

## STORED PROCEDURE DA CODICE

### **Esecuzione di una Stored Procedure da codice C#**

Per richiedere l'**Esecuzione di una Stored Procedure da codice C#**, è necessario utilizzare la *classe* `SqlCommand` e l'insieme dei record risultanti dall'esecuzione, viene restituita in un oggetto di *classe* `SqlDataReader`.

E' necessario anzitutto creare un oggetto di **classe** `SqlCommand` e specificare, come testo del comando, il *nome della Stored Procedure* che si desidera eseguire:


```
SqlCommand cmd = new SqlCommand ("RecuperaDatiPersona", cn);
```

Si procede utilizzando la **proprietà** `CommandType` assegnandovi la costante `CommandType.StoredProcedure`, per informare VB che il comando richiesto è una Stored Procedure e non un normale comando sql:

```
cmd.CommandType = CommandType.StoredProcedure;
```

Per **Attribuire un Valore ai Parametri** previsti della *Stored Procedure*, si utilizza il **collection** `Parameters` della classe `SqlCommand`: anzitutto, con il consueto **metodo** `Add`, si aggiungono a tale collection, i parametri necessari, specificandone il *nome* e il *tipo*:

```
cmd.Parameters.Add("@IdPersona", SqlDbType.Int);
```


 Le *costanti enumerative* del gruppo `SqlDbType` elencano tutti i possibili tipi ammessi da SQL Server e fra questi il tipo `Int`.

Quindi, è possibile attribuire loro un valore, utilizzando la **proprietà** `Value` degli elementi della collection:

```
cmd.Parameters["@IdPersona"].Value = 2;
```

Infine, per **Inviare la Richiesta di Esecuzione** della Stored Procedure al server SQL, si utilizza il consueto **metodo** `ExecuteReader` della classe `SqlCommand`. Come è noto, il *metodo* `ExecuteReader` restituisce un oggetto di **classe** `SqlDataReader` contenente l'insieme dei Record che costituiscono il risultato dell'esecuzione della Stored Procedure:

```
SqlDataReader dr;  
dr = cmd.ExecuteReader();
```

 Nell'esempio, dopo aver posizionato `dr` sul primo record con il metodo `dr.read`, si può accedere ai campi del primo record con `dr("Cognome")` e `dr("Nome")`, che avranno come valore rispettivamente `"De Paperoni"` e `"Paperon"`.

E' possibile anche richiedere l'esecuzione di una Stored Procedure utilizzando un **DataAdapter** e il suo metodo **Fill** per dislocare i risultati in un **DataTable**. Per farlo è sufficiente *creare il DataAdapter passando come parametro l'oggetto Command*:

```
SqlCommand cmd = new SqlCommand ("RecuperaDatiPersona", cn);  
cmd.CommandType = CommandType.StoredProcedure;  
cmd.Parameters.Add("@IdPersona", SqlDbType.Int);  
cmd.Parameters["@IdPersona"].Value = 2;
```

```
sqlDataAdapter da = new sqlDataAdapter();  
sqlDataAdapter.SelectCommand = cmd;           ' ... associa l'oggetto SqlCommand "cmd" al DataAdapter "da"  
DataTable dt = new DataTable();  
  
da.Fill(dt);
```

Il vantaggio di disporre dei dati nel *DataTable* è la sua maggiore flessibilità rispetto al *DataReader*.

 Ad esempio, è possibile *"collegare"* il *DataTable* a un *DataGridView* per visualizzare/editare i dati.