

A

abstract 78
 abstract entity 292
 ACC. *See* application client container
 access
 EJB 3 session bean from EJB
 2.x bean 503
 type 420
 access type 229, 636
 accessType 230
 ACID property 177–178
 acknowledgement mode 130
 acknowledgeMode 130
 ActionBazaar 41–221
 domain model 218
 fake bid 204
 messaging 112–113
 problem domain 220
 security problem 203
 specials 112, 117
 transactional problem 178
 web service 539
 ActionBazaarLogger 160
 activation 80, 100
 activationConfig property 128
 @ActivationConfigProperty
 125, 127, 513
 Active Directory 205, 591
 ActiveMQ, 112
 actor 222
 adapted home 508
 addBid method 45
 administered object 118
 allocationSize 268

American National Standards
 Institute 601
 annotation
 inheritance 155
 in session beans 78
 oracle.j2ee.ejb.Stateless-
 Deployment 408
 Overriding 406
 vendor specific 408
 vs. XML in O-R
 mapping 260
 annotation metadata. *See* meta-
 data annotation
 anonymous beach dweller 115
 Ant 401
 AOP 30, 74
 Also see Aspect-Oriented Pro-
 gramming
 Apache 75
 Apache Derby 654
 application assembler 388
 application client 108
 application client container 104
 Also see container
 application layer. *See* service
 layer
 application server 14, 19, 184
 application-client.xml 56
 @ApplicationException
 193–194
 rollback element 195
 ApplicationException 193
 application-exception 632
 around invoke 158, 162
 @AroundInvoke 159
 restrictions 162

ArrayIndexOutOfBoundsException
 196
 AspectJ 12, 157
 AspectJ in Action 157
 Aspect-Oriented
 Programming 157
 assembler 203
 Also see application assembler
 assembly 388
 assembly-descriptor 404, 630
 association field 304
 association. *See* relationship
 association table 281, 473
 Also see intersection table 256
 asynchronous 57, 113
 communication 111
 Asynchronous JavaScript And
 XML 429
 atomicity 179
 atomicity, consistency, isolation
 and durability. *See* ACID
 property
 attached
 managed 298
 Also see entity state
 attribute 225
 attribute-oriented
 programming 36
 @AttributeOverride 273
 auditing 157
 authentication 202, 205
 in EJB 207
 mechanism 207
 authenticationType 151
 @Author 36
 authorization 202, 205

- AUTO_ACKNOWLEDGE
136
acknowledgement mode 130
automated persistence 224
automatic schema
 generation 261
 for Hibernate 417
 TopLink 417
automatic table generation 262
- B**
-
- B2B. *See* business-to-business
backing bean 48, 104
@Basic 260, 265, 321
basic element 638
basic profile 547
batch job 167
BEA WebLogic 19, 468
bean instance 83, 91, 141
bean interface. *See* interface
Bean Managed
 Transaction 184, 196–201
begin method 199
behavior 222
best practice
 packaging and
 deployment 422
bid entity 44, 47, 63–65
BidderAccountCreatorBean 95,
 103
BidManager 83
BIDS table 64
BidStatisticsTracker 160
binary large object 260, 265
BLOB 321, 361
 Also see binary large object
blueprint 426
BMT. *See* Bean Managed Trans-
 action
boilerplate code 27
boolean property 120
bootclasspath 395
bootstrap 124
bottleneck 463
business interface. *See* interface
business method 78
business rule 219
business-to-business 113
byte stream 590
BytesMessage 121
- C**
-
- caching 18, 320
 Also see entity caching
callback lifecycle
 time delayed 168
 Also see lifecycle callback
canned solution 21
cardinality. *See* multiplicity
cascade 242
 in databases 318
 merge 328
 persist 317–318
 types 318
CascadeType 317
CategoryPK 236
CaveatEmptor 41
certificate 207
character large object 265
checked exception 165
class diagram 219
class loader 394
 application 396
 boot 395
 EAR 398
 EJB 398
 extension 395
 system 396
class loading
 delegation model 396
ClassNotFoundException 423
CLASSPATH 394
client 83
 expectation 79
 Also see session bean client
CLOB 361
 Also see character large object
cloned 298
cluster 106
clustered server farm 8
clustering 14
 EJB 106
 EJB applications 488–496
 stateful session bean 493
CMP entity bean 380, 514
CMR. *See* container-managed
 relationship
CMT 451
 Also see Container-Managed
 Transaction
coarse grained 435–436, 485,
 514
Codd, E. F. 597
@Column 64, 224, 262
column 597
columnDefinition 263
commit 177, 179
 two-phase 183
compatibility 500
 backward 508
 between platforms 540
 with EJB 2.x clients 512
compiler 124
component 5, 15
composite key. *See* primary key
composite primary key 238,
 278
concurrency 73, 458
concurrency issue 457
concurrent 105
 system 180
confirmOrder method 52
connection 117, 136
 database 80
 JMS. *See* javax.jms.Connec-
 tion
 object 91
connection factory 117–118
 Also see javax.jms.Connection-
 Factory
connection pool 415, 468
connectionFactoryJndiName
 129
connector 122
consistency 179
constraints 599
 database 180
constructor 78
consumer 114
 single 115
container 15, 48, 61
 application client 55, 156
 EJB 20, 450
 Java EE 19, 450
 lightweight 451, 578
 services 297
 web 19
container services. *See* EJB, ser-
 vices
container-managed
 relationship 249

Container-Managed
 Transaction 184–195
 container-transaction 631
 conversation 79
 conversational state 79, 93, 106
 CORBA 5, 158, 505, 542, 590
 coupling
 loose 433, 435
 tight 434, 436
 create, read, update, delete 17
 createEntityManager
 method 453
 createEntityManagerFactory
 312
 CreateException 508, 527
 createNamedQuery 470
 method 349
 createNativeQuery method 381
 createQuery method 343
 createTimer method 171
 credentials 207
 CreditCardChargeService 540
 CreditValidationException 193
 CreditVerificationService 540
 cron 167
 crosscutting 74
 concerns 157, 212, 334
 CRUD 65, 294, 313, 431, 452
 Also see create, read, update,
 delete
 cryptography library 395

