

Index

Symbols

- != (not equal) operator, 201**
- <= (less than or equal) operator, 201**
- <> (not equal) operator, 201**
- = (equality) operator, 201**
- >= (greater than or equal) operator, 201**
- [] operator (HQL), 197–198**

A

- Abstract Factory implementation (DAO), 252–253**
- ActionForm (Struts application example), 375–376**
- actions (Struts application example), 377**
- advice (AspectJ), 400**
- aggregate functions (HQL), 204**
- all predicate (HQL), 197–198**
- AndroMDA, 355–358**
- Ant**
 - Maven comparison, 327
 - SchemaExport/SchemaUpdate tool support, 341–342
- any predicate (HQL), 197–198**
- AOP (Aspect-Oriented Programming), 389. See also AspectJ; OOADP**
 - aspect languages, 397
 - component languages, 397
 - development process, 397–398
 - goals, 397
 - language implementation in AspectJ, 401–402
 - OOADP problems solved by, 396
 - weaving, 397, 401–402
- Apache DBCP, 66**
- architecture (Hibernate), 16–17**
- arrays, mapping, 157–158**
- <artifactId> element, 310**
- AspectJ. See also AOP**
 - advice, 400
 - AOP language implementation, 401–402
 - aspects, 400–401

- example class, 403
- Hibernate aspects, 404
- join points, 399
- overview, 398–399
- pointcuts, 400
- weaving, 401–402
- aspect languages, 397**
- Aspect-Oriented Programming. See AOP**
- aspects (AspectJ), 400–401**
- <assembly-descriptor> element, 245**
- assigned generator, 75**
- associations, 127–128**
 - bidirectional, 130, 158–160
 - <element> element, 129–130
 - <index> element, 128–129
 - lazy initialization, 131
 - ternary, 140
- avg(attribute) function (HQL), 204**

B

- bags, mapping, 156**
- beginTransaction() method, 226**
- bidirectional associations, 130, 158–160**
- binary data, mapping classes with, 93–95**
- booleans (HQL), 201–202**
- boolean wasCommitted() method, 227**
- boolean wasRolledBack() method, 227**
- browsing objects (HibernateIDE tool), 353**

C

- C3PO, 65–66**
- caching, 207–208**
 - EHCACHE, 216–218
 - persistent object cache, 213–214
 - query caches, 215–216
 - session caches, 216
 - SessionFactory caches, 212–215

- Survey application example, 208–211
- when not to use, 218
- calendar attributes, mapping classes with, 97–99**
- callbacks, 21–22**
- CD list servlet-based application example**
 - adding Hibernate to, 49–50
 - interface page, 50–51
 - servlet code, 51–53
- CD list standalone application example**
 - database table, 30
 - Java class, 28–30
 - loading the mapping, 38–39
 - data objects, finding, 45–46
 - deleting objects, 43
 - obtaining SessionFactory, 39
 - persistent classes, creating, 39–41
 - persistent classes, loading, 43–45
 - updating objects, 41–42
 - mapping document, 30–33
 - skeleton code, 33–38
- check attribute, 337–338**
- <class> element, 31–32, 70–72**
- classes, mapping**
 - basic classes, 88–93
 - with binary data, 93–95
 - with data/calendar attributes, 97–99
 - read-only classes, 99–102
 - serializable classes, 95–97
 - table-per-class inheritance mapping, 110–113
 - table-per-concrete-class inheritance mapping, 115–117
 - table-per-subclass inheritance mapping, 113–115
 - with versioning/timestamps, 102–107
- CMT (container managed transactions), 242–245**
- code generation, 262**
- code generation templating (XDoclet), 265**
- CodeGenerator finders, 348–349**
- Code Generator tool, 342–348**
- code scattering (OOADP), 394–396**
- code tangling (OOADP), 394–396**
- collection elements, 84–85**
- collections**
 - filtering, 171–172
 - HQL (Hibernate Query Language), 197–199
 - mapping, 127
 - arrays, 157–158
 - associations, 127–128, 129–131
 - bags, 156
 - bidirectional associations, 158–160
 - lazy initialization, 131
 - lists, 152–155
 - maps, 131–147

