# 2011

CODE ONTIME

COOKBOOK
Universal Lookup

Universal Lookup Database

Universal Lookup functionality allows extension of otherwise static database structures with custom fields, and simplifies maintenance of lookup records in a fixed set of tables. Below is a sample database diagram with a universal lookup. *Lookup* table in the middle has integer *LookupID*, string LookupText, and *LookupGroupName* field. Lookup records in the same group constitute a virtual lookup dataset.
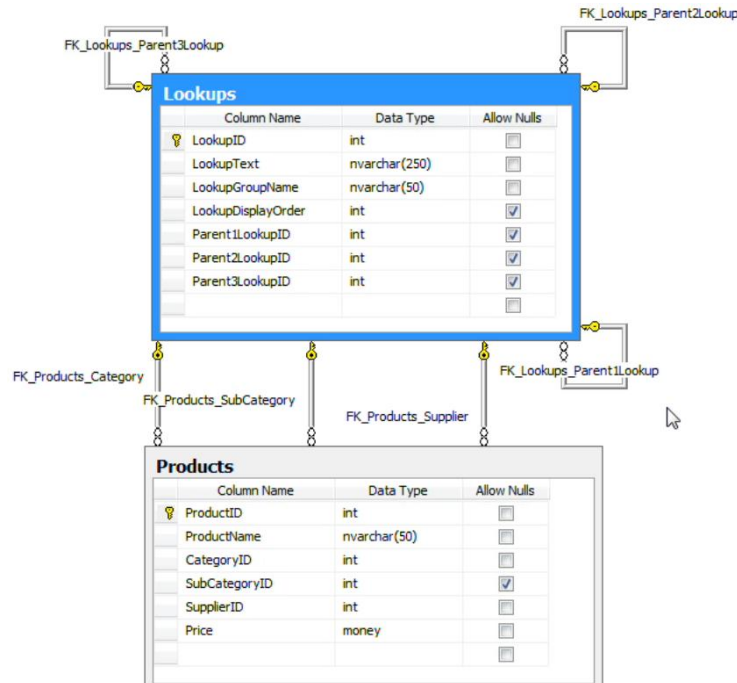
Below is a list of records available in this table. There are several records with the group *Categories*, some other ones with the groups *Sub Categories*, *Suppliers*, and *Comm Methods*.

LookupID	LookupText	LookupGroupN...	LookupDisplay...	Parent1LookupID	Parent2LookupID	Parent3LookupID
16	Category 1	Categories	1	NULL	NULL	NULL
17	Category 2	Categories	2	NULL	NULL	NULL
18	Category 3	Categories	3	NULL	NULL	NULL
19	Sub Category 1	Sub Categories	1	16	NULL	NULL
20	Sub Category 2	Sub Categories	2	16	NULL	NULL
21	Sub Category 3	Sub Categories	3	17	NULL	NULL
22	Sub Category 4	Sub Categories	4	18	NULL	NULL
23	Sub Category 5	Sub Categories	5	18	NULL	NULL
24	Sub Category 6	Sub Categories	6	18	NULL	NULL
25	Sub Category 7	Sub Categories	7	18	NULL	NULL
26	Sub Category 8	Sub Categories	8	18	NULL	NULL
27	Supplier 1	Suppliers	NULL	NULL	NULL	NULL
28	Supplier 2	Suppliers	NULL	NULL	NULL	NULL
29	Supplier 3	Suppliers	NULL	NULL	NULL	NULL
30	Comm Method 1	Comm Methods	NULL	NULL	NULL	NULL
31	Comm Method 2	Comm Methods	NULL	NULL	NULL	NULL
32	Comm Method 3	Comm Methods	NULL	NULL	NULL	NULL

The lookup table also has three self-referring keys, *Parent1LookupID*, *Parent2LookupID*, and *Parent3LookupID*. This can be used to refer to the category of a subcategory. You can see which subcategories belong to which category in the field list. The references in our example are under *Parent2LookupID* field.

The products table has *CategoryID*, *SubCategoryID*, and *SupplierID*, all of whom are referring to the same lookups table. You will have to set up foreign keys to allow *Code On Time Generator* to detect the relationship that the universal lookup table creates.

