INDEX

Note: Page numbers in **bold** refer to definition of the term.

A
‘Abort ACK’ message, 329–330
‘Aborted’ transaction state, 302
Access control, 84, **106**
Access frequency, **78**
Access path, 120, 133, **173**
Access rights, 84–86, **106**
Access rule, 209
Acknowledgment, message, 591–593
Active transaction, **351**
‘Active’ transaction state, 302
Administered objects, 583–584, **617**
ADO.NET database connectivity, 682
Advanced Encryption Standard (AES), 361, **377**
Affinity Matrix, **78**
Affinity, **78**
After trigger, **106**
Aggressive TO algorithm, **238**
American National Standards Institute (ANSI), 18
Anomalies, 191
Apache Geronimo, 635
Applet, **637**
Application Client Container, 625, **660**
Application Client, 625, **637**
Application Domains, 667, 672–673, **684**
Application processor (AP), 9, 20
Application Sever, 624, **637, 661**

Application-defined protocol, 542, **574, 587–588, 647**
Application—programming interface (API), 581, 621, 625, 667
Arbitrated digital signatures, 365, **378**
Archetypical DBE architectures, 6–13
application processor (AP), 9
basic services, 7–8
cmd command processor (CP), 9
data accessor (DA), 9
data getter (DG), 7
data read service (Drd-S), 7
data write service (Dwr-S), 7
expected services, 8–9
expected subsystems, 9–10
query processor (QP), 9
required services, 6–7
summary level diagrams, 11–13
typical DBMS services, 10–11
Architectural concepts, DBE, 4–6
components, 4
services, 4
subsystems, 4
Architectural security issues, 375–376
Architectural security requirements, 567–569, **574**
arbitrary architectural security, 568
Architectural security requirements (contd.)
  authentication, 568
  authorization, 568
  encryption, 568–569
  identity, 567
  security policies, 567
  security target, 567
Architectural vocabulary, DDBE, 537–544
  architectural visualizations, 538–542
  business services tier, 538–540
  components, 537
  Data Services Tier, 540–542
  middleware, 537, 542–544
  service consumers, 537–538
  service providers, 537–538
  service-oriented architecture (SOA), 538, 540–542
    descriptive requests, 541
    prescriptive requests, 541
  subsystems, 537
  three-tier model (3TM), 538–540
  user presentation tier, 538–540
  User Services Tier, 538–540
Assemblies, 666, 671–672, 684, 708
  deployment alternatives, 677–678
Assertions, 106
  Asynchronous messaging, 588–589, 617
  Asynchronous replication control, 276, 278–279, 292
Atomic AT, 397, 419
  Atomicity property, 186, 238
  Atomicity, consistency, isolation, and durability (ACID), 184, 238
Attribute Type (AT), 397–398, 419
Authentication, 84–85, 106, 358, 368–369, 378, 568, 574, 617
  database, 84
  operating system, 84
Authorization, 368–369, 378, 568, 574, 617
Auxiliary database, 509
Avoidance, deadlock, 249–252, 271
  Awaits, 271
B
  Bachman diagram (BD), 437, 446
  Backup types, 311–313
    cold, 312
    complete, 311
    full-image, 312
    hot, 312
    incremental, 311–312
    rolling a database back, 313
    rolling a database forward, 313
  Backward recovery, 198, 238
  Base Class Library (BCL), 666–674, 684, 708
  Basic operators in RA, 113
  Bean Managed Transaction (BMT), 633, 637, 661
  Before commit state, 351
    'Before Commit' transaction state, 302
  Before trigger, 106
  Binary relationship types, 400
  Bit vectors, 165, 173
  Blind write, 192, 238
  Block ciphers, 359, 378
  Blocking calls, 560
  Blocking messaging, 563
  Blocking, 617
  Bloom vectors, 165, 173
  Bond Contribution Calculation, 54–62
    for (C0, C3, C1) ordering, 55
    clustered affinity matrix, 57
    partial clustered affinity matrix, 56
  Bond Energy Algorithm (BEA), 53, 78
  Bottom–up design approach, 35, 37, 78
  Bottom–up methodology, 459–461, 509
  Boyce–Codd normal form (BCNF), 446
  Broadcast approach, 283, 292
  Brute force attack, 359, 378
  Business Services Tier, 538–540, 574
C
  Callee, 557–560, 574, 637
  Caller, 557–560, 574, 637
  Candidate keys (CKs), 429
  Can-wait-for, 271
  Cardinality (of an RT), 399–400, 419
  Cartesian product (CP) operator, 116
  Cascadeless schedules, 197, 239
  Cascading rollback, 197, 239
  Catalog, 173
  Centralized 2PL, 229–230, 239
  Centralized database environment (CDBE), 13, 18, 173, 386
  Centralized DBE concurrency control, 200–222
    concurrency control in Oracle, 214–222
    growing phase, 202
    locking-based concurrency control algorithms, 201–208
    locking in relational databases, 204–206
    one-phase locking, 201
    optimistic concurrency control algorithms, 201
    pessimistic concurrency control algorithms, 201
    phantom issue, 206–208
    shrinking phase, 202
    strict two-phase locking, 203
    timestamping, 201
    two-phase locking, 201
  Centralized deadlock detectors, 262
  Centralized systems
    deadlocks in, 248–258, See also under Deadlock handling
    serializability in, 194
  Centralized systems, query processing in, 126–145
    code generation, 144–145
    dynamic programming (DP), 135–141
    query optimization, 128–144
    cost estimation, 129–133
plan generation, 133–135
query tree, 130
query parsing, 127–128
translation, 127–128
Centralized Transaction Acceptance/Rejection Phase, 281
Certificate authority, 366, 378
Certificate revocation list (CRL), 366, 378
Certification authorities (CA), 365–366
Checkpointing, 318–319, 351
consistent, 318
fuzzy, 318
Child record type, 441, 446
Child transaction, 200, 239
Cipher block chaining (CBC), 360, 378–379
Ciphers, 357, 378
Cipher text, 358, 378
Circulating token algorithm, 284–287, 292
Class, 444, 446
Class inheritance, 444, 446
Clients, 3–4, 353
client activated server object, 684, 709
CLP taxonomy, 384
conceptual-level issues, 390–394
abstraction issues, 392–393
DBE design and implementation independence issues, 390–392
emphasis issues, 393–394
Clustered affinity matrix, 53, 78
after rearranging rows and columns, 57
after rearranging the columns, 57
partial, 56
Clustered index, 173
CODASYL, 436, 509
Code generation, centralized systems, 144–145
Codes, 357, 378
Cold backup, 312, 351
Command, 3
Commercial DDBES, 483–489
InfoSphere Federation Server, 488–489
Microsoft Access, 487–488
schema integration in, 485–486
software integration in, 486–487
Commercial off-the-shelf (COTS) components, 14–15, 509, 516, 533, 574
Commit point, 352
Commit protocols, 298–300, 324–351, 352
architectural requirements, 325–326
distributed commit protocols (DCPs), 326–327
network partitioning, 348–351
one-phase commit (1PC), 326–329
quorum-based commit (QBC), 326, 348–351
three-phase commit (3PC), 326, 345–349
two-phase commit (2PC), 326, 329–345, See also individual entry
'Committed' transaction state, 302
Common Intermediate Language (CIL), 666, 669, 684
Common Language Runtime (CLR), 666–671, 684
Common Language Specification (CLS), 666–671, 684
Common Object Resource Broker Architecture (CORBA), 521, 572
Common Type System (CTS), 666, 668–669, 674, 684
reference types, 668
value types, 668
Communication link failures, 298
Communication protocol, 587
Compensating transaction, 239
Compete backup, 352
Competitive synchronization, 554, 574
Compile time validation, 97, 106
Complex predicates, 121
Components/COS (component or subsystem), 5, 537, 574, 617, 637, 661, 684, 709
Computing RA operators, 119–126
computing join, 123–126
external sort, 124
hash-join, 126
nested-loop joins with indexes, 124
nested-loop joins, 123–124
sort–merge join, 124–126
computing selection, 120–122
B+tree index, 120–122
complex predicates, 121
hash Index, 122
no index, 120
nonunique index, 121
simple predicates, 121
unique index, 121
Concepts, database, 2–4
Conceptual data model (CDM), 386, 411–413, 419, 425
Conceptual modeling language (CML), 386
Conceptual schema, 388, 392
Concurrency, controlling, 183–242, 239. See also
Centralized DBE concurrency control;
Timestamp
Concurrency control algorithms in distributed database systems, 222–237
distributed optimistic concurrency control, 236–237
distributed timestamp concurrency control, 232–236
federated/multidatabase concurrency control, 237
optimistic concurrency control algorithm, 213–214
terminology, 183–189
database consistency, 184
database, 183–184
two-phase locking in distributed systems, 229–232
Concurrent access, 3
Conditional join, 173
Confidentiality, 358, 378
Conflict Graphs, 235–236, 239
Conflict matrix, 239
Conflict serializability, 239
Conflict serializable schedules, 196, 239
Conflicting operations, 3
Conflicts, 191–192, 239
overwriting uncommitted data, 191
reading uncommitted data, 191
unrepeatable reads, 191
Connection Factories and Connections, 584
Connection Factory object, 584, 617
Connection object, 584, 617
Consensus, 292
Conservative TO algorithm, 210–212, 239
Consistency property, 186, 239
Consistent checkpointing, 318, 352
Constraints, 106
Container Managed Transaction, 633, 637, 661
Content, data, 2
ConTracts, 199
Controlling concurrency, See Concurrency, controlling
Conventional cryptography, 3, 358–362, 378
block ciphers, 359
brute force attack, 359
cipher block chaining (CBC), 360
Data Encryption Standard (DES), 360
decrypt, 358
electronic code book (ECB) mode, 360
encrypt, 358
initialization vector (IV), 360
key, 358
modes, 359
padding, 360
public key cryptography, 359, 363
RC4, 361–362
stream ciphers, 359
Cooperative DDBEs, 513–527, 533
implementation specifics, 523–527
DQSP/LQSP registration and deregistration, 523
election process, 526–527
query to DDSP, 524–526
query to DQSP, 523–524
subsystems, 515–523
client, 517
DDSP, 515, 518–519
DQSP, 515, 517
GIS, 515, 519–521
LDS, 515, 521–522
LQSP, 515, 522–523
Cooperative synchronization, 554, 575
Coordinator, 325
COS closedness or openness (COS-COO), 14–15
COS distribution and deployment (COS-DAD), 13–14
Cost-based optimization (CBO), 135, 173
Create, retrieve, update, and delete (CRUD), 3
Crow’s feet notation (CFN), 390, 407–408, 419
Cryptanalysis, 357, 378
Cryptography, 357–366, 378
arbitrated digital signatures, 365
authentication, 358
certification authorities, 365–366
ciphertext, 358
certification, 358
conventional cryptography, 358
digital certificates, 358, 365–366
digital signature (DS), 358, 364–365
hash message authentication code (HMAC), 362
message authentication code (MAC), 358, 362
message digest (MD), 358, 362
nonrepudiation, 358
plaintext, 358
public key cryptography, 358, 361–364
Cyclic redundancy codes (CRCs), 362

