

A

- abstract class 268
 - See also* class, abstract
- AcceptAllChanges
 - method 155, 160, 179, 525
- AcceptChanges method 161, 179
- Accessibility class 329
- AddObject method 57, 155–157, 180
- ADO.NET provider, connecting to 247–248
- ADO.NET Self-Tracking Entities Generator 414
- aggregate 367
 - and factories 376
 - and repositories 371
- aggregation methods 85
- annotation, custom 317, 346
 - generating data annotations from 333–335
 - retrieving 319
- anonymous method 504
- anonymous type 91, 497, 508–509
 - assign name to property of 91
 - limitations 509
 - nesting 91
- application
 - creating 42–55
 - design process 358–359
 - developing using database-like structures 5–10
 - layer 358
 - loose typing problem 9–10
 - performance problem 10
 - strong coupling problem 8–9
 - three-layer architecture 358
- application layer 364
- ApplyChanges method 417–419
- ApplyCurrentValues
 - method 155, 158–159, 183, 185, 194, 402
 - performance issue 186
- ApplyOriginalValues
 - method 155, 158–159
- architecture, three-layer. *See* three-layer architecture
- Arrange, Act, Assert (AAA) pattern 452, 460
- AsEnumerable method 489
- ASP.NET
 - and Entity Framework, common scenarios 393
 - and transactions and concurrency 394
 - attaching entities 394
 - presentation logic 360
 - assembly, creating 42–43
- AssemblyCleanup attribute 455
- AssemblyInitialize
 - attribute 455
- Assert class 454
 - AreEqual method 453
- association
 - creating 50–51
 - definition 365
 - deleting 49
 - implementing 136–144
 - independent 42, 52, 170
 - projecting with 92–95
 - sorting with 101–102
- association mismatch 16–18
 - many-to-many relationships 17–18
 - one-to-many relationships 17
 - one-to-one relationships 16–17
- AssociationSet class 315
- AssociationType class 302
- Attach method 59, 154–155, 157–158
- AuditableValue class 518
- auditing 516–526
 - creating attribute for entity 517
 - creating custom context 517, 520–525
 - custom property getter in designer 520
 - custom property in designer 517–520
 - custom property setter in designer 520
 - customizing context
 - template 517, 525–526
 - customizing designer 517
 - customizing entities
 - template 517, 520
- ExtendedObjectContext class 523
- ExtendedObjectContext constructor 525
- IObjectPersistenceNotification interface 521, 523, 525
- PersistenceNotification class 521, 523, 526

B

binding

- adding data 440
 - complex properties in XAML 441
 - data source 433, 435
 - data to context in WPF 442
 - deleting and adding entities in WPF 445
 - deleting detail data 439
 - deleting detail entities in WPF 445
 - deleting master data 438
 - details 437
 - error management 431
 - in Windows Forms applications 432–441
 - in WPF applications 441–446
 - interfaces 440–441
 - lookup combo box 435
 - lookup combo box in WPF 443
 - master/detail 436
 - master/detail in WPF 443
 - modifying data 438
 - transactional entities 427
 - Windows form 426
- BindingList class 434
- BindingSource
- component 433, 435
- bottom-up design
- introduction 35
 - reasons to use 36
 - vs. top-down 36–37
- bridge class 348
- builder method 64
- business layer 358
- and business rules 361
 - interactions with other layers 362
 - responsibility 360
- Button control 361

C

C#

- anonymous method 504
- anonymous type 497, 508–509
- extension method 496–503
 - consuming 500
 - creating 500
 - importing 500
- lambda expression 496, 504–507

- object initializer 496, 507–508
 - and anonymous types 508
 - query syntax 497, 509–510
 - type inference 496, 505
 - using 500
 - var 497
- CancelEdit method 440
- casting, cost of 9
- change tracking 152, 167–170
- and MergeOption 174–175
 - attaching or adding graph 171
 - detecting collection properties modification 194
 - disabling 475, 491
 - entity not wrapped inside proxy 167–169
 - entity outside context 171
 - entity wrapped inside proxy 169–170
 - managing with ObjectStateManager 162–175
- ChangeObjectState
- method 155, 161, 183, 513
 - data loss 184
- ChangeState method 155, 161
- ChangeTracker property 415
- class
- abstract 37, 47
 - See also* abstract class
 - benefits of using 11
 - compile-time checking 11
 - creating 123–136
 - describing in conceptual schema 127–130
 - designing for binding 425–432
 - ease of development 11
 - extending through partial classes 335–336
 - isolation from fetching problems 14
 - moving to object model 13–15
 - name
 - pluralizing 45
 - singularizing 45
 - organizing data with 10–15
 - partial 335
 - creating 335
 - query 336
 - properties, filtering on 106
 - representing data with 11–13
 - storage-agnostic interface 11
 - strong typing 11
 - writing 124–126
- ClassCleanup attribute 455
- ClassInitialize attribute 455
- code
- generated. *See* generated code
 - organizing with POCO template 53
- code first 44
- CodeGenerationTools
- class 328
- column
- discriminator. *See* discriminator column
 - identity. *See* identity column
 - versioning 10
- command tree, introduction 71
- CompiledQuery class 488
- Invoke method 489
 - MergeOption 490
 - recompilation 489
- complex property
- persistence 183
 - See also* property, complex
- complex type 37
- creating 47
- ComplexType class 302, 310
- composition, definition 366
- Computed property 200
- concurrency
- and inheritance 212–213
 - automatic management 220
 - building comparison form 216–220
 - concurrent updates 204–205
 - connected scenario 209
 - disconnected scenario with ApplyCurrentValues method 210–212
 - disconnected scenario with ChangeObjectState method 209–210
 - enabling optimistic concurrency check 208–209
 - exceptions
 - catching 213–214
 - managing 214–220
 - handling 208–220
 - handling graphs 212
 - managing during persistence 204
 - optimistic approach 206
 - advantages 207

concurrency (*continued*)
 optimistic, in action 209–213
 pessimistic control 208
 pessimistic lock 205
 pessimistic/optimistic
 approach 207
 problem 204–207
 optimistic control 206–207
 pessimistic control 205–206
 pessimistic/optimistic
 control 207
 refreshing database
 values 215
 using all columns for
 versioning 207
 using version column 206,
 209
 concurrency exception 179
 connection string
 creating in code 68–69
 metadata 67
 name 66
 provider 67
 providerName 67
 setting up 66–69
 container, populating 313–315
 Contains method 375
 context 30
 attaching entity as modified
 to 514–515
 creating proxied entity 168
 Identity Map pattern in
 72–74
 lifecycle in Windows
 form 434
 managing entities returned
 by queries 72
 context entity 158
 ContextOptions.ProxyCre-
 ationEnabled property 77,
 405
 Context-per-Form pattern 434
 Context-per-Request
 pattern 389–390
 contract 399
 converting, cost of 9
 CreateChangeScope
 method 348
 CreateEdmItemCollection
 method 328
 CreateObjectSet method 65
 and tracking 74
 CreateQuery method 229
 CSDLToSSDLAndMSLActivity
 activity 338

