Project Step 4 Draft Version: DML and DDL Queries (CS340)

URL:http://web.engr.oregonstate.edu/~xuna/340_proj/

a) ***Team Member Names:***
   - i) David Kwak
   - ii) Naisen Xu

b) ***Project Title: Animal Crossing (The Deserted Island Handbook)***

c) ***Project Step 3 Final Version Changes (5/10/20)***
   - ***i) Feedback:***
     - ***1) SeriesID of Villager and Furniture should be removed***
     - ***2) Website feedback from peers on piazza***
       - ***(a) Focuses on UI not having a SELECT for every table***
       - ***(b) Make it more functional (Aware of this issue)***
         - ***(i) Graphic changes***
   - ***ii) Fixes:***
     - ***1) Removed SeriesID from villager and furniture***

d) ***Project Step 2 Changes:***
   - ***i) Fixes: The reason for switching over to this idea was due to because the last project we wanted wasn't super clear. After realizing more about what we wanted and possibly decided to make a database on Animal Crossing which is more towards something that's more fun and motivational. Became more clear what we wanted afterwards therefore improving the project as a whole.***
   - ***ii) Feedback by the peer Reviewer:***
     - ***1) Based on the feedback it claimed of how there isn't a clearly indicated table which each team member is primarily assigned for.***
   - ***iii) Actions based on feedback:***
     - ***1) Through this feedback we'll have to discuss who will do what in a more clear manner and also find a new way to communicate.***
   - ***iv) Changes to the Draft Version Version 2:***
     - ***1) No changes this version***

e) ***Project Step 3 Changes:***

   i) ***Feedback by peer/TA:***

      1) ***Fix the relationship between Furniture and Furniture Series -> should be 1:M***
      2) ***Recipe -> Material shouldn't be M:M***
      3) ***Too many entities for this project -> basically lighten the project***

   ii) ***Changes:***

      1) ***Fixed the relationship between Furniture and Furniture Series to 1:M***
      2) ***Changed Recipe:Material to 1:M relationship because for every one recipe a user must have different material in order to build it.***
      3) ***Made some changes to the entities of the project getting rid of some in order to lighten the project a bit.***
      4) ***Decided to change the recipe and get rid of it along with material and instead put it in a museum. Because material and recipe were a bit extra and are a part of furniture. Meanwhile museum can be its own entity due to the amount of bugs and fishes in the game along with the many different variables for it.***
      5) ***Used this website in order to host the website: https://oregonstate.teamdynamix.com/TDClient/1935/Portal/KB/ArticleDet?ID=45669 however the images aren't loading and I have to look into this problem.***
         (a) *http://people.oregonstate.edu/~kwakd/*

f) ***Overview**:*

   - ***Animal Crossing has been a phenomenal hit currently with it's perfect release date in a time of quarantine. And especially with a game that's meant to be soothing and relaxing it has a slow pace of gaming. Which in return means a player won't be able to view all 397 villagers in the game in order to make their ideal deserted island. This website will basically allow a user to create their perfect island with their favorite villagers along with the furniture series these villagers collect and the materials necessary for these furnitures (if applicable). Basically this site will be a simple website where you look for your favorite category of a villager (example: lazy turtle) and it'll find the villager for you. And you can choose to put the villager***

on your island with a max of 10 and it'll even save your deserted island.

- ***Entity: Villager, Player, Furniture, Material, Recipe***

- ***Work will be split evenly among group members.***

g) ***Database Outline, in Words:***

1. ***Villager****: Entity for the villagers with all their information.*
    a. VillagerID: int, auto_increment, not NULL, Primary Key
    b. VillagerName: varchar, not NULL
    c. VillagerPhrase: varchar, not NULL
    d. VillagerGender: int, not NULL
    e. VillagerPersonality: int, not NULL
    f. VillagerSpecies: int, not NULL
    g. **VillagerClothes: int, not NULL?**
    h. VillagerBirthday: Date, not NULL

    Relationship

    1. **1:M to Furniture_Series because for every villager they can only have one series/set of furniture but multiple villagers can have the same series/set of furniture. For example "Egbert" has furniture from the "Robo Series" that his house will consist of. However "Cube" also has furniture which consist from the "Robo Series". However villagers will not mix and match sets/series.????**
    2. ~~**1:1 to Island because every villager is unique to the user.**~~
    3. 1:M to island or "settledVillager" because a same villager can be settled in different islands

2. ***User****: Entity for user information to keep track of their village count and villagers.*
    a. UserID: int, auto_increment, not NULL, Primary Key
    b. UserName: varchar, not NULL
    c. UserVillagerMax: int, not NULL
        i. ***Relationship***
            1. 1:1 to Island because there's only One island per User.

