

INDEX

- aborted transaction, 432
- absolute address, 366
- abstraction. *See* generalization
- access category, 82
- access modifiers, 59
- access paths, 397
- access transparency, 525
- accessibility dimension, 617
- accessor methods, 208
- accuracy dimension, 617
- accuracy ratio (AR), 687
- ACID properties
 - defined, 15, 452–453
 - in loosely coupled systems, 535–538
 - in NoSQL, 538
- active DBMS, 232–236
- ActiveX Data Objects (ADO), 468, 502
- activity services, 607–608
- actuator, 353
- ADO.NET, 468–471, 502, 533–534
- after images, 435
- after trigger, 233
- agglomerative hierarchical clustering, 693
- aggregate functions, 157
- aggregated data, 717
- aggregation
 - in EER model, 55
 - mapping EER to relational, 137
 - in UML, 62–63
- AJAX (Asynchronous JavaScript and XML), 508–509
- ALL, 178–181
- allocation, 519, 523
- ALTER, 155–156
- alternative keys, 109
- Amazon, 593
- Amazon Redshift, 572, 621
- Amazon Relational Database Service (RDS), 621
- Amazon Web Services, 706
- Amsterdam, 8
- analytics
 - applications, 667
 - data pre-processing, 669–672
 - economic perspectives
 - in- versus outsourcing, 664–704
 - on-premises versus cloud solutions, 705–706
 - open-source versus commercial software, 706–708
 - return on investment, 702–704
 - total cost of ownership, 702
 - improving ROI of
 - cross-fertilization, 713–714
 - data quality, 711–712
 - management support, 712
 - new data sources, 708–711
 - organizational aspects, 712–713
 - post-processing, 700–701
 - predictive model evaluation, 689–696
 - privacy and security
 - accessing internal data, 717
 - anonymization, 717–718
 - definitions and considerations for, 714–715
 - encryption, 721
 - importance of, 714
 - label-based access control, 719–721
 - RACI matrix, 715–716
 - regulations, 721–723
 - SQL views, 719
 - process model, 665–666
 - success factors for, 701
 - types of
 - descriptive, 689–695
 - predictive, 673–682
 - social network, 695–700
- analytics process model, 665–666
- anonymization (data), 717–718
- ANY, 178–181
- Apache Flume, 621
- Apache Hadoop, 631
- Apache Kylin, 621
- Apache Lucene, 616
- Apache Spark
 - background of, 652–653
 - GraphX, 658
 - MLlib, 656–657
 - Spark Core, 653–654
 - Spark SQL, 654–656
 - Spark Streaming, 657–658
- Apache Sqoop, 621
- Apache Storm, 658
- Apollo program, 97
- application developer, 12
- application programming interface (API)
 - classification
 - background, 462–463
 - early binding versus late binding, 465–466
 - embedded versus call-level, 464
 - proprietary versus universal, 463–464
 - object persistence
 - Enterprise JavaBeans, 484–488
 - Entity Framework, 498–499
 - Java Data Objects, 495–498
 - Java Persistence API, 488–494
 - object-relational mapping, 483–484, 498
 - SQLAlchemy, 499–502
 - universal database
 - ADO.NET, 468–471
 - embedded API versus embedded DBMS, 480–482
 - JDBC, 471–477
 - language-integrated querying, 482–483
 - ODBC, 466–467
 - OLE DB and ADO, 467–468
 - SQL injection, 477–479
 - SQLJ, 479–480

- architecture categorization, 30–31
- architecture components
 - connection and security manager, 21–22
 - DDL compiler, 22
 - interacting with
 - DDL statements, 21
 - embedded DML statements, 21
 - interactive queries, 21
 - interfaces, 27
 - query processor, 22–25
 - storage manager, 25–26
 - utilities, 26
- archiving, 438
- arcs, 333
- area under the ROC curve (AUC), 685
- ASP (Active Server Pages), 506
- association class, 60
- association rules
 - basic setting, 689–690
 - defined, 689
 - post-processing, 691
 - support, confidence, and lift, 690–691
- associations, 59–61, *See also*
 - relationship type
- associative query, 221
- Asynchronous JavaScript and XML (AJAX), 508–509
- atomic attribute type, 42
- atomic literal, 217
- atomic search key, 402–404
- atomicity property, 15, 452, 530–532
- attribute type
 - defined, 40
 - in ER model, 42–43
 - in file organization, 362
 - in index creation, 400
 - relationship, 46
- attributes, 57
- authorization identifier, 150
- availability, 539
- AVG, 162
- Axibase, 342

- B⁺-trees, 378, 388–389, 402–404
- Bachman diagram, 98
- backup, 438
- backup and recovery utility
 - and data availability, 423–425
 - as database advantage, 15
 - defined, 27
- Baesens, B., 82, 85
- bag. *See* multiset
- BASE principle, 312
- BASE transactions, 540
- Bayer, Rudolf, 388
- BayesDB, 342
- Bean-Managed Persistence (BMP), 488
- before image, 435
- before trigger, 233
- begin_transaction instruction, 432
- behavior (OO), 244
- BETWEEN, 159
- BFI, 384
- bidirectional association, 61
- Big Data
 - Apache Spark
 - background of, 652–653
 - GraphX, 658
 - MLlib, 656–657
 - Spark Core, 653–654
 - Spark SQL, 654–656
 - Spark Streaming, 657–658
 - data integration outlook, 621
 - defined, 627
 - Hadoop
 - definition and design, 630
 - history of, 630–631
 - SQL, 643–652
 - stack, 631–643
 - scope of
 - value, 627–629
 - variety, 627–629
 - velocity, 627–629
 - veracity, 627–629
 - volume, 627–629
- BigQuery ETL, 621
- binary large object (BLOB), 155–156, 247, 360–361
- binary relationship type
 - cardinalities, 45
 - defined, 44–45
 - mapped to a relational model, 122–127
 - and ternary types, 48–50
- binary search, 364
- binary search trees, 385–386
- biometric data, 3
- bitmap index, 383
- BLOB (binary large object), 155–156, 247, 360–361
- block, 397
- block-level I/O protocols, 415
- block pointer, 371
- blockchains, 524
- blocking factor (BF), 361
- bootstrapping, 683–684
- Boyce, Raymond, 120
- Boyce–Codd normal form (BCNF), 119–120
- Brewer, Eric, 312–313, 539
- B-tree, 378, 386–388
- bucket, 365, 368–369
- buffer manager, 26
- business activity monitoring (BAM), 593
- business continuity
 - contingency planning, recovery point and recovery time, 398–421
 - defined, 421
- business intelligence (BI). *See also*
 - decision-making
 - defined, 572
 - hybrid OLAP, 575
 - multidimensional OLAP, 574–587
 - on-line analytical processing, 574
 - operational BI, 592
 - pivot tables, 573
 - query and reporting, 573–587
 - relational OLAP, 575
- business process
 - defined, 601
 - in database design, 38
- business process integration
 - data and process integration in, 606–610
 - defined and modeling, 601–602
 - managing dependencies, 604–606
 - manual processes, 602–604
- CallableStatement, 475–476
- call-level APIs, 464
- candidate key
 - defined, 108
 - in file organization, 362
 - in index creation, 400
- canonical form, 526
- CAP theorem, 312–313, 539
- Capability Maturity Model Integration (CMMI), 619–620
- cardinalities. *See also* multiplicities
 - CODASYL model, 101
 - ER model, 45
- Cartesian product, 108
- cascading rollback, 447–448
- catalog
 - and metadata role, 80
 - data types in, 401
 - defined, 10–11

