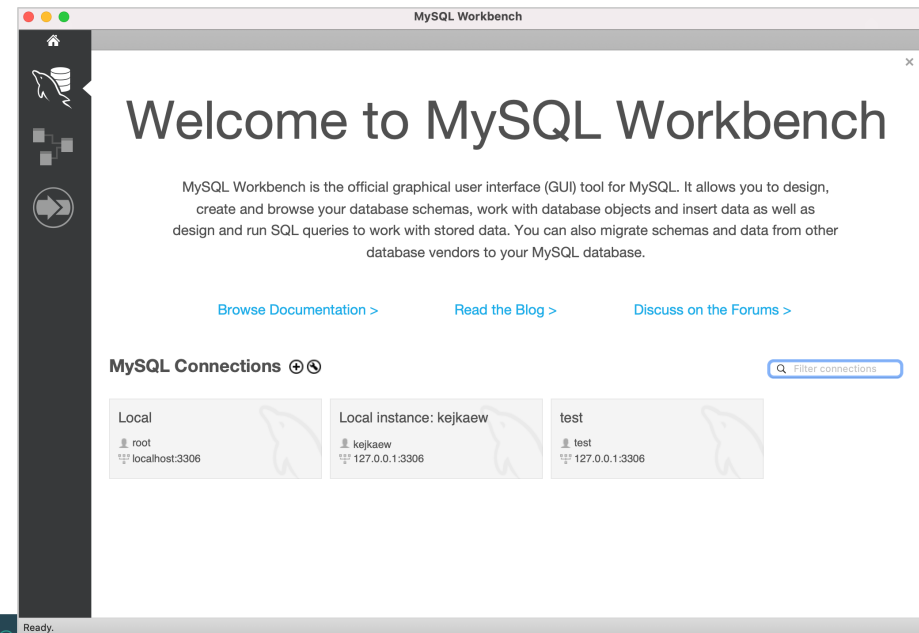
5. เนื่องจาก XAMP ประกอบไปด้วย phpMyAdmin อยู่แล้ว ดังนั้น กดที่ Admin เพื่อ
เข้า phpMyAdmin หรือ พิมพ์ URL ที่ web browser:

