CPSC 304 Project Cover Page

Milestone #: 2

Date: 15/10/23

Group Number: 135

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Yahya Abouelmagd	54403621	x0p5w	yaya.almajd@gmail.com
Andrew Piemonte	90501727	a0z2r	andrewpiemonte@gmail.com
Jana Sheirah	75867630	j4c7k	jana.sheirah@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

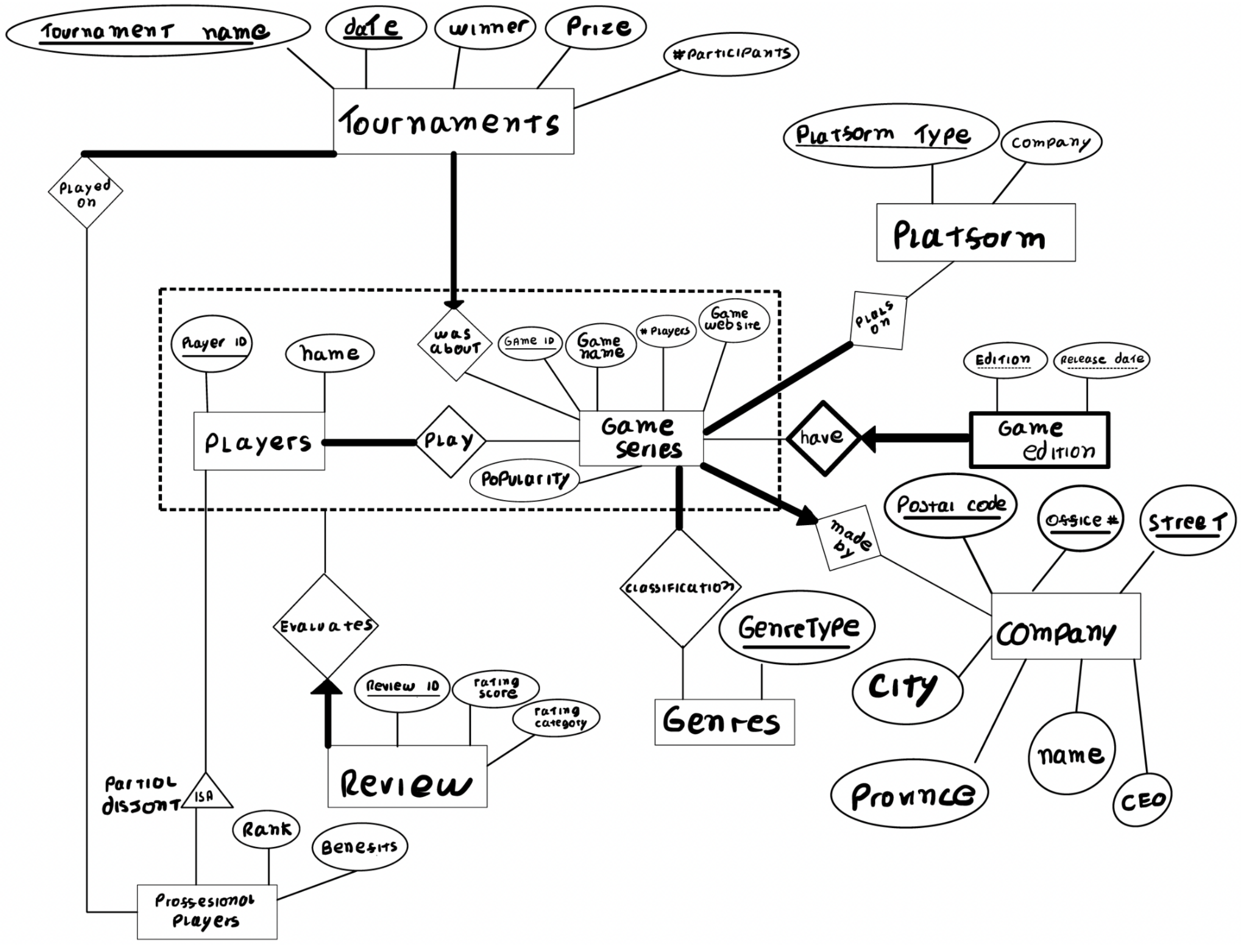
Project Description

This app is all about entertainment, focusing on video games and tournaments. It aims to include various gaming genres and tournaments within the industry. The main idea is to connect these elements through the players.

