

Autumn Hills Database

Note: Since I am using the same data design I originally decided upon, I have included some new stretch features beginning on page 4.

For the Autumn Hills Country Club web application, I will be utilizing a relational MySQL database. I have opted to utilize MySQL since I am mostly familiar with it, but also since the structure of the data stored in the database most likely won't change, so we won't need to utilize the flexibility of a document database. MySQL will also be useful as the application grows in the future so that it can scale with our data needs. For the MVP of this project, Autumn Hills needs a way for users to sign up for accounts in their online system, and from there, the users can book tee times to play a round of golf on the course.

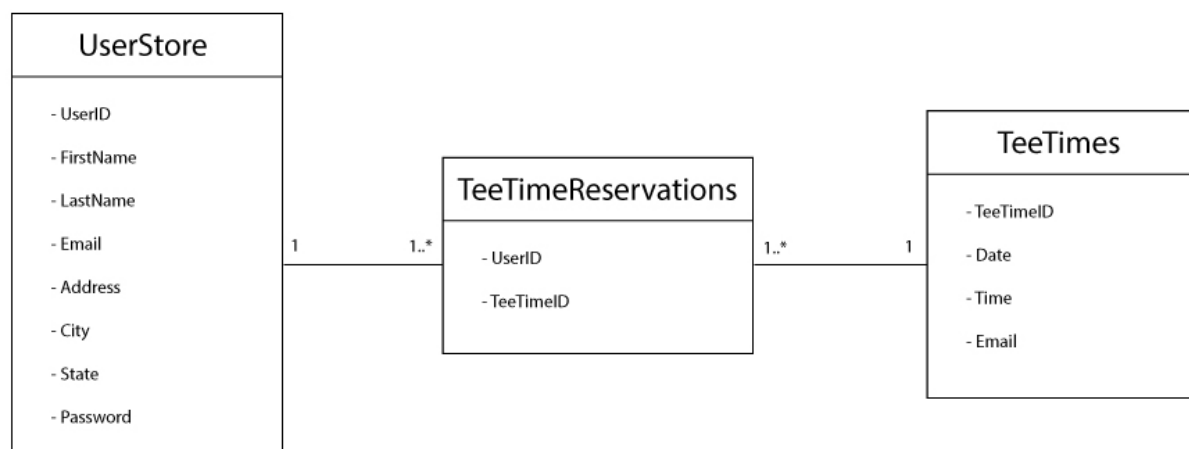
In the future if Autumn Hills plans to implement added functionality outside of the scope of the MVP; such as make a reservation at the restaurant, make a purchase at the pro shop, etc., then more and more relationships will need to be established in the database. Because of this potential expansion with new tables and new relationships, along with new rows being added to the tables, the MySQL solution makes sense to implement.

With MySQL databases, there are plenty of data structures to choose from. For Autumn Hills I will mostly be utilizing queries to the database in order to add users. If a

user has an account and wants to log in, then I will have a query cross reference the UserStore table to make sure their email address and password exist within the database, and if they do then the user can log in. Other data structures that may need to be used will be joins between tables. When a user books a tee time there will need to be a way to associate a tee time with the user who booked the tee time.

The Autumn Hills web application will be built as a server side JavaScript application using Node.js and Express with a MySQL dependency. Express will query the database and will display data on pages in the application through various routes. Server side JavaScript provides the ability for this application to be fast and lightweight while establishing a connection with the database. The database itself is housed within AWS RDS as a MySQL database.

Entity Relationship Diagram



UserStore Table - this table is what will ultimately store the user information when they sign up for an account within the Autumn Hills system. Their email address and password information will be utilized for login purposes.

