

# Index

## • Symbols •

- \* (asterisk) wildcard character, 54, 60, 63
- [^] (caret inside square brackets) wildcard character, 63
- = (equality) comparison operator, 56
- > (greater than) comparison operator, 56
- >= (greater than or equal to) comparison operator, 56
- != (inequality) comparison operator, 56
- <> (inequality) comparison operator, 56
- < (less than) comparison operator, 56
- <= (less than or equal to) comparison operator, 56
- % (percent sign) wildcard character, 63
- [ ] (square brackets)
  - SELECT statement, 52
  - wildcard characters, 63
- \_ (underscore) wildcard character, 63

## • A •

- Abs () function, 239
- accessing system stored procedures, 194–195
- Acos () function, 239
- adding
  - logins, 288–290
  - users, 293–296
- administration, 16
- AdventureWorks database, 19, 43
- AFTER trigger, 209–210, 213–220
- aggregate functions, 236
- aggregates, 65
- aliases for table names, 76–77
- ALLOW\_PAGE\_LOCKS option (CREATE INDEX statement), 262
- ALLOW\_ROW\_LOCKS option (CREATE INDEX statement), 262
- ALTER ASYMMETRIC KEY statement, 315
- ALTER AUTHORIZATION statement, 293
- ALTER DATABASE statement, 139
- ALTER FUNCTION statement, 233
- ALTER INDEX statement, 264–265
- ALTER LOGIN statement, 290
- ALTER MASTER KEY statement, 311–312
- Alter () method, 378

- ALTER PROCEDURE statement, 205
- ALTER ROLE statement, 303
- ALTER SERVICE MASTER KEY statement, 309
- ALTER SETTINGS permissions, 346
- ALTER TABLE statement, 161–162
- ALTER USER statement, 292
- alternatives to Transact-SQL (T-SQL), 24
- Analysis Services Data Mining Expressions (DMX query), 30
- Analysis Services Multidimensional Expressions (MDX) query, 30
- Analysis Services XMLA query, 30
- AND logical operator, 58–59
- application roles, 303–304
- App\_Name () function, 244
- approximate numeric data types, 20, 145–146
- AS keyword, 65, 77
- Ascii () function, 249
- Asin () function, 239
- assembly, 343, 348–350
- assigning
  - permissions, 297–300
  - privileges, 298
  - roles, 302–303
- associating users with logins, 295–296
- asterisk (\*) wildcard character, 54, 60, 63
- asymmetric keys, 312–315
- Atan () function, 240
- Atn2 () function, 240
- AttachDatabase () method, 378
- attributes, 128–129
- authentication, 27–28, 289
- auto counter, 113
- AVG aggregate, 65
- avg () function, 236, 253–254

## • B •

- BACKUP DATABASE permission, 298
- BACKUP LOG permission, 298
- BACKUP MASTER KEY statement, 311
- BACKUP SERVICE MASTER KEY statement, 309–310
- BEGIN TRANSACTION statement, 122
- BETWEEN keyword, 60–61
- bigint data type, 144