D

DAO. *See* data access object;
 design pattern
 data access object 260, 338, 429
 data integrity 599
 data modeling 223
 data source 86
 data transfer object 430, 502
 anti-pattern 328
 data type
 custom 233
 identity 234
 persistent field/property 232
 database 365
 administrator 388
 connection 86
 normalization 602
 object-oriented 597

 relational 597
 table 225, 233, 597
 DataSource
 configuration 415
 DATE 265
 DB2 9
 DBA 223, 252
 Also see database, administrator
 DDD. *See* domain-driven design
 dead message queue 138
 @DeclareRoles 208
 default-entity-listeners 336
 denormalization 602
 @DenyAll 209
 dependency 80
 dependency injection 27, 40–
 41, 75, 118, 146–155, 439,
 444, 450, 573
 avoiding 442
 EJB 3.0 48
 EJBContext 152
 environment entry 153
 from EJB 2.x 503
 JMS resource 152
 of POJO 581
 setter vs. field 150
 TimerService 169
 web service 550
 deploy
 algorithm 392
 application 392
 deployer 203, 388
 deployment 388
 deployment descriptor 26, 37–
 38, 185, 402, 408, 505
 application.xml 391
 persistence.xml 410–411
 proprietary 409
 reference 623–645
 vendor specific 519
 Derby 380
 Also see Apache Derby
 descriptor. *See* deployment
 descriptor
 deserialization 487
 design pattern 426–439
 data transfer object 517
 sequence generator
 pattern 521
 session façade design 485

 transfer object 529
 value object 529
 destination 56, 112
 destinationName 129
 destinationType 129
 detach 300
 DI. *See* dependency injection
 dirty read 457
 DiscountVerifierInterceptor
 164
 discriminator
 column 285, 642
 discriminator value 286
 @DiscriminatorColumn 286,
 288, 468
 DiscriminatorType 468
 @DiscriminatorValue 286, 288,
 468
 disk thrashing 106
 DNS. *See* Domain Name Service
 domain
 model 219
 object 220
 as Java class 225
 implementing with
 JPA 228
 domain model 219–224, 534
 anemic 11, 223, 515
 modeling 12
 rich 12, 223
 Domain Name Service 591
 domain object 11, 227
 domain-driven design 9, 11
 DriverManager 309
 DTO. *See* data transfer object
 DUPS_OK_ACKNOWLEDGE
 acknowledgement mode 130
 durability 180
 durable subscriber 131
 durable subscription 130

E

eager fetching. *See* eager loading
 eager loading 320, 378, 470
 EAO 577
 Also see entity access object
 EAR 37, 391
 Also see enterprise application
 archive
 Eclipse 659

- Eclipse Dali project 422
- @EJB 48, 103, 146, 154, 441, 590
 - class-level usage 156
 - inject home object 502
- EJB 5, 21
 - and Spring. *See* Spring: with EJB 3
 - context. *See* EJBContext
 - darkest hours 122
 - design practices 484
 - magic of 142
 - object 141
 - roles 388
 - services 141
- EJB 3 persistence provider. *See* persistence provider
- EJB framework 8
- EJB QL 355
- ejbActivate 26
- ejb-class 37
- EJBContext 143–146, 186, 191, 211
- ejbCreate 26
- EJBException 190, 194
- ejb-jar 37, 390
 - version 501
- ejb-jar.xml 37, 624
 - reference 624
- EJBLocalHome 527
- ejb-local-ref 441, 443, 504, 626
- ejb-name 37
- ejbPassivate 26
- EJBQL 28, 380
- ejb-ref 441, 443, 626
- ejbRemove 26
- @Embeddable 233, 238, 260
- embeddable 239
 - mapping 270
 - object 233
- @Embedded 224, 260
- embedded object
 - nested 239
- @EmbeddedId 237–238
- EMPTY comparison
 - operator 366
- ENC 148, 446
 - Also see* environment naming context
- enterprise application archive 390
- enterprise archive. *See* EAR
- enterprise bean provider 388
- enterprise beans 624
- Enterprise Information System 182
- Enterprise Java Beans. *See* EJB
- entities
 - packaging within an EJB
 - module 410
 - @Entity 64, 224, 228, 230, 518
 - entity 15, 18, 28, 228
 - a set of trade-offs 297
 - bean 23
 - create new record 314
 - detachment and merge
 - operations 326
 - fetch mode 321
 - identity. *See* identity
 - life cycle 297–301
 - callback 333
 - listener 334
 - locking 457
 - how optimistic locking works 459
 - locking types 458
 - managed 295
 - managing state 298
 - mapping 258
 - optimistic locking with times-tamp or version
 - column 459
 - packaging 391, 410
 - persisting 314
 - read-only 483
 - reattach 327
 - retrieval 319–325
 - state 297
 - version attribute 461
 - version column 461
 - versioning 461
 - entity access object 338, 427, 430, 574
 - factory 433
 - using session bean 433
 - entity bean 27, 218
 - entity caching 478
 - best practices 483
 - clustered cache 495
 - distributed cache
 - synchronization 495
 - in TopLink Essentials 482
 - named query 482
 - persistence unit level
 - cache 481
 - query 478
 - read-mostly 478
 - read-only 478
 - with extended persistence
 - context 479
 - entity listener 420
 - entity manager 448
 - container managed 448
 - entity performance tuning 464
 - avoid transaction 470
 - avoiding association
 - table 473
 - avoiding full table scans 474
 - caching SQL statement 468
 - choosing the right fetch
 - type 470
 - defer database updates 471
 - dividing tables 466
 - inheritance mapping
 - strategy 467
 - merge tables 465
 - ordering a collection 476
 - query performance 474
 - redesign schema 464
 - reducing database
 - operations 470
 - reducing database round trips 477
 - tuning the connection
 - pool 468
 - tuning the JDBC Layer 468
 - using bulk update 473
 - using cascade on
 - relationship 472
 - using database indexes 475
 - using DELETE
 - CASCADE 472
 - using indexes for association
 - field 475
 - using lowest-cost locking 471
 - using named query 469
 - entity state
 - attached 326
 - detached 298, 300, 320
 - removed 329
 - transient 298
 - EntityContext 522
 - @EntityListeners 334

