CPSC 304 Project Proposal

Milestone #: 2
Group Number: 2
Date: 21 July 2024

Name	Student Number	CS Alias	E-mail Address
Jessie Zhong	50544220	g1l9y	zhongjessie2003@gmail.com
Ryan Quay	88497581	i3j0y	ryanquay@outlook.com
Mathew Balsdon	21041694	g0h3b	m.mathewbalsdon@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 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.

Contents

1 Project Summary

The domain of the application is the management of data related to tournaments that take place in the online rhythm game "osu!" (game-specific information can be found in Section 5 of the M1 Project Proposal). A significant aspect of the game's culture has to do with competition and seeing "who's the best" — while the osu! website and by extension the game's underlying systems provide an *overall* leaderboard and ranking system (based only on individual performance), no such thing exists specifically for those who compete in *tournaments*. The system aims to provide tools for the tracking, filtering, and ranking of tournament data such as players, matches, beatmaps, and other such auxiliary information.

2 Entity-Relationship Diagram

The full ER diagram can be found on the last page.

2.1 Changes

We made a few changes to our entity-relationship diagram from M1 in order to better represent our data. For the sake of clarity, the "name" attributes for Artist, Song, Player, and Tournament were changed to "artistName", "songName", "username", and "tournamentName" respectively. Substantial changes are listed below.

2.1.1 Features

Switched the Features relationship (between Score and Match) from many-to-many to many-to-one, since a Score may only ever feature in a single Match.

2.1.2 Beatmap

Added a new attribute to the Beatmap entity which links to its file. Also fixed the formatting to comply with conventions for weak entities.

- hitObjectsUrl: represents a link (from the osu! CDN) to the beatmap file.

2.1.3 Score

Added new attributes to the Score entity which effectively describe how well the player did on the beatmap.

- totalScore: represents the normalized value of the play based on the player's performance.
- combo: represents the highest number of hit-objects that were hit in a row by the player without missing.
- accuracy: represents an average of how on-time hit-objects were hit by the player (including misses).

2.1.4 Tournament

Replaced the "isBadged" attribute in the Tournament entity with more verbose badge data. Badges indicate that a tournament is officially recognized by the game and can be understood as analogous to a (real-life) trophy. Also added an ID since tournament names are not unique, and an attribute describing the start-time of the tournament.

- badgeUrl: represents a unique link (from the osu! CDN) to the badge's image.
- badgeIssueDate: represents the date on which the badge was officially administered by osu! staff.
- tournamentId: represents a unique ID to be used as a primary key.
- startDate: represents when the tournament began.

2.1.5 Standard

Removed the "circleDensity" attribute from the Standard entity since it is technically a 3rd-party quantifier not native to osu! beatmap files, and would thus make automatic population of the database much trickier.

2.1.6 Taiko

Removed the "drumLeniency" attribute from the Taiko entity since we falsely assumed that osu! beatmap files contain this information (it is generally a setting in other similar games).

3 Relational Schema

The following schema was derived from the ER diagram in Section 2. Conventions follow those from lecture.

3.1 Artist

Artist(artistId: INT, isFeatured: BOOL, name: CHAR(32))

- PRIMARY KEY: artistId
- NOT NULL: isFeatured, name

3.2 Song

Song(songId: INT, bpm: INT, genre: CHAR(32), name: CHAR(64), **artistId**: INT)

- PRIMARY KEY: songId
- FOREIGN KEYS: artistId
- NOT NULL: bpm, name, artistId

3.3 BeatmapSet

BeatmapSet(beatmapSetId: INT, creationDate: DATE, **playerId**: INT, **songId**: INT)

- PRIMARY KEY: beatmapSetId
- FOREIGN KEYS: playerId, songId
- NOT NULL: creationDate, playerId, songId

3.4 Beatmap

Beatmap(**beatmapSetId**: INT, difficultyName: CHAR(64), maxCombo: INT, hpDrain: INT, stars: FLOAT, hitObjectsUrl: CHAR(256), **playerId**: INT)

