Database Application Design

Overview

In this initial phase of your group project, you will focus on the intricate process of crafting a resilient database application using the entity-relationship (E-R) model. Your tasks include listing essential entity sets with underlined primary keys and delineating associated attributes, clarifying relationships between different entity sets, and constructing a comprehensive E-R diagram. You will also offer detailed rationale behind your design choices, highlighting how each element enhances the overall functionality of the envisioned database application.

Part I: Setting Up Your Group's GitHub Repository

Now that you've formed your groups for the semester, you will create a shared GitHub repository for your group project. This repository will serve as a central workspace for all project materials including design documents, E-R diagrams, presentation slides, database scripts and application code.

Setting up your repository supports collaboration, version control, and transparency in individual contributions—and mirrors how real-world development teams work. All group members are expected to make regular, meaningful contributions to the shared repository. Commit history may be considered alongside peer evaluations when assessing individual participation in the group project.

ALL Group Members Must Complete Following:

- ☑ Complete the Install Your IDE section to install an IDE on your local device that supports web development (e.g., VS Code).

- ☑ Complete the Create A GitHub Account section if you do not already have one.

- ☑ Choose ONE of the following options:

    - ☑ Complete the Installation & Authentication section to install and authenticate GitHub Desktop (GUI) on your device.

    - ☑ Complete the Installation & Authentication section to install and authenticate Git (CLI) on your device.

ONE Group Member Must Complete Following:

☑ ~~Complete the~~ ~~Create a GitHub Repository for Your Course Projects section~~ ~~to create your group's repository.~~

☑ ~~Clone the repository to your local device using~~ ~~GitHub Desktop~~ ~~or~~ ~~Git CLI.~~

☐ Navigate to and open the repo folder you cloned to your local device (not the one on GitHub.com).

☑ ~~Inside your repo folder, create a new folder named~~ **~~documentation~~**~~.~~

☑ ~~In your group's repository on GitHub, go to the~~ **~~Settings~~** ~~tab → ~~**~~Collaborators~~**~~.~~

☑ ~~Click~~ **~~Add people~~**~~, then enter your group members' GitHub usernames and send invitations. This step ensures all group members have access to the shared repository.~~

ALL Group Members (After Receiving the Invitation) Complete Following:

☑ ~~In your GitHub dashboard, click the~~ **~~Notifications~~** ~~icon (~~ 🔔 ~~). Open the invitation message and click~~ **~~Accept Invitation~~** ~~to join the repo as a collaborator.~~

☑ ~~Clone the repository to your local device using~~ ~~GitHub Desktop~~ ~~or~~ ~~Git CLI.~~

☑ ~~Open the repository in VS Code and confirm that it contains the expected structure (e.g., the~~ **~~documentation~~** ~~folder).~~

Part II: Database Application Design

Design your group's database application using the E-R model by providing thoughtful responses to the following questions.

1. List the entity sets and their attributes with primary keys underlined in your design.

   Users(<u>username</u>, password)
   Location(<u>locationID</u>, name, description, details, partyNotes)
   Prop(<u>ID</u>, name, details, description, partyNotes)

   These inherit attributes & relations from *Prop* entity set
   {
   Creature(creatureType, population)
   NPC(opinions, race/species)
   Player(level, race/species)
   Item(itemType, rarity, quantity)
   }

2. List the relationships between different entity sets in your design.
   Users would be related to *Player* through the *Plays* relationship set, which is one-to-many.

   Location would be related to itself through the *Contains* relationship set, which is a one-to-many relationship representing how individual locations can contain sublocations.

   Prop would be related to *Location* through the *IsAt* relationship set, which is many-to-one.