- categorization
 - based on architecture, 30–31
 - based on data model, 28–30
 - based on degree of simultaneous access, 30
 - based on usage, 31–32
 - in EER model, 54–55
 - mapping EER to relational, 136–137
- central processing unit (CPU), 352
- central storage, 352
- centrality metrics, 696–698
- centralized DBMS architecture, 459–460
- centralized system architecture, 30
- chaining, 370
- changeability property, 65
- changed data capture (CDC), 598
- character large object (CLOB), 247, 360–361, 610
- CHECK constraint, 151
- checkpoints, 435
- Chen, Peter Pin-Shan, 40
- choreography, 604–606
- churn prediction, 667
- class, 57
- class diagram, 58
- class invariant, 65
- classification
 - defined, 673
 - performance measures for, 684–687
- classification accuracy, 685
- cleansing, 566
- client–server DBMS architecture, 31, 459–460
- client-side scripting, 507–508
- CLOB (character large object), 247, 360–361, 610
- cloud DBMS architecture
 - analytics, 705–706
 - data in the, 600–601
 - data warehousing in, 572
 - defined, 31
 - tiered system architectures, 462
- cloud storage, 421
- cloud-based solutions, 705–706
- CLUSTER, 398
- clustered index, 373–374, 398
- clustering
 - defined, 423, 692–693
 - hierarchical clustering, 693–695
 - K-means, 695
- CODASYL model
 - defined, 97
 - key building blocks, 97–101
- Codd, Edgar F., 104–105, 120
- coefficient of determination, 688
- collection literal, 218
- collection types (OO), 245–247
- collision, 366
- column constraints, 151
- column value, 399
- column-oriented DBMS, 331–332
- combination notation, 404
- combined approach, 270–271
- commercial analytical software, 706–708
- committed, 432
- common gateway interface (CGI), 504–507
- Common Language Runtime (CLR), 468–471
- compatibility matrix, 445
- compensation-based transaction models, 535–538
- completeness constraint, 53
- completeness dimension, 82–84
- composite aggregation, 62–63
- composite attribute type, 42
- composite key, 362
- comprehensibility, 689
- conceptual data model
 - advantages of, 13
 - defined, 9
 - in design phase, 39
 - EER, 52–57
 - ER, 40–52, 121–133
 - not stored in catalog, 10
 - physical design architecture, 357–358
 - UML class diagram, 57–66
- conceptual/logical layer, 10
- concurrency control
 - defined, 6, 14–15
 - in distributed databases, 528–534
 - locking protocol, 444–452
 - multi-version, 541–542
 - optimistic and pessimistic schedulers, 443–444
 - problems, 439–442
 - schedule and serial schedule, 442
 - serializable schedules, 442–443
 - in transactions, 431
- confidence, 690
- conformed dimensions, 568–569
- connection manager, 21–22
- Connection object, 477
- connectivity, 414
- conservative 2PL, 446
- consistency
 - in CAP theorem, 539
 - eventual, 312–313, 540
 - quorum-based, 542–544
- consistency dimension, 82–84
- consistency in databases, 300–301
- consistency property, 15, 452
- consistent hashing, 309–310, 538
- constructor, 210
- container managed relationships (CMR), 488
- container-managed persistence (CMP), 488
- contextual category, 82
- contingency plan, 398–421
- Control Objectives for Information and Related Technology (COBIT), 620
- correlated nested queries, 175–178
- cost-based optimizer, 400
- COUNT, 161, 222–223
- credit scoring models, 667
- cross-table, 573
- cross-validation, 682–683
- CRUDS functionality, 608
- CUBE, 577–578
- cube (three-dimensional), 575
- cumulative accuracy profile (CAP), 686–687
- cursor mechanism, 474
- customer relationship management system (CRM), 5, 628
- customer segmentation, 667
- cutoff, 685
- Cutting, Doug, 630
- cylinder, 354
- Cypher
 - in graph-based database, 334
 - overview of, 335–341
- data access request, 717
- data accessibility, 84
- data accuracy, 82
- Data as a Service (DaaS), 599–601
- data auditing services, 610
- data availability, 423–425
- data cleansing services, 609
- data completeness, 617
- data consistency, 617
- data consolidation, 593–595
- data definition language (DDL), 12
- data definitions, 21
 - and DDL compiler, 22
 - versus actual data, 8–9

- data dependency
 - defined, 604
 - managing, 604–610
- data enrichment services, 609
- data event services, 610
- data federation, 595–596
- data flow, 607
- data governance, 85–86, 712–713
- data in the cloud, 600–601
- data independence, 12
- data integration
 - in business process integration, 606–610
 - defined, 591
 - outlook, 621
 - searching unstructured documents
 - enterprise search, 616–617
 - full-text, 610–611
 - indexing, 611–613
 - web search engines, 613–616
- data integration pattern, 593
- data integrity rules, 14
- data item, 357
- data lake, 571–572
- data lineage, 608
- data localization, 526
- data management
 - in catalogs, 401
 - data quality, 81–85
 - defined, 79
 - governance, 85–86
 - and metadata catalogs, 80–81
 - roles in, 86–88
- Data Management Body of Knowledge (DMBOK), 620
- Data Management Maturity Model, 619–620
- data manipulation language (DML)
 - declarative, 22–25
 - defined, 12
 - procedural, 22–23
- data mart
 - as smaller data warehouse, 567–569
 - virtual, 569–570
- data model
 - categorization
 - extended relational, 29
 - hierarchical, 28
 - network, 28
 - not-only SQL, 30
 - object-oriented, 28
 - object-relational, 29
 - relational, 28
 - XML, 29
 - defined, 9–10
- data needs convergence, 591–593
- data owner, 87
- data pointers, 386
- data pooling firms, 709
- data pre-processing
 - defined, 669
 - denormalization, 669–670
 - exploratory analysis, 671
 - missing values, 671–672
 - outlier detection and handling, 672
 - sampling, 670
- data profiling services, 609
- data projections, 626
- data propagation
 - enterprise application integration, 596–597
 - enterprise data replication, 597
- data providers, 469
- data quality (DQ)
 - in analytics, 711–712
 - and data governance
 - Capability Maturity Model Integration, 619–620
 - Control Objectives for Information and Related Technology, 620
 - Data Management Body of Knowledge, 620
 - Information Technology Infrastructure Library, 621
 - Total Data Quality Management, 619
 - defined, 80–81
 - dimensions, 81–84
 - master data management, 617–618
 - problems, 84–85
- data redundancy, 14, 311, 438
- data replication, 311
- data scientist
 - and analytics processing, 666
 - defined, 88
 - job profile, 668–669
- data security, 15
- data service composition, 599
- data services, 608
- data silo, 591
- data stewards, 87
- data striping, 411–412
- data transformation services, 610
- data type, 359
- data virtualization, 598–599
- data warehouse
 - data marts, 567–569
 - definition, 553–554
 - ETL process, 565–567
 - Hive, 649–652
 - largest, 554
 - most popular vendors, 565
 - operational data store, 571
 - schemas
 - fact constellation, 557
 - snowflake, 556
 - star, 555–556
 - traditional set-up, 592
 - versus data lakes, 571–572
 - virtual, 569–570
- data adapter method, 470–471
- database, 4
- database access in the World Wide Web
 - client-side scripting, 507–508
 - common gateway interface, 504–507
 - JavaScript, 508–509
 - JSP, ASP, and ASP.NET, 506
 - original web server, 504
 - REST-based web services, 509–511
 - Simple Object Access Protocol, 509–511
- database access methods
 - atomic key index searches, 402–404
 - full table scan, 408
 - index-only access, 408
 - multiple index and multicolumn index search, 403–407
 - query optimizer functioning, 400–402
- database administrator (DBA)
 - and analytics processing, 666
 - data management role, 87
 - defined, 11
 - interaction tools, 21
 - privileges, 191–192
- database approach, 6–8
- database design phases, 38–39
- database designer, 11, 87
- database functionality, 422–423
- database languages, 12
- database management, 12–15
- database management system (DBMS)
 - architecture of, 20–27
 - categorization of
 - based on architecture, 30–31
 - based on data model, 28–30
 - based on degree of simultaneous access, 30
 - based on usage, 31–32
 - definition, 5
 - market value of, 5
 - object-oriented, 207–226