<http://localhost/phpmyadmin/>

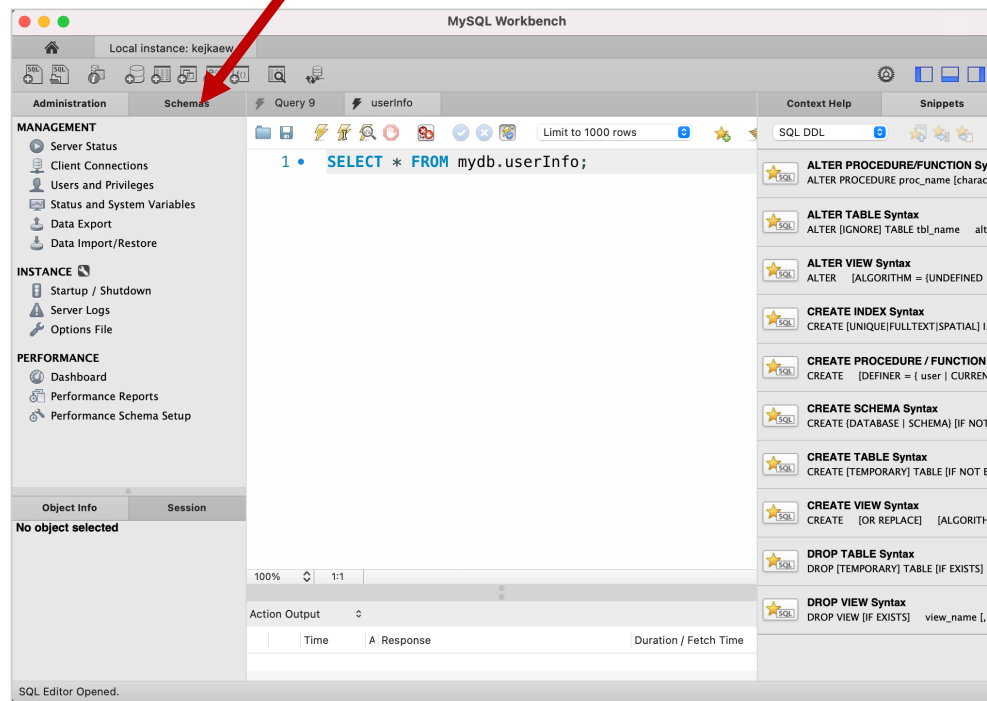


MySQL Workbench

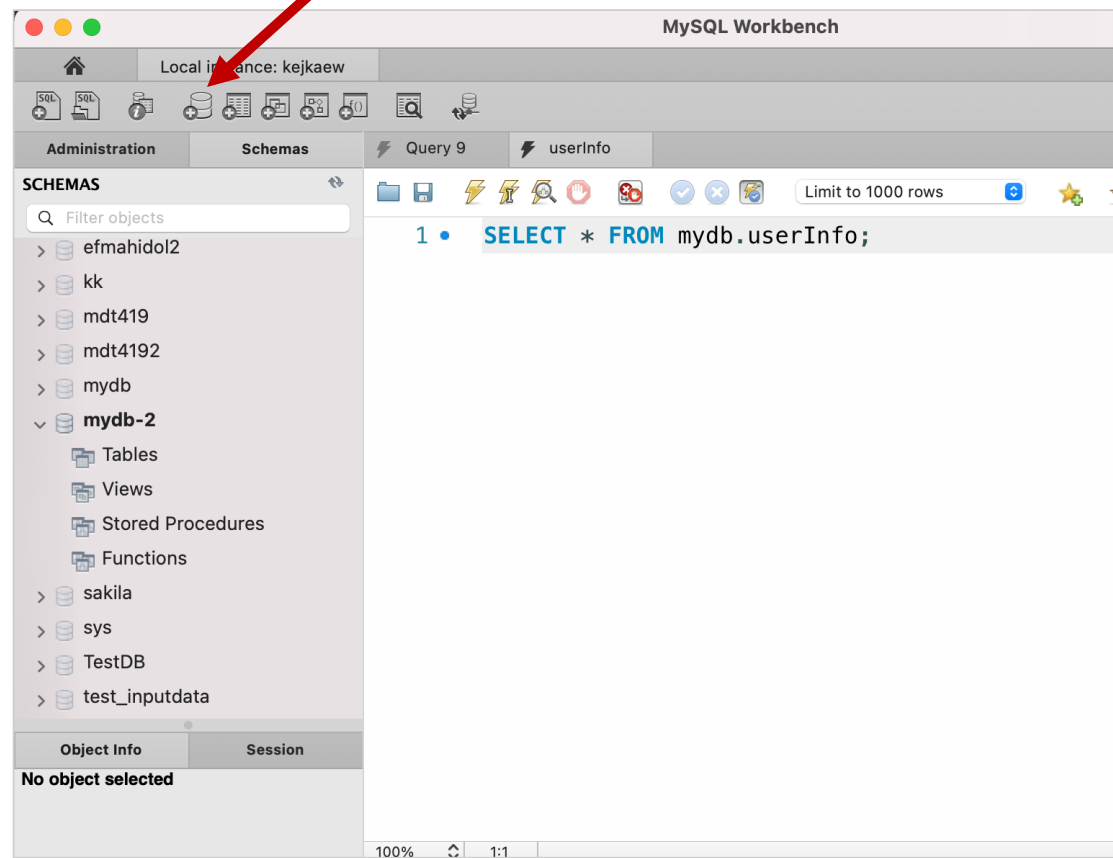
- Download MySQL Workbench ได้จาก <https://dev.mysql.com/downloads/workbench/>
- ทำการติดตั้ง MySQL Workbench



