Assignment on Week 4

Create the following tables in a database named "roster". Make sure that your database and tables are named exactly as follows including matching case.

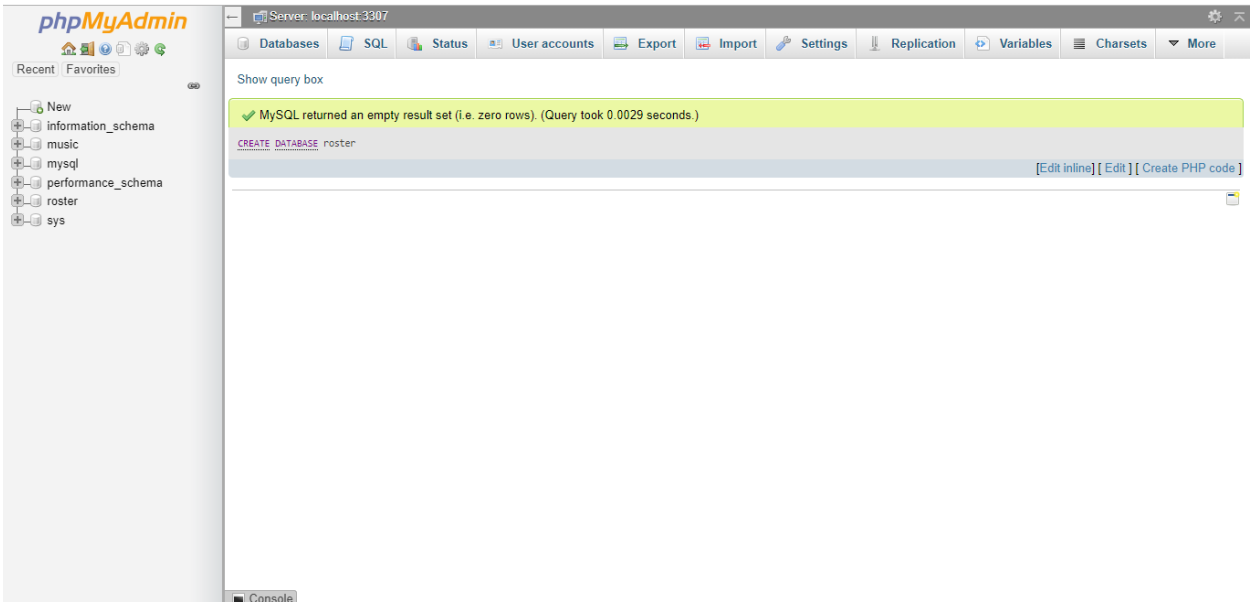
Data:

You will normalize the following data (each user gets different data), and insert the following data items into your database, creating and linking all the foreign keys properly. Encode instructor with a role of 1 and a learner with a role of 0.

```
Orin, si106, Instructor
Ishaal, si106, Learner
Kiera, si106, Learner
Rhiah, si106, Learner
Yuri, si106, Learner
Nabeel, si110, Instructor
Aonghus, si110, Learner
Charleigh, si110, Learner
Malachy, si110, Learner
Tayye, si110, Learner
Lisandro, si206, Instructor
Kallan, si206, Learner
Micheal, si206, Learner
Reese, si206, Learner
Sheigh, si206, Learner
```

