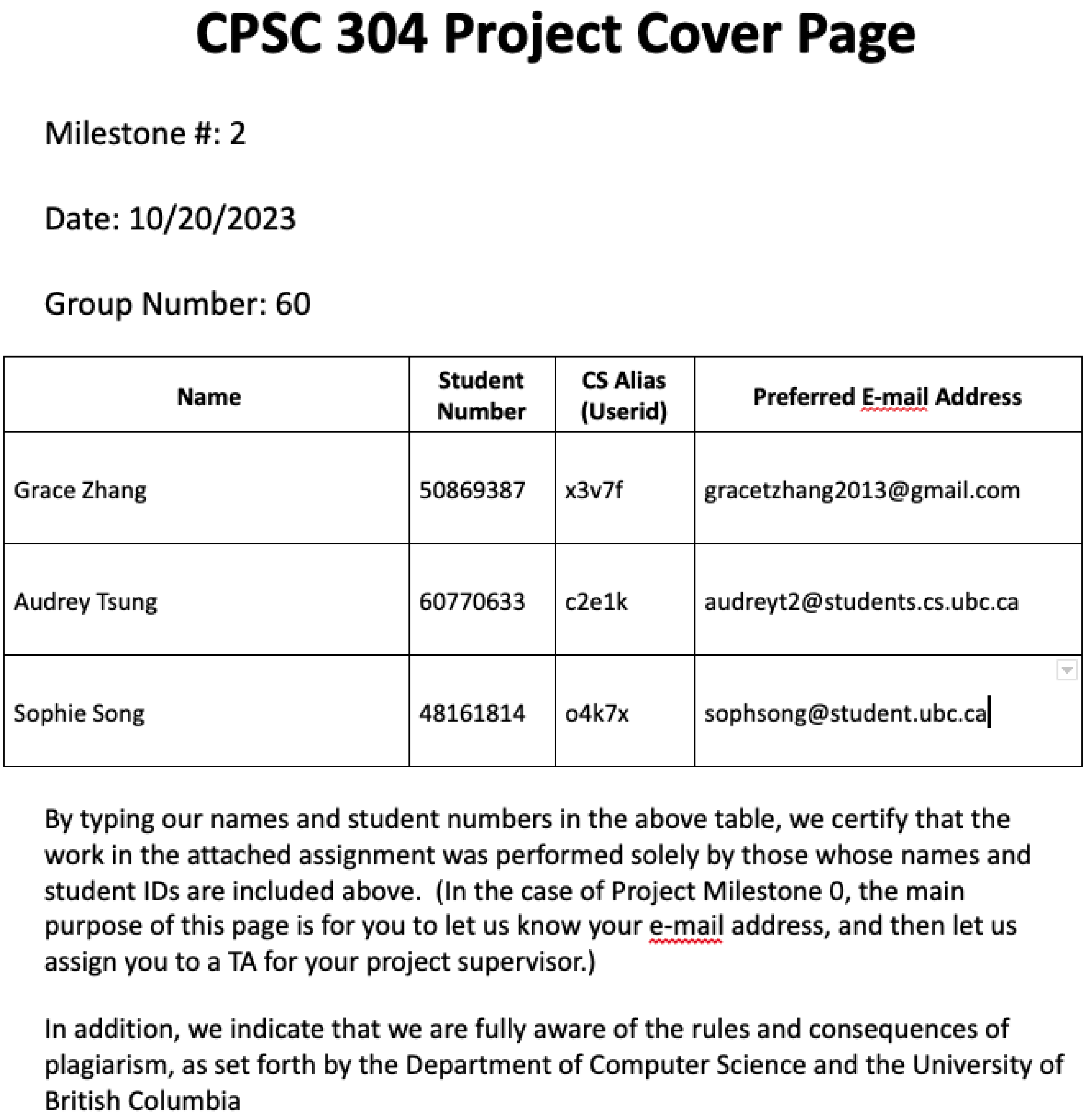
1. **Cover Page**



CPSC 304 Project Cover Page

Milestone #: 2

Date: 10/20/2023

Group Number: 60

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Grace Zhang	50869387	x3v7f	gracetzhang2013@gmail.com
Audrey Tsung	60770633	c2e1k	audreyt2@students.cs.ubc.ca
Sophie Song	48161814	o4k7x	sophsong@student.ubc.ca

