

Index

Numbers

1:Many associations, 290–291

A

abstract base types, 425–427

abstraction, EDM

adding, updating, and deleting model objects, 367–368

creating XML mapping files and object layer class files, 365–367

editing `EntityType` and `AssociationSet` properties, 368–369

EF (Entity Framework) and, 363–364

EntityQueries in Entity SQL, 376–381

entity-relationship model, 359–362

examining `ConceptualModels` group, 371–373

mapping physical layer to conceptual layer, 365

Object Services. See Object Services overview, 357–359

Persistence Ignorance controversy and, 390–391

scanning `StorageModels` group, 370–371
summary, 391–392

tracing `Mappings` group, 374–375

abstraction, leaky, 210

Active Directory. See LINQ to Active Directory

active entity instances, 540–543

Active Record pattern, 391

adding collection members, 237

adding entities, 509, 602–604

adding objects, LINQ to SQ, 215–219

addition operations

for updates, 549–550

validating, 510–513

ADO.NET

Data Services. See Data Services

Data Source. See data sources

Entity Framework. See EF (Entity Framework) query string options. See query string options

advanced queries

aggregate operators. See aggregate operators

`Contains()` SQO. See `Contains()` SQO

`GroupBy`, with aggregate queries. See `GroupBy`, with aggregate queries

mocking collections for testing projects. See mocking collections

overview, 155

query expression trees, 172–176

summary, 187–188

AdventureWorksMetadata VB project, 428–429

aggregate keyword, with VB, 161–162

aggregate operators

`Aggregate`, 152–153, 159

`Count` and `LongCount`, 150

`Let` variable, 158–159

method call syntax, 157–158

`Min`, `Max`, `Sum`, and `Average`, 151

overview, 149, 156

query output formatting, 159–160

aggregate values, for WHERE clause constraints, 454–455

`Aggregate...Into`

in advanced query expressions, 159

in grouping with associated child objects, 161–162

aggregating business documents. See business documents, grouping/aggregating

aggregating child entity values, 158

AJAX Client Library

adding, editing and removing entities, 602–604

becoming familiar with JSON, 598–599

creating an AJAX test client, 599–602

overview, 598

alias prefixes, in Entity SQL and Transact-SQL, 440–441

All, as a quantifier operator, 148–149

Amazon, LINQ to Amazon, 7

Ancestors, XML extension method, 271

AncestorsAndSelf, XML extension method, 271

ANDD (ASP.NET Dynamic Data)

EF as a data source for, 568
overview, 4

ANDS (ADO.NET Data Services). See Data Services

anonymous delegates, 75

anonymous methods

C# 2.0, 75–76
C# 3.0, 77
generic predicates and, 75–77
VB 8.0, 76, 77

anonymous structs, 269

anonymous types

C# 3.0, 71
DataSets, 263–266
overview, 60
VB 9.0, 71–72

Any operator, 148

ANYELEMENT operator, 452

AppendOnly, merge option, 482

arithmetic operators, 333–334, 578

array initializers, 70

AsDataView(), 257–260

AsEnumerable operator, 133

ASP.NET databinding

adding a LinqDataSource to a page, 226–228
eager loading EntityRef values for, 231–233
overview, 226
substituting EntityRef for ForeignKey values, 228–230

ASP.NET Dynamic Data (ANDD)

EF as a data source for, 568
overview, 4

ASP.NET EntityDataSource server control. See EntityDataSource server control

ASP.NET Web application page, 555–557

AsQueryable operator, 133–135

assemblies, installing LINQ to SharePoint, 344–345

associated entities

dependent, 508
EntityCollections, 489–491
EntityObject, 507–508
EntitySet, 507–508
lazy loading, 338–340

returning, 587–589

selecting instances of, 540–543

associated entities, deferred-loaded

enabling, 483–489
entity class serialization with, 478
with the Load() method, 484–485

associated entities, eager-loaded

enabling, 483–489
entity class serialization with, 478–479
with Include() operators, 485–489

associated objects

deferred loading with Attach() method, 539–540
deferred loading with Load() method, 537–538
eager loading with Include() method, 538–539
equi-joins of, 110–111
grouping, 161–162
SelectMany implementation, 108–109

Association subgroups

in ConceptualModels (CSDL content), 414
overview, 409
population of associations with stored procedures, 410–411

associations

1:Many and Many:1, 290–291
C# 3.0, 24
keys, 239–240
with LINQ join...on expressions, 27–29
mapping, 407–408
navigating entity set, 37–38
navigating to emulate joins, 24–27
populating with stored procedures, 410–411
sets, 545–546
unbound ComboBoxes to specify, 543–546
VB 9.0, 25–26

AssociationSet

elements or subgroups, 398–399
Mapping subgroup, 417–418
properties, editing, 368–369
selecting and adding new, 551–553

Astoria

creating AJAX test client for, 599–602
query string options, 576–580

Attach() method

deferred loading with, 539–540
deferred loading with the Attach() method, 539–540

attributes

attribute-centric mapping, 20, 210

EDM terminology definition, 361
 implicit v. explicit typing of attribute content,
 276–278
 in XML extension method, 271
 AutoGenerateOrderByClause **property**,
 EntityDataSource, **557**
 AutoGenerateOrderWhereClause **property**,
 EntityDataSource, **557**
autoincrementing primary key, 237
 AutoPage **property**, EntityDataSource, **557**
 AutoSort **property**, EntityDataSource, **557**
 Average **operator, 151**
axis properties, VB 9.0, 275–276

B

bad request errors, 338–340
base classes, querying
 disambiguate derived object types with base
 types, 425–427
 incorrect results from `is`, `Is`, `typeof` and
`OfType` operators, 423–424
 overview, 421–423
 type discrimination in Entity SQL queries,
 424–425
batching
 changes in, 597
 queries, 589–591
binding data
 with `AsDataView()`, 257–260
 databound web controls, 228–230
 LINQ, 12–15
 overview, 533–534
 summary, 565–566
 with `ToList()` operator, 12–15
 using `EntityDataSource`. See
`EntityDataSource` server control
 Windows form controls to entities. See Windows
 form controls, binding to entities
binding data, ASP.NET
 adding a `LinqDataSource` to a page, 226–228
 eager loading `EntityRef` values for, 231–233
 overview, 226
 substituting `EntityRef` for `ForeignKey`
 values, 228–230
**browser, creating/consuming Web service in,
 571–572**
business documents, grouping/aggregating
 aggregating `Order_Details` and
`OrdersSubtotals`, 292–294

`GroupJoin` to produce hierarchical
 documents, 289
 overview, 288
 taking advantage of `1:Many` and `Many:1`
 associations, 290–291

C

C#
 functional construction with C# 3.0, 280–284
 PLINQ, 324–325
 XML namespaces in C# 3.0. See XML
 namespaces, in C# 3.0
C# extensions
 anonymous methods and generic predicates,
 75–77
 anonymous types, 71–72
 array initializers with object initializers, 70
 collection initializers, 70–71
 expression trees and compiled queries,
 83–88
 extension methods, 72–75
 implicitly typed local variables, 67–68
`IQueryable<T>` implementations, 88–89
 lambda expressions, 77–80
 object initializers, 68–69
 overview, 66–67
 standard query operators, 80–83
**CAML (Collaborative Application Markup
 Language), 348–349**
canonical functions, 439–440
Canonical Query Tree (CQT)
 in ADO.NET Data Services, 569
 definition, 27, 363
 in EF architecture, 364
cascading deletions, in SQL Server, 238–239
Cast operator, 135–136
changes
 persisting, 506
 tracking, 20
child element groups, 308–312
child entity values, 158
classes, base, 421–423
classes, C#, 17–19, 98–100
classes, creating mock object, 177–182
classes, derived
 creating, 419–421
 disambiguate derived object types with base
 types, 425–427
 querying, 421–423

classes, entity

C# 3.0 and VB 9.0, 470

EntityName partial classes, 472–477

generated, 470

ModelNameEntities partial classes, 471–472

serialization, 477–479

classes, generated

C#, 17–19

examining, 206–210

VB 9.0, 17, 19–20

classes, LINQ to SQL, 16–20, 206–210

classes, partial entity

EntityName, 472–477

generating with SqlMetal.exe, 200–203

ModelNameEntities, 471–472

classes, VB 9.0, 17, 19–20, 100–101

clause constraints, 454–455

client layer, ADO.NET Data Service, 569–570

Client Library. See .NET 3.5 Client Library

client projects. See Northwind.svc WCF Service example

client views

EF components, 363–364

executing Entity SQL queries as, 376–381

working with, 375

clients, entity

queries against. See Entity SQL, queries against

EntityClient

queries as client views. See Entity SQL, queries as client views

working with client views, 375

writing EntityQueries in Entity SQL, 376

CLR (Common Language Runtime)

integrating XML into, 267–269

strongly typed DataSets stored as, 212

code

implementing optimistic concurrency, 517–520

sample classes for C# and VB, 98–100

writing LINQ to SharePoint, 346–348

Collaborative Application Markup Language (CAML), 348–349

Collation, SQL Server facet name, 401

collections

adding members, 237

ANYELEMENT and FLATTEN operators, 452

DeleteOnNull attribute, 239–240

deleting dependent records, 240–242

deleting members, 238

initializers, 60, 70–71

mocking. See mocking collections

navigating, 573–576

SQL Server cascading deletions, 238–239

column, discriminator, 419–421

ComboBoxes

event handlers for ComboBoxColumns, 543–544

setting composite primary-key members. See composite primary keys

specifying associations, 543–546

command-line options, with SqlMetal.exe, 201–202

CommandText property,

EntityDataSource, 557

Common Language Runtime (CLR)