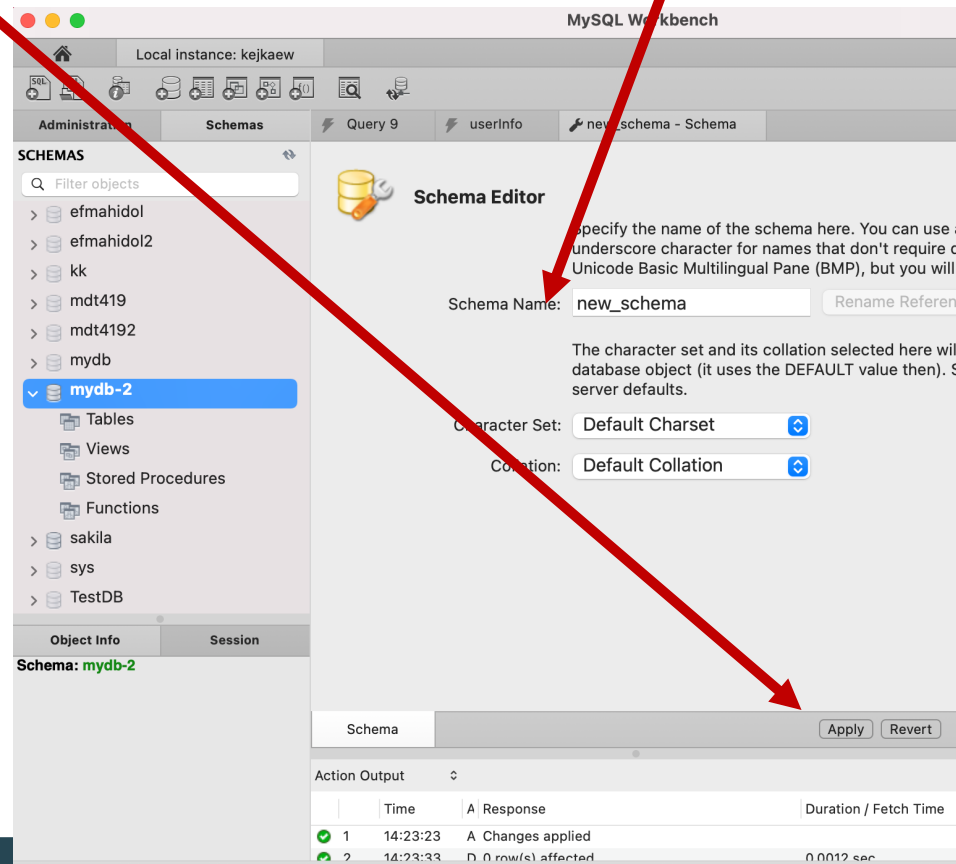
■ ทำการเลือก tab: Schemas



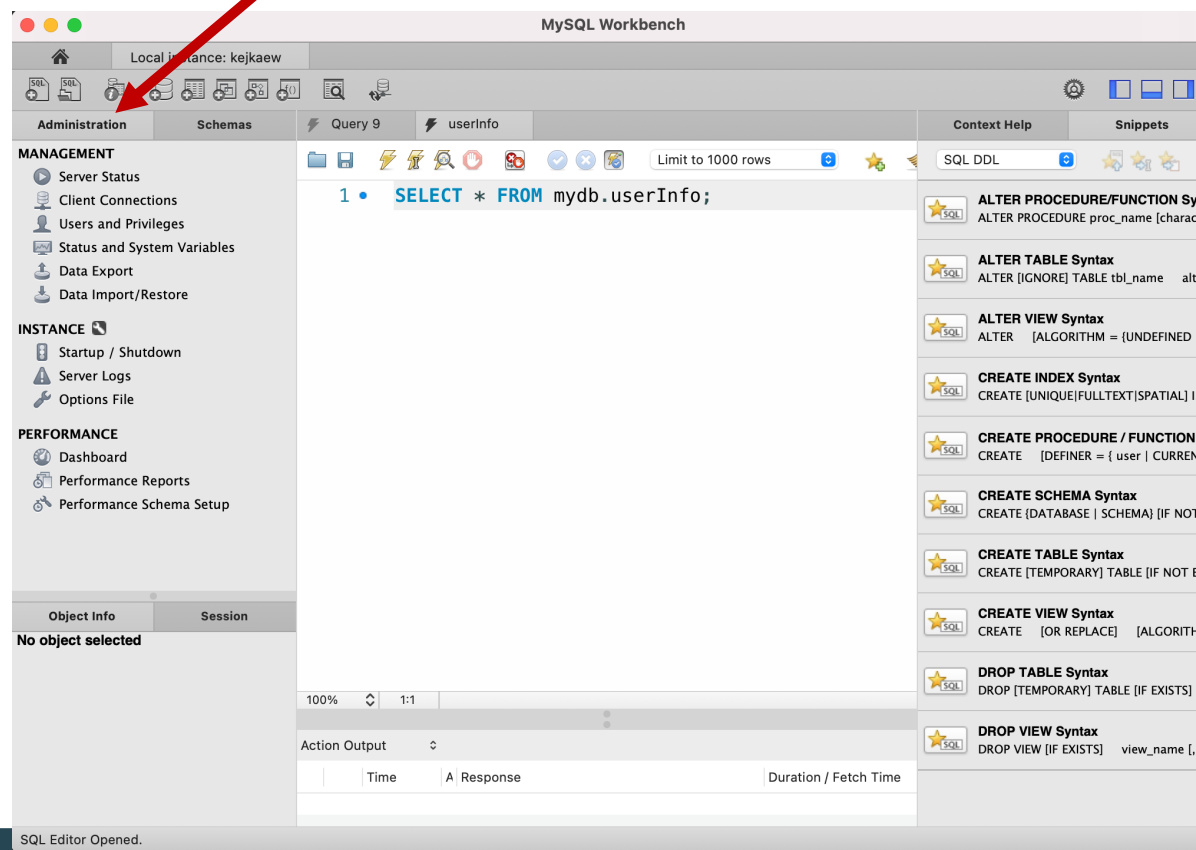
- ถ้าต้องการสร้าง Database ให้กดที่ปุ่ม