D

data
 aggregated, filtering 100
 fetching. *See* fetching
 filtering 81–89
 based on associations
 82–86
 grouping. *See* grouping
 retrieving from database
 71–72
 sorting. *See* sorting
 data access layer 358, 461
 Entity Framework as 361
 interactions with other
 layers 362
 querying 361
 simplifying 57
 testing 461–469
 data access, introduction 4–5
 data adapter
 creating 6
 using with dataset 5
 data annotation, attributes 333
 data binding 360
 with EntityDataSource
 379–383
 data container 5–8
 limitations 8–10
 data layer, introduction 12
 data reader
 inability to transparently
 retrieve data 8
 returning column index 9
 typed methods 9
 using as data container 5–8
 Data Source Configuration
 Wizard 433
 data source controls 379–383
 and DataBind method 380
 practical guide to 379–380
 data table 6
 inability to transparently
 retrieve data 8
 pouring data into 6
 data transfer object 93, 399
 database
 association mismatch 16
 associations, unique and
 bidirectional 17
 connecting to 6
 creating before model 36
 deleting objects 59–60
 connected approach 59
 disconnected approach 59
 deletion, triggering 438
 describing in storage
 schema 130–133
 designing 37–42
 entity state effect on 153
 foreign key. *See* foreign key
 granularity mismatch 18
 handling mismatches 22
 identity mismatch 21
 importing 44–46
 impossibility of representing
 inheritance graphs 20
 inheritance mismatch 20
 inserting objects 57
 managing from the
 context 79
 modifying, performance
 comparison 479–481
 natural key. *See* natural key
 never forget about 36
 number of tables vs. number
 of classes 18
 persisting entities into
 180–187
 querying 55–56
 retrieving data from 71–72
 special types 528–531
 handling using a view 528
 handling using defining
 query 529
 structure, code bound to 7
 surrogate key. *See* surrogate
 key
 transaction
 commit or rollback 179
 starting 178
 updating 56–60
 updating objects 58
 connected approach 58
 disconnected approach 58
 values, refreshing 215
 Database Generation Workflow
 property 337
 Database Schema Name
 property 338
 database-first design. *See* bot-
 tom-up design
 DataCollection attribute 415
 DataColumn, introduction 10
 DataContract attribute 405,
 415
 DataContractResolver class 400
 DataGrid element, Binding
 attribute 441

- DataMember attribute 405, 408, 415
 - DataReader
 - casting and conversion issues 10
 - introduction 10
 - DataSet 363
 - casting and conversion issues 10
 - structure of 10
 - dataset
 - using as data container 5–8
 - vs. recordset 4
 - dataspace 297
 - DataSpace enum class 300
 - DataTable 363
 - introduction 10
 - datatype mismatch 15–16
 - DbDataAdapter,
 - introduction 10
 - DbDataReader, transforming
 - into objects 249–250
 - DbDataRecord class 233, 293
 - switching to typed object 293
 - DbParameter class 111
 - DDD. *See* domain-driven design
 - DDL
 - generating with designer 336–339
 - generation,
 - customizing 339–344
 - DDL Generation Template
 - property 338
 - Debug class, Assert
 - method 453
 - DefiningQuery 265, 285–288, 529
 - mapping stored procedure to
 - classes with complex property 287–288
 - relationships 287
 - update through
 - functions 287
 - delete cascade 51, 197
 - DeleteObject method 59, 155, 159–160, 173, 187, 438
 - design
 - database-first. *See* bottom-up design
 - model-first. *See* top-down design
 - design, bottom-up. *See* bottom-up design
 - design, top-down. *See* top-down design
 - designer
 - extending 344–352
 - factory class 348–350
 - installing, debugging, and uninstalling 352
 - manifest file 351
 - property class 346–348
 - required classes 345
 - setting up project 345–346
 - extensions 336
 - debugging 352
 - installing 352
 - uninstalling 352
 - generating SSDL, MSD, and DDL 338–339
 - workflow
 - choosing 337–338
 - TablePerTypeStrategy.xml 337
 - Detach method 155, 161–162
 - detail, deleting 439
 - DetectChanges method 421, 525
 - DiffDays method 108
 - discriminator 49
 - discriminator column 38, 48, 268–269
 - Domain Driven Design* (Eric Evans) 14
 - domain layer 364
 - domain model
 - anemic 369
 - complex 376
 - designing 363
 - refactoring 370
 - referencing in unit tests 449
 - vs. object model 14
 - Domain Model pattern 14
 - domain root 371
 - domain service
 - criteria for using 371
 - definition 370
 - stateless 371
 - domain-driven design 363–371
 - aggregate 367–368
 - and foreign keys 377
 - association 365
 - collection 371
 - composition 366
 - domain root 367–368
 - domain service 370
 - entities 364–365
 - getting reference to 376
 - factory 376
 - identity field 365
 - layers 364
 - master-detail
 - relationship 367
 - model, refining 368–371
 - repository 368, 371
 - implementing 372–376
 - retrieving references to
 - domain entities 371
 - value object 365–366
 - DTC. *See* Microsoft Distributed Transaction Coordinator
 - DTO 404
 - benefits of using 412
 - developing service 409–412
 - persisting complex graph 411–412
 - See also* data transfer object
 - Dynamic Data controls 384–388
 - data annotations 386–388
 - DataTypeAttribute class 388
 - DisplayNameAttribute class 388
 - RegularExpressionAttribute class 388
 - RequiredAttribute class 388
 - Dynamic Data Entities Web Application 384
 - EntityDataSource
 - control 384
 - LinqDataSource control 384
 - MetadataTypeAttribute
 - class 386
 - registering model 384–386
 - scaffolding 386
- E**
-
- eager loading 113–115
 - navigation path 114
 - EDM. *See* Entity Data Model
 - EdmFunction attribute 289, 291
 - EdmFunction class 302
 - EdmGen tool 485
 - EdmItemCollection class 299, 328
 - EDMX file 122
 - EF Mapping Verifier 472
 - Eini, Oren 458
 - entities graph
 - persisting 187–199
 - persisting deletions 196–198

