Assignment Specifications: Many-to-Many Databases

Example: Walkthrough for assignment

Perform the instructions below and upload your JSON export of the resulting database in your individual assignment page.

Tables for the Assignment

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

```
DROP TABLE IF EXISTS Member;
DROP TABLE IF EXISTS `User`;
DROP TABLE IF EXISTS Course;

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;
```

Example: Data

You will normalize the following data (each user gets different data on the autograder page), 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.

```
Taliesin, si106, Instructor
Denver, si106, Learner
Juwairiyah, si106, Learner
Kainui, si106, Learner
Zoya, si106, Learner
Aisha, si110, Instructor
Artemis, si110, Learner
Danna, si110, Learner
Dennis, si110, Learner
Tyler, si110, Learner
Kirstin, si206, Instructor
Allisha, si206, Learner
Carra, si206, Learner
Idahosa, si206, Learner
Iliana, si206, Learner
```

You can test to see if your data has been entered properly with the following SQL statement.

```
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
```

The order of the data and number of rows that comes back from this query should be the same as above. There should be no missing or extra data in your query.

Submitting Your Assignment

When you have the data all inserted, use phpMyAdmin to Export the data as follows:

- Select the database (do not select a table within the database)
- Select the Export Tab
- Select "Custom - display all possible options"
- Select "Save output to a file"
- Set the format to JSON
- Do not select "pretty print" the output
- Leave everything else as default and run the export.

The output will be on a file named "roster.json" that should look like the following: Depending on the version of phpMyAdmin there are 2 formats it exports.

```
/**
 Export to JSON plugin for PHPMyAdmin
 @version 0.1
 */

// Database 'roster'
// roster.Course

[{"course_id":"6","title":"si106"}, ... }]

// roster.Member

[{"user_id":"1","course_id":"1","role":"1"}, ... }]

// roster.User

[{"user_id":"15","name":"Areez"}, ... }]
```

```
[
{"type":"header","version":"4.7.0","comment":"Export to JSON plugin for PHPMyAdmin"},
{"type":"database","name":"roster"},
{"type":"table","name":"course","database":"roster","data":
[
{"course_id":"1","title":"si106"},
....
]
}
,{"type":"table","name":"member","database":"roster","data":
[
{"user_id":"1","course_id":"1","role":"1"},
....
]
}
,{"type":"table","name":"user","database":"roster","data":
[
{"user_id":"6","name":"Aisha"},
....
]
}
]
```