integrating XML into, 267–269

strongly typed DataSets stored as, 212

comparison operators, 578

compiled queries

code examples, 87–88

LINQ to Entities, 497–499

overview, 87

complex types

overview, 528

reusing a ComplexType definition, 530–531

working with, 528–531

complex types, modeling

modifying the CSDL section, 529

modifying the MSL section, 530

overview, 529

reusing a ComplexType definition, 530–531

composite primary keys

detecting attempts to change values, 546–549

enforcing object deletion and addition, 549–550

overview, 546

selecting/adding EntityObject and AssociationSet, 551–553

compound OR operators, 169–171

compound Where expressions, 102–104

Concat operator, 120–121

conceptual database schema, 50–52

Conceptual Schema Definition Language

(CSDL), 529. See also ConceptualModels (CSDL)

conceptual schema, mapping physical to

adding, updating, and deleting objects, 367–368

creating the XML mapping files and object layer class file, 365–367

editing `EntityType` and `AssociationSet` properties, 368–369
 overview, 365
ConceptualModels (CSDL)
 `Association` subgroup, 414
 `EntityType` subgroup, 411–412
 `EntityType` subgroup, 413
 group, 371–373
 overview, 411
concurrency
 detecting and resolving conflicts, 219–222
 managing optimistic conflicts, 515–520, 597
 managing pessimistic, 515
 managing with original values, 525–527
 managing with timestamp property, 527–528
Connection page, for eSqlBlast, 434–435
ConnectionString property,
 `EntityDataSource`, 557
construction
 C# `XDocuments`, 300–302
 VB 9.0 in XML, 284–288
 XML documents, 44
Contains() SQO
 emulating SQL `IN()` function, 171–172
 emulating SQL `Where` clauses, 169–171
 overview, 169
 as a quantifier operator, 149
ContextTypeName property,
 `EntityDataSource`, 557
controls, data-enabled, 21–23
conversion operators
 `AsEnumerable`, 133
 `AsQueryable`, 133–135
 `Cast`, 135–136
 `OfType`, 136–138
Count operator, 150
CountValidISBNs, PLINQ, 324
CQT (Canonical Query Tree)
 in ADO.NET Data Services, 569
 definition, 27, 363
 in EF architecture, 364
create, retrieve, update, and delete. See CRUD (create, retrieve, update, and delete)
CREATEFEF, entity reference, 446–449
CRUD (create, retrieve, update, and delete)
 creating initial Data Model, 521–522
 deleting entity instances with stored procedures, 528
 `FunctionImports` to populate `EntitySets`, 522–524

inserting a new entity instance, 524–525
 Object Services and, 469
 overview, 521, 524
 summary, 532
 updating an entity instance and managing concurrency with original values, 525–527
 updating an entity instance and managing concurrency with timestamp property, 527–528
CSDL (Conceptual Schema Definition Language), 529. See also ConceptualModels (CSDL)
Customer, 290–291
Co , 61, 269

D

DAL (data access layer), 569–570. See also LINQ toSQL, as a DAL
data
 C# example of table data, 31–32
 editing form, 233–236
 retrieval, 405–406
 shapes, 71
 storage. *See* data store
 updating table data, 30–32
 validating additions and updates, 510–513
 VB example of table data, 32–34
data abstraction, EDM
 adding, updating, and deleting model objects, 367–368
 creating XML mapping files and object layer class files, 365–367
 editing `EntityType` and `AssociationSet` properties, 368–369
 EF (Entity Framework) and, 363–364
 `EntityQueries` in Entity SQL, 376–381
 entity-relationship model, 359–362
 examining `ConceptualModels` group, 371–373
 mapping physical layer to conceptual layer, 365
 Object Services. *See* Object Services overview, 357–359
 Persistence Ignorance controversy and, 390–391
 scanning `StorageModels` group, 370–371
 summary, 391–392
 tracing `Mappings` group, 374–375

data access

- EDM (Entity Data Model). See EDM (Entity Data Model)
- EF (Entity Framework). See EF (Entity Framework)
- LINQ (Language Integrated Query). See LINQ (Language Integrated Query)
- overview, 3–5
- summary, 57–58

data access layer (DAL), 569–570. See also LINQ toSQL, as a DAL

Data Model, creating initial, 521–522

Data Services

- consuming with AJAX Client Library. See AJAX Client Library
- consuming with .NET 3.5 Client Library. See .NET 3.5 Client Library

Data Services data source

- creating Web service, 571–572
- invoking service operations, 580–581
- navigating collections and their members, 573–576
- overview, 568–570
- query string options, 576–580
- REST architecture and, 570

Data Services Framework

- data flow diagram, 327
- overview, 4
- query options, 333
- three-layered structure, 569–570

data sources. See also EntityDataSource server control

- binding controls to entity data sources, 21–23
- creating and binding to, 54–56
- Data Services. See Data Services data source
- EF (Entity Framework) as, 567–568, 604–605

data store

- persisting changes, 506, 553–554
- SQL Server Compact as, 461–463
- substituting SSCE for SQL Server, 462–463

data table, 30–34

database schema

- *.dbml and *.xml files and, 202–206
- OR Designer generated, 34
- physical and conceptual, 49–52
- XML documents, 42–47

database server traffic, reducing, 231–233

databinding

- with `AsDataView()`, 257–260
- LINQ, 12–15

- overview, 533–534
- summary, 565–566
- with `ToList()` operator, 12–15
- using `EntityDataSource`. See `EntityDataSource` server control
- Windows form controls to entities. See Windows form controls, binding to entities

databinding, ASP.NET

- adding a `LinqDataSource` to a page, 226–228
- eager loading `EntityRef` values for, 231–233
- overview, 226
- substituting `EntityRef` for `ForeignKey` values, 228–230

databound web controls, 228–230

DataContext

- C# 3.0, 23–25
- `DataSets` compared with, 244–245
- emulating joins, 24–27
- emulating SQL functions, modifiers and operators, 30
- explicit joins, 27–29
- generating a SQL server database, 34
- instantiating, 210–211
- lazy or eager loading, 27
- `LinqDataSource` control, 34
- overview, 23–24
- updating table data, 30–34
- VB 9.0, 24, 25–26

data-enabled controls, binding to entity data sources, 21–23

DataRows, 262–263

DataServiceQuery, 585–586

DataServiceRequests, 589–591

DataSets. See also LINQ to DataSets

- C# 3.0, 35–37, 249–251
- `DataContexts` compared with, 244–245
- joining strongly typed, 35–36
- joining untyped, 36–37
- processing anonymous types, 263–266
- querying typed, 254–257
- querying untyped, 248–252
- T-SQL, 35
- VB 9.0, 36–37, 251–252

DataSpace members, 427

DataTables, copying LINQ query results to, 260–266

- copying typed `DataRows`, 262–263
- overview, 260–261

- processing anonymous types from projections, 263–266
- DataTables, **querying**. See **LINQ to DataSets**
- date query functions, 335**
- DateTime functions, 579**
- datetime Scalar Function, **404–405**
- DateTimeKind, **401**
- *.dbml **files, 203–204**
- Default, **facet name, 401**
- DefaultContainer **property**, EntityDataSource, **557**
- DefaultIfEmpty **operator, 146**
- deferred loading**
 - of associated entities, 478, 483–489
 - Attach() method, 539–540
 - Load() method, 484–485, 537–538
 - overview, 483
- delegates**
 - anonymous, 75
 - generic, 76–77
- DELETE **stored procedure**
 - assigned to entities, 528
 - mapping to EntityTypes, 406–407
 - substituting for dynamic SQL, 224–225
- DeleteOnNull, **association key, 239–240**
- deleting collection members**
 - DeleteOnNull attribute, 239–240
 - dependent records, 240–242
 - overview, 238
 - SQL Server cascading deletions, 238–239
- deletion operations**
 - in SQL Server, 238–239
 - transacting, 508
 - for updates, 549–550
- dependent associated entities, 508**
- dependent EntitySet, 238–242**
- dependent records, 240–242**
- dependent rows, 240–242**
- DEREF, **entity reference, 446–449**
- derived classes**
 - creating, 419–421
 - disambiguate derived object types with base types, 425–427
 - querying, 421–423
- derived object types, 425–427**
- DescendantNodesAndSelf, **in XML extension method, 271**
- DescendantNodes(T), **in XML extension method, 271**
- Descendants, **in XML extension method, 271**
- DescendantsAndSelf, **in XML extension method, 271**
- Designer, EDM**
 - EF roles, 363–364
 - mapping physical to the conceptual schema, 50–52
 - modifying storage to conceptual mapping with, 53–54
 - update stored procedure association mapping in, 407–408
- Designer, O/R**
 - editing *.dbml files in, 203–204
 - generating a SQL server database with, 34
 - mapping tables to entity sets, 15–16, 197–200
- DetailsView, **in EntityDataSource, 564–565**
- diagrams, entity-relationship, 361–362**
- Dim Query, 6**
- disambiguation, of derived object types, 425–427**
- discrimination, type, 424–425**
- discriminator column, TPH inheritance, 419–421**
- DISTINCT (eSQL), **491**
- Distinct (QBM), **491**
- Distinct **operator, SQO, 129**
- documents**
 - hierarchical, 289
 - joining, 314–315
 - joining to insert elements, 312–313
 - multiple namespaces, 306–308
- documents, grouping/aggregating**
 - aggregating Order_Details and OrdersSubtotals, 292–294
 - GroupJoin to produce hierarchical documents, 289
 - overview, 288
 - taking advantage of 1:Many and Many:1 associations, 290–291
- documents, XML**
 - C# 3.0, 40–41, 42, 44–45
 - querying, 40–44
 - transforming or creating, 44–47, 306–308
 - VB 9.0, 41–42, 43, 45–47
- domain objects, 34**
- domain-specific languages (DSLs), 63–64**

domain-specific LINQ implementations

application of, 190–191

IQueryable<T> interface and, 88–89

LINQ to Active Directory. See LINQ to Active Directory

LINQ to DataSet. See LINQ to DataSets

LINQ to SharePoint. See LINQ to SharePoint

LINQ to SQL. See LINQ to SQL

LINQ to XML. See LINQ to XML

overview, 342

dot-notation

NAVIGATE as a complex substitute for, 445–446

returning many:one navigation properties, 442–443

DropDownList, 563–564

DSLs (domain-specific languages), 63–64

Dynamic Data, ASP.NET, 4, 568

dynamic SQL, 222–225

E

eager evaluation, in SQOs, 82

eager loading

of associated entities, 478–479

of child objects, 27

EntityRef values to reduce server traffic, 231–233

Include() method, 485–489, 538–539
overview, 483

editing *.dbml files, 203–204

editing *.xml file, 205–206

editing entities, 236–237

EDM (Entity Data Model)

creating default model, 52–53

creating/binding to a data source, 54–56

mapping physical to the conceptual schema, 50–52

materializing object context, 56–57

modifying mapping fields to EntityType
properties, 53–54

overview, 48–50

raising the level of data abstraction.

