University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: February 18th, 2024

Group Number: 5

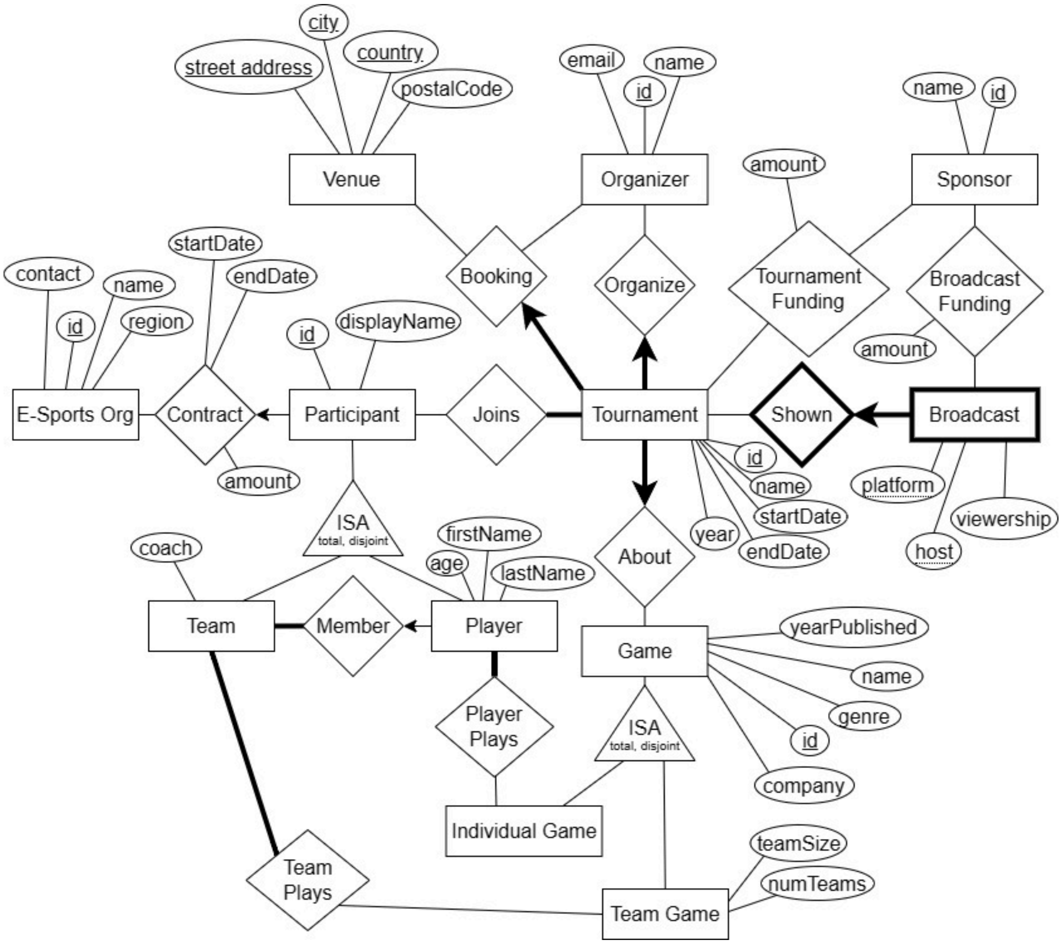
Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Bryan Jhutti	60442258	bjhutti	bryanjhutti@hotmail.com
Ronald Liu	10051118	rliu4936	ronaldliu2022@gmail.com
Victor Yao	26291401	vyao02	victoryao1039@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

2. Organization system for in-person e-sports tournaments. Users can track and manage tournaments, participants, sponsors, venues, etc.

3.



Many changes have been made to the ER diagram following feedback from Milestone 1:

- The Shown relationship set in the weak entity now has bolded borders
- ‘Name’ primary keys were changed to ‘id’ primary keys
- Attributes have been added to entities
- The Member relationship has not been changed because it makes sense to us; not only is it indicative of the real structure, but the Member relationship allows us to query Participants more granularly. The issue of Teams of single Players is not a concern for us, as it's not realistic and it doesn’t cause issues anyway. If we really wanted to avoid it for some reason, we could look into assertions in the future.
- The many-to-one relationship between Participant and Tournament has been changed to a many-to-many relationship to reflect reality
- Tournament now has total participation in Bookings, since tournaments must have venues (we will not be considering online tournaments for this project).
- The ISA relationships are meaningful now, through the TeamPlays and the PlayerPlays relationship. Our system can now document who plays what, with special distinction between individuals and teams.
- The partial key for Broadcast has been added along with all the attributes
- We removed the Broadcast Funding relationship between Sponsor and Broadcast, as we felt it was unnecessary

