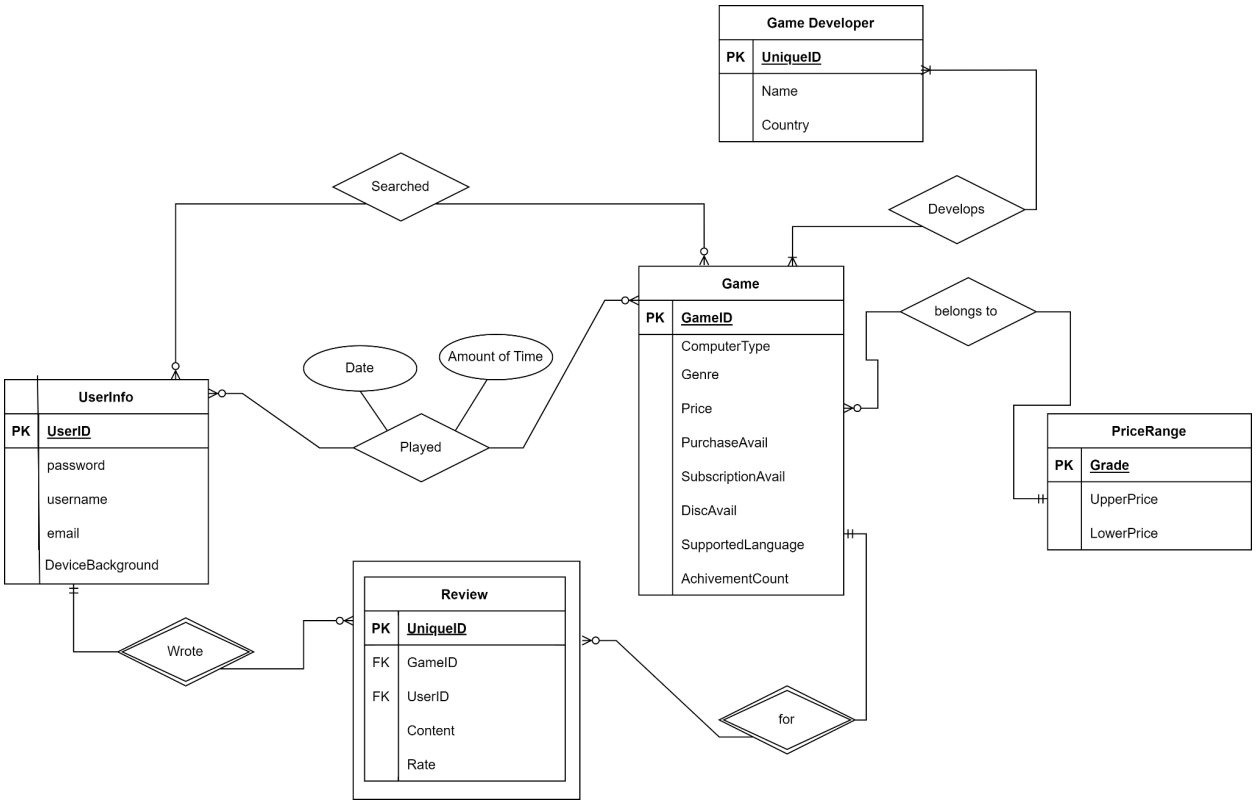
ER/UML Diagram



Entities:

UserInfo: **UserID**, Email, UserName, Password, DeviceBackground

Games: **GameID**, Computer Type, Game Genre, Price, Supported language, Achievement Count, PurchaseAvail, SubscriptionAvail, DiscAvail

Review: **UniqueID**, GameID, UserID, Content, Rating

Game Developer: **UniqueID**, Name, Country

Price Range: **Grade**, UpperPrice, LowerPrice

Relationship Assumptions & Descriptions

1. Writes (UserInfo & Reviews) 1 to Many

Users write reviews. Users can write multiple reviews and reviews must be written by 1 user for it to exist, so it is 1 to many.

2. Reviews for (Games & Reviews) 1 to Many
   Reviews are a part of game descriptions so they are for a Game
   Games can have multiple reviews
3. Developed by (Games & Game Developers) Many to Many
   Games can be developed by 1 or more developers
4. Played (UserInfo & Games) Many to Many
   A user can play 0 to many games.
   One game can be played by 0 to many users
   A table that stores user's playtime and date (could be null)
5. Searches (UserInfo & Games) Many to Many
   A user can search 0 to many games.
   One game can be searched by 0 to many users
   Save the searched attributes in a table

P.S. The purpose of searched relation is to allow users to search for which games have previously been searched for by them.Serves as a "search history" function.

6. In Price Range(Game & Price Range) one to Many
   A game is in exactly one price range.
   A price range can have multiple games.

P.S. the idea of price range is that we will divide all prices into x range (x being an integer being determined) and the information (lower price and upper price) are stored in the price range table

7. Game has reviews (Game & Review) one to many
   A game can have zero to many reviews.
   A review belongs to exactly one game.

Relational Schema

UserInfo(
    UserID: VARCHAR(20) [PK],
    Email: VARCHAR(50),
    Username: VARCHAR(20),
    Password: VARCHAR(40),
    DeviceBackground: VARCHAR(40)
)
Games(
    GameID: VARCHAR(20) [PK],
    Computer Type: VARCHAR(50),
    Game Genre: VARCHAR(20),
    Price: FLOAT,
    Support Language: VARCHAR(20),

Achievement Count: INT
PurchaseAvail: BIT DEFAULT 0,
SubscriptionAvail: BIT DEFAULT 0,
DiscAvail: INT
)

Review(
UniqueID:VARCHAR(20) [PK],
GameID:VARCHAR(20) [FK to Game.GameID],
UserID:VARCHAR(20) [FK to User.UserID],
Content:VARCHAR(1000),
Rate: INT
)

GameDeveloper(
UniqueID: VARCHAR(20) [PK],
Name: VARCHAR(40),
Country: VARCHAR(20)
)

PriceRange(
Grade:VARCHAR(20) [PK],
UpperPrice: INT,
LowerPrice: INT
)

Played(
UserID:VARCHAR(20) [PK, FK to User.UserID],
GameID:VARCHAR(20) [PK, FK to Game.GameID],
Date: DATE,
TimePlayed: INT
)

Searched(
UserID:VARCHAR(20) [PK,FK to User.UserID],
GameID:VARCHAR(20) [PK, FK to Game.GameID]
)

Develops(
DeveloperID: VARCHAR(20) [PK, FK to GameDeveloper.UniqueID],
GameID:VARCHAR(20) [PK, FK to Game.GameID]
)

Answer Previous Stage Questions

Some additional questions: Do you think you have enough data for 5 entities, and at least 3 tables with at least 1,000 rows? Is visualization the main way you will distinguish yourself from existing sites, because many existing sites allow you to search using multiple conditions? Will Reviews be the only CRUD component of your application?

We believe that we will have enough data for 5 entities, and at least 3 tables with at least 1000 rows. Visualization will be the main way for us to distinguish ourselves from existing sites. Users can report mistakes of current search results and after being reviewed by humans, the corresponding entity can be updated. User's reviews and user's collections are both CRUD components.