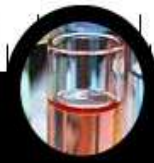
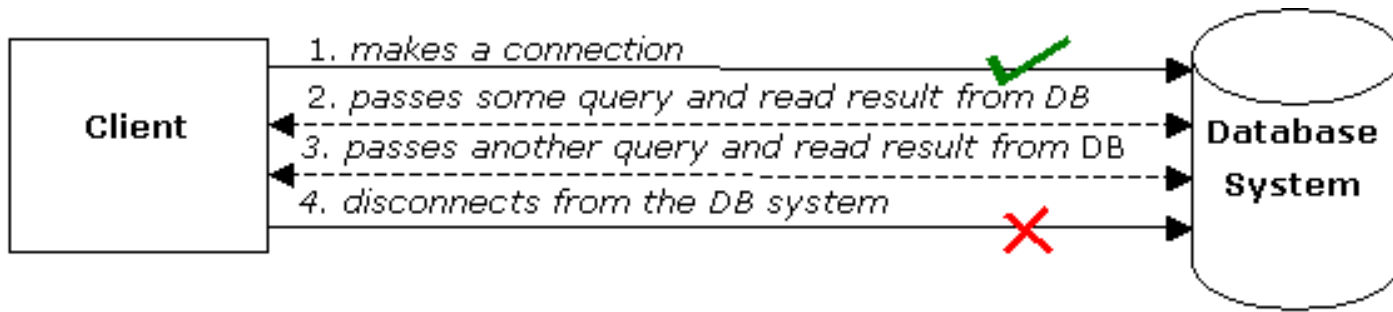


