

Health and Fitness Club

COMP 3005 Project Report

Group members:

Connor McDougall (SN 101179300)

Oliver Przednowek (SN 101240982)

Submitted to:

Professor Ahmed El-Roby and Abdelghny Orogat

GitHub Link:

<https://github.com/CGMcDougall/3005-Database-GP>

Youtube Link:

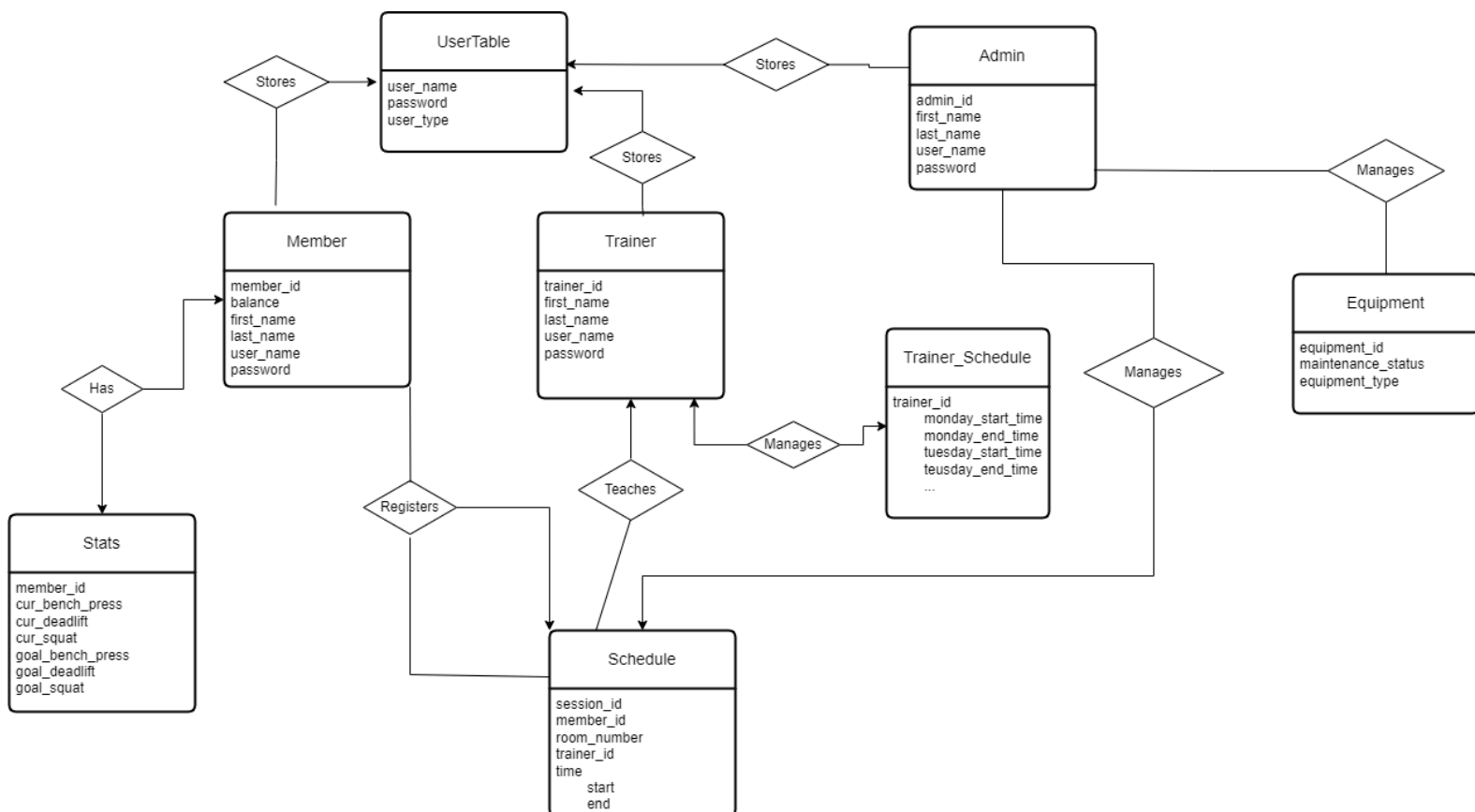
https://youtu.be/Q_5Gos1vXb8

Conceptual design

The database servers to store and manage the information regarding Some gym™, and the information associated with its members, trainers, admin, events, and more.

