



Index

Page numbers followed by “*f*” indicates figures and “*t*” indicates tables.

A

A* Search algorithm

- abstract state, 140
- applying constraints with, 138–142
- goal state, 138–140, 141

Abox, 328

Access-pattern limitations, 68, 80

- executable plans, generating, 81–84
- modeling, 81

Accuracy challenge, 95

Acyclicity constraints, 166

Adaptive methods, 426

Adaptive query process, 225–226

Affine gap measure, 100–102, 100*f*, 102*f*

Agglomerative hierarchical clustering (AHC), 180, 200

Algorithm decide completeness, 91

Analysis time vs. execution time, 225

Annotations, 436

- comments and discussions, 439–441
- data provenance, 360

Answering queries, using views, 43–44

- algorithms, comparison of, 57–58
- Bucket algorithms, 48–51
- closed-world assumption, 60–61
- interpreted predicates with, 61–62
- Inverse-rules algorithm, 56–57
- MiniCon algorithm, 51–55
- open-world assumption, 59–60
- problem of, 44–46
- relevant queries, 46–48

Archiving update logs, 444

Ashcraft, 109

Attribute correspondences, schema mappings, 351

Attribute names, mediated schema, 65–66

Attribute-level uncertainty, 347

Autonomous data sources, interfacing with, 223

Autonomy for data integration, 6

B

Backwards expansion, 403–404

Bag semantics, 41–43

Bayes’ rule, 190

Bayesian networks, 183–184, 190, 191*f* as generative model, 189–190

learning, 186–189

modeling feature correlations, 192–193, 192*f* representing and reasoning with, 184–185

Beam search, 151

BID model, *see* Block-independent-disjoint model

Bidirectional expansion, 404

Bidirectional mappings, 322

Bioinformatics, 441–442

BioSQL, 443

Bipartite graph, 116

Blank nodes in RDE, 338–339

Block-independent-disjoint (BID) model, 349–350

Blocking solution, 111

Blogosphere, 456

Boolean expression, 226

Boolean formulas, 35

Bound filtering, 116–117

Bucket algorithm, 48–51

Build large-scale structured Web databases, 454–455

By-table semantics, 353–354, 355*f*

By-tuple semantics, 354–356

C

Caching, 283–284, 457

Candidate networks, 404

- Candidate set, 155
 - Canonical database, 33
 - Canopies, 203
 - Cardinality constraint, 165–166
 - Cardinality estimation, 214
 - Cartesian product, 384
 - CDATA, *see* Character data
 - CDSS, *see* Collaborative data sharing system
 - Centralized DBMS, 217
 - Character data (CDATA), 297
 - Chase procedure, 446
 - Chase rule, 86–87
 - Classifier techniques, 133
 - Closed-world assumptions, 60–61
 - Cloud-based parallel process, 457
 - Cluster-based parallel process, 457
 - Clustering
 - collective matching based on, 200–201
 - data matching by, 180–182
 - Co-testing, 263
 - Collaboration in data integration
 - annotation
 - comments as, 439–441
 - mapping as, 438–439
 - challenges of, 435–436
 - corrections and feedback process, 436–437
 - user updates propagated, upstream/
 - downstream, 437–438
 - Collaborative data sharing system (CDSS)
 - data provenance, 447–448
 - peer in, 442
 - properties of, 441
 - reconciliation process, 449
 - trust policies, 448–449
 - update exchange process, 445–447
 - warehouse services, 441–442
 - Collective matching, 174, 198–200, 204
 - based on clustering, 200–201
 - entity mentions in documents, 201–202
 - Commercial relational databases, 23
 - CommitteeType, 299
 - Compatible data values, discovering, 408
 - Complete orderings, query refinements and,
 - 36–37
 - Complex query reformulation algorithms, 75
 - Composability, 29
 - Compose operator, 163
 - Composing scores, uncertainty, 347
 - Computational complexity, 356
 - Concordance table, 93
 - Condition variables, uncertainty, 347
 - Conditional probability table (CPT), 184*f*, 185,
 - 187, 188*f*, 191–193, 191*f*
 - Conjunctive queries, 26–28
 - interpreted predicates, 35–37
 - negation, 37–41
 - query containment of, 32–34
 - unions of, 34–35
 - Consistent target instance, 352–353
 - Constraint enforcer, 128, 135
 - Containment mapping, 32
 - interpreted predicates with, 35
 - Content-free element, 295
 - Conventional query processor, modules in,
 - 211*f*
 - Core universal solutions, 281–282
 - Corrective query processing (CQP), 232
 - cost-based reoptimization, 235–238
 - Cost-based backwards expansion, 404
 - Cost-based reoptimization, CQP, 235–238
 - Count queries, 42
 - CPT, *see* Conditional probability table
 - CQP, *see* Corrective query processing
 - Crowdsourcing, 454
 - Curation, scientific annotation and, 440
 - Cyclic mappings, 447
 - Cyclic PDMS, 420
- D**
- Data, 345
 - annotation, 436
 - cleaning, 453–454
 - creation/editing, 435
 - governance, 274
 - graph, 399–401
 - placement and shipment in DBMS, 217–218
 - profiling tools, 275
 - relationships
 - annotations on, 360
 - graph of, 361–362, 361*f*