- database management system
 - architecture, 20–27
- database model
 - advantages of, 13
 - versus instances, 8–9
- database schema, 8–9
- database schema issues in data warehousing
 - dimension table optimization, 559–560
 - fact table granularity, 558–559
 - factless fact table, 559
 - junk dimensions, 560
 - outrigger table, 561
 - rapidly changing dimension, 563–565
 - slowly changing dimensions, 561–563
 - surrogate keys, 557–558
- database state, 8
- database system, 12–15
 - definition, 5
 - elements, 8–10
- database system architecture, 459–461
- database system types
 - extended relational, 231–249
 - legacy, 93–101
 - object-oriented, 207–226
 - relational, 104–137
 - SQL, 147–194
 - XML, 255–293
- database technology applications, 3–4
- database user types, 12
- DataNodes, 632–634
- data-oriented approach, 270
- dataset, 357
- Davenport, T.H., 712
- DBCLOB (double byte character large object), 247
- DDL compiler, 22
- DDL statements, 21
- deadlock, 448
 - detection and resolution, 448–449
 - prevention, 449
- decision making, 552–553, *See also* business intelligence
- decision phase, 531
- decision support systems (DSS), 552
- decision trees
 - defined, 677
 - properties of, 680
 - regression trees, 680–681
 - splitting decision, 677–679
 - stopping decision, 679–680
- declarative DML, 22–25
- deduplication, 567
- deep equality, 217
- deferred update, 437
- degree (of relationship type), 44
- Dejaeger, K., 82
- DELETE, 185–186, 474
- deletion anomaly, 113
- delimiters, 360
- delineation, 432
- Delta Airlines, 225
- dendrogram, 694
- denormalization, 669–670
- dense indexes, 371
- dependency
 - existence, 45, 47
 - full functional, 117
 - managing, 604–606
 - multi-valued, 120
 - transitive, 118
 - trivial functional, 119
 - in UML, 66
- dependent data marts, 568
- derived attribute type, 43
- derived fragmentation, 523
- descriptive analytics
 - association rules
 - basic setting, 689–690
 - defined, 689
 - post-processing, 691
 - support, confidence and lift, 690–691
 - clustering, 692–695
 - goal of, 689
 - sequence rules, 691–692
- detection (deadlock), 449
- dicing, 577
- dimension reduction, 710
- dimension table optimization, 559–560
- dimensions
 - accessibility, 84
 - completeness, 82–84
 - conformed, 568–569
 - consistency, 82–84
 - junk, 560
 - mini, 563
 - rapidly changing, 563–565
 - slowly changing, 561–563
 - table optimization, 559–560
- direct attach, 414
- direct file organization, 365–370
- directly accessible storage devices (DASDs), 353
- directly attached storage (DAS), 416, 419
- directory, 376
- dirty read problem, 440–441
- disaster tolerance, 421, 424–425
- discretization, 717
- disjoint specialization, 52
- disjointness constraint, 52
- disk arrays, 411–413
- disk blocks, 354, 361
- disk mirroring, 412, 438
- dissemination, 308
- DISTINCT, 170
- distinct data type, 238–252
- distributed 2PL, 529–530
- distributed database systems
 - architectural implications, 518–519
 - blockchains, 524
 - defined, 517–518
 - fragmentation, 520–523
 - metadata allocation, 524
 - replication, 523–524
- distributed query processing, 525–528
- divisive hierarchical clustering, 693
- DML compiler
 - in database access, 401
 - defined, 22–25
- document metadata, 613
- document stores, 315
- Document Type Definition (DTD), 260
- document-oriented approach, 270
- DOM API, 267
- domain
 - defined, 41
 - relational database, 106–108
 - in SQL, 150–151
 - in UML, 59
- double byte character large object (DBCLOB), 247
- doubling amount, 677
- DQ frameworks, 82
- drill-across, 576
- drill-down operator, 576
- drill-up operator, 575
- Driver Manager, 471–473, 479
- DROP, 155–156
- dumb client architecture, 460
- dummy record type, 99–100
- durability property, 15, 453
- dynamic binding, 212
- dynamic hashing, 370
- dynamic random access memory (DRAM), 356
- dynamic SQL, 466
- early binding API, 465, 502
- eBay, 629

- edge, 333, 696
 efficiency, 399
 Elasticsearch, 616
 electronic vaulting, 424
 ELK stack, 616
 embedded API, 464, 502
 embedded DBMSs, 480–482
 embedded DML statements, 21
 embedded documents, 320
 embedded identification, 359
 encapsulation, 208
 encryption, 721
 end_transaction, 432
 Enhanced Entity Relationship Model (EER)
 aggregation in, 55
 categorization in, 54–55
 designing a, 56–57
 examples of, 56
 foreign keys, 109
 mapped to a relational model, 133–137
 metadata modeling, 81
 and relational model, 106
 specialization and generalization in, 52–54
 SQL for metadata management, 192
 versus UML, 66
 ensemble methods, 681–682
 Enterprise Application Integration (EAI), 284, 596–597
 Enterprise Data Replication (EDR), 597
 Enterprise Information Integration (EII), 595–596
 Enterprise JavaBeans (EJB), 484–488, 502
 enterprise resource planning (ERP), 628
 enterprise search, 616–617
 enterprise storage subsystems
 directly attached storage (DAS), 416
 iSCSI/Storage over IP, 419–421
 NAS gateway, 418–419
 network attached storage (NAS), 417–418
 overview, 414–416
 storage area network (SAN), 416–417
 Entity Relationship model (ER)
 attribute types, 40–41
 domains, 41
 entity type, 40
 examples of, 50–51
 history of, 40
 mapped to a relational model, 121–133
 relationship types, 43–46
 temporal constraints of, 51–52
 ternary relationship types, 48–50
 weak entity types, 46–47
 entity type
 defined, 40
 mapped to a relational model, 121–122
 Entity Manager, 492–493
 Ethernet
 defined, 415
 iSCSI (internet SCSI), 419–420
 NAS, 417–418, 420
 ETL (extract, transform, load), 565–566, 593–595. *See also*
 near real-time ETL
 Event Store, 342
 eventual consistency, 312–313, 318–319, 540
 EXCEPT, 183–184
 exclusive lock, 445
 ExecuteReader method, 470
 ExecuteScalar method, 470
 existence dependency
 cardinality, 45
 and weak entity type, 47
 EXISTS, 181–182
 experts, 666
 explicit networks, 708–709
 exploratory analysis, 671
 extended relational DBMS (ERDBMS)
 active RDBMS extensions, 232–236
 defined, 29
 Extensible Markup Language (XML), 256–259
 Extensible Stylesheet Language (XSL), 263–266
 external data model
 defined, 10
 SQL having, 150
 SQL views, 188–190
 external scalar function, 241
 external table function, 241
 extraction strategy, 566
 extraction transformation and loading process (ETL), 565–567
 Facebook
 API, 463
 as Big Data, 628
 Hive, 649
 Presto, 652
 fact constellation, 557
 fact table granularity, 558–559
 factless fact table, 559
 failover, 416
 failover time, 438
 failure detection, 308
 failure types, 436
 fat client variant, 31
 fat server variant, 31
 featurization, 699–700
 federated database, 519, 525
 federated DBMS, 31
 Fibre Channel (FC), 415
 field, 357
 file-level I/O protocols, 415
 file system, 418
 file-based approach
 definition, 5–6
 record organization, 359
 fill factor, 389
 filter factor, 401
 filters, 316–320
 first normal form (1 NF), 115–117
 5 Vs of Big Data, 627–629
 FlockDB, 334
 FLWOR, 280–282
 foreign key
 defined, 109–110
 mapped to a relational model, 129–130
 and SQL referential integrity constraints, 154–155
 formatting rules, 566
 fourth normal form (4 NF), 120
 fragment query, 526
 fragmentation, 520–523
 fragmentation transparency, 524
 fragments, 519
 fraud detection, 667
 free-form language, 148
 FROM, 157
 full backup, 439
 full functional dependency, 117
 full outer join, 170–172
 full table scan, 408
 full-text
 indexing, 611–613
 searching, 610–611
 full-text search (XML), 280
 functional dependency, 114–115
 fuzzy logic, 612
 gain, 678
 galaxy schema, 557

