Contents

THIS PROJECT IS CREATED BY MUHMEDSADIQ JASIM 2ND YEAR NETWORK ENGINEERING DEPT.

Portfolio: https://muhmedsadiqjasim.github.io/Portfolio/

تم إنشاء هذا المشروع بواسطة محمد صادق احمد جاسم – مرحلة ثانية – قسم شبكات

AL-NAHRAIN UNIVERSITY

Database Project

2025-04-20

Creating The Database

```
CREATE DATABASE uscsrs;
USE uscsrs;

CREATE TABLE Student (
student_id INT AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(100) NOT NULL,
email VARCHAR(100) UNIQUE NOT NULL,
department VARCHAR(50) NOT NULL);

CREATE TABLE Club (
club_id INT AUTO_INCREMENT PRIMARY KEY,
club_name VARCHAR(100) NOT NULL,
description TEXT);

CREATE TABLE Session (
session_id INT AUTO_INCREMENT PRIMARY KEY,
club_id INT,
session_title VARCHAR(100) NOT NULL,
session_date DATE NOT NULL,
location VARCHAR(100),
FOREIGN KEY (club_id) REFERENCES Club(club_id));
```

```
CREATE TABLE Registration (
registration_id INT AUTO_INCREMENT PRIMARY KEY,
student_id INT,
session_id INT,
registration_date DATE NOT NULL,
FOREIGN KEY (student_id) REFERENCES Student(student_id),
FOREIGN KEY (session_id) REFERENCES Session(session_id));
```

Normalization

What the database will looks like if we don't use normalization?

The tables of our database will look like that:

student_id	name	email	department	club_name	session_title	session_date	...
1	Muhmedsadiq	example@gmail.com	Network Dept.	IT Students	SOC Introduction	2025-05-01	...

SOOO...

This is called Unnormalized Form (UNF).
We need to organize our database to read, insert and modify the data as we like.

To do that we will use (**1NF**, **2NF** and **3NF**).

1st Normal Form (1NF)

1. Each column should contain atomic (indivisible) values (Atomicity).
2. Each row should have a unique identifier (Primary Key).

We will make 4 tables each table will includes the data in a specific place.
We will have (Student, Club, Session and Registration table).

For example, Student table will look like that:

student_id (PK)	name	session_title	department	registration_date
1	Ahmed Duraid	SOC Introduction	Network Engineering Dept.	2025-04-28

This table looks better than the previous one for sure. We can read, insert and modify the data easily now but also we have something to do to make the database much better.

2nd Normal Form (2NF) – "Before that apply the 1NF"

- Remove Partial Dependencies.

Student Table

student_id (PK)	Name
1	Ahmed Duraid

Session Table

session_id (PK)	club_name	session_date
1	IT Students	2025-05-01

Now we have to do some enhance. let's go to the **3NF**.

3rd Normal Form (3NF) – “Before that apply 1NF & 2NF”

- Remove transitive dependencies.

session_id (PK)	session_title	club_id (FK)	session_date
1	SOC Introduction	1	2025-05-01

Note: I just showed the effect of normalization in general, I didn’t show the full effect on our database.

Useful Queries For Our Database

INSERT – Add a New Student

```
INSERT INTO Student (name, email, department)

VALUES ('Ali Kareem', 'ali.kareem@email.com', 'Computer Engineering');
```

```
mysql> SELECT * FROM Student;
+------------+----------------+---------------------------+----------------------+
| student_id | name           | email                     | department           |
+------------+----------------+---------------------------+----------------------+
|          1 | Ahmed Duraid   | ahmed.duraid@gmail.com    | Network Engineering |
|          4 | Ali Jassim     | ali.jassim@email.com      | Network Engineering |
|          5 | Mohammed Ammar | mohammed.ammar@email.com  | Network Engineering |
+------------+----------------+---------------------------+----------------------+
3 rows in set (0.00 sec)
```

```
mysql> INSERT INTO club (club_name, description)
    -> VALUES ('COIE Students', 'This club is created for the students of College of Information Engineering to give the
 students the opportunity to participate in some IT activities');
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM club;
+---------+--------------+-----------------------------------------------------------------------------------------------
-----------------------------------------------------+
| club_id | club_name    | description
                                                     |
+---------+--------------+-----------------------------------------------------------------------------------------------
-----------------------------------------------------+
|       1 | COIE Students | This club is created for the students of College of Information Engineering to give the stud
ents the opportunity to participate in some IT activities |
+---------+--------------+-----------------------------------------------------------------------------------------------
-----------------------------------------------------+
1 row in set (0.00 sec)
```

```
mysql> INSERT INTO session (club_id, session_title, session_date)
    -> VALUES (1, 'Introdction to NOC', '2025-05-01');
Query OK, 1 row affected (0.01 sec)
```