- sources, 65, 67*f*, 68
- transformation modules, 275
- types, 390
- warehousing, 9, 11
 - definition, 272
 - design, 274
 - ETL, 275–276
 - MDM, 273–274
- Data exchanges, 272, 321
 - programs, 446
 - settings, 277–278
 - solutions, 278–279
 - core universal solutions, 281–282
 - materialized repository, 283
 - universal solutions, 279–281
- Data integration
 - architecture, 9, 10*f*
 - challenges of, 6
 - logical, 7–8
 - setting expectations, 9
 - social and administrative, 8–9
 - systems, 6–7
 - components of, 10–12
 - examples of, 1–5
 - goal of, 6
 - keyword search for, 407–410
 - modules in, 220*f*
- Data integration engine, 222
- Data lineage, *see* Data provenance
- Data matching, 174*f*
 - by clustering, 180–182
 - entity mentions in text, 193–198
 - learning based, 177–180
 - with Naive Bayes, 190
 - probabilistic approaches to, 182–183
 - Bayesian networks, 183–186
 - problem of, 173–174
 - rule-based, 175–177
 - scaling up, 203–205
- Data pedigree, *see* Data provenance
- Data provenance, 359, 447–448
 - annotations on, 360
 - applications of, 362–363
 - graph of relationships, 361–362, 361*f*
- Data-level heterogeneity, 92–93
- Data-level variations, 67
- Database concepts, review of
 - conjunctive queries, 26–28
 - data model, 22–23
 - datalog program, 28–29
 - integrity constraints, 24–25
 - queries and answer, 25–26
- Database instances, 23
- Database management system (DBMS)
 - parallel vs. distributed, 216–217
 - performance of, 209
 - query process, 210–211
 - control flow, 216
 - cost and cardinality estimation, 214
 - enumeration, 212–213
 - execution, 211–212
 - granularity of process, 214–216
 - interesting orders, 213
- Database reasoning vs. description logics, 333–334
- Database schemas, 22, 122*f*
- Database systems, queries, 25
- Datalog programs, 28–29
- Dataspace systems, 394–395
- DBMS, *see* Database management system
- De-duplication, 275
- Decision-support, 273
- Declarative warehousing, data exchange, 276–277
- Deep Web, 376–377, 379–380
 - surfacing, 383–385
 - vertical search engines, 380–383
- Dependent join operator, 224
- Description logics, 327–328
 - inference in, 331–333
 - semantics of, 329–331
 - syntax of, 328–329
 - vs. database reasoning, 333–334
- Desiderata, 65
- Distinguished variables, 26
- Distributed query process, 216–219
- Distributed vs. parallel DBMS, 216–217
- Document object model (DOM), 300–301
- Document root, 295

Document type definition (DTD), 296–298
 Dom relation, 84
 Domain integrity constraints, 135–137
 Domain ontology, 325
 Double pipelined join, *see* Pipelined hash join
 Dynamic content, *see* Deep Web
 Dynamic data, CDSS
 architecture, 443–444
 data provenance, 447–448
 peer in, 442
 properties of, 441
 reconciliation process, 449
 trust policies, 448–449
 update exchange process, 445–447
 warehouse services, 441–442
 Dynamic-programming algorithm, 97