- PRIMARY KEY: (difficultyName, beatmapSetId)
- FOREIGN KEYS: beatmapSetId, playerId
- NOT NULL: maxCombo, hpDrain, stars, hitObjectsUrl, playerId

3.5 Player

Player(playerId: INT, rank: INT, username: CHAR(32), joinDate: DATE, **countryName**: CHAR(32))

- PRIMARY KEY: playerId
- FOREIGN KEYS: countryName
- NOT NULL: rank, username, joinDate, countryName
- UNIQUE: rank, username

3.6 Standard

Standard(**beatmapSetId**: INT, **difficultyName**: CHAR(64), circleSize: FLOAT)

- PRIMARY KEY: (difficultyName, beatmapSetId)
- FOREIGN KEYS: (difficultyName, beatmapSetId)
- NOT NULL: circleSize

3.7 Taiko

Taiko(**beatmapSetId**: INT, **difficultyName**: CHAR(64), drumSpeed: FLOAT)

- PRIMARY KEY: (difficultyName, beatmapSetId)
- FOREIGN KEYS: (difficultyName, beatmapSetId)
- NOT NULL: drumSpeed

3.8 Mania

Mania(**beatmapSetId**: INT, **difficultyName**: CHAR(64), keyCount: INT)

- PRIMARY KEY: (difficultyName, beatmapSetId)
- FOREIGN KEYS: (difficultyName, beatmapSetId)
- NOT NULL: keyCount

3.9 Score

Score(scoreId: INT, modifier: CHAR(8), dateSet: DATE, totalScore: INT, combo: INT, accuracy: FLOAT, **playerId**: INT, **beatmapSetId**: INT, **difficultyName**: CHAR(64), **matchId**: INT)

- PRIMARY KEY: scoreId
- FOREIGN KEYS: (difficultyName, beatmapSetId), playerId, matchId
- NOT NULL: dateSet, combo, accuracy, playerId, beatmapSetId, difficultyName, matchId

3.10 Match

Match(matchId: INT, round: CHAR(32), **tournamentId**: INT)

- PRIMARY KEY: matchId
- FOREIGN KEYS: tournamentId
- NOT NULL: round, tournamentId

3.11 Tournament

Tournament(tournamentId: INT, startDate: DATE, badgeUrl: CHAR(256), badgeIssueDate: DATE, name: CHAR(64), lowerRankBound: INT, upperRankBound: INT, **countryName**: CHAR(32))

- PRIMARY KEY: tournamentId
- FOREIGN KEYS: countryName
- NOT NULL: startDate, name

3.12 Country

Country(countryName: CHAR(32), flag: BLOB)

- PRIMARY KEY: countryName
- NOT NULL: flag

4 Functional Dependencies

We identified a number of functional dependencies from the model in Section 2 and Section 3.

4.1 Artist

- artistId → (isFeatured, name)

4.2 Song

- songId → (bpm, genre, name)

4.3 BeatmapSet

- beatmapSetId → (creationDate, **playerId**, **songId**)

4.4 Beatmap

- (**beatmapSetId**, difficultyName) → (maxCombo, hpDrain, stars, hitObjectsUrl, **playerId**)
- (hitObjectsUrl, hpDrain, maxCombo) → stars

4.5 Player

- playerId → (username, rank, joinDate, **countryName**)
- username → playerId
- rank → playerId

4.6 Standard

- (**beatmapSetId**, **difficultyName**) → circleSize

4.7 Taiko

- (**beatmapSetId**, **difficultyName**) → drumSpeed

4.8 Mania

- (**beatmapSetId**, **difficultyName**) → keyCount

4.9 Score

- scoreId → (modifier, dateSet, combo, accuracy, totalScore, **beatmapSetId**, **difficultyName**, **playerId**, **matchId**)

4.10 Match

- matchId → round, **tournamentId**

4.11 Tournament

- tournamentId → startDate, badgeUrl, badgeIssueDate, name, lowerRankBound, upperRankBound, **countryName**
- badgeUrl → badgeIssueDate

4.12 Country

- countryName → flag

5 Normalization

In order to cut down on redundancy, relations have been normalized to 3NF. Any relations that haven't been listed were identified as already being within 3NF.