D
Daisy chaining, 283, 292
Data Accessor (DA), 8, 509
Data Content, 32
Data dictionary service provider (DDSP), 515, 518–519, 533
query to DDSP, 524–525
Data dictionary, 20, 173
Data distribution alternatives, 35–79
Data Encryption Standard (DES), 360, 378
Data Getter (DG), 7, 509
Data inference, 374–375
Data item, 437, 446
Data Model (DM), 2, 32
Data Modeling Language (ML), 32, 383–390
Data modeling overview, 383–421
CLP taxonomy, 384
conceptual-level goals, 394–396
forward engineering, 417–418
3LM taxonomy, 388–389
multiple models, conversion to, 417
reverse engineering, 418
using multiple types of modeling, 416–418
Data models (DMs), working with, 383–390, 411–416
CDMs, 411–413
capture the requirements, 411–412
group concepts together, 412
identify concepts in the requirements, 412
remove ambiguities, 412–413
steps to create, 411
Data processor (DP), 20
Data Read Service (Drd-S), 7, 32
Data Schema, 32
Data Services Tier, 538–540, 575
Data shipping, 152, 173
Data Source Name (DSN), 509
Data type constraints, in semantic integrity control, 88, 106
Data Write Service (Dwr-S), 7, 32
Database (DB), 1, 32, 239
Database Administrator (DBA), 83, 106
Database backups, 311
Database buffers, 309–311
Database Client (DB-Client), 32
Database consistency, 239
Database control, 83–107, See also Semantic integrity control
Database Environment (DBE), 4, 32, 510
Database management system (DBMS), 3, 32
Database recovery, 314–319, 352
Database rollback, 352
Database Server (DB-Server), 3, 32
Database Update Modes, 302
Deferred update mode, 302
Immediate update mode, 302
Deadlock Handling Service (Dead-S), 10, 32, 247–272
in centralized systems, 248–258
deadlock avoidance, 249–252
deadlock prevention, 248–249
detection, 252
preacquisition of all locks algorithm, 248
removal, 252
wait–die, 249–250
wound–wait, 249, 251
definition, 247–248
Deadlocks in distributed systems, 252–270
distributed deadlock avoidance, 255–260
distributed deadlock detection, 260–270
distributed deadlock prevention, 254–255
distributed wait–die algorithm, 254
transaction control issue, 253–254
transaction location issue, 253
Deadly embrace, 271
Decision support system (DSS), 189, 239
Decryption, 358, 378
Deferred update mode, 302
log records, 304–306
Deferred update, 313–314, 352
Degree (of a relation), 427, 447
Degree (of an RT), 400, 419
Degrees of integration, 464–465
Dem subsystem, 650–651, 706
Deployment descriptor, 637, 661
Deployment, 548–551, 575, 618, 637, 661, 684, 709
Deregistration, 523, 567, 575
Derived horizontal fragmentation (DHF), 42, 44–47, 78
Derived operators in RA, 113
Design alternatives, 38–39
distributed data, 38–39
localized data, 38
Destination object, 584–586, 618
Detection, deadlock, 252, 271
Diffie–Hellman algorithm, 363–364, 378
Digital certificates, 358, 365–366, 378, 685
Digital signature (DS), 358, 364–365, 378, 685
Directory Service Requirements, 564–567, 575, 618, 637, 681, 685
J2EE, 630–631
lookup service, 565–567
naming, 564–565
registration, 565–566
Dirty read, 239
Disjoint participation, 399, 420
Disk failures, 298
Disk I/O, 173
Distributed 2PL, 230, 239
Distributed commit protocols (DCPs), 326–327
Distributed Database (DDB), definition, 1, 32
Distributed data platform requirements, 575, 618
Distributed data, design alternatives, 38–39
fragmented or partitioned, 39
horizontal fragmentation, 39
hybrid fragmentation, 39
vertical fragmentation, 39
fully replicated, 38–39
mixed distribution, 39
nonfragmented, 38
nonreplicated, 38
partially replicated, 39
Distributed deadlock avoidance, 255–260
distributed wound–wait algorithm, 256–257
distributed wound–wait control alternatives, 257–260
Distributed deadlock detection, 260–270
centralized, 262
detectors, 262–270
graph sending rule, 267
hierarchical, 262
potential for global deadlock, 264, 266
Distributed deadlock prevention, 254–255
Distributed execution manager (DEM), 21
Distributed fund transfer transaction, 24
Distributed optimistic concurrency control, 236–237
Distributed process platform requirements, 575, 618
Distributed query optimization, 150–170, See also under Query optimization
Distributed query service provider (DQSP), 515, 517, 533
query to, 523–526
registration and deregistration, 523
Distributed systems serializability in, 195–196
splitting in, 50
two-phase locking in, 229–232
centralized 2PL, 229–230
Distributed systems (contd.)
  distributed 2PL, 230
  primary copy 2PL, 230
Distributed tables, 36
Distributed timestamp concurrency control, 232–236
  conflict graphs, 235–236
  transaction classes, 235–236
Distributed transaction, 253, 271
Distributed voting algorithm, 282–283
Distributed wait–die, 254, 271
Distributed wait-for-graph, 271
Distributed wound–wait, 271
Distribution transparency, 68–69, 78
  fragmentation, 68
  location, 68
  replication, 69
Divide operator in RA, 118
DNET starter kit, 689–710
  architectural overview, 691–697
    Abstract Factory, 692
    component identification, 695–696
    extension architectural overview, 697–700
    Proxy, 693
    subsystem interaction overview, 696–697
Dbms Subsystem, 698
Dem subsystem, 699
  design overview, 700–707
    framework exceptions area, 701
    framework interfaces area, 700–701
    framework parameters area, 701–702
  IDet, 702
  IParsedSql, 702
  IQualifiedSql, 702
  IQueryResults, 702
extension design overview, 702–707
Dbms subsystem, 703–704
Dem subsystem, 706
  Lem subsystem, 706–707
  Parser subsystem, 704–705
  Planner subsystem, 705–706
  Qualifier subsystem, 705
  goals, 691
  Lem subsystem, 699–700
  Parser subsystem, 698
  Planner subsystem, 699
  Qualifier subsystem, 699
Domains (in RDM), 426, 447
  'Done' message, 329
Dummy member, 439, 447
Durability property, 187, 239
Durable Subscriber, 591, 593–595, 598, 605, 612, 618
Dynamic programming (DP), 135–141, 173
  in distributed systems, 152–156
    1-relation plans, 153
    2-relation plans, 154–155
    3-relation plans, 155
    4-relation plans, 156
  iterative dynamic programming (IDP), 142–143
  subsequent steps, 136
E
  Electronic code book (ECB) mode, 360, 378
  Encrypted file systems (EFSs), 369, 378
  Encryption, 358, 369–370, 379, 568–569, 575
  Enterprise Archive (EAR) files, 625–626, 661
  Enterprise Java Beans (EJBs), 625, 643–644, 661
  Entity Relationship Diagram (ERD), 420
  Entity relationship model (ERM), 398–411, 420
    attribute type (AT), 396–397
    construct review, 396–402
  Crow’s foot notation or crow’s feet notation
    (CFN), 407–408
  disjoint participation, 399
  entity type (ET), 396, 398
  inheritance type (IT), 397–399
  mutually exclusive participation, 399
  overlapping participation, 399
  relationship type (RT), 396, 399–400
  rules and guidelines, 402–405
  naming convention guidelines, 403
  naming rules, 402
  traditional Chen ERD notation, 405–407
  unified modeling language (UML), 408–411
Entity Type (ET), 398, 420
Entropy, 379
Environments, DB, 3–4
Equi-join, 173
Equivalence, 192–193
Exclusive lock mode (ELM), 216, 239
Exclusive row-level lock, 217
Execution manager (EM), 144
Execution Optimization Service (Eopt-S), 8, 32
Execution Service (Exec-S), 8, 32
Exhaustive Search Optimization, 134
Explicit constraints, in semantic integrity control, 88
Export schema, 510
Extensible Markup Language (XML), 638, 661
Extent, 447
External port, 271
F
  'Failed' transaction state, 302
Failures, 297–324
  commit point, 299
  database update modes, 302
  DBMS storage types, 303–304
    nonvolatile storage (disk, tape), 304
    stable storage, 304
    volatile storage (memory), 303
  hard failure, 298
  communication link failures, 298
  disk failures, 298
INDEX 717

