

Index

NUMBERS

- 1NF. See First Normal Form (1NF)**
- 2NF. See Second Normal Form (2NF)**
- 3NF. See Third Normal Form (3NF)**
- 4NF. See Fourth Normal Form (4NF)**
- 5NF. See Fifth Normal Form (5NF)**

A

- Access. See Microsoft Access**
- access control**
 - database security and, 17
 - security planning and, 68
- ACID (Atomicity, Consistency, Isolation, Durability), 13–14**
- alternate (secondary) keys, relational databases, 53–54**
- American National Standards Institute (ANSI), 347**
- anomalies**
 - 2NF and, 147
 - 3NF and, 151
 - 4NF and, 159
 - BCNF and, 155
 - normalization and, 138
 - redundancy and, 167
- ANSI (American National Standards Institute), 347**
- applications**
 - developing for databases. *See* software development
 - multi-tier. *See* multi-tier applications
 - speed of interactive, 235
- association objects, semantic views, 102–105**
- association patterns**
 - many-to-many, 186–188
 - multiple-object, 188–191
 - one-to-many reflexive, 194–195
 - one-to-one reflexive, 193–194
 - overview of, 185–186
 - reflexive, 193
 - repeated attribute, 191–192
- association tables**
 - many-to-many relationships and, 186

- multiple-object associations and, 189
- types of tables, 176
- atomic objects, 100–101**
- atomic transactions, 8, 12–13**
- Atomicity, Consistency, Isolation, Durability (ACID), 13–14**
- attributes**
 - classes sharing in common, 249
 - database columns, 52
 - ERD, 107
 - relational databases, 52
 - repeated attribute associations, 191–192
 - semantic objects, 96
 - XML, 35
- audit trails**
 - logging and, 203–204
 - security needs and, 236
 - security planning and, 69

B

- backups, 379–382**
 - design goals and, 14–15
 - offsite, 381
 - overview of, 379–381
 - planning, 15, 381–382
- basic initialization scripts, 370**
- BCNF. See Boyce-Codd Normal Form (BCNF)**
- Boyce-Codd Normal Form (BCNF), 154–158**
 - arranging data in, 157
 - how it works, 157–158
 - qualifications/rules, 154
- brainstorming sessions**
 - customer/user needs and, 75
 - Pampered Pet case study, 229–230
- bugs, testing and, 214**
- business logic, multi-tier applications, 130**
- business rules, 121–136**
 - exercise solutions, 424–430
 - exercises, 134–136
 - extracting key, 128–129
 - identifying key, 123–128
 - list for custom woodworking shop, 126
 - multi-tier applications, 129–133
 - overview of, 121–123
 - summary, 133–134

business rules (Pampered Pet case study)

business rules (Pampered Pet case study), 263–271

Courses table, 264–266
 CustomerCourses table, 266–267
 Customers table, 267
 Employees table, 267
 exercise solutions, 449–451
 exercises, 271
 identifying, 263–264
 InventoryItems table, 268
 OrderItems table, 268
 Orders table, 267
 overview of, 263
 Persons table, 269
 Pets table, 267
 Phones table, 269
 relational model for, 270
 Shifts table, 269
 summary, 270–271
 TimeEntries table, 268–269
 Vendors table, 269–270

C

calculations

ability of databases to perform complex, 19
 business rules for parameters of, 124, 128

candidate keys

BCNF and, 154–156
 relational databases, 53–54

cardinality

enforcing referential integrity and, 295
 ERD and, 108–109
 semantic object attributes, 97

Cartesian product, relational database operations, 58

case sensitivity, SQL, 347

catchall tables, as design pitfall, 215–217

cells, in dimensional databases, 44

change

managing, 264
 not planning for, 210–212

check constraints

allowing for changes to business rules, 123
 business rules for, 122, 124
 business rules for complex checks, 125
 overview of, 55
 sanity checks and, 264
 validating fields against domain, 175

classes

building ERDs and, 250–252
 entity classes, 107
 object inheritance and, 105–106
 semantic classes, 96

semantic views, 100
 sharing common attributes, 249

cleanup scripts, 371

columns

1NF and, 139–146
 attributes or data elements, 52
 multiple values in, 276
 relational databases, 50–52

Command Line Client, MySQL, 314–318

commands, SQL

FROM clause, 359–362
 CREATE INDEX command, 355–356
 CREATE TABLE command, 350–355
 DELETE command, 365–366
 DROP command, 356
 GROUP BY clause, 362–363
 INSERT command, 357–358
 ORDER BY clause, 363–364
 overview of, 348–350
 SELECT clause, 358–359
 SELECT command, 358
 UPDATE command, 364–365
 WHERE clause, 362

comments and notes

ERDs, 113
 SOMs, 106

commit, atomic transactions, 12

compacting databases, 383

composite objects, semantic views, 101

compound (composite) keys, relational databases, 53–54

compound objects, semantic views, 101–102

configuration information

INI files, 26
 Windows system registry, 26–27

consistency

database design goals, 10
 definition of, 13
 naming conventions and, 209

constraints, 55–57

Addresses table (Pampered Pet case study), 265–266
 business rules as, 122–123
 check constraints, 55
 Courses table (Pampered Pet case study), 264–265
 CustomerCourses table (Pampered Pet case study), 267
 data integrity and, 68, 236–238
 DKNF and, 165–166
 ERDs and, 112
 field constraints in Access, 296–298
 foreign key constraints, 56–57
 generalized, 124
 not enforcing as design pitfall, 217