5.1 Beatmap

```
Beatmap = B(BSID, DN, MC, HPD, S, HOU, PID)
Functional dependencies F =
    {
        (BSID, DN) -> (MC, HPD, S, HOU, PID),
        (HOU, HPD, MC) -> S
    }

Minimal cover F' =
    {
        (BSID, DN) -> MC,
        (BSID, DN) -> HPD,
        (BSID, DN) -> HOU,
        (BSID, DN) -> PID,
        (HOU, HPD, MC) -> S
    }

(BSID, DN) is a superkey for B.
(HOU, HPD, MC) is not a superkey for B, nor is S a (minimal) key.
Decomposing:
    B1(BSID, DN, MC, HPD, PID, HOU)
    B2(HOU, HPD, MC, S)

B1 and B2 are in BCNF. No dependencies were lost, therefore they are also in 3NF.
```

5.2 Tournament

```
Tournament T = T(TID, SD, BURL, BI, N, LRB, URB, CN)
Functional dependencies F =
    {
        TID -> (SD, BURL, BI, N, LRB, URB, CN)
        BURL -> BI
    }

Minimal cover F' =
    {
        TID -> SD,
        TID -> BURL,
        TID -> N,
        TID -> LRB,
        TID -> URB,
        TID -> CN,
        BURL -> BI
    }

TID is a superkey for T.
BURL is not a superkey for T, nor is BI a (minimal) key.
Decomposing:
    T1(TID, SD, BURL, N, LRB, URB, CN)
    T2(BURL, BI)

T1 and T2 are in BCNF. No dependencies were lost, therefore they are also in 3NF.
```

6 SQL DDL

The following contains SQL DDL statements to create tables based on the normalized relations in Section 5.

6.1 Artist

```
CREATE TABLE Artist (
    artistId INTEGER,
    isFeatured BOOL NOT NULL,
    name CHAR(32) NOT NULL,
    PRIMARY KEY (artistId)
);
```

6.2 Song

```
CREATE TABLE Song (
    songId INTEGER,
    bpm INTEGER NOT NULL,
    genre CHAR(32),
    name CHAR(32) NOT NULL,
    artistId INTEGER NOT NULL,
    PRIMARY KEY (songId),
    FOREIGN KEY (artistId) REFERENCES Artist(artistId)
);
```

6.3 BeatmapSet

```
CREATE TABLE BeatmapSet (
    beatmapSetId INTEGER,
    creationDate DATE NOT NULL,
    mapperId INTEGER NOT NULL,
    songId INTEGER NOT NULL,
    PRIMARY KEY (beatmapSetId),
    FOREIGN KEY (mapperId) REFERENCES Player(playerId),
    FOREIGN KEY (songId) REFERENCES Song(songId)
);
```

6.4 Beatmap

```
CREATE TABLE Beatmap (
    beatmapSetId INTEGER
    difficultyName CHAR(64),
    maxCombo INTEGER NOT NULL,
    hpDrain INTEGER NOT NULL,
    hitObjectsUrl CHAR(256) NOT NULL,
    mapperId INTEGER NOT NULL,
    PRIMARY KEY (beatmapSetId, difficultyName),
    FOREIGN KEY (beatmapSetId) REFERENCES BeatmapSet(beatmapSetId),
    FOREIGN KEY (mapperId) REFERENCES Player(playerId)
    FOREIGN KEY (hitObjectsUrl) REFERENCES BeatmapHitObjects(hitObjectUrl)
);
```

6.5 BeatmapHitObjects

```
CREATE TABLE BeatmapHitObjects (
    hitObjectsUrl CHAR(256),
    maxCombo INTEGER NOT NULL,
    hpDrain INTEGER NOT NULL,
```

```
    stars FLOAT NOT NULL,
    PRIMARY KEY (hitObjectsUrl)
);
```