- garbage collection, 541
- garbage in, garbage out (GIGO), 81, 572, 711
- General Data Protection Regulation (GDPR), 722
- generalization
 - defined, 52
 - in UML, 62
- geographical information systems (GIS)
 - applications, 4
 - as file-based approach, 5
- global deadlock, 530
- global query optimization, 526
- Google
 - consistency, 314
- Google Cloud Dataflow, 621
- Google File System, 630
- Google Trends, 616
- grain. *See* granularity
- granularity, 558–559
- graph theory, 333
- graph-based databases, 333, 709
- GraphX, 658
- GROUP BY, 163–164, 170
- GROUPING SETS, 579–580

- Hadoop
 - and Big Data, 621
 - definition and initial design, 630
 - history of, 630–631
 - stack
 - distributed file system, 631–635
 - MapReduce, 635–641
 - pure form, 631
 - YARN, 641–643
- Hadoop Common, 631
- Hadoop Distributed File System (HDFS), 631–635
- hard disk backup, 424
- hard disk controller, 353
- hard disk drive (HDD)
 - failures of, 631
 - internals of, 353–355
 - as secondary storage device, 352
- Harris, J.G., 712
- hash file organization, 365–370
- hash function, 304–305
- hash indexes, 383
- hash join, 410
- hashing, 362
- HAVING clause (SQL DML), 163–164
- HBase, 644–648
- heap file, 363

- Hibernate, 489
- hierarchical clustering, 693–695
- hierarchical DBMS, 28
- hierarchical model, 93–97
- Hive, 649–652
- horizontal fragmentation, 521, 538
- horizontal scaling
 - defined, 301
 - in distributed databases, 519
 - and NoSQL databases, 305–308
- HTML
 - client-side scripting languages, 507
 - as original web server language, 504
 - and XML, 263–266
 - versus XML, 256–257
 - web search engines and, 615
- HTTP, 506
 - definition, 285
 - and REST, 288–289
- hybrid OLAP (HOLAP), 575

- I/O, 353, 414–416
- I/O boundary, 353
- immediate update policy, 438
- immutable object identifier (OID), 58
- impedance mismatch problem, 24
- implementation, 209
- implicit networks, 708–709
- impurity, 677–679
- IN, 159
- inconsistent analysis problem, 441–442
- incremental backups, 439
- independent data marts, 586
- index, 190–191
- index design, 398–400
- index entry, 371
- index-only access, 408
- index search, 402–404
- index spaces, 397
- indexed sequential file organization
 - clustered indexes, 373–374
 - defined, 370
 - multilevel indexes, 374–375
 - primary indexes, 373
 - terminology of, 370–385
- indexer, 613
- indexing, 362, 611–613
- information analyst, 86
- information architect
 - data management role, 86
 - defined, 11
- information hiding, 58, 209
- Information Technology Infrastructure Library (ITIL), 621
- Infrastructure as a Service (IaaS), 600–601
- inheritance
 - at data type level, 242–243
 - defined, 210
 - in EER model, 54
 - at table type level, 243–244
 - in UML, 58
- in-memory DBMS, 31
- Inmon, Bill, 553, 569
- inner join, 166–171
- inner table, 409–410
- INSERT, 185, 474
- insertion anomaly, 112
- insourcing, 664–704
- integrator standard, 620
- integrity constraints, 314–315
- integrity rules, 14
- intention exclusive lock (ix-lock), 451–452
- intention lock, 451
- intention shared lock (is-lock), 451–452
- inter-query parallelism, 397, 519
- interactive queries, 21
- interface, 27, 209
- internal data model
 - defined, 10
 - in design phase, 40
 - physical database design facilitates, 356–358
 - and physical database organization, 396
 - and SQL, 149
 - SQL indexes, 190–191
- internal layer, 10
- internal representation format, 25
- Internet of Things (IoT), 4, 627–628
- INTERSECT, 183–184
- intersection, 405
- intervals, 370
- intra-query parallelism, 397, 519
- intrinsic category, 82
- inverted file
 - characteristics, 382
 - and database access, 370
 - defined, 380
- iSCSI (internet SCSI), 419–420
- isolation levels (locking protocol), 449–450
- isolation property, 15, 452

- Java (programming language)
 - hash map, 304
 - HDFS, 631, 634
 - and MapReduce, 636–641
 - MongoDB, 321–330
 - NoSQL databases, 316–320
 - object-oriented paradigm, 223–225
 - popularity of, 148
 - and SQL, 148
- Java applets, 507
- JavaBean, 484–488
- Java Data Objects (JDO), 495–498, 502
- Java DataBase Connectivity (JDBC), 471–477, 511
- Java Persistence API, 489–494, 502
- JavaScript, 507–510
- JavaScript Object Notation (JSON), 290–292
- JavaServer Pages (JSP), 486, 506
- join condition, 408–409
- join index, 384
- join queries
 - defined, 166
 - in index creation, 399
 - in physical database organization, 408–410
 - OQL, 222
- jOOQ, 482
- JSON standard, 316
- JSONB, 331
- junk dimension, 560

- key attribute type, 42
- key performance indicators (KPIs)
 - in business intelligence, 592
 - defined, 12–16
 - monitoring of, 27
- keys, 108–110
- key-to-address transformation, 365–368
- key-value stores, 304
- keyword-based search (XML), 280
- Kibana, 617
- Kimball, Ralph, 569
- K-means clustering, 695