- entities graph (*continued*)
 - persisting graph of added entities 188–192
 - using foreign-key associations 188–189
 - using independent associations 190–192
 - persisting graph of entities in different states 192
 - persisting many-to-many relationships 199
 - persisting
 - modifications 192–196
- entity
 - adding to context 155–157
 - associating not-tracked entity to tracked one 195
 - attaching 512–516
 - attaching as Modified or Deleted 513–514
 - attaching entities in ASP.NET 394
 - attaching to context 157–158
 - committing 179
 - context entity 158
 - creating 123–136
 - creating from scratch 46–49
 - definition 365
 - describing in conceptual schema 127–130
 - description, key 286
 - designing 37–50
 - detaching from context 161–162
 - dirty, detecting 177–178
 - getting reference to 376
 - lifecycle 152–155
 - mapping to database 47–49
 - mark as Unchanged 191
 - marking as modified 183
 - marking property as modified 186
 - modifications
 - persisting 438–440
 - persisting in WPF 445–446
 - rolling back 427
 - modifying 49–50
 - modifying selected properties
 - after attaching 514–516
 - persisting added entity 180–182
 - persisting added graph with foreign-key association 188–189
 - persisting added graph with independent association 190–192
 - persisting deletion 187
 - persisting deletions in graph
 - using foreign-key association 196–197
 - persisting deletions in graph using independent association 197–198
 - persisting into database 180–187
 - persisting many-to-many relationships 199
 - persisting mixed graph 192
 - persisting modified entity 182–187
 - persisting modified graph with foreign-key association 193–195
 - in connected scenario 193
 - in disconnected scenario 193
 - persisting modified graph with independent association 195–196
 - persisting with SaveChanges 177–180
 - plain 167
 - proxied 167
 - enabling change tracking 169
 - proxy. *See* proxy entity
 - returned by a query 76–77
 - self-tracking. *See* self-tracking entity
 - set, multiple per type. *See* multiple entity set per type
 - single, retrieving 87–89
 - with context methods 88–89
 - state. *See* EntityState
 - stub, creating 190
 - vs. value object 367
 - writing 124–126
- Entity Client 31, 64
 - and Object Services 74–75
 - data provider 246–251
 - querying directly 31
- Entity Data Model 27–29
 - and Visual Studio designer 120–123
- Association element in CSDL 138
- AssociationSet element in CSDL 137
- SSDL 139
- AssociationSetMapping element in MSL 143
- CollectionType element in CSDL 295
- CommandText element in SSDL 275
- ComplexProperty element in MSL 135, 279
- ComplexType element in CSDL 127–128
- conceptual model 28
 - splitting 28
- conceptual schema 120, 127–130
- Condition element in MSL 147
- custom annotations 149–150
- DefiningExpression element in CSDL 291
- DefiningQuery element in SSDL 286
- DeleteFunction element in MSL 279
- Dependent element in CSDL 139
- End element in CSDL 138
- EndProperty element in MSL 143
- EntityContainer element in CSDL 127–128
- SSDL 139, 286
- EntityContainer element in CSDL 127
- EntityContainerMapping element in MSL 134
- EntitySet element in CSDL 128
- SSDL 131, 286
- EntitySetMapping element in MSL 134, 146
- EntityType element in CSDL 127, 129
- SSDL 132–133
- EntityTypeMapping element in MSL 134, 146, 148, 268, 279
- Function element in CSDL 291
- SSDL 256, 275
- FunctionImport element in CSDL 267
- SSDL 257

- Entity Data Model (*continued*)
 - FunctionImportMapping element in MSL 258, 264, 268
 - InsertFunction element in MSL 279
 - introduction 120–123
 - Key element 129
 - in CSDL 129
 - location 122
 - many-to-many relationship 142–143
 - in MSL 143
 - mapping 28–29
 - Mapping element in MSL 134
 - mapping schema 120, 133–136
 - MappingFragment element in MSL 134, 146, 148
 - metadata 122
 - ModificationFunctionMapping element in MSL 279
 - NavigationProperty element in CSDL 137, 142
 - one-to-many relationship 140–142
 - in CSDL 142
 - in the model 141–142
 - one-to-one relationship 136–140
 - in CSDL 137–139
 - in MSL 140
 - in SSDL 139–140
 - in the model 136
 - Parameter element in CSDL 257, 291
 - SSDL 256
 - Principal element in CSDL 139
 - Property element in CSDL 128
 - PropertyRef element 129
 - in CSDL 139
 - ReferentialConstraint element in CSDL 139
 - relationship tips 143
 - ResultMapping element in MSL 268
 - ReturnType and RowType 293
 - ReturnType element in CSDL 295
 - RowType element in CSDL 295
 - ScalarProperty element in MSL 135, 143, 279
- Schema element in CSDL 127
 - SSDL 340
- Schema element in CSDL 127
 - SSDL EntityContainer element 131–132
 - SSDL Schema element 131
- storage model 28
- storage schema 120, 130–133
- table-per-hierarchy mapping 144–147
 - design model class 144–145
 - discriminator column 147
 - in MSL 145–147
- table-per-type mapping 147–149
- UpdateFunction element in MSL 279
 - verbosity of 28
 - XML files 27
- Entity Data Model Designer
 - Extension Starter Kit 351
- Entity Designer Database Generation Power Pack 43
- Entity Designer Database Generation toolkit 336
 - customizing DDL generation 339–344
- Entity Framework
 - and ASP.NET, common scenarios 393
 - architecture 26
 - as part of infrastructure layer 364
 - benefits of using 23–26
 - current version solves past problems 25
 - database vendor independence 25
 - inclusion in .NET Framework 24
 - integration into Visual Studio 25
 - LINQ as query language 25
 - recommended for data access 26
 - current version 25
 - data access 26–32
- Entity Client. *See* Entity Client
- Entity Data Model. *See* Entity Data Model
- Entity SQL. *See* Entity SQL
- LINQ to Entities. *See* LINQ to Entities
- Object Services. *See* Object Services
 - query languages 27
 - vs. NHibernate 24
- Entity Framework designer, limitations 43
- entity set 372
 - introduction 64
- Entity SQL 27, 32, 64
 - AS clause 228
 - associations
 - filtering with collection associations 231
 - filtering with single associations 230
 - navigating 230
 - working with 230–232
 - automatic joins 231
 - complexity of 32
 - COUNT function 231
 - defining variable 229
 - DISTINCT function 231
 - enabling plan caching 490–491
 - EXISTS clause 231
 - filtering data 230–232
 - FROM clause 228
 - GROUP BY clause 237
 - grouping data 237–239
 - and filtering 238
 - and projecting 238
 - handling complex properties in code 235
 - HAVING clause 238
 - history 228
 - injection 245
 - JOIN clause 240
 - joining data 240
 - LIMIT clause 232
 - navigating associations 235
 - navigating complex properties 230
 - OFTYPE function 240
 - ORDER BY clause 232, 239
 - parameters 230
 - plan caching 475
 - projecting concatenated columns 234
 - projecting data 232–237
 - handling results 233–235
 - with associations 235–237
 - with collection associations 236–237
 - with single associations 235–236
 - projecting into complex properties 234