- overview of, 55
 - primary key constraints, 55
 - table constraints in Access, 298–300
 - unique constraints, 56
 - contextual lists, 89. See also requirements document**
 - conventions, ERD, 112–113**
 - corrupted databases, repairing, 383**
 - cost savings, database design goals, 16**
 - Courses table (Pampered Pet case study), 264–266**
 - Create, Read, Update, and Delete. See CRUD (Create, Read, Update, and Delete)**
 - CREATE DATABASE statement, 322**
 - CREATE INDEX statement, 355–356**
 - Create operations**
 - CRUD and, 9
 - speed and, 11
 - validity, 10–11
 - CREATE TABLE statement, 322**
 - creating table with, 352–355
 - MySQL, 317
 - overview of, 350–352
 - SQL, 350–355
 - CREATE TRIGGER statement, 335**
 - creative people, 75**
 - CRUD (Create, Read, Update, and Delete)**
 - atomic transactions, 11–12
 - consistency, 10
 - database operations, 9
 - error correction, 11
 - retrieval, 9–10
 - speed, 11–12
 - SQL and, 349
 - validity, 10–11
 - customer champion**
 - brainstorming and, 75
 - who's who, 71
 - customer representative, 71**
 - CustomerCourses table (Pampered Pet case study), 266–267**
 - Customers table (Pampered Pet case study), 267**
 - customer/user requirements, 65–88**
 - brainstorming sessions, 75
 - creating list of questions for, 67
 - data integrity and, 68
 - data requirements, 67–68
 - environmental requirements, 69
 - exercise solutions, 413–416
 - exercises, 86–88
 - experiencing day-to-day operations, 73–74
 - feasibility evaluation, 85
 - functional requirements, 67
 - future needs, 76
 - meeting with customer(s), 69–70
 - overview of, 65–66
 - picking customers' brains, 73
 - planning process, 66–67
 - prioritization, 78–79
 - requirements document, 80–81
 - security and, 68–69
 - studying day-to-day operations, 74–75
 - summary, 85–86
 - understanding customers' real needs, 77–78
 - understanding customers' reasoning, 76–77
 - use cases, 81–85
 - verifying approach, 79–80
 - who's who and, 70–73
 - customer/user requirements (Pampered Pet case study)**
 - data integrity needs, 236–238
 - exercise solutions, 442–446
 - exercises, 244
 - functional requirements, 229–231
 - getting feedback, 243–244
 - meeting with customer(s), 227–229
 - overview of, 227
 - performance needs, 235
 - picking customers' brains, 229
 - project look, 231
 - relationships between data, 233–235
 - security needs, 235
 - sources of data, 233
 - summary, 244
 - user interface, 232–233
 - writing requirements document, 242–243
 - writing use cases, 238–242
- D**
- data**
 - allowing for redundant, 179–180
 - determining sources of (Pampered Pet case study), 233
 - editing in MySQL, 320–321
 - organizing hierarchically, 32
 - data (database) tier, multi-tier applications, 129**
 - Data Control Language (DCL), 348–349**
 - Data Definition Language (DDL), 348–349**
 - data elements, 52**
 - data initialization scripts, 370–371**
 - data integrity**
 - customer/user needs, 68
 - customer/user needs (Pampered Pet case study), 236–238
 - Data Manipulation Language (DML), 348–349**
 - data models**
 - entity-relationship. *See* ERDs (entity-relationship diagrams)

data models (*continued*)

data models (*continued*)

exercise solutions, 417–424
 exercises, 118–119
 normalization and, 249
 overview of, 89–92
 relational. *See* relational models
 semantic object. *See* SOMs (semantic object models)
 semantic views, 99–100
 summary, 117–118
 user interface, 92–96
 well-focused tables and, 176–177

data models (Pampered Pet case study)

building combined ERD, 252–254
 building ERD, 250–252
 building initial SOM, 245–248
 ERDs, 250
 exercise solutions, 446–448
 exercises, 261–262
 improving ERD, 254–256
 improving SOM, 248–250
 integrating all models, 259–260
 overview of, 245
 relational modeling, 256–259
 SOMs, 245
 summary, 260–261

data requirements

converting requirements tables into semantic objects, 246–248
 customer/user needs, 67–68