E

Eddy
 lottery scheduling routing, 234–235
 queueing-based plan selection, 232–234
 Edges
 adjust weights on, 409–410
 directed, 399, 400
 Edit distance, 96–98, 97*f*, 98*f*
 Efficient reformulation, 70
 Enterprise information integration (EII), 283
 Equality-generating dependencies (EGDs), 24, 80, 277
 Eurocard database, 1–4, 3*f*, 7–8
 Event-condition-action rule framework, 226
 Event-driven adaptivity, 226
 handling source failures and delays, 227–228
 handling unexpected cardinalities, 228–231
 Evidence, combining, 408
 Executable plans, generating, 81–84
 Executable query plans, 81–82
 Execution time vs. analysis time, 225
 Existential variables, 26
 Expectation-maximization (EM) algorithm, 187, 188*f*, 197, 198, 205
 Explanation, provenance, 363

eXtensible Markup Language (XML), 292, 446
 document order, 295–296
 namespaces and qualified names, 294–295
 output, 317
 path matching, 313–316
 query capabilities for, 306–312
 query language
 DOM and SAX, 300–301
 XPath, 301–306
 XQuery, 306–312
 query processing for, 312–313
 schema mapping for
 nested mappings, query reformulation with, 321–322
 nesting, mappings with, 318–321
 structural and schema definitions
 DTD, 296–298
 XSD, 298–300
 tags, elements, and attributes, 293–294
 Extensional database (EDB) relations, 28
 External data, direct analysis of, 284–287
 Extract-transform-load (ETL)
 operations, 275–276
 tool, 11
 Extraction program, 246
 Extraction rules with Lixto, 267–269

F

Facebook, 456
 FindCands method, 110, 111
 FindMapping algorithm, 156
 Flat-file-based data analysis, 287
 FLWOR, 307–309
 Foreign key constraints, 24
 Fullserve company database, 1–4, 2*f*, 7–8
 Functional dependencies, 24

G

Gap penalty, 98, 101*f*
 GAV, *see* Global-as-View
 Generalized Jaccard measure, 106–108, 107*f*
 Generative model, 194–195, 194*f*, 201
 Bayesian networks as, 189–190
 learning, 196–198
 matching entity mentions, 195

Generic operators, 162
 GLAV, *see* Global-and-Local-as-View
 Global alignments, 102
 Global-and-Local-as-View (GLAV), 77–78
 mappings, 427, 428
 Global-as-View (GAV), 70–73, 415
 approach, 123
 mapping, 438
 with integrity constraints, 88–89
 Google Scholar, 454
 Google’s MapReduce programming paradigm, 284
 Granularity level, 66
 Graph expansion algorithms, 403–404
 Graph random-walk algorithms, 401
 Graphical user interface, 153
 Ground atom, 23

H

Handling limited access patterns, 224
 Hash-based exchange scheme, 217
 Hash-based operators for faster initial results, 223
 Hashes effect, 110
 Hashing, 203
 Head homomorphisms, 52
 Head variables, 26
 Head-left-right-tail (HLRT) wrappers, 249–250
 learning, 250–251
 Heterogeneity, 375
 semantic, 8
 type of, 382
 Higher-level similarity measure, 108
 HLRT, *see* Head-left-right-tail
 Homomorphism, 280
 Horizontal partitioning, 217
 HTML, *see* HyperText Markup Language
 Hybrid similarity measures
 generalized Jaccard measure, 106–108, 107*f*
 Monge-Elkan similarity measure, 109
 soft TF/IDF, 108–109, 108*f*
 HyperText Markup Language (HTML), 292
 data, 375
 tables, 376*f*

I

IDF measure, *see* Inverse document frequency measure
 Immediate consequent, provenance, 361
 Import filters, 275
 Incremental update propagation, 447
 Indexing, 203
 Information-gathering query operators, 229
 Informative inputs, 384
 Input attributes, 381
 Instance-based matchers, 132
 Integrated data, visualization, 456
 Integrity constraints, 22, 24–25, 78
 on mediated schema, 85–89
 Intensional database (IDB) relations, 28
 Interactive wrapper construction, 263
 creating extraction results with Lixto, 267–269
 identifying extraction results with poly, 264–267
 labeling of pages with stalker, 263–264
 Internet data, query execution for, 222
 Interpreted atoms, 27, 35
 Interpreted predicates, 30, 61–62
 Inverse document frequency (IDF) measure, 105–106, 105*f*
 Inverse mapping, 169
 Inverse rules, 79, 80, 86
 advantage of, 57
 algorithm, 56–57
 Invert operator, 164, 168–170
 Inverted index over strings, 111–112, 111*f*
 Iterative probing, 385
 Iterator model, 216