See data abstraction, EDM

terminology and entity-relationship, 360–361

EDM Designer

EF roles, 363–364

mapping physical to the conceptual schema, 50–52

modifying storage to conceptual mapping
with, 53–54

update stored procedure association mapping
in, 407–408

EDM Wizard

creating default EDM with, 52–53

creating the XML mapping files, 365–367

EDMX file

ConceptualModels (CSDL content). See
ConceptualModels (CSDL)

Mapping (MSL content). See Mapping (MSL
content)

overview, 395–397

StorageModels (SSDL content). See
StorageModels (SSDL content)

EF (Entity Framework)

architecture and components, comprehending, 363–364

consuming Data Services with AJAX Client
Library. See AJAX Client Library

consuming Data Services with .NET Client
Library. See .NET 3.5 Client Library

creating Data Services data source. See Data
Services data source

as a data source, 567–568, 604–605

data store, 461–463

Entity Framework futures, 467–468

future of, 354–355

history of, 353–354

introduction to, 352–353

LINQ to SQL compared with, 466–467

mapping physical to the conceptual schema, 50–52

materializing an object context, 56–57

ModelName.edmx file. See ModelName.edmx
files, analyzing

No Confidence vote, 355

overview, 4, 48–50, 466

element operators

DefaultIfEmpty, 146

ElementAt and ElementAtOrDefault, 146

First and FirstOrDefault, 144–145

Last and LastOrDefault, 145

overview, 144

Single and SingleOrDefault, 145–146

elements

implicit v. explicit typing of element content, 276–278

joining documents to insert, 312–313

in XML extension method, 271

- Empty **operator, generation operator, 147**
- EnableDelete **property**,
 - EntityDataSource, **557**
- EnableInsert **property**,
 - EntityDataSource, **557**
- EnableUpdate **property**,
 - EntityDataSource, **558**
- EnableViewState **property**,
 - EntityDataSource, **558**
- entities. See also LINQ to Entities**
 - adding, 592–594
 - adding to collections, 237
 - alias prefixes, 440–441
 - binding controls to entity data sources, 21–23
 - binding Windows forms controls to. See Windows form controls, binding to entities
 - C# 3.0, 37–38
 - comparing performance, 514–515
 - deleting, 505–508, 596–597
 - deleting from collections, 238–242
 - deleting with stored procedures, 528
 - editing, 236–237
 - EDM terminology definition, 360
 - entity-relationship model, 360–362
 - JSON for working with, 602–604
 - left outer joins emulated with entity associations, 168–169
 - logging state of, 506
 - in managing entity references, 446–449
 - managing optimistic concurrency conflicts, 515–520
 - mapping tables to, 15–16, 197–200
 - ObjectStateManager and its children, 503–505
 - optimizing theObjectContext lifetime, 513–514
 - persisting changes to the data store, 506
 - refreshing stale, 509
 - relationships and, 196
 - selecting active top-level and associated instances, 540–543
 - stored procedure used to insert new instances, 524–525
 - term usage, 244
 - type constructors, 449–450
 - updating, 505–506, 594–595
 - updating instances of, 525–528
 - validating data additions and updates, 510–513
 - VB 9.0, 38
 - working with complex types, 528–531
- entities, associated**
 - deferred-loaded, 478, 483–489
 - dependent, 508
 - eager-loaded, 478–479, 483–489
 - EntityCollections, 489–491
 - EntityObject, 507–508
 - EntitySet, 507–508
 - lazy loading, 338–340
 - returning, 587–589
 - selecting instances of, 540–543
- entity classes**
 - C# 3.0 and VB 9.0, 470
 - EntityName partial classes, 472–477
 - generated, 470
 - ModelNameEntities partial classes, 471–472
 - serialization, 477–479
- entity clients**
 - EF roles, 363–364
 - queries against. See Entity SQL, queries against EntityClient
 - queries as client views. See Entity SQL, queries as client views
 - working with client views, 375
 - writing EntityQueries in Entity SQL, 376
- Entity Data Model. See EDM (Entity Data Model)**
- Entity Data Model Wizard**
 - creating default EDM, 52–53
 - creating the initial EDM, 521–522
- Entity Data Platform, Microsoft, 393**
- Entity Framework. See EF (Entity Framework)**
- entity sets**
 - adding FunctionImports to populate, 521–522
 - adding new entities, 592–594
 - associated entities and, 507–508
 - batching changes, 597
 - data retrieval stored procedures assigned to, 405–406
 - deleting entities, 596–597
 - deleting members from collections, 238–242
 - EDM terminology definition, 360
 - elements or subgroups, 398
 - EntitySetMapping, 415–417
 - EntitySetName, 558
 - managing optimistic concurrency conflicts, 597
 - mapping tables to, 197–200
 - updating, 594–595
 - updating associated, 507–508

Entity SQL

- EF roles, 363–364
- EntityQueries, 376
- ObjectQueries, 384–387, 480–483
- overview, 433–434
- query sample, 381
- query utility. See eSqlBlast
- SQL Server Compact used as an EF data store, 461–463
- statements for composing ObjectQueries, 387
- summary, 463–464
- Transact-SQL compared with. See Entity SQL/Transact-SQL comparison
- type discrimination in queries, 424–425
- using the eSqlBlast. See eSqlBlast

Entity SQL, queries against EntityClient

- executing parameterized queries, 460–461
- measuring the performance penalty of, 459
- overview, 455–456
- parsing the IExtendedDataRecord, 456–458

Entity SQL, queries as client views

- C# 3.0, 377–378
- Entity SQL from ecmdNwind.CommandText sample, 380
- overview, 376
- T-SQL, 381
- T-SQL from ToTraceString sample, 380–381
- VB 9.0, 378–380

Entity SQL/Transact-SQL comparison

- ANYELEMENT and FLATTEN operators, 452
- dot-notation syntax and, 442–443
- entity alias prefixes, 440–441
- explicit projections, 441
- IS OF, OFTYPE, and TREAT, 453–454
- JOINS, 445
- NAVIGATE, 445–446
- one:many navigation properties, 444–445
- ORDER BY clause for paging, 452–453
- overview, 439–440
- REF, Deref, CREATeref, ROW, and KEY references, 446–449
- sorting collections, 451
- type constructors, 449–450
- UNION, INTERSECT, OVERLAPS, and EXCEPT operators, 450
- VALUE modifier, 441–442
- WHERE clause constraints throwing exceptions, 454–455

EntityCollections, **489–491**

EntityContainer **subgroup, 398–399, 411–412**

EntityDataReader, **456–458**

EntityDataSource **server control**

- adding a control to ASP.NET Web application page, 555–557
- adding a linked DetailsView, 564–565
- binding and formatting a GridView control, 562–563
- control added to ASP.NET Web application page, 555–557
- EntityDataSource events, 561–562
- EntityDataSource properties, 557–558
- overview, 4, 554–555, 557
- using the GroupBy property and a DropDownList, 563–564
- Where and GroupBy clauses, 559–560

EntityKey, **504–505**

EntityName **partial classes**

- C# 3.0, 473–475
- overview, 472
- VB 9.0, 475–477

EntityObject

- associated entities, 507–508
- selecting and adding new, 551–553
- updating associated, 507–508

EntityQueries, **376**

EntityRef

- eager loading values to reduce database server traffic, 231–233
- substituting for ForeignKey values, 228–230

entity-relationship diagrams, 361–362

entity-relationship model, 360–362

EntityState

- logging, 506
- ObjectStateManager and, 504

EntityType

- editing properties, 368–369
- filters, 558
- INSERT, UPDATE, AND DELETE functions mapped to, 406–407
- Key subgroup, 400
- modifying mapping fields to EntityType properties, 53–54
- overview, 503
- Property subelement, 400–401
- subgroups, 400–402, 413

equality operators, 142–143

equi-joins, 110–111

eSQL ObjectQueries

- C# 3.0, 480
 - merge options and execution times, 482–483
 - overview, 480
 - VB 9.0, 481
 - eSqlBlast**
 - Connection page, 434–435
 - Model page, 435–436
 - overview, 434
 - Query page, 436–437
 - Result page, 437–439
 - event handlers**
 - for ComboBoxColumns, 543–544
 - deleting dependent records with, 240–242
 - updating association sets with, 545–546
 - events, EntityDataSource, 561–562**
 - EXCEPT (eSQL), **491**
 - Except (QBM)
 - composing QBMs, 491
 - requiring sub-queries, 450
 - as a SQO set operator, 132
 - exceptions, subqueries throwing, 454–455**
 - execution times**
 - eSQL ObjectQueries, 483
 - LINQ to Entities queries, 499
 - \$expand, **ADO.NET query string, 577**
 - expand, **ANDS query, 333**
 - explicit loading, 483. See also deferred loading**
 - explicit projections, in Entity SQL and Transact-SQL, 441**
 - expression syntax, 162**
 - expression trees, 27**
 - C# 3.0, 83–84
 - overview, 83
 - query expression trees, 172–176
 - VB 9.0, 86–87
 - VS2008 SP1, 84–86
 - expressions, LINQ to Expressions, 7**
 - extension methods**
 - C# 3.0, 72–74
 - LINQ to XML, 270–271
 - overview, 60, 72
 - VB 9.0, 74–75
 - extensions, C#**
 - anonymous methods and generic predicates, 75–77
 - anonymous types, 71–72
 - array initializers with object initializers, 70
 - collection initializers, 70–71
 - expression trees and compiled queries, 83–88
 - extension methods, 72–75
 - implicitly typed local variables, 67–68
 - IQueryable<T> implementations, 88–89
 - lambda expressions, 77–80
 - object initializers, 68–69
 - overview, 66–67
 - standard query operators, 80–83
 - extensions, VB**
 - anonymous methods and generic predicates, 75–77
 - anonymous types, 71–72
 - array initializers with object initializers, 70
 - collection initializers, 70–71
 - expression trees and compiled queries, 83–88
 - extension methods, 72–75
 - implicitly typed local variables, 67–68
 - IQueryable<T> implementations, 88–89
 - lambda expressions, 77–80
 - object initializers, 68–69
 - overview, 66–67
 - standard query operators, 80–83
- ## F
- facets, 400–401**
 - \$filter, **ADO.NET query, 577**
 - filter, **ANDS query, 333**
 - First() operator, 144–145**
 - FirstOrDefault() operator, 144–145**
 - FixedLength, SQL Server facet name, 401**
 - FLATTEN operator, 452**
 - Flickr, LINQ to Flickr, 7**
 - foreign-key values**
 - EntityRef substituting for, 228–230
 - SQL Server Foreign Key Relationships, 238–239
 - form controls, Windows. See Windows form controls, binding to entities**
 - Function subelements and subgroups**
 - Function definitions, 404–405
 - Function imports, 405–406
 - INSERT, UPDATE, AND DELETE function mapping to EntityTypes, 406–407
 - overview, 402–404
 - update stored procedure association mapping, 407–408
 - functional construction**
 - C# XDocuments, 300–302
 - XML documents, 44, 280–284
 - functional languages, 60–61**
 - FunctionImportMapping element, 418**