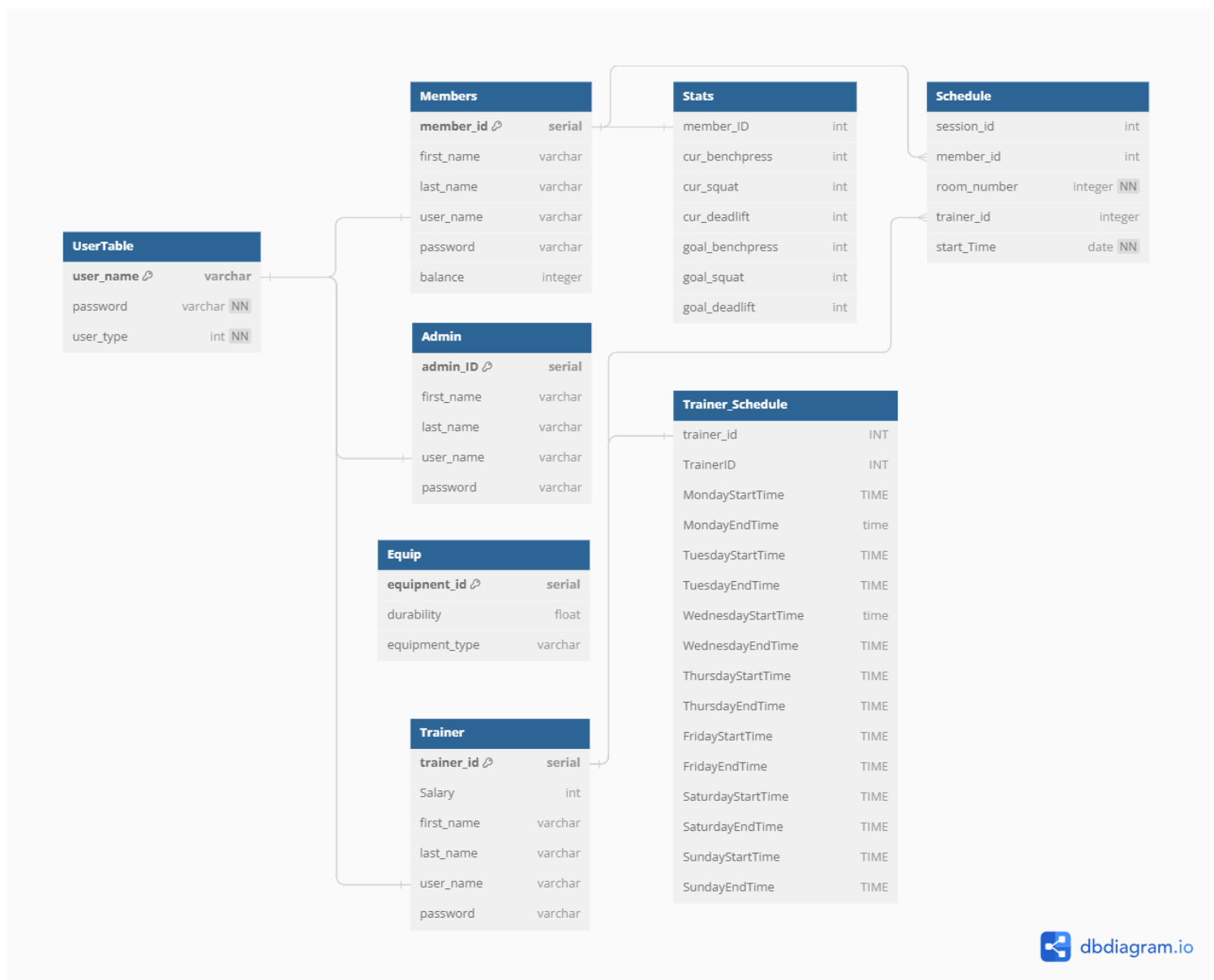
The database stores 3 separate tables for the 3 types of users on the system, members, trainers, and admin (all named after their type respectively). All users are also stored in a userTable, which ensures that the database can enforce unique usernames. The supplementary data of stats, trainer_schedule and equipment, are all stored in their respective tables, and are referenced using the id variables in the 3 user tables. There is a stats table that every member has a reference to which stores their gym goals and progress; a trainer_schedule that dictates availability which the trainer controls, and an equipment table that admin manages. Finally there is also a Schedule table that all users interact with. The schedule table manages and stores session/class information, and uses a sessionID and memberID together to form a primary key.

Members can enroll, create and drop from sessions as they please, trainers teach the sessions based upon their availability (trainer_schedule) and admins have the ability to modify any and all sessions in schedule as they see fit.



This is the ER model that represents the database. UserTable stores one to many for each of the three user types. Members have a 1-1 relation with stats, as each member has one set of stats. Trainers have 1-1 relation with trainer_schedule as each trainer manages their own weekly schedule. Admin and equipment is many to many because there is a possibility for multiple admin, and there are many pieces of equipment. Multiple members can register for group or private sessions, of which trainers teach several. Admins manage the schedule, making that relation a many to one.

Reduction to Relation Schema



Implementation

Built using Java and JDBC to connect to a PostgreSQL database, this application serves as a scheduling management system for a Health and Fitness Club. Data such as class scheduling, member, trainer, administrative staff login information, and fitness equipment maintenance status are all stored in a PostgreSQL relational database. This application implements the model-view-controller (MVC) architectural design pattern to organize the code. It also implements inheritance, having an abstract User class which Admin, Member, and Trainer all inherit from. Admin, Member, Trainer, Controller and SQLManager are all control classes. Each subclass of User implements the functionality that its respective user type needs to have access to. The Controller class controls the high level functionality of the program, delegating functionality to the different user types as necessary based on user type and input obtained through the View class. The View class handles the “view” portion of the MVC pattern, which is controlled through the Controller class. It implements a simple, easy to use Command Line Interface to allow the user to seamlessly interact with the application’s features. To handle database connectivity, all functionality related to interacting with the PostgreSQL database is encapsulated within the SQLManager class. This ensures that the application is quickly and easily scalable in the future if needed. Session is an “entity” class which stores data related to fitness sessions as stored in the Schedule table in the database. SessionList is a container class which inherits from ArrayList to provide operations to be done on the schedule as a whole. TrainerSchedule fulfills a similar purpose to SessionList, except it provides operations on a trainer’s individual schedule.