The database provides a wealth of gaming-related features for users. It allows you to easily access detailed information about a particular game, including its edition, reviews, compatible platforms, and genres. You can also discover the company behind the game and learn about both casual and professional players associated with it. Furthermore, the database offers comprehensive tournament details, such as the number of participants, the roster of players who competed, and the ultimate victor.

Section 1: ER Diagram

Figure 1. Project ER Diagram



Changes to ER Diagram from Milestone 1:

- Added some more meaningful attributes to show valid FDS
    - Modification to GameSeries:
        - Added Popularity,
        - Added #player,
        - Added GameWebsite.
        - Replaced Gross income was replaced for #players which makes it more meaningful for our focus.
    - Modification to Company
        - Address was broke into Postal Code, Office#, street, City, Province
    - Modification to Review:
        - AddedRating category, which review are placed in a category depending on Rating score, ex good and bad
- Some of the names were changed for more clarity (ex: ID replaced by Player ID)

Section 2: Schema

Schema based on Figure 1

Player(Player ID, Name)

Professional Player(**Player ID**, Rank, Benefits)

Platform(Type, Company)

Company(PostalCode, OfficeNum, Street, City, Province, Name, CEO)

Genres(Type)

PlayedOn(**TournamentName**, **TournamentDate**, **PlayerID**)

Classification(GenreType, **GameID**)

PlaysOn(**PlatformType**, **GameID**)

Play(**PlayerID**, **GameID**)

GameEditionHave(**GameID**, Edition, ReleaseDate)

GameSeriesMadeBy(GameID**, OfficeNum**, **Street**,**Postal Code**,GameName, PlayersNum , Popularity, GameWebsite)

ReviewEvaluates(Review ID, **Player ID**, **Game ID**, Rating Score, Rating Category)

TournamentWasAbout(TournamentName, TournamentDate, **GameID**, Winner, Prize ,ParticipantsNum)

Section 3: Functional Dependencies

Functional dependencies based on section 2 schema

- Player(Player ID, Name)
    - ID -> Name
    - CKs: {Player ID}
- Professional Player(**PlayerID**, Rank, Benefits)
    - PlayerID -> Rank, Benefits
    - Rank -> Benefits
    - CKs: {Player ID}
- Platform(Type, Company)
    - Type -> Company
    - CKs: {Type}
- Company(Postal Code, OfficeNum, Street, City, Province, CompanyName, CEO)
    - Postal Code, OfficeNum, Street -> CEO, CompanyName, City, Province
    - Postal Code -> City, Province
    - CKs:{ Postal Code, OfficeNum, Street}
- TournamentWasAbout(TournamentName, TournmentDate, **GameID**, Winner, Prize ,ParticipantsNum)
    - TournamentName, TournamentDate -> GameID, winner, prize, ParticipantsNum
    - CKs: {TournamentName, TournamentDate}
- GameEditionHave(**GameID**, Edition, ReleaseDate)
    - GameID -> Edition, ReleaseDate
- GameSeriesMadeBy(GameID**, OfficeNum, Street, Postal Code**, GameName, PlayersNum, Popularity, GameWebsite)
    - GameID -> OfficeNum, Street, Postal Code, GameName, PlayersNum, GameWebsite, Popularity
    - GameName, Postal Code, Street, OfficeNum -> GameID, PlayersNum, GameWebsite, Popularity
    - PlayersNum -> Popularity
    - GameWebsite -> GameName
    - CKs: {GameID}
- ReviewEvaluates(Review ID, **Player ID**, **Game ID**, Rating Score, Rating Category)
    - Review ID -> Player ID, Game ID, Rating Score, Rating Category

- Player ID, Game ID -> Review ID, Rating Score, Rating Category
- Rating Score -> Rating Category
- CKs: {Review ID}

Due to the nature of our database, many relations typically rely on a single candidate key for identification, mainly because the attributes themselves do not effectively serve as unique identifiers, as we explained in Milestone 1. Ultimately, that causes our relations to not have any other candidate keys since they are not minimal.

Furthermore, we've carefully considered all possible functional dependencies. Adding more functional dependencies could introduce constraints that we want to avoid. For example, a functional dependency like "Winner -> TournamentName, TournamentDate" would imply that a person cannot win two tournaments with the same name on the same day.