Update – Add a New Location to the Session

```
mysql> UPDATE session
    -> SET location = '206 New Building'
    -> WHERE club_id = 1;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM session;
+------------+---------+--------------------+--------------+------------------+
| session_id | club_id | session_title      | session_date | location         |
+------------+---------+--------------------+--------------+------------------+
|          1 |       1 | Introdction to NOC | 2025-05-01   | 206 New Building |
+------------+---------+--------------------+--------------+------------------+
1 row in set (0.00 sec)
```

DELETE – Remove a Student's Registration

```
DELETE FROM Registration

WHERE registration_id = 5;
```

SELECT & JOIN – List Students and Their Sessions

```
mysql> SELECT s.name AS student_name, se.session_title, se.session_date, c.club_name
    -> FROM Registration r
    -> JOIN Student s ON r.student_id = s.student_id
    -> JOIN Session se ON r.session_id = se.session_id
    -> JOIN Club c ON se.club_id = c.club_id;
+--------------+--------------------+--------------+---------------+
| student_name | session_title      | session_date | club_name     |
+--------------+--------------------+--------------+---------------+
| Ahmed Duraid | Introdction to NOC | 2025-05-01   | COIE Students |
+--------------+--------------------+--------------+---------------+
1 row in set (0.00 sec)
```

SELECT, COUNT & UNION – Show the Number of registers in the Sessions

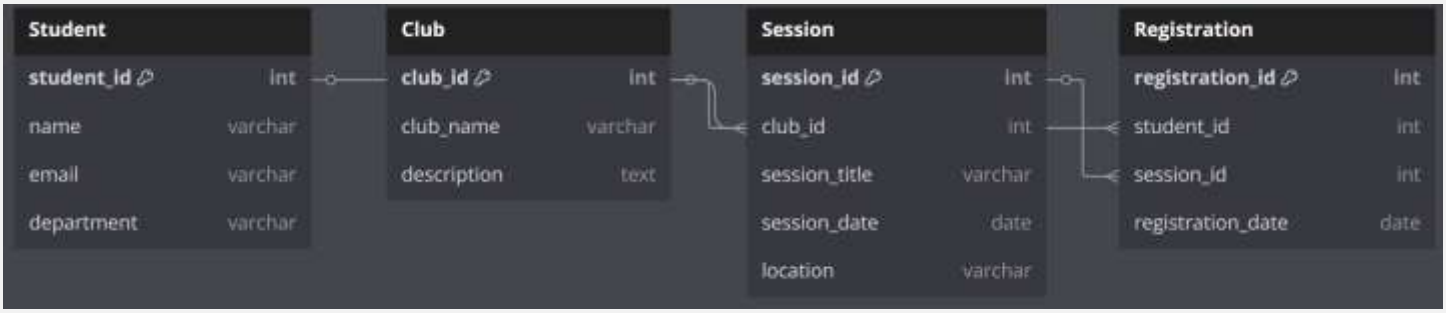
```
mysql> SELECT COUNT(*) FROM Registration WHERE session_id = 1
    -> UNION
    -> SELECT COUNT(*) FROM Registration WHERE session_id = 2;
+----------+
| COUNT(*) |
+----------+
|        1 |
|        0 |
+----------+
2 rows in set (0.00 sec)
```

There are a lot more of queries that will help use in this database, but this is enough to demonstrate the benefits of use SQL for our data.

