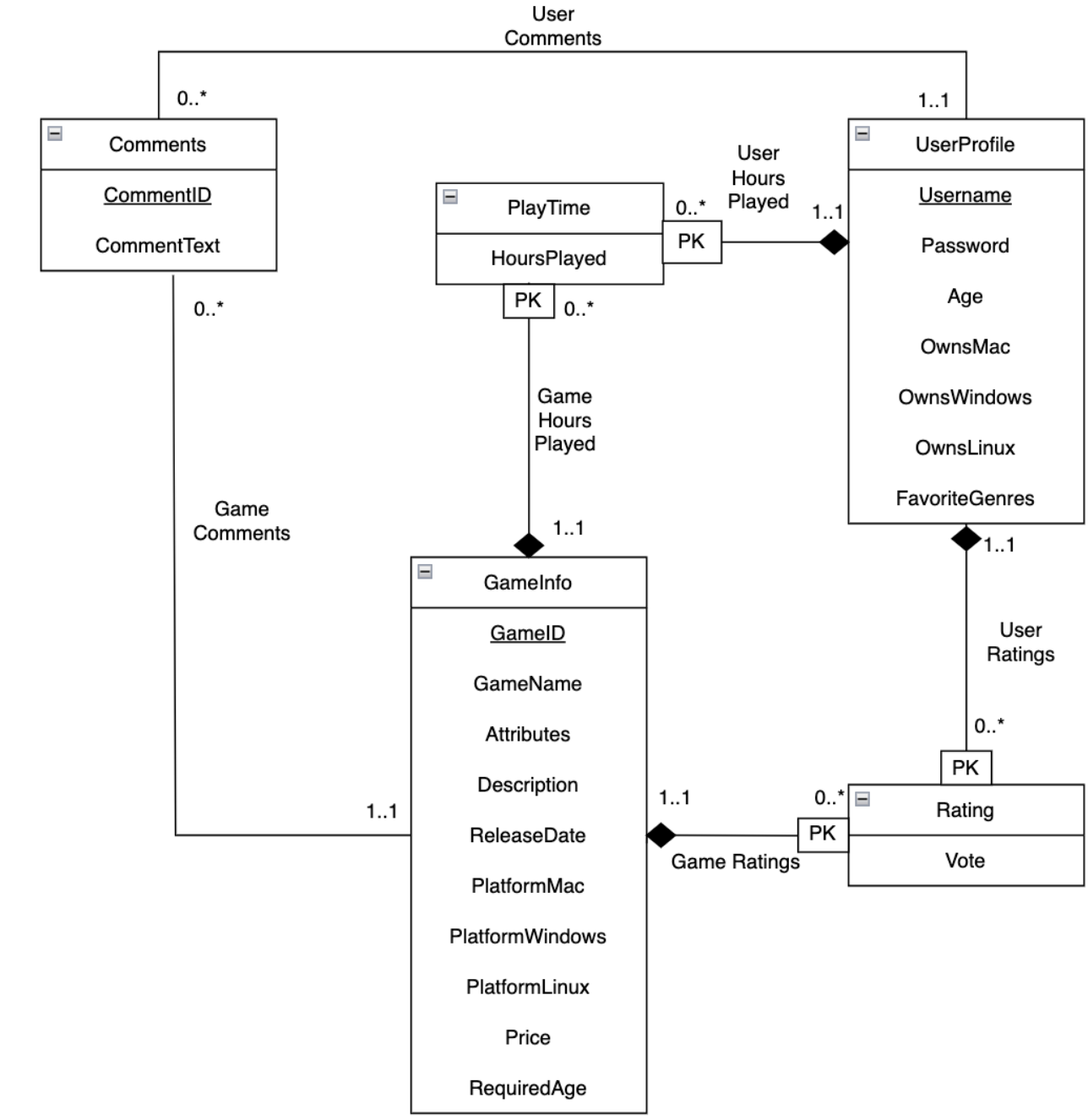
Krishna Konda, Praveen Kalva, Justin Ansell, Wyatt Sass

CS 411 Database Systems

TeamName: PreQL

Stage 2: Database Design - PreQL

UML Diagram:



Explanation of each relation/relationship:

UserProfile contains basic user info, including the user's username, password, age, favorite genres, and the types of machines they own.