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. Brief Summary

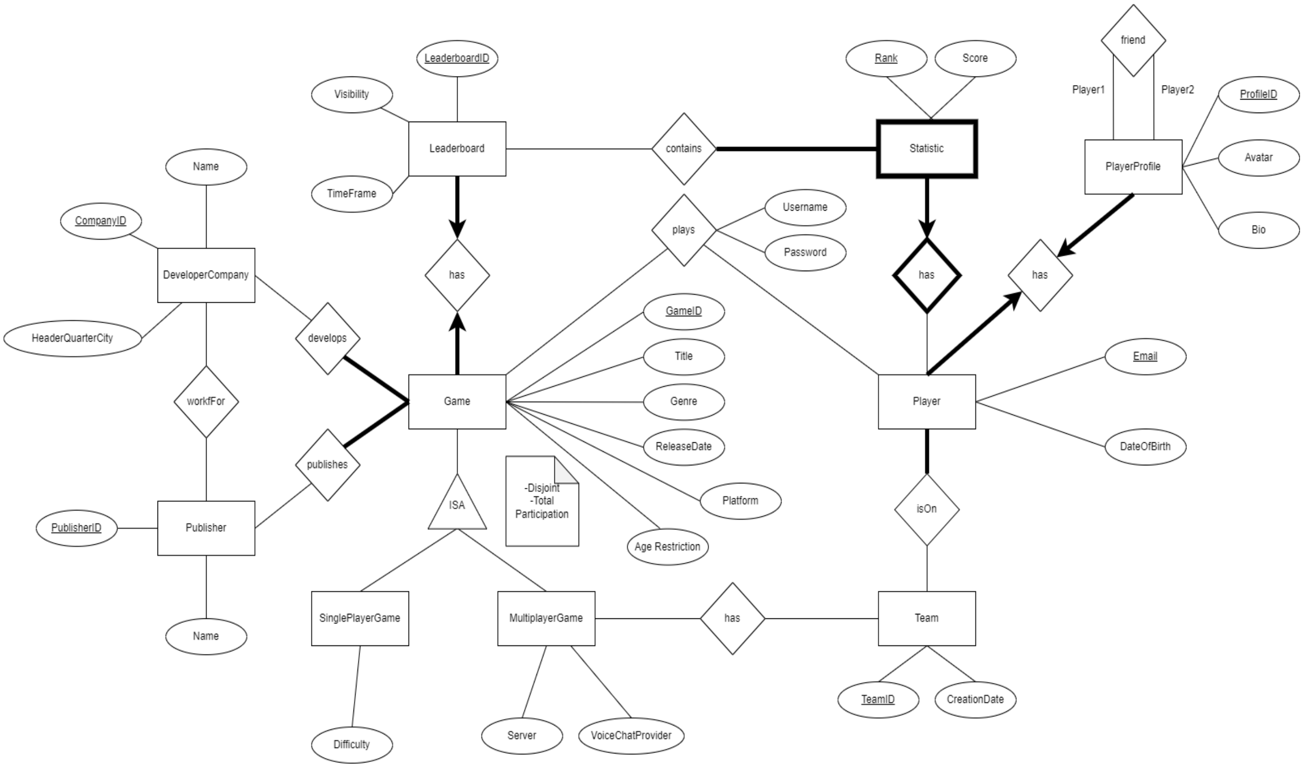
Our project involves designing a database to represent a system that manages games of various types, tracks player statistics, supports player profiles, and leaderboard tracking for each game. We are aiming to model how a company might want to handle and track data related to the games they provide and the players that play them. For example, game platforms such as Steam that host different amounts of games, and players can access it all in one place.

3. ER Diagram from before

We took the TA suggestions from the previous Milestone and made some changes involving our ER Diagram. Namely, we changed the primary key of Player to playerEmail as it is a more specific, and minimal way to describe a unique tuple in the Player table. As well, we previously did not have a partial key for our weak entity, so we identified that to be the “rank” attribute.

To make a more meaningful decomposition into 3NF, we added another FD to Game: ageRestriction. Now, we created a meaningful dependency as a genre determines what the ageRestriction of a game will be. For example, violent horror games will have an ageRestriction of 18+, while a farming RPG will have an ageRestriction of 0+.