FunctionImports, **521–522**
functions, query, 335
functions, SQL, 30
futures, EF (Entity Framework), 467–468

G

generated classes

C#, 17–19
examining, 206–210
VB 9.0, 17, 19–20

generated entity classes

C# 3.0 and VB 9.0, 470
EntityName partial classes, 472–477
ModelNameEntities partial classes,
471–472
overview, 470
serialization, 477–479

generation operators

Empty, 147
overview, 147
Range, 147
Repeat, 147

generic predicates, 75–77

graphs, object

deferred loading of associated objects with
Load() method, 537–538
deferred loading with the Attach() method,
539–540
eager loading of multiple associated objects
with Include() method, 538–539
shaping, 535–537

GridView control, 562–563

Group By

clauses, 557–560
displaying customers by country, 563–564
EntityTypeSource property name, 558
GROUP BY (eSQL), 491
GroupBy (QBM), 491
grouping operators, 125–128
Join and, 164–166

Group By, with aggregate queries

emulating left outer joins, 168–169
grouping associated child objects, 161–162
grouping joined child objects, 163–164
Join and Group By operations combined,
164–166
joins and nested queries compared with,
166–168

overview, 160–161

group joins

C# and VB examples, 118–119
combining Join and GroupBy, 164–166
compared to nested LINQ queries,
166–168
creating object graphs, 182–187
overview, 117
producing hierarchical documents, 289

grouping

associated child objects, 161–162
joined child objects, 163–164

grouping business documents. See business documents, grouping/aggregating

grouping operators

ADO.NET Data Services, 333–334, 578
GroupBy overview, 125–126
GroupBy with method call syntax, 126–127
GroupBy with query expression syntax,
127–128

groups.child element, 308–312

H

Haskell (<http://www.haskell.org>), 60–61

heterogeneous joins

joining documents and LINQ to SQL or LINQ to
Object entities, 314–315
joining documents to insert elements,
312–313
lookup operations to add child element groups,
308–312
overview, 308

hierarchical data, 233–236

hierarchical documents, 289

hierarchical group join expressions

combining Join and GroupBy with, 164–166
compared to nested LINQ queries, 166–168

HTTP 400, 338–340

I

IExtendedDataRecord, 456–458

implicitly typed lambda expression, 79–80

implicitly typed local variables

C# and VB examples, 67–68
overview, 60

imports, Function, 405–406

IN() function, 171–172

Include()
 eager loading with, 485–489, 538–539
 EntityDataSource, 558
 LINQ to Entities queries, 495–497

incubator projects, 47

index arguments, 104–105

index value expressions, 107–108

IndexOf(), 104–105

InDocumentOrder(T), in XML extension
 method, 271

inheritance, TPH (table-per-hierarchy)

base and derived class queries, 421–423
 disambiguate derived object types with base
 types, 425–427
 incorrect results from `is`, `Is`, `typeof` and
`OfType` operators, 423–424
 overview, 419
 specifying discriminator column/creating a
 derived class, 419–421
 type discrimination in Entity SQL queries,
 424–425

initialization code

C# example, 98–100
 VB example, 100–101

initializers

array, 70
 collection, 60, 70–71
 mock object, 177–182

initializers, object

with array initializers, 70
 C#, 68, 69, 70
 overview, 60
 VB, 69

INSERT stored procedures

function mapping to EntityTypes, 406–407
 inserting a new entity instance, 524–525
 substituting for dynamic SQL, 224–225

instantiating DataContext, 210–211

IntelliSense

enabling for namespaces, 302–304
 enabling for VB queries, 273–275

Intersect

SQO set operator, 131–132
 sub-queries required by, 450

INTERSECT (eSQL), 491

Intersect (QBM), 491

Into expression, with VB, 161–162

inverted enumeration, 321

IOrderedQueryable<T>, 88–89

IQueryable<T>, 88–89

IS OF operation, for polymorphic queries,
 453–454

Is operator, incorrect results from, 423–424

is operator, incorrect results from, 423–424

IsAggregate attribute, Entity SQL, 403

IsBuiltIn attribute, Entity SQL, 403

IsComposable attribute, Entity SQL, 403

ISingleResult<TEntity>, 223–224

IsNiladic attribute, Entity SQL, 403

J

JavaScript Object Notation (JSON)

overview, 598–599

WCF DataContractJsonSerializer
 (DCJS), 477

Join operators

combined with GroupBy, 164–166

GroupJoin, 117–119

Join, 116–117

overview, 115–116

joined child objects, 163–164

join...on expressions

code examples, 28–29

to create explicit joins, 27–29

joins, as last resort, 445

joins, group

C# and VB examples, 118–119

combining Join and GroupBy, 164–166

compared to nested LINQ queries, 166–168

creating object graphs, 182–187

overview, 117

producing hierarchical documents, 289

joins, heterogeneous

joining documents and LINQ to SQL or LINQ to
 Object entities, 314–315

joining documents to insert elements, 312–313

lookup operations to add child element groups,
 308–312

overview, 308

JSON (JavaScript Object Notation)

overview, 598–599

WCF DataContractJsonSerializer
 (DCJS), 477

K

KEY, entity reference, 446–449

Key subgroup, EntityType, 400

keys

- adding members to a collection with, 237
- association, 239–240
- composite primary. See composite primary keys
- entity primary, 360
- natural, 546
- surrogate, 546

keywords. See also specific keywords

- SQOs as, in C# and VB, 93–94
- unclassified, 93
- unsupported in LINQ, 494

L

lambda expressions

- C# 3.0, 78–79
- overview, 60, 77–78
- VB (8.0 and 9.0), 79–80

lambda function, 158

Language Integrated Query. See LINQ (Language Integrated Query)

Last () operator, 145

LastOrDefault () operator, 145

layered mapping, in ADO.NET Entity Framework, 393–394

lazy evaluation, in SQOs, 82

lazy loading

- associated entities, 338–340
- child objects, 27
- LINQ toSQL and, 214
- SQOs and, 82

LDAP, LINQ to LDAP, 7

leaky abstraction, 210

left outer joins, 168–169

Let

- index value expressions with, 107–108
- as temporary local aggregate variable, 158–159

libraries

- AJAX Client Library. See AJAX Client Library
- .NET 3.5 Client Library. See .NET 3.5 Client Library

lifetime,ObjectContext, 513–514

LIMIT subclauses, of ORDER BY clause, 452–453

LIMOG (LINQ in-memory object generator)

- creating mock object classes and initializers, 177–178
- creating object graphs, 182, 186, 188
- LINQ to Objects codes and, 96–97

LINQ (Language Integrated Query)

- anonymous methods and generic predicates, 75–77
- anonymous types, 71–72
- array initializers with object initializers, 70
- C# and VB Extensions, 66–67
- collection initializers, 70–71
- compiled queries, 87–88
- copying query results to DataTables, 260–266
- defined, 3
- emerging implementations, 317–318
- expression trees, 83–87
- extension methods, 72–75
- form of basic query, 6–7
- history of, 61–62
- implicitly typed local variables, 67–68
- IQueryable<T> and IOrderedQueryable<T>, 88–89
- lambda expressions, 77–80
- namespaces, 64–65
- object initializers, 68–69
- overview, 5–8, 63–64
- standard query operators, 80–83
- summary, 89
- third-party implementations, 7
- unsupported keywords, 494

LINQ in-memory object generator. See LIMOG (LINQ in-memory object generator)

LINQ Project Sample Query Explorer, 95–96

LINQ to Active Directory

- implementation of, 342–344
- overview, 190, 318

LINQ to Amazon, 7

LINQ to DataSets

- comparing DataSets and DataContexts, 244–245
- copying query results to DataTables, 260–261
- copying typed DataRows, 262–263
- customizing lookup lists, 253–254
- data binding with AsDataView (), 257–260
- defined, 4, 190
- as domain-specific implementation, 193
- features, 246–247
- joining DataSets, 35–37
- navigating entity set associations, 37–38
- overview, 34–35
- processing anonymous types from projections, 263–266
- querying DataTables with, 243–244
- querying typed DataSets, 254–257