4. Broadcast(**tid: int**, platform: string, host: string, viewership: int)

- includes Shown relationship

ESportsOrg(id: int, name: string, region: string, contact: string)
- contact is also a candidate key

Game(id: int, name: string, genre: string, company: string, yearPublished: int)

IndividualGame(**id**: int)

TeamGame(**id**: int, teamSize: int, numTeams: int)

Organizer(id: int, name: string, email: string)
- email is also a candidate key

Participant(id: int, displayName: string)

Player(**id**: int, firstName: string, lastName: string, age: int)

Team(**id**: int, coach: string)

Sponsor(id: int, name: string)

Tournament(id: int, name: string, startDate: date, endDate: date, year: int, **streetAddress**: string, **city**: string, **country**: string, **organizerid**: int)
- Includes Booking and Organize relationships

Venue(streetAddress: string, city: string, country: string, postalCode: string)

Contract(**pid**: int, **orgid**: string, startDate: date, endDate: date, amount: int)

Joins(**pid**: int, **tid**: int)
- can’t model total participation in many-to-many relationship (need assertions)

Member(**playerid**: int, **teamid**: int)
- doesn’t model total participation on Team

TournamentFunding(**tid**: int, **sid**, amount: int)

5. Broadcast: tid → platform, host, viewership
host → platform
e.g. we know EVO tournaments will be on Twitch

ESportsOrg:	id → name, region, contact contact → id, name, region
Game:	id → name, genre, company, yearPublished
IndividualGame:	N/A
TeamGame:	id → teamSize, numTeams
Organizer:	id → name, email email → id, name
Participant:	id → displayName
Player:	id → firstName, lastName, age
Team:	id → coach
Sponsor:	id → name
Tournament:	id → name, startDate, endDate, year, streetAddress, city, country, organizerid startDate → year endDate → year
Venue:	streetAddress, city, country → postalCode postalCode → city, country
Contract:	pid, orgName → startDate, endDate, amount
Joins:	N/A
Member:	playerid → teamid
TournamentFunding:	tid, sid →amount

6. All tables are in BCNF except for Broadcast, Tournament, and Venue, so we will normalize them to BCNF.

   In Broadcast, we decompose on host → platform to get HostPlatform(host, platform) and Broadcast(**tid: int**, host: string, viewership: int), and those are now in BCNF.

For Tournament, we decompose on startDate → year to get StartDate(startDate: date, year: int) and Tournament(id: int, name: string, startDate: date, endDate: date, **streetAddress**: string, **city**: string, **country**: string, **organizerid**: int), and those are now in BCNF (but we lose the dependency endDate → year).

For Venue, we decompose on postalCode → city, country to arrive at PostalCode(postalCode: string, city: string, country: string) and Venue(streetAddress: string, postalCode: string), and those are now in BCNF.

Here are all of our tables after normalization:

Broadcast(**tid: int**, host: string, viewership: int)

HostPlatform(host, platform)

ESportsOrg(id: int, name: string, region: string, contact: string)

Game(id: int, name: string, genre: string, company: string, yearPublished: int)

IndividualGame(**id**: int)

TeamGame(**id**: int, teamSize: int, numTeams: int)

Organizer(id: int, name: string, email: string)

Participant(id: int, displayName: string)

Player(**id**: int, firstName: string, lastName: string, age: int)

Team(**id**: int, coach: string)

Sponsor(id: int, name: string)

Tournament(id: int, name: string, startDate: date, endDate: date, **streetAddress**: string, **city**: string, **country**: string, **organizerid**: int)

StartDate(startDate: date, year: int)

Venue(streetAddress: string, postalCode: string)

PostalCode(postalCode: string, city: string, country: string)

Contract(**pid**: int, **orgid**: string, startDate: date, endDate: date, amount: int)

Joins(**pid**: int, **tid**: int)

- can’t model total participation in many-to-many relationship (need assertions)

Member(**playerid**: int, **teamid**: int)

- teamid cannot be null

TournamentFunding(**tid**: int, **sid**, amount: int)