system failures, 298
transaction failures, 298
log contents, 304–308
soft failure, 297–298
system crash, 297
termology, 297–308
transaction log, 302–303
Transaction Roll Forward (Redo), 300
Transaction Rollback (Undo), 299–300
transaction states, 300–302
UNDO/REDO and database recovery, 308–313

Fallback and Recovery Service (Rec-S), 10, 32
Federated databases (FDB), 239, 510
Federated/Multidatabase Concurrency Control, 237
Field, 447
Firewalls, 369, 379
First normal form (1NF), 436, 447
Followers, 325
Foreign key (FK), 429, 447
Forward engineering (FE), 387, 417–418, 425
ETs, 433–434
of ITs, 434–435
of RTs, 433
rules for, 429–435
forward generation of ATs, 429–433
Forward recovery, 198, 239
Fragment, 79
Fragmentation Directory (FD), 20
Fragmentation transparency, 68, 79
Fragmentation, design alternative, 39–68, See also
   Horizontal fragmentation (HF); Hybrid
   fragmentation (HyF); Vertical fragmentation
   (VF)
   replication, 68
Framework, 546–547, 575, 606–613, 648–649,
   700–702
Free or open-source software (FOSS), 535–536, 575
Full-image backup, 312
Fully replicated database, 292
Fully replicated, design alternative, 38–39
Functional Dependency (FD), 436, 447
Fundamental Platform Requirements, 544–555, 618
Fuzzy checkpointing, 318, 352

G
Generalized Voting Algorithm for Replication
   Control, 287–290, 292
   'Global Abort' message, 329–330
Global Assembly Cache (GAC), 667, 672, 685, 709
Global conceptual model (GCM), 35, 79
Global conceptual schema (GCS), 19–20
Global data dictionary (GDD), 20–21, 79, 146, 510
   data directory (DD), 20
   fragmentation directory (FD), 20
   global conceptual schema (GCS), 20
   network directory (ND), 20
   replication directory (RD), 20
Global interface schema (GIS), 515, 519–521, 533
   P2P DDBEs, 529
Global query, 173
Global timestamp, 271
Global Transaction Manager (GTM), 21, 510
Global transaction, 198, 239, 253, 271, 352
Global wait-forgraph (GWFG), 262, 271
Globally unique identifiers (GUIDs), 520–521, 533
Graph Sending Rule, 267
Greedy algorithm, 143, 173
Group, 437, 447
Grouping AT, 397, 420
Grouping, in vertical fragmentation, 49
Growing phase, 240