- Sets and SortedSets, 147–151
- tagging (XDoclet), 277–281
- <column> element, 32**
- column formulas, 122–124**
- comment blocks, 263–264**
- commit() method, 226**
- committing transactions, 223**
- <component> element, 82**
- component languages, 397**
- <composite-id> element, 75–76**
- concerns (OOADP), 391–392**
- Configuration class, 18–19**
- configuration file, 26**
- configuration file wizard (Hibernate Synchronizer), 370**
- configuring**
 - Hibernate properties, 58–64, 223–224
 - Struts
 - <plug-in> element, 378–382
 - struts-conf.xml file, 378
 - Tomcat, 46, 236
- connection pooling, 64–65**
 - with Apache DBCP, 66
 - with built-in pooling, 65
 - with C3PO, 65–66
 - with Proxool, 66–67
- container managed transactions (CMT), 242–245**
- controlling resultsets, 176–177**
- count(*) function (HQL), 204**
- count(distinct attribute) function (HQL), 204**
- count(query) function (HQL), 204**
- Criteria object, 181–184**
- crosscutting (OOADP), 393**

D

- DAO design pattern, 247–250**
 - Abstract Factory implementation, 252–253
 - example application
 - application code, 259–260
 - class diagram, 253
 - factories, building, 254–255
 - objects, building, 255–259
 - Factory Method implementation, 250–251
- Data Access Object design pattern. See DAO design pattern**
- data attributes, mapping classes with, 97–99**
- databases**
 - configuring for unit testing, 320–322
 - connecting to, 67–69
 - DB2, 413–414

DB2/400, 414
 Hibernate-supported, 55–56
 HypersonicSQL, 414
 Interbase, 414
 McKoi SQL, 415
 Microsoft SQL Server, 415
 MySQL, 416
 Oracle, 416
 PointBase, 416
 PostgreSQL, 417
 SAP DB, 417
 Sybase, 417–418
 transaction support, 224–225
database server hookup, 26–27
database tables, standalone CD list application example, 30
data objects, finding, 45–46
DB2/400 database, 414
DB2 database, 413–414
DBCP (Apache), 66
deleting objects, 166–168
dialect determination (SQL), 62–64
dirty reads (transactions), 225
<discriminator> element, 76–77
documenting code, 263–264

E

Eclipse

HiberClipse plug-in, 368–370
 Hibernate Synchronizer plug-in, 370–372
 Hibernate plug-in, 363–367
 integrating with Maven, 328
 plug-ins, 22

EHCACHE, 216–218

EJB (Enterprise JavaBeans)

finders, 348–349
 stateful session bean example, 240–242
 stateless session bean example, 238–239 **EJBGen, 262**

<element> element, 86, 129–130

elements predicate (HQL), 197–198

Enterprise JavaBeans. See EJB

enumerations. See Enums

Enums, mapping, 119–121

equality operators (HQL), 201

exists predicate (HQL), 197–198

F

factories, building (DAO), 254–255

Factory Method implementation (DAO), 250–251

fetch joins (HQL), 206

filtering collections, 171–172

finders, 348–349

finding

large resultsets, 171
 objects, 169–170

flushing objects, 166

forced rollbacks (transactions), 229

foreign generator, 75

foreign-key attribute, 336

formulas, 122–124

forums on Hibernate, 419–420

full joins (HQL), 206

functions, scalar, 200–201

G

<generator> element, 32, 73–75

getCurrentLockMode(Object) method, 233

getters, tagging (XDoclet), 282

goals (Maven), 311–312

greater than or equal (>=) operator, 201

group by clause (HQL), 204–205

<groupid> element, 310

groupings (HQL), 200

H

HAVING clause (HQL), 204–205

HiberClipse plug-in (Eclipse), 368–370

Hibern8IDE tool, 350–353

browsing objects, 353
 querying objects, 353–355

Hibernate

configuration file, 26
 database server hookup, 26–27
 downloading, 23–24
 forums, 419–420
 installing, 25
 issue tracking, 421
 mailing lists, 421
 Microsoft SQL Server setup, 27–28
 MySQL setup, 27
 overview, 15–16