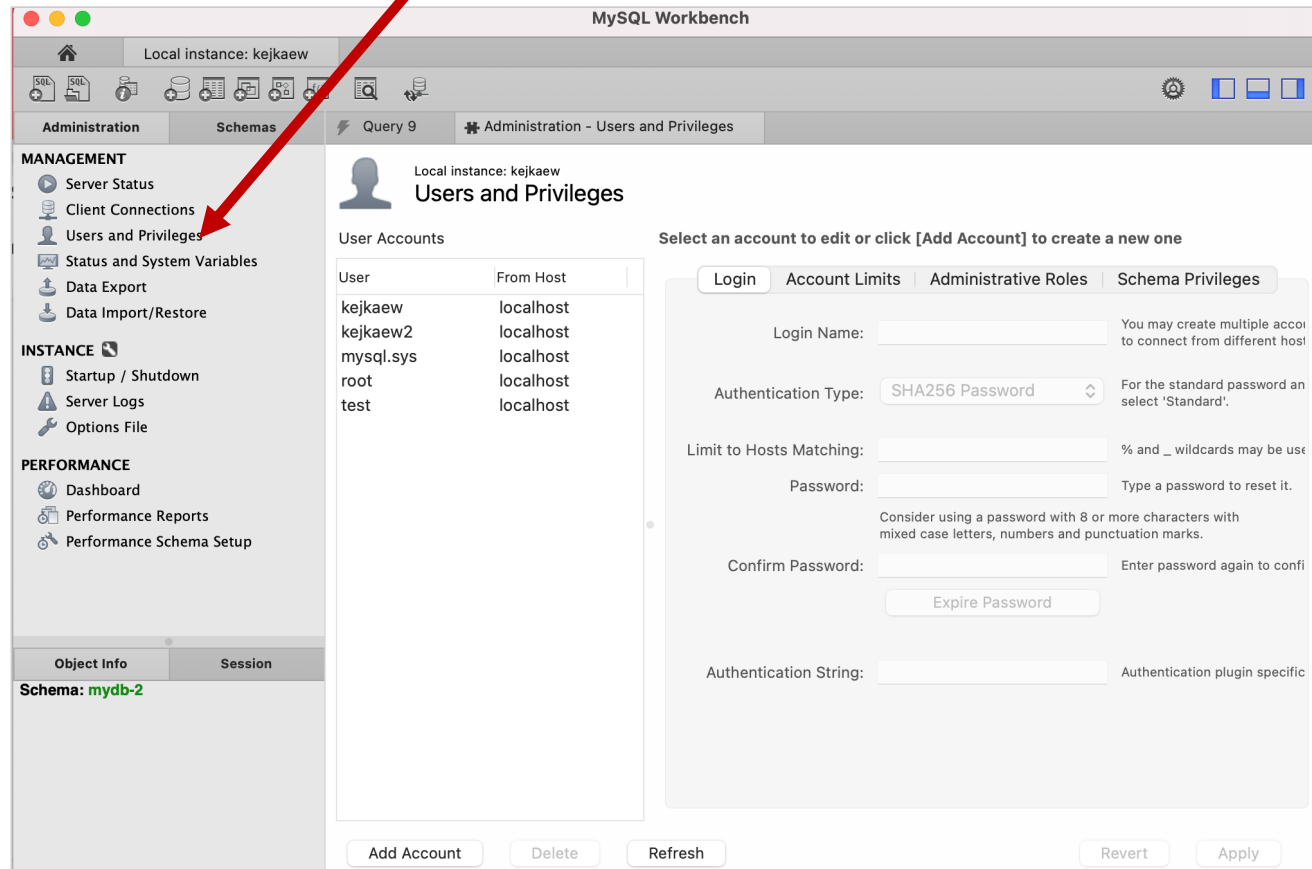
- ทำการตั้งชื่อ database ที่ Schema Name และกด Apply (เราจะตั้งชื่อว่า: mydb)