J

Jaccard measure, 104, 132
 Jaccard similarity measures, 200
 Jaro measure, 103
 Jaro-Winkler measure, 104
 Java model, 167

K

Key constraints, 24
 Keyword matching, 401–403

Keyword search

- for data integration, 407–410

- over structured data, 399–403

Knowledge representation (KR) systems,
325–327

L

LAV, *see* Local-as-View

Learning algorithm, 177

Learning techniques, 410

Learning-based wrapper construction, 249

Left outer join operator, 317

Levenshtein distance, 96

Lightweight integration, 455–456

Linearly weighted matching rules, 176

Lixto system, creating extraction rules with,
267–269

Local completeness, 89–90

Local contributions table, 447

Local data, direct analysis of, 284–287

Local rejections table, 447

Local-as-View (LAV), 73, 415

- approach, 123

- reformulation in, 75–76

- syntax and semantics, 74–75

- with integrity constraints, 85–87

Local-completeness constraint, 89–90

Logical query plan, 65, 68–70, 212*f*

Logistic regression matching rules, 175–176

Lottery scheduling scheme for routing,
234–235

M

Machine learning techniques, 409

Manual wrapper construction, 247–249

Many-to-many matches, 124, 150–152, 150*f*

Many-to-one matches, 124

Mappings, 163

- rule, 364, 365

MapReduce framework, 285

Margin-Infused Ranking Algorithm (MIRA),
410

Mashups, 388

Master data management (MDM), 273–274

Match combinations, 135, 144

- searching the space of, 137–143

Match operator, 161, 163

Match predictions, combining, 134

Match selector, 143–144

Matchers, 128–134

Materialized repository, 283

Materialized view, 25

Max queries, 43

MCD, *see* MiniCon description

m-estimate method, 187

Mediated schema, 11–13, 65, 67*f*, 133, 145, 346,
381, 413

- integrity constraints on

- GAV, 88–89

- LAV, 85–87

Mendota, 115

Merge operator, 161, 163–166

Message-passing systems, 162

Meta-learner, 146, 147, 149–150

Meta-meta-model, 168

Meta-model, 163

- translations between, 166

Metadata, 274, 395

Mid-query reoptimization, 228, 238

Middle-tier caching, 284

MiniCon algorithm, 51–55, 424

MiniCon description (MCD), 51, 424

- combining, 54–55

- definition, 52–54

Model management operators, 162–164, 162*f*

- developing goal of, 168

- use of generic set of, 161

Model management systems, 163, 170

ModelGen operator, 163, 166–168, 167*f*

Models, 163

Modern database optimizers, 212

Monge-Elkan similarity measure, 109

Multi-set semantics, 23

Multi-strategy learning, 146

N

Naive Bayes

- assumption, 190

- classification technique, 134

- data matching with, 190

- learner, 148–149

- Name-based matchers, 130–132
- Namespaces, 294–295
- Needleman-Wunch measure, 98–100, 99f
- Negative log likelihood, 367
- Nested mappings, query reformulation with, 321–322
- Nested tuple schemas, 251–252
- Nested tuple-generating dependency (Nested tgds), 320–321
- Nodes, 400
 - adjust weights on, 409–410
- O**
- Object-oriented database schemas vs. description logics, 334
- ObjectRank, 401
- OLAP, *see* Online analytic processing queries
- One-to-many matches, 123
- One-to-one matches, 123, 127
- Online analytic processing (OLAP) queries, 273
- Online learning, 409
- Open DataBase Connectivity (ODBC) wrapper, 223
- Open-world assumption, 59–60
- Optimizer, runtime reinvocation of, 231
- ORCHESTRA system, 366, 366f
- Output attributes, 381
- Overlap similarity measure, 104, 113
- OWL, *see* Web Ontology Language
- P**
- P-mappings, *see* Probabilistic mappings
- PageRank, 401
- Parallel vs. distributed DBMS, 216–217
- Pay-as-you-go
 - data integration, 456
 - data management, 394–395
- Pc-table, *see* Probabilistic conditional table
- Peer data management systems (PDMSs), 413
 - complexity of query answering in, 419–421
 - for coordinating emergency response, 415