3. **Museum**: Entity for the museum for creatures such as fish and bugs
    a. MuseumID: int, auto_increment, not NULL, Primary Key
    b. museumName: varchar, not NULL
    c. museumType: varchar, not NULL
    d. museumSell: int
    e. museumLocation: varchar, not NULL
    f. museumShadow: int
    g. museumTime: int
    h. museumDate: int
        i. Realized museum could be its own entity due to the amount of fishes and bugs in the game. Connects to the user and has a 1:M relationship because a fish can only be caught per one user or at the least record it.

4. ***Furniture**: Entity for all the different furnitures in the game.*
    a. FurnitureID: int, auto_increment, not NULL, Primary Key
    ~~**b. RecipeID: int**~~
    c. FurnitureSize: int
    d. FurniturePriceSold: int
    e. FurniturePriceBought: int
    f. FurnitureSource: int
    g. FurnitureCustomizable: int
        i. Relationship
            1. 1:1 to Recipe because for every one recipe you can only have one type of furniture. For example an “acoustic guitar diy recipe” can only make “acoustic guitars.” Therefore it is a 1:1 relationship to Recipe.
            2. 1:M to Furniture Series because each furniture belongs to a series/set in the game, however these series/sets have multiple furniture in them. For example the “Antique Bed” can only belong to the “Antique Set.” However the “Antique Set” will also consist of “Antique Clock, Antique Bed, etc.” Therefore although a furniture can only belong to one set, these sets have multiple furniture in them.

~~*5. Furniture_Series: Connects the Furniture Series to a villager. Because certain villagers have certain furniture from a series.*~~
    ~~*a.* SeriesID*:* int, auto_increment, not NULL, Primary Key~~

~~b. VillagerID: int~~

~~c. FurnitreID: int~~

~~i. ***Relationship***~~

~~1. M:1 for Furniture because as stated above because a furniture can belong to one series/set however there can be multiple furniture inside this one series/set.~~

~~2. 1:M for Villagers because for every one villager they will only have one set of furniture however multiple villagers can use the same set.~~

6. *Island: Connects all the information for Players and their Villagers basically the island itself in a way.*

    a. IslandID*:* int, auto_increment, not NULL, Primary Key

    b. IslandName*: varchar, not NULL*

    c. VillagerID*: int*

    d. *UserID: int*

    i. ***Relationship***

    1. M:M to User because it is all the different VillagerID's or entities in order to make sure that the user's island loads with their saves. Therefore to many user's there will also be many different villagers therefore making it a M:M relationship.

    2. M:M to Villager because of the same reasons above, due to how it connects the user's specific settings to their certain villagers.

7. *Furniture_items: Connect all furnitures and needed materials to made themselves*

    a. itemID: int, auto_increment, not NULL, Primary Key

    b. furnitureID: int, not NULL

    c. materialID: int, not NULL

    d. number(number of materials): int, not NULL

    I. RELATIONSHIP

    M:M to Material because a furniture might be made by more than one kind of material and a kind of material can make at least one kind of furniture

8. *Materials: Entity for all kinds of materials which can make a series of furniture*