- querying untyped `DataSets`, 248–252
- read-only queries, 247–248
- T-SQL, 35
- LINQ to Entities**
 - comparing performance, 499
 - compiling queries, 497–498
 - conventional queries, 494–495
 - creating object graphs, 182–187
 - defined, 4
 - implementing LINQ to Entities provider, 388–390
 - `Include()` operator, 495–497
 - mock collections for testing, 176–177
 - mock object classes and initializers, 177–182
 - Out-of-Band SQL updates compared to, 514–515
 - overview, 38–39
 - provider role in EF, 363–364
 - unsupported LINQ keywords, SQOs and overloads, 494
 - writing queries, 493–494
- LINQ to Expressions, 7**
- LINQ to Flickr, 7**
- LINQ to LDAP, 7**
- LINQ to NHibernate, 7**
- LINQ to Objects**
 - aggregate operators, 149–153
 - `AsEnumerable` operator, 133
 - `AsQueryable` operator, 133–135
 - C# class definition and initialization code example, 98–100
 - `Cast` operator, 135–136
 - concatenation operator: `Concat`, 120–121
 - conversion operators, 132
 - data binding with `ToList()` operator, 12–15
 - `DefaultIfEmpty()` operator, 146
 - element operators, 144
 - `ElementAt()` and `ElementAtOrDefault()` operators, 146
 - equality operators, 142–143
 - `First()` and `FirstOrDefault()` operators, 144–145
 - generation operators, 147
 - grouping operators, 125–128
 - `GroupJoin` operator, 117–119
 - index arguments and `IndexOf()`, 104–105
 - `Join` operators, 115–117
 - joining documents and, 314–315
 - `Last()` and `LastOrDefault()` operators, 145
 - LINQ Project Sample Query Explorer, 95–96
 - `OfType` operator, 136–138
 - ordering operators, 121–124
 - overview, 3, 8–9, 91–92
 - partitioning operators, 111
 - projection operators, 105
 - quantifier operators, 148–149
 - sample classes for code examples, 96–98
 - sample queries, 9–12
 - `Select` operator, 106–108
 - `SelectMany` operator, 108–111
 - set operators, 128–132
 - `Single()` and `SingleOrDefault()` operators, 145–146
 - `Skip` operator, 112–113
 - `SkipWhile` operator, 114–115
 - SQOs as keywords, 93–94
 - SQOs by group, 92–93
 - summary, 153–154
 - `Take` operator, 112
 - `TakeWhile` operator, 113–115
 - `To...` operators, 138–142
 - VB class definition and initialization code example, 100–101
 - `Where` restriction operator, 101–104
- LINQ to REST. See also Northwind.svc WCF Service example**
 - adopting URIs as a query language, 328–329
 - C# 3.0 and VB 9.0 queries, 586–587
 - executing queries, 337–338
 - as a LINQ implementation, 325–328
 - overview, 190, 317
- LINQ to SharePoint**
 - creating a sample project, 345–346
 - installing assemblies, 344–345
 - installing SharePoint 3.0 SDK, 345
 - overview, 190, 318, 344
 - translating query expressions, 348–349
 - writing query code, 346–348
- LINQ to SQL**
 - application of (overview), 190, 195–196
 - binding controls to entity data sources, 21–23
 - creating mock object classes and initializers, 177–182
 - creating object graphs, 182–187
 - domain-specific LINQ implementations, 191–192
 - vs. EF (Entity Framework), 466–467
 - emulating joins by navigating entity associations, 24–27

LINQ to SQL (continued)

- emulating SQL functions, modifiers and operators, 30
- examining the generated classes, 16–20
- explicit joins, 27–29
- generating SQL server database with OR designer, 34
- joining documents and, 314–315
- LinqDataSource control, 34
- loading child objects lazily or eagerly, 27
- mapping tables to entities, 15–16
- mock collections for testing, 176–177
- ObjectContext and, 469
- object/relational mapping. *See* object/relational mapping
- overview, 4, 15
- query pipeline, 212–215
- uploading table data, 30–34

LINQ to SQL, as a DAL

- adding, updating and removing objects, 215–219
- detecting and resolving concurrency conflicts, 219–222
- LINQ to SQL query pipeline, 212–215
- moving the LINQ to SQL layer to a middle tier, 225
- overview, 212
- substituting stored procedures, 222–225

LINQ to stored XML

- emerging LINQ implementations, 341–342
- overview, 317

LINQ to Streams, 7

LINQ to XML

- 1:many and many:1 associations, 290–291
- aggregating order details and subtotals by customer, 292–294
- class hierarchy, 270
- composing XML Infosets, 278–280
- document manipulation, 267
- domain-specific LINQ implementations, 193
- functional construction with C# 3.0, 280–284
- grouping elements and aggregating numeric values, 288
- GroupJoin for hierarchical documents, 289
- heterogeneous joins and lookup operations, 308
- implicit vs. explicit typing, 276–278
- inferring a schema and enabling IntelliSense for VB queries, 273–275
- integrating XML into the CLR, 267–269

- joining documents, 312–315
- LINQ to XML objects, 39–40
- literal construction with VB 9.0, 284–288
- lookup operations for adding child element groups, 308–312
- overview, 4, 39, 190
- queries, namespaces in C#, 297–298
- querying basic XML Infosets, 271–273
- querying XML documents, 40–44
- querying XML with C#, 269
- summary, 316
- System.Xml.Linq namespace, 269–271
- transforming or creating XML documents, 44–47
- VB axis properties and, 275–276
- Xen for minimizing XML object mismatch, 268
- XML namespaces, 295–297
- XML namespaces in C# 3.0, 297–302
- XML namespaces in VB 9.0, 302–308

LINQ to XSD, 44–47

- emerging implementation of, 340–341
- emerging LINQ implementations, 340–341
- history of, 341
- overview, 4, 47–48, 317

LINQ4SP, 7

LinqDataSource control, 34

- adding to a page, 226–228
- eager loading EntityRef values for, 231–233
- overview, 4, 226
- substituting EntityRef for ForeignKey values, 228–230

LinqDataViews, 257–260

LINQRayTracer, 324–325

literal construction, with VB 9.0 in XML, 284–288

literal queries, 304–305

literal XML

- construction with VB 9.0, 284–288
- multiple namespaces transformed with code, 306–308

Load() method, 484–485, 537–538

loading, deferred

- of associated entities, 478, 483–489
- Attach() method, 539–540
- Load() method, 484–485, 537–538
- overview, 483

loading, eager

- of associated entities, 478–479
- of child objects, 27

EntityRef values to reduce server traffic, 231–233
 Include() method, 485–489, 538–539
 overview, 483

loading, lazy

associated entities, 338–340
 child objects, 27
 LINQ toSQL and, 214
 SQOs and, 82

local names, XML. See XML namespaces

local variable type inference

C# and, 67
 implicitly typed, 67–68
 overview, 60
 VB and, 68

logical operators, ADO.NET Data Services, 333–334, 578

LongCount, as an aggregate operator, 150

lookup operations

C# 3.0 functional construction, 309–310
 customizing lookup lists, 253–254
 overview, 308–309
 VB 9.0 functional construction, 311–312
 VB 9.0 literal XML, 310–311

M

Many:1 associations

in business documents, 290–291
 navigation properties, 442–443

mapping. See also object/relational mapping

associations, 407–408
 attribute-centric, 20, 210
 EDM Designer and, 53–54
 EDM terminology definition, 360
 files with SqlMetal.exe, 200–203
 function mapping to EntityTypes, 406–407
 from physical to conceptual schema, 50–52
 tables to entity sets, 197–200

Mapping (MSL content)

AssociationSetMapping subgroup, 417–418
 EntitySetMapping subgroup, 415–417
 FunctionImportMapping element, 418
 overview, 414

mapping files

*.dbml and *.xml files, 203–206
 generating with SqlMetal.exe, 200–203
 role as EF (Entity Framework) component, 363–364

Mapping Schema Language. See MSL (Mapping Schema Language)

Mapping subgroup, AssociationSet, 417–418

mappings group, 373–375

mashup, Web, 326

math functions, ADO.NET Data Services, 335, 579

Max operator, 151

MaxLength facet, SQL Server, 401

members, adding to collections, 237

merge options

eSQL ObjectQueries, 482
 updating associated EntitySets or EntityObjects, 507–508

\$metadata, ADO.NET query string option, 577

MetadataWorkspace

C# 3.0, 428–429

ObjectContext and, 383–384

traversing, 427–428

VB 9.0, 429–431

method calls

GroupBy with, 126–127

index arguments and, 104–105

numerical operators and, 157–158

translating query expressions to, 348–349

Microsoft Entity Data Platform, 391–393

Microsoft Synchronization Framework, 5

middle tier, LINQ to SQL layer, 225

Min operator, 151

mocking collections

creating mock object classes and initializers, 177–182

creating object graphs, 182–187

testing LINQ to SQL and LINQ to Entities, 176–177

model, entity-relationship, 360–361

Model page, for eSqlBlast, 435–436

modeling a complex type

modifying the CSDL section, 529

modifying the MSL section, 530

overview, 529

reusing a ComplexType definition, 530–531

ModelName.edmx files, analyzing

ConceptualModels group, 371–373

mappings group, 373–375

StorageModels group, 370–371

ModelNameEntities partial classes

C# 3.0, 471

overview, 471

VB 9.0, 472

models, conceptual

- Association subgroup, 414
- EntityContainer subgroup, 411–412
- EntityType subgroup, 413
- group, 371–373
- overview, 411

models, Data Model, 521–522

models, storage

- association subgroups, 409–411
- EntityContainer subgroup, 398–399
- EntityType subgroups, 400–401
- Function subelements and subgroups. See Function subelements and subgroups overview, 397
- Product EntityType subgroup, 402
- scanning StorageModels group, 370–371

modifications, collection, 236–237

modifier, VALUE, 441–442

modifiers, SQL, 30

MSL (Mapping Schema Language)