H
Hard failure, 298, 352
Hash index on R, 122
Hash-join, 126, 173
Hash message authentication code (HMAC), 362, 379
Heterogeneous DDBE, 174
   architecture, 170–171
   query optimization, 170–172
Heuristics-based optimization, 134, 174
Hierarchical data model (HDM), 386, 425, 440–443,
   447
   fields, 441
   forward engineering, rules for, 442
   nomenclature, 441
   parent–child record types, 441
   record types, 441
   records, 441
   reverse engineering, rules for, 442–443
   virtual link types, 441
   virtual record types, 441
Hierarchical Deadlock Detectors, 262
History, 240
Homogeneous DDBE, 174
Horizontal fragmentation (HF), 39, 42–47, 79
   completeness of, 63–66
   correctness rules, 66–67
   completeness, 66
   disjointness, 66
   reconstruction, 66
   derived horizontal fragmentation (DHF), 42,
   44–47
   design alternative, 39
   generation guidelines, 62–66
   minimality of, 63–66
   primary horizontal fragmentation (PHF), 42–43
Hot backup, 312, 352
Hybrid fragmentation (HyF), 39, 47–49, 79
Hybrid shipping, 152, 174

I
Identifying AT, 397, 420
Identifying RT, 398, 401, 420
INDEX
Identity, 575
IDet, 702
Immediate update mode, 302
log records, 306–308
Immediate update transaction steps, 314, 352
Inactive Transactions, 352
Incremental backup, 313, 352
Index page locking, 207–208, 240
InfoSphere Federation Server, 488–489
Inheritance Type (IT), 398–399, 420
Initialization vector (IV), 360, 379
Input port, 271
Integrity, 362, 379
constraints, 106
Intention lock modes, 205
Intention locking, 204–205, 240
Intentional model, 384, 420
Intermediate Language (IL), 666, 669, 674, 685
Interprocess communication (IPC), 555
Intrusion detection systems (IDSs), 369, 379
IParsedSql, 702
IQualifiedSql, 702
IQueryResults, 702
Isolation property, 187, 240
Iterative dynamic programming (IDP), 142–143, 174
greedy algorithms, 143
Iterator concept, 131

J
Java 2 Enterprise Edition (J2EE) platform, 571–572, 621–639
application clients, 625
application servers, 624
container managed transactions, 633
containers, 624
deployment descriptor, 625
directory services, J2EE, 630–631
distributed data platform requirements, 631–633
J2EE Security, 632
Java Database Connectivity (JDBC), 632
Enterprise Java Beans (EJBs), 624–625
fundamental J2EE vocabulary terms, 622–625
implementation alternatives, 633–635
Apache Geronimo, 635
Jboss, 634
JOnAS, 635
J2EE Remote-Code Execution, 627–629
J2EE remote messaging, 629–630
J2EE support for platform requirements, 626–631
J2EE transactions, 633
J2SE Development Kit (JDK), 623
Java 2 Standard Edition (J2SE) platform, 623
Java beans, 624
Java Message Service (JMS), 629–630
Java Naming and Directory Interface (JNDI), 630–631
Java Runtime Environment (JRE), 623–624
Lightweight Directory Access Protocol (LDAP) service, 630
Message Driven Bean (MDB), 629
overview, 622–626
Remote Method Invocation, 627
Session Beans, 628–629
Stateful Session Beans, 629
Stateless Session Beans, 628
Universal Description, Discovery, and Integration (UDDI), 631
XML Web Services and SOAP, 627–628
Java 2 Enterprise Edition starter kit (J2EE-SKIT), 641–662
architectural overview, 643–648
Ddbms Subsystem, 648
design overview, 651–659
Ddbms Subsystem, 652–653
Dem subsystem, 655–656
extension design, 654–656
framework design, 651–654
Lem subsystem, 656
Parser Subsystem, 653–654
Planner subsystem, 655
Qualifier subsystem, 655
exception type classes, 658–659
extension exception base classes, 658
framework exception base classes, 658
subsystem-specific exception base classes, 659
extension architectural overview, 649–651
Dem subsystem, 650–651
Lem subsystem, 651
Planner subsystem, 650
Qualifier subsystem, 649–650
goals, 642–643
parameter type interfaces, 656–658
DET, 657
ParsedSQL, 657
QualifiedSQL, 657
QueryResultSet, 657–658
Parser Subsystem, 648–649
subsystem identification, 644–645
subsystem interaction overview, 645–648
Java Archive (JAR) files, 551, 638, 661
Java Bean, 638
Java Message Service (JMS) starter kit, 579–619
administrator objects, 583–584
connection factories and connections, 584
collection factory object, 584
destination object, 584–586
application–programming interface (API), 581
architecture, 581–586
creating new extensions, 615–616
framework goals, 607–609
communication topography, 607–608
deployment topography, 607–608
message reliability, 607–608
software complexity, 607–609
framework implementation package overview, 612–613
framework interface package overview, 609–612
ChatRoom, 611
Hub, 611
HubFactory, 611
Msg, 609
Node, 610–611
NodeFactory, 611
Java 2 Enterprise Edition (J2EE), 580
Java 2 Standard Edition (J2SE), 580
JMS API constructs, 598–602
connection factory, 598
connection, 598–599
destination, 600–601
message consumer, 601
message producer, 601
session, 599–600
JMS provider implementation alternatives, 602–606
administration facilities, 604–605
Apache ActiveMQ, 603–605
configuration, 603–604
software installation, 603
startup and shutdown, 604
OpenJMS, 605–606
administration facilities, 606
configuration, 605–606
software installation, 605
startup and shutdown, 606
sample extension architectural overview, 614
sample extension deployment overview, 615
sample extension design overview, 614–615
sample extension implementation overview, 615
sample extension overview, 613–615
service terminology, 582–583
thread-safety considerations, 584–585
message consumer objects, 585
message objects, 585
message producer objects, 585
not thread-safe, 584
session objects, 584
Java Message Service (JMS), 629–630
Java Naming and Directory Interface (JNDI), 598, 630–631
Java Server Pages (JSPs), 625, 638
Java Virtual Machine (JVM), 547
JBoss, 634
Join operator in Relational Algebra, 117
conditional joins, 117
equi-joins, 117
natural join, 117
JonAS, 635