- ถ้าต้องการกำหนด username และ password ใหม่ให้กดที่ Administration tab



■ ทำการเลือก Users and Privileges

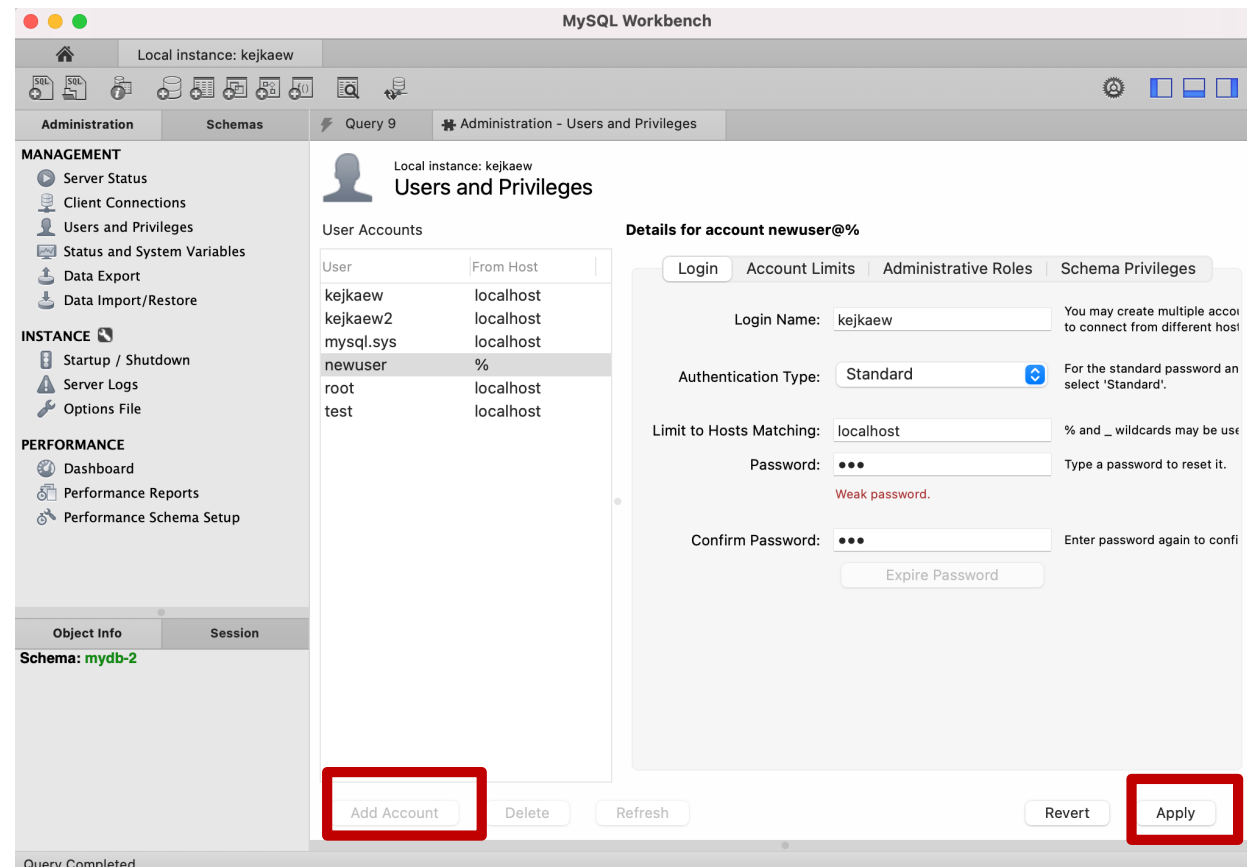


The screenshot shows the MySQL Workbench interface for the 'Local instance: kejkaw'. The 'Administration' tab is active, and the 'Users and Privileges' option is selected in the left sidebar. The main window displays the 'Users and Privileges' configuration page. A table lists existing user accounts, and a form on the right allows for creating or editing a user account.

User	From Host
kejkaw	localhost
kejkaw2	localhost
mysql.sys	localhost
root	localhost
test	localhost

Buttons at the bottom: Add Account, Delete, Refresh, Revert, Apply.

- ทำการกด Add Account ตรง Details for account newuser@% ให้กำหนดรายละเอียด เช่น



Install MySQL module

- การใช้ MySQL ต้อง install MySQL module ก่อน โดยใช้
- To install it, use the following code

```
npm install mysql
```

- การใช้ MySQL ต้องทำการ import ก่อน

```
const mysql = require('mysql');
```

ตัวอย่างที่จะใช้

Add data
Please enter a username:

Please enter an email:

Update email
Please enter a username:

Please enter a new email:

Delete data
Please enter a username you want to delete:

เชื่อมต่อ database