Another modification we made to our ER diagram was a naming change, changing the attribute in DeveloperCompany from “headquarter” to “headerQuarterCity” instead. Thus, in the insertion it’s more clear to the user that they need to input the city, and not include other information that may describe the location where a headquarter may be. For example, headerQuarter may elicit insertions like “Calgary, AB, Canada” or “Vancouver, Canada”. Now, it is clear to the user that they should insert strings like “Calgary” or “Vancouver”.



4. Table Definitions

Leaderboard(leaderboardID: INT, visibility: BIT, timeFrame: VARCHAR, gameID: VARCHAR)

- PK: leaderboardID
- CK: N/A
- FK: gameID
- Not null: gameID
- Unique: gameID

DeveloperCompany(companyID: INT companyName: VARCHAR, headquarterCity: VARCHAR)

- PK: companyID
- CK: N/A
- FK: N/A
- Not null: name
- Unique: N/A

Publisher(publisherID: INT, publisherName: VARCHAR)

- PK: publisherID
- CK: N/A
- FK: N//A
- Not null: name
- Unique: N/A

Game(gameID: INT, title: VARCHAR, genre: VARCHAR, releaseDate: DATE, platform: VARCHAR, leaderboardID: INT, ageRestriction: VARCHAR)

- PK: gameID
- CK: N/A
- FK: leaderboardID
- Not null: title
- Unique: N/A

SinglePlayerGame(singlePlayerGameID: INT, difficulty: INT)

- PK: singlePlayerGameID
- CK: N/A
- FK: singlePlayerGameID
- Not null: N/A
- Unique: N/A

MultiplayerGame(multiplayerGameID: INT, server: VARCHAR, voiceChatProvider: VARCHAR)

- PK: multiplayerGameID
- CK: N/A
- FK: multiplayerGameID
- Not null: N/A
- Unique: N/A

Player(playerEmail: VARCHAR, dateOfBirth: DATE, profileID: INT)

- PK: playerEmail
- CK: N/A
- FK: profileID
- Not Null: DateOfBirth
- Unique: N/A

Statistic(playerEmail: VARCHAR, rank: INT, score: INT)

- PK: playerEmail, rank
- CK: N/A
- FK: playerEmail
- Not null: N/A
- Unique: rank

PlayerProfile(profileID: INT, avatar: IMG, bio: VARCHAR, playerEmail: VARCHAR)

- PK: profileID
- CK: N/A
- FK: playerEmail
- Not null: N/A
- Unique: N/A

Team(teamID: INT, creationDate: VARCHAR)

- PK: teamID
- CK: N/A
- FK: N/A

- Not null: creationDate
- Unique: N/A

workFor(companyID: INT, publisherID: INT)

- PK: companyID, publisherID
- CK: N/A
- FK: companyID, publisherID
- Not null: N/A
- Unique: N/A

develops(companyID: INT, gameID: INT)

- PK: gameID, companyID
- CK: N/A
- FK: companyID, gameID
- Not null: N/A
- Unique: N/A

publishes(publisherID: INT, gameID: INT)

- PK: publisherID, gameID
- CK: N/A
- FK: publisherID, gameID
- Not null: N/A
- Unique: N/A

contains(leaderboardID: INT, rank: INT, playerEmail: VARCHAR)

- PK: leaderboardID, rank, playerEmail
- CK: N/A
- FK: leaderboardID, rank, playerEmail
- Not null: N/A
- Unique: N/A

hasGAMETEAM(gameID: INT, teamID: INT)

- PK: gameID, teamID
- CK: N/A
- FK: gameID, teamID
- Not null: N/A
- Unique: N/A

isOn(playerEmail: VARCHAR, teamID: INT)

- PK: playerEmail, teamID
- CK: N/A
- FK: playerEmail, teamID
- Not null: N/A
- Unique: N/A

plays(username: VARCHAR, password: VARCHAR, gameID: INT, playerEmail: VARCHAR)

- PK: gameID, playerEmail
- CK: N/A
- FK: gameID, playerEmail
- Not null: username, password

- Unique: username

friend(profileID: VARCHAR, friendProfileID: VARCHAR)

- PK: profileID, friendProfileID
- CK: N/A
- FK: profileID, friendProfileID
- Not null: N/A
- Unique: N/A