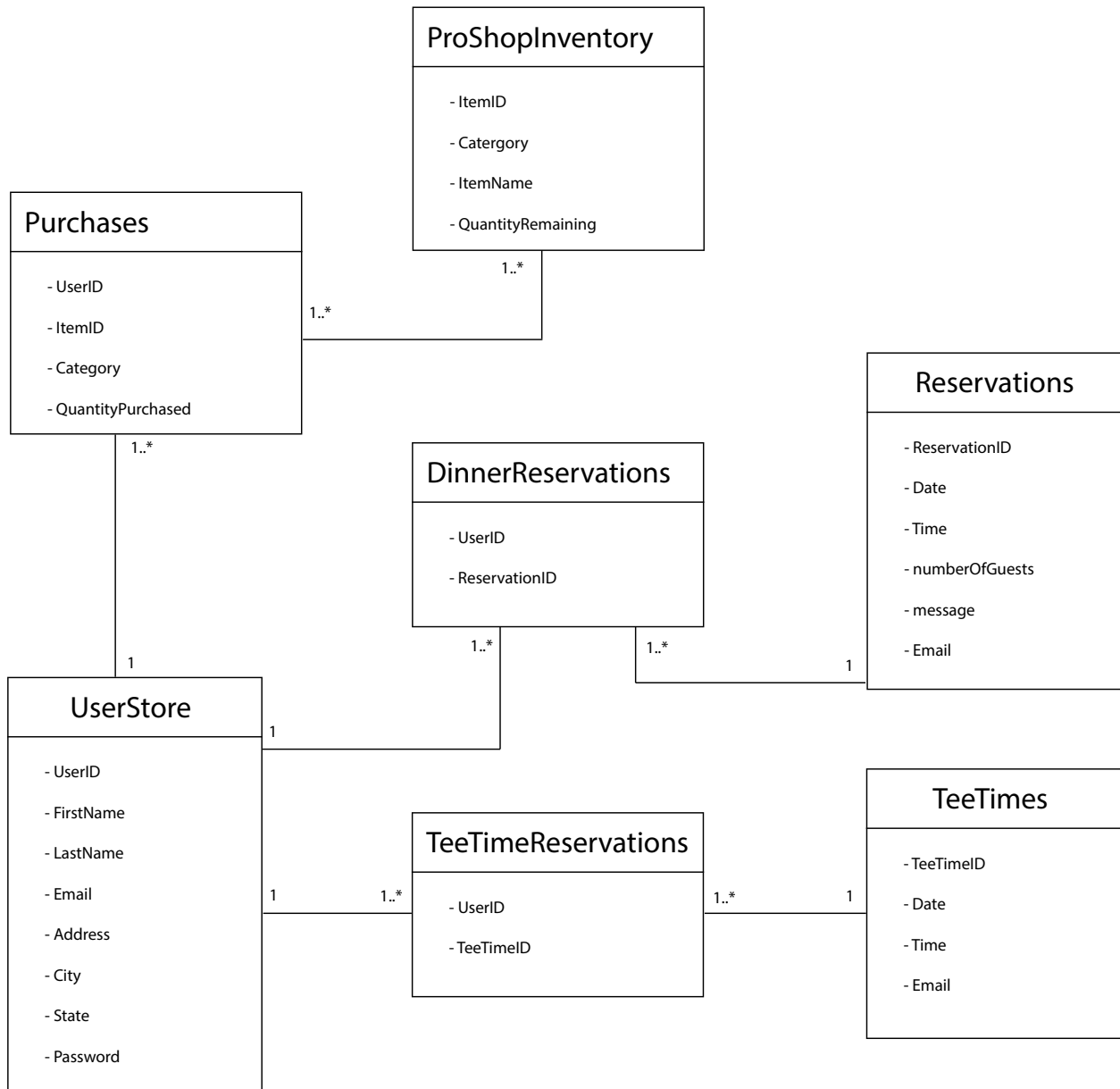
TeeTimes Table - this table will store the tee time reservations. An ID is associated with each time, and the date and time are stored in the table. The email address associated with the user acts as the secondary key to associate a tee time to a user.

TeeTimeReservations Table - this table combines the primary key UserID from the UserStore table and the TeeTimeID of the TeeTimes table. This table simply shows the UserID associated with a TeeTime ID.

In the above database system, there will be an admin user who will most likely be the owner of the country club and they will have read/write access to the database. The reason for this is, again, should Autumn Hills decide to expand their application to more clubhouse services such as pro shop purchases, then more management of the system will be required. Employees would also have database access which would be read only. If they need to make modifications then they will need manager approval.

Stretch Features

As we look to the future, we also need to plan for stretch features in our database table. I've expanded upon the existing MVP database and included new tables to incorporate the stretch features:



The above database system includes the following tables to incorporate the stretch features that can be incorporated in the future after the MVP:

Reservations Table - this table will store the dinner time reservations at the country club. An ID is associated with the time, date, number of guests, and message (if provided) in the table. The email address associated with the user acts as the secondary key to associate a reservation to a user.

DinnerReservations Table - this table combines the primary key UserID from the UserStore table and the ReservationID of the Reservations table. This table simply shows the UserID associated with a ReservationsID.

ProShopInventory Table - this table stores the inventory of the online pro shop where users can purchase items online. This table organizes each item by its ItemID and then its Category and the ItemName within that category. Finally, the quantity remaining of those items is stored.

Purchases Table - this table simply stores purchase history made at the pro shop, and which user purchased which items. This would be useful in the event of a user needing to make a return on an item. The table associates the UserID with the ItemID purchased along with how many items the user purchased.