

# Academy Week 4




**Roberto Ajolfi**

Senior Developer@icubedsrl

[roberto.ajolfi@icubed.it](mailto:roberto.ajolfi@icubed.it)



# Week 4 - Agenda

- ADO.NET
  - Connected Mode
  - Disconnected Mode
- Entity Framework
  - Definizione del Modello Dati
  - LINQ
  - Migrations
-  Esercitazione



*"Ci sono domande ingenuie, noiose, mal formulate, domande poste senza avere esercitato un'autocritica sufficiente. Ogni domanda è però un grido per capire il mondo.*

*Non esistono domande stupide."*

(C. Sagan)



# ADO.NET

- **ADO.NET** è una tecnologia di accesso ai dati di Microsoft
- È costituito da un insieme di componenti software che i programmatori possono utilizzare per accedere ai dati e ai servizi dati da un database
- Fornisce un accesso coerente
  - alle origini dati come SQL Server
  - alle origini dati esposte tramite OLE DB e ODBC

# ADO.NET - Provider

- ADO.NET include i **Provider** di dati per
  - la connessione a un database
  - l'esecuzione di comandi
  - il recupero dei risultati
- I **provider** di dati .NET Framework forniscono una serie di oggetti per gestire l'accesso ai dati
  - L'oggetto **Connection** fornisce connettività a un'origine dati tramite una Connection String
  - L'oggetto **Command** consente l'accesso ai comandi del database per restituire dati, modificare dati, eseguire procedure memorizzate e inviare o recuperare informazioni sui parametri

# ADO.NET - Provider

- Insieme di oggetti che ereditano da queste classi astratte di base:
  - *DbConnection*
  - *DbCommand*
  - *DbDataReader*
  - *DbDataRecord*
  - *DbDataAdapter*
- Creati come layer leggero tra la fonte dati e l'applicazione

# ADO.NET - Provider

- Realizzati come codice managed all'interno del .NET Framework (namespace *System.Data.Common*)
- Costituiscono la parte “connessa” alle varie fonti dati
- Permettono l'utilizzo di comandi diretti verso le fonti dati attraverso gli oggetti *Command*
  - Chiamata a Stored Procedure
  - Codice SQL dinamico

# ADO.NET - Provider

Nel namespace *System.Data.Common*

DbCommand	DbCommandBuilder	DbConnection
DataAdapter	DbDataAdapter	DbDataReader
DbParameter	DbParameterCollection	DbTransaction
DbProviderFactory	DbProviderFactories	DbException



# ADO.NET – Connection String

Una stringa di connessione contiene le informazioni di inizializzazione fondamentali per creare una connessione con un database.

```
Server=tcp:platone.database.windows.net,1433; Initial Catalog=University; Persist  
Security Info=False; User ID={your user}; Password={your_password};  
MultipleActiveResultSets=False; Encrypt=True; TrustServerCertificate=False;  
Connection Timeout=30;
```

# ADO.NET - Modalità

**ADO.NET** consente l'accesso ai dati in due modalità distinte:

- **Connected Mode**  
(*Connection, Commands, DataReader ...*)
- **Disconnected Mode**  
(*Connection, Adapter, DataSet, DataTable ...*)