5. **FD**

leaderboardID → visibility, timeFrame, gameID
companyID → companyName, headQuarterCity
publisherID → publisherName
playerEmail → dateOfBirth, profileID (for Player relation)
playerEmail → avatar, bio (for playerProfile relation)
playerEmail, rank → score
teamID → creationDate

gameID→ title, genre, releaseDate, platform, leaderboardID, ageRestriction
genre → ageRestriction

multiPlayerGameID → server, voiceChatProvider
server → voiceChatProvider

singlePlayerGameID → difficulty

6. Normalization into 3NF

Leaderboard(leaderboardID, visibility, timeFrame, gameID)
leaderboardID → visibility, timeFrame, gameID
The key for Leaderboard is leaderboardID such that leaderboardID$^+$ = {leaderboardID, visibility, timeFrame, gameID} and thus, satisfies 3NF.

DeveloperCompany(companyID, companyName, headQuarterCity)
companyID → companyName, headQuarterCity
The key for DeveloperCompany is companyID and companyID$^+$ = {companyID, companyName, headQuarterCity} which satisfies 3NF.

publishes(publisherID, gameID)
publisherID → publisherName
The trivial case where two attribute relations are by default in 3NF.

Player(playerEmail, dateOfBirth, profileID)
playerEmail → dateOfBirth, profileID
playerEmail is the key for player where playerEmail$^+$ = {playerEmail, dateOfBirth, profileID} so this relation is in 3NF.

PlayerProfile(profileID, avatar, bio, playerEmail)
playerEmail → avatar, bio

The key for PlayerProfile is profileID and playerEmail. So, because playerEmail is part of the key, PlayerProfile is in 3NF.

Statistic(playerEmail, rank, score)
playerEmail, rank → score
The primary key for Statistic is playerEmail and rank together such that (playerEmail, rank)$^+$ = {playerEmail, rank, score} which satisfies 3NF.

Team(teamID, creationDate)
teamID → creationDate
Team is a two attribute relation which is by default is in 3NF.

Game(gameID, title, genre, releaseDate, platform, difficulty, leaderboardID, ageRestriction)
gameID→ title, genre, releaseDate, platform, leaderboardID
genre → ageRestriction

Game violates in 3NF because genre is not part of the minimal key.

So first, we need to find the minimal cover:

1. gameID→ title
   gameID→ genre
   gameID→ releaseDate
   gameID→ platform
   gameID→ leaderboardID
   gameID → ageRestriction
   genre → ageRestriction
2. LHS doesn't need to be reduced because they each have one attribute
3. We can remove genre → ageRestriction because gameID already gives us ageRestriction.
   gameID→ title, genre, releaseDate, platform, leaderboardID

This minimal set of Game FDs no longer violates 3NF.

SinglePlayerGame(singlePlayerGameID: INT, difficulty: INT)
singlePlayerGameID → difficulty

SinglePlayerGame is in 3NF because singlePlayerGameID$^+$={singlePlayerGameID, difficulty}, so singlePlayerGameID is a key.

MultiplayerGame(multiplayerGameID: INT, server: VARCHAR, voiceChatProvider: VARCHAR)
multiPlayerGameID → server, voiceChatProvider
server → voiceChatProvider

MultiplayerGame is not in 3NF since server is not in the key. The key is multiPlayerGameID.

So, first we need to find minimal cover:

1. multiPlayerGameID → server
   multiPlayerGameID → voiceChatProvider
   server → voiceChatProvider
2. LHS are all one attribute, so we can't reduce them further.
3. We can remove server → voiceChatProvider since multiPlayerGameID gives us voiceChatProvider. So, the FDs left are:
   multiPlayerGameID → server, voiceChatProvider

The minimal cover doesn't violate 3NF, since multiPlayerGameID is the key.

7. SQL DDL statements

Entities

Notes: total participation, one-to-one relationship with game

```
CREATE TABLE Leaderboard(leaderboardID INT PRIMARY KEY,
                         visibility  BIT,
                         timeFrame VARCHAR,
                         gameID INT NOT NULL,
                         UNIQUE gameID,
                         FOREIGN KEY (gameID) REFERENCES Game(gameID))
```

```
CREATE TABLE DeveloperCompany(companyID INT PRIMARY KEY,
                              companyName VARCHAR,
                              headQuarterCity VARCHAR
```

```
CREATE TABLE Publisher(publisherID INT PRIMARY KEY,
                               publisherName VARCHAR)
```

