Design: Database

By Gary Deken

Department of Computer Science, Maryville University

SWDV 691: SOFTWARE DEVELOPMENT CAPSTONE 1W 25/SP2

Professor: Joseph Gradecki

March 21, 2025

Design: Database

Overview

I will be using a relational database, either Maria DB or My SQL, for my website. I am choosing this type of database due to the interconnectivity of each element of data. The Clubs need to connect to the types of clubs, so that users can search over them. I will need the Events to connect to the Clubs which would connect to a user, for displaying current events based on the day. There are several many-to-many relationships in the data.

Users will interact with the web frontend, which in turn will connect to a service layer for talking with the database. The frontend will communicate through REST calls. Each REST call will return data specific to a users action. The web frontend will not make calls into the database directly, only going through the REST service layer.

Data specifications

Table structure and relationships between tables is in the Entity Relationship diagram below. All tables have Audit field for the Created date and time.

Database accounts

The application will have an account to select, delete, insert and update. The username is HobbyClubApp.

Clubs

The clubs table is the main table, storing the Hobby Clubs. It contains the club name, description, and the category and state associated with it. A Club will have only one category and one state, but a category can be associated to many Clubs and a state can be associated to many clubs. The Clubs will also be associated to the Account of the user that created it, and its creation time. Clubs also have an active column to state if a club is active or deleted.

Accounts

The Accounts table stores information about each user including the name, state and city. It will join to the State table as a one-to-many relationship, and the User_Interests as a one-to-many relationship. As members join clubs it will connect to the Clubs table in a many-to-many relationship. When users post and replay, the account will join to those in a one-to-many relationship. When users create events and mark they will attend will join to the Events table in a one-to-many relationship. Accounts also have an active column to state if the user is active of deleted.

Categories

The Categories table stores dynamic list of club categories. It can be added to by request from users. It joins to the Clubs table in a one-to-many relationship, and it joins to the User_Interest as a many-to-many relationship to the Accounts table.

Events

The Events table stores information about a planned event. It will have a title, description, a date and time, who organized it, and optional City and State. There can be many Events to on Club, and an Event can be in only one City and State.

Event_Attendance

The Event_Attendance stores the users that have marked their intentions about an event. An event can have multiple Event_Attendance entries, and many Accounts. A single user can only mark their intentions once. The attending column stores a constant for their intentions. The following are valid intentions: WA = Will Attend; MA = May Attend; IE = Interested in event and NI = Not Interested. These constant values are not abbreviated on the frontend, but spelled out.

Topics

The Topics table stores users posts on a Clubs home page. It is linked to a single Club and a single Account. The title is limited to fifty characters and the details are limited to three hundred characters.

Replies

The Replies table stores the users' reply to a given Topic. A Topic can have many replies, but a Reply is only associated with one Topic and one user. The message of the Reply is limited to three hundred characters.

User_Interest

The User_Interest table is the many-to-many relationship between an Account and Categories table. It is used a help users find clubs they may be interested in.

User_Clubs

The User_Clubs table is a many-to-many relationship between an Account and Clubs, where a user can join many clubs and a club can have many users associated with it.

