**Raef Hany, 900213194**

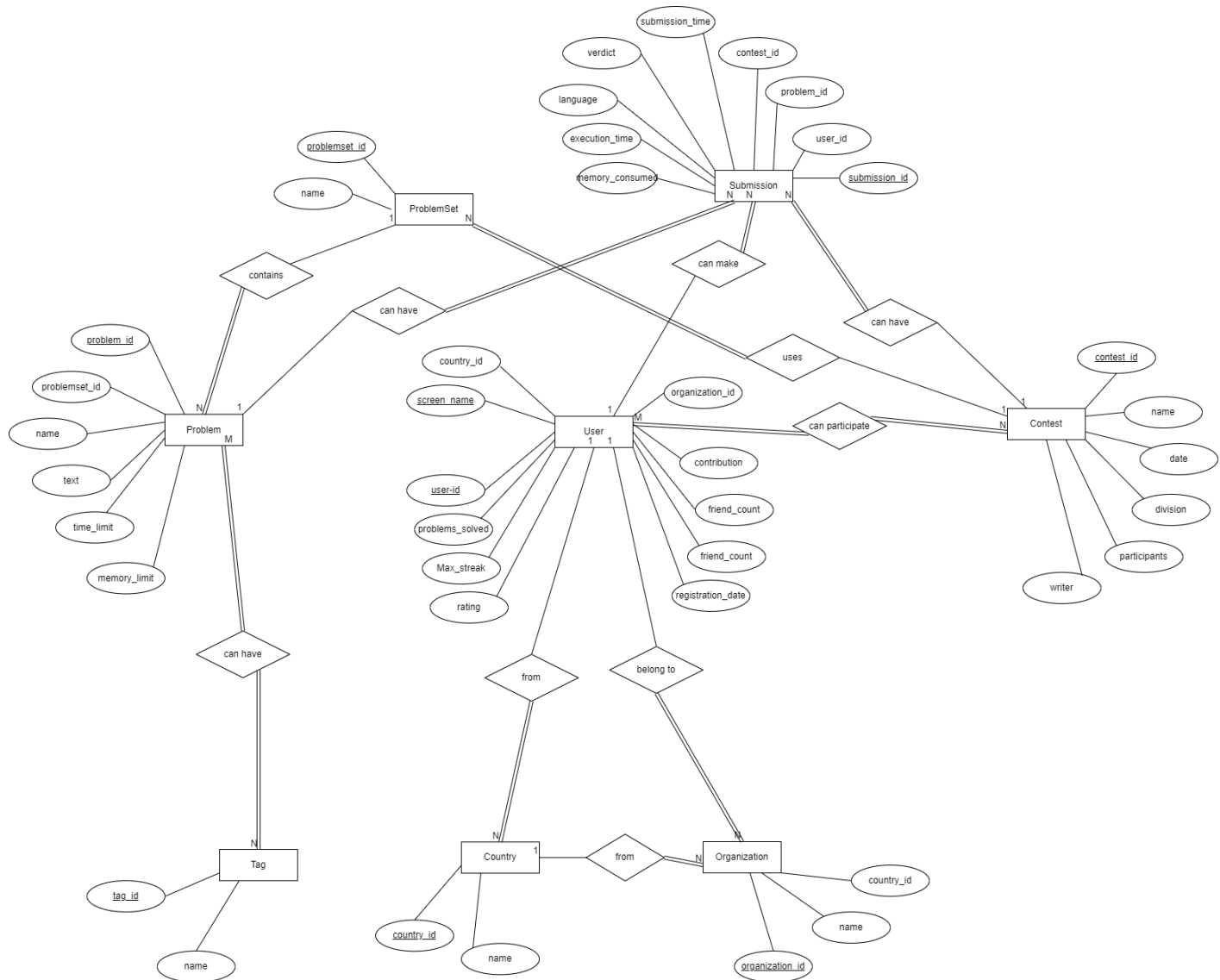**Project fall 24'**

**CSCE 2501**

**Dr. Sharara, Hossam**

**entity relationship diagram:**

**Tables:**

- **User:**
    - user_id (INT, PRIMARY KEY, AUTO_INCREMENT)
    - screen_name (VARCHAR(255), UNIQUE, NOT NULL)
    - city (VARCHAR(255))
    - country_id (INT, FOREIGN KEY referencing Country)
    - organization_id (INT, FOREIGN KEY referencing Organization)
    - contribution (INT)
    - friend_count (INT)
    - registration_date (DATETIME)
    - problems_solved (INT) *(Can be derived, but we can choose to store it)*
    - max_streak (INT) *(Can be derived)*
    - rating (INT)
- **Contest:**
    - contest_id (INT, PRIMARY KEY, AUTO_INCREMENT)
    - name (VARCHAR(255), NOT NULL)
    - date (DATETIME)
    - division (ENUM('Div. 1', 'Div. 2'))
    - participants (INT)
- **ProblemSet:**
    - problemset_id (INT, PRIMARY KEY, AUTO_INCREMENT)
- **Problem:**
    - problem_id (INT, PRIMARY KEY, AUTO_INCREMENT)
    - problemset_id (INT, FOREIGN KEY referencing ProblemSet)
    - name (VARCHAR(255)) *(e.g., "A", "B", "1000A")*
    - text (TEXT) *(For the bonus part)*
    - time_limit (INT) *(Milliseconds)*
    - memory_limit (INT) *(KB)*
- **Submission:**
    - submission_id (INT, PRIMARY KEY, AUTO_INCREMENT)
    - user_id (INT, FOREIGN KEY referencing User)
    - problem_id (INT, FOREIGN KEY referencing Problem)
    - contest_id (INT, FOREIGN KEY referencing Contest) *(Can be NULL)*
    - submission_time (DATETIME)
    - verdict (VARCHAR(255))
    - language (VARCHAR(255))
    - execution_time (INT) *(Milliseconds)*
    - memory_consumed (INT) *(KB)*
- **Organization:**
    - organization_id (INT, PRIMARY KEY, AUTO_INCREMENT)
    - name (VARCHAR(255))
    - country_id (INT, FOREIGN KEY referencing Country)
- **Country:**