K
Key, 358, 379

L
Late transaction rule, 209
Lem subsystem, 699–700, 706–707
Lightweight Directory Access Protocol (LDAP) service, 369, 379, 630, 638
Link type (LT), 437, 447
Linked table, 510
Linking member, 439, 447
Local conceptual model (LCM), 35, 79
Local conceptual schema, 510
Local COSs, 556
Local database environment (LDBE), 515, 533
Local databases (LDBs), 515, 533
Local deployment schema (LDS), 515, 521–522, 533
P2P DDBEs, 529
Local execution managers (LEM), 22
Local query service provider (LQSP), 515, 522–523, 533
registration and deregistration, 523
Local query, 174
Local recovery manager (LRM), 308–313
Local timestamp, 271
Local transaction, 53, 199, 240, 271, 326, 352
Local transaction monitor (LTM), 308, 510
Local Wait-For-Graph (LWFG), 262, 271
Localized data, design alternatives, 38
Location transparency, 68, 79
Lock conversion, 205, 240
Lock downgrade, 205, 240
Lock escalation, 205, 240
Lock matrix, 201, 240
growing phase, 202
one-phase locking (1PL), 201–202
shrinking phase, 202
strict two-phase locking, 203
two-phase locking (2PL), 201–204
Lock upgrade, 205, 240
Locking Service (Lock-S), 10, 32
Locking-based concurrency control algorithms, 201–208
Locking in relational databases, 204–206
index page locking, 207
intention lock modes, 205
intention locking, 204–205
lock conversion, 205
lock downgrade, 205
lock escalation, 205
lock upgrade, 205
multiple-granularity locking (MGL), 204–205
predicate locking, 207
Logging process, 314–316
transaction abort, 315
Logging process (contd.)
transaction commit, 315
transaction read, 315
transaction start, 314
transaction write, 315
Logical clock reading, 209, 240
Logical data models (LDMs), 386, 413–415, 420, 425–449, 447
purpose, 413–414
steps to create, 414–415
Logical level modeling language (LLML), 527
Logical modeling language (LML), 386
Long-lived transactions (LLTs), 197, 240
Lookup service, 566–567, 575, 618

M
Majority Consensus Algorithm, 283–284
Majority voting, 292
Managed component, 669–671, 685, 709
Managed/Unmanaged Components, 666, 669–671
Management, database, 3
Man-in-the-middle attack, 365, 379
Marshalling, 558, 575, 618, 638, 685, 709
Master–slave distributed transaction control, 27
Master–slave replication control algorithm, 280–282, 292
centralized transaction acceptance/rejection phase, 281
evaluation net [Nutt72], 280
Petri net, 280
Materialization, 133, 174
Maximum cardinality, 400
Meet-in-the-middle attack, 360, 379
Memory frame, 174
Memory page, 174
Message authentication code (MAC), 358, 362, 379
Message Consumer objects, 585
Message digest (MD), 358, 362, 379
Message Driven Bean (MDB), 629
Message objects, 585
Message Oriented Middleware (MOM), 562, 575, 618
Message passing, 553, 555, 575
Message Producer objects, 585
Messaging, 586–595, 618, 638, 685
acknowledgment, 591–593
application-defined protocols, 587–588
durable destinations, 591
effects and exceptions, 593–594
message persistence, 589–591
message reliability, 591
ordering, 594–595
reliability, 593
synchronous versus asynchronous messaging, 588–589
temporary destinations, 591
time-to-live (TTL) value, 590
Messaging domains, 595–598
point-to-point (PTP) messaging, 595–597
publish-and-subscribe (Pub/Sub) messaging, 595, 597–598
Metadata, 32, 392
Microsoft .NET platform, 572, 663–686
assembly deployment overview, 675–679
application configuration file, 676
assembly deployment alternatives, 677–678
configuration, 676–677
deploying a private assembly, 678–679
deploying shared assembly, 678
machine configuration file, 676
security configuration, 676
development overview, 674–675
directory services requirements, 681
distributed data platform requirements, 681–683
ADO.NET database connectivity, 682
.NET security, 682
.NET transactions, 682–683
fundamental vocabulary terms, 666–673
Application Domains, 667, 672–673
Assemblies, 666, 671–672
Base Class Library (BCL), 666–672
Common Language Runtime (CLR), 666
Common Language Specification (CLS), 666–672
Common Type System (CTS), 666, 668–669
Global Assembly Cache (GAC), 667, 672
Intermediate Language (IL), 666, 669
Managed/Unmanaged Components, 666, 669–671
.NET Remoting, 680–681
remote messaging requirements, 681
remote-code execution requirements, 680–681
Microsoft Intermediate Language (MSIL), 685
Middleware, 537, 542–544, 575, 618, 638, 685
Minimum cardinality, 400
Minterm predicate, 79
Mixed distribution, design alternative, 39
Modeling language (ML), data, 2, 32, 420, 425, 447
Modes, 359, 379
Modulus (MOD) function, 174
Multidatabase system (MDBS), 240, 458, 510
architecture classifications, 457–458
Multidatabase, 240
Multinumber set type, 438
Multiple-granularity locking (MGL), 204–205, 240
Multiple site failures in 2PC, 342
Multiple site failures in 3PC, 349
Multiprocessing System, 552–553, 576
Multitasking System, 551–553, 576
Multithreaded Programming, 552, 576
Multitransection processing systems, 189–200
conflicts, 191–192
equivalence, 192–193
schedule, 189–191
parallel schedule, 189, See also Serializable schedules
INDEX