- architecture, 16–17
- callbacks, 21–22
- configuration, 18–19
- persistent classes, 17–18
- server support, 17
- sessions, 20–21
- technologies supported, 22
- requirements, 24–25
- source code, 420–421
- wiki, 420

hibernate:aggregate-mappings goal (Maven), 317

- @hibernate.array tag, 284–285**
- @hibernate.bag (0..1) tag, 285**
- @hibernate.cache tag, 283**
- hibernate.cfg.xml file, generating, 268–270**
 - with Maven, 314, 316–317
- hibernate.cglib.use_reflection_optimizer property, 60**
- @hibernate.class tag, 282**
- @hibernate.collection-cache tag, 285–286**
- @hibernate.collection-composite-element tag, 286**
- @hibernate.collection-element tag, 286**
- @hibernate.collection-index tag, 286**
- @hibernate.collection-key-column (0..*) tag, 287**
- @hibernate.collection-key tag, 287**
- @hibernate.collection-many-to-many tag, 287**
- @hibernate.collection-one-to-many tag, 287–288**
- @hibernate.column tag, 288**
- @hibernate.component tag, 288**
- hibernate.connection.datasource property, 26, 58**
- hibernate.connection.driver_class property, 26**
- hibernate.connection.isolation property, 60–61**
- hibernate.connection.password property, 26, 27, 56, 58**
- hibernate.connection.pool_size property, 26, 58**
- hibernate.connection.provider_class property, 61**
- hibernate.connection.url property, 26, 58**
- hibernate.connection.username property, 26, 27, 58**
- hibernate.default_schema property, 59**
- hibernate.dialect property, 26, 59, 62–64**
- @hibernate.discriminator tag, 283**
- @hibernate.generator-param tag, 288**
- @hibernate.id tag, 289**
- @hibernate.index-many-to-many tag, 289**
- hibernate.jdbc.batch_size property, 60**
- hibernate.jdbc.fetch_size property, 60**
- hibernate.jdbc.use_scrollable_resultset property, 60**
- hibernate.jdbc.use_streams_for_binary property, 60**
- hibernate.jndi.<property> property, 60**
- hibernate.jndi.class property, 27, 59**
- hibernate.jndi.url property, 27, 59**
- @hibernate.joined-subclass-key tag, 284**
- @hibernate.joined-subclass tag, 283**

- @hibernate.list tag, 290**
- @hibernate.many-to-one tag, 290**
- @hibernate.map tag, 291**
- hibernate.max_fetch_depth property, 59**
- @hibernate.one-to-one tag, 291**
- @hibernate.primitive-array tag, 292**
- @hibernate.property tag, 292**
- @hibernate.query tag, 284**
- hibernate:schema-export goal (Maven), 317–319**
- hibernate.Session_factory_name property, 59**
- @hibernate.set tag, 292–293**
- hibernate.show_sql property, 61–62**
- @hibernate.subclass tag, 284**
- @hibernate.timestamp tag, 293**
- hibernate.transaction.factory_class property, 61**
- hibernate.use_outer_join property, 59**
- @hibernate.version tag, 293**
- hibernatedoclet task, 266–267**
- Hibernate Extensions**
 - CodeGenerator finders, 348–349
 - Code Generator tool, 342–348
 - Hibern8IDE tool, 350–353
 - browsing objects, 353
 - querying objects, 353–355
 - Map Generator tool, 349–350
- <hibernate-mapping> element, 31–32, 69–70**
- Hibernate plug-in for Maven, 317–319**
- Hibernate Query Language. See HQL**
- Hibernate Synchronizer plug-in (Eclipse), 370–372**
- Hibernator plug-in (Eclipse), 363–364**
 - creating schemas, 367
 - database connection, 364–366
 - mapping documents, creating, 364–366
 - queries, 367
 - updating schemas, 367
- hilo generator, 74**
- hooking up database servers, 26–27**
- HQL (Hibernate Query Language)**
 - aggregate functions, 204
 - booleans, 201–202
 - collections, 197–199
 - equality operators, 201
 - group by clause, 204–205
 - groupings, 200
 - HAVING clause, 204–205
 - joins, 205–206
 - logical operators, 199
 - named parameters, 201
 - newsgroup example
 - application code, 191–192
 - classes, 185–189

