

# ADO.NET – Disconnected Mode

Il **Disconnected Mode** consente di

- manipolare i dati recuperati dall'origine dati
- **successivamente riconciliarli con l'origine dati**

Le classi disconnesse forniscono un modo comune di lavorare con i dati disconnessi, indipendentemente dall'ambiente di origine dati.

# ADO.NET – Disconnected Mode

Principali classi utilizzate in Disconnected Mode:

- *DataSet*
- *DataTable*
- *DataColumn*
- *DataRow*
- *Constraint*
- *DataRelation*

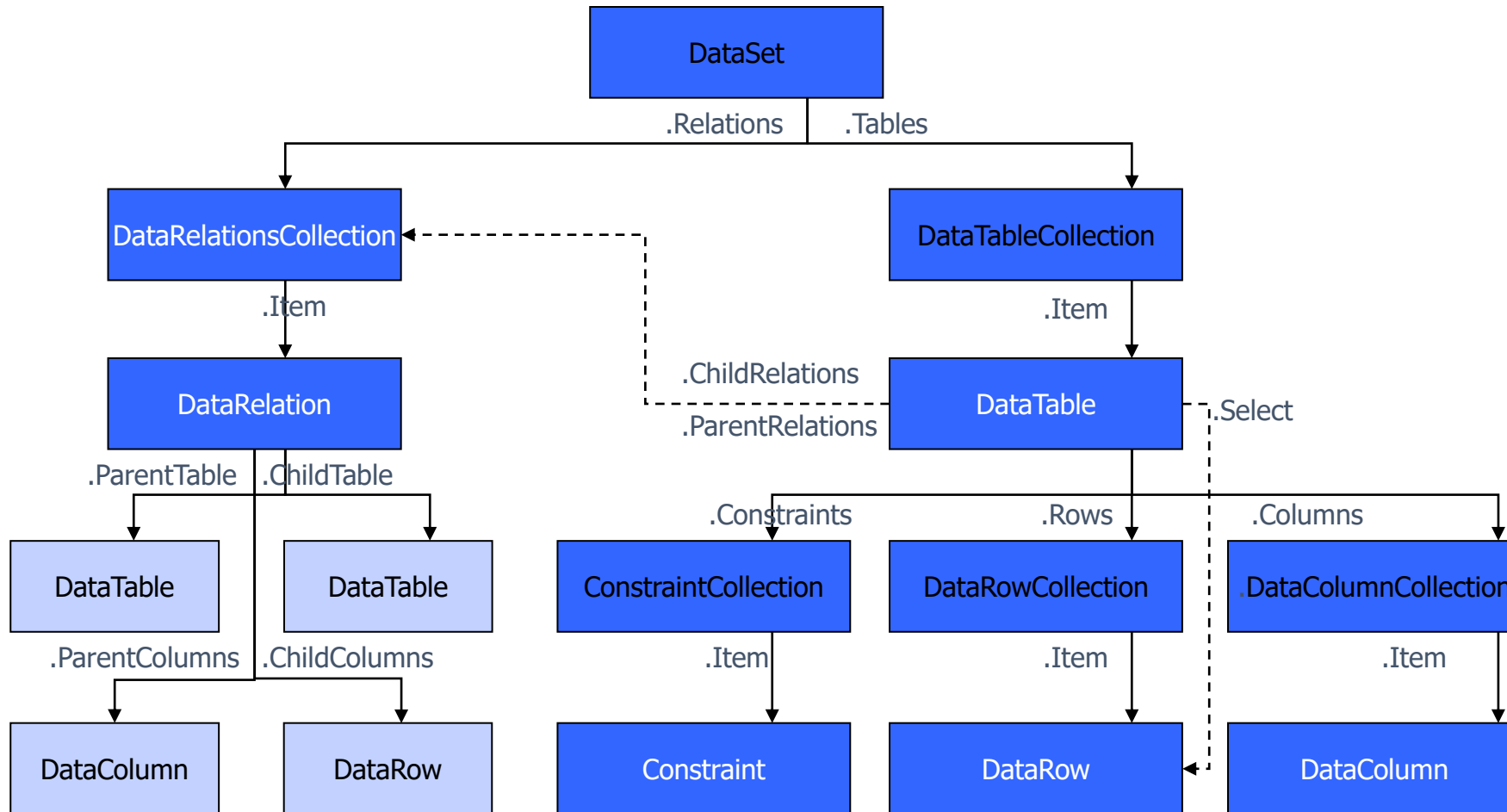
# ADO.NET - DataSet

Il **DataSet** è progettato esplicitamente per l'accesso ai dati indipendentemente da qualsiasi origine dati: può essere utilizzato con origini dati multiple e diverse o utilizzato per gestire i dati locali per l'applicazione.

Il **DataSet** contiene una raccolta di uno o più oggetti **DataTable** costituiti da

- righe e colonne di dati
- informazioni relative a chiave primaria, chiavi esterne, vincoli e relazione sui dati

# ADO.NET - DataSet



# Architettura del DataSet

L'oggetto *DataSet* rappresenta un contenitore di dati in memoria e contiene una serie di

- Tabelle
- Relazioni tra le tabelle
- Vincoli di integrità dei dati

L'utilizzo di tali oggetti è assolutamente simile nelle funzionalità ad un database relazionale

È possibile salvare i dati contenuti in formato XML e i relativi metadati in XSD (Xml Schema Definition)

# Popolare il DataSet

Il *DataSet* non dipende dalla fonte dati utilizzata per popolarlo

- È possibile utilizzare un oggetto di tipo *DataAdapter* per popolare il *DataSet*
