Entity Framework

Entity Framework – What?

- Entity Framework (EF) is an object-relational mapper that enables .NET developers to work with relational data using domain-specific objects

Entity Framework – When?

- When use of ORM weighs over using a traditional DB
- When the database growth is predictable and organic

Entity Framework Developer Workflows

- Model First(*New Database*)
  - Create model in designer
  - Database created from Model
  - Classes auto generated from model

Entity Framework Developer Workflows

- Database First (*Existing Database*)
  - Reverse engineer model in designer
  - Classes auto generated from model

Entity Framework Developer Workflows

- Code First (*New Database*)
  - Define classes and mapping in code
  - Database created from Model
  - Use Migrations to evolve the model

Entity Framework Developer Workflows

- Code First (*Existing Database*)
  - Define classes and mapping in code
  - Reverse engineer tools available

System.Data.Entity Namespace

- Contains classes that provide access to the core functionality of the Entity framework
- Enable you to query, insert, update and delete data using CLR strongly typed objects

DbContext

- DbContext instance represents a combination of the Unit Of Work and Repository patterns such that it can be used to query from a database and group together changes that will then be written back to the store as a unit

DbSet<TEntity>

- A DbSet represents the collection of all entities in the context, or that can be queried from the database, of a given type.

POCO Model

- Plain Old CLR Objects

- Ref: http://www.entityframeworktutorial.net/