Disconnected Application

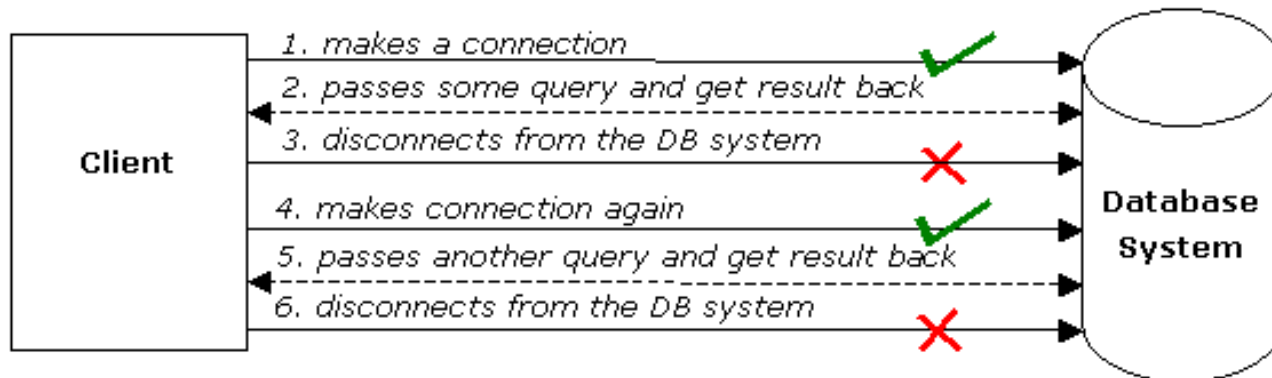


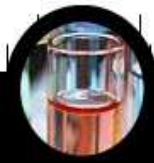
Data Access Architecture

Traditional Data Access Architecture



ADO.NET Disconnected Data Access Architecture



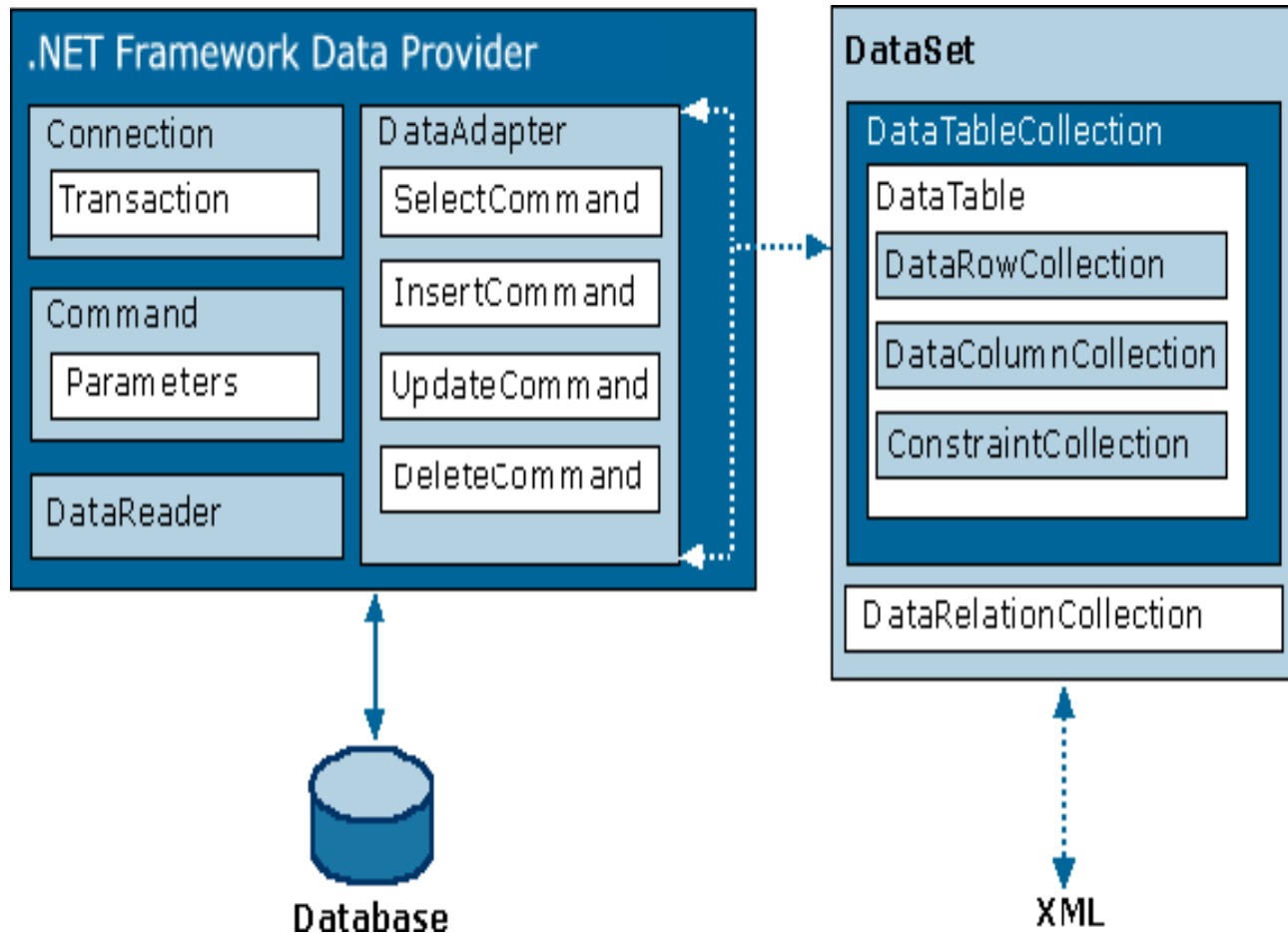


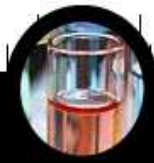
Disconnected Data Access Architecture of ADO.NET

- ADO.NET introduces the concept of disconnected data architecture
- ADO.NET solves this problem(traditional data access) by managing a local buffer of persistent data called dataset
- Your application automatically connects to the database server when it needs to pass some query and then disconnects immediately after getting the result back and storing it in dataset
- This design of ADO.NET is called disconnected data architecture and is very much similar to the connection less services of http over the internet