- Il *DataSet* può essere popolato con informazioni provenienti da differenti fonti dati
  - SQL Server, Oracle, File XML, ecc.
- Si può costruire la struttura via codice e utilizzarlo come data store senza nessun legame con una fonte dati

Le *DataTable* utilizzano tipi di dati .NET, indipendenti dalla sorgente

Non funzionano via ws/remoting:

- Dalla 2.0 sono serializzabili di default

# DataTable

Rappresenta il contenitore delle informazioni in memoria

Struttura descritta da una collection di *DataColumn* e da una di *Constraint*

- *Primary Key*
- *UniqueConstraint*
- *ForeignKeyConstraint*
  - Supporta la Cascading DRI attraverso le proprietà *DeleteRule* e *UpdateRule* (come SQL Server...)
  - La proprietà *AcceptRejectRule* consente di propagare le modifiche fatte su una tabella “master” ad una o più “details”

# ADO.NET - Adapter

Il **DataAdapter** fornisce il ponte tra l'oggetto **DataSet** e l'origine dati.

**DataAdapter** utilizza oggetti *Command* per eseguire comandi SQL sull'origine dati per caricare sia *DataSet* con dati sia per riconciliare le modifiche apportate ai dati nel *DataSet* con l'origine dati

La classe **DataAdapter** permette di collegare l'origine dati e le classi disconnesse tramite le classi connesse.



# Il metodo Fill

L'operazione di query e l'apertura della connessione verso la fonte dati viene eseguita alla chiamata del metodo *Fill* dell'oggetto *DataAdapter*

- Accetta come parametri il *DataSet* da utilizzare e il nome di una tabella o un oggetto *DataTable* da riempire
- Se la tabella non esiste viene creata

