Assignment 3

Sarpreet Singh Buttar

Songho Lee

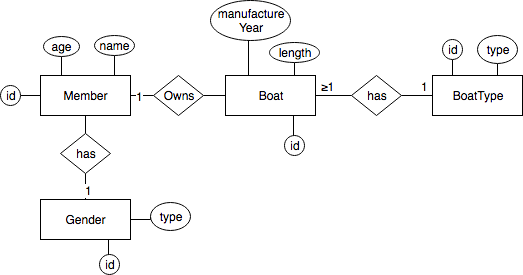
For the third assignment, we chose to create a management system for a boat club. The intended user of our programme is an administrator who wishes to manage a list of members and boats of a boat club. Our solution is suitable for a manager of a club as we provide a user-friendly web interface to add and edit members and boats. Besides, the user can search members and boasts in certain criteria. Therefore, the solution fulfils following:

# 1. Requirements

1. An admin can perform CRUD functionalities.
2. An admin can also perform search such as
   1. Get all members
      1. Get members by name
      2. Get members by gender
      3. Get members by age [≤, ≥, =, >, <]
   2. Get all boats
      1. Get all boats by length [≤, ≥, =, >, <]
      2. Get all boats by manufacture year.
      3. Get all boats by its type.
   3. Get all boats of a specific member
   4. Get a member from a specific boat.

We chose to use MySQL as a database management system, and programming language as JavaScript with React. You can find instructions on how to execute the project programme in a readme file.

# 2.Logical Model



## Entities and Attributes

A database requires two major entities which are Member and Boat, as we are developing the system for these. We decided to include age, name attributes to identify a member, and for boats, we have its manufacture year, length as attributes.

We could treat gender of a member and type of a boat as attributes as well, but we decided to have these as separate entities. For boat type, we want to manage various sort of boats and having it as a separate entity eases updating and removing it, which several boats may already belong to. In case of the gender of a member, it would normally be treated in binary or short string form such as F/M. However, having it as a separate entity provides more flexibility of managing diverse type, such as gender identity. Furthermore, making these in entity sets makes the model fulfil normalised form

## Relations

## A single member can have many boats, but one boat is only allowed to have one owner in our design. A boat only has one boat type, but each boat types have various boats. A member has only one gender, and cannot belong to different genders simultaneously. If that is the case, the new type of gender can be defined instead. Several members can belong to a gender type.

# 3.SQL Design

We translate our E/R diagram in following tables. We denote primary keys in underlined form and foreign keys with asterisk mark.

|  |
| --- |
| member(id, name,age,gender)  gender(id, type)  boat(id, year, length, member\_id\*, type\_id\*)  boat\_type(id,type) |

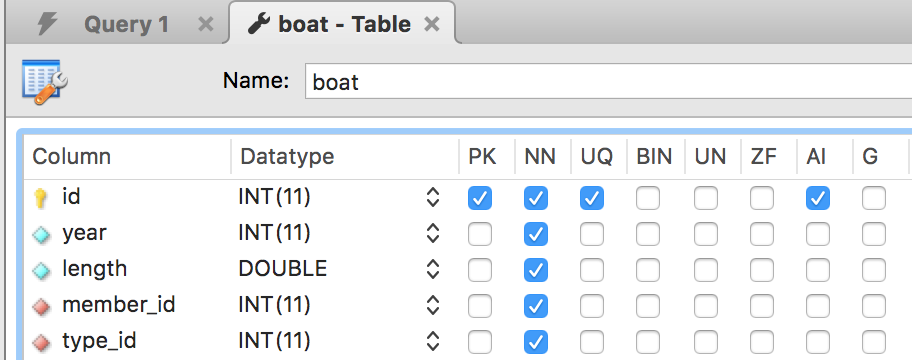


Figure 1 Boat table

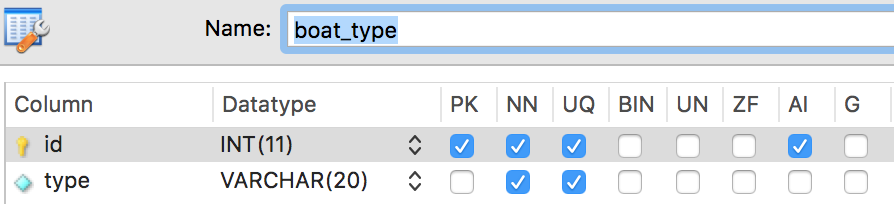


Figure 2 Boat type table

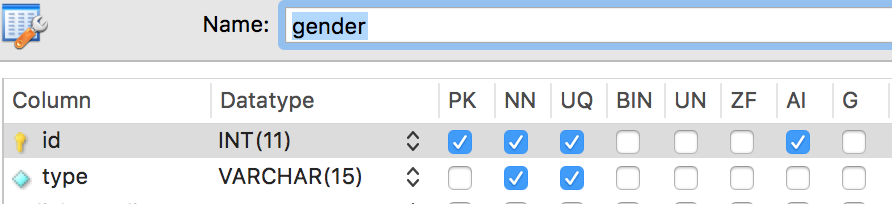


Figure 3 Gender table

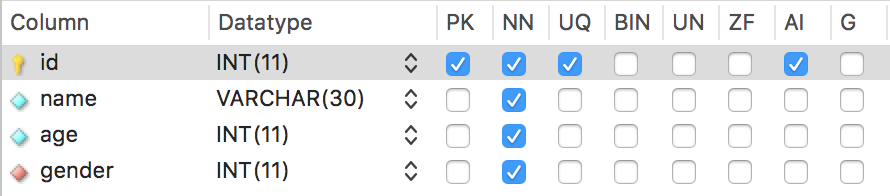


Figure Member table

The red square indicates that these are foreign key. We applied to cascade on updates and restrict on deletion.

# 4.SQL Queries

table