6.6 Player

```
CREATE TABLE Player (
    playerId INTEGER,
    rank INTEGER NOT NULL,
    username CHAR(32) NOT NULL,
    joinDate DATE NOT NULL,
    countryName CHAR(32) NOT NULL,
    UNIQUE (username, rank),
    PRIMARY KEY (playerId),
    FOREIGN KEY (countryName) REFERENCES Country(countryName)
);
```

6.7 Standard

```
CREATE TABLE Standard (
    beatmapSetId INTEGER,
    difficultyName CHAR(64),
    circleSize FLOAT NOT NULL,
    PRIMARY KEY (beatmapSetId, difficultyName),
    FOREIGN KEY (beatmapSetId, difficulty Name) REFERENCES Beatmap(beatmapSetId, difficultyName)
);
```

6.8 Taiko

```
CREATE TABLE Taiko (
    beatmapSetId INTEGER,
    difficultyName CHAR(64),
    drumSpeed FLOAT NOT NULL,
    PRIMARY KEY (beatmapSetId, difficultyName),
    FOREIGN KEY (beatmapSetId, difficulty Name) REFERENCES Beatmap(beatmapSetId, difficultyName)
);
```

6.9 Mania

```
CREATE TABLE Mania (
    beatmapSetId INTEGER,
    difficultyName CHAR(64),
    keyCount INTEGER NOT NULL,
    PRIMARY KEY (beatmapSetId, difficultyName),
    FOREIGN KEY (beatmapSetId, difficulty Name) REFERENCES Beatmap(beatmapSetId, difficultyName)
);
```

6.10 Score

```
CREATE TABLE Score (
    scoreId INTEGER,
    modifier CHAR(8),
    totalScore INTEGER,
    combo INTEGER NOT NULL,
    accuracy FLOAT NOT NULL,
    dateSet DATE NOT NULL,
    playerId INTEGER NOT NULL,
    beatmapSetId INTEGER NOT NULL,
```

```
    difficultyName CHAR(64) NOT NULL,
    matchId INTEGER NOT NULL,
    PRIMARY KEY (scoreId),
    FOREIGN KEY (playerId) REFERENCES Player(playerId),
    FOREIGN KEY (beatmapSetId, difficultyName) REFERENCES Beatmap(beatmapSetId, difficultyName),
    FOREIGN KEY (matchId) REFERENCES Match(matchId)
);
```

6.11 Match

```
CREATE TABLE Match (
    matchId INTEGER,
    round CHAR(32) NOT NULL,
    tournamentId INT NOT NULL,
    PRIMARY KEY (matchId),
    FOREIGN KEY (tournamentId) REFERENCES Tournament(tournamentId)
);
```

6.12 Tournament

```
CREATE TABLE Tournament (
    tournamentId INTEGER,
    name CHAR(64) NOT NULL,
    lowerRankBound INTEGER,
    upperRankBound INTEGER,
    startDate DATE NOT NULL,
    badgeUrl CHAR(256),
    countryName CHAR(32),
    PRIMARY KEY (tournamentId),
    FOREIGN KEY (countryName) REFERENCES Country(countryName),
    FOREIGN KEY (badgeUrl) REFERENCES TournamentBadge(badgeUrl)
);
```

6.13 TournamentBadge

```
CREATE TABLE TournamentBadge (
    badgeUrl CHAR(256),
    badgeIssueDate DATE NOT NULL,
    PRIMARY KEY (badgeUrl)
);
```

6.14 Country

```
CREATE TABLE Country (
    countryName CHAR(32),
    flag BLOB NOT NULL,
    PRIMARY KEY (countryName)
);
```

7 Tentative Queries

The following contains SQL DDL statements to populate the normalized tables from Section 5 and Section 6.

Data obtained from:

- osu! World Cup 2023
- osu!taiko World Cup 2023
- osu!mania 7K World Cup 2023
- osu!Collegiate League: 10th Edition
- osu!UBC Sunset Series '22

NOTE: We are planning to automate the population of our database for this project.