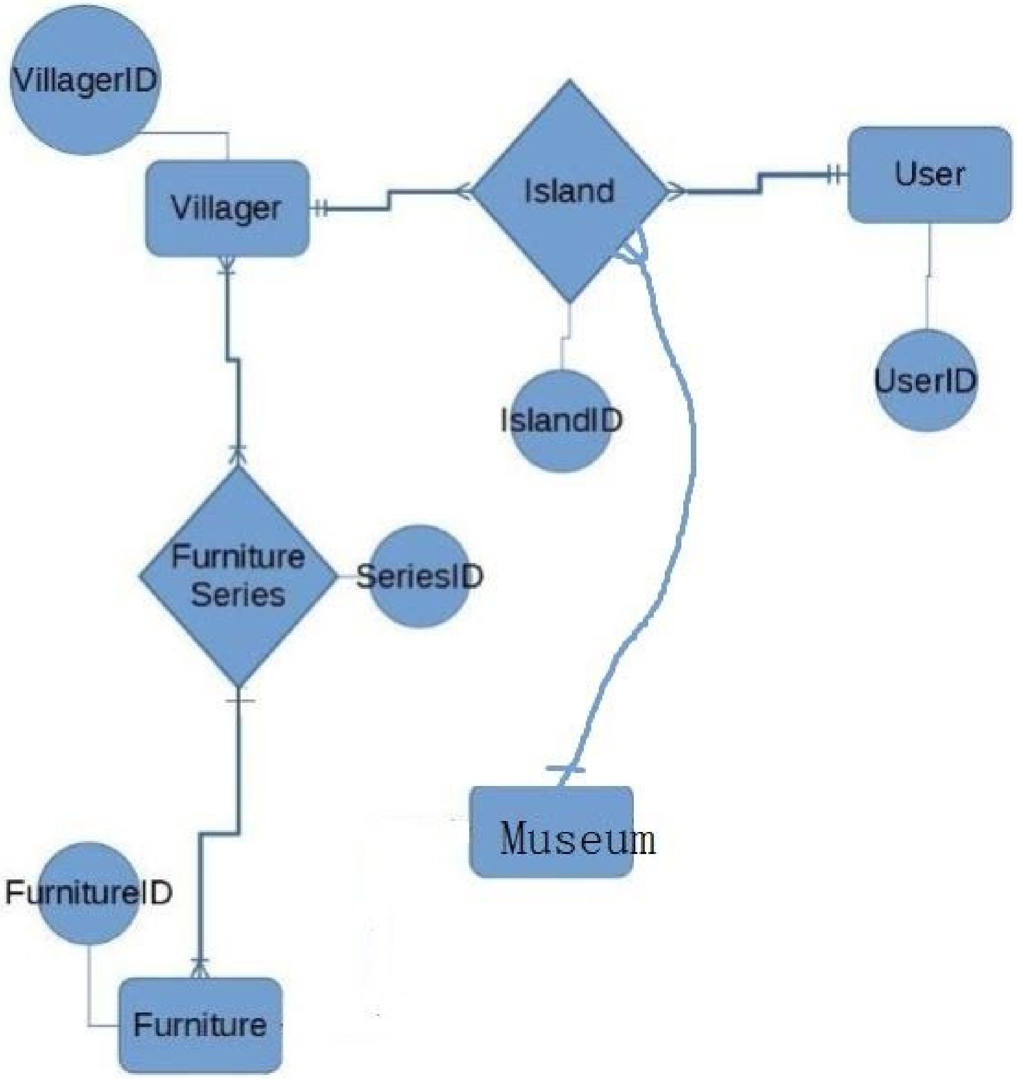
a. *materialID: int, auto_increment, not NULL, Primary Key*

b. *materialName: varchar, not NULL*

c. *Rarity: int, DEFAULT NULL*

9. *settledVillager: Connect villagers and islands*
   a. *settledID: int, auto_increment, not NULL, Primary Key*
   b. *villagerID: int*
   c. *islandID: int*
   d. *RELATIONSHIP:*
      i. *M:1 to island because an island can have at most 10 settled villagers.*

h) Entity-Relationship Diagram:


VillagerID
Villager
Island
User
IslandID
UserID
Furniture Series
SeriesID
Museum
FurnitureID
Furniture

i) Schema:

Attribute in Blue is Foreign Key
Attribute in Purple is Primary Key

Villager(
VillagerID,
VillagerPhrase,
VillagerGender,
VillagerSpecies,
VillagerClothes,
~~VillagerCoffee~~
)

User(
UserID,
UserName,
~~UserVillagerMax,~~
)

Furniture(
FurnitureID,
~~**RecipeID,**~~
FurnitureSize,
FurniturePriceSold,
FurniturePriceBought,
FurnitureSource,
FurnitureCustomizable
)

