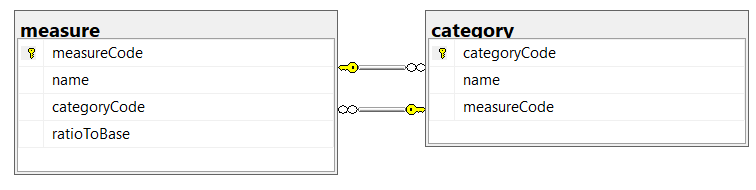
## PR: Microsoft Web Technologies – Mid-Term Exam

You are just starting a cooking site. The first tables you’re working with are for weights and measures … one, the ***Category*** table, lists the measurement categories: weight, length, volume, energy, etc. The other is the ***measure*** table, listing units of measure: grams, pounds, litres, etc. Each category has a base *measureCode* (such as gram for weight, litre for volume), and each measure has a *ratioToBase* number allowing it to be converted to the base measure for the category. For example: a gram is the base measure for weight, so it has a ratio of 1 … a pound (lb) is 453.592 grams, so this becomes its ratio to the base measure for weights.

1. You have an ASP.NET Core MVC project called ***Cooking*** and an SQL script called ***Cooking.sql***. **Only what is uploaded to “MidTerm - Exam Dropbox” (under Assignments) will be marked.**
   1. Use SQL Server Management Studio to generate the Cooking database from the SQL script.
   2. **It is your responsibility to upload the solution to Dropbox before the end time.**

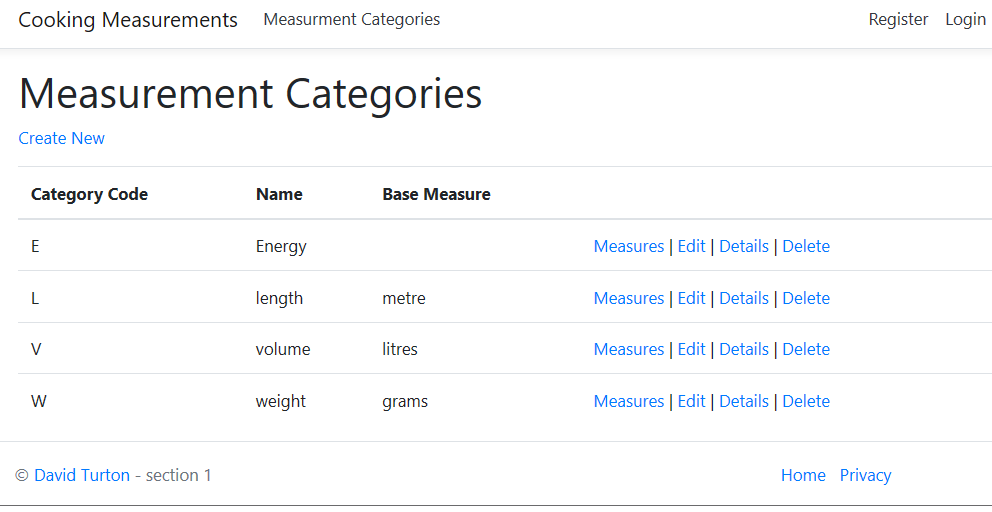


1. Modify the main layout, ***\_layout.cshtml***:
   1. Add a link to the ***Category*** controller in the menu
   2. Move the *Home* and *Privacy* menu items to the right side of the footer, aligned horizontally as shown.
   3. Display the TempData variable *message*, bold and red, before the main body of the guest page.
   4. Put your name and section in the footer, as an e-mail hyperlink to your college address.

### Category Controller

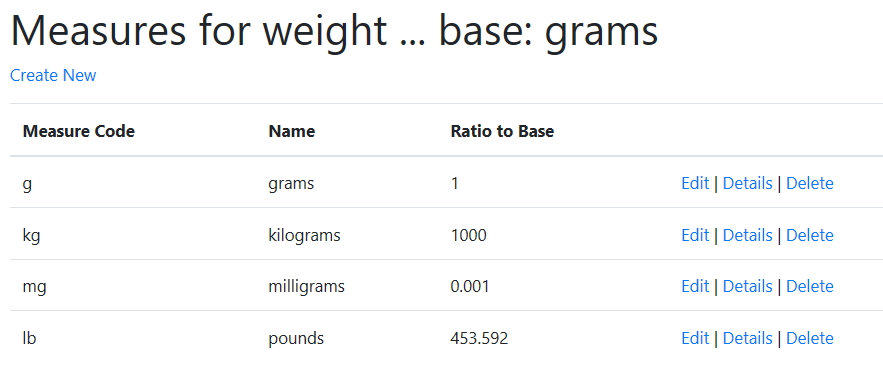
1. Generate the ***Category*** controller with full CRUD support for the *category* table. Modify its listing to be similar to the one shown below …
   1. The page title and main heading should be “Measurement Categories”.
   2. Sequence the categories by name
   3. Show only the columns in the example
   4. Add a hyperlink to each row that passes the *categoryCode* to the default page of the ***Measure*** controller.