- AssociationSetMapping subgroup, 417–418
- EntitySetMapping subgroup, 415–417
- FunctionImportMapping element, 418
- mapping physical to the conceptual schema, 50–52
- modification in modeling a complex type, 530
- overview, 414

multiple associated objects, 538–539

multiple namespaces

- transforming with literal XML code, 306–308
- XDocuments constructed with, 300–302
- XML literal queries, 304–305

multiple Select projection expression, 107

MULTISET type constructor, 449–450

N

namespaces

- C# expanded, 298–299
- IntelliSense for, 302–304
- LINQ to XML, 297–299
- .NET Framework 3.5, 64–65
- System.Xml.Linq, 269–271
- VB 9.0, 302–308

namespaces, multiple

- transforming documents containing, 306–308
- XDocuments constructed with, 300–302
- in XML literal queries, 304–305

namespaces, XML

- enabling IntelliSense for namespaces, 302–304

- examples, 295–296
- functionally constructing XDocuments, 300–302
- multiple namespaces in XML literal queries, 304–305
- namespaces in C# LINQ to XML queries, 297–298
- overview, 295, 302
- removing expanded namespaces, 298–299
- transforming documents with multiple namespaces, 306–308

natural keys, 546

NAVIGATE, 445–446

navigation properties

- many:one, 442–443
- one:many, 444–445

nested queries

- NAVIGATE as a complex substitute for, 445–446
- one:many navigation, 444–445
- sorting collections requiring, 451

.NET 3.5 Client Library

- adding new entities, 592–594
- batching changes, 597
- deleting entities, 596–597
- executing batch queries with DataServiceRequests, 589–591
- executing DataServiceQuery, 585–586
- executing LINQ to REST queries, 586–587
- executing queries from Windows Forms clients, 583
- executing query-string URIs from clients, 583–585
- managing optimistic concurrency conflicts, 597
- overview, 581–583
- returning associated entities, 587–589
- updating entities, 594–595

.NET BinarySerializer, 477

.NET Framework 3.5 namespaces, 64–65

.NET XmlSerializer, 477

Next Generation Data Access, 58

NHibernate, LINQ to NHibernate, 7

Nodes (T), in XML extension method, 271

Northwind.svc WCF Service example

- experimenting with URI query syntax, 329–332
- logical, arithmetic and grouping operators, 333–334
- query options, 333
- string, date, math, and type query functions, 335
- in XML (Atom 1.0), 332

NoTracking, merge option, 482

NQueens, PLINQ, 324

Nullable facet, SQL Server, 401

numeric values of business documents. See

business documents, grouping/aggregating

numerical operators, 157–158

NwindServicesClient services requests

executing LINQ to REST queries, 337–338

executing URI queries, 337

overview, 336

preventing HTTP 400, 338–340

O

object addition operations, for updates, 549–550

Object data sources, binding Windows forms controls to entities

overview, 534–535

persisting changes to the data store, 553–554

selecting active top-level/associated entity instances, 540–543

setting composite primary-key members. See composite primary keys

using unbound ComboBoxes, 543–546

object deletion operations, for updates, 549–550

object graphs

creating with GroupJoin expressions, 182–187

deferred loading of associated objects with Load() method, 537–538

deferred loading with the Attach() method, 539–540

eager loading of multiple associated objects with Include() method, 538–539

instantiating, 210–211

overview, 155

shaping, 535–537

object initializers

with array initializers, 70

C#, 68, 69, 70

overview, 60

VB, 69

object layer class file, 365–367

Object Services

compiling LINQ to Entity queries, 497–499

composing ObjectQueries, 491–492

conventional LINQ to Entities queries, 494–495

deferred loading, 484–485

eager loading, 485–489

EF roles, 363–364

entity class serialization, 477–479

EntityName partial class, 472–477

eSQL ObjectQueries, 480–483

generated entity classes and, 470

Include() operator in LINQ to Entity queries, 495–497

LINQ to Entities provider, 388–390

ModelNameEntities partial class, 471–472

ObjectContext, 382–384

ObjectQueries with Entity SQL, 384–387

ObjectQueries with QBMs, 387–388

ordering and filtering associated

EntityCollections, 489–491

ordering/filtering EntityCollections, 489–491

overview, 381–382, 469

parameterizing ObjectQueries, 500

performance of LINQ to Entity queries, 499

summary, 500–501

unsupported LINQ keywords, SQOs and overloads, 494

writing LINQ to Entities queries, 493–494

object types, derived, 425–427

ObjectContext

creating ObjectQuery and, 382

Object Services and, 469

ObjectStateManager and children of, 503–505

optimizing the lifetime of, 513–514

ObjectDataSource, 226

ObjectQuery

composing with QBMs, 387–388, 491–492

creating ObjectContext and, 382

writing with Entity SQL, 384–387

object/relational mapping

examining the generated classes, 206–210

generating partial entity classes/mapping files, 200–203

instantiating DataContext, 210–211

mapping tables to entity sets, 197–200

overview, 197

working with *.dbml and *.xml mapping files, 203–206

objects. See also LINQ to Objects

adding, updating, and deleting, 215–219, 367–368

child, 161–163

code examples, 56–57

deletion/addition operations for updating, 549–550

objects. See also LINQ to Objects (continued)

- domain, 34
- graphs. See object graphs
- initializers, 60, 68–70
- LINQ to XML, 39–40
- materializing object context, 56–57
- persistence, 195–196
- relationship to resources, 395
- XML/object mismatch, 268

objects, associated

- deferred loading with `Attach()` method, 539–540
- deferred loading with `Load()` method, 537–538
- eager loading with `Include()` method, 538–539
- equi-joins of, 110–111
- grouping, 161–162
- `SelectMany` implementation, 108–109
- `ObjectStateManager`
 - methods and event, 504
 - understanding, 503–505
 - in working with `ObjectContext`, 384

OFTYPE (eSQL), 491

- `OfType` (eSQL)
 - as a conversion operator, 136–138
 - incorrect results from, 423–424
 - as type operator for polymorphic queries, 453–454

OfType (QBM), 491

one:many, navigation properties, 444–445

operators. See also specific SQOs

- code examples, 30
- emulating SQL, 30
- operator groups list, 92–93

operators, aggregate

- `Aggregate`, 152–153, 159
- `Count` and `LongCount`, 150
- `Let` variable, 158–159
- method call syntax, 157–158
- `Min`, `Max`, `Sum`, and `Average`, 151
- overview, 149, 156
- query output formatting, 159–160

operators, arithmetic, 578

operators, comparison, 578

operators, compound OR, 169–171

operators, concatenation, 120–121

operators, conversion

- `AsEnumerable`, 133
- `AsQueryable`, 133–135

- `Cast`, 135–136
- `OfType`, 136–138
- `To...operators`. See `To...operators`

operators, element

- `DefaultIfEmpty`, 146
- `ElementAt` and `ElementAtOrDefault`, 146
- `First` and `FirstOrDefault`, 144–145
- `Last` and `LastOrDefault`, 145
- overview, 144
- `Single` and `SingleOrDefault`, 145–146

operators, equality, 142–143

operators, generation

- `Empty`, 147
- `Range`, 147
- `Repeat`, 147

operators, grouping

- ADO.NET Data Services and, 333–334, 578
- method call syntax for, 126–127
- overview, 125–126
- query expression syntax for, 127–128

operators, Join

- combined with `GroupBy`, 164–166
- `GroupJoin`, 117–119
- `Join`, 116–117
- overview, 115–116

operators, logical, 333–334, 578

operators, ordering

- examples, 123–124
- `OrderByDescending`, 122
- overview, 121–122
- `ThenBy`, 122
- `ThenByDescending`, 123

operators, partitioning

- overview, 111
- `Skip`, 112
- `Skip/Take` example, 112–113
- `SkipWhile`, 114
- `SkipWhile/TakeWhile` example, 114–115
- `Take`, 112
- `TakeWhile`, 113–114

operators, project

- overview, 105
- `Select`, 106–108
- `SelectMany`, 108–111

operators, quantifier

- `All`, 148–149
- `Any`, 148
- `Contains`, 149
- overview, 148

operators, restriction, 101–105

- compound `Where` expressions, 102–104
- method calls with index arguments and use of `IndexOf`, 104–105
- simple `Where` expressions, 102
- `Where` overview, 101

operators, set

- `Distinct`, 129
- `Except`, 132
- `Intersect`, 131–132
- overview, 128
- `Union`, 129–131

operators, standard query, 80–83**operators, To...**

- overview, 138
- `ToArray`, 138
- `ToDictionary`, 139–140
- `ToList`, 139
- `ToLookup`, 141–142

operators, type, 453–454**optimistic concurrency management**

- changing `EntitySets` and, 597
- code for implementing, 517–520
- enabling for entity properties, 516–517
- overview, 515–516

O/R Designer

- editing *.dbml files in, 203–204
- generating a SQL server database with, 34
- mapping tables to entity sets, 15–16, 197–200

OR operators, compound, 169–171

- `ORDER BY` clause, 452–453
- `ORDER BY` (eSQL), 491
- `Order_Details`, aggregating, 290–291
- `OrderBy` (QBM), 491
- `OrderBy`, ADO.NET Data Services, 333
- `$orderby`, ADO.NET query string option, 577
- `OrderBy` property, `EntityDataSource`, 558
- `OrderByDescending`, 122

ordering operators

- C# 3.0 example, 123–124
- `OrderByDescending`, 122
- overview, 121–122
- `ThenBy`, 122
- `ThenByDescending`, 123
- VB 9.0 example, 124

OrdersSubtotals, aggregating, 290–291**Out-of-Band SQL, 514–515**

- `OVERLAPS` set operators, 450
- `OverwriteChanges`, 482

P**pages, eSqlBlast**

- Connection page, 434–435
- Model page, 435–436
- Query page, 436–437
- Result page, 437–439

paging, SKIP and LIMIT subclauses and, 452–453**Parallel Computing Initiative (PCI), 319****Parallel LINQ. See PLINQ (Parallel LINQ)****Parameter Source types, 560****parameterized queries**

- execution of, 460–461
- `ObjectQuery`, 500

ParameterTypeSemantics attribute,**Entity SQL, 403****partial entity classes**

- `EntityName`, 472–477
- generating with `SqlMetal.exe`, 200–203
- `ModelNameEntities`, 471–472

partitioning operators

- overview, 111
- `Skip`, 112
- `Skip/Take` example, 112–113
- `SkipWhile`, 114
- `SkipWhile/TakeWhile` example, 114–115
- `Take`, 112
- `TakeWhile`, 113–114

