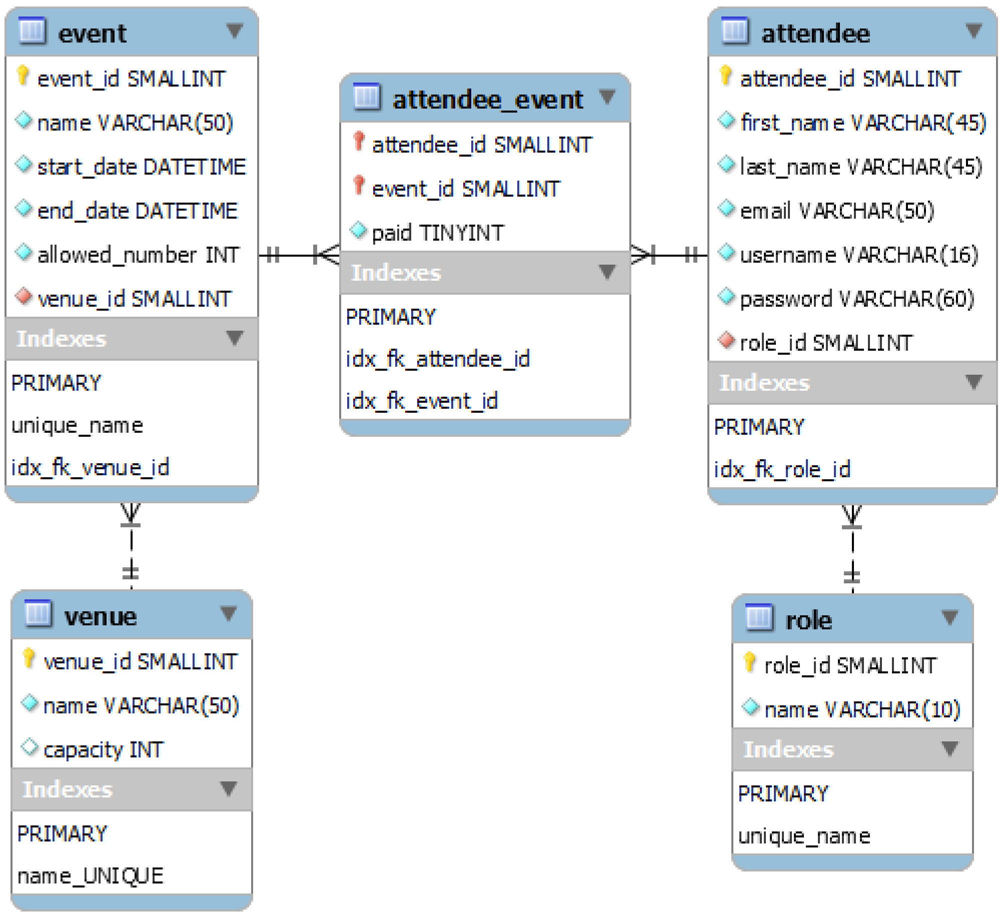
PROJECT 1: OBJECT-ORIENTED DATABASE DRIVEN WEB APPLICATION USING PHP & MYSQL

You are required to create an object-oriented database-driven web application for an event management system using PHP & MySQL. The application should also provide login and registration functionality to only allow access to registered users.

Database

- Use the provided SQL script to create the tables on Solace or/and localhost. This is the EER diagram of the DB the script is going to create:



  - The **attendee** table will contain 2 records – do not delete them. The **role** table will also be populated by the SQL script. There are **2** types of users depending on the permissions: **attendee** (view permission) and **admin** (full control).
  - You need to populate the tables with a minimum of 2 venues, 2 events & 2 attendees per event.
- Passwords need to be hashed.

Web Application

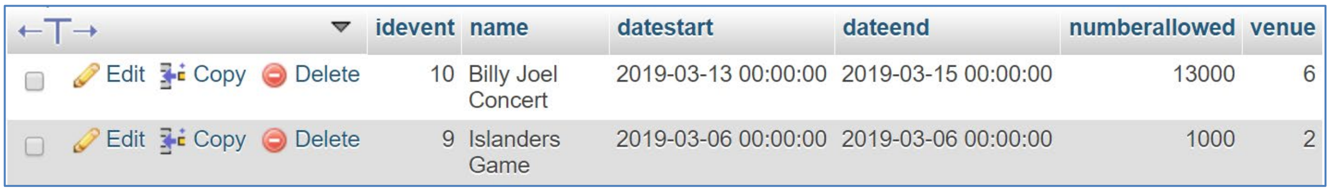
- You will use **PHP sessions** to control access and interaction with the web pages based on roles:
  - **ATTENDEE ROLE**
    - View all events.
    - View events for which the user has registered.
    - Select a particular event for inspection.
    - Register/unregister for an event.

- **ADMIN ROLE**
  - All functionality of an attendee AND
  - Add/Edit/Delete/View any user, venues, events, attendees.
- The home page should allow user login and registration. Once logged in, the layout is adjusted based on the user role. For the logged in user, the logout option should be available.
- To visualize the MySQL data efficiently, you will have to extract records from two or more tables based on certain conditions and join them.