Step 1: Create a Database

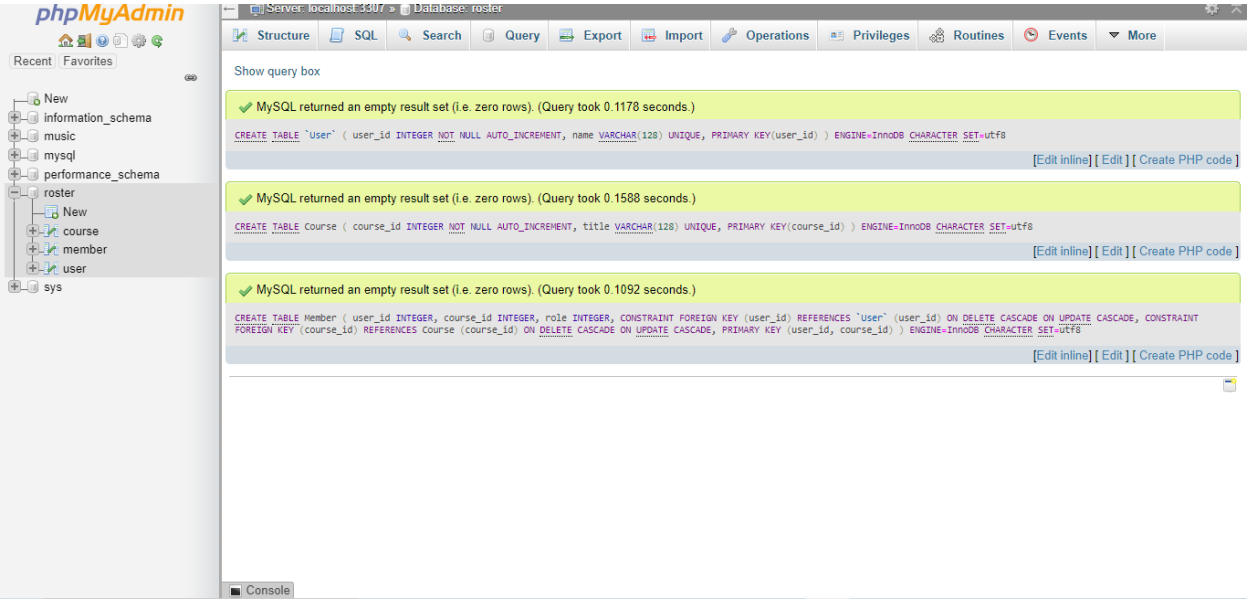
```
CREATE DATABASE roster;
```



Step 2: Create Tables (Course, Members, User)

```
USE roster;
CREATE TABLE `User` (
    user_id     INTEGER NOT NULL AUTO_INCREMENT,
    name        VARCHAR(128) UNIQUE,
    PRIMARY KEY(user_id)
) ENGINE=InnoDB CHARACTER SET=utf8;

CREATE TABLE Course (
    course_id     INTEGER NOT NULL AUTO_INCREMENT,
    title         VARCHAR(128) UNIQUE,
    PRIMARY KEY(course_id)
) ENGINE=InnoDB CHARACTER SET=utf8;

CREATE TABLE Member (
```

```
    user_id       INTEGER,
    course_id     INTEGER,
    role          INTEGER,

    CONSTRAINT FOREIGN KEY (user_id) REFERENCES `User` (user_id)
      ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT FOREIGN KEY (course_id) REFERENCES Course (course_id)
       ON DELETE CASCADE ON UPDATE CASCADE,

    PRIMARY KEY (user_id, course_id)
) ENGINE=InnoDB CHARACTER SET=utf8;
```



