Connor Simmons

Music Theory Query

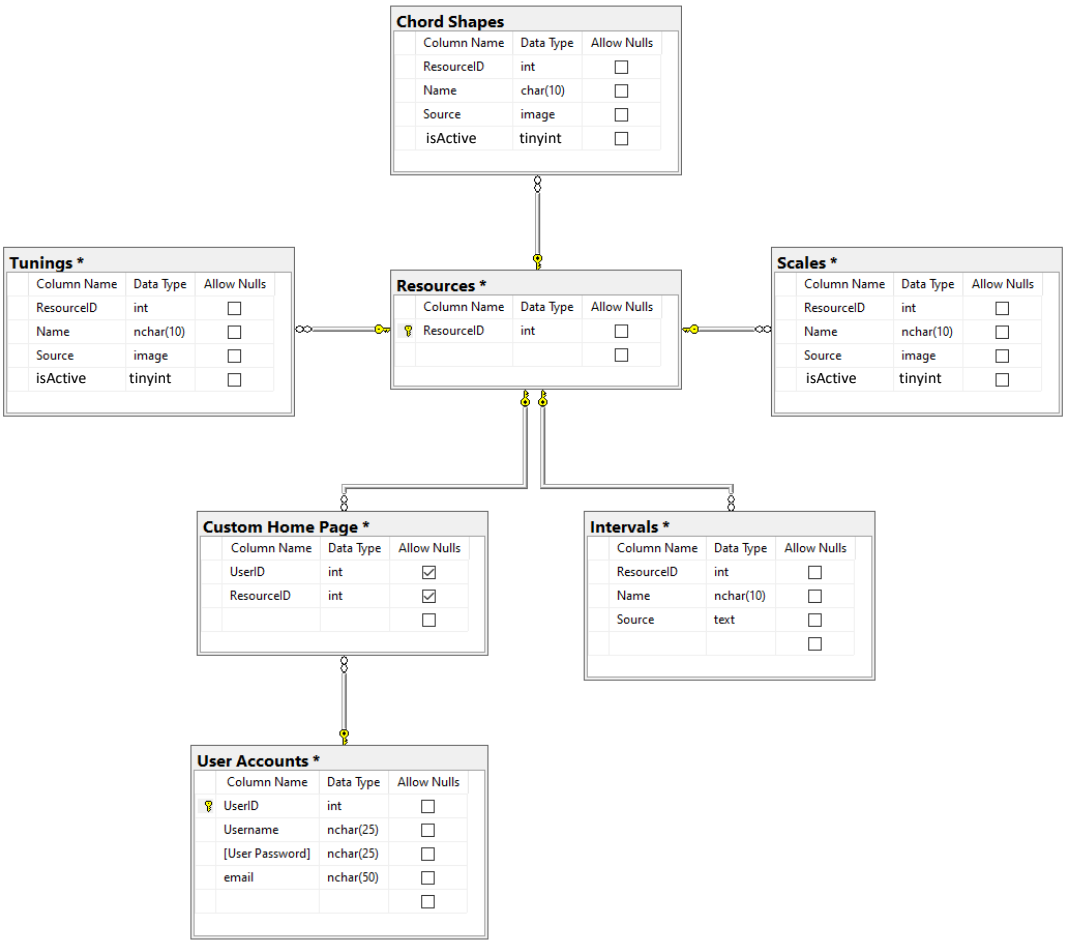
Database Redesign

Overview

Original justification:

Since my application will make use of one-to-many relationships and rows may change frequently in at least one of the tables, I have decided to use a relational database to support Music Theory Query. Below is my first proposal for a database design which I will break down in the text that follows.

I have decided to continue forward with a relational database. Despite some changes to the database and service layers, I think a relational database is still the best option to support Music Theory Query. A tinyint column named “isActive” has been added to the resource tables to support a minor redesign at my peer reviewer’s suggestion. This column will act like an on/off switch for the customizable homepage’s population.



Summary

The focal point of my application is the ability to customize a home page with music theory resources for musicians to practice with. These resources therefore need to be stored in the database so that they may be accessed when necessary. To that end I have designed a generic resource table which stores all ResourceID values. These IDs will be used to make sure the appropriate resource is returned when the user wants it on their home page. Connected via the primary key of the resource table are several subcategories which will hold the resources themselves. These contain the name of the resource and the data to be returned to the application. While most of these will be images, if I am able to quickly deliver the MVP I may also allow for more complicated tools to be added to the home page as well.

The user account table will store user data and links to a bridge table between itself and the resource table. After my redesign, the "Custom Home Page" table no longer serves an integral purpose for the MVP, however I am going to leave it in the design as I may use it to implement stretch features such as sorting/reordering the homepage.

Example Scenario

When a user wants to add a resource to their home page, they will use a button associated with the resource. The button will call a function which will grab the resource's ID and pass it through the API to query the database. The resource will be found via its ID, its "isActive" value will be set to 1, and it will be inserted into the homepage via the onload function the next time it is visited.