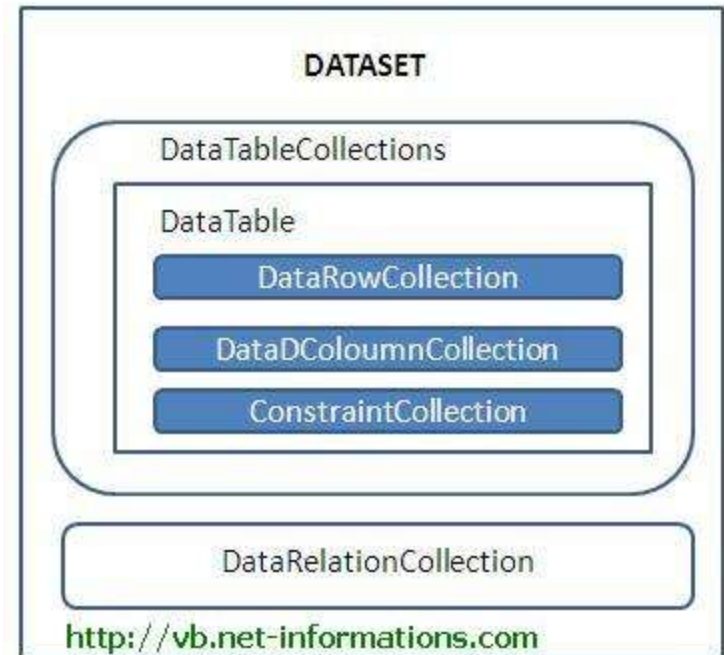
ADO.NET Architecture

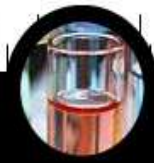




Dataset

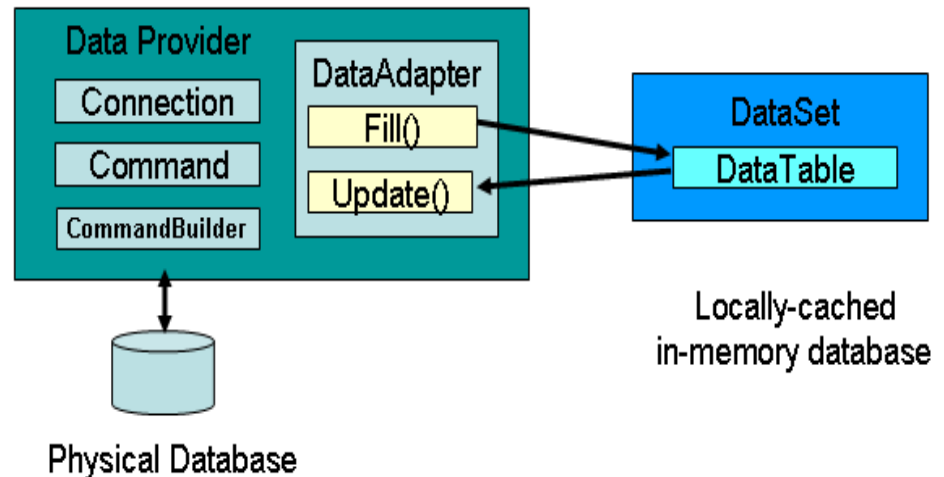
- The ADO.NET DataSet contains DataTableCollection and their DataRelationCollection
- It represents a collection of data retrieved from the Data Source.
- The Dataset can work with the data it contain, without knowing the source of the data coming from
- That is, the Dataset can work with a disconnected mode from its Data Source