- Entity SQL (*continued*)
 - projecting to DTO 233
 - query, basics 228–229
 - query-builder methods 245
 - chaining 242–243
 - creating query at runtime 243
 - defining variable 242–243
 - preventing injection 244–245
 - using parameters 244–245
 - vs. LINQ to Entities methods 243–244
 - querying for
 - inheritance 240–241
 - results, paging 232
 - returning single value from
 - collection association 237
 - ROW function 234, 293
 - SELECT clause 228, 232
 - SKIP clause 232
 - sorting data 239–240
 - based on associations 239–240
 - SUM function 231
 - syntax, vs. LINQ to Entities 231
 - vs. LINQ to Entities 32, 251
 - WHERE clause 230
 - EntityCollection 53
 - EntityCommand class 246, 248
 - EnablePlanCaching property 248, 490
 - ExecuteNonQuery method 251
 - ExecuteReader method 248
 - EntityConnection class 66, 246–248, 298
 - andObjectContext 74
 - connection string 247
 - GetMetadataWorkspace method 248
 - StoreConnection property 247
 - EntityConnectionStringBuilder class 68–69
 - EntityContainer class 302
 - EntityDataReader class 246, 248–250
 - EntityDataSource control 379–383
 - binding 381–383
 - ConnectionString property 383
 - Data Source Configuration Wizard 381
 - DefaultContainerName property 383
 - EntitySetName property 383
 - in ASP.NET 3.5 380
 - in depth 380–383
 - ObjectContext 382
 - EntityDesignerExtendedProperty attribute 349
 - EntityDesignerExtendedProperty class 517
 - EntityDesignerSelection enum 350
 - EntityEntry class 163, 199
 - EntityFunctions class 108
 - mathematical functions 109
 - EntityKey class 88, 166
 - EntityObject 52
 - EntityParameter class 246, 248
 - EntityReference 53
 - EntitySet class 315
 - EntityState 152–153
 - Added 152, 154
 - changing 153–162
 - database impact 153
 - Deleted 152, 155
 - Detached 152–153
 - managing 155
 - Modified 152, 154
 - modifying from entry 166–167
 - Unchanged 152, 154
 - vs. entry state 166
 - EntityTransaction class 246
 - EntityType class 302, 310
 - entry, changing original values 211
 - equality by reference 21
 - Equals method 125
 - and GetHashCode 195
 - ErrorProvider component 440
 - Evans, Eric 25
 - Domain Driven Design* 14
 - event handler, and business logic 361
 - Execute method, and tracking 74
 - ExecuteFunction method 258, 283, 492
 - and multiple resultsets 274
 - ExecuteStoreCommand method 201, 282
 - ExecuteStoreQuery method 110, 112, 260
 - overloading 111
 - execution, deferred 510–511
 - expandability 358
 - ExpectedException
 - attribute 453
 - Export attribute 349
 - extensibility 495
 - Extension Manager 53
 - extension method 466, 496, 498
 - consuming 500
 - creating 500
 - importing 500
 - extension, adding 53
- F**
-
- factory
 - definition 376
 - using Entity Framework's foreign keys 377
 - factory class 345
 - creating 348–350
 - fault injection 450
 - fetching 113–118
 - and Include method 114
 - eager loading 113–115
 - lazy loading 113, 115–116
 - manual deferred loading 116–118
 - strategy, choosing 118
 - filtering
 - based on properties of base class 106
 - based on type 105
 - First method 56
 - fluent interface 460
 - foreign key
 - association 42, 52
 - in database design 40
 - introduction 17
 - mapping 51
 - property 136
 - foreign-key property 377
 - Fowler, Martin 25
 - full-text search 528
 - function 107–109
 - canonical 108–109
 - and mathematical functions 108
 - custom database 108
 - custom in SSDL 281
 - database 108–109
 - defining in conceptual schema 257
 - defining in SSDL 275
 - embedding in storage model 274–275

function (*continued*)
 generating 329–333
 model defined 108
 populating 312–313
 scalar-valued. *See* scalar-valued function 288
 user-defined. *See* user-defined function
 FunctionImportParameter class 333

G

Generate Database Script from Model 54
 Generate T-SQL Via T4 (TPH) .xaml file 339
 generated code, organizing 52–54
 Geometry data type 528
 GetAllEntityType method 340
 GetHashCode method 125 and Equals 195
 GetItem method 301, 305
 GetItemCollection method 301, 304
 GetItems method 301, 303–304
 GetMetadataWorkspace method 298
 GetObjectByKey method 88, 319
 GetObjectStateEntries method 165
 GetObjectStateEntry method 166
 GetUpdatableOriginalValues method 211, 218
 granularity mismatch and inheritance 20–21 and many-to-many relationships 18 and value types 18–20
 grouping 96–100
 grouped data
 changing name of 97
 filtering 100
 projecting 99
 key 98