Section 4: Normalization

For functional Dependencies refer to Section 3.

ReviewEvaluates Relation to 3NF:

Let (Review ID, Player ID, Game ID, Rating Score, Rating Category) = (A,B,C,D,E)

1. **Minimum cover: A ->B A ->C B,C->A, B,C->D, D -> E**
2. **Decompose into 3NF: R1(ADCB) R2(DE)**
3. **All functional dependencies are represerved.**

Final Relations	FDs
R1(ReviewID, PlayerID, GameID, **RatingScore**)	ReviewID ->PlayerID,GameID PlayerID,GameID->ReviewID,RatingScore
R2(RatingScore, RatingCategory)	RatingScore -> RatingCategory

GameSeriesMadeBy Relation to 3NF:

Let (GameID, OfficeNum, Street, Postal Code, GameName, PlayersNum, Popularity, GameWebsite) = (A,B,C,D,E,F,G,H)

1. **Minimum cover: A ->B A ->C A->D A->E BDCE -> A**
   BDCE -> A, BDCE -> F BDCE -> H F->G H->E
2. **Decompose into 3NF: R1(ABCDEFH) and R2(FG)**
3. **Decompose into 3NF: R3(ABCDFH) and R4(HE)**

4. **All functional dependencies are represerved.**

Final Relations	**FDs**
R2(PlayersNum, Popularity)	PlayersNum -> Popularity
R3(GameID, OfficeNum, Street, Postal Code, **PlayersNum**, **GameWebsite**)	GameID -> OfficeNum, Street, Postal Code, PlayersNum, GameWebsite OfficeNum, Street, Postal Code, GameWebsite -> PlayersNum, GameID
R4(GameWebsite, GameName)	GameWebsite **->** GameName

Company Relation to 3NF:

Let Company(Postal Code, OfficeNum, Street, City, Province, CompanyName, CEO) = R(A,B,C,D,E)

- **Minimal Cover: ABC->F, ABC->G, A->D, A-> E**
- **-Decompose into 3NF: R1(A,D,E) and R2(A,B,C,F,G)**
- **All functional dependencies are preserved.**

R1(PostalCode, City, Province)	PostalCode -> City , Province
R2(**PostalCode**, OfficeNum, Street, Name, CEO)	PostalCode, OfficeNum, Street ->Name, CEO

Professional Player to 3NF:

Let ProfessionalPlayer(**PlayerID**, Rank, Benefits) = R(A,B,C)

So we have functional dependencies: A-> B, C and B->C (B->C violates BCNF since B is not a superkey)

- **Minimal Cover: A->B and B->C**
- **Decompose into 3NF: R1(B,C) and R2(A,B)**
- **All functional dependencies are represerved.**

Final Relations	**FDs**
R1(**PlayerID**, **Rank**)	PlayerID -> Rank
R2(Rank, Benefits)	Rank -> Benefits