serial schedule, 189
Multi-user System, 553, 576
Multiversion (MV) concurrency control algorithm, 212–213, 240
Mutual consistency, 292
Mutually exclusive participation, 399, 420
N
Naming Service, 576
Navigation, 448
Nested-loop joins, 123–124
.Net Remoting, 680–681
Network Data Language (NDL), 436
Network data model (NDM), 425, 436–440, 448
data items, 437
data storage, 438
forward engineering, rules for, 438–440
multimember set type, 438
navigation, 438
records, 437
reverse engineering, rules for, 440
set types, 437–438
system-owned set type, 438
Network Directory (ND), 20
Network partitioning, 348–351
New coordinator election in 2PC, 342
New DDBE architectures, 513–532, See also
Cooperative DDBEs; Peer-to-peer (P2P) DDBEs
New taxonomy, 13–17
Nickname, 510
Nonblocking, 560, 563, 576, 618
Nonclustered Index, 174
Nonfederated database (NFDB), 510
Nonfragmented, design alternative, 38
Nonidentifying AT, 397, 420
Nonidentifying RT, 401, 420
Nonpersistent messaging, 590, 619
Nonprocedural language, 174
Nonreplicated, design alternative, 38
Nonrepudiation, 358, 379, 685
Nonunique index, 121, 174
Nonvolatile storage (disk, tape), 304, 352
Normal conjunctive form, 174
Normal disjunctive form, 174
Normalization, 435–436, 448
‘Not Ready’ message, 329
O
Object-oriented data model (OODM), 425, 443–446, 448
class inheritance, 444
classes, 444
forward engineering, rules for, 444–445
methods, 443
nomenclature, 443–444
object identifier (OID), 444
properties, 443
reverse engineering, rules for, 445
One time pad (OTP), 362, 379
One-phase commit (1PC), 326–329, 352
One-phase locking (1PL), 201–202, 240
On-line analytical processing (OLAP), 189, 240
On-line transaction processing (OLTP) systems, 183, 240
On-the-fly, 174
Operating system authentication, 84
Operation shipping, 151, 174
Operations, database, 2–3
Optimistic concurrency control algorithm, 201, 213–214, 240
commit phase (CP), 213
execution phase (EP), 213
validation phase (VP), 213
Oracle Lock Compatibility Matrix, 220
Oracle Lock Duration, 216
Oracle Lock Escalation, 220
Oracle row-level Locks, 216
exclusive row-level lock, 217
row exclusive (RX) table lock, 217–218
row share (RS) table lock, 217–218
shared row-level lock, 216
table-level locks, 217
Ordering, message, 594–595
Output port, 271
Overlapping participation, 399, 420
Overwriting uncommitted data, 240
P
Padding, 360, 380
Parallel schedule, 189, 240
Parent–child record type, 441, 448
Parent record type, 441, 448
Parent transaction, 200
Parser subsystem, 648, 653–654, 698, 704–705
Partial commitment order (PCO), 194, 240
Partial data visibility (PDV), 16
Partial participation, 399, 420
Partial schema visibility (PSV), 16
Partial visibility (PV), 16
Partially replicated database, 292
design alternative, 39
Partitioning, 174
Partitions, 79
Peer-to-peer (P2P) DDBEs, 528–530, 533
client, 529
COOP versus, 530–532
COOP/P2P versus traditional DDBEs, 531
GIS, 529
implementation specifics, 530
LDS, 529
QSP, 528–530
Persistent data, 2, 32
Persistent messaging, 619
Pessimistic concurrency control algorithms, 201, 241
Petri net, 280
Phantom issue, 206–208, 241
Physical data models (PDMs), 386, 415–416, 420, 425
Physical level modeling language (PLML), 527
Physical modeling language (PML), 386
Ping, 326
Pipelining, 133
Plaintext, 358, 380
Plan generation, centralized systems, 133–135
access paths, 133
exhaustive search algorithm, 134
heuristics-based algorithm, 134
materializing, 133
pipelining, 133
query plan, 133
Planner subsystem, 650, 655, 699, 705–706
Platform requirements, DDBE, 535–577, 638, 661, 685, 709
database connectivity requirements, 569
distributed data, 537, 567–570
architectural security requirements, 567–569
distributed process, 555–567
directory service requirements, 564–567
remoteability requirements, 556–564
fundamental, 544–555
binding requirements, 546
competitive synchronization, 554
concurrent execution requirements, 553–555
cooperative synchronization, 554
COS configuration requirements, 548, 550
COS deployment requirements, 548, 550–551
COS pooling requirements, 548–549
COS versioning requirements, 548–550
deployment requirements, 548–551
development requirements, 544–548
execution requirements, 551–555
extensibility requirements, 545–547
integration requirements, 544–545
message passing, 553, 555
multitasking requirements, 551–553
multitasking via multiple threads, 552
multitasking via multiple users, 553
namespace requirements, 545
portability requirements, 545, 547–548
reentrancy, 553
reference requirements, 546
semaphores, 553–555
synchronization, 553–554
thread safety, 553, 554
Java 2 Enterprise Edition (J2EE) platform, 571–572
Java Message Service (JMS) platform, 571
Microsoft .NET framework-based platform, 572
transaction management requirements, 570
Plural AT, 397–398, 421
Plural Identifying Atomic (PIA), 397–398, 429
Plural Identifying Grouping (PIG), 397–398, 429
Plural Nonidentifying Atomic (PNA), 397–398, 430
Plural Nonidentifying Grouping, 397–398, 432
Point-in-time (FIT) recovery, 197, 241, 352
Point-to-point (FTP) messaging, 595–597
Pooling, 576
Portable Executable (PE) Format, 671, 685
Posted update approach (PUA), 290–291, 292
Postexecution time validation, 97, 106
Potential for a global deadlock, 272
Preacquisition of all locks, 248, 272
Predicate locking, 207, 241
‘Prepare’ message, 329
Pre-write, 241
Primary copy 2PL, 230, 241
Primary copy approach, 278, 292
Primary horizontal fragmentation (PHF), 42–43, 79
Primary key (PK), 428–429, 448
Primary site, 278
Private key, 380, 685
Probing, 174
Procedural languages, 174
Project operator in RA, 114
Property, 443–444, 448
Public key cryptography, 358, 359, 361–364, 380, 685
Diffie–Hellman algorithm, 358, 359, 361–364, 380, 685
RSA algorithm, 363
Publish-and-subscribe (Pub/Sub) messaging, 595, 597–598
Publisher, 292
Q
Qualifier subsystem, 649–650, 655, 699, 705
Quel, 112, 174
Query, 3, 32
Query optimization, 8, 32, 111–175, 174, See also
Centralized systems, query processing in
distributed, 150–170
data shipping, 152
distributed resources, utilization, 151–152
dynamic programming in, 152–156
hybrid shipping, 152
operation shipping, 151
query trading in, 156–157
distributed query solution space reduction,
157–170
apply select and project as soon as possible, 157
Bit Vectors Use, 165
materialize common subexpressions once, 161
perform operations where most of the data is,
158–159
simplify join operations, 159
simplify operations on horizontal fragments,
158
use semi-join to reduce communication cost,
162
heterogeneous database systems, 170–172
multiphase approach, space reduction, 166
data localization (Phase 2), 167
data shipping for binary operations (Phase 4), 168
function shipping (Phase 3), 168
tree selection (Phase 1), 167
sample database, 112
Query Optimization Service (Qopt-S), 32
Query parsing in centralized systems, 127–128
Query plan, 174
Query processing in distributed systems, 145–172, 174
architecture, 146
mapping global query into local queries, 146–150
Query Request Service (Qreq-S), 8, 33
Query service provider (QSP), P2P DDBEs, 529–530
Query trading, 174, 532
Query tree, 175
Queue, 596–598, 600–606, 619
Quiescent point, 352
Quorum-based commit (QBC) protocol, 348–351