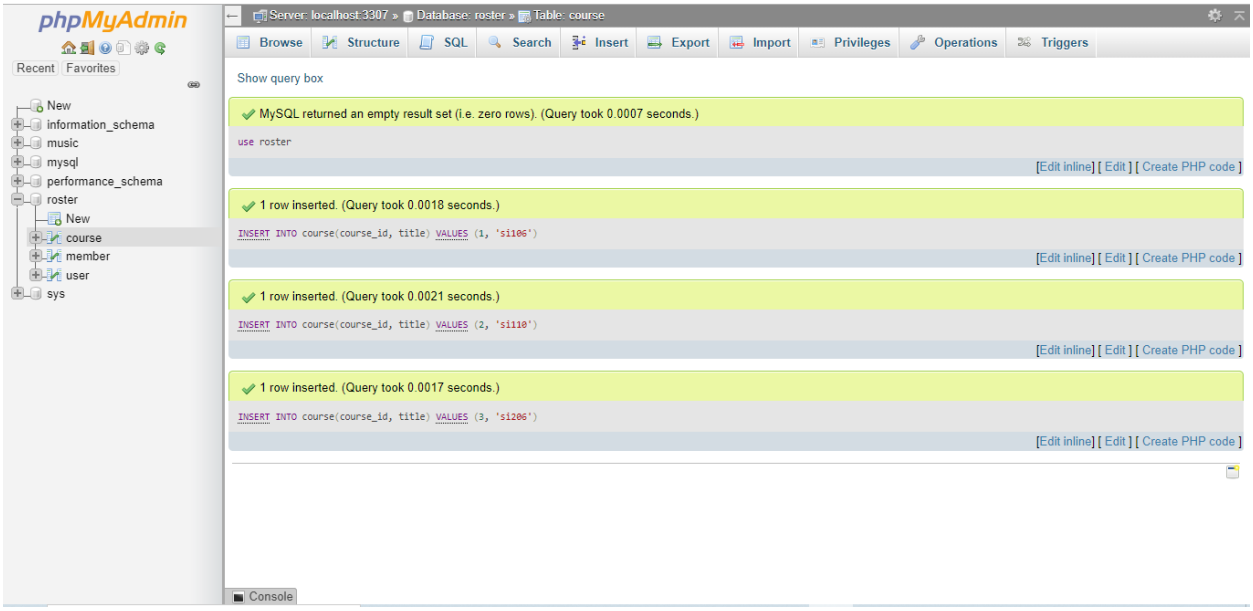
Step 3: Insert Users in the User Table

```
use roster;
INSERT INTO user (user_id,name) VALUES (1, "Orin");
INSERT INTO user (user_id,name) VALUES (2, "Ishaal");
INSERT INTO user (user_id,name) VALUES (3, "Kiera");
INSERT INTO user (user_id,name) VALUES (4, "Rhiah");
INSERT INTO user (user_id,name) VALUES (5, "Yuri");
INSERT INTO user (user_id,name) VALUES (6, "Nabeel");
```

```
INSERT INTO user (user_id,name) VALUES (7, "Aonghus");
INSERT INTO user (user_id,name) VALUES (8, "Charleigh");
INSERT INTO user (user_id,name) VALUES (9, "Malachy");
INSERT INTO user (user_id,name) VALUES (10, "Tayye");
INSERT INTO user (user_id,name) VALUES (11, "Lisandro");
INSERT INTO user (user_id,name) VALUES (12, "Kallan");
INSERT INTO user (user_id,name) VALUES (13, "Micheal");
INSERT INTO user (user_id,name) VALUES (14, "Reese");
INSERT INTO user (user_id,name) VALUES (15, "Sheigh");
```

Step 4: Insert Courses in the Course Table

```
use roster;
INSERT INTO course(`course_id`, `title`) VALUES (1, 'si106');
INSERT INTO course(`course_id`, `title`) VALUES (2, 'si110');
INSERT INTO course(`course_id`, `title`) VALUES (3, 'si206');
```