Notes:

- total participation, one-to-one relationship with Leaderboard

```
CREATE TABLE Game(gameID INT PRIMARY KEY,
                  title VARCHAR NOT NULL,
                  genre VARCHAR,
                  ageRestriction VARCHAR,
                  releaseDate DATE,
                  platform VARCHAR,
                  leaderboardID INT NOT NULL,
                  UNIQUE leaderboardID,
                  FOREIGN KEY (leaderboardID) REFERENCES
                Leaderboard(leaderboardID) )


CREATE TABLE SinglePlayerGame(singlePlayerGameID INT PRIMARY KEY,
                              difficulty INT,
                              FOREIGN KEY (singlePlayerGameID) REFERENCES
                              Game(GameID))

CREATE TABLE MultiPlayerGame(multiPlayerGameID INT PRIMARY KEY,
                              server VARCHAR,
                              voiceChatProvider VARCHAR,
                              FOREIGN KEY (multiPlayerGameID) REFERENCES
                              Game(GameID))

CREATE TABLE Player(playerEmail VARCHAR PRIMARY KEY,
                  dateOfBirth DATE,
                  profileID INT,
                  rank INT,
                  UNIQUE profileID,
                  FOREIGN KEY (profileID) REFERENCES
                                   PlayerProfile(profileID)
                                   ON DELETE CASCADE
                 FOREIGN KEY(rank) REFERENCES
                                   Statistic(rank)
                                   ON DELETE CASCADE )
```

Notes: should delete statistic if player gets deleted

```
CREATE TABLE Statistic(playerEmail VARCHAR,
                       rank INT,
                       score INT,
                       PRIMARY KEY (playerEmail, rank)
                       FOREIGN KEY(playerEmail) REFERENCES
                                     Player(playerEmail)
                                     ON DELETE CASCADE)

CREATE TABLE PlayerProfile(profileID INT PRIMARY KEY,
                          friendProfileID INT,
                          avatar IMG,
                          bio VARCHAR,
                          playerEmail VARCHAR,
                          UNIQUE playerEmail,
                          FOREIGN KEY (playerEmail) REFERENCES
                                                      Player(playerEmail)
                                                      ON DELETE CASCADE,
                        FOREIGN KEY (friendProfileID) REFERENCES
                                                      PlayerProfile(profileID))

CREATE TABLE Team(teamID INT PRIMARY KEY,
                      creationDate DATE)
```

<u>Relationships</u>

```
CREATE TABLE workFor(companyID INT,
                         publisherID INT,
                         PRIMARY KEY(companyID, publisherID),
                         FOREIGN KEY (companyID) REFERENCES
                                                     DeveloperCompany(companyID)),
                        FOREIGN KEY (publisherID) REFERENCES
                                                     Publisher(publisherID))

CREATE TABLE develops(companyID INT,
                          gameID INT NOT NULL,
                          PRIMARY KEY (companyID, teamID),
                          FOREIGN KEY (gameID) REFERENCES Game(gameID)
                                                   ON DELETE CASCADE

                          FOREIGN KEY (companyID) REFERENCES
                     DeveloperCompany(companyID))

CREATE TABLE publishes(publisherID INT,
                          gameID INT NOT NULL,
                          PRIMARY KEY (publisherID, gameID),
                          FOREIGN KEY (publisherID) REFERENCES Publisher(publisherID),
                          FOREIGN KEY (gameID) REFERENCES Game(gameID))

CREATE TABLE contains(leaderboardID INT,
                         rank INT,
                         playerEmail INT),
                         PRIMARY KEY (leaderboardID, rank, playerEmail),
                         FOREIGN KEY (leaderboardID) REFERENCES
                         Leaderboard(leaderboardID),
                         FOREIGN KEY (rank) REFERENCES Statistic(rank),
                                    FOREIGN KEY (playerEmail) REFERENCES Player(playerEmail))

CREATE TABLE hasGameTeam(multiPlayerGameID INT,
                                   teamID INT),
                                  PRIMARY KEY (multiPlayerGameID, teamID),
                                  FOREIGN KEY (multiPlayerGameID) REFERENCES
                                  MultiPlayerGame(multiPlayerGameID),
                                  FOREIGN KEY (teamIDI) REFERENCES Team(teamID))

CREATE TABLE isOn(playerEmail INT,
                        teamID INT,
                        PRIMARY KEY (playerEmail, teamID),
                        FOREIGN KEY (playerEmail) REFERENCES
                                        Player(playerEmail),
                       FOREIGN KEY (teamID) REFERENCES
                                        Team(teamID))

CREATE TABLE plays(username VARCHAR NOT NULL,
                         password VARCHAR NOT NULL,
                         gameID INT,
                         playerEmail VARCHAR,
                         UNIQUE(username),
                         PRIMARY KEY(playerEmail, gameID),
                         FOREIGN KEY (playerEmail) REFERENCES
                                        Player(playerEmail),
                         FOREIGN KEY (teamID) REFERENCES
                                       Team(teamID))

CREATE TABLE friend(profileID INT,
                         friendProfileID INT,
                         PRIMARY KEY(profileID, friendProfileID),
                         FOREIGN KEY (profileID) REFERENCES
                                        PlayerProfile(profileID),
                         FOREIGN KEY (teamID) REFERENCES
                                       PlayerProfile(profileID)))
```