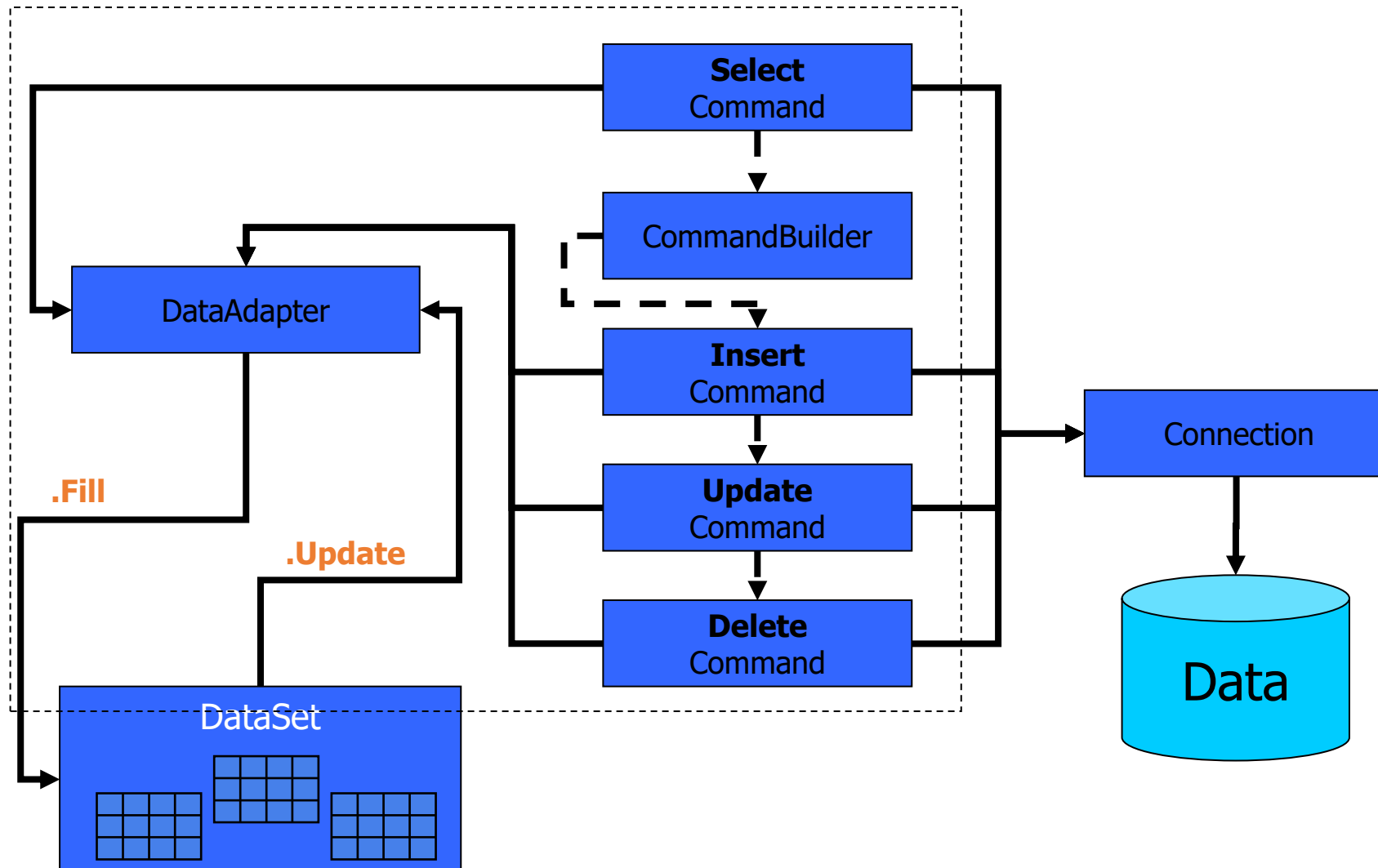
La connessione al database viene creata (se necessario), aperta e chiusa automaticamente al riempimento del *DataSet*