This script creates the database tables for *Microsoft SQL Server*.

```sql
USE [UniversalLookups]
GO
/****** Object:  Table [dbo].[Lookups] ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Lookups](
      [LookupID] [int] IDENTITY(1,1) NOT NULL,
      [LookupText] [nvarchar](250) NOT NULL,
      [LookupGroupName] [nvarchar](50) NOT NULL,
      [LookupDisplayOrder] [int] NULL,
      [Parent1LookupID] [int] NULL,
      [Parent2LookupID] [int] NULL,
      [Parent3LookupID] [int] NULL,
 CONSTRAINT [PK_Lookups] PRIMARY KEY CLUSTERED
(
      [LookupID] ASC
)WITH (PAD_INDEX  = OFF, STATISTICS_NORECOMPUTE  = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS  = ON, ALLOW_PAGE_LOCKS  = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/****** Object:  Table [dbo].[Products]    ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Products](
      [ProductID] [int] NOT NULL,
      [ProductName] [nvarchar](50) NOT NULL,
      [CategoryID] [int] NOT NULL,
      [SubCategoryID] [int] NULL,
      [SupplierID] [int] NOT NULL,
      [Price] [money] NOT NULL,
 CONSTRAINT [PK_Products] PRIMARY KEY CLUSTERED
(
      [ProductID] ASC
)WITH (PAD_INDEX  = OFF, STATISTICS_NORECOMPUTE  = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS  = ON, ALLOW_PAGE_LOCKS  = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/****** Object:  ForeignKey [FK_Lookups_Parent1Lookup]    ******/
```

```sql
ALTER TABLE [dbo].[Lookups]  WITH CHECK ADD  CONSTRAINT
[FK_Lookups_Parent1Lookup] FOREIGN KEY([Parent1LookupID])
REFERENCES [dbo].[Lookups] ([LookupID])
GO
ALTER TABLE [dbo].[Lookups] CHECK CONSTRAINT [FK_Lookups_Parent1Lookup]
GO
/****** Object:  ForeignKey [FK_Lookups_Parent2Lookup]    ******/
ALTER TABLE [dbo].[Lookups]  WITH CHECK ADD  CONSTRAINT
[FK_Lookups_Parent2Lookup] FOREIGN KEY([Parent2LookupID])
REFERENCES [dbo].[Lookups] ([LookupID])
GO
ALTER TABLE [dbo].[Lookups] CHECK CONSTRAINT [FK_Lookups_Parent2Lookup]
GO
/****** Object:  ForeignKey [FK_Lookups_Parent3Lookup]    ******/
ALTER TABLE [dbo].[Lookups]  WITH CHECK ADD  CONSTRAINT
[FK_Lookups_Parent3Lookup] FOREIGN KEY([Parent3LookupID])
REFERENCES [dbo].[Lookups] ([LookupID])
GO
ALTER TABLE [dbo].[Lookups] CHECK CONSTRAINT [FK_Lookups_Parent3Lookup]
GO
/****** Object:  ForeignKey [FK_Products_Category]    ******/
ALTER TABLE [dbo].[Products]  WITH CHECK ADD  CONSTRAINT
[FK_Products_Category] FOREIGN KEY([CategoryID])
REFERENCES [dbo].[Lookups] ([LookupID])
GO
ALTER TABLE [dbo].[Products] CHECK CONSTRAINT [FK_Products_Category]
GO
/****** Object:  ForeignKey [FK_Products_SubCategory]    ******/
ALTER TABLE [dbo].[Products]  WITH CHECK ADD  CONSTRAINT
[FK_Products_SubCategory] FOREIGN KEY([SubCategoryID])
REFERENCES [dbo].[Lookups] ([LookupID])
GO
ALTER TABLE [dbo].[Products] CHECK CONSTRAINT [FK_Products_SubCategory]
GO
/****** Object:  ForeignKey [FK_Products_Supplier]    ******/
ALTER TABLE [dbo].[Products]  WITH CHECK ADD  CONSTRAINT
[FK_Products_Supplier] FOREIGN KEY([SupplierID])
REFERENCES [dbo].[Lookups] ([LookupID])
GO
ALTER TABLE [dbo].[Products] CHECK CONSTRAINT [FK_Products_Supplier]
GO
```

Using the Standard Application

Here is the standard *Web Site Factory* project that is generated straight from the *UniversalLookup* database. Navigate to the *Lookups* page, and you will see a list of records similar to what *SQL Management Studio* provides (with some more advanced functionality, such as adaptive filtering), and you can easily add new records to the table.