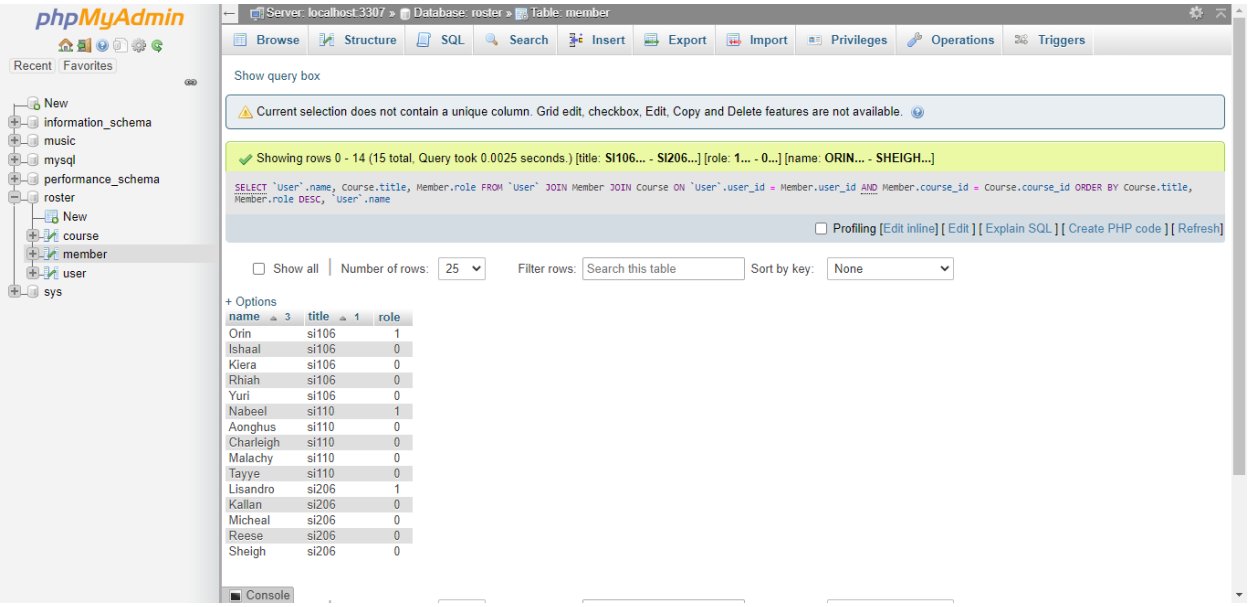


Step 5: Insert Members with the 1 and 0 roles in the Member Table

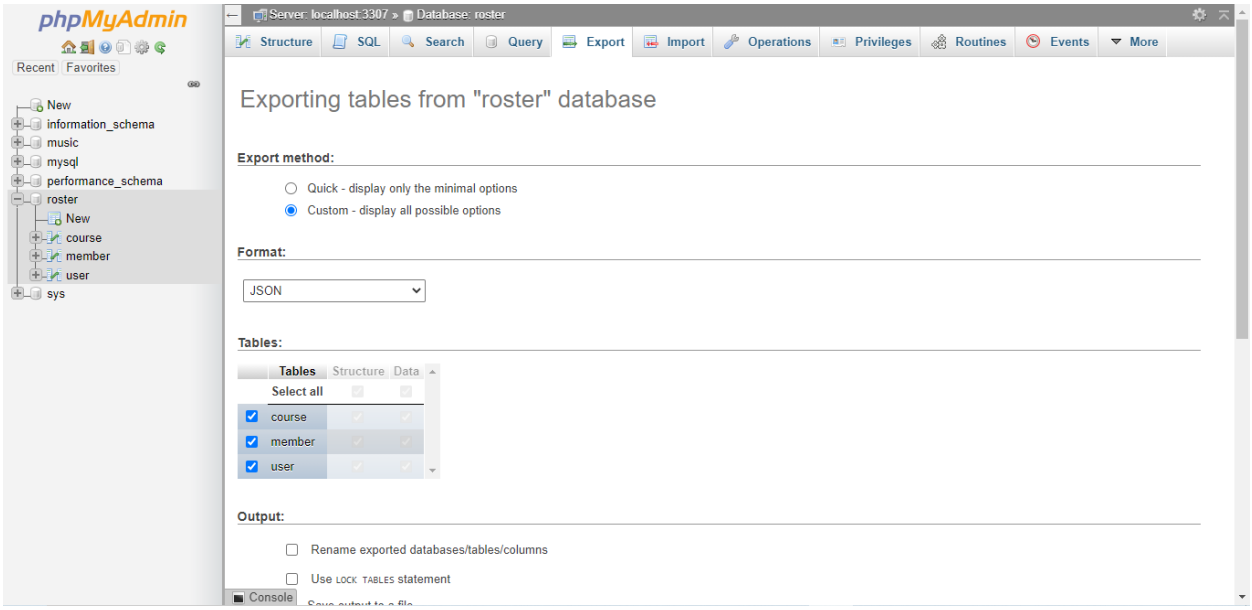
```
use roster;
INSERT INTO member (user_id,course_id,role) VALUES (1, 1, 1);
INSERT INTO member (user_id,course_id,role) VALUES (2, 1, 0);
INSERT INTO member (user_id,course_id,role) VALUES (3, 1, 0);
INSERT INTO member (user_id,course_id,role) VALUES (4, 1, 0);
INSERT INTO member (user_id,course_id,role) VALUES (5, 1, 0);
INSERT INTO member (user_id,course_id,role) VALUES (6, 2, 1);
INSERT INTO member (user_id,course_id,role) VALUES (7, 2, 0);
INSERT INTO member (user_id,course_id,role) VALUES (8, 2, 0);
INSERT INTO member (user_id,course_id,role) VALUES (9, 2, 0);
INSERT INTO member (user_id,course_id,role) VALUES (10, 2, 0);
INSERT INTO member (user_id,course_id,role) VALUES (11, 3, 1);
INSERT INTO member (user_id,course_id,role) VALUES (12, 3, 0);
INSERT INTO member (user_id,course_id,role) VALUES (13, 3, 0);
INSERT INTO member (user_id,course_id,role) VALUES (14, 3, 0);
INSERT INTO member (user_id,course_id,role) VALUES (15, 3, 0);
```