- Label-Based Access Control (LBAC), 719–721
- LAN-free backup, 416–417
- language-integrated querying, 482–483
- large objects (LOBs), 247
- late binding, 465–466, 502
- latency, 354, 591–593
- left outer join, 170–172
- legacy, 93
- legacy databases
 - CODASYL model, 98–101
 - hierarchical model, 93–97
- legacy file-based system, 6
- Lemahieu, W., 82
- lift, 690
- lift curve, 686
- LIKE, 159
- linear decision boundary, 676
- linear list, 375–377
- linear regression, 673–675
- linear search, 362
- Linear Tape-Open (LTO), 424
- linked data, 283
- linked list
 - defined, 370
 - represented as tree structure, 378
- LinkedIn, 628
- LINQ, 483
- Lismont, J., 713
- list data organization, 375–378
- lists, 360, 375
- literal, 217
- loading factor, 368
- loading utility, 26
- local area network (LAN), 519
- local query optimization, 526
- location transparency, 524
- lock manager, 26
- lock table, 445
- locking, 444–446
- locking protocol
 - cascading rollbacks, 447–448
 - in concurrency control, 444
 - deadlocks, 448–449
 - defined, 26
 - isolation levels, 449–450
 - lock granularity, 450–452
 - purposes of, 444–446
 - Two-Phase Locking Protocol, 446–448
- log records, 435
- logfile, 435
- logical data independence, 12
- logical data model
 - defined, 9–10
 - in design phase, 39–40
 - physical design architecture, 357–358
 - SQL having, 150
 - transparency in, 524
- logistic regression, 675–677
- Logstash, 616
- long-running transactions, 534–535
- long-term locks, 449
- loosely coupled systems, 535
- Lorenz curve, 686
- lost update problem, 440, 447, 450

- macroeconomic data, 709
- mainframe architecture, 460
- manual failover, 422
- MapReduce
 - and Hadoop, 631, 635–641
 - and Hive, 651–652
 - innovative aspects of, 321–330
 - parallelization in, 630
- marketing analytics, 667
- master data management (MDM), 617–618
- MATCH, 335–341
- mathematical notation, 404
- maturity (analytics applications), 713–714
- McCreight, Edward, 388
- mean absolute deviation (MAD), 688
- mean squared error (MSE), 688
- media failure, 436
- media recovery, 438–439
- member record type, 98
- membership protocol, 308
- Memcached, 306–308, 353
- message-oriented middleware (MOM), 284–285
- metadata
 - and catalogs, 80
 - in database approach, 6
 - distribution and replication of, 524
 - document, 613
 - modeling, 80–81
 - NameNode, 632
 - semantic, 567
 - SQL for, 192–193
 - structural, 567
- metadata services, 610
- metamodel, 80–81
- method overloading, 209
- method overriding, 211–212
- metrics (social network), 696–698
- Microsoft
 - ADO.NET, 468–471, 502, 533–534
 - AJAX, 508–509
 - ASP.NET, 506
 - LINQ, 483
 - ODBC, 466–467
 - OLE DB, 467–468, 502
 - purchase of LinkedIn, 629
- mini-dimension table, 563
- mirroring, 424

- misclassification rate, 685
 - missing values, 671–672
 - mixed file, 357
 - mixed fragmentation, 521–522
 - MMLib, 656–657
 - mobile DBMSs, 32
 - modulo (mod), 366–367
 - Moges, H.T., 82
 - MongoDB
 - complex queries and aggregations, 320–330
 - eventual consistency, 318–319
 - filters and queries, 316–318
 - items with keys, 316
 - Monsanto, 334
 - Morison, R., 712
 - multi-user database, 430
 - multicolumn index, 382–383, 403–407
 - multidimensional DBMS (MDBMS), 574–587
 - multidimensional OLAP (MOLAP), 574–587
 - multilevel indexes, 374–375, 384–385
 - multimedia data
 - applications, 3–4
 - technology for, 32
 - multimedia DBMSs, 32
 - Multiple Granularity Locking Protocol (MGL protocol), 451–452
 - multiple indexes, 403–407
 - multiplicities, 60, *See also* cardinalities
 - multiset, 157
 - multi-user system, 30
 - multi-valued attribute type
 - defined, 43
 - mapped to a relational model, 130–131
 - multi-valued dependency, 120
 - multi-version concurrency control, 541–542
 - MySQL (interactive environment), 147–148

 - named row type, 239
 - NameNode server, 631–634
 - namespace, 266–267
 - n-ary relationship, 129–130
 - NAS gateway, 418–419
 - natural key, 717
 - navigational query, 221
 - near real-time ETL, 598, *See also* ETL
 - Neo4j (graph-based database), 334, 658
 - nested queries, 172–175
 - nested-loop join, 409–410
 - Netflix, 593
 - network attach, 414
 - network attached storage (NAS), 417–418, 518
 - network DBMSs, 28
 - network sockets, 462
 - neural networks, 681–682
 - NewSQL, 343
 - nodes, 696
 - defined, 333
 - tree data structures, 386–389
 - nonlinear list, 375
 - nonrepeatable read, 442
 - non-volatile, 553–554
 - normalization
 - anomalies in unnormalized relational model, 113–114
 - defined, 111–112
 - functional dependencies, 114–115
 - informal guidelines, 114
 - prime attribute type, 115
 - normalization form tests
 - Boyce–Codd normal form, 119–120
 - first normal form, 115–117
 - fourth normal form, 120
 - second normal form, 117–118
 - third normal form, 118–119
 - NoSQL databases
 - and Big Data, 31
 - blended systems, 343
 - column-oriented databases, 331–332
 - and consistency, 301–302
 - data distribution/transaction management
 - BASE transactions, 540
 - CAP theorem, 539
 - horizontal fragmentation and consistent hashing, 538
 - multi-version concurrency control and vector clocks, 541–542
 - quorum-based consistency, 542–544
 - defined, 30
 - graph-based databases
 - and Hbase, 644
 - Cypher query language, 335–341
 - defined, 333–335
 - fragmentation, 520–521
 - inconsistency, 440
 - key-value stores 304
 - consistent hashing, 309–310
 - eventual consistency, 312–313
 - hash function, 304–305
 - horizontal scaling, 305–308
 - integrity constraints and querying, 314–315
 - replication and redundancy, 311–312
 - request coordination, 308–309
 - stabilization, 314
 - modern background, 302–304
 - movement emergence, 302
 - REST-based web services, 509–511
 - tuple and document stores
 - complex queries and aggregations, 320–330
 - defined, 315–316
 - filters and queries, 316–320
 - items with keys, 316
 - SQL interface, 330–331
- NOT EXISTS, 182
- NOT NULL, 151
- n-tier DBMS architecture, 31, 461
- Nutch project, 630
-
- object, 57
- object constraint language (OCL), 64–66
- Object Data Management Group (ODMG), 217
- object definition language (ODL), 218–221
- object equality, 217
- object identifier (OID), 216–217
- object identity, 217
- Object Management Group (OMG), 57
- object manipulation language (OML), 223
- object model, 217–218
- object persistence, 483–484
 - basic principles of, 214
 - serialization in, 214–215
- object query language (OQL), 221–223
- object-relational mapping (ORM), 226–227, 484, 501–502
- object storage, 420
- object-oriented database management systems (OODBMS), 208–227