 - data instance for, 418, 419
 - with looser mappings
 - mapping table, 430–432
 - similarity-based mappings, 429–430
 - mapping composition, 426–429
 - peer mappings, 414, 417–418
 - query reformulation algorithm, 421–426
 - query to, 415
 - reformulation construction, 426
 - rule-goal tree for, 422f, 424f, 425
 - semantics of mappings in, 418–419
 - storage descriptions, 414, 417
 - structure of, 414
- Peer mappings, 413, 414, 421
 - compositions of, 426, 429
 - definitional, 417, 422, 422f
 - inclusion and equality, 417
 - interpreted predicates in, 421
- Peer relations, 413–415, 417, 418
- Peer schema, 414, 414f, 415
- Performance-driven adaptivity, 231–232
- Phonetic similarity measures, 109–110
- Physical database, 9
 - design, 274
- Physical query plan for data integration, 223
- Physical-level query operators, 217
- Piazza-XML mappings language, 318–319
- Pipelined hash join, 222–223, 224f
- Position filtering, 115
- Prefix filtering, 113–115, 113f
- Probabilistic conditional table (Pc-table), 348–349
- Probabilistic data representations
 - BID model, 349–350
 - c-table, 348
 - tuple-independent model, 349
- Probabilistic generative model, 201
- Probabilistic mappings (P-mappings), 350, 352
 - semantics of, 352–353
 - semi-automatic schema mapping tool, 351
- Probabilistic matching method, 204, 205
- Probability
 - distribution, 183, 183f
 - of perturbation types, 196, 197
 - smoothing of, 187
 - theory, 183
- Procedural code, 273
- Processing instruction, 293, 295
- Prolog programming language, 430

Provenance, 453–454

- annotations on data, 360
- data, applications of, 362–363
- graph of data relationships, 361–362, 361*f*
- semiring formal model, 364–365
 - applications of, 366–368
- storing, 368–369
- token, 362, 362*f*, 364
- trust policies and, 448–449

pSQL, 440

Publishing update logs, 444

Q

- Qualified names, 294–295
- Quasi-inverses of mapping, 169–170
- Queries, 346
- Query annotations, 318
- Query answer-based feedback, 401
- Query answering inference in description
 - logics, 332–333
- Query capabilities and limited data statistics,
 - 209–210
- Query containment, conjunctive queries,
 - 32–34
- Query equivalence, 31
- Query execution, 228
 - engine, 211, 214
 - for Internet data, 222
 - selection of, 211–212
- Query optimization, 211
- Query optimizer, 211
- Query plans, generating initial, 221–222
- Query process, 66*f*
 - adaptive, 225–226
 - for data integration, 219–221
 - DBMS, *see* Database management system (DBMS), query process
 - execution, 14–15
 - optimization, 13–14
 - reformulation, 13
- Query refinements, 36–37
- Query rewrite stage, 211
- Query tree, score as sum of weights in, 402–403
- Query unfolding, 29–30
 - stage, 211

R

- RDF, *see* Resource Description Framework
- RDFS, *see* Resource Description Framework Schema
- Real-world data matching systems, 177
- Reconciliation process, CDSS, 449
- Recurrence equation
 - for affine gap measure, 100*f*, 101
 - for Needleman-Wunch score, 99, 99*f*
- Recursive query plan, 83–84, 86
- Reformulation
 - GAV, 71–72
 - GLAV, 77–78
 - LAV, 75–76
- Rehash operation, 217
- Reification, RDF, 339–340
- Relation names, mediated schema, 65–66
- Relational schema, 22
- Reoptimization
 - mid-query, 228, 238
 - predetermined, 229–230
- Resolving cycle constraints, 166
- Resource Description Framework (RDF),
 - 335–337
 - blank objects in, 338–339
 - literals in, 338
 - query of, 342–343
 - reification, 339–340
- Resource Description Framework Schema (RDFS), 335, 340–341
- Rewriting queries, length of, 47–48
- Root element, 293
- Root-leaf costs, score as sum of, 403
- Rule-based learner, 147–148
- Rule-based matching, 175–177
 - scaling up, 203–204
- Runtime re-invocation of optimizer, 231

