

Part B: Database Design

David Norton 3616902

Tye Shuttly 3605045

CS3103 Rick Wightman

March 14, 2021

Project idea	1
Some user stories to help guide this assignment:	1
ERD	2
Dictionary of stored procedures (names only) and their functionality:	2
Create	3
Read	3
Update	3
Delete	3

Project idea

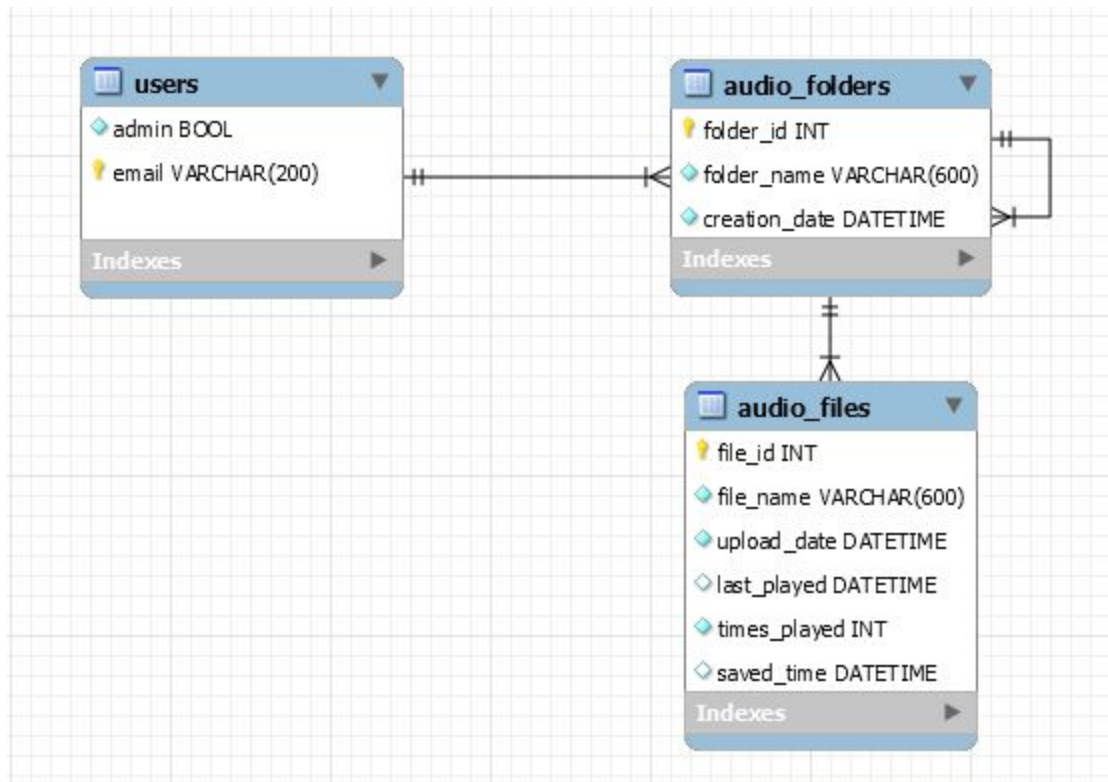
A personal online audio storage/streaming service.

Some user stories to help guide this assignment:

- Create personal account
- Create admin account
- Log into account
- Change email
- Create folders
- Change folder name
- Organize folders into folders
- Upload audio files
- Download audio files
- View all uploaded audio files
- Change audio file name
- Organize audio files into groups (folders)
- Delete audio files
- Delete account
- Delete other accounts (admin)

- Delete other users' data (admin)
- Delete other users' files (admin)
- View number of times file played
- View number of times file was downloaded/saved
- View upload date of file
- View creation date of folder
- Play and pause audio file in browser with volume control

ERD



Notes:

- The file_name includes the extension indicating format
- The path to each file is derived from the user email, parent folders/folders, and file_name
- audio files will not be stored in a database, just their metadata
- Each user has a root folder, and the root folder has child folders, and so on.
- The UNB CS system will store username and password used to access many parts of the system

Dictionary of stored procedures (names only) and their functionality:

(All access to the database will be restricted to stored procedures so there will be quite a few.)

Create

- `createUser`: given email, creates a user for the audio platform, also creates root folder with default name and description
- `createFolder`: given folder ID, folder name, and folder description, creates a folder with that name under that folder. `creation_date` is automatically populated.
- `createFile`: given file name, file description, and folder ID, adds a file with that metadata and a foreign key to that folder. `times_played` is set to 0 and the current system time is set as `upload_date`.

Read

- `readFolderContents` : given folder ID, returns all immediate child file and folder IDs and names
- `readFolderTree` : given user email, return tree of all subfolder names and folder IDs
- `searchFolderName` : given folder substring and user email, return all matching user folder names with folder ID
- `readUserEmail` : given user email, returns admin status
- `readUserAdmin` : given admin status, returns all user emails with that status (non admin can only search for admins)
- `readFolderDescription` : given folder ID, returns folder name and descriptive text (such as folder genres, number of songs)
- `readFileDescription` : given file ID, returns file name and descriptive text (such as genre, length, popularity)
- `searchFileName` : given file substring and user email, return all matching user file names with file ID
- `searchUniqueEmail` : given an email, returns true if unique, else false. Helps when signing up.

Update

- `updateFolderDescription` : given folder ID and text, replaces description and name
- `updateFileDescription` : given file ID and text, replaces description and name
- `updateFolderParent` : given folder ID, changes folder parent to given existing folder
- `updateFilePlayed` : given file ID, increments `times_played` and timestamp
- `updateUserEmail` : given user email, changes email to given email
- `updateUserAdmin` : given user email, changes admin status of given user (only available to admins)

Delete

- `deleteFolder` : given folder ID, deletes given folder and all contents, except the root folder
- `deleteUser` : given user email, deletes user records and all user folders and files, including root folder
- `deleteFile` : given file ID, the audio file will be removed