- defined, 216
- evaluation of, 225–227
- identifiers, 216–217
- as NoSQL niche database, 303
- standard
 - defined, 217
 - language bindings, 223–225
 - object definition language, 218–221
 - object model, 217–218
 - object query language, 221–223
- object-oriented DBMS (OODBMS). *See* object-oriented database management systems
- object-oriented paradigm (OO)
 - advanced concepts of, 209–213
 - basic concepts of, 208–209
 - defined, 207
 - object persistence in, 214–215
- object-relational DBMS (ORDBMS)
 - behavior, 244
 - collection types, 245–247
 - defined, 29, 236
 - inheritance, 242–244
 - large objects, 247
 - polymorphism, 245
 - recursive SQL queries, 249–253
 - user-defined functions, 240–241
 - user-defined types, 236–240
- odds ratio, 676
- OLE DB, 467–468, 502
- ON DELETE CASCADE, 154
- ON UPDATE CASCADE, 154
- one-way linked list, 375–376
- on-line analytical processing (OLAP), 31
- on-line transaction processing DBMS (OLTP) 31
 - in business intelligence, 574
 - in decision-making, 552
 - operators, 575–577
 - SQL queries, 577–583
- on-premises analytics, 705–706
- opaque data type, 238
- open addressing, 369
- open API, 463
- Open Database Connectivity (ODBC), 464, 466–467
- open-source analytical software, 706–708
- open-source DBMS
 - defined, 32
 - and Hadoop, 630–631
 - Spark, 652
- operational BI, 592
- operational data store (ODS), 571
- operational efficiency, 689
- operational level, 552
- operations, 57
- optimistic protocol, 443–444, 534–536
- orchestration pattern, 604–606
- ORDER BY, 163–164
- ordinary least squares (OLS), 674
- outer joins (SQL DML), 170–172
- outer table, 409–410
- outliers, 672
- outrigger table, 561
- outsourcing, 664–704
- overfitting, 679
- overflow
 - and database access, 408
 - defined, 366
 - retrieval of, 368
- overflow area, 369
- overflow handling technique, 369–370
- overlap specialization, 53
- owner entity type, 46
- owner record type, 98
- page, 397
- parallel databases, 519
- parent–child relationship
 - defined, 94–95
 - foreign keys, 110
- Parquet, 332
- partial categorization, 55
- partial participation, 45
- partial shredding, 270–271
- partial specialization, 53
- participants, 528
- PARTITION BY, 582
- partition tolerance, 539
- partitions, 370
- passive, 232
- Pearson correlation coefficient, 687
- performance measures
 - classification models, 684–687
 - comprehensibility, 688
 - operational efficiency, 689
 - regression models, 687–688
- performance monitoring utilities
 - and CAP theorem, 539
 - defined, 27
 - in modern DBMSs, 397
- performance utilities
 - defined, 16
 - in modern DBMSs, 397
- persistence
 - by class, 214
 - by creation, 214
 - by inheritance, 214
 - by marking, 214
 - by reachability, 214
 - independence, 214
 - orthogonality, 214
- persistent object, 214
- persistent storage media, 352
- personal computer (PC), 459
- pessimistic protocol, 444, 533
- phantom reads, 442
- PHP, 505
- physical data independence, 12
- physical database, 357
- physical database design, 356–358
- physical database organization
 - business continuity, 421–425
 - database access methods, 400–408
 - disk arrays and RAID, 411–413
 - enterprise storage subsystems, 413–421
 - join implementations, 408–410
 - physical design architecture, 356–358
 - record organization, 359–361
 - records and files, 396–400
 - storage hierarchy, 352–353
- physical file, 357
- physical file organization
 - bitmap index, 383
 - hash index, 383
 - heap file, 363
 - indexed sequential file, 370–375
 - join index, 384
 - list data organization, 375–378
 - random file/hashing, 365–370
 - secondary indexes and inverted files, 379–384
 - sequential file, 363–365
 - terminology, 362–363
- Pig platform, 648–649
- pivot or cross-table, 573
- Platform as a Service (PaaS), 600
- pointers
 - block, 371
 - data, 386
 - defined, 357–358, 360
 - record, 371
 - tree, 386

- point-of-sale (POS)
 - application (POS), 552
 - defined, 4
 - as on-line transactional processing, 31
- polymorphism, 212, 245
- PostgreSQL, 330
- power curve, 686
- precedence graph, 443
- precision, 685
- predictive analytics
 - decision trees, 677–681
 - evaluating, 689–696
 - goal and types of, 673
 - linear regression, 673–675
 - logistic regression, 675–677
- PreparedStatement interface, 475
- Presto, 652
- primary area, 369
- primary copy 2PL, 529
- primary file organization methods, 362
- primary index
 - defined, 373, 398
 - range query, 404
- primary key
 - defined, 109
 - in file organization, 362
 - in index creation, 400
 - NoSQL databases, 316
- primary site 2PL, 529
- primary storage, 352
- prime attribute type, 115
- privacy (in analytics)
 - anonymization (data), 717–718
 - defined, 714
 - encryption, 721
 - internal data access, 717
 - LBAC, 719–721
 - RACI matrix, 715–716
 - regulations, 721–723
 - SQL views, 719
- privilege, 191–192
- procedural DML, 22
- process engine, 602
- process integration
 - and data integration, 606–610
 - defined, 591
- product notation, 404
- projection, 527
- proprietary API, 463–464
- proximity, 612
- qualified association, 61
- query and reporting, 573–587
- query by example (QBE), 573
- query cardinality (QC), 401
- query decomposition, 526
- query executor, 25
- query optimizer
 - in database access, 400–402
 - defined, 25
 - and index design, 398
- query parser, 25, 401
- query predicate, 401
- query processor
 - in database access, 401
 - defined, 22
 - in a distributed database, 525–528
 - DML compiler, 22–25
 - function of, 25
 - optimization of, 25
 - query executor, 25
- query rewriter, 24, 401
- QueryDSL, 483
- querying, 314–320
- quorum-based consistency, 542–544
- RACI matrix, 715–716
- RAID controller, 411
- RAID levels, 412–413
- RAIN (redundant array of independent nodes), 420
- random file organization
 - defined, 365
 - efficiency factors, 368–370
 - key-to-address transformation, 365–368
- ranking, 580
- ranking module, 613
- rapidly changing dimension, 563–565
- raw data, 6
- RDF Schema, 283, 341
- read committed, 450
- read lock, 26
- read uncommitted, 450
- read/write heads, 353
- receiver operating characteristic curve (ROC curve), 685
- recommender systems, 667
- record organization, 359–361
- record pointer, 371
- record type
 - CODASYL model, 98–101
 - defined, 94
 - variable length, 360–361
- record-at-a time DML, 22–23
- recovery, 431
- recovery facilities, 15
- recovery manager, 26
- recovery point objective (RPO), 422
- recovery time objective (RTO), 421–422
- recovery utilities, 27
- REDO, 436–438
- redundancy
 - and clustering, 311
 - defined, 95
 - in disk arrays, 412
- Redundant Array of Independent Disks (RAID), 411–413, 422
- referential integrity constraints
 - ALTER command, 155–156
 - DROP command, 155–156
 - SQL having, 154–155
- regression, 673
- regression tree
 - defined, 680–681
 - performance measures, 687–688
- relation, 105, *See also* set type
- relational DBMS (RDMS)
 - active extensions, 232–236
 - and SQL, 147–149
 - defined, 28, 147
 - relational model, 105–112
 - SQL, 147–149
- relational databases
 - basic concepts, 105–106
 - compared to NoSQL, 303
 - constraints, 111
 - examples of, 111–112
 - formal definitions, 106–108
 - history of, 104–105
 - mapping conceptual EER model to, 133–137
 - mapping conceptual ER model to, 121–133
 - mathematical underpinning, 105
 - normalization, 111–120
 - types of keys, 108–110
 - versus XML databases, 271–272
- relational model, 231–232
- Relational OLAP (ROLAP), 575
- relationship, 43
- relationship type. *See also* associations
 - with attribute type, 46
 - defined, 44
 - dependency, 66
 - in ER model, 43–46