7.

```
-- Broadcast table
CREATE TABLE Broadcast (
   tid INT,
   host VARCHAR(20),
   viewership INT,
   PRIMARY KEY (tid, host)
   FOREIGN KEY (tid) REFERENCES Tournament(id)
);

-- HostPlatform table
CREATE TABLE HostPlatform (
   host VARCHAR(20) PRIMARY KEY,
   platform VARCHAR(20)
);

-- ESportsOrg table
CREATE TABLE ESportsOrg (
   id INT PRIMARY KEY,
   name VARCHAR(20),
   region VARCHAR(20),
   contact VARCHAR(20)
);

-- Game table
CREATE TABLE Game (
   id INT PRIMARY KEY,
   name VARCHAR(20),
   genre VARCHAR(20),
   company VARCHAR(20),
   yearPublished INT,
);

-- IndividualGame table
CREATE TABLE IndividualGame (
   id INT PRIMARY KEY,
   FOREIGN KEY (id) REFERENCES Game(id)
          ON DELETE CASCADE
);

-- TeamGame table
CREATE TABLE TeamGame (
   id INT PRIMARY KEY,
   teamSize INT,
   numTeams INT,
   FOREIGN KEY (id) REFERENCES Game(id)
          ON DELETE CASCADE
);

-- Organizer table
CREATE TABLE Organizer (
   id INT PRIMARY KEY,
   name VARCHAR(20),
   email VARCHAR(20)
);
-- Participant table
CREATE TABLE Participant (
   id INT PRIMARY KEY,
   displayName VARCHAR(20)
);

-- Player table
CREATE TABLE Player (
   id INT PRIMARY KEY,
   firstName VARCHAR(20),
   lastName VARCHAR(20),
   age INT,
   FOREIGN KEY (id) REFERENCES Participant(id)
          ON DELETE CASCADE
);

-- Team table
CREATE TABLE Team (
   id INT PRIMARY KEY,
   coach VARCHAR(20),
   FOREIGN KEY (id) REFERENCES Participant(id)
          ON DELETE CASCADE
);
```

```
-- Sponsor table
CREATE TABLE Sponsor (
   id INT PRIMARY KEY,
   name VARCHAR(20)
);

CREATE TABLE Tournament (
   id INT PRIMARY KEY,
   name VARCHAR(20),
   startDate DATE,
   endDate DATE,
   streetAddress VARCHAR(30),
   city VARCHAR(20),
   country VARCHAR(20),
   organizerid INT,
   FOREIGN KEY (streetAddress, city, country)
          REFERENCES Venue(streetAddress, city, country)
          ON UPDATE CASCADE,
    FOREIGN KEY (organizerid) REFERENCES Organizer(id)
);

-- StartDate table
CREATE TABLE StartDate (
   startDate DATE PRIMARY KEY,
   year INT
);

-- Venue table
CREATE TABLE Venue (
   streetAddress VARCHAR(30),
   postalCode CHAR(10),
   PRIMARY KEY (streetAddress, postalCode)
);

-- PostalCode table
CREATE TABLE PostalCode (
   postalCode CHAR(10) PRIMARY KEY,
   city VARCHAR(20),
   country VARCHAR(20)
);

-- Contract table
CREATE TABLE Contract (
   pid INT PRIMARY KEY,
   orgid INT,
   startDate DATE,
   endDate DATE,
   amount INT,
   FOREIGN KEY (pid) REFERENCES Participant(id)
          ON DELETE CASCADE,
   FOREIGN KEY (orgid) REFERENCES ESportsOrg(id)
          ON DELETE CASCADE
);

-- Joins table
CREATE TABLE Joins (
   pid INT,
   tid INT,
   PRIMARY KEY (pid, tid),
   FOREIGN KEY (pid) REFERENCES Participant(id),
          ON DELETE CASCADE,
   FOREIGN KEY (tid) REFERENCES Team(id)
          ON DELETE CASCADE
);

-- Member table
CREATE TABLE Member (
   playerid INT PRIMARY KEY,
   teamid INT NOT NULL,
   FOREIGN KEY (playerid) REFERENCES Player(id)
          ON DELETE CASCADE,
   FOREIGN KEY (teamid) REFERENCES Team(id)
          ON DELETE CASCADE
);

-- TournamentFunding table
CREATE TABLE TournamentFunding (
   tid INT,
   sid INT,
   amount INT,
   PRIMARY KEY (tid, sid),
   FOREIGN KEY (tid) REFERENCES Tournament(id)
          ON DELETE CASCADE,
   FOREIGN KEY (sid) REFERENCES Sponsor(id)
          ON DELETE CASCADE
);
```

