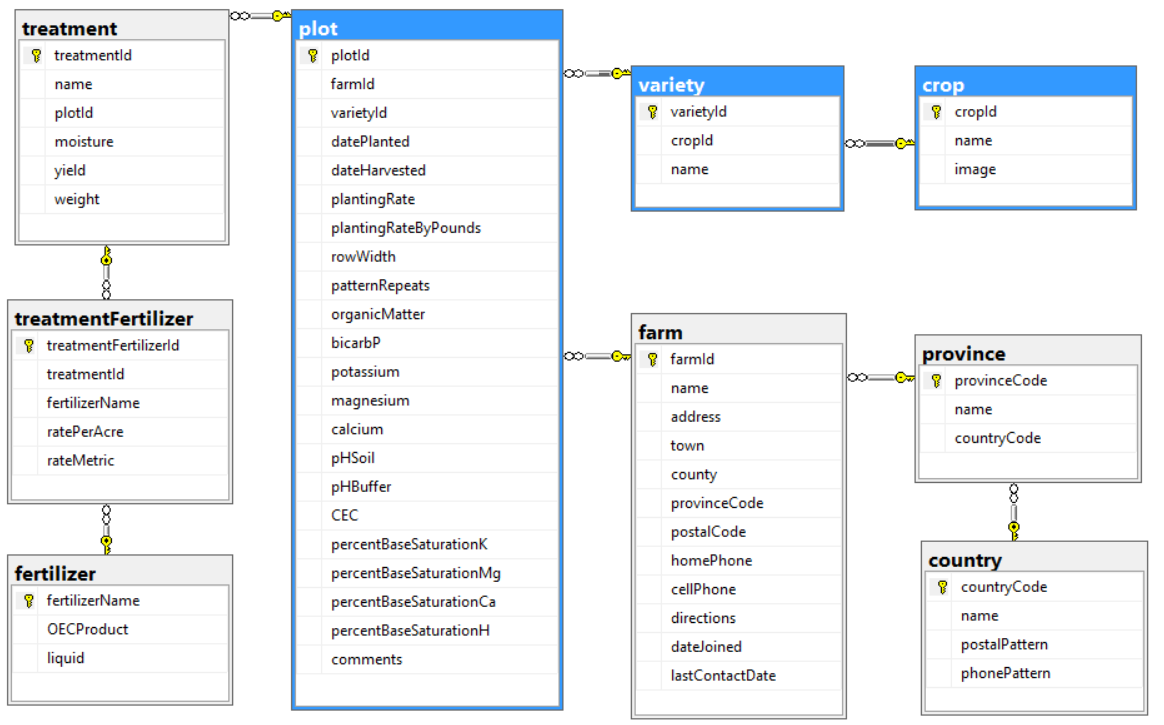
## Ms Web Tech Assignment – Persistence, sorting & filtering

In this assignment, you will pass & persist (remember) keys, to show all varieties for a selected crop, and all plots planted with a specific variety. As always, whenever you see “XX”, substitute your initials.



### Project Setup

1. ~~You can continue using your~~ ***~~XXOEC~~*** ~~project from assignment 1 or you can unzip & use the~~ ***~~a1OEC~~*** ~~project from your instructor’s site.~~
2. ~~On~~ ***~~~/Views/Shared/\_layout.cshtml~~***~~, display the TempData variable “message” before the guest page, bold and red.~~
   1. ~~If using the~~ ***~~a1OEC~~*** ~~project, add your name and section to the footer~~
3. ~~Put an agricultural-themed logo (image) in the header with a height of 100 pixels, and a better site heading than just “OEC”.~~
   1. ~~The logo will enlarge the banner on each page. The body of the guest page passes under the banner: modify the site CSS so no content like page headings or @TempData[“message”]) is overlaid.~~
4. ~~Add support for Session variables to your site’s Startup.cs~~