# Il metodo Update

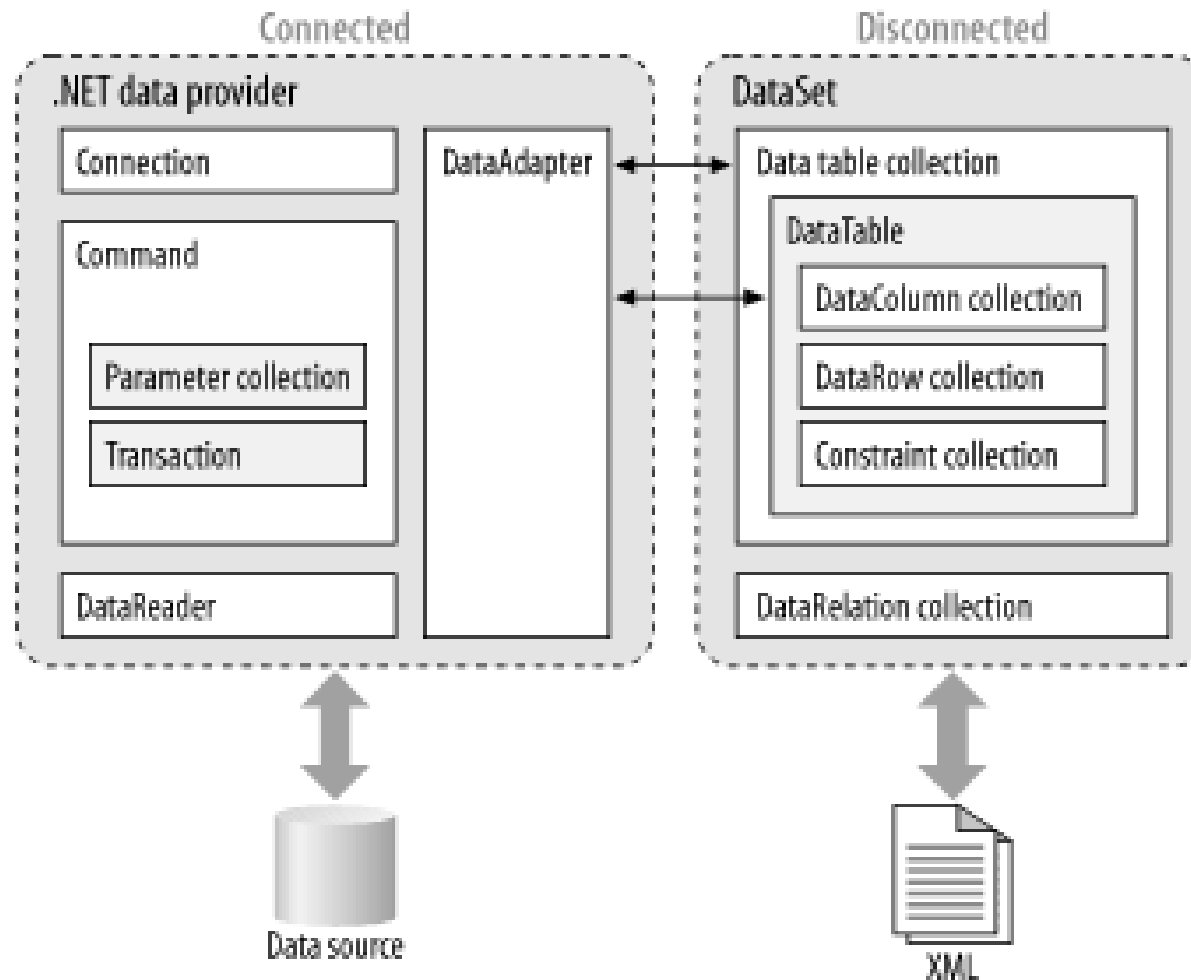
L'operazione di aggiornamento dei dati nella fonte dati viene eseguita alla chiamata del metodo *Update* dell'oggetto *DataAdapter*

- Accetta come parametri il *DataSet* da utilizzare e il nome di una tabella o un oggetto *DataTable*
- L'aggiornamento viene eseguito riga per riga
- Per ogni riga inserita, modificata ed eliminata, il metodo *Update* determina il tipo di modifica che è stata eseguita su di essa (Inserisci, Aggiorna o Elimina) e chiama i rispettivi *Command* INSERT, UPDATE o DELETE
- Se le istruzioni INSERT, UPDATE o DELETE non sono state specificate, il metodo genera un'eccezione