PCI (Parallel Computing Initiative), 319**performance**

- Entity SQL queries and, 459
- LINQ to Entities updates, 514–515
- Out-of-Band SQL updates, 514–515
- query expression trees for improving, 172–176

Persistence Ignorance (PI), 20, 210, 390–391**pessimistic concurrency management, 515****physical layer, mapping to conceptual**

- adding, updating, and deleting the model's objects, 367–368
- creating the XML mapping files and object layer class file, 365–367
- editing `EntityType` and `AssociationSet` properties, 368–369
- overview, 50–52, 365

PI (Persistence Ignorance), 20, 210, 390–391**pipelined processing, 321**

PLINQ (Parallel LINQ), 5

- overview, 191, 317–319
- processing queries, 321–322
- programming with, 319–321
- running the samples, 322–325

pluralization, 16

polymorphic queries, 453–454

Precision facet, **SQL Server, 401**

predicates, generic, 75–77

prefixes, entity alias, 440–441

Preserve Seconds facet, **SQL Server, 401**

PreserveChanges, **merge option, 482**

primary keys, adding members to a collection with, 237

primary keys, composite

- detecting attempts to change values, 546–549
- enforcing object deletion and addition, 549–550
- overview, 546
- selecting/adding `EntityObject` and `AssociationSet`, 551–553

procedures, stored. See stored procedures

Product EntityType subgroup, storage models and, 402

Product stored procedures, **404–405**

programming, `DataContext`. **See DataContext**

programming, with PLINQ, 319–321

project operators

- overview, 105
- Select, 106–108
- SelectMany, 108–111

projection expression

- multiple Select, 107
- simple Select, 106–107

projections

- anonymous. **See anonymous types**
- explicit, 441

properties

- `EntityDataSource`, 557–558
- `EntityType` subgroups, 400–401
- timestamp, 527–528
- values, 560

Q

QBM (Query Builder Methods), 387–388, 491–492

quantifier operators

- All, 148–149
- Any, 148

Contains, 149

overview, 148

queries

- ADO.NET Data Services options, 333
- batch, 589–591
- conventional, 494–495
- copying results to `DataTables`, 260–266
- derived classes, 421–423
- Dim Query, 6
- expressions. **See query expressions**
- formatting output of, 159–160
- LINQ. **See LINQ (Language Integrated Query)**
- polymorphic, 453–454
- processing in PLINQ, 321–322
- query language, 328–329
- standard query operators. **See SQOs (standard query operators)**
- typed `DataSets`, 253–254
- URI query syntax, 329–332
- var, 6
- XML namespaces, 297–298, 304–305

queries, advanced

- aggregate operators. **See aggregate operators**
- `Contains()` SQO. **See Contains() SQO**
- `GroupBy`, with aggregate queries. **See GroupBy**, with aggregate queries
- mocking collections for testing projects. **See mocking collections**
- overview, 155
- query expression trees, 172–176
- summary, 187–188

queries, base classes

- disambiguate derived object types with base types, 425–427
- incorrect results from `is`, `Is`, `typeof` and `OfType` operators, 423–424
- overview, 421–423
- type discrimination in Entity SQL queries, 424–425

queries, compiled

- code examples, 87–88
- LINQ to Entities, 497–499
- overview, 87

queries, Entity SQL

- as client views. **See Entity SQL, queries as client views**
- against `EntityClient`. **See Entity SQL, queries against EntityClient**

queries, from Windows Form clients

- executing a `DataServiceQuery`, 585–586

executing a query-string URI, 583–585
 executing batch queries with
 DataServiceRequests, 589–591
 executing LINQ to REST queries, 586–587
 returning associated entities, 587–589

queries, nested

- NAVIGATE as a complex substitute for, 445–446
- one:many navigation, 444–445
- sorting collections requiring, 451

queries, parameterized

- execution of, 460–461
- ObjectQuery, 500

queries, read-only LINQ to DataSet

- customizing lookup lists, 253–254
- querying typed DataSets, 254–257
- querying untyped DataSets, 248–252

queries, URI

- adopting URI as a query language, 328–329
- executing, 337
- executing a query-string URI, 583–585
- experimenting with URI query syntax, 329–332

queries, XML Infosets

- C# 3.0 example, 271–272
- implicit v. explicit content typing, 276–278
- inferring a schema/enabling IntelliSense, 273–275
- overview, 271
- taking advantage of VB 9.0 axis properties, 275–276
- VB 9.0 example, 272–273

Query Builder Methods (QBM)s, 387–388, 491–492

query code, 346–348

query expressions

- advanced. *See* advanced queries
- overview, 80–83
- syntax of, 127–128
- translating, 348–349
- trees, 172–176

query language, 328–329

Query page, for eSqlBlast, 436–437

query pipeline, in LINQ to SQL, 212–215

query string options, 576–580

- comparison, logical, arithmetic, and grouping operators, 578
- DateTime, math and type functions, 579–580
- option descriptions, 577
- overview, 576

query-string URI, 583–585

R

Range **operator, 147**

ray tracing, 322

read-only LINQ to DataSet queries

- customizing lookup lists, 253–254
- querying typed DataSets, 254–257
- querying untyped DataSets, 248–252

RectangleUnion, PLINQ, 324

REF, **entity reference, 446–449**

relationship sets, 360

relationships

- EDM terminology definition, 360
- entities and, 196
- entity-relationship model, 360–362

Remove, **in XML extension method, 271**

removing objects, in LINQ to SQL, 215–219

Repeat **operator, 147**

Representational State Transfer. *See* REST (Representational State Transfer)

resources, relationship to objects, 395

REST (Representational State Transfer).

- See also* LINQ to REST**
- architecture, 570
- definition, 325

restriction operator

- compound Where expressions, 102–104
- method calls with index arguments and use of IndexOf, 104–105
- simple Where expressions, 102
- Where overview, 101

Result page, for eSqlBlast, 437–439

roles, EDM, 360

ROW

- entity reference, 446–449
- type constructors creating, 449–450

S

schema

- for *.dbml and *.xml files, 202–206
- enabling IntelliSense, 273–275
- OR Designer generated, 34
- physical and conceptual, 49–52
- stored procedures and, 223
- XML, 395–397
- XML documents, 42–47

schema, database

- *.dbml and *.xml files and, 202–206
 - OR Designer generated, 34
 - physical and conceptual, 49–52
 - XML documents, 42–47
- schema, mapping physical to conceptual**
- adding, updating, and deleting objects, 367–368
 - creating the XML mapping files and object layer class file, 365–367
 - editing `EntityType` and `AssociationSet` properties, 368–369
 - overview, 365
- `SELECT` (eSQL), **491**
- `Select` (QBM), **491**
- `Select` **project operators**
- index value expressions with the `Let` keyword, 107–108
 - multiple `select` projection expression, 107
 - overview, 106
 - simple `select` projection expression, 106–107
- `Select` **projection expression, 106–107**
- `Select` **property, EntityDataSource, 558**
- `SELECT` **stored procedure, 223–224**
- `SELECT VALUE` (eSQL), **491**
- `SelectedIndexChanged` **event handler, 545–546**
- `SelectMany` **project operator**
- basic implementation for associated objects, 108–109
 - overload for equi-joins of associated objects, 110–111
 - overview, 108
- `SelectValue` (QBM), **491**
- `SequenceEqual` **operator, 142–143**
- serialization, entity class, 477–479**
- server controls, ASP.NET. See**
- `EntityDataSource` **server control**
- service operations, invoking in ADO.NET Data Services, 580–581**
- services requests, processing. See**
- `NwindServicesClient` **services requests**
- set operators**
- `ANYELEMENT` and `FLATTEN` work on collections, 452
 - `Distinct`, 129
 - `Except`, 132
 - `Intersect`, 131–132
 - overview, 128

- sorting collections returned by, 451
 - `Union`, 129–131
- sets, entity**
- adding `FunctionImports` to populate, 521–522
 - adding new entities, 592–594
 - associated entities and, 507–508
 - batching changes, 597
 - data retrieval stored procedures assigned to, 405–406
 - deleting entities, 596–597
 - deleting members from collections, 238–242
 - EDM terminology definition, 360
 - elements or subgroups, 398
 - `EntitySetMapping`, 415–417
 - `EntitySetName`, 558
 - managing optimistic concurrency conflicts, 597
 - mapping tables to, 197–200
 - updating, 594–595
 - updating associated, 507–508
- SharePoint 3.0 SDK, 345. See also LINQ to SharePoint**
- SharePoint Mapping Language (SMPL), 345**
- `Single` **operator, 145–146**
- `SingleOrDefault` **operator, 145–146**
- `SKIP` (eSQL), **491**
- `Skip` (QBM)
- C# and VB examples, 112–113
 - composing QBMs, 491
 - partitioning operator, 112
- `skip`, **ADO.NET Data Services, 333**
- `SKIP` **subclauses, ORDER BY clause, 452–453**
- `SkipWhile`
- C# and VB examples, 114–115
 - partitioning operator, 114
- Slambda, in Entity Framework futures, 466–467**
- SMPL (SharePoint Mapping Language), 345**
- sorting collections, 451**
- SQL. See also LINQ to SQL**
- `IN` () function, 171–172
 - functions, modifiers and operators, 30
 - overview, 7, 61
 - Where clauses, 169–171
- SQL Server**
- cascading deletions, 238–239
 - facets, 401
 - OR Designer generated database, 30–34
- SQL Server Compact (SSCE), 5, 461–463**
- SQL Server (Express), 462–463, 494**
- `SqlMetal.exe`, **200–203**

SQOs (Standard Query Operators). See also specific SQOs

advanced query operators/expressions. See advanced queries
 C# 3.0 expressions, 80–82
 by group, 92–93
 as keywords in C3 3.0 and VB 9.0, 93–94
 lazy and eager evaluation, 82
 overview, 80
 unsupported, 494
 VB 9.0 expressions, 83