State

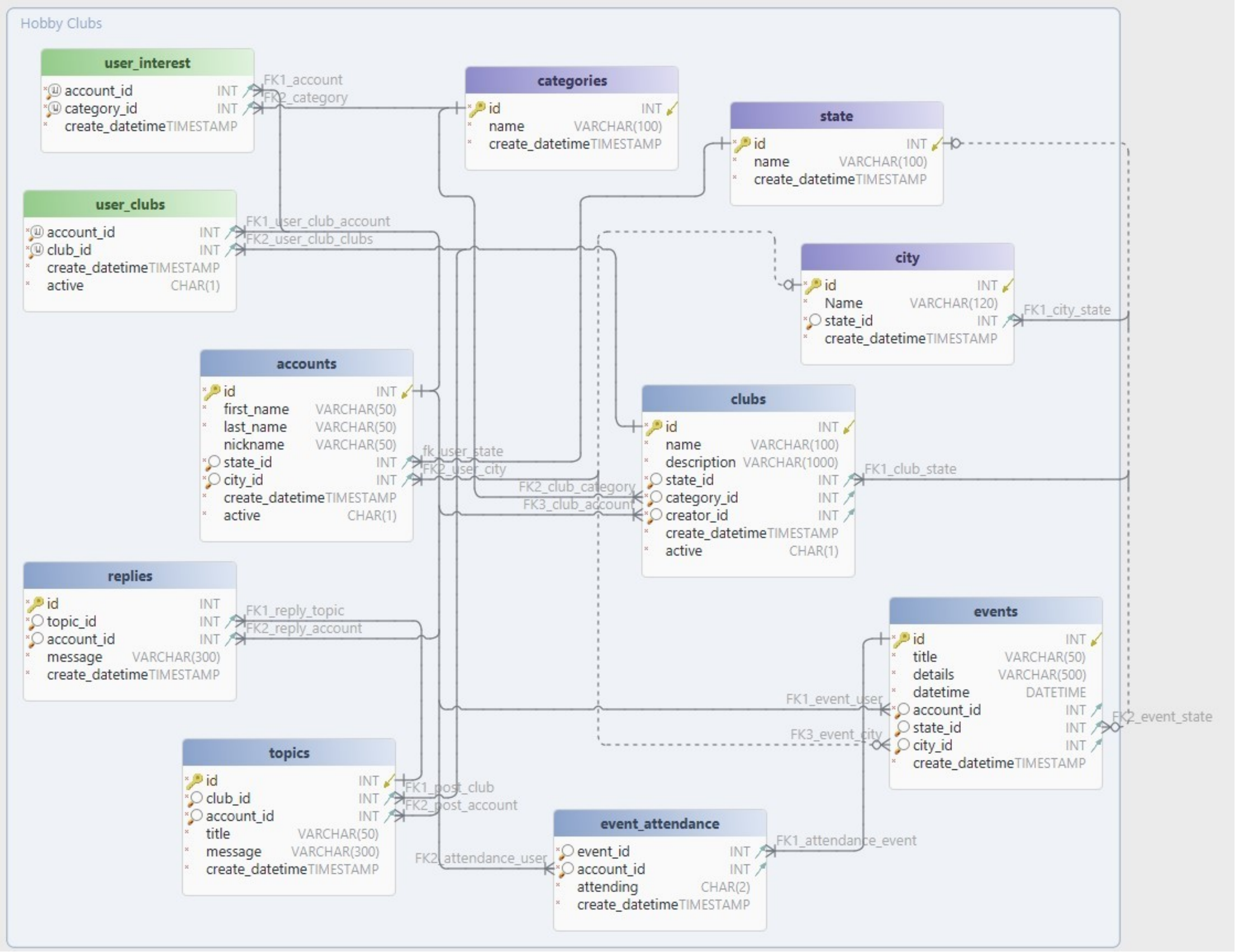
The State table is a constant listing of all fifty US states. It joins to the Accounts and Clubs tables in a one-to-many relationship.

City

The city table is a dynamical list of city for a state. It will start out with major cities for each state. It joins to the State table with one state to many cities. On a larger scale, the cities will have associations with other cities, but that is out of scope for the MVP.

Entity Relationship Diagram

The main data is color in blue. Constant data is colored in purple, and green tables represent connecting tables. The foreign key names are listed on the diagram connectors.



Hobby Clubs

user_interest	
account_id	INT
category_id	INT
create_datetime	TIMESTAMP

user_clubs	
account_id	INT
club_id	INT
create_datetime	TIMESTAMP
active	CHAR(1)

categories	
id	INT
name	VARCHAR(100)
create_datetime	TIMESTAMP

state	
id	INT
name	VARCHAR(100)
create_datetime	TIMESTAMP

city	
id	INT
Name	VARCHAR(120)
state_id	INT
create_datetime	TIMESTAMP

accounts	
id	INT
first_name	VARCHAR(50)
last_name	VARCHAR(50)
nickname	VARCHAR(50)
state_id	INT
city_id	INT
create_datetime	TIMESTAMP
active	CHAR(1)

clubs	
id	INT
name	VARCHAR(100)
description	VARCHAR(1000)
state_id	INT
category_id	INT
creator_id	INT
create_datetime	TIMESTAMP
active	CHAR(1)

replies	
id	INT
topic_id	INT
account_id	INT
message	VARCHAR(300)
create_datetime	TIMESTAMP

events	
id	INT
title	VARCHAR(50)
details	VARCHAR(500)
datetime	DATETIME
account_id	INT
state_id	INT
city_id	INT
create_datetime	TIMESTAMP

topics	
id	INT
club_id	INT
account_id	INT
title	VARCHAR(50)
message	VARCHAR(300)
create_datetime	TIMESTAMP

event_attendance	
event_id	INT
account_id	INT
attending	CHAR(2)
create_datetime	TIMESTAMP