- ○ country_id (INT, PRIMARY KEY, AUTO_INCREMENT)
  - ○ name (VARCHAR(255), UNIQUE)
- **Tag:**
  - ○ tag_id (INT, PRIMARY KEY, AUTO_INCREMENT)
  - ○ name (VARCHAR(255), UNIQUE)
- **UserContest (Junction Table):**
  - ○ user_contest_id (INT, PRIMARY KEY, AUTO_INCREMENT)
  - ○ user_id (INT, FOREIGN KEY referencing User)
  - ○ contest_id (INT, FOREIGN KEY referencing Contest)
- **ProblemTag (Junction Table):**
  - ○ problem_tag_id (INT, PRIMARY KEY, AUTO_INCREMENT)
  - ○ problem_id (INT, FOREIGN KEY referencing Problem)
  - ○ tag_id (INT, FOREIGN KEY referencing Tag)

**mySQL script:**

CREATE DATABASE IF NOT EXISTS codeforces;

USE codeforces;


CREATE TABLE Country (

   country_id INT PRIMARY KEY AUTO_INCREMENT,

   name VARCHAR(255) UNIQUE NOT NULL

);


CREATE TABLE Organization (

   organization_id INT PRIMARY KEY AUTO_INCREMENT,

   name VARCHAR(255),

   country_id INT,

   FOREIGN KEY (country_id) REFERENCES Country(country_id)

);

```sql
CREATE TABLE User (

    user_id INT PRIMARY KEY AUTO_INCREMENT,

    screen_name VARCHAR(255) UNIQUE NOT NULL,

    city VARCHAR(255),

    country_id INT,

    organization_id INT,

    contribution INT,

    friend_count INT,

    registration_date DATETIME,

    problems_solved INT,

    max_streak INT,

    rating INT,

    FOREIGN KEY (country_id) REFERENCES Country(country_id),

    FOREIGN KEY (organization_id) REFERENCES Organization(organization_id)

);


CREATE TABLE Contest (

    contest_id INT PRIMARY KEY AUTO_INCREMENT,

    name VARCHAR(255) NOT NULL,

    date DATETIME,
```

```sql
    division ENUM('Div. 1', 'Div. 2'),

    participants INT

);



CREATE TABLE ProblemSet (

    problemset_id INT PRIMARY KEY AUTO_INCREMENT

);



CREATE TABLE Problem (

    problem_id INT PRIMARY KEY AUTO_INCREMENT,

    problemset_id INT,

    name VARCHAR(255),

    text TEXT,

    time_limit INT,

    memory_limit INT,

    FOREIGN KEY (problemset_id) REFERENCES ProblemSet(problemset_id)

);



CREATE TABLE Tag (

    tag_id INT PRIMARY KEY AUTO_INCREMENT,
```

```sql
    name VARCHAR(255) UNIQUE NOT NULL

);


-- ProblemTag (Junction Table)

CREATE TABLE ProblemTag (

    problem_tag_id INT PRIMARY KEY AUTO_INCREMENT,

    problem_id INT,

    tag_id INT,

    FOREIGN KEY (problem_id) REFERENCES Problem(problem_id),

    FOREIGN KEY (tag_id) REFERENCES Tag(tag_id)

);




CREATE TABLE Submission (

    submission_id INT PRIMARY KEY AUTO_INCREMENT,

    user_id INT,

    problem_id INT,

    contest_id INT,  -- Can be NULL for practice submissions

    submission_time DATETIME,

    verdict VARCHAR(255),

    language VARCHAR(255),

    execution_time INT,

    memory_consumed INT,
```

```sql
    FOREIGN KEY (user_id) REFERENCES User(user_id),

    FOREIGN KEY (problem_id) REFERENCES Problem(problem_id),

    FOREIGN KEY (contest_id) REFERENCES Contest(contest_id)

);



-- UserContest (Junction Table)

CREATE TABLE UserContest (

    user_contest_id INT PRIMARY KEY AUTO_INCREMENT,

    user_id INT,

    contest_id INT,

    FOREIGN KEY (user_id) REFERENCES User(user_id),

    FOREIGN KEY (contest_id) REFERENCES Contest(contest_id)

);


CREATE TABLE ContestWriter (

    contest_writer_id INT PRIMARY KEY AUTO_INCREMENT,

    contest_id INT,

    user_id INT,

    FOREIGN KEY (contest_id) REFERENCES Contest(contest_id),

    FOREIGN KEY (user_id) REFERENCES User(user_id)

);
```