Accessing Data

DataView & DataRelation

DataView

- Represents a particular "view" of a table, often filtered by a particular point of interest.
- Previous video on DataGridView fetched orders records for a customer.
- Hit the database each time customer changed.

DataView

- DataView can eliminate multiple "hits".
- Get all customers records in one DataTable.
- Get all orders records in a second DataTable.
- Now create a DataView object based on the table containing order records:
- DataView dataView = new DataView(dataset.Tables[1]);

DataView

- Assign the data view as DataSource for a second data grid.
OrderGrid.DataSource = dataView;
- Now, when you get the customer id, you "filter" the view to retrieve only the order records with this customer id.
dataView.RowFilter = "CustID = " + custID;
- This is equivalent to sql syntax "WHERE custid = 'value'";

DataRelations

- Because a dataset is a disconnected model of the database, it has to be able to capture relationships among the tables in the database.
- DataSet object captures table relationships in a *DataRelationCollection* that you access through the read only *Relations* property.

DataRelations

- Each **DataRelation** object relates a pair of DataTable objects to each other through **DataColumn** Objects.
- Example: CustomerID is a foreign key into the Orders table (1:Many)
- Declare two DataColumns for the 2 columns of interest
System.Data.DataColumn dataColumn1;
System.Data.DataColumn dataColumn2;

DataRelations

- Get references to the two columns of interest: CustID in Customers and CustID in Orders.

dataColumn1 = dataset.Tables["Customers"].Columns["custID"];
dataColumn2 = dataset.Tables["Orders"].Columns["custID"];

DataRelations

- Initialize a DataRelation ("name", parentColumn, childColumn)

dataRelation = new DataRelation("CustToOrders", datacolumn1, datacolumn2);

- Add the relation to DataRelationCollection

dataset.Relations.Add(dataRelation);