- ทำการสร้างการเชื่อมต่อก่อน โดยสร้าง connection object การเรียก method: `createConnection()` ของ `mysql` และกำหนด รายละเอียดการเข้าถึง database ประกอบด้วย `host`, `user`, `password` and `database`
- ทำการเชื่อมต่อ database โดยใช้เรียก method: `connect()` ของ connection object

ตัวอย่าง

```
1  const express = require('express');
2  const app = express();
3  const hostname = 'localhost';
4  const port = 3001;
5  const bodyParser = require('body-parser');
6  const mysql = require('mysql');
7
8  // app.use(express.static(__dirname));
9  app.use(express.static('public'));
10 app.use(bodyParser.json());
11 app.use(bodyParser.urlencoded({extended: false}));
12
13 // ใส่ค่าตามที่เราตั้งไว้ใน mysql
14 const con = mysql.createConnection({
15   host: "localhost",
16   user: "kejkaew",
17   password: "mdt419",
18   database: "mydb"
19 })
```

```
21 con.connect(err => {
22   if(err) throw(err);
23   else{
24     console.log("MySQL connected");
25   }
26 })
```

สร้าง table ใน database

- สร้าง table โดยจะใช้คำสั่ง sql ดังนี้

```
CREATE TABLE table_name (
```

```
    column1 datatype,
```

```
    column2 datatype,
```

```
    column3 datatype,
```

```
    ....
```

```
);
```

- Column parameters เป็นการกำหนด names ของ columns ใน table.