SSCE (SQL Server Compact), 5, 461–463**SSDL (Store Schema Definition Language), 370–371. See also StorageModels (SSDL content)****Standard Query Operators. See SQOs (standard query operators)****stop-and-go processing, 321****StorageModels (SSDL content)**

association subgroups, 409–411
 EntityContainer subgroup, 398–399
 EntityType subgroups, 400–401
 Function subelements and subgroups. See Function subelements and subgroups overview, 397
 Product EntityType subgroup, 402
 scanning StorageModels group, 370–371

Store Schema Definition Language (SSDL), 370–371. See also StorageModels (SSDL content)**stored procedures**

CRUD. See CRUD (create, retrieve, update, and delete)
 data retrieval to an EntitySet, 405–406
 entity instances deleted with, 528
 Function definitions for, 404–405
 INSERT, UPDATE AND DELETE procedures, 224–225
 inserting a new entity instance, 524–525
 population of associations with, 410–411
 SELECT procedure, 223–224
 substituting for dynamic SQL, 222–225
 update association mapping, 407–408

StoreGenerated Pattern facet, SQL**Server, 401**

StoreOriginal ValueInViewState property, EntityDataSource, 558

string query functions, 335**structs, anonymous, 269****subclauses, LIMIT, 452–453****subclauses, SKIP, 452–453****subelement, Property, 400–401****subqueries**

throwing exceptions, 454–455
 UNION, INTERSECT, OVERLAPS, and EXCEPT require, 450

Sum operator, 151**surrogate keys**

application of, 546
 autoincrementing primary key, 237

Synchronization Framework, Microsoft, 5**syntactic sugar, 159****syntax**

aggregate operators, 157–158
 C# grouping expression, 162
 dot-notation, 442–443
 expression syntax with Let, 158–159
 GroupBy with method call, 126–127
 GroupBy with query expression, 127–128
 method call with numerical operators, 157–158

System.Xml.Linq namespace, 269–271**T****table data**

C# 3.0, 31–32
 updating, 30–32
 VB 9.0, 32–34

table-per-hierarchy inheritance. See TPH (table-per-hierarchy) inheritance**tables**

data table, 30–34
 mapping to entities, 15–16, 197–200

Take operator, 112–113**TakeWhile operator, 113–115****temporary local aggregate variable, 158–159****test client, AJAX, 599–602****testing LINQ to SQL and LINQ to Entities. See mocking collections****ThenBy operator, 122****ThenByDescending operator, 123****third-party LINQ implementations, 7, 317–318, 342–344****tier, middle, 225****timestamp property, concurrency management and, 527–528****ToArray operator, 138****ToDictionary operator, 139–140**

ToList operator

- LINQ data binding with, 12–15
 - overview, 139
- ToLookup operator, **141–142**
- To . . . operators
- overview, 138
 - ToArray, 138
 - ToDictionary, 139–140
 - ToList, 139
 - ToLookup, 141–142
- Top (QBM), **491**
- \$top, **ADO.NET query string option, 577**
- TOP AND LIMIT (eSQL), **491**
- top-level entity instances, 540–543**
- TPH (table-per-hierarchy) inheritance**
- base and derived class queries, 421–423
 - disambiguate derived object types with base types, 425–427
 - incorrect results from `is`, `Is`, `TypeOf` and `OfType` operators, 423–424
 - overview, 419
 - specifying discriminator column/creating a derived class, 419–421
 - type discrimination in Entity SQL queries, 424–425
- Transact-SQL. See Entity SQL/Transact-SQL comparison**
- TREAT operator, **453–454**
- tuples, 71–72, 360**
- type discrimination, in Entity SQL queries, 424–425**
- type functions, in ADO.NET Data Services, 579**
- type operators, for polymorphic queries, 453–454**
- type query functions, in ADO.NET Data Services, 335**
- typed DataRows, 262–263**
- typed DataSets**
- joining, 35–36
 - querying, 254–257
- TypeOf operator, **423–424**
- types, anonymous**
- C# 3.0, 71
 - DataSets, 263–266
 - overview, 60
 - VB 9.0, 71–72
- types, base**
- disambiguate derived object types with base types, 425–427

- incorrect results from `is`, `Is`, `TypeOf` and `OfType` operators, 423–424
- overview, 421–423
- type discrimination in Entity SQL queries, 424–425

types, complex

- modeling, 529–531
- overview, 528
- reusing a `ComplexType` definition, 530–531
- working with, 528–531

typing, explicit, 276–278

typing, implicit

- of element and attribute content, 276–278
- of lambda expression, 79–80
- of local variables, 60, 67–68

U

unbound ComboBoxes, to specify associations

- adding event handlers, 543–544
- updating association sets, 545–546

unclassified keywords, 93

Unicode facet, **SQL Server, 401**

UNION (eSQL), **491**

Union (QBM), **129–131, 491**

UNION ALL (eSQL), **491**

UNION set operators, **450**

UnionAll (QBM), **491**

untyped DataSets

- joining, 36–37
- querying, 248–252

UPDATE stored procedures

- function mapping to `EntityTypes`, 406–407
- substituting for dynamic SQL, 224–225
- updating an entity instance and managing concurrency with original values, 525–527
- updating an entity instance and managing concurrency with timestamp property, 527–528

updates

- enforcing object deletion and addition operations for, 549–550
- LINQ to Entities vs. Out-of-Band SQL, 514–515
- LINQ to SQL, 215–219
- transacting, 508
- validating, 510–513

URI queries

- adopting URI as a query language, 328–329
- executing, 337
- executing a query-string URI, 583–585
- experimenting with URI query syntax, 329–332
- UserDeletingRows, **dependent records and**, 240–242

V**validating data additions and updates, 510–513**

\$value, **ADO.NET query string option**, 577

VALUE **modifier**, **Entity SQ**, 441–442

value objects, 528**values**

- aggregation of child entity values, 158
- composite primary key, 546–549
- concurrency management with, 525–527

value-type association keys, 239–240**var query, 6****variables, implicitly typed local, 60, 67–68****VB 9.0**

- axis properties of, 275–276
- inferring a schema for a XML document, 273–275
- literal XML construction with, 284–288
- PLINQ, 324–325
- XML namespaces in, 302–305

VB extensions

- anonymous methods and generic predicates, 75–77
- anonymous types, 71–72
- array initializers with object initializers, 70
- collection initializers, 70–71
- expression trees and compiled queries, 83–88
- extension methods, 72–75
- implicitly typed local variables, 67–68
- IQueryable<T> implementations, 88–89
- lambda expressions, 77–80
- object initializers, 68–69
- overview, 66–67
- standard query operators, 80–83

views, client

- EF components, 363–364
- executing Entity SQL queries as, 376–381
- working with, 375

VS2008 SP1 ExpressionTreeVisualizer, in SQOs, 84–86**W****WCF (Windows Communication Framework) Service templates, 4**

WCF DataContractJsonSerializer (DCJS), 477

WCF DataContractSerializer (DCS)
entity class serialization with, 477
XML from, 478, 479

Web service, creating and consuming in a browser, 571–572

WHERE (eSQL)

- clause constraints, 454–455
- composing QBM, 491
- compound, 102–104
- method calls with index arguments, 102–104
- simple, 102
- SQL clauses, 169–171

Where (QBM), 491

Where **clauses, in EntityDataSource**, 557–560

Where **expressions**

- compound, 102–104
- restriction operators, 102

Where **property, EntityDataSource**, 558

Windows Communication Framework (WCF) Service templates, 4**Windows Form clients, queries from**

- executing a DataServiceQuery, 585–586
- executing a query-string URI, 583–585
- executing batch queries with
DataServiceRequests, 589–591
- executing LINQ to REST queries, 586–587
- returning associated entities, 587–589

Windows form controls, binding to entities

- adding members to collections, 237
- autogenerating hierarchical data editing form, 233–236
- collection modifications, 236–237
- composite primary-key members. See composite primary keys
- deferred loading of associated objects with
Load() method, 537–538
- deferred loading with the Attach() method, 539–540
- deleting members from a collections, 238–242
- eager loading of multiple associated objects
with Include() method, 538–539
- entity edits, 236–237
- overview, 233, 534–535

Windows form controls, binding to entities (continued)

- persisting changes to the data store, 553–554
- persisting entity edits/collection modifications, 236–237
- selecting active top-level/associated entity instances, 540–543
- shaping object graphs, 535–537
- using unbound `ComboBoxes`, 543–546

WordScrambler, PLINQ, 324

X

XDocuments, 300–302

Xen, 268

XML. See also LINQ to XML

- literal queries, 304–305
- mapping files, 365–367
- Northwind.svc WCF Service in, 332
- object mismatch, 268
- objects, 39–40
- schema, 395–397
- transforming documents with literal code, 306–308
- from WCF `DataContractSerializer` (DCS), 478, 479

XML, integrating into the CLR

- minimizing XML/object mismatch, 268
- overview, 267–268
- querying XML with `Cω`, 269

XML documents

- C# 3.0, 40–41, 42, 44–45
- querying, 40–44
- transforming or creating, 44–47, 306–308
- VB 9.0, 41–42, 43, 45–47

XML editor, 205–206

***.xml files, 205–206**

XML Infosets

- C# 1.0 example, 279
- functional construction with C# 3.0, 280–284
- literal construction with VB 9.0, 284–288
- overview, 278–279
- VB 7.0 example, 280

XML Infosets, querying

- C# 3.0 example, 271–272
- implicit v. explicit content typing, 276–278
- inferring a schema/enabling IntelliSense, 273–275
- overview, 271
- taking advantage of VB 9.0 axis properties, 275–276
- VB 9.0 example, 272–273

XML namespaces

- examples, 295–296
- overview, 295
- transforming documents with multiple namespaces, 306–308

XML namespaces, in C# 3.0

- functionally constructing `XDocuments`, 300–302
- namespaces in C# LINQ to XML queries, 297–298
- removing expanded namespaces, 298–299

XML namespaces, in VB 9.0

- enabling IntelliSense for namespaces, 302–304
- multiple namespaces in XML literal queries, 304–305
- overview, 302

XSD, 48. See also LINQ to XSD