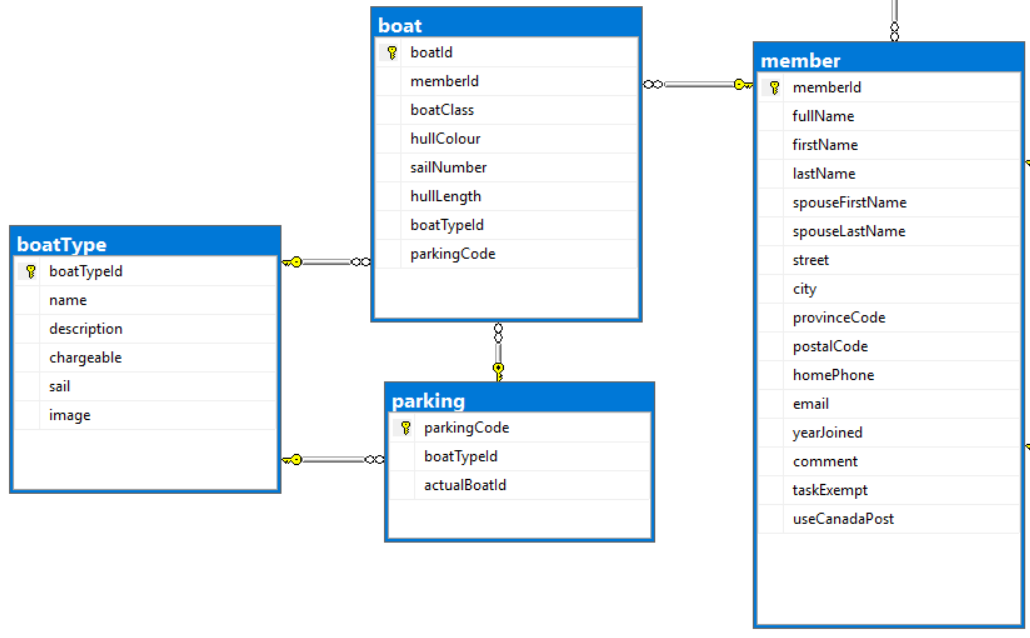
## Ms Web Tech Assignment 3 – Filtering Dropdowns

In this assignment, you’ll create pages to register boats for each member and assign them to appropriate parking spaces. A member will have two or more boats (one is the “no fertilizer” control). Each boat can have zero or more fertilizers … occasionally, a stock fertilizer mix needs to be augmented, like extra nitrogen, because of the soil analysis. After formulating the boats, you have to climb back up the reference chain.



### Project Setup

1. You can continue using your ***XXSail*** project from previous assignments

### XXMemberController

1. Add a *manage boats* hyperlink to each entry in the Index listing, passing the *memberId* to the *XXBoat* controller. You might want to pass the full name as well … member 250 doesn’t help anyone.

#### XXBoatController

1. Generate a controller for the *boat* table, called ***XXBoatController***, with full CRUD support and views. Ensure all pages work.
2. In the *Index* action:
   1. If the *memberId* is in the URL or a QueryString variable, save it to a cookie or session variable.
   2. If no *memberId* was passed in the URL or QueryString, see if there’s a cookie or session variable with it.
      1. If there’s no *memberId* in the cookie or session variables either, return to the member controller with a message asking them to select a member to see their boats.
   3. Regardless of where you got the *memberId* from, use it to filter the boat list to only show boats on file for that member, ordered by their *class*.
   4. If the member’s full name was passed as well, persist it the same way as memberId … otherwise, take the memberId, fetch the record from the member table, then extract & persist the full name.
   5. In the Index listing:
      1. Show the boat type’s name, not the boatTypeId.
      2. Do not list anything about the member … the heading (later) should suffice.
3. For the Create and Edit views:
   1. Do not show an input field for *memberId*: the user has requested boats for a specific member, so stay in that mindset.
      1. The *memberId* field still has to post back for new or updated records, so hide it on the view.
   2. The boat type drop-down should display the type *name* (sorted) and return the *boatTypeId*.
   3. For the *parkingCode* drop-down, sort & filter its entries:
      1. If the boat’s *parkingCode* is empty or null, list all parking spaces not currently assigned to a boat, plus an empty entry.
      2. Otherwise, list only parking spaces of the same type as the boat:
         1. Small boats can be towed by hand … with no standing rigging, they are parked with their mast removed, meaning they can be parked close together under the trees. The other types are stowed fully rigged. Catamarans are 8-10’ wide, so they need wider spaces but most can be towed by two people, using a 12’ wheeled tow-bar. Cruisers and over-17 boats need enough space to manoeuvre a tow vehicle.
4. In all views for this controller:
   1. Modify the browser title and page heading to indicate its purpose and the member it’s for, like “Boats for Chesterfield, Stanley & Mabel”, “Add Boat for Chesterfield, Stanley & Mabel”, and so on.

#### XXParkingController

1. Generate a controller for the *parking* table, called ***XXParkingController***, with full CRUD support and views. Ensure all pages work.
2. The Index view should be ordered by *parkingCode*.
3. On Create & Edit, the *boatTypeId* dropdown should display the type *name* (ordered) and return the *boatTypeId.*

### Hand In

1. Zip and upload your project folder to the drop box (course tools 🡪 assignments) for this assignment.