7.1 Artist

```
INSERT INTO Artist VALUES(0, TRUE, "Kommisar");
INSERT INTO Artist VALUES(1, TRUE, "ZxNX");
INSERT INTO Artist VALUES(2, TRUE, "Aether");
INSERT INTO Artist VALUES(3, FALSE, "Denkishiki Karen Ongaku Shuudan");
INSERT INTO Artist VALUES(4, FALSE, "Noah");
```

7.2 Song

```
INSERT INTO Song VALUES(0, 245, "Video Game", "AKARI BEAM CANNON LAST BOSS", 0)
INSERT INTO Song VALUES(1, 222, "Electronic", "Fana", 1);
INSERT INTO Song VALUES(2, 240, "Rock", "Lunate Elf", 2);
INSERT INTO Song VALUES(3, 264, "Metal", "E.E.L.S.", 3);
INSERT INTO Song VALUES(4, 223, "Video Game", "Necrofantasia", 4);
```

7.3 BeatmapSet

```
INSERT INTO BeatmapSet(2095159, "2023-11-26", 10466315, 0);
INSERT INTO BeatmapSet(2095138, "2023-11-26", 5745865, 1);
INSERT INTO BeatmapSet(2095121, "2023-11-26", 4960893, 2);
INSERT INTO BeatmapSet(2095134, "2023-11-26", 7715620, 3);
INSERT INTO BeatmapSet(2095157, "2023-11-26", 5194391, 4);
```

7.4 Beatmap

```
INSERT INTO Beatmap(2095159, "TEMPORAL BLAST", 1820, 5,
    "https://osu.ppy.sh/beatmapsets/2095159/download", 10466315);
INSERT INTO Beatmap(2095138, "Annihilation", 1299, 6,
    "https://osu.ppy.sh/beatmapsets/2095138/download", 5745865);
INSERT INTO Beatmap(2095121, "Extra Stage", 1468, 4,
    "https://osu.ppy.sh/beatmapsets/2095121/download", 4960893);
INSERT INTO Beatmap(2095134, "Consumed", 1516, 4,
    "https://osu.ppy.sh/beatmapsets/2095134/download", 7715620);
INSERT INTO Beatmap(2095157, "Extra Stage", 1812, 5,
    "https://osu.ppy.sh/beatmapsets/2095157/download", 5194391);
```

7.5 BeatmapHitObjects

```
INSERT INTO BeatmapHitObjects("https://osu.ppy.sh/beatmapsets/2095159/download", 1820, 5, 8.05);
INSERT INTO BeatmapHitObjects("https://osu.ppy.sh/beatmapsets/2095138/download", 1299, 6, 7.93);
INSERT INTO BeatmapHitObjects("https://osu.ppy.sh/beatmapsets/2095121/download", 1468, 4, 7.60);
INSERT INTO BeatmapHitObjects("https://osu.ppy.sh/beatmapsets/2095134/download", 1516, 4, 7.73);
INSERT INTO BeatmapHitObjects("https://osu.ppy.sh/beatmapsets/2095157/download", 1812, 5, 7.61);
```

7.6 Player

```
INSERT INTO Player(7813296, 26, "hydrogen bomb", "2016-01-23", "United States");
INSERT INTO Player(7075211, 161, "tekkito", "2016-09-11", "United States");
INSERT INTO Player(4108547, 19, "WindowLife", "2014-03-06", "United States");
INSERT INTO Player(7562902, 2, "mrekk", "2015-12-12", "Australia");
INSERT INTO Player(7341183, 16, "ASecretBox", "2015-10-30", "Australia");
```

7.7 Standard

```
INSERT INTO Standard(2095159, "TEMPORAL BLAST", 3.8);
INSERT INTO Standard(2095138, "Annihilation", 4.2);
INSERT INTO Standard(2095121, "Extra Stage", 4.0);
INSERT INTO Standard(2095134, "Consumed", 4.0);
INSERT INTO Standard(2095157, "Extra Stage", 4.0);
```

7.8 Taiko

```
INSERT INTO Taiko(1980705, "Hell Oni", 10.0);
INSERT INTO Taiko(1720005, "X", 9.6);
INSERT INTO Taiko(1980775, "Super Macaroni", 5.0);
INSERT INTO Taiko(1664805, "Cosmic Cruise", 8.0);
INSERT INTO Taiko(1980692, "WereOni", 10.0);
```