H

HttpModule 392

I

ICollection interface 372
 Add method 372
 Remove method 372
 IDataErrorInfo interface 425, 429–431
 customizing template 432
 default property 430
 Error property 430
 ErrorProvider component 440
 implementing 430
 identity column 38
 identity field 365
 identity map 72
 Identity Map pattern 72–74
 identity mismatch 21–22
 IDisposable interface 66
 IDynamicDataSource interface 380
 IEditableObject interface 425–429, 440
 BeginEdit method 428
 CancelEdit method 428
 customizing template 432
 EndEdit method 428
 inheritance 429
 IEntityDesignerExtendedProperty interface 344, 348, 350
 CreateProperty method 350
 IEnumerable interface 70, 502
 IEqualityComparer interface 195
 IGrouping interface 96
 IModelConversationExtension interface 344
 IModelGenerationExtension interface 344
 IModelTransformExtension interface 344
 Include method 56, 114
 improving 527–528
 with lambda expression 527
 independent association. *See* association, independent
 information, selected, showing 437
 infrastructure layer 364
 and foreign keys 377
 Entity Framework as part of 364
 repository

implementation 372

inheritance
 adding 49
 and concurrency 212–213
 and polymorphic queries 105
 mapping 144–149
 persisting 37–38
 See also table per concrete type
 querying with 105–107
 inheritance mismatch 20–21
 InnerException class 199
 INotifyPropertyChanged interface 416, 425–426
 advantage of 440
 automatic properties 426
 customizing template 431
 NotifyPropertyChanged event 426
 PropertyChanged event 426
 integration test 449, 461
 IntelliSense 11
 InternalsVisibleTo attribute 457
 Inversion of Control 376
 Unity container 389
 IObjectChangeTracker interface 415–416
 IObjectContext interface 463
 IObjectPersistenceNotification parameter 525
 IObjectSet interface 157, 463, 467
 IOrderedQueryable interface 102
 IQueryable interface 70, 390
 IQueryProvider interface 70
 IRepository class 374
 ISet interface 141
 isolation level 223
 IsolationLevel enum 223
 ItemCollection class 328

J

joining 102–105
 group join 104
 JUnit 451

K

key, foreign. *See* foreign key

L

lambda expression 496, 504–507

layer 358

lazy loading 113, 115–116

- enabling 77
- switching off 116

link table 40

- introduction 17

LINQ

- chaining methods 85
- Enumerable class 500
- flavors 496
- functional
 - programming 496
- method chaining 501–502
- method evaluation 502–503
- methods list 501
- provider architecture 496
- query execution 510
- reasons for creation of 494–497
- runtime query
 - composition 511
- unified query language 494
- unified query result 495

LINQ in Action 496

LINQ to DataSet 496

LINQ to Entities 27, 31, 64

- All method 84
- Any method 84
- Cast method 106
- Contains method 82
- Count method 85
- Distinct method 85
- dynamic querying 89
- execution 77–78
- filtering 81
 - on collection
 - associations 84–85
 - on single reference
 - association 82–83
- First method 87
 - vs. Single 88
- FirstOrDefault method 87
- Group By and groupby
 - methods 96
 - into 98
- grouping 96–100
- grouping properties 91
- join method 103
- joining 102–105
 - group join 104

OfType method 105, 107

orderby and Order By

- methods 87, 100

paging results 86

projection 90–96

- with collection
 - association 94
 - with single association 92

queries

- compiling 474, 487–490
- unit testing 466–469
- vs. standard LINQ
 - queries 70

querying with

- inheritance 105–107

Select method 85, 92

Single method 87

- vs. First 88

SingleOrDefault method 88

Skip method 86

sorting 100–102

- with associations 101–102

Sum method 84

syntax, vs. Entity SQL 231

Take method 86

vs. Entity SQL 32, 251

Where method 81

LINQ to Objects 496

LINQ to SQL 496

- future of 81

LINQ to XML 496

LinqDataSource control 379–380

ListBox 360

LoadProperty method 116

logic contract, n-tier

- application 398

loose typing problem 9–10

M

maintainability 358

- as benefit of using O/RM 24

Managed Extensibility Framework 344

manifest file 351

manual deferred loading 116–118

many-to-many relationship, mapping 52

mapping

- Mapping Details window 48
- table per hierarchy 48

mapping file 29, 133–136

Mapping Specification Language (MSL). *See* MSL

MarkAsAdded method 416

MarkAsDeleted method 416

MarkAsModified method 416

MarkAsUnchanged

- method 416, 420

master-detail relationship 367

- avoiding 368

materialization 27

Math class 109

max length 47

MEF. *See* Managed Extensibility Framework

MergeOption class 491

MergeOption enum 174–175, 259

- AppendOnly value 174
- NoTracking value 174
- OverwriteChanges value 174
- PreserveChanges value 174

MergeOption property

- AppendOnly 73
- NoTracking 73
- OverwriteChanges 73
- PreserveChanges 73

MEST. *See* multiple entity set per type

metadata

- accessing 297–299
 - using connection 298
 - using context 298
 - using, Metadata-Workspace 299
- availability 301
- basics 297–301
- building explorer 306–316
- extracting from EDM 303–306
 - with GetItem and TryGetItem 305–306
 - with GetItemCollection and TryGetItemCollection 304
 - with GetItems 303–305
- force loading 301
- internal organization 300
- loading 301
- object model 302
- primitive types 303
- retrieving 301–306
- retrieving complex types 311
- retrieving containers 314
- retrieving entities 306–307