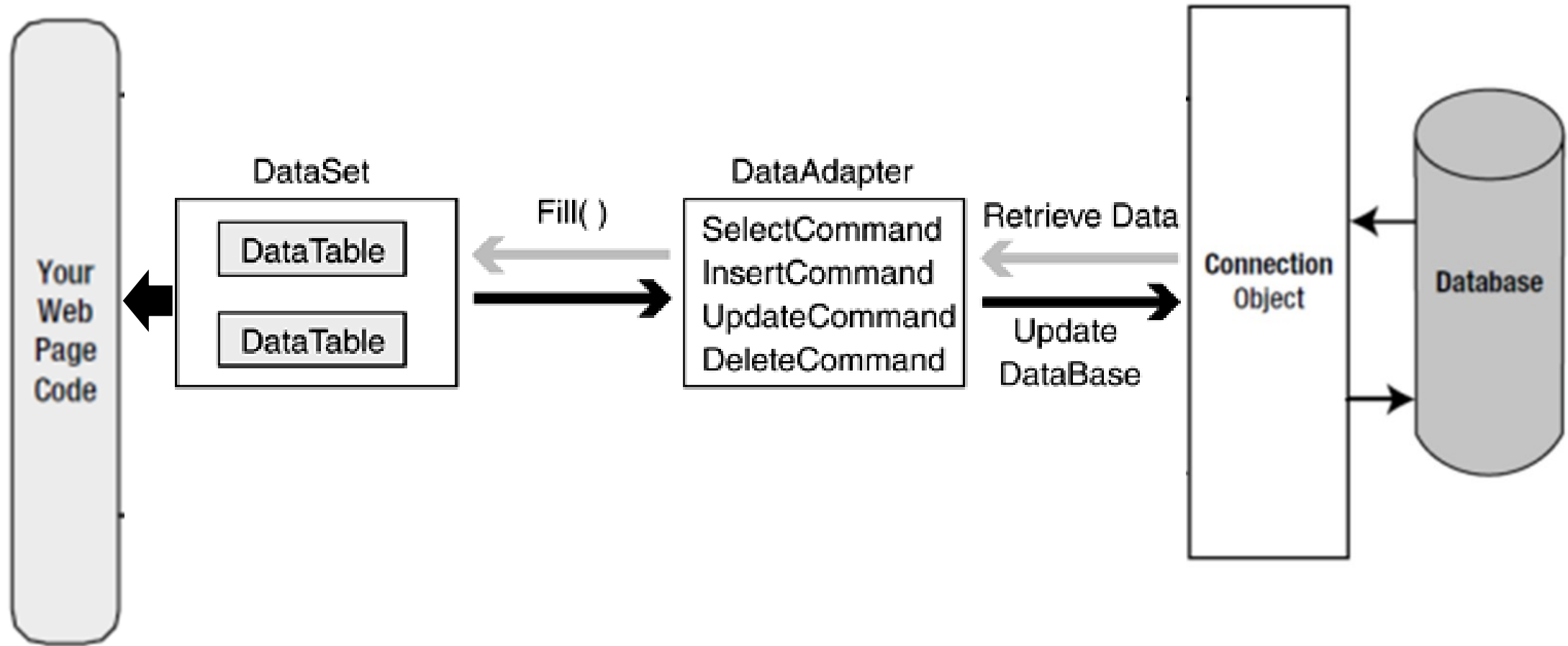
# Architettura del DataAdapter



# ADO.NET - Adapter



# ADO.NET – Disconnected Mode



# Esercitazione 2

Realizzare una Console app che acceda al database Ticketing utilizzando il Disconnected Mode di ADO.NET e che implementi le stesse operazioni:

- Stampi la lista dei Ticket in ordine cronologico (dal più recente al più vecchio)
- Permetta l'inserimento di nuovi ticket (i dati devono essere inseriti dall'utente)
- Permetta la cancellazione di un Ticket (utilizzare l'ID univoco per identificarlo)