S

- Sarbanes-Oxley Act, 274
- SAX, 300–301
- Scalability challenge, 96
- Scalable automatic edge inference, 407–408
- Scalable query answering, 409
- Scale, 375

- Schema, 125
 - combined similarity matrix for, 138*t*
 - data instances of, 132
 - with integrity constraints, 137*f*
 - node, 142
 - propagating constraints, 142
 - standards of, 126
 - tree representation of, 143*f*
- Schema mappings, 11, 65–68, 121, 124, 129, 168, 345, 351, 442
 - challenges of, 124–127
 - composing, 426
 - formalisms, 92
 - languages
 - GAV, 70–73
 - GLAV, 77–78
 - LAV, 73–77
 - logical query plan, 68
 - principles, 69–70
 - tuple-generating dependencies, 78–80
- matches into, 152
- space of possible, 153, 154, 156–158
- uncertainty
 - by-table semantics, 353–354, 355*f*
 - by-tuple semantics, 354–356
 - p-mappings, 350–353
- Schema matching, 121, 124, 127–129
 - challenges of, 124–127
 - components of, 128
 - learners for, 147–150
 - learning techniques, 145
- Scientific data sharing setting, 440–441
- Score components, 409
- Score matrix, 98, 99*f*
- Scoring
 - models, 401–403, 410
 - provenance, 363
- Select-project-join (SPJ) expression, 211, 212
- Semantics
 - compatibility, considering, 408
 - cues, 375–376
 - GAV, 71
 - GLAV, 77
 - heterogeneity, 8, 67
 - reconciling, 125
 - LAV, 74–75
 - mappings, 11, 122–123
 - matches, 123–124
 - schema mappings, 69
 - Web, 325, 335
- Semi-supervised learning, 409, 456
- Semiautomatic techniques, 345
- Semiring formal model, 364–365
 - applications of, 366–368
- Sensors, 453
- Sequence-based similarity measures
 - affine gap measure, 100–102, 100*f*, 102*f*
 - edit distance, 96–98, 97*f*, 98*f*
 - Jaro measure, 103
 - Jaro-Winkler measure, 104
 - Needleman-Wunch measure, 98–100, 99*f*
 - Smith-Waterman measure, 102–103, 103*f*
- Sequential covering, 255
- Set-based similarity measures
 - Jaccard measure, 104
 - overlap measure, 104
 - TF/IDF measure, 105–106, 105*f*
- SGML, *see* Structured Generalized Markup Language
- Similarity measures
 - hybrid
 - generalized Jaccard measure, 106–108, 107*f*
 - Monge-Elkan similarity measure, 109
 - soft TF/IDF, 108–109, 108*f*
 - phonetic, 109–110
 - sequence-based
 - affine gap measure, 100–102, 100*f*, 102*f*
 - edit distance, 96–98, 97*f*, 98*f*
 - Jaro measure, 103
 - Jaro-Winkler measure, 104
 - Needleman-Wunch measure, 98–100, 99*f*
 - Smith-Waterman measure, 102–103, 103*f*
 - set-based
 - Jaccard measure, 104
 - TF/IDF measure, 105–106, 105*f*
- Simple delete-insert update model, 449–450
- Single-database context, 401
- Size filtering, 112
- Skolem function, 80

Skolem terms, 56
 Skolem values, 446
 Smith-Waterman measure, 102–103, 103*f*
 Social media, integration of, 456
 Soft TF/IDF similarity measure, 108–109, 108*f*
 Softened overlap set, 107
 Sorting, 203
 Soundex code, 109
 Source descriptions, vertical-search engine, 382
 SparQL language, 342–343
 SPJ expression, *see* Select-project-join expression
 Spreading activation, 404
 SQL queries, 25, 158
 STAIRs, 235
 Stalker extraction rules, 254
 Stalker wrappers, 251–252
 learning, 254–256
 model, 252–253, 256
 Standard data integration applications, 388
 State modules (STeMs), 235
 Statistics collection operators, 229
 Steiner tree algorithms, 402, 403
 STeMs, *see* State modules
 Stitch-up plans, creating, 238–240
 Storing provenance, 368–369
 Streaming XPath evaluation, 312
 String matching
 problem description of, 95–96
 scaling up
 blocking solution, 111
 bound filtering, 116–117
 inverted index over strings, 111–112, 111*f*
 position filtering, 115
 prefix filtering, 113–115, 113*f*
 size filtering, 112
 techniques, 117
 similarity measures
 hybrid, 106–109
 phonetic, 109–110
 sequence-based, 96–104
 set-based, 104–106
 Structured data, keyword search, 399–403