- metadata (*continued*)
 - retrieving entity base type 307–308
 - retrieving entity derived types 308
 - retrieving entity foreign-key properties 309
 - retrieving entity key properties 309
 - retrieving entity properties 308–311
 - retrieving functions 312
 - retrieving storage schema 315–316
 - writing generic code 316–320
 - Metadata Artifact Processing property 123
 - MetadataItemCollection-Factory class
 - CreateStoreItemCollection method 329
 - MetadataLoader class 328–329
 - MetadataTools class 328
 - GetElementType method 333
 - MetadataWorkspace class 297, 299
 - constructor 299
 - RegisterItemCollection method 299
 - MetadataWorkspace property, ObjectContext 298
 - method
 - aggregation family 85
 - chaining 85
 - projection family 85
 - set family 84
 - Microsoft Distributed Transaction Coordinator 221
 - Microsoft Unit Testing Framework 453–455
 - mocking framework 458–461
 - model
 - creating before database 36
 - defining relationships 136–144
 - designing 37–42
 - Model Browser window 47
 - model. *See* object model
 - model-first design. *See* top-down design
 - MSDTC. *See* Microsoft Distributed Transaction Coordinator
 - MSL, generating in designer 55
 - MSTest 451
 - multiple entity set per type 128
- N**
-
- namespace, retrieving 330
 - natural key, avoiding 21
 - navigation path 114
 - navigation property 41
 - and independent association 42
 - navigation, conceptual 82
 - NHibernate, vs. Entity Framework 23
 - Nilsson, Jimmy 25
 - n-tier application
 - choosing data to exchange between client and server 398–399
 - logic contract 398
 - problems and solutions 397–400
 - serialization 399–400
 - tracking changes on client 397–398
 - NUnit 451
- O**
-
- O/RM
 - and maintainability 24
 - and performance 24
 - and productivity 24
 - choosing whether to use 26
 - introduction to 23
 - object
 - benefits of using 5
 - deleting 59–60
 - connected approach 59
 - disconnected approach 59
 - equality 5
 - inheritance 5
 - inserting 57
 - modifying 56–60
 - relationships 5
 - tracking
 - and projection 95–96
 - n-tier application 397–398
 - updating 58
 - connected approach 58
 - disconnected approach 58
 - vs. dataset 4
 - object initializer 496, 507–508
 - and anonymous types 508
 - object materialization 30
 - object model 359, 361, 363
 - introduction 13–15
 - moving to domain model 14
 - vs. domain model 14
 - Object Model pattern 14
 - object persistence 5
 - Object Services 27, 29–30, 64–79
 - and Entity Client 74–75
 - context 30
 - data organization 30
 - object materialization 30
 - query transformation 29
 - tasks 66
 - object tracking. *See* change tracking
 - object/relational mapping. *See* O/RM
 - object/relational mismatch 5, 15–22
 - example 8
 - ObjectChangeTracker class 415
 - AcceptChanges method 417
 - ChangeTrackingEnabled property 415
 - ExtendedProperties property 415–416
 - ObjectsRemovedFromCollectionProperties property 415–416
 - ObjectState property 415–416
 - OriginalValues property 415–416
 - StartTracking method 417
 - StopTracking method 417
 - ObjectContext 30
 - ContextOptions.UseLegacy-PreserveChangesBehavior property 174
 - ObjectContext class 64–66, 153, 155, 463
 - and EntityConnection 74
 - Connection property 75
 - ContextOptions.LazyLoading-Enabled property 116
 - Context-per-Request pattern 390
 - CreateDatabase method 79
 - CreateDatabaseScript method 79
 - CreateObject method 377

- ObjectContext class (*continued*)
 - DatabaseExists method 79
 - decoupling from ASP.NET 393
 - DeleteDatabase method 79
 - implementing interface on 464
 - lifecycle in ASP.NET 388–393
 - using HttpModule 392–393
 - n-tier application 397
 - Refresh method 214
 - Translate method 249–250
 - unit test 466
 - wrapping for repository 390–391
 - ObjectDataSource
 - control 379–380
 - ObjectMaterialized event 71, 73
 - ObjectParameter class 230, 245, 272
 - ObjectQuery class 65, 76, 88, 241
 - CommandText property 242
 - objects graph. *See* entities graph
 - ObjectSet class 64–65, 74, 153, 155
 - AddObject method 462
 - EnablePlanCaching property 490
 - Execute method 434
 - queries written against 71
 - unit test 466
 - ObjectsRemovedFromCollectionProperties
 - property 416
 - ObjectStateEntry class 163–164
 - AcceptChanges method 167, 217
 - ChangeState method 167, 171
 - CurrentValues property 163
 - Delete method 167
 - Entity property 163
 - EntityKey property 163
 - automatic creation 164
 - temporary 164
 - EntitySet property 163
 - EntityState property 163
 - entry, retrieving 164–166
 - GetModifiedProperties method 163
 - IsRelationship property 163
 - members 163
 - methods that modify entity state 167
 - OriginalValues property 163
 - SetModified method 167
 - SetModifiedProperty method 167
 - ObjectStateManager class 27, 162–175, 464
 - accessing 162
 - adding an element in collection properties with foreign-key association 173
 - adding an element in collection properties with independent association 173
 - automatic entry-entity synchronization 169
 - ChangeRelationshipState method 171
 - DetectChanges method 168
 - identifying object by key 163–164
 - lifecycle 162
 - monitoring entity modification 167–170
 - relationship change 170
 - relationship for collection association 170
 - relationship in collection property 172–173
 - relationship in single-reference property 172
 - relationship management during attach 170
 - relationship with foreign key 171
 - relationship with independent association 170
 - removing element in collection properties with foreign key 173
 - retrieving single entry 166
 - ObservableCollection
 - class 442, 445
 - CollectionChanged event 445
 - OfType method 104
 - ObjectStateManager class
 - removing element in collection properties with independent association 173
 - OptimisticConcurrencyException class 213
 - OrderDetails property 141
 - OrderIT
 - model and database, designing 35–42
 - use cases 35
 - OrderIT example
 - introduction 34–35
 - requirements 34
 - OriginalValueRecord class 211, 218
 - output pragma 327
 - overnormalization, avoiding 19
- P**
-
- paging results. *See* result, paging parameter
 - classic 112–113
 - numbered lists 111–112
 - working with 111–113
 - partial class 529
 - performance
 - as benefit of using O/RM 24
 - bulk insert 481
 - comparison between compiled queries and ADO.NET 488
 - compiling LINQ to Entities query 488–489
 - database-modification comparison 479–481
 - with ADO.NET 481
 - disabling change tracking 475, 491
 - for stored procedure 492
 - enabling plan caching 490–491
 - Entity SQL plan caching 475
 - LINQ to Entities query compilation 474
 - optimizing 484–492
 - compiling LINQ to Entities queries 487–490
 - pregenerating views 484–487
 - speeding up first execution 484–487
 - stored procedures 491–492
 - pregenerating view
 - with EdmGen 485
 - with template 486
 - query comparison 481–484
 - between Object Services and Entity Client layer 490
 - with ADO.NET with tracking disabled 491
 - with all techniques 484
 - with view pregeneration 485

performance (*continued*)
 stored-procedure comparison with ADO.NET 491
 test configuration 475
 timer 476–479
 visualizer 476