# ADO.NET – Connected Mode

Il **Connected Mode** fornisce

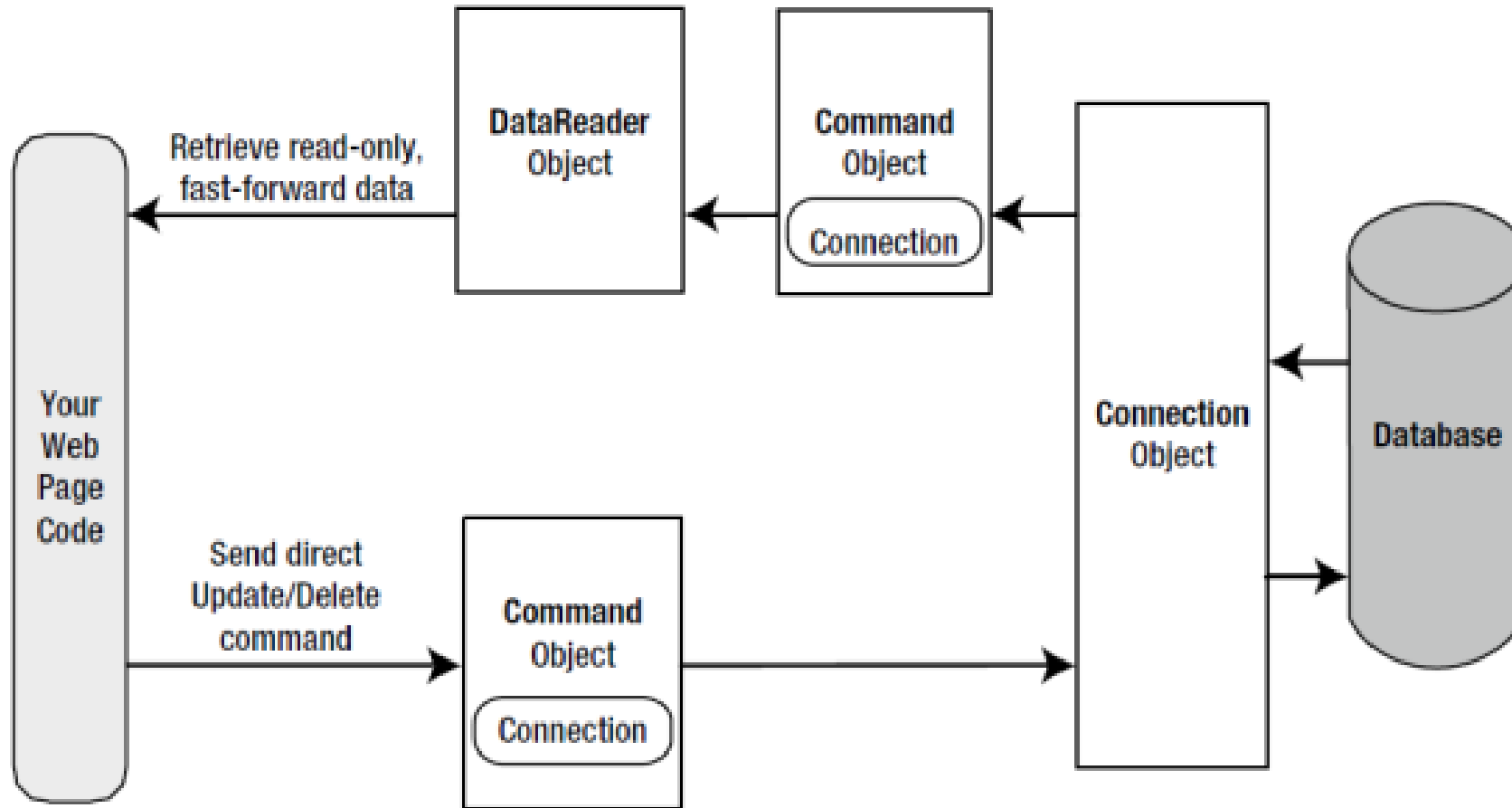
- accesso di sola lettura (e forward-only) ai dati nell'origine dati
- la possibilità di eseguire comandi sull'origine dati

# ADO.NET – Connected Mode

Principali classi utilizzate in Connected Mode:

- Connection (*SqlConnection*)
- Command (*SqlCommand*)
- DataReader (*SqlDataReader*)
- Parameter (*SqlParameter*)

# ADO.NET – Connected Mode



# ADO.NET – Connected Mode

## Step

1. Creare e Aprire una Connessione
2. Creare un Command
  - Creare Parametri se necessario
3. Eseguire Command (DataReader/NonQuery/Scalar)
4. Lettura Dati a schermo
5. Chiudere Connessione

# Esercitazione 1

Creare un nuovo database *Ticketing* con una sola tabella *Tickets*. Le colonne sono:

- *ID* (int, PK, auto-incrementale)
- *Descrizione* (varchar(500))
- *Data* (datetime)
- *Utente* (varchar(100))
- *Stato* (varchar(10)) – New, OnGoing, Resolved

Realizzare una Console app che acceda al database Ticketing utilizzando il Connected Mode di ADO.NET e che:

- 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)