- EntityManager 18, 45, 65, 294–333, 449
 - application managed 308
 - within Java EE container 308
 - as air traffic controller 300
 - best practices 338
 - container managed 305, 307
 - creating 304
 - flush mode 331
 - flushed 315, 330
 - in a ThreadLocal variable 313
 - in ActionBazaar 303
 - injection 305
 - interface 294
 - lifecycle 451
 - methods 295
 - scope 306
 - thread safety 307
 - using from EJB 2.x 504
 - EntityManagerFactory 309, 450
 - injection 309
 - @EntityResult 381
 - EntityTransaction 311, 414
 - enum 264
 - @Enumerated 260, 264
 - enumeration 264
 - envelope. *See* SOAP
 - env-entry 153, 627
 - env-entry-type 153
 - environment entry 153
 - type. *See* env-entry-type
 - environment naming
 - context 147, 446, 594
 - equals method 233, 236
 - exception
 - application 194
 - checked 91, 165, 194
 - system 194
 - unchecked 194
 - exception handling 193
 - exchange of information 79
 - @ExcludeClassInterceptors 161
 - @ExcludeDefaultInterceptors 161
 - @ExcludeDefaultListeners 337, 420
 - @ExcludeSuperClassListeners 337
 - executeUpdate method 379
 - extend 78
 - extended persistence scope 306
- F**
-
- façade 582
 - failover 14, 489
 - fault-tolerant 8
 - fetch 242, 323
 - FetchType 321, 325
 - field access 230, 258
 - field-based persistence. *See* field access
 - finalize method 124
 - find method 299, 304, 319
 - with composite primary keys 319
 - FinderException 508, 527
 - fine-grained 435–436
 - control 308
 - FlushMode 353, 471
 - setting on a per query basis 471
 - FlushModeType 331, 353, 471
 - Flux 167
 - foreign key 276, 279, 475
 - Fowler, Martin 219
 - Fragile 120
 - framework 7
 - FULL TABLE SCAN 475
- G**
-
- @GeneratedValue 64, 268
 - GenerationType 64
 - generator-value 642
 - Geronimo 13
 - getBusinessObject 145
 - getCallerPrincipal 211
 - getContextData method 164
 - getEJBHome 144–145
 - getEJBLocalHome 145
 - getEJBLocalObject 145
 - getEJBObject 145
 - getInvokedBusinessInterface 145
 - getMessageContext 145
 - getParameters method 163
 - getPrimaryKey method 522
 - getResultList method 351
 - getRollbackOnly 144, 192
 - getSingleResult method 350
 - getStatus method 200
 - getTarget method 163
 - getter 227
 - method 149, 229
 - getTimeRemaining method 173
 - getTimerService 144
 - getTransaction method 312
 - getUserTransaction 144
 - method 198
 - getXX method. *See* getter, method
 - GlassFish 19, 56, 87, 647
 - global JNDI name 443
 - GoldBidManagerBean 104
 - group 203
 - system 203
- H**
-
- @HandlerChain 561
 - HAR 391
 - HelloUser EJB 27
 - HelloUserExample 24
 - HelloUserBean 37
 - helper class 401, 442, 444
 - Hibernate 14, 17, 24, 28, 218, 313, 345, 500, 533, 573
 - hbm2java 433
 - Hibernate in Action 41
 - high availability 106, 488
 - high-performance 8
 - computing 14
 - holder class 559
 - home interface 508
 - HSQL 355
 - HTTP 50, 107, 489
 - HttpSession 107, 442, 446
 - object 490
- I**
-
- iBATIS 225
 - IBM MQSeries. *See* IBM WebsphereMQ
 - IBM WebsphereMQ 56, 112, 182
 - @Id 64, 224, 230, 234, 236
 - using 234

- @IdClass 235–236, 320
 - IDE 401
 - IDENTITY 315
 - identity 202, 233, 314, 319
 - column 601
 - idle time
 - for a bean instance 106
 - IllegalStateException 192, 198, 330, 351
 - immutable object
 - references 559
 - impedance mismatch 251–258
 - Indigo. *See* Windows Communication Platform
 - infrastructure 12
 - concerns 157
 - @Inheritance 284, 467
 - inheritance 12, 28, 223, 228, 256, 514
 - mapping 284–291
 - session bean 155
 - inheritance mapping 284–291
 - comparisons 290
 - joined table 287
 - single table 285
 - table-per-class 288
 - inheritance strategy
 - single table per entity hierarchy 516*Also see* inheritance mapping
 - InheritanceType 286
 - inherited column 289
 - @Init 509
 - InitialContext 49, 156, 492, 593
 - initialValue 269
 - inject 133
 - Also see* dependency injection
 - injection 86
 - stateful session bean 104*Also see* dependency injection
 - injection-target 626, 629
 - INSERT statement 263
 - instance
 - field 227
 - stateful bean 93*Also see* bean instance
 - instance variables 107, 303, 487
 - instanceof operator 371
 - instantiate 49
 - integration
 - application-to-application 538
 - business-to-business 538
 - interception 158
 - interceptor 13, 31, 82–167, 408
 - business method 158
 - class level 160
 - difference between business method and lifecycle 166
 - disable 161
 - for programmatic security 212
 - implementing 162
 - lifecycle callback 165
 - method level 160
 - interceptor-binding 407, 631
 - interceptor-class 407
 - @Interceptors 159
 - interface 76
 - based programming 76
 - business 76
 - component 23, 506
 - endpoint 539
 - home 23
 - local 87
 - remote 88
 - service endpoint(SEI) 98
 - Interface21 569
 - interlocking 219
 - interoperability 8, 539
 - issues 500
 - web services 505
 - with CORBA 505
 - intersection table 256, 366
 - Also see* association table
 - inverseJoinColumns 283
 - Inversion of Control 30
 - invocation
 - local 590
 - remote 590
 - InvocationContext 159, 162, 561
 - methods 163
 - proceed method 159
 - invoke
 - EJB 2.x from EJB 3.0 502
 - IoC. *See* Inversion of Control
 - irrecoverable 179
 - isCallerInRole 145, 211
 - method 211
 - isolation 180
 - ItemManagerBean 303
- ## J
-
- JAAS 13
 - Also see* Java Authentication and Authorization Service
 - Java Authentication and Authorization Service 13, 205
 - Java Community Process 8, 13, 75
 - Java Connector Architecture. *See* JCA
 - Java Database Connectivity. *See* JDBC
 - Java EE 13
 - platform 388
 - specification 388
 - Java EE 5 SDK
 - installing and configuring 646
 - Java Generics 247
 - Java Messaging Service. *See* JMS
 - Java Naming and Directory Interface 588, 591–595
 - Also see* JNDI
 - Java Persistence API 67
 - JPA 15
 - using from EJB 2.x 504
 - using from outside container 451–452
 - using from web tier 447–451
 - Java Persistence Query Language 17, 19, 341–379
 - Also see* JPQL
 - Java Remote Method Invocation 588–591
 - Java Remote Method Protocol. *See* Remote Method Invocation
 - Java RMI. *See* RMI
 - Java SE 5 647
 - Java Transaction API. *See* JTA
 - Java XML Binding 545
 - JavaBeans
 - naming conventions 227
 - rules 227
 - Javadoc 36
 - java.io.Serializable 78, 101, 590