- mapping document, 189–190
- rows, 192
- tables, 191
- NULL expression, 199–200
- order by clause, 203
- polymorphism, 202–203
- query substitution, 62
- scalar functions, 200–201
- SELECT clause, 194–196
- subselects, 202
- WHERE clause, 196–197

HypersonicSQL database, 414

I

- <id> element, 32, 72–73**
- identity generator, 73**
- increment generator, 73**
- index() property, 198**
- <index> element, 86, 128–129**
- index attribute, 337**
- indices predicate (HQL), 197–198**
- inheritance, 8–9**
 - lowest-child inheritance mapping, 9–10
 - table-per-class, 10–11, 110–113
 - table-per-concrete-class, 11, 115–117
 - table-per-subclass, 113–115
- inheritance hierarchies, 107–110**
- inner joins (HQL), 206**
- InnoDB table type (MySQL transactions), 228–229**
- in predicate (HQL), 197–198**
- integrity. See transactions**
- Interbase database, 414**
- Interceptor interface**
 - building interceptors, 408–412
 - methods, 407–408
- interfaces, persisting, 117–119**
- ISAM table type (MySQL transactions), 228**
- issue tracking, 421**

J

- J2EE, 22**
 - application servers, 235
 - CMT (container managed transactions), 242–245
 - DAO design pattern, 247–250
 - Abstract Factory implementation, 252–253
 - example application, 253–260

- Factory Method implementation, 250–251
- Hibernate configuration file, 236–237
- SessionFactory singleton class, 237–238
- stateful session bean example, 240–242
- stateless session bean example, 238–239
- Tomcat configuration, 235–237
- JARs, producing (Maven), 311**
- Java classes, creating (standalone CD list application example), 28–30**
- Javadoc, 263–264**
 - with Maven, 311
- Java Naming and Directory Interface (JNDI), 17**
- Java Transaction API (JTA), 17**
- Java Virtual Machine (JVM), 207**
- JBoss, 270–272**
- JDBC, 17**
 - configuring connections (HiberClipse), 369
 - properties, 58–62
- JDBC database connections, 67–69**
- JIRA issue tracker, 421**
- JNDI (Java Naming and Directory Interface), 17, 236**
- <joined-subclass> element, 83–84**
- join points (AspectJ), 399**
- joins (HQL), 205–206**
- JTA (Java Transaction API), 17**
- jta.UserTransaction property, 61**
- JVM (Java Virtual Machine), 207**

K

- <key> element, 85**
- keys, primary, 8**

L

- large resultsets, finding, 171**
- lazy initialization, 131**
- left outer joins (HQL), 205–206**
- length attribute, 334**
- less than or equal (<=) operator, 201**
- like operator, 201**
- lists, mapping, 152–155**
- load() method, 233**
- loading data into objects, 164–165**
- lock() method, 233**
- locking, 230–233**
- LockMode.NONE lock, 233**
- LockMode.READ lock, 233**

LockMode.UPGRADE lock, 233
LockMode.UPGRADE-NO WAIT lock, 233
LockMode.WRITE lock, 233
LockMode getCurrentLockMode(Object) method, 233
logical operators (HQL), 199
lower() function (HQL), 200–201
lowest-child inheritance mapping, 9–10

