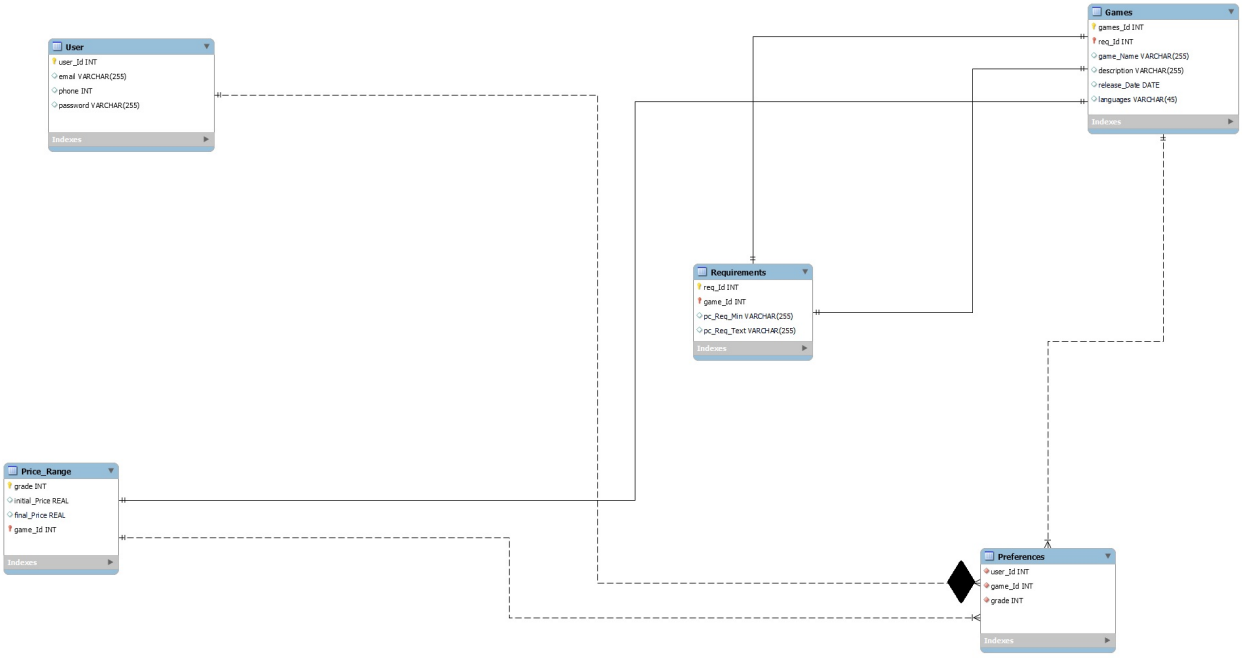

Relational Schema:

User(UserId:INT[PK], Email:VARCHAR(255), PhoneNumber:REAL,
Password:VARCHAR(50))

Games(GameId:INT[PK], ReqId:INT[FK to Requirements.ReqId],
GameName:VARCHAR(255), Description:VARCHAR(255), ReleaseDate:DATE,
Languages:VARCHAR(45))

Requirements(ReqId:INT[PK], GameId:INT[FK to Games.GameId],
PCReqMin:VARCHAR(255), PCReqText:VARCHAR(255))

Preferences(UserId:INT[FK to User.UserId], GameId:INT[FK to Games.GameId],
Grade:INT[FK to PriceRange.Grade])

PriceRange(Grade:INT[PK], GameId:INT[FK to Games.GameId], Initial:REAL,
Final:REAL)

Assumptions/Relationships:

a. Games and Requirements Cyclic Dependency:
i. The Games table and the Requirements table have a cyclic relationship. Both tables reference each other through foreign keys. Every game has requirements, and every requirement corresponds to a specific game.
b. User and Preferences Cyclic Dependency:
i. The User table references the Preferences table through a foreign key, and the Preferences table references the User table. This cyclic dependency indicates that a user has preferences, and each preference set can be traced back to a specific user.
c. Game Pricing:
i. The Price_Range table defines a range of prices (initial_Price to final_Price) for games based on a grade, indicating a tiered pricing system for games.
d. User Preferences for Games:
i. Users have specific game preferences stored in the Preferences table.
ii. The Preference_Games table is a bridge table that relates games to user preferences, indicating that users can have preferences for multiple games.
e. Preferences and Games Relationship:
i. The Preferences_has_Games table is an associative table that maps preferences to games. The presence of multiple foreign keys from both the Preferences and Games tables in the primary key of this table indicates a many-to-many relationship between games and preferences.
f. User Identification:
i. Users are identified by a unique user_Id and have associated email, phone, and password details.

To determine if the provided DDL is in 3NF, let's first lay out the functional dependencies (FDs) we identified:
1. User
Attributes:

- user_Id (Primary Key)
- email
- phone
- password

FDs:

- user_Id→email,phone,password
- user_Id→email,phone,password

2. Requirements
Attributes:

- req_Id (Primary Key, part of a composite key with game_Id)
- game_Id (Foreign Key referencing Games)

- pc_Req_Min
- pc_Req_Text

FDs:

- req_Id,game_Id→pc_Req_Min,pc_Req_Text
- req_Id,game_Id→pc_Req_Min,pc_Req_Text

3. Games

Attributes:

- games_Id (Primary Key, part of a composite key with req_Id)
- game_Name
- description
- release_Date
- languages

FDs:

- games_Id→game_Name,description,release_Date,languages
- games_Id→game_Name,description,release_Date,languages

4. Price_Range

Attributes:

- grade (Primary Key, part of a composite key with game_Id)
- initial_Price
- final_Price
- game_Id (Foreign Key referencing Games)

FDs:

- grade,game_Id→initial_Price,final_Price
- grade,game_Id→initial_Price,final_Price

5. Preferences

Attributes:

- user_Id (Foreign Key referencing User)
- game_Id (Foreign Key referencing Games)
- grade (Foreign Key referencing Price_Range)

FDs:

- user_Id,game_Id→grade
- user_Id,game_Id→grade

Normalization Analysis:

First Normal Form (1NF):

- No repeating groups.
- Data values are atomic.
- Each table has a primary key.

The provided DDL satisfies 1NF.

Second Normal Form (2NF):

- All requirements of 1NF.
- No partial dependency.

For tables with composite keys (Requirements, Games, Price_Range), there are no non-key attributes that are dependent on just a part of the primary key. Thus, the provided DDL satisfies 2NF.

Third Normal Form (3NF):

- All requirements of 2NF.
- No transitive dependencies of non-key attributes on the primary key.

Conclusion:

The DDL provided is in 2NF and 3NF due to the transitive dependency observed in the Games table.