performance problem 10

persistence
 deletion
 in connected scenario 187
 in disconnected scenario 187
 executing custom SQL 200–201
 concurrency 201
 simplicity 201
 handling exceptions 199–200
 logical delete 522
 modifications
 in connected scenario 182–183
 in disconnected scenario 183–186
 optimized update command 183
 performing operation before and after 521
 tricks 199–201

persistence by reachability 57

persistence ignorance 11, 123

persistence test 470–472
 transactions in 471

plain entity. *See* entity, plain

plain old CLR object (POCO). *See* POCO

pluralizing object names 45

POCO (plain old CLR objects) 11, 53
 and persistence ignorance 123
 importance of 123
 template 322, 328–329
 installing 53
 organizing code with 53

polymorphic association 20

polymorphic query. *See* query, polymorphic

presentation layer 358, 364
 interactions with other layers 362
 logic 360
 triggering 361

productivity, as benefit of using O/RM 24

Profiler API 450

projecting results. *See* projection

projection 90–96
 and grouped data 99
 and object tracking 95–96
 nested 94
 with associations 92–95

projection methods 85

property
 complex 125
 navigation. *See* navigation
 property
 read-only 50
 scalar, adding 47
 virtual 126

property class 345
 creating 346–348

PropertyExtensionContext class 348

provider model 495

proxy entity 76
 disabling 77

proxy, and serialization 400

ProxyDataContractResolver class 400, 405
 SetResolver method 407

ProxyResolver attribute 407

Q

query
 and transactions 223–224
 compiled
 internals 489–490
 writing 488–489
 defining 285–288
See also DefiningQuery

different techniques 494

double execution 78

dynamic 89

entities returned by 76–77

executing with
 EntityCommand 248

manual 110–113

parameters 111–113
 classic 112–113
 numbered lists 111–112

performance
 comparison 481–484

polymorphic 105

results, processing with
 EntityDataReader 248–250

writing against classes 70

query syntax 497, 509–510
 limitation 510

query transformation 29

Queryable class, Where method 466

querying methods 64

querying, with inheritance 105–107

R

recordset, vs. dataset 4

refactoring, for testability 456–458

Reflector 76, 466

RefreshMode enum 215
 ClientWins 215
 StoreWins 215

relational database, working around inheritance 20

relationship tracking 170–173

relationship, designing 50–52

RelationshipEntry class 163, 170, 199

repository 368, 371, 389
 implementation, separating from interface 372
 implementing 372–376
 interface 372
 reasons to build 372
 test 462–466
 typical usage 376

Repository class 373
 constructor 373

result
 paging 86–87
 projecting. *See* projection

result column bindings 277

Rhino Mocks 458
 building mock 460
 building stub 459
 concrete entities 463
 VerifyAll method 461

robustness 358

S

Save button 438

SaveChanges method 57, 164, 168, 177–180
 committing entities 179
 database persistence 179
 detecting dirty entities 177–178
 generating SQL 178
 overriding 180, 520
 starting database transactions 178
 workflow 177

SaveOptions enum 179, 222

- SavingChanges event 164–165
- scalar property, retrieving in
 - template 334
- scalar-valued function 288–295
 - in LINQ to Entities 289
 - passing object as
 - parameter 292
 - returning a list of typed
 - objects 295
 - returning generic object 293
 - returning list of generic
 - objects 294
 - returning list of scalar
 - values 294
 - returning typed object 293
- search, full-text. *See* full-text search
- self-tracking entity 398
 - adding or deleting
 - entities 419
 - ADO.NET Self-Tracking Entities Generator 414
 - change tracking 413
 - complex graph 419
 - context 417
 - developing service 413–422
 - enabling 414–415
 - entity change tracker
 - 415–416
 - detecting modifications 416
 - generating 413
 - inner workings 415–417
 - internal state and context
 - state, automatic
 - synchronization 421
 - logic contract 414
 - managing entity state
 - 416–417
 - many-to-many
 - relationships 420–421
 - MarkAs* methods 416
 - POCO 415
 - pros and cons 421–422
 - proxy creation disabled 417
 - service proxy 417
 - using 417–421
- SerializableAttribute class 394
- serialization
 - and lazy loading 408
 - and proxies 400
 - ASP.NET ViewState 393
 - in WCF 405
 - lazy loading 400
- service
 - and DTOs 404
 - developing with DTOs
 - 409–412
 - developing with STEs
 - 413–422
 - entities as contract 400–409
 - interface 401
 - logic contract 403
 - optimizing data
 - exchanges 404–405
 - persisting complex
 - graph 403, 411–412
 - serialization 405–409
- set methods 84
- SetModifiedProperty
 - method 186, 402, 514
 - using lambda expression 514
- singularizing object names 45
- Smalltalk, unit testing in 448
- SmartAttach method 515
- sorting 100–102
 - with associations 101–102
- SpaceAfter method 328
- SpaceBefore method 328
- SQL
 - generated by Entity Framework 83
 - generated, capturing 75–76
 - monitoring with profiler
 - tool 75
- SQL Profiler, and transaction promotions 222
- SqlDataSource control 379
- SqlException class 191
- SqlFunctions class 109
- SqlGeometry .NET type 529
- SSDL, generating in
 - designer 54
- SsdlToDdlActivity activity 338
- state manager 27
 - customizing 30
- state manager component. *See* ObjectStateManager class
- STE. *See* self-tracking entity
- storage schema,
 - retrieving 329–330
- Store Schema Definition Language (SSDL). *See* SSDL
- stored procedure
 - binding to function in mapping schema 258
 - concurrency 279–280
 - configuring 279
 - defining in storage
 - schema 256–257
 - definition 285
- downgrading entity 282
- executing 258
- importing
 - manually 256, 258
 - using designer 254–255
- in conceptual schema. *See* function
- mapping 254, 258
- mapping persistence
 - 276–277
- mapping scalar resultset 266
- mapping to
 - class with complex
 - property 265
 - classes with complex
 - properties 287–288
 - complex type 261
 - complex type using
 - designer 262–264
- mapping TPH inheritance
 - hierarchy 268
- mapping TPT inheritance
 - hierarchy 269
- materialization of navigation
 - properties 260
- multiple resultset 274
- not connected to entity 282
- not lined up with entity 264
- output parameters 258, 271–274
 - in second resultset 274
- performance,
 - optimizing' 491–492
- persisting entity 276–279
- persisting inheritance
 - hierarchy 280–282
- querying inheritance
 - hierarchy 268
- querying TPT inheritance
 - hierarchy 269
- results
 - column names different
 - from entity
 - properties 264–265
 - columns that don't line up
 - with entity 261–264
 - entity with complex
 - type 265
 - fetching 272
 - inheritance
 - hierarchy 268–271
 - mapping to class 260–261
 - scalar values 266–267
 - that don't match an
 - entity 261–265