Section 5: SQL DDL Statements

- It's important to mention that Oracle doesn't support "ON UPDATE CASCADE", so we've included comments in the relevant sections where it should be applied.
- Please take note of the following changes in our database design from Section 4: In our efforts to enhance clarity and maintain consistency, we've updated the names. For instance, "company R2" has been replaced with "company," and "company R1" is now "companyAddressInfo."

```
CREATE TABLE Player (
        PlayerID INTEGER PRIMARY KEY,
        Name VARCHAR(20)
);

CREATE TABLE ProfessionalPlayerRank (
        Rank VARCHAR(20) PRIMARY KEY,
        Benefits INTEGER
);
CREATE TABLE ProfessionalPlayer (
        PlayerID INTEGER PRIMARY KEY,
        Rank VARCHAR(20),
        FOREIGN KEY (PlayerID) REFERENCES Player(playerID),
        FOREIGN KEY (Rank) REFERENCES ProfessionalPlayerRank(Rank)
        ON DELETE CASCADE
);
// ON UPDATE CASCADE

CREATE TABLE Platform (
        Type VARCHAR(20) PRIMARY KEY,
        Company VARCHAR(20)
);

CREATE TABLE CompanyAddressInfo (
        PostalCode CHAR(6) PRIMARY KEY,
        City VARCHAR(30),
        Province VARCHAR(30)
);
CREATE TABLE Company (
        PostalCode CHAR(6),
        OfficeNum INTEGER,
        Street VARCHAR(20),
        Name VARCHAR(30),
    CEO VARCHAR(20),
    PRIMARY KEY(PostalCode, OfficeNum, Street),
    FOREIGN KEY (PostalCode) REFERENCES CompanyAddressInfo(PostalCode)
);


CREATE TABLE Genres (
    Type VARCHAR(20) PRIMARY KEY
);

CREATE TABLE GameSeriesPlayers (
    PlayersNum INTEGER PRIMARY KEY,
    Popularity VARCHAR(20)
);
CREATE TABLE GameSeriesWebsite (
    GameWebsite VARCHAR(20) PRIMARY KEY,
    GameName VARCHAR(20)
);
CREATE TABLE GameSeriesMadeBy (
    GameID INTEGER PRIMARY KEY,
    OfficeNum INTEGER NOT NULL,
    Street VARCHAR(20) NOT NULL,
    PostalCode CHAR(6) NOT NULL,
    PlayersNum INTEGER,
    GameWebsite VARCHAR(20),
    FOREIGN KEY (PlayersNum) REFERENCES GameSeriesPlayers(PlayersNum),
    FOREIGN KEY (GameWebsite) REFERENCES GameSeriesWebsite(GameWebsite),
    FOREIGN KEY (PostalCode, OfficeNum, Street) REFERENCES Company(PostalCode, OfficeNum,
    Street)
);
-- Note: in the example above for all company foreign key references
–  While we intend to use ON UPDATE CASCADE and ON DELETE NO ACTION,
-- please be aware that Oracle does not support ON UPDATE CASCADE.
-- Therefore, only ON DELETE NO ACTION will be enforced by Oracle as the default behavior.

CREATE TABLE TournamentWasAbout (
    TournamentName VARCHAR(20),
    TournamentDate DATE,
    GameID INTEGER NOT NULL,
    Winner VARCHAR(20),
    Prize VARCHAR(20),
    ParticipantsNum INTEGER,
    PRIMARY KEY(TournamentName, TournamentDate),
    FOREIGN KEY (GameID) REFERENCES GameSeriesMadeBy(GameID)
);
-- Note: in the example above
–  While we intend to use ON UPDATE CASCADE and ON DELETE NO ACTION,
-- please be aware that Oracle does not support ON UPDATE CASCADE.
-- Therefore, only ON DELETE NO ACTION will be enforced by Oracle as the default behavior.

CREATE TABLE PlayedOn (
    TournamentName VARCHAR(20),
    TournamentDate DATE,
    PlayerID INTEGER NOT NULL,
    PRIMARY KEY(TournamentName, TournamentDate, PlayerID),
    FOREIGN KEY (TournamentName, TournamentDate) REFERENCES
TournamentWasAbout(TournamentName, TournamentDate),
    FOREIGN KEY (PlayerID) REFERENCES Player(PlayerID)
);

CREATE TABLE Classification (
    GenreType VARCHAR(20),
    GameID INTEGER,
    PRIMARY KEY(GenreType, GameID),
    FOREIGN KEY (GenreType) REFERENCES Genres(Type),
    FOREIGN KEY (GameID) REFERENCES GameSeriesMadeBy(GameID)
);

CREATE TABLE PlaysOn (
    PlatformType VARCHAR(20),
    GameID INTEGER,
    PRIMARY KEY(PlatformType, GameID),
    FOREIGN KEY (PlatformType) REFERENCES Platform(Type),
    FOREIGN KEY (GameID) REFERENCES GameSeriesMadeBy(GameID)
);

CREATE TABLE Play (
    PlayerID INTEGER,
    GameID INTEGER,
    PRIMARY KEY(PlayerID, GameID),
    FOREIGN KEY(PlayerID) REFERENCES Player(PlayerID),
    FOREIGN KEY (GameID) REFERENCES GameSeriesMadeBy(GameID)
);

CREATE TABLE GameEditionHave (
    GameID INTEGER,
    Edition VARCHAR(20),
    ReleaseDate DATE,
    PRIMARY KEY(GameID, Edition, ReleaseDate),
    FOREIGN KEY (GameID) REFERENCES GameSeriesMadeBy(GameID) ON DELETE CASCADE
);

CREATE TABLE ReviewEvaluatesRating (
    RatingScore INTEGER PRIMARY KEY,
    RatingCateogry VARCHAR(20)
);
CREATE TABLE ReviewEvaluates (
    ReviewID INTEGER PRIMARY KEY,
    PlayerID INTEGER NOT NULL,
    GameID INTEGER NOT NULL,
    RatingScore INTEGER,
    FOREIGN KEY (PlayerID) REFERENCES Player(PlayerID),
    FOREIGN KEY (GameID) REFERENCES  GameSeriesMadeBy(GameID),
    FOREIGN KEY (RatingScore) REFERENCES ReviewEvaluatesRating(RatingScore)
);
```