7.9 Mania

```
INSERT INTO Mania(1939259, "[7K] Blazing Inferno", 7);
INSERT INTO Mania(1939327, "[7K] rickrollab", 7);
INSERT INTO Mania(1939270, "[7K] Event Horizon", 7);
INSERT INTO Mania(1657994, "[7K] Love You (Cut Ver.)", 7);
INSERT INTO Mania(1939330, "[7K] Schadenfreude", 7);
```

7.10 Score

```
INSERT INTO Score(1, "HR", 760905, 1518, 96.48, "2023-12-01", 7813296, 2095159,
    "TEMPORAL BLAST", 111534249);
INSERT INTO Score(2, "HD", 599548, 1145, 96.39, "2023-12-01", 7075211, 2095138,
    "Annihilation", 111534249);
INSERT INTO Score(3, NULL, 485149, 682, 98.13, "2023-12-01", 4108547, 2095121,
    "Extra Stage", 111534249);
INSERT INTO Score(4, "HD", 680844, 1163, 98.74, "2023-12-01", 7562902, 2095134,
    "Consumed", 111534249);
INSERT INTO Score(5, NULL, 532323, 1186, 95.78, "2023-12-01", 7341183, 2095157,
    "Extra Stage", 111534249);
```

7.11 Match

```
INSERT INTO Match(111534249, "GF (Winners)", 1);
INSERT INTO Match(108221558, "GF (Winners)", 2);
INSERT INTO Match(106911235, "GF (Winners)", 3);
INSERT INTO Match(111072404, "SF (Losers), 4);
INSERT INTO Match(103540083, "RO32", 5);
```

7.12 Tournament

```
INSERT INTO Tournament(1, "osu! World Cup 2023", NULL, NULL, "2023-09-21",
    "https://assets.ppy.sh/profile-badges/owc2023-winner.png", NULL);
INSERT INTO Tournament(2, "osu!taiko World Cup 2023", NULL, NULL, "2023-02-16",
    "https://assets.ppy.sh/profile-badges/otwc-2nd-2023.png", NULL);
```

```
INSERT INTO Tournament(3, "osu!mania 7K World Cup 2023", NULL, NULL, "2022-12-15",
    "https://assets.ppy.sh/profile-badges/mwc7k2023-winner.png", NULL);
INSERT INTO Tournament(4, "osu!Collegiate League: 10th Edition", NULL, NULL, "2023-08-31",
    "https://assets.ppy.sh/profile-badges/OCL10.png", NULL);
INSERT INTO Tournament(5, "osu!UBC Sunset Series '22", NULL, 100000, "2022-08-07",
    "https://assets.ppy.sh/profile-badges/oubc-2022.png", NULL);
```

7.13 TournamentBadge

```
INSERT INTO TournamentBadge("https://assets.ppy.sh/profile-badges/owc2023-winner.png",
    "2023-12-02");
INSERT INTO TournamentBadge("https://assets.ppy.sh/profile-badges/otwc-2nd-2023.png",
    "2023-04-30");
INSERT INTO TournamentBadge("https://assets.ppy.sh/profile-badges/mwc7k2023-winner.png",
    "2023-02-21");
INSERT INTO TournamentBadge("https://assets.ppy.sh/profile-badges/OCL10.png",
    "2024-01-15");
INSERT INTO TournamentBadge("https://assets.ppy.sh/profile-badges/oubc-2022.png",
    "2022-10-07");
```

7.14 Country

NOTE: Placeholder filenames have been used for flag BLOBs.

```
INSERT INTO Country("United States", LOAD_FILE("us.svg"));
INSERT INTO Country("Australia", LOAD_FILE("au.svg"));
INSERT INTO Country("Japan", LOAD_FILE("jp.svg"));
INSERT INTO Country("Taiwan", LOAD_FILE("tw.svg"));
INSERT INTO Country("Canada", LOAD_FILE("ca.svg"));
```