M

mailing lists, 421
<many-to-many> element, 135–140
many-to-many relationships, 13–14
<many-to-one> element, 80–81
Map Generator tool, 349–350
mapping
 classes
 basic classes, 88–93
 with binary data, 93–95
 with data/calendar attributes, 97–99
 inheritance hierarchies, 107–110
 read-only classes, 99–102
 serializable classes, 95–97
 table-per-class inheritance mapping, 110–113
 table-per-concrete-class inheritance mapping, 115–117
 table-per-subclass inheritance mapping, 113–115
 with versioning/timestamps, 102–107
 collections, 127
 arrays, 157–158
 associations, 127–131
 bags, 156
 bidirectional associations, 158–160
 lazy initialization, 131
 lists, 152–155
 maps, 131–147
 Sets and SortedSets, 147–151
 Enums, 119–121
 inheritance, 8–9
 lowest-child inheritance mapping, 9–10
 table-per-class inheritance mapping, 10–11
 table-per-concrete-class inheritance mapping, 11
 loading into an application, 38–39
 data objects, finding, 45–46
 deleting objects, 43
 obtaining SessionFactory, 39
 persistent classes, creating, 39–41
 persistent classes, loading, 43–45
 updating objects, 41–42

overview, 3–7
primary keys, 8
relationships, 11
 many-to-many, 13–14
 one-to-many, 12–13
 one-to-one, 11–12
timestamps, 8
version numbers, 8
mapping documents, 69
 <class> element, 31–32, 70–72
 <component> element, 82
 <composite-id> element, 75–76
 creating
 with Eclipse, 364–366
 with HiberClipse, 369
 with Hibernate Synchronizer, 371–372
 with Maven, 313–315, 319–320
 <discriminator> element, 76–77
 <element> element, 86, 129–130
 <generator> element, 32, 73–75
 <hibernate-mapping> element, 31–32, 69–70
 <id> element, 32, 72–73
 <index> element, 86, 128–129
 <joined-subclass> element, 83–84
 <key> element, 85
 <many-to-many> element, 131–135
 <many-to-one> element, 80–81
 newsgroup example (HQL), 189–190
 <one-to-one> element, 81–82
 <property> element, 78–80
 formula attribute, 122–124
 standalone CD list application example, 30–33
 <subclass> element, 82–83
 <timestamp> element, 78
 <version> element, 77
mapping document wizard (Hibernate Synchronizer), 371–372
maps
 mapping, 131
 object maps, 131–135
 TreeMaps, 140–147
 values maps, 131–135
Maven, 22
 Ant comparison, 327
 compiling Java code, 311
 database configuration for unit testing, 320–322
 downloading, 308
 Eclipse integration, 328
 goals, 311–312
 Hibernate files, generating, 312

- hibernate.cfg.xml file, 314
- mapping files, 313–314, 319–320
- XDoclet plug-in, 313
- Hibernate plug-in, 317
 - hibernate:aggregate-mappings goal, 317
 - hibernate:schema-export goal, 317–319
- installing, 308
- JARs, producing, 311
- Javadocs, generating, 311
- with multiple Ant tasks, 327
- overview, 308
- plug-ins, 311–312
- POM (Project Object Model), 309–310
- project organization, 328
- properties, 311–312
- scripting
 - generating hibernate.cfg.xml file, 316–317
 - generating mapping files, 314–315
 - setting up dependencies, 314–315
- setup, 328
- Survey application example code, 322–327

max(attribute) function (HQL), 204

maximum index values (HQL), 197–198

MBean descriptor, generating, 270–272

McKoi SQL database, 415

Microsoft SQL Server database, 415

Microsoft SQL Server setup, 27–28

MiddleGen, 358–361

min(attribute) function (HQL), 204

minimum index values (HQL), 197–198

MySQL, 416

- setup, 27
- transaction example, 227–228
 - forced rollbacks, 229
 - InnoDB table type, 228–229
 - ISAM table type, 228
 - optimal session use, 229–230

N

named parameters, 179–180, 201

named queries, 178–180

native generator, 75

newsgroup example (HQL)

- application code, 191–192
- classes, 185–189
- mapping document, 189–190
- rows, 192