   Item would be related to *Prop* through the *OwnedBy* relationship set, which is many-to-one and represents the items owned by an individual and the individual parts of a larger item. This would have an attribute *number* to show how many of that item are owned by the prop.
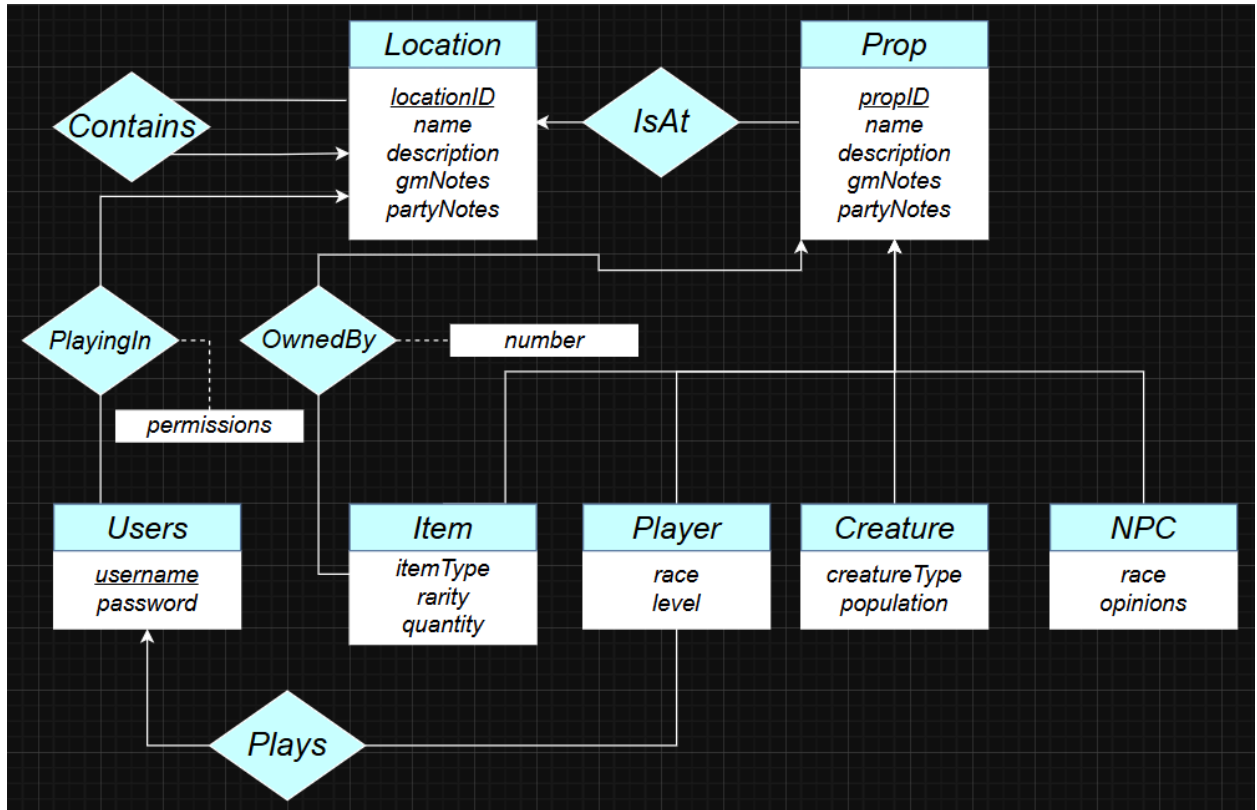
   Users would be related to *Location* through the *PlayingIn* relationship set, which is many-to-one and represents players being part of an overarching game world. This would have an attribute *permissions* to determine what information that user can see.

3. Construct an E-R diagram for your database application. Use the following steps to design your database using the E-R model:

   ● Start by identifying essential entity sets to be included in your database.

   ● Choose relevant attributes representing the values to be captured in the database for each entity set.

   ● Formulate relationship sets among entities, addressing potential redundancy in attributes.

   ● Incorporate any necessary constraints (ie, relationship cardinalities, total/partial participation, descriptive attributes, weak entity sets).

You can use draw.io to create your E-R diagram. To collaborate with your group, click **File → Share** and grant edit access to your group members so everyone can work on the diagram in real time.

[Link to E-R diagram](#)

4. Provide a detailed explanation of the rationale behind your design choice. Highlight how each element contributes to the overall functionality of your database application.

For our Entity–Relationship diagram, we decided to include three parent entities. This design choice was chosen in order to highlight an object-oriented framework, in which 'objects' of similar type can inherit common attributes in order to reduce overhead. The three parent entities we chose were 'Prop', 'Location', and 'User'. While 'Location' and 'Prop' also have similar attributes, their relationships are specifically defined, justifying their need to be separate entities. Any generic 'Prop" can be related to 'Location' using the 'IsAt' relationship. A 'Location' also has a specific relationship with itself using the 'Contains' relationship. This allows sublocations to exist, from globular, to national, to regional differences. A 'Prop' has many sub-entities, like 'Item', 'Player', 'Creature', and 'NPC'. Each of these have specific attributes tied to them. The 'Item' sub-entity can be related to any generic 'Prop' using the OwnedBy relationship, with its own relationship attribute defining its *number*. The 'Player' entity is related to the 'User' entity through the relationship 'Plays', which connects a specific user to a specific player. Finally, a 'User' entity is related to the 'Location' entity through the 'PlayingIn' relationship, which requires the permissions relationship attribute.

Submission

When you're finished, complete the following steps to submit your work:

☐ Export your document with responses as a **PDF file AND save** it **inside** your **documentation** folder. Refer to the following for documentation on how to do this:

- [Google Docs](#) (*File → Download → PDF Document*)

- [Microsoft Word](#) (*File → Save As / Export → PDF*)

- [Pages](#) (*File → Export To → PDF*)

☐ Export your **E-R diagram** as an **image file** and save it **inside** your **documentation** folder. Be sure your file is clearly named (i.e., **er_diagram.png**).

☐ Upload all your changes to GitHub.

  ☐ **If you're using GitHub Desktop (GUI)**, complete the [Uploading Changes (GitHub Desktop) section](#) to upload your changes from your local device to GitHub.

  ☐ **If you're using Git (CLI)**, complete the [Uploading Changes (GitHub CLI) section](#) to upload your changes from your local device to GitHub.

ONE group member must paste the URL of your GitHub repository in the provided textbox in Brightspace. Click the blue *Submit* button to successfully submit your work for this assignment.

Grading Rubric

You can refer to the **Database Application Design grading rubric** given in Brightspace for this assignment to find details on how your submission will be graded.