R
Read quorum, 292
Reading uncommitted data, 191, 241
Read-only transaction, 241
Real DBMS (Oracle), concurrency control in, 214–222
enforcing serializability in Oracle, 216–222
Oracle lock compatibility matrix, 220
oracle lock duration, 216
Oracle lock escalation, 220
oracle lock types, 216
Oracle row-level locks, 216
read committed level, 216
serializable level, 216
Reconstructiveness, 40
Record, 437, 448
Recoverable schedules, 197, 241
Recovery process, 316–319, See also Transaction recovery
based on REDO/UNDO processes, 322–323
checkpointing, 318–319
complete recovery algorithm, 323–324
deferred update, 317
from disk failure, 317–318
immediate update, 317
log information analysis, 316
from a power failure, 317
recovery to a point-in-time in the past, 319–320
recovery to current, 319
Recovery protocol, 332, 340–341, 352
Recovery to current, 353
Recursive relationship type, 400–401, 404, 438–440
Redo scripts, 353
Redundant array of independent disks (RAID), 303
Reentrancy, 553–554, 558
Reentrant Code, 576
Reference DDBE architecture, 18–24
DDBE information architecture, 18–20
Referential constraints, in semantic integrity control, 88
Referential integrity (RI), 90–91, 106
Registration service, 565, 576
Relation, 426–428, 448
Relation constraints, in semantic integrity control, 88, 107
Relational algebra (RA), 112–119, 175
computing RA operators, 119–126, See also individual entry
nonprocedural, 112
procedural, 112
structured query language (SQL), 112
subset of RA commands, 113–119, See also individual entry
Relational algebra derived operators, 116–119
declare operator in, 118
join operator, 117
conditional joins, 117
equi-joins, 117
natural join, 117
set intersect (SI) operator, 117
Relational calculus, 112, 175
Relational constraints, 107
Relational data model (RDM), 386, 426–436, 448, 527
candidate keys (CKs), 429
foreign key (FK), 429
forward engineering, rules for, 429–435
keys, 426	nomenclature, 426–429
domains, 426
Relational data model (RDM) (contd.)
relations, 426
tuples, 426
primary key (PK), 428
Relational schema, 449
Relationship type (RT), 399–401, 421
Reliability, message, 571, 593, 607–608
Remote COS, 556
Remote Messaging, 560–564
blocking, 563
communication errors and exceptions, 564
durability, 562
J2EE, 629–630
message oriented programming, 562
messages, 561
nonblocking, 563
ordering, 562
requirements, 681
timeouts, 562
Remote Method Invocation (RMI), 576, 627, 638
Remote Procedure Call (RPC), 558, 576
Remoteability requirements, 556–564
general-purpose mechanisms, 557
Remoteability requirements (contd.)
local COSs, 556
Remote-Code Execution, 557–560
remote COS, 556
remote messaging, 557, 560–564
task-specific mechanisms, 557
Remote-Code Execution, 557–560
blocking calls, 560
communication errors and exceptions, 560
marshalling parameters, 558
nonblocking calls, 560
Remote Method Invocation (RMI), 558
Remote Procedure Calls (RPC), 558
requirements, 680–681
RPC versus RMI, 558
unmarshalling parameters, 558
Remoting, 619, 638, 685, 709
Removal, deadlock, 252
Repeating group, 449
Replication, 68
Replication control, 275–293
algorithms, 279–291
architectural considerations, 279–280
asynchronous replication control, 276, 278–279
circulating token algorithm, 284–287
distributed voting algorithm, 282–283
generalized voting algorithm for, 287–290
majority consensus algorithm, 283–284
master–slave replication control algorithm, 280–282
posted update approach (PUA), 290–291
scenarios, 276–279
synchronous replication control, 276–278
transaction application rule, 284
voting rule, 284
Replication Directory (RD), 20
Replication Service (Repl-S), 11, 33
Replication transparency, 69, 79
Resolution rule, 284
Reverse engineering (RE), 388, 418, 425
rules for, 435
Rijndael, 380
Role, concept, 85, 107
Rollout, 278
Row exclusive (RX) table lock, 217–218
Row share (RS) table lock, 217–218
Row trigger, 107
Rowid, 175
Rule-based optimization (RBO) approaches, 134, 175
Run time validation, 97, 107
S
Sagas, 198–199, 241
Save point, 198, 241
Schedule, 189–191, 241, See also Serializable schedules
Schema and data control (SAD-CON), 16–17
Schema and data visibility (SAD-VIS), 15–16
Schema integration issues, 35, 79, 466–477, 510
centralized implementation of a physical GCS, 467
distributed implementation of physical GCS, 468
GCS creation issues, 469–477
discovery (Step 3.2), 472
export schema generation (Step 2), 470–471
merging (Step 3.4), 474
resolution (Step 3.3), 473
schema integration (Step 3.1), 471
translation to a common data model (Step 1), 469
GCS maintenance issues, 466–469
logical implementation of GCS, 467
maintaining currency for a physical GCS, 467
physical implementation of GCS, 467
Schema translation, 510
Second normal form (2NF), 449
Secondary copy, 292
Secondary Storage, 33
Secure Shell (SSH), 368, 380
Secure Sockets Layer (SSL) protocol, 366–367
Securing communications, 366–368
IPSec, 368
secure shell (SSH), 368
SSL/TLS, 366–367
Uniform Resource Locator (URL), 367
Virtual Private Networks (VPNs), 366–368
Securing data, 368–375
authentication, 368–369
authorization, 368–369
bypassing password for change password, 372
bypassing password for customer login, 371–372
data auditing, 375
data encryption, 369–370
data inference, 374–375
encrypted file systems (EFSs), 369
firewalls, 369
intrusion detection systems (IDSs), 369
lightweight directory access protocol (LDAP), 369
single sign on (SSO), 369
SQL injection, 370–374
unvalidated input, 370–374
vandalizing DDBE, 372–374
Security, 84
Security policy, 567–568, 576
Security Service (Sec-S), 7, 33
Security target, 567, 576
Select operator in RA, 114
Semantic impedance issues, 392
Semantic integrity control, 86–94, 107
distributed, 94–97
compile time validation, 97
postexecution time validation, 97
run time validation, 97
referential integrity, 87
semantic integrity constraints, 88–94
data type, 88
explicit, 88
referential, 88
referential integrity (RI), 90–91
relation, 88
stored procedures, 91
triggers, 91
user-defined data type, 88
Semantic integrity enforcement, cost of, 97–106
in distributed system, 98–106
compile time validation, 102–103
postexecution time validation, 104–106
run time validation, 103
variables used, 100–102
Semantic Integrity Service (Semi-S), 7, 33
Semaphores, 553–555, 576
Semi-join, 175
Sequential schedule, 241
Serial schedule, 189, 241
Serializable schedules, 193–197, 241
cascadeless schedules, 197
in centralized system, 194
conflict serializable schedules, 196
in distributed system, 195–196
point-in-time (PIT) recovery, 197
recoverable schedules, 197
view serializable schedules, 196
Service Consumer, 33, 576, 619, 709
Service-oriented architecture (SOA), 538, 540–542, 576
Service Provider, 33, 576, 619, 709
Services, 4–5, 33, 576, 619
Servlet, 638
Session Bean, 638, 661
Session object, 619
Set Difference Operator in Relational Algebra, 115
Set intersect (SI) operator in relational algebra, 117
Set Type, 437–438, 449
Share lock mode (SLM), 216, 241
Shared row-level lock, 216
Short-lived transactions (SLTs), 197, 241
Shrinking phase, 241
Simple Object Access Protocol (SOAP), 627–628, 638, 685
Simple Predicate, 79
Single on (SSO), 369, 380
Singlecall, 685, 709
Singleton, 685, 709
Singular AT, 397–398, 421
Singular Identifying Atomic (SIA), 397–398, 429
Singular Identifying Grouping (SIG), 397–398, 429–430
Singular Nonidentifying Grouping, 397–398, 430
Sites, 5–6
Slaves, 325
Soap, 627–628, 638, 685–686
Soft failure, 297–298, 353
Software architecture, DDBE, 20–24
application processor (AP), 20
components of, 21–23
bottom-up, 22
data processor (DP), 20
components of, 23–24
generic, 23
top-down, 21
Software integration issues, 465–466
Solution space, 175
Sort–merge join, 124–126, 175
external sort, 124
merge phase (Phase 2), 125
sort phase (Phase 1), 125
Splitting, in vertical fragmentation, 49–62
bond contribution calculation, 54–62
in distributed systems, 50
access frequency, 50
affinity matrix, 52
bond energy algorithm (BEA), 53
clustered affinity matrix, 53
process column affinity matrix, 52
usage matrix, 50
SQL injection, 370–374, 380
Stable storage, 304, 353
Standard Planning and Requirements Committee (SPARC), 18
Stateful Session Beans, 629, 638, 661
Stateless Session Beans, 628, 639, 662
Statement trigger, 107
Steganography, 357, 380
Store and forward, 292
Stored procedures, semantic integrity control, 91, 107
Stream ciphers, 359, 380
Strict two-phase locking, 203, 241
Strong ET, 398, 421
Structure, data, 2
Structured Query Language (SQL) approach, 112, 175, 368, 523
Subplatform, 639
Subqueries, 3
Subscriber, 292
Subset of RA commands, 113–119
basic operators, 113
Cartesian product (CP) operator, 116
combining select and project, 114
derived operators, 113
project operator, 114
relational algebra derived operators, 116–119
select operator, 114
set difference operator, 115
union operators, 115
Subsystems, 5, 33, 577, 619, 639, 662, 686, 709
Summary Level Diagrams, 11–13
Symmetric replication, 279, 292
Synchronization, 278, 553–554, 577
Synchronous messaging, 588–589, 619
Synchronous replication, 276–278, 292
System change number (SCN), 215
System crash, 297, 353
System failures, 298
System lock, 241
System-owned set type, 438, 449