Comments contains comment info, including the comment ID, the username of the user who left the comment, the gameID of the game the comment is on, and the comment text itself. This has a many-to-one relationship with UserProfile because there can be many comments related to each user, and every comment needs to have exactly one user associated with it. Similarly, it has a many-to-one relationship with GameInfo, as many comments can be related to a single game, and each comment must be related to exactly one game.

GameInfo contains information about the game, such as its id, name, attributes (genre, number of players, etc.), release date, machine compatibility, price, and required age.

PlayTime is a weak entity set that is uniquely identified by the username of the user and the gameID of the game. It contains the number of hours a specific user plays a specific game.

Rating is also a weak entity set that is uniquely identified by the username of the user and the gameID of the game. It contains the rating that a specific user left for a specific game.

Relational Schema:

UserProfile(Username:VARCHAR(50) [PK], Password:VARCHAR(50), Age:INT, OwnsMac:BOOLEAN, OwnsLinux:BOOLEAN, OwnsWindows:BOOLEAN, FavoriteGenres:VARCHAR(255))

Username→Password, Age, OwnsMac, OwnsWindows, OwnsLinux, FavoriteGenres

Explanation of Normalization: Username uniquely identifies Password, Age, OwnsMac, OwnsWindows, OwnsLinux, and FavoriteGenres; however, none of the other keys uniquely identifies any other key. Therefore, we have no troublesome FDs in this entity and it is in BCNF as UserID is a superkey for this relation.

PlayTime((Username: VARCHAR(50)[PK][FK to UserProfile.Username], GameID:INT[PK][FK to GameInfo.GameID, HoursPlayed: INT)

Username, GameID→HoursPlayed

Explanation of Normalization: Username and GameID uniquely determine HoursPlayed, but HoursPlayed does not uniquely identify any other keys. Therefore, we have no troublesome FDs in this entity and it is in BCNF as Username, GameID is a superkey for this relation.

GameInfo(GameID:INT [PK], GameName:VARCHAR(255), Attributes:VARCHAR(255), Description:VARCHAR(1023), ReleaseDate:DATE, PlatformMac:BOOLEAN, PlatformWindows:BOOLEAN, PlatformLinux:BOOLEAN, Price:INT, RequiredAge:INT)

GameID→GameName, Attributes, Description, ReleaseDate, PlatformMac, PlatformWindows, PlatformLinux, Price, RequiredAge

Explanation of Normalization: GameID uniquely determines GameName, Attributes, Description, ReleaseDate, PlatformMac, PlatformWindows, PlatformLinux, Price, and RequiredAge; however, none of the other keys uniquely identifies any other key. Therefore, we have no troublesome FDs in this entity and it is in BCNF as GameID is a superkey for this relation.

Comments(CommentID:VARCHAR(255) [PK], GameID:INT [FK to GameInfo.GameID], Username:VARCHAR(50) [FK to UserProfile.Username], CommentText:VARCHAR(1000))

CommentID→GameID, Username, CommentText

Explanation of Normalization: CommentID uniquely identifies GameID, Username, and CommentText; however, none of the other keys uniquely identifies any other key. In our application, a user can write multiple comments on the same game, so the GameID and Username can have duplicates in this relation! Therefore, we have no troublesome FDs in this entity and it is in BCNF as CommentID is a superkey for this relation.

Rating((Username: VARCHAR(50)[PK][FK to UserProfile.Username], GameID:INT[PK][FK to GameInfo.GameID, Vote:INT)

Username, GameID → Vote

Explanation of Normalization: Username and GameID uniquely determine Vote, but Vote does not uniquely identify any other keys. Therefore, we have no troublesome FDs in this entity and it is in BCNF as Username, GameID is a superkey for this relation.

Explanation of Normalization:

Since the only functional dependencies in our schema are from the primary key to the other keys in the table, there are no troublesome functional dependencies present. Thus, our database is normalized.

UML Assumptions:

We assumed that users can only rate/vote on a specific game 1 time. However, we assumed that users could comment on the same game multiple times. Additionally, users can report playtime for every game they played but each playtime entry is associated with a distinct game and user.