stored procedure, results
 (continued)
 that match an entity 258–261
 returning code with 258–274
 scenarios 258
 updating data with 275–283
 upgrading entity 282
 wrapper method 259
 StoreGeneratedPattern 50
 StoreItemCollection class 299, 329
 Strategy design pattern 370
 strong coupling problem 8–9
 StructuralType class 310
 stub 190
 in DTO 411
 stub method
 building 330–333
 consuming 289
 generating code via template 289
 prerequisite 289
 scalar-valued function 289
 user-defined function 291
 surrogate key 21
 syntactic sugar 510

T

T4 323
 table
 designing 37–42
 joining. *See* link table
 table per concrete type 37
 table per hierarchy 38
 discriminator column 182
 vs. table per type 39
 table per type 37
 drawbacks 39
 vs. table per hierarchy 39
 TablePerTypeStrategy.xaml
 workflow 337
 table-valued function 287–288
 Tangible Engineering, Visual
 Studio extension 324
 template 322
 adding 53
 assembly directive 325
 block 325
 class block tag 325
 code generation 323
 code section 323, 326–328
 conceptual to mapping 342–343
 conceptual to storage 340–342
 creating 323
 creating method 327
 customizing 328, 336
 directive tag 325
 directives 323
 evaluation block tag 325
 extension 327
 import directive 325
 include directive 326
 markup 324
 output directive 325
 output pragma 327
 pragma section 323, 325–326
 storage to database script 343–344
 tags 324–325
 template directive 325
 TEntity type 156
 Test Project template 451
 test suite, writing 451–455
 test, injecting fake ObjectSet to
 ObjectContext stub 468
 TestApi 450
 TestClass attribute 451
 TestCleanUp attribute 454, 471
 test-driven development 450
 TestInitialize attribute 454, 471
 TestInitialize method 460
 TestMethod attribute 453
 Text Template Transformation
 Toolkit. *See* T4
 three-layer architecture 358
 and complexity 363
 business layer 358
 code simplicity 362
 data access layer 358
 object model 363
 presentation layer 358
 typical 359–363
 top-down design
 in the designer 54–55
 introduction 35
 reasons to use 36
 vs. bottom-up 36–37
 ToTraceString method 76, 301
 TPC. *See* table per concrete type
 TPH. *See* table per hierarchy
 TPT. *See* table per type
 TrackableCollection class 416
 CollectionChanged event 416

tracking
 and CreateObjectSet method 74
 and Execute method 74
 transaction
 and queries 223–224
 and SQL Profiles 222
 lifetime, extending 220
 managing 220–224
 ObjectContext 222–223
 performance counters 222
 promoting 221
 TransactionScope class 220, 222
 Complete method 220
 IsolationLevel property 223
 TryGetItem method 301, 305
 TryGetItemCollection
 method 301, 304
 TryGetObjectByKey method 88
 TryGetObjectStateEntry
 method 166
 type
 anonymous. *See* anonymous type
 complex. *See* complex type
 filtering based on 105
 type inference 496–498, 505
 type safety, introduction 8

U

UML, sequence diagram 362
 unit test 448–451
 avoiding bugs in 450
 build integration 450
 data access layer 461–469
 dependencies 449, 456, 458, 461
 exception 453
 fake 456, 463
 faking ObjectSet 467
 initialization 454
 isolating dependencies 455–461
 isolation 461
 JUnit 451
 LINQ to Entities query 466–469
 loosely coupled
 dependency 458
 Microsoft Unit Testing
 Framework 453–455
 mock 458–461, 466
 MSTest 451

unit test (*continued*)
 naming convention 451–452
 NUnit 451
 persistence 470–472
 refactoring 458, 463
 remote service 456
 repository 462–466
 source control
 integration 450
 stub 457, 460, 466
 test method 449
 test report 453
 testing a method 451–453
 UpdateException class 191,
 199
 StateEntries property 199
 user-defined function 290–295
 and collection results
 294–295
 designer support 290
 output 290
 passing object as
 parameter 291–292
 returning list of generic
 objects 294
 returning list of scalar
 values 294
 returning list of typed
 objects 295
 returning nontyped
 object 292–293
 returning typed object
 293–294
 Using pattern 75

V

value object
 definition 366
 immutability 366
 modifying 366
 vs. entity 367
See also complex type

VB
 anonymous method 504
 anonymous type 497,
 508–509
 creating extension
 method 500
 Dim 497
 extension method 496
 consuming 500
 importing 500
 Function 506
 Imports 500
 lambda expression 496,
 505–507
 object initializer 496,
 507–508
 and anonymous types 508
 query syntax 497, 509–510
 type inference 496, 505
 view
 creating on database 285
 introduction 285
 mapping as table 285
 mapping via
 DefiningQuery 285
 pregenerating 484–487
 Visual Studio
 debugging 352
 designer 29
 generating code to create
 entities 123
 EDMX file 29
 generating classes 323–328
 generating database
 DDL 336
 generating mapping
 information 29
 mapping files 29

W

WCF. *See* Windows Communica-
 tion Foundation

web service
 and DTOs 404
 contract 396, 399
 developing with DTOs
 409–412
 developing with STEs
 413–422
 entities as contract 400–409
 exchanging data with
 client 398
 interface 401
 logic contract 403
 optimizing data
 exchanges 404–405
 persisting complex
 graph 403, 411–412
 proxy class 457
 serialization 399–400,
 405–409
 Where method, extension
 method 466
 Windows Communication
 Foundation 399
 contract resolution 405
 contract resolver 400
 known types 399
 serialization 405–409
 Windows Forms application,
 binding 432–441
 Windows Presentation founda-
 tion, binding in 441–446
 wrappable entity. *See* entity,
 proxied
 wrapped entity. *See* entity, prox-
 ied
 WriteAttribute method 334
 WriteProperties method 308
 WriteTypeBaseTypes
 method 307

X

XElement class 348