8. Insert into Tables

```
INSERT
INTO TABLE Leaderboard(leaderboardID, visibility, timeFrame, gameID)
Values (1111111111, TRUE, “2023-10-12 - 2023-10-19”, 1111111111)
        (2222222222, TRUE, “2023-10-12 - 2023-10-19”, 2222222222)
```

```
           (3333333333, FALSE, “2023-10-12 - 2023-10-19”, 3333333333)
           (4444444444, TRUE, “2023-10-12 - 2023-10-19”, 4444444444)
          (5555555555, FALSE, “2023-10-12 - 2023-10-19”, 5555555555)

INSERT
INTO TABLE DeveloperCompany(companyID, companyName, headquarter)
Values (1111111111, “Awesome Games”, “Calgary”)
           (2222222222, “Cool Games”,  “Vancouver”)
          (3333333333, “Simple Machine Games”,  “Montreal”)
          (4444444444, “Riot”,  “Los Angeles”)
           (5555555555, “Blizzard”,  “New York”)

INSERT
INTO TABLE Publisher(publisherID, publisherName)
Values (1111111111, “Ink Splatter Press“)
          (2222222222, “Game Publishers“)
          (3333333333, “Wonderful Whimsy“)
          (4444444444, “Reach for the Stars“)
          (5555555555, “Midnight Flourish“)

INSERT
INTO TABLE Game(gameID, title, genre, releaseDate, platform)
Values (1111111111, 'Maple Story', 'RPG', '2023-10-19', 'iOS'),
          (2222222222, 'League of Legends', 'MOBA', '2009-10-27', 'Microsoft Windows'),
          (3333333333, 'Valorant', 'Shooter', '2020-06-19', 'Microsoft Windows'),
          (4444444444, 'TeamFight Tactics', 'Auto Chess', '2019-06-26', 'Microsoft Windows'),
          (5555555555, 'Genshin Impact', 'Action RPG', '2020-09-28', 'Microsoft Windows”'),
          (6666666666, 'The Witcher 3: Wild Hunt', 'RPG', '2023-10-19', 'PC'),
          (7777777777, ‘Assassin's Creed Valhalla', 'Action-Adventure', '2023-10-19', 'Xbox'),
          (8888888888, ‘'The Legend of Zelda: Breath of the Wild', 'Action-Adventure', '2023-10-19',
          'Nintendo Switch'),
          (9999999999, Cyberpunk 2077', 'RPG', '2023-10-19', 'PC'),
          (1000000000, 'Red Dead Redemption 2', 'Action-Adventure', '2023-10-19', 'PlayStation');


INSERT
INTO TABLE SinglePlayerGame(singlePlayerGameID, difficulty)
Values (6666666666,  6),
          (7777777777,  8),
          (8888888888, 10),
          (9999999999, 9),
          (1000000000, 3)

INSERT
```

```
INTO TABLE MultiPlayerGame(multiPlayerGameID, server, voiceChatProvider)
Values
        (1111111111, ‘America-1001', ‘Voices’),
        (2222222222, ‘America-1002', 'VoicePro'),
        (3333333333,  ‘America-1003', 'TeamSpeak'),
        (4444444444, ‘America-1006’, ‘Noise’),
        (5555555555, ‘America-1005’,  'Mumble')

INSERT
INTO TABLE Statistic(playerEmail, rank, score)
Values (“grace@gmail.com”, 200, 49988)
        (“sophie@gmail.com”, 7, 60000)
        (“audrey@gmail.com”, 1, 8000878)
        (“jane@gmail.com”, 20, 4482748)
        (“sam@gmail.com”, 9000, 434)

INSERT
INTO TABLE Player(playerEmail, dateOfBirth, profileID)
Values
        ('grace@gmail.com', “2002-08-12”, 1),
        ( 'sophie@gmail.com, “2002-08-1”, 2),
        ('audrey@gmail.com', “2002-08-2”, 3),
        ('“jane@gmail.com', “2002-08-3”, 4),
        ('sam@gmail.com', “2002-08-4”, 5)

INSERT
INTO TABLE PlayerProfile(profileID, avatar, bio, playerEmail)
Values
        (1111111111, '/path/to/avatar1.jpg', 'Gamer, explorer, and thrill-seeker.', 'grace@gmail.com'),
        (2222222222, '/path/to/avatar2.jpg', 'Casual gamer', 'sophie@gmail.com'),
        (3333333333, '/path/to/avatar3.jpg', 'competitive.', 'audrey@gmail.com'),
        (4444444444, '/path/to/avatar4.jpg', 'live laugh love.', '“jane@gmail.com'),
        (5555555555, '/path/to/avatar5.jpg', 'i will win', 'sam@gmail.com');

INSERT
INTO TABLE Team(teamID: VARCHAR, creationDate: VARCHAR)
Values (1111111111, '2023-10-19'),
        (2222222222, '2023-09-15'),
        (3333333333, '2023-08-03'),
        (4444444444, '2023-07-22'),
        5555555555, '2023-06-01');

INSERT
INTO TABLE workFor(companyID, publisherID)
```

```
Values (1111111111, 1111111111)
       (1111111111, 2222222222)
       (2222222222, 1111111111)
       (2222222222, 2222222222)
       (2222222222, 3333333333)

INSERT
INTO TABLE develops(companyID: VARCHAR, gameID: VARCHAR)
Values (1111111111, 1111111111)
       (1111111111, 2222222222)
       (1111111111, 3333333333)
       (1111111111, 4444444444)
       (1111111111, 5555555555)
```