- java.lang.Object 233
- java.lang.RuntimeException 91
- java.rmi.Remote 88
- java.rmi.RemoteException 25, 589
- java.sql.Blob 265
- java.sql.Clob 265
- java.sql.Connection 101
- java.sql.Time 265
- java.sql.Timestamp 265
- java.util.List 244
- java.util.Set 244
- javax.annotation.PostConstruct 39, 609
- javax.annotation.PreDestroy 39, 609
- javax.annotation.Resource 39, 607
- javax.annotation.security.DeclareRoles 39, 608
- javax.annotation.security.DenyAll 39, 609
- javax.annotation.security.PermitAll 39, 609
- javax.annotation.security.RolesAllowed 608
- javax.annotation.security.RunAs 39, 609
- javax.ejb.LocalHome 605
- javax.ejb.RemoteHome 605
- javax.ejb.ActivationConfigProperty 606
- javax.ejb.ApplicationException 608
- javax.ejb.EJB 39, 606–607
- javax.ejb.Init 605
- javax.ejb.Local 605
- javax.ejb.MessageDriven 606
- javax.ejb.NoSuchEJBException 442, 447
- javax.ejb.PostActivate 609
- javax.ejb.PrePassivate 610
- javax.ejb.Remote 605
- javax.ejb.Remove 604
- javax.ejb.SessionBean 25
- javax.ejb.Stateful 604
- javax.ejb.Stateless 604
- javax.ejb.Timeout 610
- javax.ejb.TransactionAttribute 608
- javax.ejb.Transaction-Management 607
- javax.interceptor.AroundInvoke 610
- javax.interceptor.ExcludeClassInterceptors 610
- javax.interceptor.ExcludeDefaultInterceptors 610
- javax.interceptor.Interceptors 610
- javax.jaxm.OneWayMessageListener 128
- javax.jms.Connection 117, 136
- javax.jms.ConnectionFactory 118
- javax.jms.MapMessage 121
- javax.jms.Message 61, 119
- javax.jms.MessageListener 61
- javax.jms.Queue 152
- javax.jms.QueueConnectionFactory 152
- javax.jms.Session 117
- javax.jms.TextMessage 121
- javax.jms.Topic 131, 152
- javax.jws.WebServiceRef 39
- javax.mail.Session 154
- javax.persistence.AssociationOverride 617–618
- javax.persistence.AttributeOverride 614
- javax.persistence.Basic 614
- javax.persistence.Column 613
- javax.persistence.DiscriminatorColumn 618
- javax.persistence.DiscriminatorValue 619
- javax.persistence.Embeddable 611
- javax.persistence.Embedded 611
- javax.persistence.EmbeddedId 611
- javax.persistence.Entity 611
- javax.persistence.EntityListeners 620
- javax.persistence.EntityManager 341
- javax.persistence.Enumerated 612
- javax.persistence.ExcludeDefaultListeners 621
- javax.persistence.ExcludeSuperClassListeners 620
- javax.persistence.GeneratedValue 614
- javax.persistence.Id 611
- javax.persistence.IdClass 611
- javax.persistence.Inheritance 618
- javax.persistence.JoinColumn 616
- javax.persistence.JoinTable 617
- javax.persistence.Lob 612
- javax.persistence.ManyToMany 616
- javax.persistence.ManyToOne 615
- javax.persistence.MapKey 618
- javax.persistence.NamedNativeQuery 619
- javax.persistence.NamedQueries 619
- javax.persistence.NamedQuery 619
- javax.persistence.OneToMany 616
- javax.persistence.OneToOne 615
- javax.persistence.OrderBy 618
- javax.persistence.Persistence 453
- javax.persistence.PersistenceContext 39, 622
- javax.persistence.PersistenceUnit 39, 622
- javax.persistence.PostLoad 622
- javax.persistence.PostPersist 621
- javax.persistence.PostRemove 621
- javax.persistence.PostUpdate 621
- javax.persistence.PrePersist 621
- javax.persistence.PreRemove 621
- javax.persistence.PreUpdate 621
- javax.persistence.PrimaryKeyJoinColumn 617
- javax.persistence.Query 341
- javax.persistence.SecondaryTable 613

- javax.persistence.Sequence-Generator 615
- javax.persistence.SqlResultSet-Mapping 620
- javax.persistence.Table 613
- javax.persistence.Table-Generator 614
- javax.persistence.Temporal 612
- javax.persistence.Transient 612
- javax.persistence.Unique-Constraint 613
- javax.persistence.Version 620
- javax.sql.DataSource 118, 147
- JAXM 128
- JBoss 19, 106
- JCA 122
- JDBC 13, 80, 117, 134, 225, 258, 452, 500
 - data source 146
 - driver 656
 - fetch size 477
- JDBCTemplate 225
- JDeveloper 26, 659
- JDO 14, 28, 31, 533
 - migrating to JPA 534
- JDO QL 355, 534
- Jetty 313
- JMS 13, 56, 111, 117
- jms/OrderBillingQueue 57
- jms/QueueConnectionFactory 117
- JMSCorrelationID 120
- JMSMessageID 120
- JMSTimestamp 120
- JND 439
- JNDI 27, 40, 87, 103, 118, 156
 - context 591
 - lookup 49, 155, 503
 - mapping 147
 - name 147
 - global 147
 - namespace 578
 - registry 141, 452
 - tree 87, 592