data types

column, 50–51
 requirements, 125
 SQL, 351

data warehousing, 382–383

database creation scripts, 370

database drivers, Access and, 287

database maintenance. *See* maintenance

database security. *See* security

Database Tool tab, Access, 293

database types, 23–47

benefits of different types, 24
 deductive, 44
 dimensional, 44
 document-oriented, 44
 exercise solutions, 408–411
 exercises, 47
 flat files, 24–27
 hierarchical, 31–34
 networks, 40–42
 object, 42–43
 object-relational, 43
 overview of, 23–24
 relational, 27–30
 spreadsheets, 30–31

summary, 45–46

temporal, 45

XML, 34–40

databases

business rules for, 122

definition of, 5

as information containers, 5

Datasheet views, Access, 289

day-to-day operations

experiencing, 73–74

studying, 74–75

DCL (Data Control Language), 348–349

DDL (Data Definition Language), 348–349

deductive databases, 44

Delete operations

CRUD and, 9

speed and, 11

validity and, 10–11

DELETE statement, SQL, 365–366

deleted objects, temporal data, 202

deletion anomalies

2NF and, 147

3NF and, 151

BCNF and, 155

deliverables, requirements document, 80–81

dependencies

5NF and, 162–163

DKNF and, 166

design

consequences of good and bad designs, 9–10

importance of, 4–5

design patterns

associations, 185–186

exercise solutions, 436–440

exercises, 205–206

hierarchical data and, 195–197

logging and locking, 203–205

many-to-many associations, 186–188

multiple-object associations, 188–191

network data and, 198–200

one-to-many reflexive associations, 194–195

one-to-one reflexive associations, 193–194

overview of, 185

reflexive associations, 193

repeated attribute associations, 191–192

summary, 205

temporal data, 200–203

design pitfalls

exercise solutions, 440–442

exercises, 222–223

insufficient normalization, 213–214

insufficient testing, 214

lack of preparation, 207–208

mishmash tables, 215–217

not defining natural keys, 220–221

field-level check constraints

- not enforcing constraints, 217
 - not planning for change, 210–212
 - overuse of IDs, 218–220
 - overview of, 207
 - performance anxiety, 214–215
 - poor documentation, 208
 - poor naming standards, 208–209
 - summary, 221
 - thinking too small, 210
 - too much normalization, 212–213
 - Design view, Access, 289**
 - determinants, BCNF and, 154–156**
 - development environment, Access, 288**
 - devil's advocate, who's who and, 72**
 - difference operation, relational databases, 58**
 - dimensional (multi-dimensional) databases, 44**
 - divide operation, relational databases, 58**
 - DKNF. See Domain/Key Normal Form (DKNF)**
 - DML (Data Manipulation Language), 348–349**
 - documentation**
 - of existing system in planning projects or applications, 69
 - importance of documenting everything, 174
 - naming conventions and, 208
 - poor or insufficient, 208
 - document-oriented databases, 44**
 - domain constraints, 55**
 - Domain/Key Normal Form (DKNF), 165–167**
 - arranging data in, 166
 - qualifications/rules, 165
 - domains**
 - columns and, 50–51
 - converting into tables, 175–176
 - data integrity and, 68
 - sanity checks, 263
 - DROP DATABASE statement, 318**
 - DROP statement, 356**
 - DROP TABLE statement, 317**
 - durability, ACID, 14**
- E**
- ease of use, database design goals, 16**
 - EER diagrams, creating with MySQL Workbench, 331–333**
 - effective/valid dates, temporal data, 200–202**
 - elements, XML nesting rules, 35**
 - Employees table, (Pampered Pet case study), 267**
 - encryption**
 - right level of security and, 389
 - security planning and, 68
 - end users, modeling and, 91**
- entities**
- 1NF and, 140
 - building ERDs, 250–252
 - classes, 107
 - converting requirements tables into semantic objects, 246–248
 - ERDs and, 107
 - primary key constraints for integrity of, 55
- entity sets, 107**
- enumerated types, with fixed values, 125**
- enumerated values, types of business rules, 124**
- environment, customer/user needs, 69**
- ERDs (entity-relationship diagrams), 106–113**
- building combined (Pampered Pet case study), 252–254
 - building initial (Pampered Pet case study), 250–252
 - cardinality, 108–109
 - comments and notes, 113
 - conventions for, 112–113
 - converting to relational models, 116–117
 - entities, attributes, and identifiers, 107
 - improving (Pampered Pet case study), 254–256
 - inheritance, 109–110
 - integrating all models (Pampered Pet case study), 259–260
 - overview of, 106–107
 - Pampered Pet case study, 250
 - relationships, 107–108
 - working with, 111
- errors**
- avoiding vs. fixing, 173–174
 - database design goals and, 11
- executive champion, who's who and, 71**
- executive overview, in requirements document, 242–243**
- extensibility, database design goals, 16**
- eXtensible Markup Language (XML). See XML (eXtensible Markup Language)**
- F**
- feasibility evaluation, customer/user needs, 85**
 - feedback, development process and, 243–244**
 - field-level check constraints**
 - business rules and, 122
 - Courses table (Pampered Pet case study), 264–265
 - creating in Access, 296–298
 - Employees table (Pampered Pet case study), 267
 - not enforcing as design pitfall, 217
 - overview of, 55
 - Phones table (Pampered Pet case study), 269

field-level check constraints (*continued*)

field-level check constraints (*continued*)