ER Diagram & Relational Schema

I use this site to help me doing the ER Diagram with the Relational Schema

https://dbdiagram.io/d



Explination

1. **Student to Registration**
   Type: One-to-Many
   One student can register for many sessions.
2. **Registration to Student**
   Type: Many-to-One
   Each registration is linked to one student.
3. **Session to Registration**
   Type: One-to-Many
   One session can have many student registrations.
4. **Registration to Session**
   Type: Many-to-One
   Each registration is for one session.
5. **Club to Session**
   Type: One-to-Many
   One club can create and manage many sessions.
6. **Session to Club**
   Type: Many-to-One
   Each session is part of one club.

Source Code

db.php

```php
<?php
$hostname = "localhost";
$username = "root";
$password = "HamHamHam9911";
$database = "uscsrs";

$conn = new mysqli($hostname, $username, $password, $database);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
?>
```

index.php (main)

```php
<?php include "db.php"; ?>
<!DOCTYPE html>
<html>
<head>
    <title>Events</title>
    <link rel="stylesheet" href="stylez.css">
</head>
<body>
    <h2>Available Events</h2>
    <?php
    $q = $conn->query("
    SELECT se.session_id, se.session_title, se.location, c.club_name
    FROM Session se
    JOIN Club c ON se.club_id = c.club_id
    WHERE se.session_date >= CURDATE()
");
    while ($row = $q->fetch_assoc()):
    ?>
        <div class="event">
            <p><?php echo $row['session_title']; ?></p>
            <p><?php echo $row['club_name']; ?></p>
            <p><?php echo $row['location']; ?></p>
            <a href="register.php?session_id=<?php echo $row['session_id']; ?>">Join</a>
        </div>
    <?php endwhile; ?>
</body>
</html>
```

register.php

```php
<?php include "db.php"; ?>
<!DOCTYPE html>
<html>
<head>
    <title>Join Event</title>
    <link rel="stylesheet" href="stylez.css">
</head>
<body>
<?php
if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $name = $_POST["name"];
    $email = $_POST["email"];
    $department = $_POST["department"];
    $session_id = $_POST["session_id"];

    $stmt = $conn->prepare("INSERT INTO Student (name, email, department) VALUES (?, ?, ?)");
    $stmt->bind_param("sss", $name, $email, $department);
    $stmt->execute();
    $student_id = $conn->insert_id;

    $stmt2 = $conn->prepare("INSERT INTO Registration (student_id, session_id,
registration_date) VALUES (?, ?, CURDATE())");
    $stmt2->bind_param("ii", $student_id, $session_id);
    $stmt2->execute();

    echo "<h2>You joined the event successfully!</h2>";
} else {
    $session_id = $_GET['session_id'] ?? 1;
?>
    <h2>Join Event</h2>
    <form method="post">
        <input type="hidden" name="session_id" value="<?php echo $session_id; ?>">
        <input type="text" name="name" placeholder="Your name" required><br><br>
        <input type="email" name="email" placeholder="Your email" required><br><br>
        <select name="department" required>
            <option value="">Select Department</option>
            <option value="Information Engineering">Information Engineering</option>
            <option value="Computer Science">Computer Science</option>
            <option value="Arts">Arts</option>
        </select><br><br>
        <button type="submit">Join Now</button>
    </form>
<?php } ?>
</body>
</html>
```

stylez.css

```css
body {
    background-color: #f5f5dc;
    font-family: Arial, sans-serif;
    color: #333;
    text-align: center;
    padding: 20px;
}

h2 {
    color: #5a4e3c;
    margin-bottom: 20px;
}

.event {
    background-color: #fff8dc;
    border: 1px solid #ddd;
    margin: 20px auto;
    padding: 15px;
    width: 300px;
    border-radius: 6px;
}

.event a {
    display: inline-block;
    margin-top: 10px;
    background-color: #d2b48c;
    color: white;
    padding: 8px 12px;
    text-decoration: none;
    border-radius: 4px;
}

.event a:hover {
    background-color: #c2a77c;
}

form {
    background-color: #fffaf0;
    padding: 20px;
    margin: 30px auto;
    width: 280px;
    border: 1px solid #ddd;
    border-radius: 6px;
}
```

```css
input,
select {
    width: 90%;
    padding: 6px;
    margin-bottom: 12px;
    border: 1px solid #ccc;
    border-radius: 4px;
}

button {
    background-color: #d2b48c;
    color: white;
    border: none;
    padding: 8px 16px;
    border-radius: 4px;
    cursor: pointer;
}

button:hover {
    background-color: #c2a77c;
}
```

What is solved?