Also see Java Naming and Directory Interface
- jndi.properties 56, 593
- join table. *See* association table
- join the existing transaction
 - join existing 188
- @JoinColumn 274–276, 279–281, 283
 - many-to-one 279
 - one-to-one mapping 274
- join-column 640–641
- @JoinColumns 275
- JOINED 288
- @JoinTable 280
 - many-to-many mapping 281
- join-table 640, 642
- joinTransaction 312
 - method 311
- JPA 11
 - Also see* Java Persistence API
- JPA provider 279, 355
 - Also see* persistence provider
- JpaDaoSupport 575
- JpaTemplate 572–576
- JPQL 28, 65
 - ANY / ALL / SOME operator 375
 - arithmetic functions 368
 - ascending order 374
 - BETWEEN operator 363
 - boolean operation involving null 365
 - bulk delete 379
 - bulk operation 379
 - bulk update 378
 - cartesian product 376
 - collection value path expression 360
 - complex conditional expression 362
 - CONCAT 368
 - conditional expression 362
 - constructor in a SELECT clause 370
 - DATETIME expression 363
 - DELETE statement 357
 - DISTINCT keyword 370
 - domain 358
 - empty collection 365
 - empty string 365
 - EXISTS 375
 - FROM clause 358
 - GROUP BY 373
 - HAVING clause 367, 373
 - identifier variable 359
 - IN operator 363
 - in subquery 375
 - INNER JOIN 377
 - IS NOT NULL operator 365
 - IS NULL operator 365
 - joining entities 376
 - LEFT JOIN 378
 - LIKE operator 364
 - logical operators 362
 - MEMBER OF operator 367
 - null value 365
 - operators 362
 - ORDER BY clause 360, 374
 - outer join 377
 - parameter type 362
 - parent query 363
 - path expression 360
 - pattern_value 364
 - relational operators 362
 - relationship join 377
 - reserved identifier 359
 - SELECT clause 370
 - single-value path expression 360
 - string functions 367
 - subquery 363, 374
 - SUBSTRING 368
 - temporal functions 369
 - theta join 377
 - UPDATE statement 357
 - using SELECT 356
 - WHERE clause 360–361
 - wildcard 364

Also see Java Persistence Query Language
- JPQL query 524
- JSF 19
 - backing bean 48, 435
 - managed bean. *See* backing bean
- JSON 429
- JSP 47
- JTA 13, 184
- JUnit 29, 54
- JVM 55, 106, 394
 - Sun 395

K

KabadiBazaar 503, 540
Kodo 17

L

layer
 business logic 9
 persistence 9
 presentation 9
layered architecture 9
lazy fetching. *See* lazy loading
lazy loading 265–266, 320, 324, 378, 470
LDAP. *See* Lightweight Directory Access Protocol
lifecycle callback 82
 creation 80
 destruction 80
 in interceptor class 165
 listener 165
 PostActivate 94
 postconstruct 91
 predestroy 91, 100
 prepassivate 94
 stateless session bean 90
Lightweight Directory Access Protocol 205, 591
Linux 647
listener 123
load balancing 14, 489
 stateless session bean request 490
loader. *See* class loader
loading behavior 324
@Lob 260, 265
@Local 87
local 37
local interface. *See* interface
@LocalHome 513
local-home 504
lock method 462
locking strategy 458
 optimistic 459
 pessimistic 458
LockModeType 462
Log4J 159
login-config 206
lonely castaway 115
lookup method
 EJBcontext 144, 155
loosely coupled
 application 539
 interface 539

M

mail session. *See* javax.mail.Session
managed class 439
managed objects 141
Manifest 55
Manifest Class-Path 396
@ManyToMany 224, 247, 323, 640
 definition 247
 many-to-many 224, 247, 256, 281, 640
 relationship mapping 282
@ManyToOne 244, 278
 definition 246
 many-to-one 223, 244, 638
 self-referencing relationship 281
@MapKey 640
map-key 640
mappedBy 276, 279
 in @OneToMany 246
mappedName
 Stateless annotation 87
mapped-superclass 636
mapping metadata 418
 orm.xml 410
 reference 634–645
 using annotation 418
mapping-file element 418
marshaling 590
Maven 401
MDB. *See* Message Driven Bean
merge method 327
merging relation 328
message 56
 acknowledgement 130
 dead 138
 envelope 119
 header 120
 in a bottle 115
 JMS. *See* javax.jms.Message
 model
 choosing 137
 payload 121
 poisoning. *See* poison message
 property 120
 data type 120

redelivery 138
selector 131
sending 113
store 113
type 121
 choosing 137
 usage pattern 121
 version 120
message consumers 112
 multi-threaded 123
message destination 115
 Also see destination
Message Driven Bean 56–61, 111–139
 life-cycle callback 132
 performance tuning 487
 pooling 488
 programming rules 124
 tips and tricks 137
 why use 122
message endpoint 142
message oriented
 middleware 112
 Also see MOM
message producer 111
 JMS 117
message provider
 tuning 487
MessageConsumer 131
@MessageDriven 61, 125–126, 513
MessageDrivenBean
 interface 513
MessageDrivenContext 125, 144–145, 152, 198
MessageListener 111, 125
messageListenerInterface 128
MessageProducer 59, 117, 119
messaging
 concepts 111, 114
 destination 57
 model 114, 116
Metadata 25
metadata annotation 35–36
 common 208
 EJB 3.0 46
 metadata-complete 634, 645
method invocation 50, 80
method local variable 303