**T**
Table-level locks, 217
Temporary Queue, 619
Temporary Topic, 597–598, 600–606, 619
Three-level model (3LM) taxonomy, 388–389
Three-phase commit protocol (3PC), 345–349, 353
Three-tier model (3TM), 538–540, 577
Timestamp concurrency control algorithms, 209–213
access rule, 209
basic algorithm, 209–210
commit phase, 210
conservative TO algorithm, 210–212
late transaction rule, 209
logical clock reading, 209
multiversion (MV) concurrency control algorithm, 212–213
pre-write phase, 210
timestamp ordering (TO), 209
younger transaction wait rule, 210
Timestamp ordering (TO), 209, 241
Timestamping Service (Time-S), 10, 33
Time-to-live (TTL) value, 590, 619
Top–down design approach, 35, 36, 79, 459, 510
Topic, 597–598, 600–606, 619
Total commitment order (TCO) graph, 194, 242
Total data visibility (TDV), 16
Total participation, 399, 421
Total schema visibility (TSV), 16
Total visibility (TV), 16
Traditional DDBE architectures, 451–511
classifying, 453–454
database distribution, 489–490
deployment of DDBE software, 461–463
developing approaches, 459–461
top–down methodology, 459
federated database architecture, classifying, 454–456
COS-COO Level, 455
COS-DAD Level, 455
SAD-CO Level, 455–456
SAD-VIS Level, 455
integration challenges, 463–477
DBMS autonomy, 463
degrees of integration, 464–465
heterogeneity, 463
schema integration issues, 466–477
software integration issues, 465–466
transparency, 463
linking data sources
into access, 495–497
into federation server, 497–502
MDBS architecture classifications, 457–458
nonfederated database architecture, classifying, 456–457
setting up data source names, 492–497
setup, 490–492
taxonomy, 452–457
Transaction acceptance phase, 293
Transaction application rule, 284, 293
Transaction attribute, 633–634, 637, 639, 662, 682–683, 686
Transaction classes, 235–236
Transaction control issue, 254
Transaction failures, 298
Transaction log, 302–303, 353
Transaction management in distributed systems,
24–31
distributed fund transfer transaction, 24
master–slave distributed transaction control, 27
Transaction management requirements, 570
Transaction Management Service (Tran-S), 10, 33
Transaction monitor (TM), 193, 242, 325
Transaction recovery, 320–322, 353
from disk failure, 321
from power failure, 321
using REDO scripts, 321–322
using UNDO scripts, 320
Transaction Roll Forward (Redo), 300
Transaction Rollback (Undo), 299–300
Transaction starvation, 249, 272
Transaction states, 300–302, 313–314
‘aborted’ state, 302
‘active’ state, 302
‘before commit’ state, 302
‘committed’ state, 302
defered update transaction steps, 313–314
‘failed’ state, 302
immediate update transaction steps, 314
Transactions, 184–189, 242
Multitran, 184
‘child transaction’, 200
INDEX 727

compensating transactions, 198
consistency property, 186
ConTracts, 199
in distributed system, 199–200
durability property, 187
End–Tran, 184
forward recovery, 198
global transaction, 199
isolation property, 187
J2EE, 633
life cycle, 189
local transaction, 199
long-lived transactions (LLTs), 197
parent transaction, 200
save point, 198
short-lived transactions (SLTs), 197
Translation in centralized systems, 127–128
Transport Layer Security (TLS) protocol, 366–367, 380
Triggers, semantic integrity control, 91, 107
after trigger, 93
before trigger, 93
row trigger, 93
statement trigger, 93
Triple DES, 380
Tuples, 426–429, 449
Two-phase commit (2PC), 326, 329–345, 353
multiple site failures in 2PC, 342
new coordinator election in 2PC, 342
2PC coordinator recovery process, 341–342
2PC slave recovery process, 342
recovery protocol, 332, 340–341
independent recovery, 340
recovering from a failure using state information, 340
recovery by discovery, 340
resiliency of 2PC and failure handling, 332–336
termination protocol, 332, 336–340
two-phase commit performance, 343–345
presumed abort, 343
presumed commit, 343
Two-phase locking (2PL), 201–204, 242
in distributed systems, 229–232

U
Undo scripts, 353
Undo, 353
UNDO/REDO and database recovery, 308–313
backup types, 311–313
complete backup, 311
database backups, 311
database buffers, 309–311
incremental backup, 311
local recovery manager (LRM), 308–313
local transaction monitor (LTM), 308
log archival, 311
log buffers, 309–311
write-ahead-logging (WAL), 309
Unified Modeling Language (UML), 390, 408–411, 421
Unified schema, 35, 79, 510
Uniform Resource Locator (URL), 367, 598
Union operators in relational algebra, 115
Unique index, 121, 175
Universal description, discovery, and integration (UDDI) facility, 517, 533, 631, 639, 686
Unmanaged component, 686, 709
Unmarshalling, 639, 686, 710
Unmarshalling parameters, 558, 577
Unrepeatable read, 191, 242
Unvalidated Input, 370–374
Usage Matrix, 79
User-defined data type, 88, 107
User interface (UI) module, 21
User Interface Service (UI-S), 8
User presentation tier, 538–540, 577
User queries, distribution impact on, 69–73
GDD containing location information—location transparency, 72–73
no GDD—no transparency, 70–72
User Services Tier, 538–540, 577

V
Vector, 449
Vertical fragmentation (VF), 40–42, 79
correctness rules, 62
design alternative, 39, 79
generation guidelines, 49–62
grouping, 49
splitting, 49–62, See also individual entry
View serializability, 196, 242
Virtual Link Type (VLT), 441, 449
Virtual parent–child record type, 441, 449
Virtual Private Networks (VPNs), 366, 368, 380
Virtual record type, 441, 449
Volatile storage (memory), 303, 353
Voting rule, 284

W
Wait–die, 249–250, 272
distributed wait–die algorithm, 255–256
Wait-for-graph (WFG), 247, 272
Weak ET, 398, 421
Web archive (WAR) files, 551, 639
Web container, 639
Web service, 533, 686
Web services description language (WSDL), 517, 533, 639, 686
Weighted voting, 293
Wellknown server activated object, 686, 694, 710
Window of vulnerability (WoV), 327
Wound–wait, 249, 251, 252
  distributed wound–wait algorithm, 256–257
Wrapper, 175, 511
Write-ahead-logging (WAL), 309
Write quorum, 293

X
XML Web Services, 627–628, 639, 686, 710

Y
Younger transaction wait rule, 210