Set *Lookup Group Name* filter to *Suppliers* (using the column header), click on the dropdown menu next to *Supplier 3*, and press *New*.
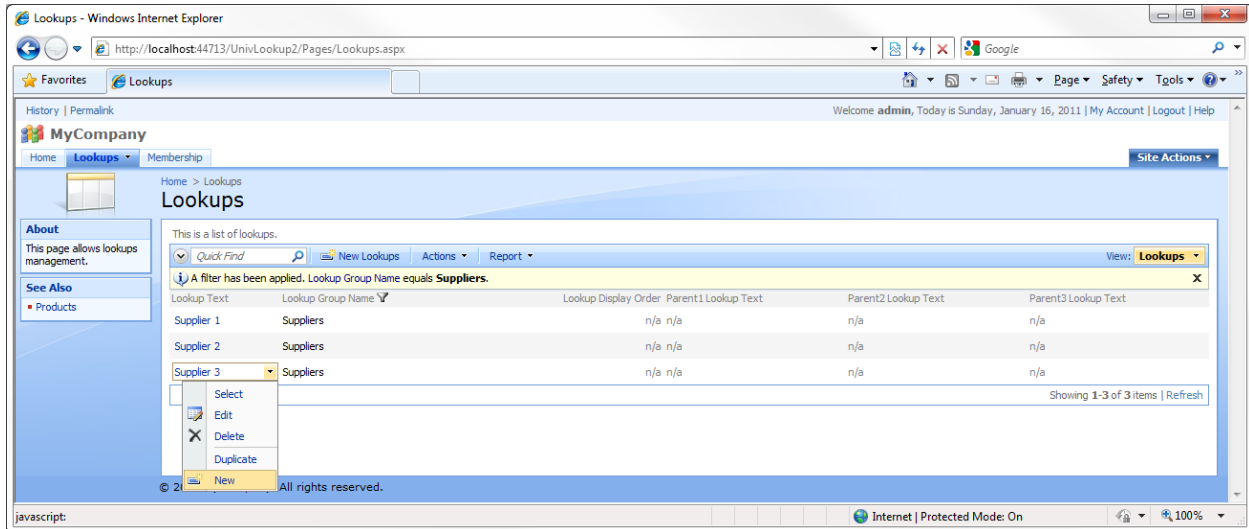
A new inline row will appear. *Lookup Text* field will be called "Supplier 4", with *Lookup Group Name* of "Suppliers". Press *Insert*, and the record will be saved.
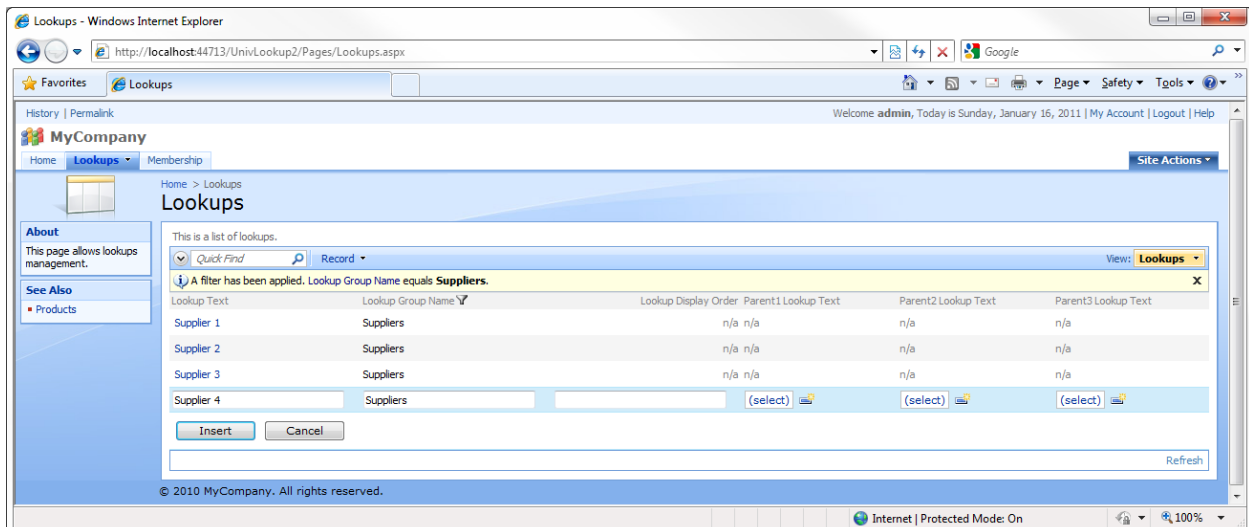