Section 5: INSERT Statements

```
INSERT INTO Player(playerID, name) VALUES (1, 'Yahya');
INSERT INTO Player(playerID, name) VALUES (2, 'Jana');
INSERT INTO Player(playerID, name) VALUES (3, 'Andrew');
INSERT INTO Player(playerID, name) VALUES (4, 'Jessica');
INSERT INTO Player(playerID, name) VALUES (5, 'Sam');


INSERT INTO ProfessionalPlayerRank(Rank, Benefits) VALUES (‘Bronze’,3);
INSERT INTO ProfessionalPlayerRank(Rank, Benefits) VALUES (‘Silver’, 3);
INSERT INTO ProfessionalPlayerRank(Rank, Benefits) VALUES (‘Gold’, 2);
INSERT INTO ProfessionalPlayerRank(Rank, Benefits) VALUES (‘Platinum’, 2);
INSERT INTO ProfessionalPlayerRank(Rank, Benefits) VALUES (‘Diamond’, 1);

INSERT INTO ProfessionalPlayer(PlayerID, Rank) VALUES (1, ‘Bronze’);
INSERT INTO ProfessionalPlayer(PlayerID, Rank) VALUES (2, ‘Silver’);
INSERT INTO ProfessionalPlayer(PlayerID, Rank) VALUES (3, ‘Gold’);
INSERT INTO ProfessionalPlayer(PlayerID, Rank) VALUES (4, ‘Platinum’);
INSERT INTO ProfessionalPlayer(PlayerID, Rank) VALUES (5, ‘Diamond’);


INSERT INTO Platform(Type, Company) VALUES (‘XBOX’, ‘Microsoft’);
INSERT INTO Platform(Type, Company) VALUES (‘Playstation 4’, ‘Sony’);
INSERT INTO Platform(Type, Company) VALUES (‘M18 Gaming Laptop’, ‘Alienware’);
INSERT INTO Platform(Type, Company) VALUES ('G16 Gaming Laptop’, ‘DELL’);
INSERT INTO Platform(Type, Company) VALUES ('Victus Gaming Laptop’, ‘HP’);


INSERT INTO Company (PostalCode, OfficeNum, Street, Name, CEO)
VALUES (‘K1A0A1’, 111, ‘Main Street’, 'ABC Inc', 'John Doe');
INSERT INTO Company (PostalCode, OfficeNum, Street, Name, CEO)
VALUES (‘M5V2H1’, 234, ‘Ross Street', 'Six Guys LLC', ‘Ahmed Bin Sulaiman');
INSERT INTO Company (PostalCode, OfficeNum, Street, Name, CEO)
VALUES (‘H3B4G7’, 244, 'Jordan Street', 'Jordan Inc', 'Jordan Jordan');
```

```
INSERT INTO Company (PostalCode, OfficeNum, Street, Name, CEO)
VALUES (‘V6B4M9’, 455, 'Wall Street', ‘New York’, ‘Jordan Belfort’);
INSERT INTO Company (PostalCode, OfficeNum, Street, Name, CEO)
VALUES (‘R3C0K6’, 678, 'Main Street', 'DEF Inc', 'Dohn Joe');

INSERT INTO CompanyAddressInfo ( PostalCode, City, Province)
VALUES (‘K1A0A1’, 'Ottawa’, 'Ontario’);
INSERT INTO CompanyAddressInfo ( PostalCode, City, Province)
VALUES (‘M5V2H1’, ‘Toronto’, ‘Ontario’);
INSERT INTO CompanyAddressInfo ( PostalCode, City, Province)
VALUES (‘H3B4G7’, ‘Montreal’, ‘Quebec’);
INSERT INTO CompanyAddressInfo ( PostalCode, City, Province)
VALUES (‘V6B4M9’, ‘Vancouver’, ‘British Columbia’);
INSERT INTO CompanyAddressInfo( PostalCode, City, Province)
VALUES (‘R3C0K6’, Winnipeg’, ‘Manitoba’);


INSERT INTO Genres(Type) VALUES (‘RPG’);
INSERT INTO Genres(Type) VALUES (‘MMO’);
INSERT INTO Genres(Type) VALUES (‘ACTION’);
INSERT INTO Genres(Type) VALUES (‘ROMANCE’);
INSERT INTO Genres(Type) VALUES (‘TERROR’);


