

Entity Framework

Steven Bucksbaum, February 28, 2011

An Outline

Need to close the impedance mismatch between code and data

1. As programming progressed from structured programming to object oriented programming to functional programming, data has remained in the same: stored in tables in row- column format
2. Closing the programming-data gap
 - a. Entity Framework
 - b. Language-Integrated Query (LINQ)
 - c. Entity SQL
3. Entity Framework
 - a. Model the nouns (entity types) on a design surface.
 - b. Model relationships (associations) between entities
 - c. In our code we program the entities and associations.
4. Modeling
 - a. Entity Data Model (EDM): formal structure for defining data used in the applications created with the Entity Framework
 - b. EDM defines the data types, what types of relationships are allowed, the schemas that support the model and the mapping between the schemas.
 - c. Each model has 3 layers
 - i. *Conceptual layer*- what most developers see, uses Conceptual Schema Definition Language (CSDL)
 - ii. *Storage layer* – defines the data store, includes the tables, columns and data types. Syntax is the Schema Definition Language (SDL)
 - iii. *Mapping layer*- mapping between the conceptual layer and the storage layer. Defines how properties on entities map to columns on tables. Mapping Specification Language (MSL)
5. Terminology
 - a. *EntityType* – properties: named value with specific data type
 - b. *Association* between two EntityTypes: in terms of 0:M or 1:M or M:M
 - c. *EntityKey* – unique identifier
 - d. *EntitySet* – holds instance of an EntityType or one of its derived types.
 - e. *ComplexType* – used to group related properties together to be reused in a model.
6. Code
 - a. Creating classes to implement EntityTypes
 - b. Workflow is managed by Windows Workflow Foundation (WF)
 - c. Database First
 - i. Process that starts coding from the database up to the user interface.