### XXCropController

1. ~~In the Index View:~~ 
   1. ~~Change the name of each crop into a hyperlink that passes its~~ *~~cropId~~* ~~to the~~ *~~XXVariety~~* ~~controller’s Index action. You might also want to pass the crop’s name.~~
   2. ~~Add a “show plots” hyperlink to each line which passes the~~ *~~cropId~~* ~~and crop~~ *~~name~~* ~~as QueryString variables to the XXPlot controller’s Index action. This is to list all plots for the selected crop, regardless of the variety.~~
   3. ~~Ensure the crop listing is in order by~~ *~~name.~~*

### XXVarietyController

1. ~~Generate a controller for the~~ *~~variety~~* ~~table, called~~ ***~~XXVarietyController~~***~~, 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~~ *~~cropId~~* ~~is in the URL or a QueryString variable, save it to a cookie or session variable.~~
   2. ~~If there is no~~ *~~cropId~~* ~~passed in the URL or QueryString, see if there’s a cookie or session variable with it.~~
      1. ~~If there’s no~~ *~~cropId~~* ~~in the cookie or session variables either, return to the~~ ***~~XXCrop~~*** ~~controller with a message asking them to select a crop to see its varieties.~~
   3. ~~Regardless of where you got the~~ *~~cropId~~* ~~from, use it to filter the listing to only show varieties on file for that crop, ordered by the variety~~ *~~name~~*~~.~~
   4. ~~If the crop name was passed as well, persist it the same way as cropId … otherwise, take the cropId, fetch the record from the crop table, then extract & persist its name.~~
   5. ~~In the Index View’s listing:~~
      1. ~~Show just the variety~~ *~~name~~*~~, in ascending sequence.~~
      2. ~~Make the variety~~ *~~name~~* ~~a hyperlink that passes the~~ *~~varietyId~~* ~~and the variety~~ *~~name~~* ~~as QueryString variables to the~~ *~~XXPlot~~* ~~controller’s index view.~~
3. ~~In the Create and Edit views:~~ 
   1. ~~Do not show an input field for~~ *~~cropId~~* ~~… the user has requested varieties for a specific crop, so stay in that mindset.~~
   2. ~~When posted back, the~~ *~~cropId~~* ~~field still has to be inserted into the new/modified record, so hide it on the views. Warning: if a hidden field is not given a value, it posts back as an empty string, which kills numeric receiving parameters … even nullable ones.~~
4. ~~In each view for this controller:~~ 
   1. ~~Modify each browser title and page heading to indicate its purpose and the crop it’s for, like “Varieties of wheat, winter”, “Add a Corn Variety”, and so on … watchyourpunctuation.~~
   2. Do not show the *cropId* or crop *name* in the property listing area … just in the page headings.

### XXPlotController

A user may be searching for all plots for a specific farm, plots that used a specific crop variety, that used any variety of a specific crop, that used a specific fertilizer or fertilizer combination (treatment) … or that had particular soil characteristics. When this controller’s index action is called, it must identify which criteria was passed and persist the criteria type and its value, since both are needed to filter the listing. Subsequent calls must determine if the criteria was passed or if it must look for persisted criteria.

1. Modify the Index view:
   1. Restrict the listing to the following fields: farm name, crop name, variety name, date planted, CEC and the treatment names (slash or <br /> separated)
   2. Make the farm, variety and CEC column headings into hyperlinks that reorder the listing by that field.
2. In the Index action:
   1. Have the Index action identify which criteria it’s been provided with (cropId or varietyId for now), persist it with a criteria identifier (and the crop/variety name).
   2. If a criteria has been persisted, use it to restrict the listing to plots with that criteria
   3. If no criteria has been passed or persisted, display all plots on file.
   4. By default, sort the plot listing by date planted, most recent first.
   5. In all views, the page title and heading should identify the criteria type and value. You’ll also have to *.Include()* associated tables so you can display names, not keys.
3. The Create view should identify the selected criteria and default its input element (usually a drop-down) to the specified value. If the criteria is two tables away (like *cropId*), that should be used to filter the intervening table’s (variety’s) dropdown.
4. Sort all drop-downs (on Edit & Create) in ascending order by the displayed field. The displayed field should be text like name, not the numeric key.

### 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.