# Esercitazione di Fine Settimana – Week 5

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Nome | Michela |
|  |  | Cognome | Putzu |
|  |  | Data | 8/10/2021 |

Leggete attentamente ogni domanda e argomentare quanto più possibile fornendo anche degli esempi.

1. ***Descrivere le modalità di utilizzo di ADO.NET***

Abbiamo due diverse modalità:

**Connected Mode 🡪** fornisce accesso di sola lettura ai dati nell’origine dati e la possibilità di eseguire comandi sull’origine dei dati. Le principali classi utilizzate in questa modalità sono: *Connection*, *Command*, *DataReader* e *Parameter*

**Disconnected Mode 🡪** consente di manipolare i dati recuperati dall’origine e successivamente riconciliarli con l’origine dati. Tra le principali classi utilizzate abbiamo: *DataSet, DataTable, DataColumn, DataRow, Constraint* e *DataRelation*.

1. ***Quali sono i metodi di esecuzione della classe DbCommand disponibili e in quali casi vanno utilizzati?***

Il DbCommand consente l’accesso ai comandi del database per restituire, modificare dati, eseguire procedure memorizzate e inviare o recuperare informazioni sui parametri.

1. ***Descrivere l’utilizzo della classe DataAdapter***

Il DataAdapter fornisce il ponte tra l’oggetto DataSet e l’origine dati. Utilizza oggetti come Command:

* Per eseguire comandi SQL
* Per caricare DataSet con dati
* Per conciliare le modifiche apportate ai dati nel DataSet con l’origine dati

1. ***Come si definisce il modello dati in Entity Framework?***

Entity Framework è un linguaggio ORM, tecnica che converte dati da type system incompatibili. Ha tre caratteristiche fondamentali:

* Mapping 🡪 definisce come il database si incastra negli oggetti e viceversa
* Fetching 🡪 sa come recuperare i dati dal database e materializzare i rispettivi oggetti
* Persistenza del grafo🡪 sa come salvare le modifiche agli oggetti, generando le query SQL

Utilizza:

* DbContext : rappresenta una combinazione dei modelli Unit Of Work e Repository in modo tale da poter essere utilizzata per eseguire query da un database e raggruppare le modifiche che verranno quindi riscritte nell'archivio.
* DbSet: classe che rappresenta le entity serve per fare operazioni CRUD
* Definizione di Primary Key
* Navigation properties 🡪 rappresenta la relazione tra due entità

1. ***Descrivere l’utilizzo delle Migration e i vantaggi che ne derivano***

Le migrations sono un meccanismo che consente, utilizzando l’approccio code-first, l’aggiornamento del database a fronte di modifiche al modello.

Esistono due tipi di migrations:

* Automatiche
* Manuali o code-based

Sono utili quando vogliamo più controllo sulle modifiche automatiche e servono due comandi per utilizzarle, tramite la Package Manager Console:

* Add-Migration “Migration name”
* Update-Database

**Esercitazione Pratica**

Realizzare un sistema di gestione delle spese che si basi su:

* Un database **GestioneSpese** (SQL Server), costituito dalle tabelle
  + **Spese**
    - *Id* (int, PK, auto-incrementale)
    - *Data* (datetime)
    - *CategoriaId* (int, FK)
    - *Descrizione* (varchar(500))
    - *Utente* (varchar(100))
    - *Importo* (decimal)
    - *Approvato* (bit)
  + **Categorie**
    - *Id* (int, PK, auto-incrementale)
    - *Categoria* (varchar(100))
* Una **Console app** che consenta di:
  + Inserire nuove Spese
  + Approvare le Spese esistenti (impostare il campo *Approvato*)
  + Cancellare le Spese esistenti
  + Mostri
    - l'elenco delle Spese Approvate
    - L'elenco delle Spese di uno specifico Utente
    - Il totale delle Spese per Categoria

VINCOLI TECNICI

* Utilizzare Entity Framework
* Utilizzare l'approccio Code-First e attivare le Migrations

**OPZIONALE**: Implementare una o più delle funzionalità utilizzando ADO.NET (Connected o Disconnected mode, a scelta)

*Es. Inserire una nuova spesa*