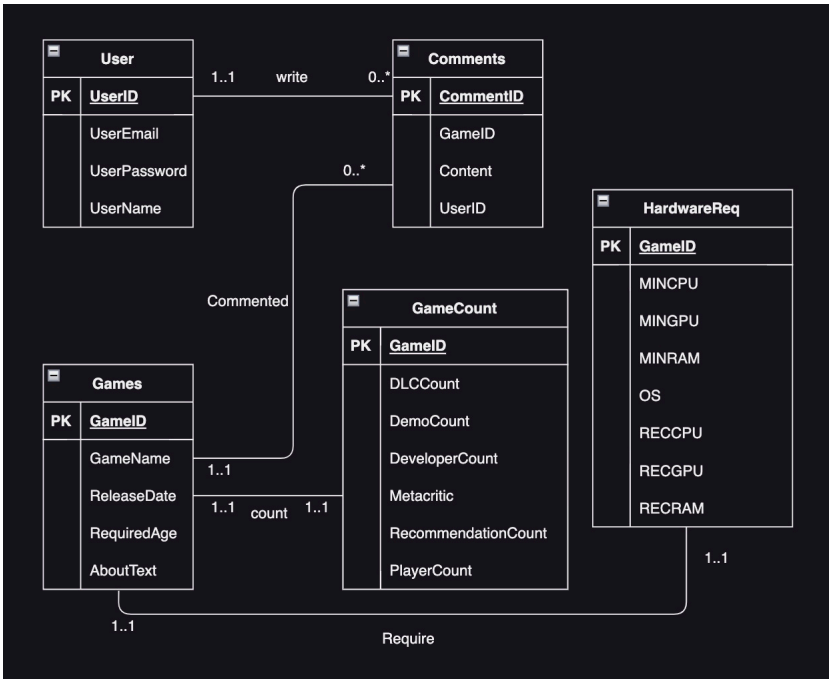
UML Diagram



Assumptions

- No duplicated UserName.
- Each email address can only be used to sign up one user login info.
- A user can only leave multiple comments on each game (one-many) and they can comment on various games.
- One comment can only be written by one user (one-one).
- Each game can only have one section of GameCount and HardwareReq which displays the number of attributes used to filter and order by.

Normalization

Functional Dependencies (based on assumptions):

1. `User` Table:
   - UserID -> UserEmail, UserPassword, UserName

2. `Games` Table:
   - GameID -> GameName, ReleaseDate, RequiredAge, AboutText

3. `Comments` Table:
   - CommentID -> GameID, content, userID

4. `GameCount` Table:
   - GameID -> DLCCount, DemoCount, DeveloperCount, Metacritic, RecommendationCount, PlayerCount

5. `HardwareReq` Table:
   - GAMEID -> MINCPU, MINGPU, MINRAM, OS, RECCPU, RECGPU, RECRAM

Analysis for *3NF*:
A relation R is in *3NF* if:
1. Whenever there is a nontrivial dependency A1,A2, ... , An -> B for R, then {A1, A2, ... , An} is a super-key for R
2. OR B is part of a key.

Based on the FDs:
1. All primary keys are single attributes, so there's no issue of partial dependency.
2. There don't appear to be any transitive dependencies. For example, in the `Games` table, none of the attributes like `ReleaseDate` or `RequiredAge` depend on each other; they all depend on `GameID`.

Analysis for BCNF:
A relation is in BCNF if:
1. It's in 3NF.
2. For every non-trivial functional dependency `X -> Y`, `X` should be a super key.

Given the identified functional dependencies:
1. Each table's attributes depend only on the primary key of that table. There don't seem to be any attributes that depend on non-super keys.

Conclusion:
From the provided ERD and the assumed functional dependencies, it appears that all tables are in BCNF. This is because:
1. All non-trivial functional dependencies are determined by the primary key.
2. There are no partial or transitive dependencies observed.

Relational Schema
- User(UserID: INT [PK], UserEmail: VARCHAR(50), UserPassword: VARCHAR(20), UserName: VARCHAR(20))
- Games(GameID: INT [PK], GameName: VARCHAR(20), ReleaseDate: VARCHAR(50), RequiredAge: INT, AboutText: VARCHAR(200))
- Comments(CommentID: INT [PK], GameID: INT [FK to Games.GamesID], content: VARCHAR(200), userID: INT [FK to User.UserID])
- GameCount(GameID: INT [FK to Games.GameID], DLCCount: INT, DemoCount: INT, DeveloperCount: INT, Metacritic: INT, RecommendationCount: INT, PlayerCount: INT)
- HardwareReq(GAMEID: INT [FK to Games.GameID], MINCPU: VARCHAR(50), MINGPU: VARCHAR(50), MINRAM: VARCHAR(50), OS: VARCHAR(50), RECCPU: VARCHAR(50), RECGPU: VARCHAR(50), RECRAM: VARCHAR(50))

Description

We have five entities: User, Comments, Games, GameCount and HardwareReq. Entity User has UserID, UserEmail, UserPassword and UserName, which is used to store information of each user. And entity Comments store information of each comment: CommentID, GameID, content of each comment and UserId who comments. Entity Games is used to store the information of each game so there will be GameID, ReleaseDate, RequiredAge, AboutText and GameName for each game. What's more, there are DLCCount, DemoCount, DeveloperCount, Metacritic, Row and GameID for each game in entity GameCount. Finally, entity HardwareReq stores the information of GameID, MINCPU, MINGPU, MINRAM, OS, RECCPU, RECGPU and RECRAM, which is about the information of hardware requirement for each game. For each user, he/she can have the number of comments ranging from zero to infinite. But there must be only one user for each comment. Also, each game has the number of comments ranging from zero to infinite, and each comment can only be for one game. For each game, there will be only one GameCount, and each GameCount is only for one game. Finally, each game has only one HardwareReq(Hardware requirments), and each HardwareReq is only for one game.