INSERT INTO GameSeriesPlayers(PlayersNum, Popularity) VALUES (10, 'Not Popular');
INSERT INTO GameSeriesPlayers(PlayersNum, Popularity) VALUES (50, 'Not Popular');
INSERT INTO GameSeriesPlayers(PlayersNum, Popularity) VALUES (200, 'Moderate
Popular');
INSERT INTO GameSeriesPlayers(PlayersNum, Popularity) VALUES (1000, 'Popular');
INSERT INTO GameSeriesPlayers(PlayersNum, Popularity) VALUES (10000, 'Overwhelmingly
Popular');


INSERT INTO GameSeriesWebsite(GameWebsite, GameName) VALUES (‘Activision’, 'Call
of Duty');
INSERT INTO GameSeriesWebsite(GameWebsite, GameName) VALUES (‘EASports.com’,
'FIFA');
INSERT INTO GameSeriesWebsite(GameWebsite, GameName) VALUES (‘Steam.com’, 'FIFA');
INSERT INTO GameSeriesWebsite(GameWebsite, GameName) VALUES
(‘PlaystationStore.com’, 'Batman');
INSERT INTO GameSeriesWebsite(GameWebsite, GameName) VALUES (‘Nintendo.com’,
'Mario Kart');

INSERT INTO GameSeriesMadeBy (GameID, OfficeNum, Street, PostalCode, PlayersNum, GameWebsite)
```

```
VALUES (12118088, 111, 'Main Street', 'K1A0A1', 10, 'Steam.com');
INSERT INTO GameSeriesMadeBy (GameID, OfficeNum, Street, PostalCode, PlayersNum, GameWebsite)
VALUES (12446708, 234, 'Ross Street', 'M5V2H1', 50, 'EASports.com');
INSERT INTO GameSeriesMadeBy (GameID, OfficeNum, Street, PostalCode, PlayersNum, GameWebsite)
VALUES (1255546, 244, 'Jordan Street', 'H3B4G7', 1000, 'Nintendo.com');
INSERT INTO GameSeriesMadeBy (GameID, OfficeNum, Street, PostalCode, PlayersNum, GameWebsite)
VALUES (26419909, 455, 'Wall Street', 'V6B4M9', 1000, 'EASports.com');
INSERT INTO GameSeriesMadeBy (GameID, OfficeNum, Street, PostalCode, PlayersNum, GameWebsite)
VALUES (7887676, 678, 'Main Street', 'R3C0K6', 200, 'Activision');


INSERT INTO TournamentWasAbout (TournamentName, TournamentDate, GameID, Winner, Prize,
ParticipantsNum)
VALUES ('Tournament A', '2023-10-20', 12118088, 'Player X', '$10,000',
100);
INSERT INTO TournamentWasAbout (TournamentName, TournamentDate, GameID, Winner, Prize,
ParticipantsNum)
VALUES ('Tournament B', '2023-10-21', 12446708, 'Player Y', '$9,000', 75);
INSERT INTO TournamentWasAbout (TournamentName, TournamentDate, GameID, Winner, Prize,
ParticipantsNum) VALUES ('Tournament C', '2023-10-22', 1255546, 'Player Z',
'$7,500', 50);
INSERT INTO TournamentWasAbout (TournamentName, TournamentDate, GameID, Winner, Prize,
ParticipantsNum) VALUES ('Tournament D', '2023-10-23', 26419909, 'Player W',
'$2,500', 40);
INSERT INTO TournamentWasAbout (TournamentName, TournamentDate, GameID, Winner, Prize,
ParticipantsNum) VALUES ('Tournament E', '2023-10-24', 7887676, 'Player V',
'$0', 30);

INSERT INTO PlayedOn (TournamentName, TournamentDate, PlayerID) VALUES ('Tournament A',
'2023-10-20', 1);
INSERT INTO PlayedOn (TournamentName, TournamentDate, PlayerID) VALUES ('Tournament A',
'2023-10-20', 2);
INSERT INTO PlayedOn (TournamentName, TournamentDate, PlayerID) VALUES ('Tournament A',