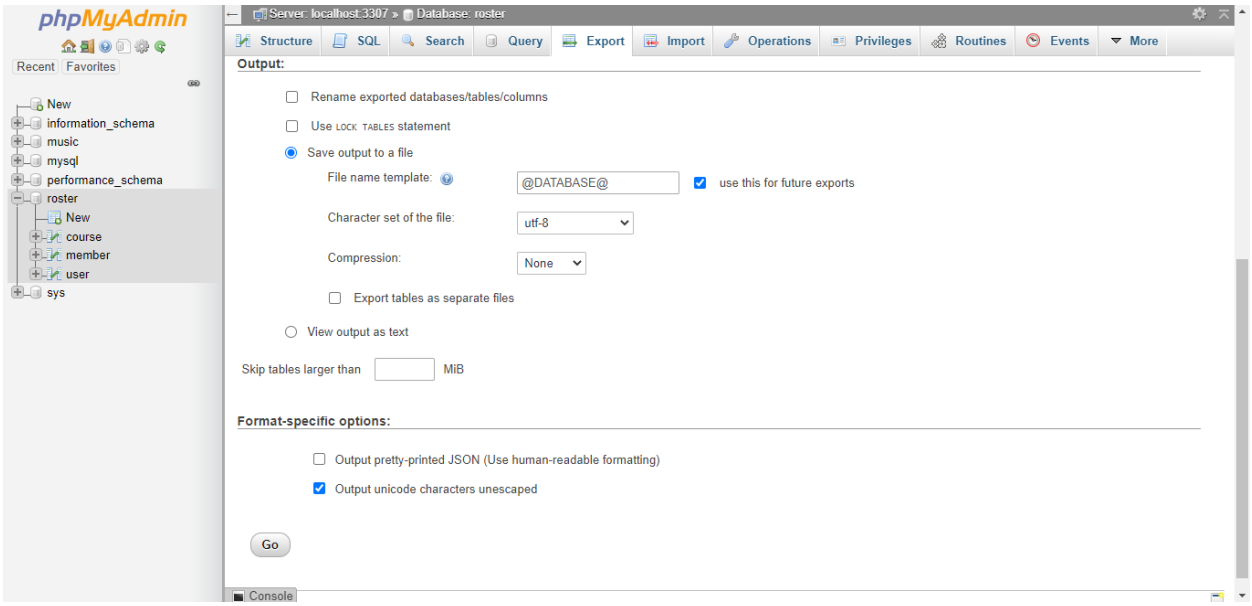
Step 6: Join all the tables

```
SELECT `User`.name, Course.title, Member.role
    FROM `User` JOIN Member JOIN Course
    ON `User`.user_id = Member.user_id AND Member.course_id = Course.course_id
    ORDER BY Course.title, Member.role DESC, `User`.name;
```



Step 7: Export as JSON (The output should be JSON)


phpMyAdmin
Recent Favorites
New
information_schema
music
mysql
performance_schema
roster
New
course
member
user
sys
Server: localhost:3307 » Database: roster
Structure SQL Search Query Export Import Operations Privileges Routines Events More
Exporting tables from "roster" database
Export method:
Quick - display only the minimal options
Custom - display all possible options
Format:
JSON
Tables:
Tables Structure Data
Select all
course
member
user
Output:
Rename exported databases/tables/columns
Use LOCK TABLES statement
Console


phpMyAdmin
Recent Favorites
New
information_schema
music
mysql
performance_schema
roster
New
course
member
user
sys
Server: localhost:3307 » Database: roster
Structure SQL Search Query Export Import Operations Privileges Routines Events More
Output:
Rename exported databases/tables/columns
Use LOCK TABLES statement
Save output to a file
File name template: @DATABASE@ use this for future exports
Character set of the file: utf-8
Compression: None
Export tables as separate files
View output as text
Skip tables larger than MiB
Format-specific options:
Output pretty-printed JSON (Use human-readable formatting)
Output unicode characters unescaped
Go
Console

Voila!