- tables, 191

non-repeatable reads (transactions), 225

nonrestrict read-write cache usage, 214

not equal operators, 201

not-null attribute, 335

NOT operator (HQL), 199

NULL expression (HQL), 199–200

O

object maps, mapping, 131–135

object-oriented analysis, design, and programming. See OOADP

object-oriented database systems, 3

object references, tagging, 274–277

Object Relational mapping model, 15

objects, 390

- browsing (Hibern8IDE tool), 353
- DAO, building, 255–259
- data objects, finding, 45–46
- deleting, 43
- locking, 230–233
- mapping. *see* mapping overview
- persistent
 - creating, 161–164
 - Criteria object, 181–184
 - deleting, 166–168
 - filtering collections, 171–172
 - finding, 169–170
 - flushing, 166
 - large resultsets, finding, 171
 - loading data into, 164–165
 - named, 178–180
 - previous sessions, using, 165–166
 - refreshing, 168
 - scalar queries, 172–173
 - SQL, 181
 - straight queries, 173–178
 - timeouts, 180–181
 - updating, 168–169
- querying (Hibern8IDE tool), 353–355
- serializing, 2
- updating, 41–42

one-to-many relationships, 12–13

<one-to-one> element, 81–82

one-to-one relationships, 11–12

OOADP (object-oriented analysis, design, and programming), 390

- benefits of, 390–392
- code scattering, 394–396
- code tangling, 394–396
- concerns, 391–392
- crosscutting, 393
- objects, 390
- problems resulting from, 392–395
- tangled code, 396

open source resources, 422

AND operator (HQL), 199

optimal session use, 229–230

Oracle database, 416

order by clause (HQL), 203

O/R mapping model, 15

OR operator (HQL), 199

P

parameters, named, 179–180, 201

persistent classes, 17–18

- creating
 - CD list application example, 39–41
 - guidelines, 87–88
- loading, 43–45

persistent object caches, 213

- nonrestrict read-write cache usage, 214
- read-only cache usage, 213–214
- read-write cache usage, 214
- transactional cache usage, 214

persistent objects

- creating, 161–164
- Criteria object, 181–184
- deleting, 166–168
- filtering collections, 171–172
- finding, 169–170
- flushing, 166
- large resultsets, finding, 171
- loading data into, 164–165
- previous sessions, using, 165–166
- queries
 - named, 178–180
 - scalar, 172–173
 - SQL, 181
 - straight, 173–178
 - timeouts, 180–181
- refreshing, 168
- updating, 168–169

persisting interfaces, 117–119

phantom inserts (transactions), 225

<plug-in> element (Struts), 378–382

plug-ins (Maven), 311–312

PointBase database, 416

pointcuts (AspectJ), 400

polymorphism (HQL), 202–203

POM (Project Object Model), 307, 309–310

pools. See connection pooling

PostgreSQL database, 417

previous sessions, using, 165–166

primary keys, 8

project.xml file, 309

Project Object Model. See POM

properties

- fully specifying (XDoclet), 281
- Maven, 311–312
- tagging (XDoclet), 272–274

<property> element, 32, 78–80, 122–124

Proxool, 66–67

Q

queries

- aggregate functions, 204
- booleans, 201–202
- caches, 208, 215–216
- collections
 - [] operator, 198–199
 - all predicate, 198–199
 - any predicate, 198–199
 - elements predicate, 198–199
 - exists predicate, 198–199
 - index() property, 198
 - indices predicate, 198–199
 - minimum/maximum index values, 197–198
 - in predicate, 198–199
 - size property, 197
 - some predicate, 198–199
- equality operators, 201
- group by clause, 204–205
- groupings, 200
- HAVING clause, 204–205
- Hibernate plug-in, 367
- joins, 205–206
- logical operators, 199
- named, 178–180
- named parameters, 201
- newsgroup example
 - application code, 191–192
 - classes, 185–189

- mapping document, 189–190
- rows, 192
- tables, 191
- NULL expression, 199–200
- order by clause, 203
- polymorphism, 202–203
- query substitution, 62
- scalar, 172–173
- scalar functions, 200–201
- SELECT clause, 194–196
- SQL, 181
- SQL dialect determination, 62–64
- straight, 173–178
 - controlling resultsets, 176–177
 - single row queries, 178
- subselects, 202
- substitution, 62
- timeouts, 180–181
- WHERE clause, 196–197
- querying objects (Hibern8IDE tool), 353–355**
- Query interface, 20**