This page was intentionally left blank

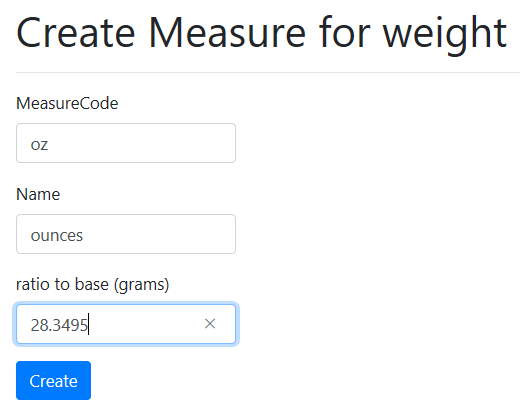
******

### *Measure* controller

1. Generate the ***Measure*** controller with full CRUD support for the *measure* table.
2. For the *Index* action and its view:
   1. If the *categoryCode* was passed in the URL or a QueryString, save it to a cookie or session variable.
   2. If the *categoryCode* is not in the QueryString/URL, look for it in a cookie or session variable … if it’s not there either, go back to the category listing and display a message requesting they select one.
   3. Limit the listing of *measure* records to the selected *categoryCode*, in sequence by name.
   4. Fetch the category record for the given categoryCode if there is a base measureCode, pass the name of that measure to the view. If not, pass “N/A” to the view.
   5. On the view:
      1. Change the page title and heading to “Measures for xx … base: yy”, where “xx” is the name of the selected category, and “yy” is the name of the category’s base measureCode, determined above.
      2. Modify the view’s fields to those shown on the sample below.



1. For the Create actions and their view:
   1. Modify the view’s fields to those shown on the sample below.
   2. Change the browser title and page heading to “Create Measure for xxx”, where “xxx” is the name of the selected measurement category.
   3. When the user clicks *Create*:
      1. Add the proper categoryCode to the new measure record & save it to the database
      2. If the new measure saved correctly and its ratioToBase is 1:
         1. Fetch the current category’s record.
         2. Change the current category’s base MeasureCode to the one the user just saved and write it back to the category table.
      3. If the save(s) worked correctly, display a message to that effect on the Measure index page.



|  |  |
| --- | --- |
| **\_layout**   * TempData variable displayed before guest page, red & bold * Menu link sends browser to index page of CategoryController * Student name & section in footer invokes e-mail client to college account * Home and Privacy hyperlinks moved to right side of footer, still aligned horizontally | \_\_\_\_\_\_\_\_/ 2  \_\_\_\_\_\_\_\_/ 1  \_\_\_\_\_\_\_\_/ 1  \_\_\_\_\_\_\_\_/ 1 |
| **Category Listing**   * Browser title & page heading correct, includes category name and base measure name * Listing shows all categories, in sequence by category name * Correct fields showing, including primary key and base measure name * Hyperlink on each line redirects user to Measure listing   + Link passes categoryCode | \_\_\_\_\_\_\_\_/ 3  \_\_\_\_\_\_\_\_/ 2  \_\_\_\_\_\_\_\_/ 2  \_\_\_\_\_\_\_\_/ 1  \_\_\_\_\_\_\_\_/ 1 |
| **Measure Listing**   * Check made for *categoryCode* in URL or a QueryString   + If found, is persisted to a cookie or session variable   + Otherwise, checks for a cookie or session variable with *categoryCode* & uses that     - If not found, returns to Category listing & displays message to pick a category * Browser title & page heading correct, includes category name * View shows only measures for the requested category, sequenced by name * View shows only fields specified | \_\_\_\_\_\_\_\_/ 1  \_\_\_\_\_\_\_\_/ 1  \_\_\_\_\_\_\_\_/ 1  \_\_\_\_\_\_\_\_/ 2  \_\_\_\_\_\_\_\_/ 2  \_\_\_\_\_\_\_\_/ 2  \_\_\_\_\_\_\_\_/ 2 |
| **Measure Create**   * Browse title & page heading correct, includes category name * View shows only fields specified * On save:   + CategoryCode added to record before writing   + Record saves correctly   + ratioTobase checked and, if it is 1:     - Category record for categoryCode fetched     - Base MeasureCode replaced     - Category record rewritten to database   + Message displayed on measure listing if measure successfully added | \_\_\_\_\_\_\_\_/ 2  \_\_\_\_\_\_\_\_/ 1  \_\_\_\_\_\_\_\_/ 1  \_\_\_\_\_\_\_\_/ 1  \_\_\_\_\_\_\_\_/ 1  \_\_\_\_\_\_\_\_/ 2  \_\_\_\_\_\_\_\_/ 2  \_\_\_\_\_\_\_\_/ 1  \_\_\_\_\_\_\_\_/ 1 |
| **Base Mark**  **Fixes for: build errors run errors @1/fix**  **Final Mark** | \_\_\_\_\_\_\_\_/ 37  - \_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_/ 37 |