method-permission 631
 Microsoft Message Queue 117
 migration
 agile approach 535
 all-at-once 535
 approaches 535–536
 bean managed
 persistence 514
 client application of session beans 511
 CMP entity beans to JPA 513–528
 component interfaces 507
 container managed relationships 524
 converting interfaces 519
 DAO implementation Class to Use the EntityManager 530
 entity bean classes and interfaces 519
 entity bean client 526
 entity bean life cycle Methods 521
 finder and select methods 523
 finding an alternative to EntityContext 522
 home as a factory interface 526
 home interface to a session façade 527
 JDBC DAO to Use JPA 529
 local interface for creating entity class 520
 manual vs automated 536
 O-R frameworks to Use JPA 533–535
 O-R mapping metadata 519
 phased approach 535
 primary key 521
 proprietary mapping metadata to JPA 534
 redesigning your domain model 514
 resource usage 510
 session bean 505–513
 session bean class 508
 SQL Queries for JPQL 531 to EJB 3 500–536

transaction and security setting 511
 why to migrate to EJB 3 505
 model-view-controller 427
 modifier 35
 MOM 56–112
 Also see message oriented, middleware
 multiple tier 463
 multiplicity 222
 multi-threaded 442, 446
 multithreading 123, 137, 442, 446
 MVC. *See* model-view-controller
 MySQL 380, 656

N

name
 Stateless annotation 87
 named native query 382, 635
 named query 469, 634
 @NamedQueries 345
 @NamedQuery 344, 533
 naming service 591
 NamingException 508
 native SQL. *See* SQL
 natural key 267, 599
 .NET 47
 .Net platform 538
 NetBeans 659
 Network Information Service 591
 NoClassDefFoundException 424
 NonDurable 131
 non-durable subscription 131
 non-jta-data-source 416
 non-managed class 439, 442
 non-repeatable read 457
 NonUniqueResultException 350
 NoResultException 350
 normal form 602
 Boyce-Codd 602
 JPQL
 IS 366
 NotSupportedException 199
 nouns 18, 223
 NullPointerException 196

O

O/R. *See* Object Relational Mapping
 object 219
 Object Relational Mapping 17, 224
 introduction 257
 with JPA 251–291
 ObjectMessage 117, 119, 121
 ocean 115
 OHare Logistics 114
 ojdbc14.jar 656
 @OneToMany 224, 244–246, 278, 323, 525
 definition 245
 one-to-many 223, 244, 639
 mapping 278
 @OneToOne 240–246, 472, 525
 bidirectional 243
 definition 241
 one-to-one 223, 639
 bidirectional 243
 mapping 273
 unidirectional 240
 @OneWay 561
 onMessage method 61, 126
 open source 12
 OptimisticLockingException 460, 462–463
 optionality 222–223, 243
 Oracle 106, 182, 380
 Oracle Advanced Queuing 112
 Oracle Application Server 469
 Oracle Application Server 10g 19
 Oracle Internet Directory 205
 Oracle TopLink 320
 Oracle XE 656
 order billing 57
 order entity 44
 OrderBillingMDB 44, 59–61
 @OrderBy 476, 640
 order-by 640
 OrderManagerBean 185
 ordinality 264
 Also see optionality
 ORM 45, 573
 Also see Object Relational Mapping
 orm.xml 447

P

- packaging
 - application 389
 - EJB 2.x and EJB 3 501
 - sessionbeans and MDB 400
- packaging standard 389
- PAR. *See* persistence archive
- parameter 349
 - binding 469
 - named 350
 - positional 349
- pass-by-value 590
- passivation 74, 80, 93, 100
- password 202
- pattern
 - design 9
- PeopleSoft 182
- performance
 - concerns 456
 - consideration 484
 - tuning 463–464
 - tuning of EJB 3 Component 483–488
- @PermitAll 209–210
- persist
 - cascade 317
 - entity relation 316
 - method 66, 304, 314–319
- Persistence 312
- persistence archive 391
- persistence context 301–303
 - extended 438, 479
 - Also see* PersistenceContext
 - extended scope 301
 - transaction scoped 301
 - using it in caching entity 480
- persistence engine. *See* persistence provider
- persistence module 391
 - without persistence.xml 411
- persistence operation
 - propagate 318
- persistence provider 15, 20, 231, 299, 353, 388, 414, 452, 462
- persistence unit 305, 452
 - ActionBazaar 306
 - defined at the EAR level 413
 - local 413
 - name 412
 - scoping 413
- @PersistenceContext 66, 304–305, 343, 379, 449
 - definition 305
- PersistenceContext 306
 - extended 102
- persistence-context-ref 449, 504, 628
- PersistenceContextType 306
- PersistenceException 315, 462
- @PersistenceUnit 309, 312, 450
- PersistenceUnit 482–483
 - persistence-unit 413, 633
 - sub elements 633
- persistence-unit-defaults 420, 634
- persistence-unit-metadata 634
- persistence-unit-ref 629
- persistence.xml 312, 447, 452, 579, 624
 - reference 632–633
- persistent field 232
- phantom read 457
- PlaceBidBean 77
- PlaceBidServlet 47, 62
- PlaceOrderBean 44, 51–54
- placeSnagItOrder method 187
- playBoyTopic 131
- plugin principle 572
- point cuts 158
- point-to-point 114
- poison message 138
- POJI 25–26, 46
- POJO 4, 25–26, 46, 228
- polymorphic
 - association 291
 - operation 371
 - query 229, 371
- polymorphism 12, 28, 229, 291, 514
- pooling
 - bean 92
 - managed 133
 - MDB 123
 - stateless bean 83
- PortableRemoteObject 49, 511
- portType 545
- @PostActivate 82
- post-activate 630
- postage and postmark 120
- @PostConstruct 81, 488
 - message driven bean 125
- PostConstruct
 - message driven bean 134
- post-construct 630
- PostgreSQL 656
- @PostRemove 336
- @PreDestroy 81, 92, 488
 - message driven bean 125
- PreDestroy
 - message driven bean 134
- pre-destroy 630
- PreDestroy method
 - message driven bean 124
- premium sellers 115
- pre-passivate 630
- @PrePersist 334
- presentation tier 427
- @PreUpdate 334
- primary key 64, 234, 277, 314, 599
 - AUTO generation 270
 - composite 235
 - generation 65, 267
 - using database sequence 269
 - using identity constraint 268
- @PrimaryKeyJoinColumn 277
 - inheritance mapping 288
 - one-to-one relation 276
 - secondary table 267
- primary-key-join-column 642
- @PrimaryKeyJoinColumns 278
- primitives 94
- Principal 205
 - getName method 212
- proceed method 165
- producer. *See* message producer
- profiling 157
- property
 - access 230
- property based persistence. *See* property access
- propertyName 132
- propertyValue 132
- proxy 141
- PTP. *See* point-to-point
- public key 207