Structured Generalized Markup Language (SGML), 292
 Structured queries, 25
 Sub-instances, 282
 Suboperators for eddy, 233
 Subsumption inference in description logics, 331–332
 Super-model, 168
 Support vector machines (SVM), 178
 Surfacing, 383–385

T

TA, *see* Threshold Algorithm
 Tabular organization, 66
 Target data instance, 276
 Tbox, 328
 Term frequency (TF) measure, 105–106, 105*f*
 Text content, 294
 TF measure, *see* Term frequency measure
 Threshold Algorithm (TA), 404, 405
 Threshold value, 405
 Threshold-based merging, 404–407
 Top-*k* query processing, 404
 Topical portals, 378, 385–388
 Training data, 177
 Transactions, challenges of, 449–450
 Transformations, 92
 modelGen performing, 167
 Transient data integration tasks, 378
 Trust policies, 448–449
 Tuple router, eddy, 233, 234
 Tuple-generating dependencies (tgds), 24, 78–80, 277
 Tuple-independent model, 349
 Tuple-level uncertainty, 347
 Tuples, 23
 Twittersphere, 456
 Two-way Bloomjoin operator, 218–219
 Two-way semijoin operator, 218–219

U

Umbrella set, 111
 Uncertainty, 453–454
 and data provenance, 356
 possible worlds, 346–347

- probabilistic data representations, 348–350
- to probabilities, 350
- schema mappings, 351
 - by-table semantics, 353–354, 355*f*
 - by-tuple semantics, 354–356
 - p-mappings, 350–353
- types of, 347
- Uniform resource indicator (URI), 294, 338
- Universal solutions, 279–281
- Unstructured queries, 25
- Update exchange process, 445–447
- URI, *see* Uniform resource indicator
- User-supervised techniques, 392

V

- Variable mappings, 32
- Variable network connectivity and source
 - performance, 210
- Vertical partitioning, 217
- Vertical-search engines, 378, 385
- Virtual data integration, 9, 10
- Virtual integration system, caching, 284

W

- Web data, 377–379
 - lightweight combination of
 - data types, 390
 - data, importing, 391–393
 - mashups, 388
 - multiple data sets, combining, 393
 - structured data, discovering, 391
- Web end user information sharing, 440
- Web Ontology Language (OWL), 335, 341–342
- Web search, 378–379
- Web Service Description Language (WSDL), 300
- Web sites with databases of jobs, 4–5, 5*f*
- Web-based applications, 284
- Web-oriented data integration systems, 225
- Weighted-sum combiners, 134

- Wikipedia, 455
- World-Wide Web, 375
- Wrappers
 - construction
 - categories of solutions, 246–247
 - challenges of, 245–246
 - interactive, *see* Interactive wrapper
 - construction
 - learning-based, 249
 - manual, 247–249
 - problem, 244
 - generation tools, 162
 - HLRT, 249–250
 - learning, 250–251
 - learning, 245
 - inferring schema, 258–263
 - modeling schema, 257–258
 - without schema, 256–257
 - operator, 223–224
 - program, 10
 - stalker, *see* Stalker wrappers
 - task of, 243
 - vertical-search engine, 382–383

X

- XML, *see* eXtensible Markup Language
- XML Schema (XSD), 298–300, 299*f*
- XML Stylesheet Language Transformations (XSLT), 300
- XML wrapper, 223
- XPath language, 301–306
- XQuery, 306–312
 - optimization, 317
 - queries, 25
- XSD, *see* XML Schema
- XSLT, *see* XML Stylesheet Language Transformations

Z

- Zipcode, 165