R

- read-only cache usage, 213–214**
- read-only classes, mapping, 99–102**
- read-write cache usage, 214**
- refresh(Object, LockMode) method, 233**
- refreshing objects, 168**
- RegisterForm (Struts application example), 376**
- registration system example**
 - ActionForm, 375–376
 - actions, creating, 377
 - compiling, 387
 - configuration file, 382
 - flow diagram, 373–374
 - register.vm template, 385–387
 - RegisterAction code, 377
 - RegisterForm, 376
 - register page, 385–387
 - running, 387
 - success.vm template, 385
 - success page, 385
 - toolbox.xml file, 385
 - web.xml file, 382–384
- relationship elements, 80**
- relationships, 11**
 - many-to-many, 13–14
 - one-to-many, 12–13

- one-to-one, 11–12
- resources**
 - forums, 419–420
 - issue tracking, 421
 - mailing lists, 421
 - on open source, 422
 - source code, 420–421
 - wiki, 420
- resultsets, controlling, 176–177**
- right outer joins (HQL), 206**
- rollback() method, 226**
- rolling back transactions, 223**
- rows, newsgroup example (HQL), 192**

S

- SAP DB database, 417**
- scalar functions, 200–201**
- scalar queries, 172–173**
- SchemaExport/SchemaUpdate tool, 329–333**
 - advanced options, 333–334
 - check attribute, 337–338
 - foreign-key attribute, 336
 - index attribute, 337
 - length attribute, 334
 - not-null attribute, 335
 - sql-type attribute, 334–335
 - unique attribute, 337
 - unique-key attribute, 338
 - Ant support, 341–342
 - application support, 340–341
 - command-line options, 332
 - updating tables, 338–340
- schemas**
 - creating, 367
 - exports, creating (HiberClipse), 370
 - updating
 - HiberClipse plug-in, 370
 - Hibernate plug-in, 367
- scripting (in Maven)**
 - generating hibernate.cfg.xml file, 316–317
 - generating mapping files, 314–315
 - setting up dependencies, 314–315
- scroll() method, 176**
- SELECT clause (HQL), 194–196**
- seqhilo generator, 74**
- sequence generator, 73–74**
- serialization, 2**

servers

- database server hookup, 26–27
- Hibernate support, 17
- multiple, connecting to, 68–69
- Tomcat
 - configuring, 46
 - downloading, 46
 - installing, 46
 - setting up database access, 47–48

servlet-based CD list application example

- adding Hibernate to, 49–50
- interface page, 50–51
- servlet code, 51–53

session caches, 216

SessionFactory caches, 212–213, 214–215

SessionFactory class, 20

- J2EE example, 237–238
- obtaining, 39

SessionFactory-level class (caching), 207

Session object, 20

session pattern, 124–126

sessions

- locking, 230–233
- optimal use, 229–230

setFirstResults() method, 176

setLockMode() method, 233

setMaxResults() method, 176

Sets, mapping, 140–147

setters, tagging (XDoclet), 282

setTimeout() method, 180–181

shopping cart EJB example, 240–242

single row queries, 178

size property (HQL), 197

some predicate (HQL), 197–198

SortedSets, mapping, 140–147

source code (Hibernate), 420–421

SQL

- dialect determination, 62–64
- in formula attributes, 122–124
- queries, 181

SQL Server database, 415

sql-type attribute, 334–335

standalone CD list application example

- database table, 30
- Java class, 28–30
- loading the mapping, 38–39
 - data objects, finding, 45–46
 - deleting objects, 43
 - obtaining SessionFactory, 39
 - persistent classes, creating, 39–41