- Datatype parameter เป็นการกำหนด type of data ใน column (e.g. varchar, integer, date, etc.).

- sql สำหรับสร้างตาราง

```
let sql = "CREATE TABLE IF NOT EXISTS userInfo (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    reg_date TIMESTAMP,  
    username VARCHAR(255),  
    email VARCHAR(100),  
    password VARCHAR(100),  
    img VARCHAR(100));"
```

- ใช้ query method ใน connection object เพื่อทำการร้องขอ ตามคำสั่ง sql ที่กำหนดใน query method

ตัวอย่าง query method

```
28 let tablename = "userInfo";
29
30 const queryDB = (sql) => {
31   return new Promise((resolve, reject) => {
32     con.query(sql, (err, result, fields) => {
33       if (err) reject(err);
34       else
35         resolve(result)
36     })
37   })
38 }
```

เพิ่มข้อมูลลงในตาราง

- เพิ่มข้อมูลลงในตารางจะใช้คำสั่ง ดังนี้

```
INSERT INTO table_name (column1, column2, column3, ...)  
VALUES (value1, value2, value3, ...);
```

ตัวอย่างเช่น

```
sql = `INSERT INTO userInfo (username, email) VALUES  
("${req.body.username}", "${req.body.email})`;
```

ตัวอย่าง

```
41 // create tabl and add data to database
42 app.post("/addDB",async (req,res) => {
43   let sql = "CREATE TABLE IF NOT EXISTS userInfo (id INT AUTO_INCREMENT PRIMARY KEY,
44     reg_date TIMESTAMP, username VARCHAR(255), email VARCHAR(100),password VARCHAR(100),img
45     VARCHAR(100))";
46   let result = await queryDB(sql);
47   sql = `INSERT INTO userInfo (username, email) VALUES ("${req.body.username}", "${req.body.
48     email})`;
49   result = await queryDB(sql);
46   console.log("New record created successfully");
47   res.end("New record created successfully");
48 })
```

Add data

Please enter a username:

Please enter an email:

Add

Output: server

```
Server running at http://localhost:3001/
MySQL connected
New record created successfully
```

Output: html

← → ↺ 🏠 ⓘ localhost:3001/addDB

📱 Apps 📁 Home | ACS 📖 Dictionary and Th... 🔍

New record created successfully

แก้ไขข้อมูลใน database

- แก้ไขข้อมูลในตารางจะใช้คำสั่ง ดังนี้

UPDATE *table_name*

SET *column1* = *value1*, *column2* = *value2*, ...

WHERE *condition*;

ตัวอย่างเช่น

```
let sql = `UPDATE ${tablename} SET email = '${req.body.email}' WHERE  
username = '${req.body.username}'`;
```

ตัวอย่าง

```
51 // update data
52 app.post("/updateDB", async (req, res) => {
53   let sql = `UPDATE ${tablename} SET email = '${req.body.email}' WHERE username = '${req.
54   body.username}'`;
55   let result = await queryDB(sql);
56   console.log(result);
57   res.end("Record updated successfully");
58 })
```

Update email

Please enter a username:

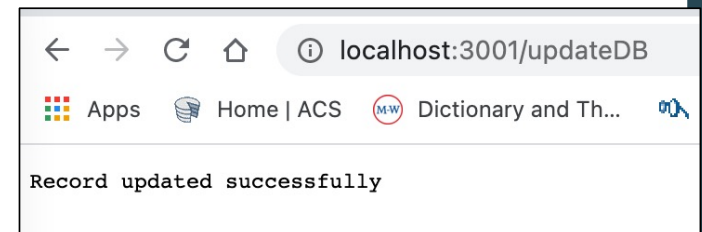
Please enter a new email:

Delete data

Output: server

```
MySQL connected
OkPacket {
  fieldCount: 0,
  affectedRows: 1,
  insertId: 0,
  serverStatus: 34,
  warningCount: 0,
  message: '(Rows matched: 1 Changed: 1 Warnings: 0',
  protocol41: true,
  changedRows: 1
}
```

