

# **CompeteNow: a Tournament Organization Platform**

## **Milestone: Implementation in MySQL Group 24**

**Charles Schatmeyer**  
**716-713-1492**  
**[schatmeyer.c@northeastern.edu](mailto:schatmeyer.c@northeastern.edu)**

**Percentage of Effort Contributed by Student1: 100%**

**Signature of Student 1:** 

**Submission Date: 10/23/2024**

Code:

```
DROP database if exists competition;
CREATE DATABASE IF NOT EXISTS competition;

USE competition;

CREATE TABLE IF NOT EXISTS user (
    userID int NOT NULL UNIQUE AUTO_INCREMENT PRIMARY KEY,
    name varchar(50) NOT NULL,
    location text,
    gender varchar(75) NOT NULL,
    email text,
    age int
);

CREATE TABLE IF NOT EXISTS team_manager (
    teamManagerID int NOT NULL UNIQUE,
    foreign key (teamManagerID) references user (userID) on update cascade on
delete cascade
);

CREATE TABLE IF NOT EXISTS player (
    playerID int NOT NULL UNIQUE,
    foreign key (playerID) references user (userID) on update cascade on delete
cascade
);

CREATE TABLE IF NOT EXISTS venue_manager (
    venueManagerID int NOT NULL UNIQUE,
    foreign key (venueManagerID) references user (userID) on update cascade on
delete cascade
);

CREATE TABLE IF NOT EXISTS spectator (
    spectatorID int NOT NULL UNIQUE,
    foreign key (spectatorID) references user (userID) on update cascade on
delete cascade
);

CREATE TABLE IF NOT EXISTS tournament_organizer (
    tournamentOrganizerID int NOT NULL UNIQUE,
    foreign key (tournamentOrganizerID) references user (userID) on update
cascade on delete cascade
);
```

```
CREATE TABLE IF NOT EXISTS venue (  
    venueID int NOT NULL UNIQUE AUTO_INCREMENT PRIMARY KEY,  
    size text,  
    name text,  
    hours text,  
    location text,  
    venueManager int NOT NULL,  
    foreign key (venueManager) references venue_manager (venueManagerID) on  
update cascade on delete cascade  
);  
  
CREATE TABLE IF NOT EXISTS sport (  
    sportID int NOT NULL UNIQUE AUTO_INCREMENT PRIMARY KEY,  
    name text  
);  
  
CREATE TABLE IF NOT EXISTS spectator_follows (  
    sport int NOT NULL,  
    spectator int NOT NULL,  
    foreign key (sport) references sport (sportID) on update cascade on delete  
cascade,  
    foreign key (spectator) references spectator (spectatorID) on update cascade  
on delete cascade  
);  
  
CREATE TABLE IF NOT EXISTS tournament (  
    tournamentID int NOT NULL UNIQUE AUTO_INCREMENT PRIMARY KEY,  
    prize text,  
    name text,  
    location text,  
    organizer int,  
    sport int,  
    foreign key (organizer) references tournament_organizer  
(tournamentOrganizerID) on update cascade,  
    foreign key (sport) references sport (sportID) on update cascade  
);  
  
CREATE TABLE IF NOT EXISTS team (  
    teamID int NOT NULL UNIQUE AUTO_INCREMENT PRIMARY KEY,  
    name text,  
    location text,  
    manager int NOT NULL,  
    foreign key (manager) references team_manager (teamManagerID) on update  
cascade on delete cascade  
);
```

```
CREATE TABLE IF NOT EXISTS team_player (
    teamID int NOT NULL,
    playerID int NOT NULL,
    foreign key (teamID) references team (teamID) on update cascade on delete
cascade,
    foreign key (playerID) references player (playerID) on update cascade on
delete cascade
);

CREATE TABLE IF NOT EXISTS match_gen (
    matchID int NOT NULL UNIQUE AUTO_INCREMENT PRIMARY KEY,
    time text,
    scoreAlpha int,
    scoreBeta int,
    venueID int,
    tournamentID int NOT NULL,
    foreign key (venueID) references venue (venueID) on update cascade,
    foreign key (tournamentID) references tournament (tournamentID) on update
cascade on delete cascade
);

CREATE TABLE IF NOT EXISTS teams_match (
    matchID int not null UNIQUE,
    teamAlpha int not null,
    teamBeta int not null,
    foreign key (matchID) references match_gen (matchID) on update cascade on
delete cascade,
    foreign key (teamAlpha) references team (teamID) on update cascade on delete
cascade,
    foreign key (teamBeta) references team (teamID) on update cascade on delete
cascade
);

CREATE TABLE IF NOT EXISTS individuals_match (
    matchID int not null UNIQUE,
    playerAlpha int not null,
    playerBeta int not null,
    foreign key (matchID) references match_gen (matchID) on update cascade on
delete cascade,
    foreign key (playerAlpha) references player (playerID) on update cascade on
delete cascade,
    foreign key (playerBeta) references player (playerID) on update cascade on
delete cascade
);
```