- Shifts table (Pampered Pet case study), 268–269
- TimeEntries table (Pampered Pet case study), 268–269
- validating fields with, 175

fields

- 1NF and, 140
- breaking into pieces to handle multiple values, 276
- data integrity and, 68
- documentation, 174
- naming conventions, 209
- relational databases, 52
- sanity checks, 263–264

Fifth Normal Form (5NF), 161–165

- how it works, 164–165
- qualifications/rules, 161
- working with, 163–164

file system, as hierarchical database, 31–32

final deliverables, 81

First Normal Form (1NF), 138–146

- arranging data in, 141–142
- databases vs. alternative methods of data storage, 181
- how it works, 142–146
- qualifications/rules, 138–141
- verifying (Pampered Pet case study), 275–278

flat files, 24–27

- INI files, 26
- overview of, 24–25
- pros/cons, 25
- Windows system registry, 26–27

foreign key constraints

- allowing for changes to business rules, 123
- business rules and, 122
- converting domains into tables, 175
- CustomerCourses table (Pampered Pet case study), 267
- InventoryItems table (Pampered Pet case study), 268
- OrderItems table (Pampered Pet case study), 268
- Orders table (Pampered Pet case study), 267
- overview of, 56–57
- Persons table (Pampered Pet case study), 267
- Phones table (Pampered Pet case study), 269
- Shifts table (Pampered Pet case study), 268–269
- between tables in Access, 294, 296
- TimeEntries table (Pampered Pet case study), 268–269
- validation list, 128
- Vendors table (Pampered Pet case study), 270

foreign keys

- data integrity and, 68
- relational databases and, 54

formats, verifying field formats, 217

forms (Pampered Pet case study), 231

Fourth Normal Form (4NF), 158–161

- arranging data in, 160–161
- how it works, 161
- overview of, 158–159
- qualifications/rules, 159

FROM clause, SQL statements, 359–362

full backups, 380

functional requirements

- customer/user needs, 67
- Pampered Pet case study, 229–231

future needs, customer/user needs, 76

G

Garbage In, Garbage Out (GIGO), 4

generic bad guy, who's who, 72

get it done viewpoint, 50

GIGO (Garbage In, Garbage Out), 4

goals, database design, 3–23

- ACID transactions, 13–14
- atomicity of transactions, 12–13
- complex calculations, 19
- consequences of good and bad designs, 19–20
- consistency, 10
- CRUD, 9
- databases as information containers, 5–7
- ease of use, 16
- error correction, 11
- exercise, 22
- exercise solutions, 403–408
- low cost and extensibility, 16
- overview of, 3–5
- Pampered Pet case study, 229
- persistence and backups, 14–15
- portability, 16–17
- retrieval, 9–10
- security, 17–18
- sharing, 18
- speed, 11–12
- strengths and weaknesses of information containers, 7–8
- summary, 21
- validity, 10–11

GRANT statement, SQL, 393

graphical user interface (GUI), 313

graphics, mockups and, 230

group attributes

- ERDs, 250–251
- SOMs, 96

many-to-many associations

GROUP BY clause, SQL, 362–363
GUI (graphical user interface), 313

H

hierarchical data, 195–197
 improving ERDs and, 253
 overview of, 195–196
 working with, 196–197
hierarchical databases
 examples of, 32–33
 overview of, 31–32
 pros/cons, 34
 Windows system registry, 26–27
 XML databases, 34
 XML structures and, 37
hives, registry, 27
HTML, 34–35
hybrid objects, semantic views, 102

I

identifiers
 ERDs, 107
 SOMs, 97
IDs
 identifying, 258
 overuse of, 218–220
incremental backups, 380
indexes
 relational databases, 54
 repairing/compacting databases and, 383
information containers
 databases as, 5–7
 strengths and weaknesses of, 7–8
information theft, 17–18
inheritance
 ERDs, 109–110, 253
 SOMs, 105–106
INI files, 26
initialization scripts, 370–371
INSERT statement, SQL, 357–358
insertion anomalies
 2NF and, 147
 BCNF and, 155
integrity, determining need for (Pampered Pet case study), 236–238
integrity constraints, 68
interactive applications, performance needs and, 235
International Organization for Standardization (ISO), 347
intersection operation, relational databases, 58

InventoryItems table, (Pampered Pet case study), 268
ISO (International Organization for Standardization), 347
isolation, 13

J

join operation, relational databases, 58

K

key constraints
 DKNF and, 165–166
 overview of, 112
keys
 enforcing uniqueness with, 192
 entity identifiers, 107
 ID numbers vs. natural keys, 218–220
 not defining natural keys, 220–221
 relational databases, 52–54