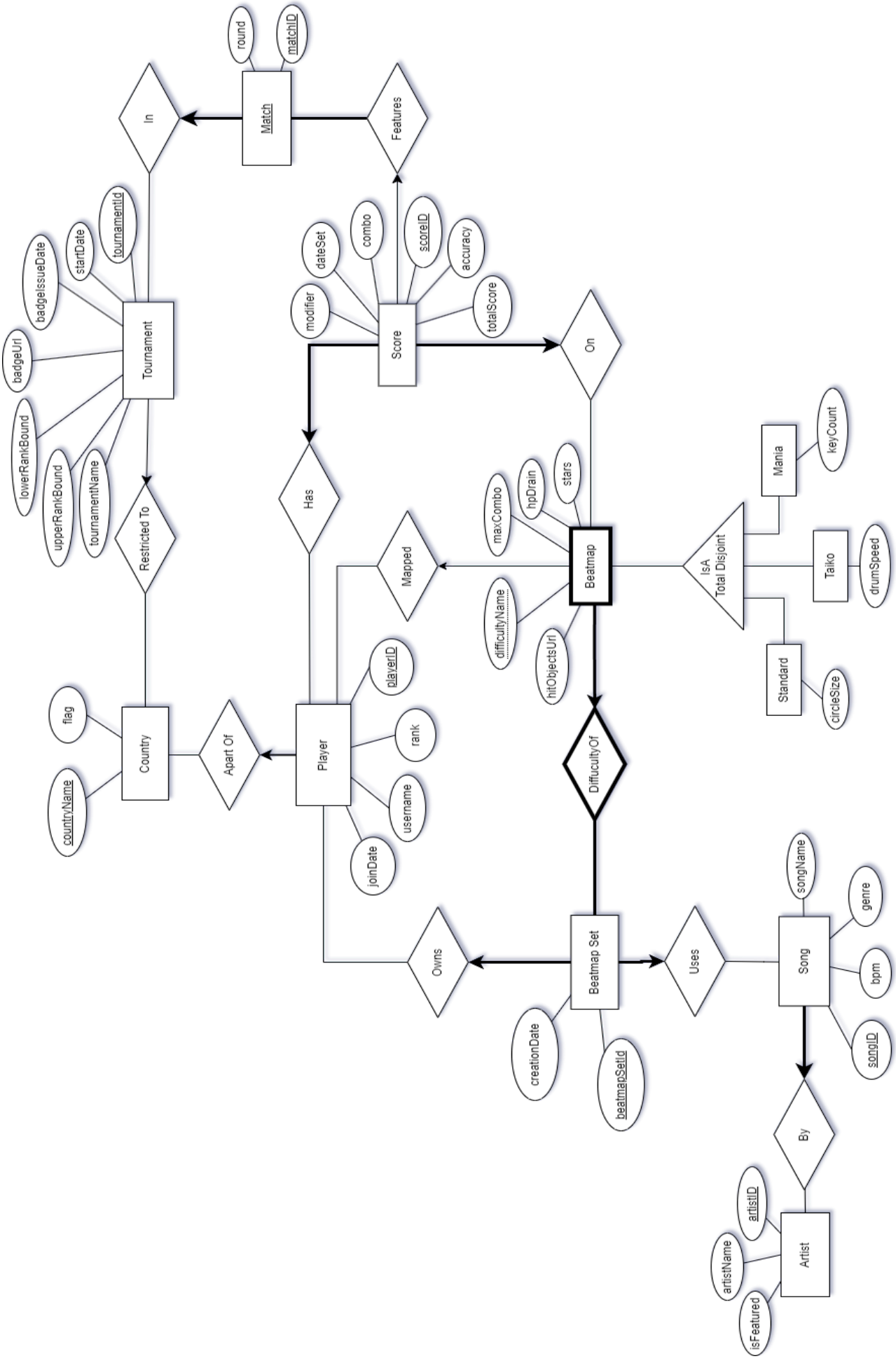

Figure 1: Entity-Relationship Diagram.