# Esercitazione di Fine Settimana – Week 6

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Nome | Laura |
|  |  | Cognome | Pennetta |
|  |  | Data | 25/06/2021 |

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

1. Descrivere le modalità di definizione del modello dati in Entity Framework?

L’entity Framework è un ORM (Object relational mapping) di miscrosoft basato su .NET Framework. È un insieme di tecnolgie ADO.NET per lo sviluppo software ed, inoltre, è un prodotto open source. Effettua una traduzione dell’oggetto in tabella e definisce un modello di astrazione dei dati. Le operazioni fondamentali alla base sono: mapping, fetching e persistenza del grafo. Fra gli strumenti impiegati troviamo: l’entity client data provider e l’entity data model. Nel primo caso si ha un livello di astrazione che permette di usare l’entity framework con più sorgenti dati; nel secondo caso, invece, viene rappresentato il modello di mapping fra database e oggetti. Esistono diversi approcci, quali:

* DataBase First: il modello viene importato da un database esistente.
* Model First: è un approccio intermedio, infatti, il database viene creato dal designer di Visual Studio.
* Code First: il modello viene creato dal nostro codice.

1. Scrivere con un esempio pratico come definire una chiave primaria ed una chiave esterna utilizzando le data annotation e fluent api.

Chiave primaria:

[key]

Public string ID {get;set;}

Public string ID {get;set;} //nella classe

Builder.HasKey(k => k.ID); //scritta per come la troviamo in una configuration

Chiave esterna:

//Navigation property:

public BlaBla BlaBla { get; set; }

//Chiave esterna per il blog:

public string blablaID { get; set; }

//nella classe che ha una foreign key che fa riferimento all’entità Uff

public int UffID { get; set; }

public Uff Uff { get; set; }

//nella configuration

builder.HasOne(u => u.Uff).WithMany(l => l.Lagne)

.HasForeignKey(f => f.UffID);

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

Le migrations servono in caso di modifiche al database e ne esistono di due tipi: Automatiche (poco invasive) e Manuali (richiedono un intervento specifico). L emigrations risultano utili quando siamo in produzione, quando lo stato del databaseè già definito e quando vogliamo più controllo sulle modifiche automatiche.

**Esercitazione Pratica**

Sulla base del modello definito in allegato realizzare i requisiti richiesti.

L'applicazione deve consentire di gestire i clienti dell'assicurazione. In particolare deve essere possibile:

- Inserire nuovi clienti

- Inserire una polizza (rcauto, furto, vita) per un cliente

- Stampare i dati delle polizze presenti a db (comprese le info sul cliente che l'ha stipulata\*).

\*In fase di stampa dei dati del cliente deve essere visualizzabile anche la spesa totale mensile che il cliente sostiene per tutte le sue polizze.

Requisiti tecnici:

Utilizzare le Migration di EF per la modifica del DB

Utilizzare le Fluent-Api per la gestione dei requisiti del database

Realizzare un menù per l’interazione con l’utente