publish-subscribe 114
 Pub-Sub. *See* publish-subscribe

Q

Quartz 167
 query
 anatomy 343
 comparing API with
 JDBC 342
 dynamic 343, 347
 flush mode 352
 hint 353
 introducing API 342
 JPQL 341, 383
 named 343–347
 pagination property 352
 SQL 341, 383
 timeout 353
 Query API 341
 Query object 343
 Query Parser 355
 Query Processor. *See* Query
 Parser
 @QueryHint 354, 477
 queryHint 382
 queue 115

R

RAR. *See* resource adapter
 archive
 RDBMS. *See* relational database
 management system
 read-only 263
 realm 207
 reference implementation 647
 referencedColumnName 275,
 277
 refresh method 304, 331
 relational database manage-
 ment system 9
 relational paradigm 251
 relations
 polymorphic 291
 relationship 219, 221, 239
 bi-directional 223
 container managed 526
 inverse side of 280
 mapping 273–283
 owner 244, 280
 owning side 283
 self-referential 227
 subordinate 283
 uni-directional 223
 reliability 113
 @Remote 46, 54, 88, 590
 remote
 communication 588
 interface 588
 Remote Method Invocation 490
 Also see RMI
 remoteability 297
 RemoteException 194
 @RemoteHome 513
 remoting 13
 @Remove 54, 94, 98, 101, 107,
 442, 486
 remove method 300, 304, 329
 REpresentational State
 Transfer 542
 request-reply 116
 @Resource 58, 86, 98, 117–
 118, 125, 134, 146
 resource
 binding 148
 reference 148
 resource adapter archive 390
 resource injection 92
 Also see dependency injection
 resource manager 182
 resource-env-ref 627
 resource-ref 147, 626
 resources
 managed 80
 res-ref-name 147
 result list 351
 offset of 352
 pagination 351
 retrieve 304
 RMI 13, 55, 74, 88, 103, 111
 client 589
 registry 588, 591
 server 589
 Also see Java Remote Method
 Invocation
 RMI-IIOP 505, 590
 @RolesAllowed 209
 rollback 177, 179
 RollbackException 189
 @RunAs 210
 RuntimeException 124, 194

S

sandbox 179
 scalability 8, 488
 scalar value 381
 scheduled tasks 167
 scheduler 167
 scheduling 13
 schema 597
 @SecondaryTable 266–267
 security 74, 202–213
 bean managed 144
 concepts 202
 declarative 208, 297
 in Java EE 205
 programmatic 210
 web tier 206
 security-constraint 206
 SecurityException 211, 213
 SecurityInterceptor 213
 security-role 630
 security-role-ref 630
 sender 114
 Also see message producer
 separation of concerns 402,
 570, 572
 sequence 270, 601
 sequence generator 268, 635
 @SequenceGenerator 268
 Serializable 119, 238
 Also see java.io.Serializable
 serialization 487
 serialized 99, 298, 326
 server farm 14
 service 21, 539
 service description 539, 542
 service discovery 542
 service layer 12
 service locator 444
 service messaging 542
 service oriented
 architecture 538
 service provider interface 591
 service registry 588
 service transport 542
 service-oriented
 application 539
 Service-Oriented Architecture 8
 service-ref 628
 servlet 19, 47, 104
 controller 441

- session 50, 79
 - closing 119
 - commit 119
 - HTTP 50
 - JMS 119
 - management 74
 - replication 106
 - transactional 119
- session bean 15–16
 - alternative 75
 - anatomy 76
 - best practices 108
 - copy-by-value semantic 484
 - home interface 504
 - interface 76
 - life cycle 79
 - performance tuning 484
 - programming rules 78
 - unit testing 77
 - why use 73
- session bean client 102
 - remote client 102
 - using dependency injection 104
- session façade 427, 434–439, 446
 - stateful 438
- session replication
 - multicasting 493
 - peer-to-peer 493
- SessionBean interface 506
- SessionContext 144, 152, 198
- SessionSynchronization 102
- session-type 37
- setEntityContext method 522
- setFirstResult method 352
- setFlushMode method 331, 353
- setHint method 353–354
- setMaxResults method 352
- setParameter 350
- setParameters method 163
- setRollbackOnly 144
 - method 186, 192
- setter 227
 - injection 149
 - method 149
- setTransactionTimeout
 - method 200
- setXX method. *See* setter, method
- shareable 151
- shared-library 396
- shipping request 113, 117
 - table 124
- ShippingRequestProcessorMDB 125
- ShippingRequestQueue 117–118
- Simple Object Access Protocol 542
- SingleThreadModel 449
- Snag-It Ordering 178
 - in BMT 196
 - using CMT 185
- SOA. *See* Service-Oriented Architecture
- SOAP 74
 - body 543
 - envelope 541
 - header 543, 560
 - message 543
 - Also see* Simple Object Access Protocol
- @SOAPBinding 553, 555, 566
- SoC. *See* separation of concerns
- SonicMQ 56, 112
- Spring 41, 75
 - AbstractJMSMessageDrivenBean 581
 - AbstractStatefulSessionBean 581
 - AbstractStatelessSessionBean 581
 - bean factory 581–583
 - BeanFactoryPath 583
 - combining power with EJB 3 581–585
 - configuration 582
 - configuring
 - to use the EntityManager 578–579
 - to use the JPA 577
 - DataAccessException 575
 - developing Spring-enabled EJB 581–583
 - EAO 574
 - enabled stateless session bean 582
 - JDBCTemplate 30
 - JMSTemplate 30
 - JndiObjectFactoryBean 578, 584
- JpaDaoSupport 574, 580
- JpaTemplate 434, 574
- LocalContainerEntityManagerFactoryBean 579
- MVC 428
- onEjbCreate method 581
- Pitchfork project 30, 573
- support class to integrate EJB 581
- using JPA 573
 - outside container 579
 - wiring 584
- wiring Entity Access Objects 580
 - with EJB 3 30–31, 568–586
- SQL 19, 67, 225, 365
 - dynamic query 381
 - generation 258
 - named query. *See* named native query
 - query plan 476
 - recursive join 380
 - START WITH ...
 - CONNECT BY 380
 - Also see* structured query language
- SQL Server 9
- SQL-92 601
- @SqlResultSetMapping 381
- sql-result-set-mapping 635
- SSL 207
- stale data 478
- state 222
- State replication. *See* session replication
- state transition 79
- @Stateful 53, 98
- stateful session bean 16, 50–54, 93–108
 - alternative 107
 - caching 486
 - instance variable 106
 - passivation policy 106, 486
 - performance consideration 105
- @Stateless 46, 78, 86
- stateless object 105
- stateless session bean 16, 45–47, 83–93
 - pool 485

