Malcolm Howard

SWDV 691

Database Design: (ShineZone)

For the ShineZone application I have chosen to go with an MySQL setup to meet the requirements for the MVP. MySQL quick start capability is the reason I chose to with it. It's reliable, cost-effective, efficient, and has a self-management system that allows me to focus on other necessities of the application. The MVP was dialed down a bit to meet time constraints so the MVP now consists of:

- Fully functioning game
- Ability to see leaderboards
- Ability to choose game mode

The MySQL database will be used with AWS services managed with MySQL Workbench. The database (ShineZone) would consist of one table entitled "Leaderboards". This table would be used to capture the information from the user after completion of the game to add to the leaderboard view. The goal is to at least show the top 10-20 top scores. The database table script is listed below (Figure 1).

Data Structure

The original structure is updated to incorporate user data, stored in the users table. The users table has a one to many relationship with the leaderboard table since the user could appear many times on the leaderboard. The tables are linked by the UserID, which acts as the FK in the Leaderboards table and the PK in the Users table.

Stretch features would be tricky to manage, but I believe just requires a bit more planning. The questions provided for the app are done via XML which would eliminate the need for storing them in the database. The problem with this is that the stretch features would become complicated to implement since changes to the XML file would need to happen to update changes in the app. With further planning this may be an easy fix. I believe building a user interface for instructors could be beneficial to allow custom levels to added; yet, this still would require the XML to be overwritten on save.

Table values are most important for the leaderboard to view correctly (game_mode, score). These would be used to query the results in descending order to ensure the data board is sorted with highest points scored first. The date timestamp is for admin purposes only so far.

This database would be only accessible to the private security group set in AWS. The database is small in structure, but would be contain a large set of data. The database would also be able to be upgraded to meet the requirements of new features as well.

Key note: Database use for now is strictly for the leaderboard functionality

(Figure 1.)

```
CREATE TABLE top_scores (
        ID INT AUTO_INCREMENT PRIMARY KEY,
        userID FK KEY,
        gamemode VARCHAR,
        score INT UNSIGNED NOT NULL DEFAULT 0,
        date TIMESTAMP DEFAULT CURRENT_TIMESTAMP

CREATE TABLE users(
        userID INT AUTO_INCREMENT PRIMARY KEY,
        firstname VARCHAR NOT NULL,
        lastname VARCHAR NOT NULL,
        email VARCHAR,
        username VARCHAR NOT NULL,
        password VARCHAR NOT NULL,
```

STRETCH FEATURES (Requires more planning)

```
CREATE TABLE instructors(
        instructorID INT AUTO_INCREMENT PRIMARY KEY,
        instructor_FN VARCHAR NOT NULL,
        instructor_LN VARCHAR NOT NULL,
        instructor_email VARCHAR,
        i_username VARCHAR NOT NULL,
        i_password VARCHAR NOT NULL,

CREATE TABLE category(
        categoryID INT AUTO_INCREMENT PRIMARY KEY,
        game_mode VARCHAR NOT NULL,

CREATE TABLE game(
        gameID INT AUTO_INCREMENT PRIMARY KEY,
        gamename VARCHAR NOT NULL,
        categoryID fk
```