L

link tables, 176
links, network, 40
Linux, 314
load handling, multi-tier applications and, 210
locking data, 204–205
logs/logging
 audit trails, 203–205
 durability and, 14
lookup tables
 Courses table (Pampered Pet case study), 265
 example of use of, 175–176
 Pets table (Pampered Pet case study), 267
 types of tables, 177

M

maintenance, 379–387
 backups, 379–382
 compacting database, 383
 data warehousing, 382–383
 exercise solutions, 459–460
 exercises, 387
 overview of, 379
 performance tuning, 383–386
 repairing database, 383
 summary, 386–387
many-to-many associations, 186–188
 CustomerCourses table (Pampered Pet case study), 266

many-to-many associations (*continued*)

many-to-many associations (*continued*)

- improving ERDs and, 254–256
- multiple many-to-many associations, 186–188
- network data and, 198
- OrderItems table (Pampered Pet case study), 268
- overview of, 186

many-to-many reflexive associations, 198

matrix management, 40

Microsoft Access, 287–311

- 1NF and, 139
- building test scores database, 291–293
- creating field constraints, 296–298
- creating queries, 301–304
- creating table constraints, 298–300
- defining relationships, 293–296
- exercise solutions, 454
- exercises, 308–311
- getting started, 288–290
- overview of, 287–288
- pros/cons, 287
- Query Design View, 304–306
- SQL View, 306–308
- summary, 308

middle (business) tier, multi-tier applications, 130

milestones, requirements document, 80–81

mishmash tables, as design pitfall, 215–217

mission statement, in requirements document, 242–243

models. *See* data models

modification anomalies

- 3NF and, 151
- 4NF and, 159
- causes of, 180
- redundancy and, 167

multiple-object associations, 188–191

multi-tier applications, 129–133

- benefits of, 175
- business rules for, 131–133
- goals and advantages of, 130
- load handling and, 210
- overview of, 129
- tiers of, 129–130

multi-valued dependencies

- 4NF and, 159
- 5NF and, 162–163

MySQL

- 1NF and, 139
- building database with, 324–327
- Command Line Client, 314–318
- creating/modifying databases, 321–324
- defining triggers, 334–336
- editing data, 320–321
- editing databases, 333–334

- EER diagrams, 331–333

- executing queries, 320

- executing scripts, 327–329

- exercise solutions, 454

- exercises, 338–341

- exporting scripts, 336–337

- installing, 313–314

- loading, 330–331

- overview of, 313

- privileges, 393–395

- pros/cons, 313

- Query Browser, 319–320

- SQL scripts, 318–319

- SQL syntax help feature, 329–330

- summary, 338

- Workbench, 330

N

naming conventions

- database objects, 178–179

- poor use of, 208–209

nesting rules, XML elements, 35

network data, 198–200

networks

- overview of, 40–42

- pros/cons, 42

nodes, network, 40

normalization, 137–172

- Boyce-Codd Normal Form (BCNF), 154–158

- data modeling and, 249

- determining best level of, 168–169

- disadvantages of higher forms of, 179–180

- Domain/Key Normal Form (DKNF),

- 165–167

- exercise solutions, 430–434

- exercises, 170–172

- Fifth Normal Form (5NF), 161–165

- First Normal Form (1NF), 138–146

- Fourth Normal Form (4NF), 158–161

- overview of, 137–138

- redundancy and, 167–168

- Second Normal Form (2NF), 146–150

- summary, 169

- Third Normal Form (3NF), 150–153

- too little, 213–214

- too much, 212–213

normalization/refinement (Pampered Pet case study), 273–284

- exercise solutions, 451–454

- exercises, 284

- improving flexibility, 273–275

- overview of, 273

- summary, 283

- verifying 1NF, 275–278

prioritization, of customer/user needs

verifying 2NF, 278–281
 verifying 3NF, 281–283
not null constraints, 55
null values, 55

O

object attributes, 96
object database management systems (ODBMS), 42
object databases, 42, 43
object identifiers, 97
object tables, 177
object-relational databases (ORD), 43
object-relational mapping, 43
objects
 association objects, 102–105
 composite objects, 101
 compound objects, 101–102
 deleted or modified objects in temporal data, 202
 hierarchical data and, 195
 hybrid objects, 102
 inheritance, 105–106
 multiple-object associations, 188–191
 naming conventions and, 178–179, 209
 network data, 198
 as programming abstraction, 42
 reflexive associations, 193
 semantic object identifiers, 97
 simple (atomic) objects, 100–101
 SOs (semantic objects), 96
ODBMS (object database management systems), 42
one-to-many reflexive associations
 hierarchical data and, 195
 overview of, 194–195
one-to-many relationships, 274
one-to-one reflexive associations, 193–194
operating system passwords, 391
operations, relational databases, 57–58
ORD (object-relational databases), 43
ORDER BY clause, SQL, 363–364
OrderItems table (Pampered Pet case study), 268, 279–280
Orders table, (Pampered Pet case study), 267
org charts, as example of hierarchical data, 195