Now, switch to the *Products* page. The product records are displayed, but with a few deficiencies. The "Text" suffix is included for three of the fields (*Category Lookup*, *Sub Category Lookup*, *Supplier Lookup*).
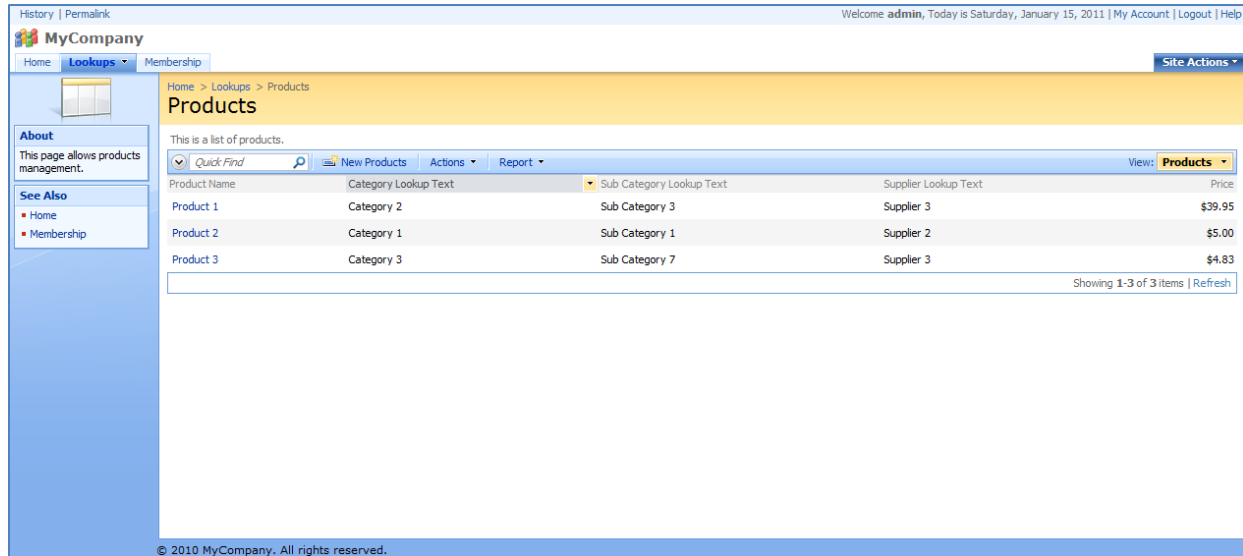
When a record is edited, the lookup values for those fields are not limited by *Lookup Group Name*. All values are presented.
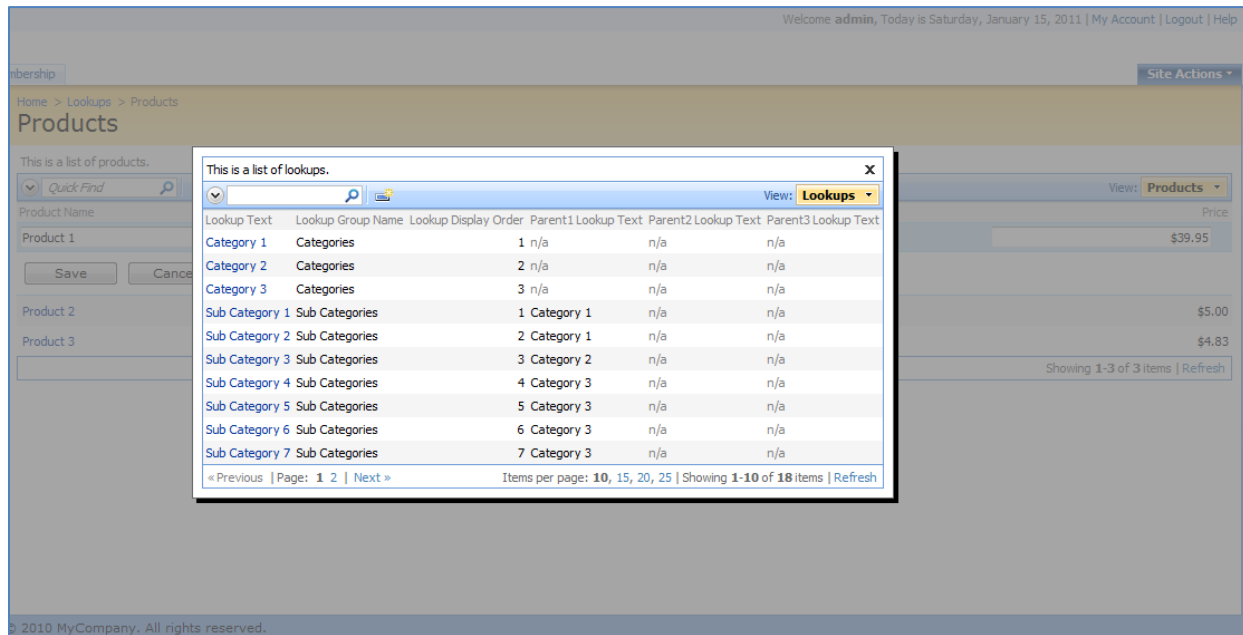
Modifying the Standard Application