- binary data types, 20, 148
  - binary\_checksum() function, 236
  - binary(n) data type, 148
  - bit data type, 144
  - Books Online, 10, 39–42, 100, 395–396
  - B-tree (binary tree), 257
  - budgeting time, 392
  - Business Intelligence Development Studio, 39, 351
  - Business Intelligence Web site, 397
  - business rules
    - constraints, 12, 163–178
    - CREATE RULE permission, 298
    - triggers, 11, 209–223
- C •
- caret inside square brackets ([^]) wildcard character, 63
  - cartesian product, 89
  - case sensitivity, 142
  - Cast() function, 96–100, 244
  - catalog views, 286, 313–314
  - causes of errors, 267–268
  - Ceiling() function, 240
  - Certificate object, 18
  - certificates, 312–314, 316–317
  - changing
    - default schema for a user, 292
    - ownership of a schema, 293
  - Channel 9 Web site, 398
  - Char() function, 249
  - character data types, 21, 146–147
  - Charindex() function, 249
  - char(n) data type, 146
  - CHECK constraint, 164, 173–178
  - checking installation status of SQLCMD utility, 36
  - checksum() function, 236
  - checksum\_agg() function, 236
  - CLOSE MASTER KEY statement, 312
  - CLR (Common Language Runtime). *See* Common Language Runtime (CLR)
  - clustered index, 257–260
  - Coalesce() function, 244
  - Code Access Security, 344
  - Code Editor, 30
  - code templates, 31–33
  - Collation\_Property() function, 244
  - Col\_Length() function, 242
  - Col\_Name() function, 242
  - column-level constraints, 179–180
  - ColumnProperty() function, 242
  - columns
    - creating, 149, 156
    - data types, 143
    - identity column, 113–116, 151–152
    - maximum number of bytes in key columns, 262–264
    - modifying, 161–162
    - naming, 142–143
  - Columns\_Update() function, 244
  - combining comparison operators, 57–58
  - COMMIT TRANSACTION statement, 122
  - Common Language Runtime (CLR)
    - advantages of using, 342–343
    - Code Access Security, 344
    - configuring SQL Server 2005 for CLR, 345–347
    - defined, 341–342
    - enabling, 345–347
    - functions, 227–228
    - Microsoft Intermediate Language (MSIL), 344
    - security, 344
    - stored procedures, 198
  - common table expressions, 100–104
  - CompareUrn() method, 378
  - comparison operators, 56–58
  - Compatibility Level object, 18
  - compatibility levels, 21
  - configuration functions, 237
  - configuring
    - network protocols, 37–38
    - security settings, 284–286
    - SQL Server 2005 for CLR, 345–347
  - connecting, 27–29, 285
  - @@Connections function, 251
  - constraints
    - CHECK, 164, 173–178
    - column-level, 179–180
    - DEFAULT, 164, 168–170
    - defined, 12, 163
    - deleting, 178–179
    - integrity of data, 163
    - NOT NULL, 163–168
    - PRIMARY KEY, 70–71, 164
    - REFERENCES, 164
    - table-level, 179–180
    - UNIQUE, 163–164, 170–173
  - CONTAINS keyword, 60–62
  - Containstable() function, 247
  - controlling transactions, 122
  - Convert() function, 96–100, 244

- converting data types, 96–100
  - Copy Database Wizard, 105–110
  - copying databases, 105–110
  - Cos() function, 240
  - Cot() function, 240
  - count() function, 236
  - count\_big() function, 236
  - @Cpu\_Busy function, 251
  - CREATE APPLICATION ROLE statement, 304
  - CREATE ASSEMBLY statement, 343, 348–350
  - CREATE ASYMMETRIC KEY statement, 314–315
  - CREATE CERTIFICATE statement, 316–317
  - CREATE DATABASE permission, 298–299
  - CREATE DATABASE statement, 136–137
  - CREATE DEFAULT permission, 298
  - CREATE FUNCTION permission, 298
  - CREATE FUNCTION statement, 227–232
  - CREATE INDEX statement, 192, 255, 260–261
  - CREATE LOGIN statement, 288–289
  - CREATE MASTER KEY statement, 310
  - CREATE PROCEDURE permission, 298
  - CREATE PROCEDURE statement, 200–201
  - CREATE ROLE statement, 303
  - CREATE RULE permission, 298
  - CREATE SCHEMA statement, 292–293
  - CREATE SYMMETRIC KEY statement, 313, 315–316
  - CREATE TABLE permission, 298–299
  - CREATE TABLE statement, 149–150
  - CREATE TRIGGER statement, 210, 214–215
  - CREATE USER statement, 294
  - CREATE VIEW permission, 298
  - CREATE VIEW statement, 182–190
  - CREATE XML SCHEMA COLLECTION statement, 326–327
  - creating
    - assembly, 343, 348–350
    - asymmetric key, 314–315
    - certificates, 316–317
    - columns, 149, 156
    - cross join, 88–89
    - database master key, 310–311
    - database references, 363–364
    - databases, 136–138, 154–155
    - functions, 227–232
    - indexes, 192, 255, 260–261
    - inner join, 73–84
    - logins, 288–290
    - outer join, 84–88
    - projects, 363–372
    - queries, 30–31, 33–35
    - relationships, 151–154, 157–161
    - roles, 303–304
    - schemas, 292–293
    - SMO applications, 381–388
    - stored procedures, 200–201
    - symmetric keys, 315–316
    - tables, 149–150, 154–158
    - triggers, 209–210, 214–215
    - users, 293–296
    - views, 181–190
  - cross join, 70, 88–89
  - cryptographic properties, 313–314
  - CryptProtectData method, 308
  - CryptUnprotectData method, 308
  - Current\_Timestamp function, 245
  - Current\_User function, 245, 304, 306
  - cursor data type, 21, 148
  - cursor functions, 238
    - @@Cursor\_Rows function, 238
    - @@Cursor\_Status function, 238
- D •
- data entry, 11–12
  - data integrity, 163, 209, 212
  - data protection API (DPAPI), 308
  - data rows (indexes), 257–258
  - data transfers, 107–108
  - data types
    - bigint, 144
    - binary, 20, 148
    - binary(n), 148
    - bit, 144
    - character (non-Unicode), 146–147
    - char(n), 146
    - columns, 143
    - converting, 96–100
    - cursor, 21, 148
    - date-related, 20, 146
    - datetime, 146
    - decimal, 144
    - float, 145–146
    - Image, 148
    - int, 144
    - money, 144
    - nchar(n), 147
    - ntext, 147
    - numeric, 20, 143–145
    - numeric, 144
    - nvarchar(n), 147
    - real, 145
    - smalldatetime, 146
    - smallint, 144
    - smallmoney, 145