- legacy databases, 94
 - mapped to a relational model, 122–132
- relative block address, 366
- relative location, 359
- remote procedure call (RPC), 284
- reorganization utility, 27
- repeatable read, 450
- repeated group, 98
- replicas, 311
- replication
 - and data availability, 424
 - and distributed 2PL, 529–530
 - in distributed databases, 520, 523–524
- replication transparency, 524
- representation category, 82
- representational state transfer (REST), 288–289, 509–511
- request coordinator, 308
- requirement collection and analysis, 38
- resilient distributed datasets (RDDs), 653–654
- resolution (deadlock), 449
- Resource Description Framework (RDF), 282–284
- response modeling, 667
- response time
 - defined, 354
 - as KPI, 16
- REST (representational state transfer), 288–289, 509–511
- RESTRICT, 154–155
- ResultsSets, 474, 476
- return on investment (ROI)
 - in analytics, 708–714
 - defined, 702–704
- right outer join, 170–172
- rigorous 2PL, 446
- ring topology, 309
- risk analytics, 667
- Roesch, D., 85
- roles, 44
- rollback, 432
- roll-down, 576
- rollforward recovery, 439
- roll-up, 575
- ROLLUP, 578–579
- rotational delay, 354
- row, *See* tuple
- sampling, 670
- SAX API (simple API for XML), 268
- scaling out, 301
- scaling up, 301
- schedule, 432, 442
- scheduler, 433, 439–440
- schema-aware mapping (XML), 275–276
- schema-level triggers, 234
- schema-oblivious mapping/shredding, 273–275
- Scheule, H., 85
- SCSI (Small Computer Systems Interface), 414–415
- search key, 362
- search key values, 386
- search tree, 385
- second normal form (2 NF), 117–118
- secondary file organization methods, 363
- secondary index, 363, 380–381, 398, 404
- secondary storage, 352
- sectors, 354
- security (in analytics)
 - anonymization (data), 717–718
 - defined, 714–715
 - encryption, 721
 - internal data access, 717
 - LBAC, 719–721
 - RACI matrix, 715–716
 - regulations, 721–723
 - SQL views, 719
- security manager, 22
- seek time, 354
- SELECT
 - correlated queries, 175–178
 - full syntax of, 156–157
 - GROUP BY/HAVING clause, 163–164
 - join queries
 - defined, 166
 - inner joins, 166–171
 - outer joins, 170–172
 - nested queries, 172–175
 - queries with aggregate functions, 161–163
 - queries with ALL/ANY, 178–181
 - queries with EXISTS, 181–182
 - queries with ORDER BY, 165
 - queries with set operators, 183–184
 - SELECT/FROM subqueries, 182–183
 - simple queries, 157–160
- selective inheritance, 55
- self-service BI, 573
- semantic metadata, 567
- semantic search, 282–284
- semantical rules
 - defined, 14
 - in UML, 59
- semi-structured data
 - defined, 13
 - and document stores, 315–318
- sensitivity, 685
- sensor DBMS, 32
- sequence rules, 691–692
- sequential file organization, 363–365
- sequentially accessible storage device (SASD), 353
- serial schedule, 442
- serial transactions, 431
- serializable, 442–443
- serializable level, 450
- serialization, 214–215
- server-free backups, 416–417
- service oriented architecture (SOA), 599
- service time, 354
- SET DEFAULT, 154
- SET NULL, 154
- set operators (SQL DML), 183–184
- set type, 97–101, *See also* relation
- set-at-a-time DML, 22–25, 147
- shallow equality, 217
- shard, 306
- sharding, 306, 538
- shared aggregation, 62–63
- shared and intention exclusive lock (six-lock), 451–452
- shared-disk architecture, 518
- shared lock, 445
- shared-memory architecture, 518
- shared-nothing architecture, 518
- short-term locks, 449
- shredding, 270, 273–275
- similarity measures, 612
- simple attribute type, 42
- Simple Object Access Protocol (SOAP), 285–288, 509–511
- simultaneous access, 30
- single points of failure, 422
- single-user system, 30
- single-valued attribute, 43
- singular value decomposition (SVD), 710
- slicing, 587
- slowly changing dimensions, 561–563

- Small Computer Systems Interface (SCSI), 414–415
- snowflake schema, 556
- SOAP (Simple Object Access Protocol), 285–288, 509–511
- Sober (fictional taxi company)
 - analytics, 723–724
 - background of, xxiii
 - Big Data, 660
 - data management, 88
 - data quality and governance, 622
 - data warehousing and business intelligence, 583–584
 - database access methods, 426
 - database architecture and categorization, 33
 - distributed transaction management, 545
 - EER model for, 67–68
 - file organization methods, 390
 - legacy databases, 102
 - mapping relational models, 138–139
 - NoSQL databases, 300
 - SQL, 194–195
 - transaction management, 453
 - UML model for, 69–70
 - XML, 294–296
 - web-based access, 512–513
- social network, 696
- social network analytics
 - definitions, 696–697
 - learning, 699–700
 - metrics, 696–698
 - popularity of, 695–696
- sociogram, 696–697
- Software as a Service (SaaS), 600
- solid state drive (SSD)
 - integrated circuitry of, 355–356
 - as secondary storage device, 352
- sort-merge join, 410
- sourced function, 241
- space utilization, 16
- Spark, 652
- Spark Core, 653–654
- Spark SQL, 654–656
- Spark Streaming, 657–658
- SPARQL, 284, 341
- sparse indexes, 371, 380
- spatial DBMS, 32
- specialization
 - defined, 52
 - mapping EER to relational, 133–136
 - in UML, 62
 - specialization hierarchy, 54
 - specificity, 685
 - spindle, 353
 - splitting decision, 677–679
 - splitting up the dataset techniques, 682–684
 - Spotify, 32, 463
 - SQL data types, 150–151
 - SQL injection, 477–479
 - SQL schema, 150
 - SQL table, 150–151
 - SQL views, 719
 - SQL/XML, 276–279
 - SQL/XML mapping, 276–279
 - SQLite, 480–482
 - SQLJ, 479–480
 - stabilization, 314
 - star schema, 555–556
 - starvation, 446
 - static 2PL, 446
 - static binding, 212
 - static SQL, 466
 - StAX (Streaming API for XML), 269
 - stopping decision, 679–680
 - storage area network (SAN), 416–417, 419, 518
 - storage devices, 352–353, 422
 - storage manager
 - and buffer manager, 26
 - defined, 25
 - and lock manager, 26
 - and recovery manager, 26
 - and transaction manager, 26
 - stored data manager, 434
 - stored database, 357
 - stored procedure, 234–236, 465
 - stored record, 357
 - stored table, 397
 - strategic level of decision-making, 552
 - streaming data, 593
 - stretched cluster, 425
 - strong entity type, 46
 - Strong, D.M., 82
 - structural metadata, 567
 - structured data
 - defined, 13
 - versus unstructured data, 613
 - structured literal, 218
 - Structured Query Language (SQL)
 - binding in, 465–466
 - data definition language, 149–156
 - data manipulation language, 156–188
 - for metadata management, 192–193
 - in index creation, 398–400
 - indexes, 190–191
 - and JPA, 494
 - key characteristics of, 147–148
 - in NoSQL queries, 330–331
 - OLAP queries, 577–583
 - and ORDBMSs, 249–253
 - popularity of, 148
 - privileges, 191–192
 - in relational databases, 147
 - search keys, 362
 - three-layer architecture, 149
 - views, 188–190
 - Structured Query Language (SQL) on Hadoop
 - background for using, 643–644
 - HBase, 644–648
 - Hive, 649–652
 - Pig, 648–649
 - Structured Query Language data definition language (SQL DDL)
 - example of, 151–154
 - key concepts of, 150–151
 - referential integrity constraints, 154–155
 - Structured Query Language data manipulation language (SQL DML)
 - DELETE statement, 185–186
 - INSERT statement, 185
 - purpose of, 156
 - SELECT statement, 156–184
 - UPDATE statement, 186–188
 - structured search, 280–282
 - subject-oriented, 553
 - SUM, 162
 - summation notation, 404
 - superkey, 108
 - supply chain management systems (SCM), 628
 - support, 690
 - support vector machines (SVM), 681–682
 - surrogate keys, 557–558
 - synonyms, 366
 - syntactical rules, 14
 - system database, 396
 - system failure, 436
 - system recovery, 436–438
 - table cardinality (TC), 401
 - table data type, 240