P

Pampered Pet case study
 business rules. *See* business rules (Pampered Pet case study)

customer/user requirements. *See* customer/user requirements (Pampered Pet case study)
 data models. *See* data models (Pampered Pet case study)
 normalization. *See* normalization/refinement (Pampered Pet case study)
 tables. *See* tables (Pampered Pet case study)
participation constraints, 112
partitioning, performance tuning and, 384
passwords, 390–392
 operating system, 391
 overview of, 390
 qualities of good, 391–392
 security planning and, 68
 single-password databases, 390
 user passwords (individual), 390–391
pattern validation, 263
performance
 anxiety related to too much focus on, 214–215
 determining performance needs (Pampered Pet case study), 235
 tuning, 383–386
permissions, database security and, 17
persistence, database design goals, 14–15
Persons table, (Pampered Pet case study), 269
Pets table (Pampered Pet case study), 267, 279
Phones table, (Pampered Pet case study), 269
physical security, 399–400
planning
 benefits of planning ahead, 173–174
 for change, 210–212
 customer/user needs and, 66–67
 lack of preparation as design pitfall, 207–208
 making backup plan, 381–382
points of view, relational databases, 49–50
policies, password, 391. *See also* business rules
portability, database design goal, 16–17
PRD (product requirements document), 80–81
primary key constraints
 OrderItems table (Pampered Pet case study), 268
 overview of, 55
 Vendors table (Pampered Pet case study), 269
primary keys
 1NF and, 140
 2NF and, 146
 5NF and, 168
 building into database, 264
 enforcing uniqueness with, 192
 entity identifiers, 107
 ID numbers vs. natural keys, 218–220
 relational databases, 53–54
 tables in Access and, 289
prioritization, of customer/user needs, 78–79

privileges

privileges, 392–398

- example of use of, 395–396
- how it works, 396–398
- overview of, 392–395

problem solving, understanding user needs, 207

product requirements document (PRD), 80–81

programmers, benefits of documentation to, 174

programs. *See* applications

project look, (Pampered Pet case study), 231

project management, 4

projection operation, relational databases, 57

Project-Join Normal Form. *See* Fifth Normal Form

(5NF)

prototypes, as deliverables, 81

Q

queries

- creating in Access, 301–304
- executing in MySQL, 320

query analyzers, 384

Query Browser, MySQL

- building database with, 324–327
- creating/modifying databases, 321–324
- editing query data, 320–321
- executing queries, 320
- executing scripts, 327–329
- overview of, 319–320

Query Design tool, Access, 301, 304–306

query results, SQL queries, 59

Query Wizard, Access, 301–304

questions

- creating list of questions in problem solving, 67
- Pampered Pet case study, 228–229

R

Read operations

- atomicity of transactions and, 12
- consistency and, 10
- CRUD and, 9
- retrieval and, 9–10
- speed and, 11

records

- business rules for cross-record checks, 124
- compacting databases, 383
- deleted or modified objects in temporal data, 202
- XML structures and, 38

recursive associations. *See* reflexive associations

redundancy

- allowing for redundant data, 179–180
- normalization and, 167–168

referential integrity constraints, 56, 295

reflexive associations

- one-to-many reflexive associations, 194–195
- one-to-one reflexive associations, 193–194
- overview of, 193

registry, Windows system, 26–27

relational databases

- constraints, 55–57
- exercise solutions, 411–412
- exercises, 59–61
- features, 29
- indexes, 54
- keys, 52–54
- many-to-many relationships and, 186
- operations, 57–58
- overview of, 27–29, 49
- points of view, 49–50
- pros/cons, 29
- relations, attributes, and tuples, 52
- summary, 59
- tables, rows, and columns, 50–52

relational models, 113–117

- business rules (Pampered Pet case study), 270
- converting ERDs to, 116–117
- converting ERDs to (Pampered Pet case study), 256–259
- converting SOMs to, 113–116
- converting SOMs to (Pampered Pet case study), 256–259
- integrating with other models (Pampered Pet case study), 256–259
- overview of, 113

relations, as formal term for a table, 52

relationships

- broken, 255
- building ERDs, 251–252
- defining relationships between tables in Access, 293–296
- determining (Pampered Pet case study), 233–235
- in ERD (entity-relationship diagram), 107–108

reliability, data integrity and, 68

remote access, database security and, 17

repairing databases, 383

repeated attribute associations, 191–192

reports (Pampered Pet case study), 231

required values, 125

requirements document

- customer/user needs, 80–81
- preparation and, 207
- writing (Pampered Pet case study), 242–243