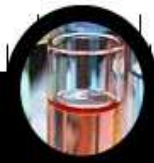




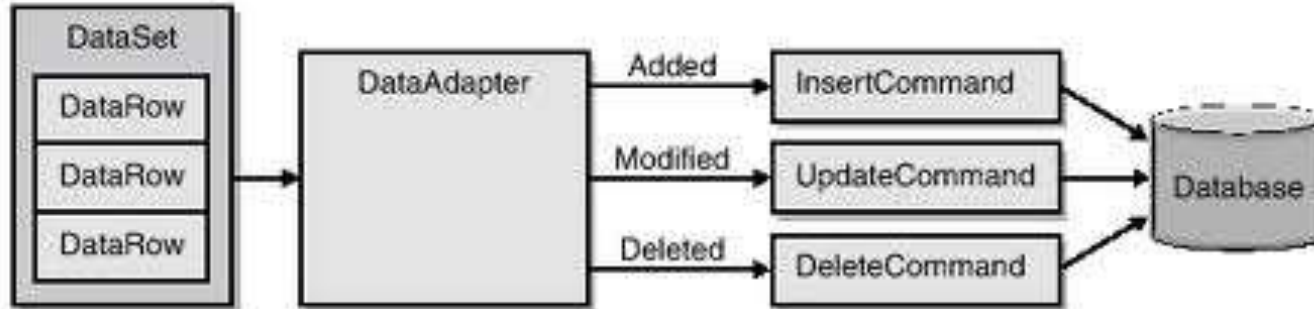
SqlDataAdapter

- An SqlDataAdapter object works as a mediator between a DataSet and a Database.
- The main functions performed by DataAdapter are:
 1. Populate the dataset by fetching data from database, using the 'Fill' method
 2. Update changes made to the dataset back to the database, using the 'Update' method

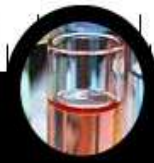




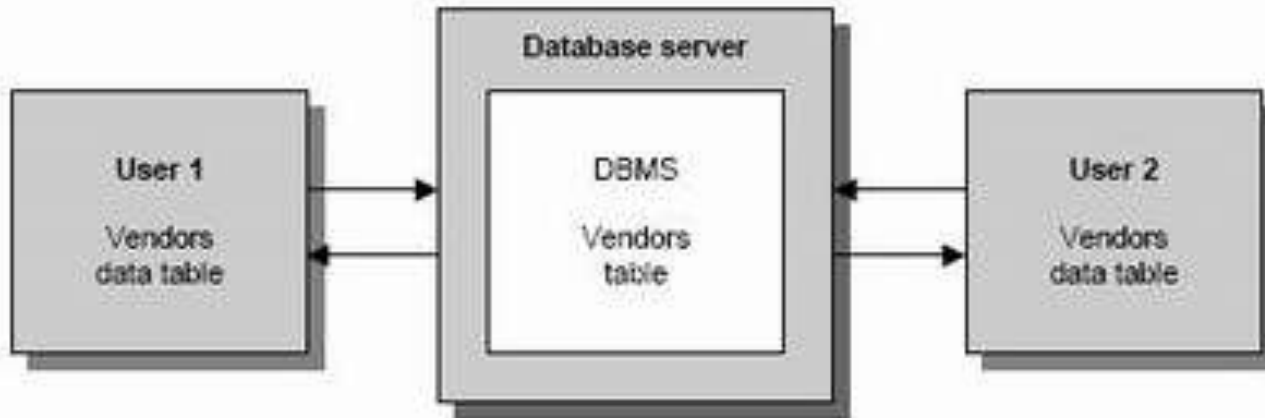
SqlCommandBuilder



- To generate INSERT, UPDATE, or DELETE statements, the SqlCommandBuilder uses the SelectCommand property to retrieve a required set of metadata automatically.
- If you change the SelectCommand after the metadata has been retrieved, such as after the first update, you should call the RefreshSchema method to update the metadata.