Bring up *Code On Time Generator*, select the project name, and press *Design*. Select the *Lookups* controller and switch to the *Views* tab. Edit View *grid1*, and change *Sort Expression* to

```
LookupGroupName, LookupDisplayOrder
```


Now, go back to *All Controllers*, and select *Products* controller. Switch to *Fields* tab. Edit the fields with "Lookup Text" in the *Label*, and remove the relevant text from their respective *Label*.

We will also need to make some modifications to a few fields (*CategoryID*, *SubCategoryID*, and *SupplierID*) to make sure the lookup values are filtered properly. Edit the *CategoryID* field, and scroll down to the *Dynamic Properties* section. Change *Context Fields* value to

```
LookupGroupName='Categories'
```

Dynamic Properties

Context fields may be listed to limit the lookup records by values of other fields of this controller. You can list multiple fields separated by comma.

Field configuration can be used to provide dynamic values for the field properties. The values are derived from other fields in the same data row. List one property per line in format *Property=FieldName*.

Context Fields

LookupGroupName='Categories'

Dynamic Configuration

Save the record, and select *SubCategoryID*. Scroll down to the *Dynamic Properties* section, and in *Context Fields*, insert

```
LookupGroupName='Sub Categories', Parent1LookupID=CategoryID
```

Dynamic Properties

Context fields may be listed to limit the lookup records by values of other fields of this controller. You can list multiple fields separated by comma.

Field configuration can be used to provide dynamic values for the field properties. The values are derived from other fields in the same data row. List one property per line in format *Property=FieldName*.

Context Fields

iroupName='Sub Categories', Parent1LookupID=CategoryID

Dynamic Configuration

Save the change, and edit *SupplierID*. Scroll down to *Dynamic Properties*, and type in

```
LookupGroupName='Suppliers'
```

Dynamic Properties

Context fields may be listed to limit the lookup records by values of other fields of this controller. You can list multiple fields separated by comma.

Field configuration can be used to provide dynamic values for the field properties. The values are derived from other fields in the same data row. List one property per line in format *Property=FieldName*.

Context Fields

LookupGroupName='Suppliers'

Dynamic Configuration

Save the record, close the Designer, and generate the application.

View the Modifications

The columns are now displaying proper headers, with no extraneous "Lookup Text". The records are now sorted according to *Lookup Group Name* and *Lookup Display Order*.

If you navigate to the *Products* page, you can see the extraneous text has been removed here as well.

If you edit a record, and select a *Category* with the lookup, you will only see *Categories*. If you open the *Sub Category* lookup, only sub categories relevant to the selected *Category* will be shown. *Supplier* lookup will only show suppliers.

Further Modification

Nevertheless, lookups for those fields are not really that convenient for the end user. Let's modify the presentation to display a dropdown list for *Category*, radio button list for *Sub Category*, and a list box for *Supplier*.

Open the *Designer*, select the *Products* controller, switch to *Fields* tab, and edit *CategoryID*. Scroll down to *Lookup* section, and select "Drop Down List" for *Items Style*. *Data Value Field* is "LookupID" and *Data Text Field* is "LookupText".

Save, and edit *SubCategoryID* field. Change *Items Style* to "Radio Button List".

Save, and edit *SupplierID* field. Change *Items Style* to "List Box".

Save the field, close the *Designer*, and generate the application. When the web page opens, navigate to *Products* page. If you edit a record, you can see the new item lookup styles in action.