'2023-10-20', 3);
INSERT INTO PlayedOn (TournamentName, TournamentDate, PlayerID) VALUES ('Tournament A',
'2023-10-20', 4);
INSERT INTO PlayedOn (TournamentName, TournamentDate, PlayerID) VALUES ('Tournament A',
'2023-10-20', 5);


INSERT INTO Classification (GenreType, GameID) VALUES ('Action',12118088);
INSERT INTO Classification (GenreType, GameID) VALUES (‘Romance’, 12446708 );
INSERT INTO Classification (GenreType, GameID) VALUES ('Action', 1255546);
INSERT INTO Classification (GenreType, GameID) VALUES (‘RPG’, 26419909);
INSERT INTO Classification (GenreType, GameID) VALUES ('Action', 7887676);

INSERT INTO PlaysOn (PlatformType, GameID) VALUES ('‘XBOX’', 12118088);
INSERT INTO PlaysOn (PlatformType, GameID) VALUES ('Playstation 4', 12446708);
INSERT INTO PlaysOn (PlatformType, GameID) VALUES (‘M18 Gaming Laptop’, 1255546);
INSERT INTO PlaysOn (PlatformType, GameID) VALUES (‘G16 Gaming Laptop’, 26419909);
INSERT INTO PlaysOn (PlatformType, GameID) VALUES (‘Victus Gaming Laptop’, 7887676);

INSERT INTO Play (PlayerID, GameID) VALUES (1, 12118088);
INSERT INTO Play (PlayerID, GameID) VALUES (2, 12118088);
INSERT INTO Play (PlayerID, GameID) VALUES (3, 12446708);
INSERT INTO Play (PlayerID, GameID) VALUES (4, 26419909);
INSERT INTO Play (PlayerID, GameID) VALUES (5, 7887676);



INSERT INTO GameEditionHave (GameID, Edition, ReleaseDate) VALUES (12118088, 'Black
OPS III', '2016-10-25');
INSERT INTO GameEditionHave (GameID, Edition, ReleaseDate) VALUES (12446708, 'Black
OPS IIII', '2020-11-25');
INSERT INTO GameEditionHave (GameID, Edition, ReleaseDate) VALUES (1255546, 'Black
OPS II', '2009-05-25');
INSERT INTO GameEditionHave (GameID, Edition, ReleaseDate) VALUES (26419909, 'Black
OPS I', '2016-12-25');
INSERT INTO GameEditionHave (GameID, Edition, ReleaseDate) VALUES (7887676, 'Modern
Warfare', '2008-9-25');

INSERT INTO ReviewEvaluatesRating (RatingScore, RatingCateogry) VALUES (1,
'Positive');
INSERT INTO ReviewEvaluatesRating (RatingScore, RatingCateogry) VALUES (2,
'Positive');
INSERT INTO ReviewEvaluatesRating (RatingScore, RatingCateogry) VALUES (3,
'Positive');
INSERT INTO ReviewEvaluatesRating (RatingScore, RatingCateogry) VALUES (4,
‘Negative');
INSERT INTO ReviewEvaluatesRating (RatingScore, RatingCateogry) VALUES (5,
‘Negative');

INSERT INTO ReviewEvaluates (ReviewID, PlayerID, GameID, RatingScore)
VALUES (1, 1, 12118088, 1);
```

```
INSERT INTO ReviewEvaluates (ReviewID, PlayerID, GameID, RatingScore)
VALUES (2, 2, 12446708, 2);
INSERT INTO ReviewEvaluates (ReviewID, PlayerID, GameID, RatingScore)
VALUES (3, 3, 12446708, 3);
INSERT INTO ReviewEvaluates (ReviewID, PlayerID, GameID, RatingScore)
VALUES (4, 4, 1255546, 4);
INSERT INTO ReviewEvaluates (ReviewID, PlayerID, GameID, RatingScore)
VALUES (5, 5, 26419909, 5);
```