Managing Concurrency

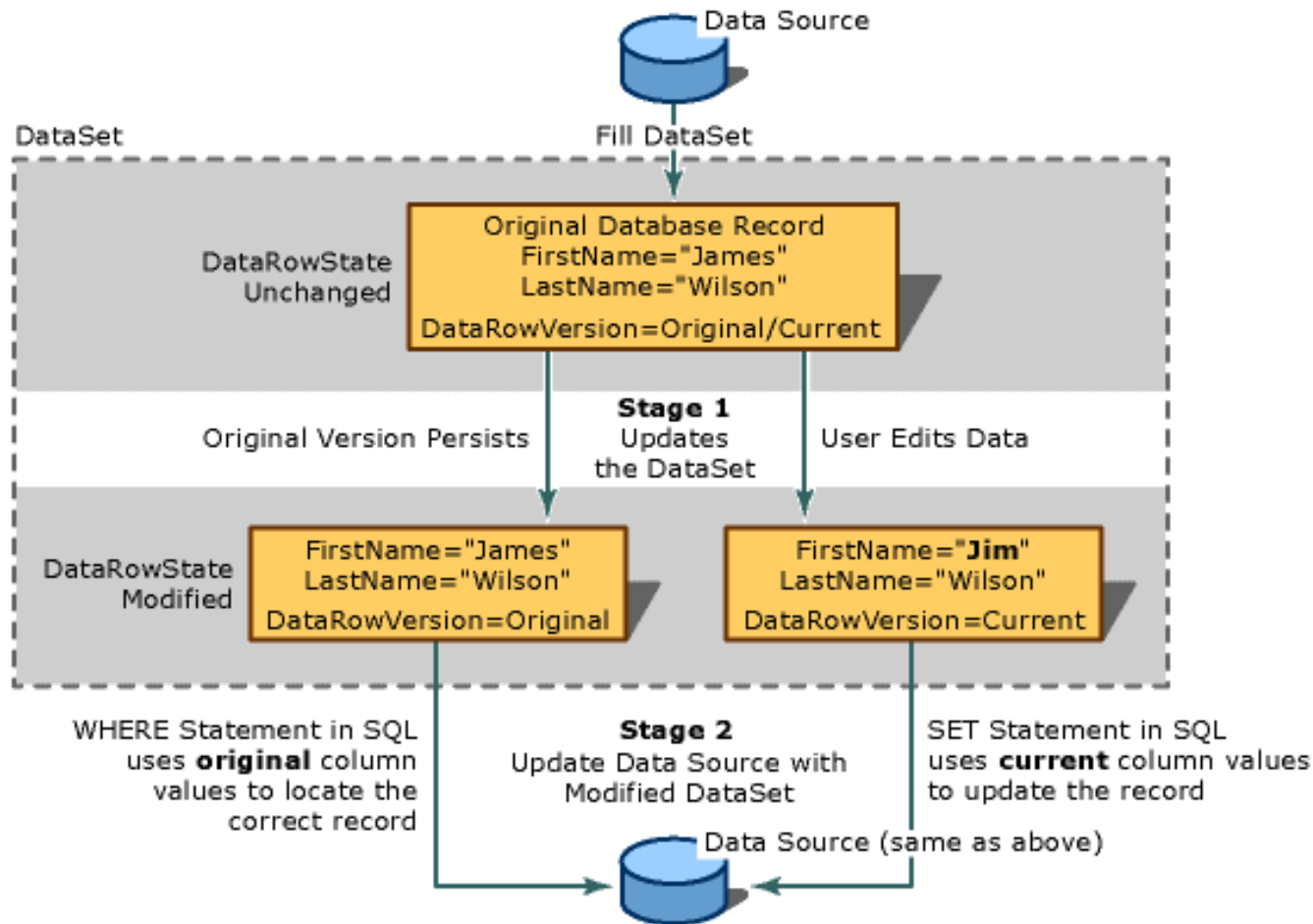


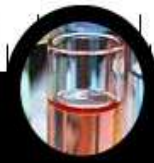
What happens when two users try to update the same row ??

- When two or more users retrieve the data in the same row of a database table at the same time, it is called concurrency. Because ADO.NET uses a disconnected data architecture, the database management system can't prevent this from happening.
- By default, ADO.NET uses *optimistic concurrency* (the program checks to see whether the database row that's going to be updated or deleted has been changed since it was retrieved.)