GUI Requirements (Visualizing the Data)

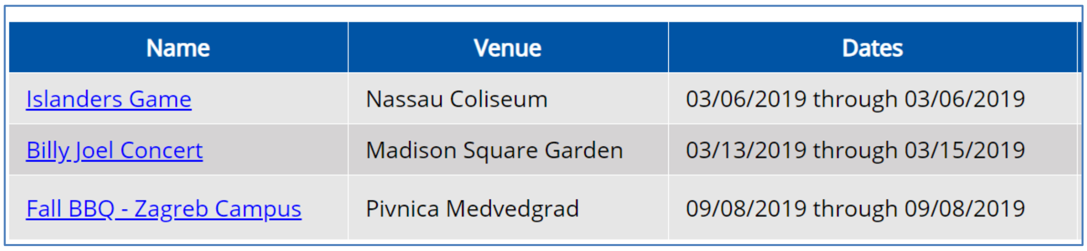
When it comes to the graphical user interface (GUI), it should not simply replicate the database tables. If you provide the same tables, and in the same format as found in the Database itself, then, there is no point in having a web application. The web application should ease the process of editing the DB tables.

For example, let's consider you are the **admin** and would like to edit the **Events** data. As an admin, you do not want to see and edit data that you do not understand as in here (dealing with events and venues IDs):



←T→	idevent	name	datestart	dateend	numberallowed	venue
Edit Copy Delete	10	Billy Joel Concert	2019-03-13 00:00:00	2019-03-15 00:00:00	13000	6
Edit Copy Delete	9	Islanders Game	2019-03-06 00:00:00	2019-03-06 00:00:00	1000	2

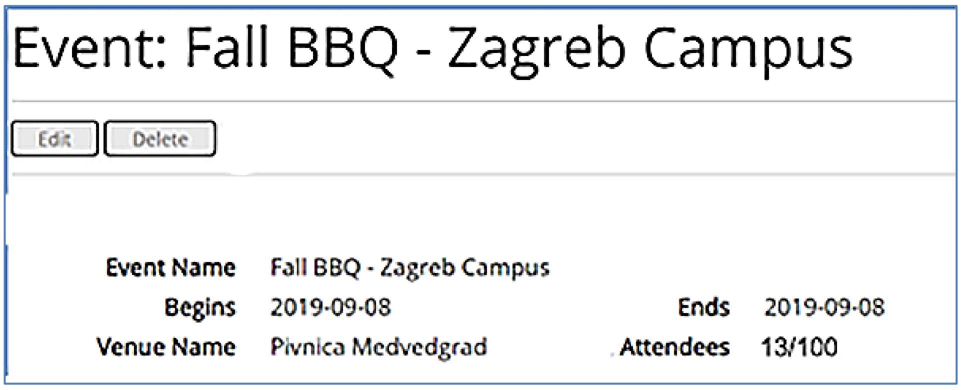
The above table is a replica of the DB table. The admin should see the joined data table as this:



Name	Venue	Dates
Islanders Game	Nassau Coliseum	03/06/2019 through 03/06/2019
Billy Joel Concert	Madison Square Garden	03/13/2019 through 03/15/2019
Fall BBQ - Zagreb Campus	Pivnica Medvedgrad	09/08/2019 through 09/08/2019

where useful data is presented. This kind of table is possible to create only if you collect data from multiple DB tables, and this combined data is what you need to provide and present to the end users (admins & attendees).

Another example: if you as an admin want to edit an event, you should be able to update (in one web page) everything that is event related as in:



Event: Fall BBQ - Zagreb Campus

Event Name	Fall BBQ - Zagreb Campus		
Begins	2019-09-08	Ends	2019-09-08
Venue Name	Pivnica Medvedgrad	Attendees	13/100

Deliverables

- Export your DB tables when you have completed the application to the project's **assets** folder.
- Create a **passwords.txt** file with all usernames and passwords you use and save it to the project's **assets** folder.
- Zip up your entire project (file structure intact) in a folder with your last name and upload it to the dedicated MyCourses Assignments folder.
- Include a link to your application on Solace with all usernames and passwords in the comments section when you upload your project to MyCourses Assignment folder.

- DUE DATE: Beginning of W06

Grading

The project will be graded based on a series of milestones, with each stage requiring timely submission and an in-depth understanding of the work completed. For each milestone, students must schedule an Office Hours (OH) meeting during the corresponding week to explain and defend their progress. During these meetings, students will be evaluated on their ability to articulate their design choices, demonstrate functionality, and explain their implementation. Failure to effectively explain any part of the project will result in a score of zero for the corresponding section of the grading rubric.

Submitting a milestone late will result in a 50% point deduction for that milestone, as applied to the final project's total grade. Additionally, a final OH meeting is mandatory for the completed project. During this session, students must demonstrate a comprehensive understanding of their project. If a student cannot explain their work or demonstrate their understanding, they may receive a grade of zero for the entire project. This policy emphasizes the importance of both technical skills and the ability to communicate your work effectively. Punctual submissions and preparation for milestone discussions are key to success.

Item	**Points**
Frontend designed and implemented following good web design principles. CSS should be defined in external style sheets & the site should pass HTML5 validation.	10
Backend designed and implemented following good SW design principles and patterns (OO approach based on MVC as explained in class)	10
Attendee role functionality (eligible for grading if data is joined) – View all events (2 pts) – **View events for which the user has registered (2 pts)** – **Select a particular event for inspection (2 pts)** – Register/unregister for an event (4 pts)	10
Admin role functionality (eligible for grading if data is joined) – All functionality of an attendee AND – Add/Edit/Delete/View any user, venues, events, attendees. (4*5 pts)	20
PHP sessions used to control access and preserve the state of the web application.	10
All PDO DB queries are parametrized using prepared statements. Note: To access the DB you need to use the PDO extension. Other solutions will lead to 0.	10
The MySQL script is exported and included as part of the project submission (project's assets folder) to ensure the database schema and necessary queries are properly documented and can be easily imported for project review and testing. The DB records include a minimum of 2 venues, 2 events with sessions & 2 attendees per event.	10
All input sanitized	10
All input validated	10
TOTAL	**100**