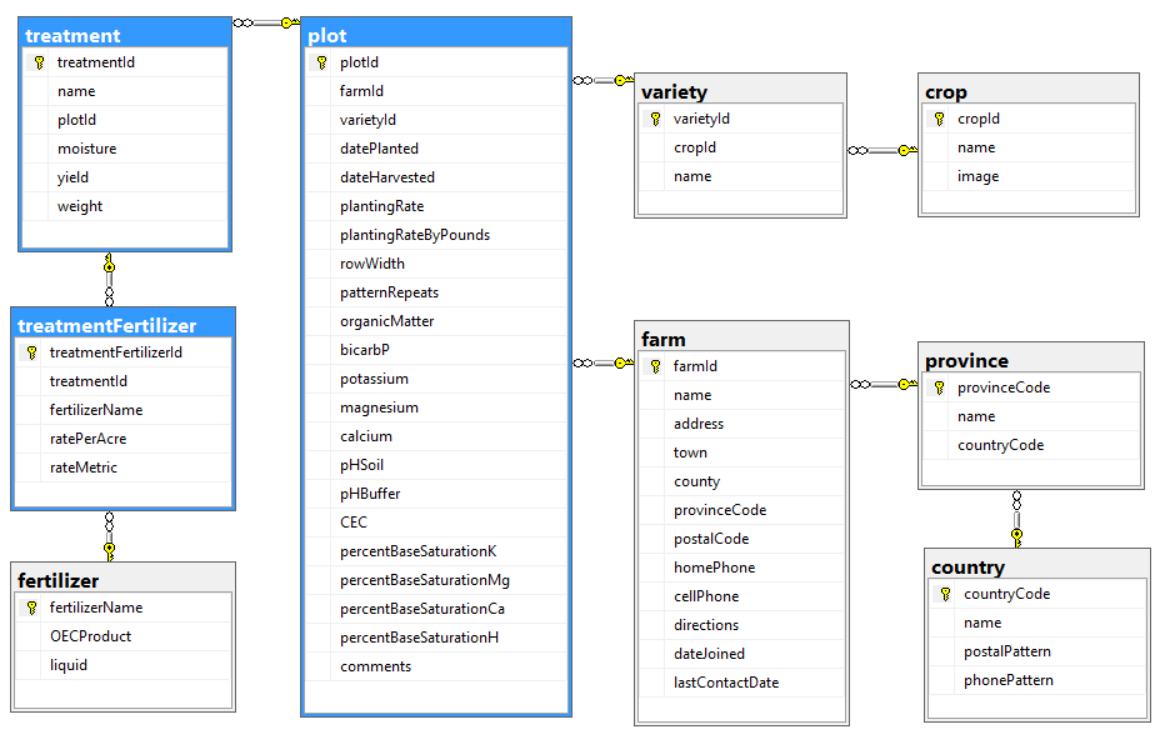
## Ms Web Tech Assignment 3 – Multiple Relations

In this assignment, you’ll manage the fertilizer treatments used in a test plot. A plot will have two or more treatments (one is the “no fertilizer” control). Each treatment 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 treatments, you have to climb back up the reference chain.



### Project Setup

1. You can continue using your ***XXOEC*** project from assignment 2 or you can unzip & use the ***a2OEC*** project from your instructor’s site … it’ll be available a week after assignment 2 was due.
   1. If using the ***a2OEC*** project, add your name and section to the footer

### XXPlotController

1. Add a *manage treatments* hyperlink to each entry in the Index listing, passing the plotId to the XXTreatment controller. You might want to pass the farm name as well … plot 3287 doesn’t help anyone.
2. In the Index action, add code to look for a passed plotId to the criteria search code that should already be there.
   1. If found, persist it as the new filter criteria and limit the listing to the selected plot (on initial and subsequent loadings).
   2. If not found, look for and go with the criteria in assignment 2, or default to all-plots-on-file if no criteria has been passed or persisted.

#### XXTreatmentController

1. Generate a controller for the *treatment* table, called ***XXTreatmentyController***, with full CRUD support and views. Make sure that create, edit, delete and details pages all work.
2. In the *Index* action:
   1. If the *plotId* is in the URL or a QueryString variable, save it to a cookie or session variable.
   2. If no *plotId* was passed in the URL or QueryString, see if there’s a cookie or session variable with it.
      1. If there’s no *plotId* in the cookie or session variables either, return to the ***XXPlot*** controller with a message asking them to select a plot to see its treatments.
   3. If the plotId was passed to this action, continue to step d. If you retrieved a persisted value, you’re possibly returning from an action that modified a treatment. You now need to derive the treatment names from the fertilizers they use … so, for each treatment in this plot:
      1. Retrieve the treatment record, with its treatmentFertilizer relations and their related fertilizer.
      2. Extract the fertilizer names into a list, and sort the list alphabetically
      3. Concatenate the sorted fertilizer names, putting “ + “ between them, into the *name* field on the treatment record. If there are no treatmentFertilizer records for this treatment, put “no fertilizer” in as its name.
      4. Call your post-back Edit action, passing it the treatment record with the updated name and whatever else it needs.
   4. Regardless of where you got the *plotId* from, use it to filter the treatment list to only show treatments on file for that plot, ordered by the treatment *name*.
   5. If the farm name was passed as well, persist it the same way as plotId … otherwise, take the plotId, fetch the record from the plot table (including its farm record) then extract & persist the farm name.
   6. In the Index listing:
      1. Make the treatment *name* a hyperlink that passes the *treatmentId* to the *XXTreatmentFertilizer* controller’s index view.
      2. Do not show anything about the plot … the heading should suffice (later).
   7. At the bottom of the listing, **provide a link that passes the *plotId* back to your *XXPlot* controller**.
   8. In the Create and Edit views, do not show an input field for *plotId* … the user has requested treatments for a specific plot, so stay in that mindset. When posted back, the *plotId* field still has to be inserted into the new or modified record, so hide it on the views.
3. In each view for this controller:
   1. Modify each browser title and page heading to indicate its purpose and the plot it’s for, like “Treatments for plot at Glenview ACHErs”, “Add Treatment for plot at Donnybrook”, and so on.

#### XXTreatmentFertilizerController

1. In the index action & view:
   1. If the *treatmentId* is in the URL or a QueryString variable, save it to a cookie or session variable.
      1. If no *treatmentId* was passed in the URL or QueryString, check for a cookie or session variable with it.
      2. If there’s no *treatmentId* in the cookie or session variables either, return to the ***XXTreatment*** controller with a message asking them to select a treatment to see its fertilizer composition.
   2. Regardless of where you got the *treatmentId* from, use it to filter the treatmentFertilizer list to only show fertilizers on file for that treatment, ordered by the fertilizer *name*.
   3. Add a hyperlink to the bottom of the Index listing returning to the XXTreatment controller’s Index action, without passing anything … it’ll figure out where you’re coming from.
2. On all pages:
   1. Do not show the treatmentId or name on the property listings or as input fields. However, Create and Edit … you know the drill.
   2. Order the fertilizer drop-downs by name.
   3. The *rateMetric* is either“Gal” or “LB”, depending on whether the fertilizer is liquid or not … so we know what the *ratePerAcre* means.

### Hand In

1. Zip and upload your project folder to the drop box (course tools 🡪 assignments) for this assignment.
2. During the class period, print and hand in the marking sheet (Content 🡪 assignments) with your name, section, and the instructor’s name on it. Outside of class, drop this into the slot in your instructor’s office door.