James Zafiri

12/04/2023

CSC6302

**Project Six: BCNF**

1. Functional Requirements:
2. Store information about characters like their name, level, class they belong in
3. Assign a character to a specific player
4. Allow character appearance to be customized, for example their hair, hair color clothing, skin color, character type/trait
5. Be able to add skills/abilities to a character
6. Keep an inventory of the character’s weapons/tools
7. Rough Draft of Database Design:

GameUser Table

Username varchar(30)

UserAge int

UserLevel int

PlayerCharacter Table

CharacterName varchar(30)

Username varchar(30) FK to User Table

CharacterLevel int

SkinColor varchar(20)

Hairstyle varchar(20)

HairColor varchar(20)

Clothing varchar(30)

Type varchar(20)

Skill Table

SkillName varchar(30)

SkillLevel int

CharacterName varchar(30) FK to PlayerCharacter Table

Equipment Table

ItemName varchar(30)

Quantity int

CharacterName varchar(30) FK to PlayerCharacter Table

1. I think that each of the relations in my rough draft appear to be in 1NF. They each have unique column names, well defined domains within the columns, and the columns have one distinct value per row. Of course, each table could use an Id for its primary key, but for the simplicity of this project we are not doing that right now. I do not need to make any changes at this step.
2. Candidate Keys:

GameUser Table

(Username, UserAge)

(Username, UserLevel)

**(Username)**  -> makes most sense as primary key.

PlayerCharacter Table

(CharacterName)

(CharacterName, CharacterLevel)

(CharacterName, CharacterLevel, CharacterType)

(CharacterName, Username, CharacterLevel)

**(CharacterName, Username)** -> makes most sense as primary key.

Skill Table

(SkillName)

(SkillName, SkillLevel)

(SkillName, SkillLevel, CharacterName)

**(SkillName, CharacterName)** -> makes most sense as primary key.

Equipment Table

(ItemName)  
(ItemName, Quantity)

(ItemName, Quantity, CharacterName)  
**(ItemName, CharacterName)** -> makes most sense as primary key.

1. I think that all the relations in my draft so far are not completely in 2NF. For example, in the “PlayerCharacter” table, the character attributes depend on the entire primary key of (CharacterName, Username), but not solely on Username. To fix this, I think it’s a good idea to make a separate table just for appearances of a character. This way I can make the attributes solely depend on the primary key. I will make these changes and create a new ER diagram to reflect all the appearance rows just depend on the character name.
2. I think that all the relations in my draft so far appear to be in 3NF. This is because I made it so that the columns in a table strictly depend on that table’s primary key for uniqueness/context, so there are no transitive dependencies. For the appearance, any attribute will solely depend on just the name of the character. For the user, the level and age solely depend on the username. For the player’s character, their level and what type of character depend on the user’s name and who the character is. For skill, the level of the skill depends on what the skill name is and which character it belongs to. For equipment, the quantity depends on the item and which character it belongs to.