resources, for SQL, 346

retrieval, database design goals, 9–10

REVOKE statement, SQL, 393

rollback, atomicity of transactions and, 12

rows

- 1NF and, 139–146
- relational databases and, 51–52
- tuples and, 52

S**sanity checks, 125**

- Addresses table (Pampered Pet case study), 264
- coding directly into databases, 264
- Courses table (Pampered Pet case study), 264
- InventoryItems table (Pampered Pet case study), 268
- OrderItems table (Pampered Pet case study), 268
- Pets table (Pampered Pet case study), 267
- Shifts table (Pampered Pet case study), 268–269
- TimeEntries table (Pampered Pet case study), 268–269
- uses of, 263

scale, thinking too small, 210**script editor, MySQL, 327****scripts, 369–377**

- categories of, 370–371
- executing with MySQL Query Browser, 327–329
- exercise solutions, 457–459
- exercises, 377
- exporting in MySQL, 336–337
- loading scripts with MySQL Workbench, 330–331
- ordering SQL commands and, 371–376
- overview of, 318–319, 369
- summary, 376–377

searches, data storage and, 180. See also queries**Second Normal Form (2NF), 146–150**

- anomalies, 147
- arranging data in, 149
- how it works, 149–150
- qualifications/rules, 146
- verifying (Pampered Pet case study), 278–281

secondary (alternate) key, relational databases, 53–54**security, 389–401**

- customer/user needs, 68–69
- databases vs. alternative methods of data storage, 180
- determining needs (Pampered Pet case study), 235
- exercise solutions, 460–466
- exercises, 401
- goals of database design, 17–18
- initial configuration, 398

overview of, 389

- passwords, 390–392
- physical security, 399–400
- privileges, 392–398
- right level of, 389
- summary, 400
- too much, 398–399

SELECT clause, SQL, 358–359**SELECT statement, 301**

- FROM clause, 359–362
- GROUP BY clause, 362–363
- MySQL, 317
- ORDER BY clause, 363–364
- SELECT clause, 358–359
- SQL, 358
- SQL view, in Access, 307
- WHERE clause, 362

selection operation, relational databases, 57**semantic classes, 96****semantic object models. See SOMs (semantic object models)****semantic views**

- association objects, 102–105
- class types, 100
- comments and notes, 106
- composite objects, 101
- compound objects, 101–102
- hybrid objects, 102
- inherited objects, 105–106
- overview of, 99–100
- simple (atomic) objects, 100–101

sensitive data, protecting, 203**shadowing, durability and, 14****sharing, database design goals, 18****Shifts table, (Pampered Pet case study), 269****short-timer, who's who and, 71–72****SHOW DATABASES statement, 316****SHOW TABLES statement, 316****sidekick/gopher, who's who and, 71****simple (atomic) objects, 100–101****simple attributes, semantic objects, 96****sketches/mockups**

- of forms, 231
- planning process and, 230

snapshot views, of temporal data, 201**software development**

- allowing for redundant data, 179–180
- benefits of planning ahead, 173–174
- considering what to include in a database, 180–181
- converting domains into tables, 175–176
- documentation, 174
- exercise solutions, 434–436
- exercises, 182–183
- multi-tier architecture for, 175

software development (*continued*)

software development (*continued*)

naming conventions for, 178–179
 overview of, 173
 summary, 181–182
 types of tables, 176–177
 well-focused tables, 176–177

SOMs (semantic object models)

association objects, 102–105
 building initial (Pampered Pet case study), 245–248
 cardinality of attributes, 97
 class types, 100
 classes and objects, 96
 comments and notes, 106
 composite objects, 101
 compound objects, 101–102
 converting to ERD, 250
 converting to relational models, 113–116
 example of, 97–98
 hybrid objects, 102
 improving (Pampered Pet case study), 248–250
 inherited objects, 105–106
 integrating all models (Pampered Pet case study), 248–250
 object identifiers, 97
 Pampered Pet case study, 245
 semantic views, 99–100
 simple (atomic) objects, 100–101
 working with, 99

SOs (semantic objects), 96

spaces, naming conventions and, 209

special characters, naming conventions and, 179, 209

specifications (specs), 80–81

speed

database design goals, 11–12
 normalization and, 179–180
 performance needs and, 235

spreadsheets, 30–31

overview of, 30
 pros/cons, 30–31

SQL (Structure Query Language)

background of, 345–346
 FROM clause, 359–362
 commands, 348–350
 CREATE INDEX command, 355–356
 CREATE TABLE command, 350–355
 DELETE command, 365–366
 DROP command, 356
 exercise solutions, 454–457
 exercises, 366–368
 GRANT and REVOKE for privileges, 393
 GROUP BY clause, 362–363
 INSERT command, 357–358
 ORDER BY clause, 363–364

ordering, 371–376
 overview of, 345–368
 resources for, 346
 scripts. *See* scripts
 SELECT clause, 358–359
 SELECT command, 301, 358
 SQL view in Access, 306–308
 standards, 346–347
 summary, 366
 syntax, 347–348
 UPDATE command, 364–365
 WHERE clause, 362