static 78
 Status interface 200
 StreamMessage 121
 structured query language 601
 Struts 428, 449
 stub 491
 client 492
 subscriber 115
 subscriptionDurability 130
 Sun Glassfish. *See* Glassfish
 superclass 78
 surrogate key 238, 268, 599
 Swing 451
 SWT 451
 synchronize 297, 316
 synchronous 111
 communication 114
 system administrator 388
 System Task Scheduler 167
 system universe 219

T

@Table 64, 224, 261
 table generator 269, 636
 @TableGenerator 269
 Tapestry 428
 targetEntity 242
 use in @OneToMany 246
 @Temporal 260, 265
 TestNG 29
 thread safe 448
 ThreadLocal pattern 313
 ThreadLocal session pattern.
 See ThreadLocal pattern
 thread-safety 73
 Tibco Rendezvous 112
 tier
 business logic 45
 @Timeout 170
 timeout method 168
 implementing 171
 timer 167–175
 cancel 173
 creation 169
 how it works 168
 interval. *See* timer, recurring
 limitations 174
 merits of 174
 recurring 171

single-event 171
 when it make sense 174
 timer interface 170
 using 172
 timer service 74, 144, 154, 167
 accessing 170
 using it 169
 TimerHandle 173
 TimerService 154
 TIMESTAMP 265
 Tomcat 19, 313
 topic 115
 TopicConnectionFactory 152
 TopLink 14, 17, 28, 218, 500, 533, 573
 TopLink Essentials 270, 414
 torn-down 124
 transaction 74
 all or nothing 177
 value proposition 178
 asking container rollback 191
 atomic 179
 attribute 188
 boundary 184
 consistent 180
 container managed 137, 144
 create a new 189
 declarative 185
 defined 177
 distributed 183
 exception handling 193
 isolation level 458
 joining 187
 JTA 311, 448
 local 182
 phase 183
 programmatic 450
 requires existing 190
 resource local 311
 suspended 189
 timeout 201
 uncommitted 457
 with MDB 191
 transaction management
 in EJB 184
 internals 181
 transaction manager 180, 182
 distributed 184
 transactional system 459

@TransactionAttribute 186–187, 485
 using 187
 TransactionAttributeType 186, 188
 MANDATORY 190
 NEVER 190
 NOT_SUPPORTED 189
 REQUIRED 188
 REQUIRES_NEW 189
 SUPPORTS 190
 @TransactionManagement 186, 197
 using 186
 TransactionManagementType 186
 BEAN 187, 197
 CONTAINER 186
 TransactionRequiredException 316, 330
 transaction-type 186, 312, 414
 JTA 414
 RESOURCE_LOCAL 312, 414
 transfer object 430
 @Transient 231
 transient 299
 field 229
 modifier 232, 487
 object 487
 tuning. *See* performance, tuning
 Turtle server 113
 Turtle Shipping Company 113

U

UML 219
 uncommitted change 457
 @UniqueConstraint 262
 uniqueConstraints 262
 unmarshaling 590
 unSetEntityContext
 method 522
 UPDATE statement 263
 user 203
 USER_PICTURES table 260
 UserAccountRegistrationBean 105
 username 202

UserTransaction 184, 196
 getting 198
 getting from EJBContext 198
 JNDI lookup 198
 methods 199

V

verbs 18, 223
@Version 461

W

WAR 37, 447
weaving 579
web
 container 308
 framework 429
 module 447
web application archive 390
 Also see WAR
web service 8, 16, 47, 74, 103,
 429, 538
 access from EJB 562
 basic profile 540
 benefits 539
 best practices 565
 bottom up approach to
 development 548
 components 541

 consumer 547
 definition 539
 developing EJB web
 service 551–565
 document oriented 547
 document style 545
 EJB 3 vs. regular java
 class 551
 encoded 547
 handler 561
 literal 547
 meet in the middle approach
 to development 549
 messaging style 547
 metadata 552
 popular style/messaging
 548
 protocol 546
 REST. *See* REpresentational
 State Transfer
 RPC style 547
 style 547
 top down approach to
 development 548
 why choose EJB over
 POJO 550
Web Services Description
 Language 539
Web Services-Interoperability
 (WS-I) Organization 540

web tier 205, 427
WebLogic. *See* BEA WebLogic
@WebMethod 553, 560
 definition 557
@WebParam 553, 558
 definition 559
@WebResult 553, 560
@WebService 47, 551–552
 definition 554
@WebServiceRef 564
webservices.xml 552
WebSphere 106
WebSphereMQ. *See* IBM Web-
 sphereMQ
WebSphere Studio 26
Webwork 428
Webworks in Action 41
web.xml 206, 441
Windows 647
Windows Communication
 Foundation 539
workflow 93
WSDL 539

X

XDoclet 36
XML 12
XML schema 548
XML-RPC 542