Dataset Changes

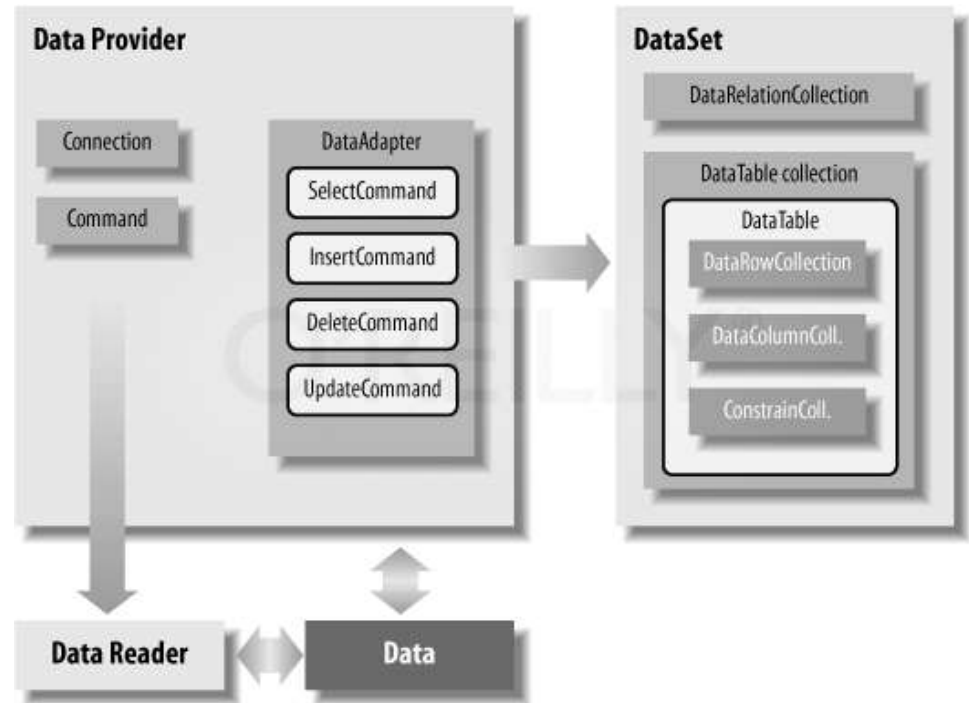


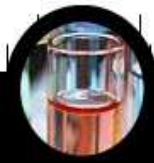


Initializing DataAdapters for Update

- In ADO.NET, you must add your own code for submitting **database updates** to the **DataAdapter** object. There are three ways of doing this:

1. You can supply your own updating logic.
2. You can use the Data Adapter Configuration Wizard to generate the updating logic.
3. You can use the **CommandBuilder** object to generate the updating logic.





SqlDataAdapter Syntax

' The DataSet that holds the data

```
Dim ds As New DataSet(DATASET_NAME)
```

' Create the SqlDataAdapter

```
Dim da As SqlDataAdapter
```

```
da = New SqlDataAdapter(SELECT_STRING, CONNECT_OBJECT)
```

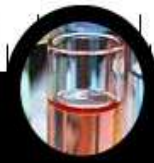
' Create the SqlCommandBuilder

```
Dim cb As SqlCommandBuilder = New SqlCommandBuilder(da)
```

' Fill the DataSet

```
da.MissingSchemaAction=MissingSchemaAction.AddWithKey
```

```
da.Fill(ds)
```



Changing the Order of Insert, Update, and Delete Operations

```
Dim dt As DataTable = ds.Tables("Authors")
```

```
' First process deletes.
```

```
da.Update(dt.Select(Nothing, Nothing, _  
    DataRowState.Deleted))
```

```
' Next process updates.
```

```
da.Update(dt.Select(Nothing, Nothing, _  
    DataRowState.ModifiedCurrent))
```