Output: html



ลบข้อมูล

- ลบข้อมูลในตารางจะใช้คำสั่ง ดังนี้

DELETE FROM *table_name* WHERE *condition*;

ตัวอย่างเช่น

```
let sql = `DELETE FROM ${tablename} WHERE username =  
'${req.body.username}'`;
```

ตัวอย่าง

```
59 // delete data
60 app.post("/deleteDB", async (req, res) => {
61     let sql = `DELETE FROM ${tablename} WHERE username = '${req.body.username}'`;
62     let result = await queryDB(sql);
63     console.log(result);
64     res.end("Record deleted successfully");
65 })
```

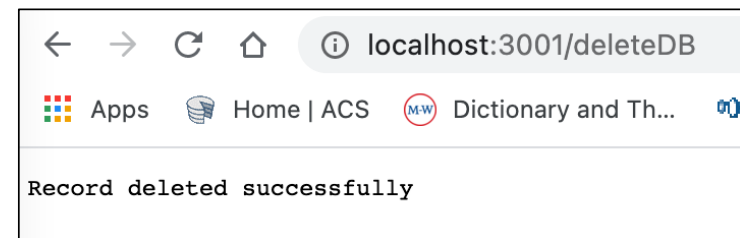
Delete data

Please enter a username you want to delete:

Output: server

```
MySQL connected
OkPacket {
  fieldCount: 0,
  affectedRows: 1,
  insertId: 0,
  serverStatus: 34,
  warningCount: 0,
  message: '',
  protocol41: true,
  changedRows: 0
}
```

Output: html



เลือกข้อมูลในตาราง

- เลือกข้อมูลในตารางจะใช้คำสั่ง ดังนี้

```
SELECT column1, column2, ...  
FROM table_name;
```

- หรือ ทำการเลือกทุก columns

```
SELECT * FROM table_name;
```

ตัวอย่างเช่น

```
67 // show data
68 app.get("/showDB", async (req,res) => {
69     // let sql = `SELECT * FROM ${tablename}`;
70     let sql = `SELECT id, username, email FROM ${tablename}`;
71     let result = await queryDB(sql);
72     result = Object.assign({},result);
73     console.log(result);
74     res.json(result);
75 })
```

Show data

Output: server

```
{
  '0': RowDataPacket {
    id: 2,
    username: 'kejkaew',
    email: 'ketnas.wawa@gmail.com'
  }
}
```

Output: html

Show data

id	username	email
2	kejkaew	ketnas.wawa@gmail.com

Assignment 12

- Download assignment12.zip file
- เราจะทำการสร้างระบบ Register และ Login สำหรับ social media โดย Register จะทำการ add วันที่, username, password และรูป avatar ลงใน table ใน database
- หน้า Login จะทำการตรวจสอบ username and password จากข้อมูลใน database เพื่อเข้าไปหน้า feed.html หลังจากนั้นในหน้า feed เราจะทำการเรียก database เพื่อแสดงข้อความที่เคย post ไปแล้วออกมา
- มีการ check cookie ก่อนที่จะเข้าหน้า feed ถ้าไม่มี cookie ที่ชื่อ username จะไม่สามารถเข้าหน้า feed ได้

-
- ทำระบบ Post ข้อความให้สมบูรณ์โดย save เพิ่มในอีก table และ post บนลงในพื้นที่ที่กำหนดให้
 - สร้าง profile picture สำหรับ social media ของเรา โดยการ
 - Upload image file ลง server
 - แก้ไขข้อมูลใน database เมื่อมีการ อัปเดต image file
 - แสดง รูป profile picture ที่เรา upload ลงไปในพื้นที่ที่กำหนดให้
 - ส่งด้วยนะคะ (แค่ 2 files คือ feed.js และ server.js)

localhost

Register

Username:

Email:

Password:

Register

Week 12: Classroom game

- ทำ classroom game ใน LEB2 ก่อนสัปดาห์หน้า