This project solves the problem of managing student participation in university clubs and events. In many universities, it can be difficult to organize events, track which students joined which sessions, and collect student information efficiently.

With this simple website:

1) Students can view all upcoming events without needing to register first.
2) When interested in an event, they can easily sign up with their basic information (name, email, and department).
3) Organizers can later use the database to see who registered for each session.
4) Events that are already expired are not shown, which keeps the interface clean and relevant.

Challenges & Lessons Learned

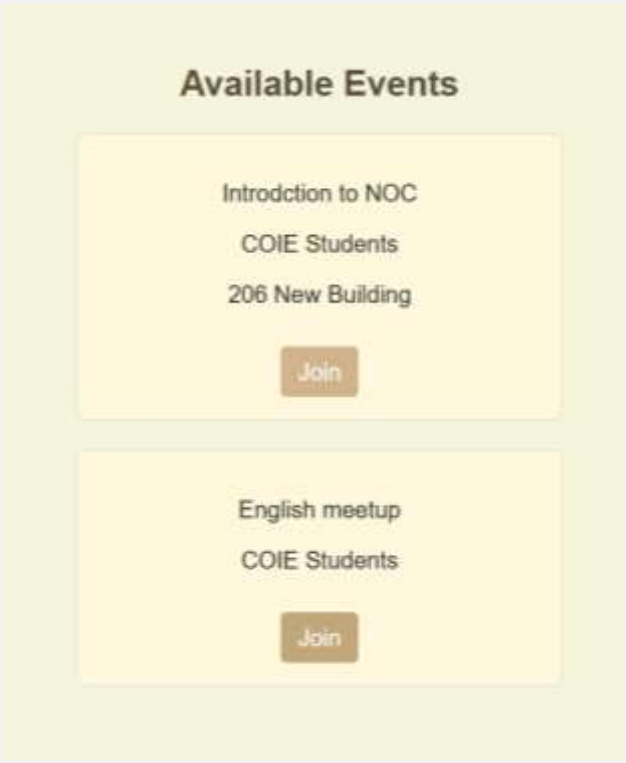
Firstly I hate PHP very very very much especially when I used it at this project.

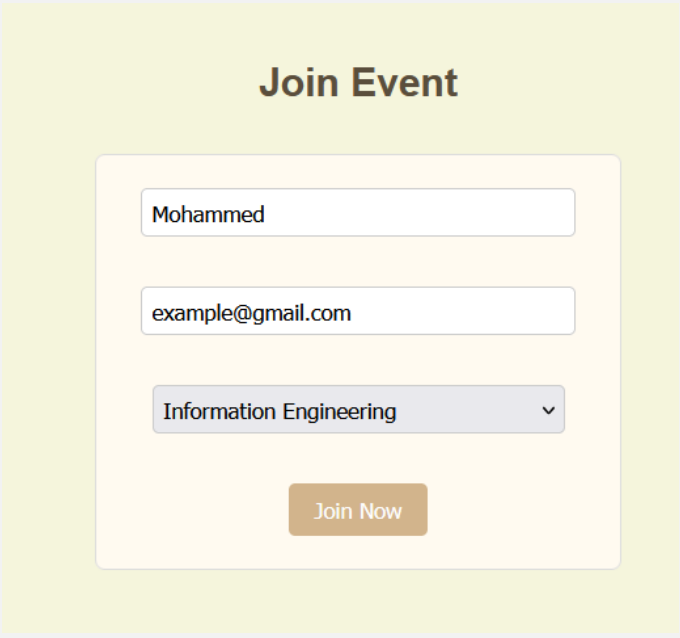
Secondly when I created this project I learned all the basics that I want to have a great foundation in Databases, and that is it.

Thanks :)

Application Sample

What users can do





What we will have

	student_id	name	email	department
Edit Copy Delete	12	Mohammed	example@gmail.com	Information Engineering

	registration_id	student_id	session_id	registration_date
Edit Copy Delete	8	12	2	2025-04-19

Check all *With selected:* Edit Copy Export

Mohammed