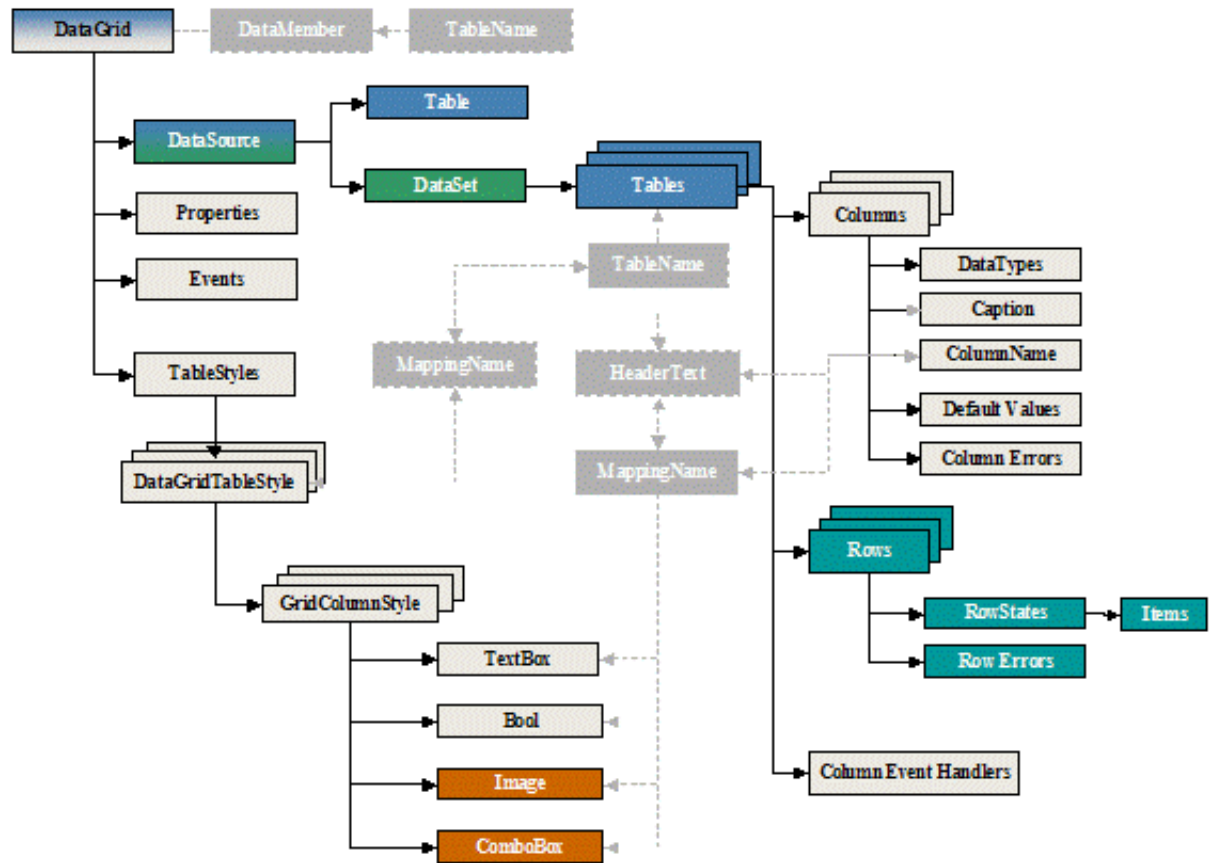
```
' Finally process inserts.
```

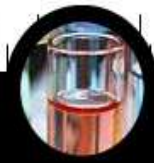
```
da.Update(dt.Select(Nothing, Nothing, _  
    DataRowState.Added))
```



Dataset Manipulation

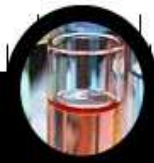
After your dataset is populated with data, you will typically perform some kind of manipulation of the data before sending it back to the data source or to another process or application