8.

```
-- Broadcast table
INSERT INTO Broadcast (tid, host, viewership) VALUES
(1, 'Host1', 1000),
(2, 'Host2', 2000),
(3, 'Host3', 1500),
(4, 'Host4', 1800),
(5, 'Host5', 2200);

-- HostPlatform table
INSERT INTO HostPlatform (host, platform) VALUES
('Host1', 'Platform1'),
('Host2', 'Platform2'),
('Host3', 'Platform3'),
('Host4', 'Platform4'),
('Host5', 'Platform5');

-- ESportsOrg table
INSERT INTO ESportsOrg (id, name, region, contact) VALUES
(1, 'Org1', 'Region1', 'Contact1'),
(2, 'Org2', 'Region2', 'Contact2'),
(3, 'Org3', 'Region3', 'Contact3'),
(4, 'Org4', 'Region4', 'Contact4'),
(5, 'Org5', 'Region5', 'Contact5');

-- Game table
INSERT INTO Game (id, name, genre, company, yearPublished) VALUES
(1, 'Game1', 'Genre1', 'Company1', 2000),
(2, 'Game2', 'Genre2', 'Company2', 2005),
(3, 'Game3', 'Genre3', 'Company3', 2010),
(4, 'Game4', 'Genre4', 'Company4', 2015),
(5, 'Game5', 'Genre5', 'Company5', 2020);

-- IndividualGame table
INSERT INTO IndividualGame (id) VALUES
(1),
(2),
(3),
(4),
(5);

-- TeamGame table
INSERT INTO TeamGame (id, teamSize, numTeams) VALUES
(1, 5, 10),
(2, 6, 8),
(3, 4, 12),
(4, 3, 15),
(5, 5, 10);

-- Organizer table
INSERT INTO Organizer (id, name, email) VALUES
(1, 'Organizer1', 'organizer1@example.com'),
(2, 'Organizer2', 'organizer2@example.com'),
(3, 'Organizer3', 'organizer3@example.com'),
(4, 'Organizer4', 'organizer4@example.com'),
(5, 'Organizer5', 'organizer5@example.com');

-- Participant table
INSERT INTO Participant (id, displayName) VALUES
(1, 'Participant1'),
(2, 'Participant2'),
(3, 'Participant3'),
(4, 'Participant4'),
(5, 'Participant5');

-- Player table
INSERT INTO Player (id, firstName, lastName, age) VALUES
(1, 'First1', 'Last1', 25),
(2, 'First2', 'Last2', 28),
(3, 'First3', 'Last3', 22),
(4, 'First4', 'Last4', 30),
(5, 'First5', 'Last5', 27);

-- Team table
INSERT INTO Team (id, coach) VALUES
(1, 'Coach1'),
(2, 'Coach2'),
(3, 'Coach3'),
(4, 'Coach4'),
(5, 'Coach5');

-- Sponsor table
INSERT INTO Sponsor (id, name) VALUES
(1, 'Sponsor1'),
(2, 'Sponsor2'),
(3, 'Sponsor3'),
(4, 'Sponsor4'),
(5, 'Sponsor5');
```

```
-- Tournament table
INSERT INTO Tournament (id, name, startDate, endDate, streetAddress, city, country, organizerid) VALUES
(1, 'Tournament1', '2024-01-01', '2024-01-10', 'Street1', 'City1', 'Country1', 1),
(2, 'Tournament2', '2024-02-01', '2024-02-10', 'Street2', 'City2', 'Country2', 2),
(3, 'Tournament3', '2024-03-01', '2024-03-10', 'Street3', 'City3', 'Country3', 3),
(4, 'Tournament4', '2024-04-01', '2024-04-10', 'Street4', 'City4', 'Country4', 4),
(5, 'Tournament5', '2024-05-01', '2024-05-10', 'Street5', 'City5', 'Country5', 5);

-- StartDate table
INSERT INTO StartDate (startDate, year) VALUES
('2024-01-01', 2024),
('2024-02-01', 2024),
('2024-03-01', 2024),
('2024-04-01', 2024),
('2024-05-01', 2024);

-- Venue table
INSERT INTO Venue (streetAddress, postalCode) VALUES
('Street1', 'Postal1'),
('Street2', 'Postal2'),
('Street3', 'Postal3'),
('Street4', 'Postal4'),
('Street5', 'Postal5');

-- PostalCode table
INSERT INTO PostalCode (postalCode, city, country) VALUES
('Postal1', 'City1', 'Country1'),
('Postal2', 'City2', 'Country2'),
('Postal3', 'City3', 'Country3'),
('Postal4', 'City4', 'Country4'),
('Postal5', 'City5', 'Country5');

-- Contract table
INSERT INTO Contract (pid, orgid, startDate, endDate, amount) VALUES
(1, 1, '2024-01-01', '2024-01-10', 1000),
(2, 2, '2024-02-01', '2024-02-10', 2000),
(3, 3, '2024-03-01', '2024-03-10', 1500),
(4, 4, '2024-04-01', '2024-04-10', 1800),
(5, 5, '2024-05-01', '2024-05-10', 2200);

-- Joins table
INSERT INTO Joins (pid, tid) VALUES
```

```
(1, 1),
(2, 2),
(3, 3),
(4, 4),
(5, 5);

-- Member table
INSERT INTO Member (playerid, teamid) VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4),
(5, 5);

-- TournamentFunding table
INSERT INTO TournamentFunding (tid, sid, amount) VALUES
(1, 1, 10000),
(2, 2, 20000),
(3, 3, 15000),
(4, 4, 18000),
(5, 5, 22000);
```