SQL query operation, relational databases, 58–59

SQL Server, 139

SQL view, Access, 306–308

stakeholders

getting input from, 73
 Pampered Pet case study, 228
 who's who, 71

standards

naming, 208–209
 SQL, 346–347

states table, 175

superkeys

BCNF and, 154
 relational databases, 53–54

syntax, SQL, 329–330, 347–348

system reliability, 68

T

Table Editor, MySQL, 324

table-level check constraints

business rules and, 122
 overview of, 55

tables

1NF and, 139–146
 2NF and, 146–150
 3NF and, 150–154
 4NF and, 158–161
 BCNF and, 154–158
 business rules for cross-table checks, 124
 constraints, in Access, 298–300
 converting domains to, 175–176
 creating in SQL, 352–355
 data needed as basis of design, 67
 DKNF, 165–167
 documentation, 174
 ID numbers vs. natural keys, 218–220
 mishmash or catchall tables as design pitfall, 215–217
 naming conventions, 209
 ordering, 373–376
 in relational databases, 50–52

- relations/relationships, 52, 293–296
 - types of, 176–177
 - well-focused, 176–177
 - XML structures and, 37–38
 - tables (Pampered Pet case study)**
 - Courses table, 264–266
 - CustomerCourses table, 266–267
 - Customers table, 267
 - Employees table, 267
 - InventoryItems table, 268
 - OrderItems table, 268
 - Orders table, 267
 - Persons table, 269
 - Pets table, 267
 - Phones table, 269
 - Shifts table, 269
 - TimeEntries table, 268–269
 - Vendors table, 269–270
 - TCL (Transaction Control Language), 348–350**
 - team members**
 - team building, 230
 - who's who, 70–73
 - temporal data, 200–203**
 - deciding what to temporalize, 203
 - deleted objects, 202
 - effective/valid dates, 200–202
 - overview of, 200
 - temporal databases, 45**
 - test scores database, 291–293**
 - testing**
 - insufficient, 214
 - preparation and, 208
 - text, XML files as, 39**
 - theoretical viewpoint, problem response, 50**
 - Third Normal Form (3NF), 150–153**
 - arranging data in, 152
 - how it works, 153
 - qualifications/rules, 150
 - stopping normalization at, 154
 - transitive dependencies and, 150–151
 - verifying (Pampered Pet case study), 281–283
 - TimeEntries table (Pampered Pet case study), 268–269, 280–281**
 - training courses, 229**
 - Transaction Control Language (TCL), 348–350**
 - Transact-SQL language, 347**
 - transitive dependencies, 3NF and, 150–151, 281–282**
 - tree structures, hierarchical data and, 31, 195**
 - triggers, defining in MySQL, 334–336**
 - tuples, relational databases, 52**
 - turnkey records, 204–205**
 - two-way relationships, 248**
- U**
- union operation, relational databases, 58**
 - unique constraints, 56**
 - unique keys, relational databases, 53–54**
 - unrelated multi-valued dependencies, 4NF and, 159**
 - update anomalies**
 - 2NF and, 147
 - BCNF and, 155
 - Update operations**
 - atomicity of transactions and, 12
 - CRUD and, 9
 - speed and, 11
 - validity, 10–11
 - UPDATE statement**
 - performance tuning and, 384
 - SQL (Structure Query Language), 364–365
 - URLs, as alternative to database storage, 180**
 - use cases, 81–85**
 - diagrams in, 82–83
 - list of what might be included in, 83
 - modeling and, 91
 - overview of, 81
 - preparation and, 207
 - specifications in, 81–82
 - working with, 84–85
 - writing (Pampered Pet case study), 238–242
 - writing before vs. after building database, 240
 - user classes**
 - Pampered Pet case study, 236
 - security planning and, 69
 - user interface models, 92–96**
 - building with sketches and forms, 92–94
 - overview of, 92
 - working with, 95
 - user interface, (Pampered Pet case study), 232–233**
 - user interface tier, multi-tier applications, 129**
 - user needs, 218–220**
- V**
- valid time, in temporal database, 45**
 - validation data, placing in separate tables, 124**
 - validation list, foreign key constraints, 128**
 - validation rules, Access, 296–298**
 - validity**
 - business rules for parameters of, 124
 - database design goals, 10–11
 - values**
 - data integrity and, 68
 - list of allowed, 124
 - repeated values causing anomalies, 168
 - required values, 125

values (*continued*)

values (*continued*)

- storing multiple values, 140
- XML structures and, 38

Vendors table, (Pampered Pet case study), 269–270

verifying approach, customer/user needs, 79–80

version control, 371

virtual tables, SQL queries, 59

W

weak entity, 112

WHERE clause, SQL, 362

whitespace

- SQL, 348
- XML, 35

who's who, 70–73

Windows system registry, 26–27

Workbench, MySQL

- creating EER diagrams, 331–333
- editing databases, 333–334
- exporting scripts, 336–337
- loading scripts with, 330
- overview of, 314

X

XML (eXtensible Markup Language), 34–40

- basics of XML, 34–37
- overview of, 34
- structures, 37–39
- summary, 39–40