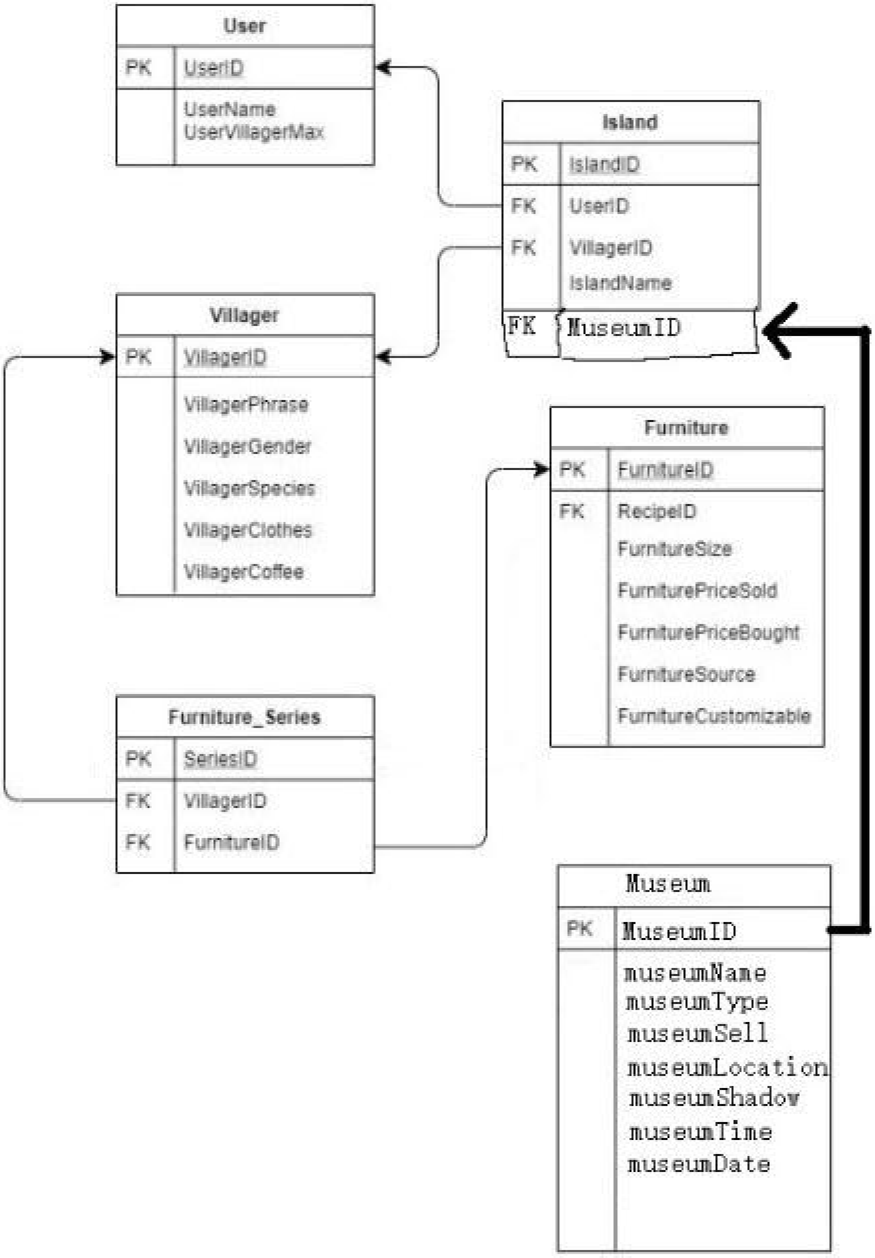
Museum(
MuseumID,
museumName
museumType
museumSell
museumLocation
museumShadow
museumTime
museumDate
)

~~Furniture_Series(~~
~~SeriesID,~~
~~VillagerID,~~
~~FurnitureID,~~
~~)~~

Island(
IslandID,
UserID,
~~VillagerID,~~
IslandName
)

recipe_items(
itemID,
furnitureID,
materialID
)

material(
materialID,
rarity
)
settled(
settledID,
villagerID,
islandID
)



User	
PK	UserID
	UserName UserVillagerMax

Island	
PK	IslandID
FK	UserID
FK	VillagerID
	IslandName
FK	MuseumID

Villager	
PK	VillagerID
	VillagerPhrase VillagerGender VillagerSpecies VillagerClothes VillagerCoffee

Furniture	
PK	FurnitureID
FK	RecipeID
	FurnitureSize FurniturePriceSold FurniturePriceBought FurnitureSource FurnitureCustomizable

Furniture_Series	
PK	SeriesID
FK	VillagerID
FK	FurnitureID

Museum	
PK	MuseumID
	museumName museumType museumSell museumLocation museumShadow museumTime museumDate