- table-based mapping, 272–273
- tablespace, 396–397
- tactical level, 552
- tape backup, 423
- technical key, 717
- template-based mapping, 278
- temporal constraints
 - of EER model, 56–57
 - of ER model, 51–52
 - relational database, 111
- ternary relationship types
 - defined, 48
 - in ER model, 48–50
- text analytics, 667
- text mining, 612
- textual data, 710
- thesaurus, 612
- third normal form (3 NF), 118–119
- thread, 22
- three-layer architecture
 - defined, 10
 - SQL having, 149
- three-schema architecture, 10
- three-tier architecture, 460
- throughput rate, 12–16
- tiered system architecture, 460–462
- tightly coupled
 - in distributed databases, 528
 - in Hadoop, 631
 - primary site 2PL and primary copy 2PL, 529
- time variant, 554
- timeliness, 82
- timestamping, 444, 533
- total categorization, 55
- total cost of ownership (TCO), 569, 702
- Total Data Quality Management (TDQM), 85–86, 619
- total participation, 45
- total specialization, 53
- tracks, 354
- transaction, 430–431
- transaction coordinator, 528
- transaction failure, 436
- transaction management
 - ACID properties of, 452–453
 - compensation-based models, 534–538
 - concurrency control, 528–534
 - DBMS components, 433–435
 - delineating, 432–433
 - distributed and concurrency control, 528–534
 - distributed query processing, 525–528
 - logfile, 435
 - transparency, 524–525
 - transaction manager, 26, 433
 - transaction recovery
 - defined, 425
 - failure types, 436
 - media recovery, 438–439
 - system recovery, 436–438
 - transaction transparency, 525
 - transfer time, 354
 - transient object, 214
 - transitive dependency, 118
 - transitive persistence, 214
 - transparency
 - defined, 517
 - in distributed databases, 524–525
 - in transaction management, 524–525
 - tree data structures
 - B⁺-tree, 388–389
 - binary search, 385–386
 - B-tree, 386–388
 - defined, 377–379
 - tree pointers, 386
 - trigger, 232–236
 - trivial functional dependency, 119
 - tuple
 - defined, 105
 - in relationships, 106–108
 - SQL DML, 161–164
 - tuple stores, 315
 - Twitter
 - API, 463
 - as Big Data, 628
 - as social network, 334
 - Two-Phase Commit Protocol (2PC), 530–532
 - Two-Phase Locking Protocol (2PL), 446–448, 529
 - two-tier architecture, 460
 - two-way linked list, 377
 - type orthogonality, 214
- unary relationship type, 44–45, 127–128
- uncommitted dependency problem, 440–441, 448, 537
- UNDO, 436–438
- unidirectional association, 60
- Unified Modeling Language (UML)
 - access modifiers, 59
 - aggregation in, 62–63
 - associations, 59–61
 - changeability property, 65
 - classes, 58
 - defined, 57
 - dependency relationships in, 66
 - examples of, 64
 - OCL in, 64–66
 - specialization and generalization in, 62
 - variables, 59
 - versus EER, 66
- uniform distribution, 368
- UNION, 183–184
- UNIQUE constraint
 - defined, 151
 - in index creation, 398
- unique index, 399
- universal API, 463–464, 525
- universal data access, 468
- universal data storage, 468
- unnamed row type, 238
- unrepeatable read, 442
- unstructured data
 - in analytics ROI, 710
 - defined, 13
 - versus structured data, 613
- UPDATE, 186–188, 474
- update anomaly, 113
- usage categorization, 31–32
- user database, 396
- user interface, 27
- user management utilities, 27
- user-defined functions (UDF), 240–241
- user-defined types (UDT)
 - defined, 236–237
 - distinct data type, 238–252
 - named row type, 239
 - opaque data type, 238
 - table data types, 240
 - unnamed row type, 238
- utilities, 26–27
- valid, 260
- value, 629
- value distortion, 717
- variable length records, 360
- variables, 59
- VARIANCE, 162
- variety, 628
- vector, 98
- vector clocks, 541–542
- velocity, 628
- veracity, 629
- vertical fragmentation, 520
- vertical scaling, 301
- vertices, 333

- victim selection, 449
- view
 - defined, 10
 - SQL, 188–190
- virtual child record type, 95
- virtual data mart, 569–570
- virtual data warehouse, 569–570
- virtual nodes, 311
- virtual parent record type, 95
- virtual parent–child relationship type, 95–97
- volatile data, 4
- volatile memory, 352
- volume, 628
- voting phase, 531

- wait-for graph, 449
- Wang, R.Y., 82, 88
- weak entity type
 - defined, 46
 - mapped to a relational model, 131–132
 - qualified associations for, 62
- web crawler
 - and Big Data, 630
 - defined, 613
- Web Ontology Language (OWL), 283
- web search engines, 613–616
- web services, 285–288
- Web Services Description Language (WSDL), 286
- WeChat, 628
- Weibo, 628

- well-formed, 259
- WHERE, 159, 221–223
- wide area network (WAN), 519
- windowing, 582
- workflow service, 607
- wrappers, 525, 569, 595
- write ahead log strategy, 435
- write lock, 26
- WS-BPEL, 602–604

- XML and XML DBMS
 - AJAX, 508–509
 - and JPA, 489
 - and relational databases, 271–272
 - as NoSQL niche database, 303
 - basic concepts of, 256–259
 - defined, 29
 - document storage, 269–271
 - document stores, 316
 - Document Type and Schema Definitions, 260–263
 - Extensible Stylesheet Language, 263–266
 - for information exchange
 - JavaScript Object Notation, 290–292
 - message-oriented middleware, 284–285
 - REST-based web services, 288–289
 - SOAP-based web services, 285–288

- web services and databases, 289–290
- YAML Ain't a Markup Language, 292–293
- mapping between object-relational databases
 - schema-aware, 275–276
 - schema-oblivious, 273–275
 - SQL/XML, 276–279
 - table-based, 272–273
- namespaces, 266–267
- processing documents, 267–269
- searching
 - full-text, 280
 - keyword-based, 280
 - semantic search with RDF and SPARQL, 282–284
 - structured search with XQuery, 280–282
- XML element, 256
- XML Schema Definition (XSD), 260
- XML-enabled DBMS, 271
- XPath, 256
- XQuery, 280–282
- XSL Formatting Objects (XSL-FO), 263–266
- XSL Transformations (XSLT), 263–266

- YAML Ain't a Markup Language, 292–293, 316
- YARN (Yet Another Resource Negotiator), 631, 641–643