- persistent classes, loading, 43–45
 - updating objects, 41–42
- mapping document, 30–33
- skeleton code, 33–38

stateful session bean example, 240–242

stateless session bean example, 238–239

straight queries, 173–178

- controlling resultsets, 176–177
- single row queries, 178

Struts

- configuring
 - <plug-in> element, 378–382
 - struts-conf.xml file, 378
- installing, 374
- registration system example
 - ActionForm, 375–376
 - actions, creating, 377
 - compiling, 387
 - configuration file, 382
 - flow diagram, 373–374
 - register.vm template, 385–387
 - RegisterAction code, 377
 - RegisterForm, 376
 - register page, 385–387
 - running, 387
 - success.vm template, 385
 - success page, 385
 - toolbox.xml file, 385
 - web.xml file, 382–384

struts-conf.xml file, 378

<subclass> element, 82–83

subselects (HQL), 202

sum(attribute) function (HQL), 204

supported databases, 55–56

SupportProperty class, 131–132

Survey application example

- caching, 208–211
- Maven
 - compiling Java code, 311
 - complete source code, 322–327
 - goals, 311–312
 - JARs, producing, 311
 - Javadocs, generating, 311
 - plug-ins, 311–312
 - properties, 311–312
- XDoclet, 266–268
 - collections, tagging, 277–281
 - final code, 294–304
 - fully specifying properties, 281
 - getters versus setters, tagging, 282

hibernate.cfg.xml file, generating, 268–270
 MBean descriptor, generating, 270–271
 object references, tagging, 274–277
 properties, tagging, 272–274

Sybase database, 417–418

T

table-per-class inheritance mapping, 10–11, 110–113

table-per-concrete-class inheritance mapping, 11, 115–117

table-per-subclass inheritance mapping, 113–115

tables, newsgroup example (HQL), 191

tagging source files (XDoclet)

collections, 277–281
 fully specifying properties, 281
 getters versus setters, 282
 object references, 274–277
 properties, 272–274

tangled code, 396

tangled code (OOADP), 394–396

ternary associations, 140

timeouts (queries), 180–181

<timestamp> element, 78

timestamps, 8, 102–107

Tomcat

CMT (container managed transactions), 242–245
 configuring, 46
 for JNDI, 236
 downloading, 46
 installing, 46
 setting up database access, 47–48

toolbox.xml file (Struts application example), 385

transactional cache usage, 214

transactions, 223

basic use example, 225–226
 CMT (container managed transactions), 242–245
 committing, 223
 configuration options, 223–224
 database support, 224–225
 dirty reads, 225
 errors, 222
 MySQL example, 227–228
 forced rollbacks, 229
 InnoDB table type, 228–229
 ISAM table type, 228
 optimal session use, 229–230
 non-repeatable reads, 225

phantom inserts, 225

rolling back, 223

Transaction API, 226–227

<transaction-type> element, 245

<trans-attribute> element, 245

TreeMaps, mapping, 140–147

U

unique attribute, 337

unique-key attribute, 338

uniqueResult() method, 178

unit testing, configuring databases for, 320–322

updating objects, 168–169

upper() function (HQL), 200–201

uuid.hex generator, 74–75

uuid.string generator, 75

V

values maps, mapping, 131–135

Velocity. See registration system example

<version> element, 77

versioning, mapping classes with, 102–107

version numbers, 8

W

wasCommitted() method, 227

wasRolledBack() method, 227

weaving (AOP), 397, 401–402

web.xml file (Struts application example), 382–384

WHERE clause (HQL), 194–196

wiki, 420

X

XDoclet

best practices, 304

development of, 262

hibernate.cfg.xml file, generating, 268–270

installing, 266

JAR files, 266

Maven plug-in, 313

MBean descriptor, generating, 270–272

- modules, 262
 - overview, 262–266
 - code generation templating, 265
 - development simplification, 265–266
 - Javadoc and, 263–264
 - support for, 262
 - Survey application example code, 294–304
 - tagging source files
 - collections, 277–281
 - fully specifying properties, 281
 - getters versus setters, 282
 - object references, 274–277
 - properties, 272–274
 - tags, 282–293
 - task configuration, 267
 - task definition, creating, 266–267
- XDoclet2, 262**
- XDoclet Spring, 22**
- XML, 2–3**