The same publisher publishes 5 different games.

```
INSERT
INTO TABLE publishes(publisherID: VARCHAR, gameID: VARCHAR)
Values (1111111111, 1111111111)
       (1111111111, 2222222222)
       (1111111111, 3333333333)
       (1111111111, 4444444444)
       (1111111111, 5555555555)

INSERT
INTO TABLE contains(leaderboardID, rank, playerEmail)
Values (1111111111, 200, “grace@gmail.com”)
        (2222222222, 7, “sophie@gmail.com”)
        (3333333333, 1, “audrey@gmail.com”)
        (4444444444, 20, “jane@gmail.com”)
       (5555555555, 9000, “sam@gmail.com”)

INSERT
INTO TABLE hasGAMETEAM(gameID, teamID)
Values (1111111111, 1111111111)
       (1111111111, 2222222222)
       (1111111111, 3333333333)
       (1111111111, 4444444444)
       (1111111111, 5555555555)

INSERT
INTO TABLE isOn(playerEmail,  teamID)
Values (“grace@gmail.com”, 1111111111)
       (“sophie@gmail.com”, 1111111111)
       (“audrey@gmail.com”, 1111111111)
```

```
        (“jane@gmail.com”, 2222222222)
        (“sam@gmail.com”, 2222222222)

INSERT
INTO TABLE plays(username, password, gameID, playerEmail)
Values (“aud”, “12345667”, 1111111111, “audrey@gmail.com”)
        (“zhanginc”, “1667”, 1111111111, “grace@gmail.com”)
        (“soneif”, “1278797779”, 1111111111, “sophie@gmail.com”)
        (“jan”, “hello1”, 1111111111, “jane@gmail.com”)
        (“sam2”, “goodbye3”, 1111111111, “sam@gmail.com”)

INSERT
INTO TABLE friend(profileID, friendProfileID)
Values (1111111111, 2222222222)
        (1111111111, 3333333333)
        (1111111111, 4444444444)
        (1111111111, 5555555555)
        (1111111111, 6666666666)
```