Inserting New Records into a Dataset

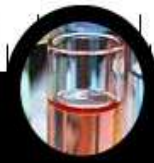
```
Dim rowBaru As DataRow = dtPenjualan.NewRow  
  
rowBaru("Tanggal") = _  
   .dtpPenjualan.Value.ToShortDateString  
rowBaru("Kode_Pelanggan")=tbKodePelanggan.Text  
dtPenjualan.Rows.Add(rowBaru)
```



Updating Existing Records in a Dataset

```
Dim rowUbah As DataRow = dtPenjualan.Rows(0)

rowUbah.BeginEdit()
rowUbah("Tanggal") = _
    dtpPenjualan.Value.ToShortDateString
rowUbah("Kode_Pelanggan") = tbKodePelanggan.Text
rowUbah.EndEdit()
```

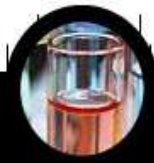


Deleting Records in a Dataset

```
Dim rowHapus As DataRow = dtPenjualan.Rows(0)  
dtPenjualan.Rows.Remove(rowHapus)
```

Or

```
dtPenjualan.Rows(0).Delete()
```

Committing Changes in the Dataset

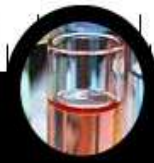
- You can commit the pending changes to the dataset by calling the **AcceptChanges** method.
- Typically, **AcceptChanges** would be called at the following times in your application.

```
`commit
```

```
dtPenjualan.AcceptChanges ()
```

```
`rollback
```

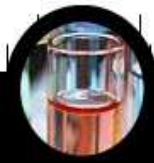
```
dtPenjualan.RejectChanges ()
```



Finding Rows in ADO.NET

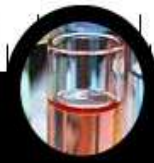
ADO.NET implements three basic forms to locate Rows :

1. Find Method
2. Contains Method
3. Select Method



Find Method

```
`nilai pada sebuah kolom primary key  
Dim drCari As DataRow  
drCari = dtPenjualan.Rows.Find  
    ("Data Yang Dicari")  
If Not drCari Is Nothing Then  
    MessageBox.Show("Data Ditemukan")  
    `menampilkan isi kolom pertama pada data  
    yang ditemukan  
    MessageBox.Show(drCari.Item(0))  
Else  
    MessageBox.Show("Data Tidak Ada")  
End If
```



Contains Method

```
`metode Contains
```

```
Dim statusCari As Boolean
```

```
statusCari = dtPenjualan.Rows.Contains  
    ("Data Yang Dicari")
```

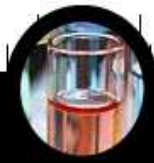
```
If statusCari = True Then
```

```
    MessageBox.Show("Data Ditemukan")
```

```
Else
```

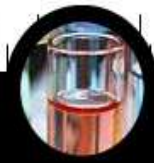
```
    MessageBox.Show("Data Tidak Ada")
```

```
End If
```



Select Method

```
Dim drCari() As DataRow
drCari = dtPenjualan.Select("Kode_Pelanggan=" & _
    "'Data Yang Dicari'")
If drCari.Length > 0 Then
    MessageBox.Show("Data Ditemukan")
    `menampilkan isi kolom pertama pada data yang
    ditemukan
    Dim dr As DataRow
    For Each dr In drCari
        MessageBox.Show(dr.Item(0))
    Next
Else
    MessageBox.Show("Data Tidak Ada")
End If
```



Daftar Pustaka

<http://www.codeproject.com/KB/grid/practicalguidedatagrids1.aspx>

<http://www.programmersheaven.com/2/FAQ-ADONET-Disconnected-Data-Access>

<http://vb.net-informations.com/dataset/ado.net-dataset.htm>

http://msdn.microsoft.com/en-us/library/ee817654.aspx#daag_performingupdates

<http://www.microsoft.com/mspress/books/sampchap/5199b.aspx>

http://www.akadia.com/services/dotnet_rowstate.html