- data types (*continued*)
  - sql\_variant, 148
  - string, 20
  - table, 148
  - text, 147
  - time-related, 20, 146
  - timestamp, 148
  - tinyint, 145
  - typecasting, 96–100
  - Unicode, 21, 147
  - uniqueidentifier, 149
  - varbinary(n), 148
  - varchar(n), 147
  - variants, 21, 148
  - xml, 13, 15, 21, 149, 322
- database design
  - definition phase, 126
  - delivery phase, 131
  - design phase, 127–130
  - documentation phase, 131
  - entity-relationship diagram (ERD), 127–130
  - evaluation phase, 127
  - implementation phase, 130–131
  - importance of, 125
  - maintenance phase, 131
  - requirements phase, 127
  - testing phase, 131
- Database Diagram Designer (Visual Studio 2005), 351, 356–357
- database engine, 2, 15–17
- database engine queries, 30
- Database Explorer (Visual Studio 2005), 352
- database indexes. *See* indexes
- database level encryption, 308, 310–312
- database master key, 310–312
- Database object, 17
- database references, 363–364
- database roles, 302–303
- DatabaseProperty() function, 242
- databases
  - compatibility levels, 21
  - copying, 105–110
  - CREATE DATABASE permission, 298–299
  - creating, 136–138, 154–155
  - default size, 137–138
  - deleting, 140
  - dropping, 140
  - instances, 27
  - modifying, 139
  - normal forms, 131–136
  - pending changes, 122
  - permissions, 297–302
  - planning, 125
  - privileges, 298
    - properties, 137–138
    - sample databases, 19–20, 42–44
    - system databases, 19, 194
    - viewing, 48–50
    - views, 181–182, 184–191, 193
- Datalength function, 245
- date and time functions, 238–239
- Dateadd() function, 238
- Datediff() function, 238
- @@DateFirst function, 237
- Datename() function, 238
- date-related data types, 20, 146
- datetime data type, 146
- Day() function, 239
- db\_accessadmin role, 302
- db\_backupoperator role, 302
- db\_datareader role, 302
- db\_datawriter role, 302
- db\_ddladmin role, 302
- db\_denydatareader role, 303
- db\_denydatawriter role, 303
- Db\_Id() function, 242
- Db\_Name() function, 242
- db\_owner role, 303
- db\_securityadmin role, 303
- DDL triggers, 209–213, 223–224
- debugging triggers, 224–225
- decimal data type, 144
- decryption, 309–311
- DEFAULT constraint, 164, 168–170
- Default object, 18
- default schema, 292
- default size of databases, 137–138
- DefaultFileGroup object, 18
- DefaultSchema object, 18
- definition phase of database design, 126
- Degrees() function, 240
- DELETE permission, 298
- DELETE statement, 116–120
- DeleteBackupHistory() method, 378
- deleting
  - assembly, 350
  - asymmetric key, 315
  - constraints, 178–179
  - databases, 140
  - functions, 233–234
  - indexes, 266
  - logins, 290
  - roles, 303
  - rows from a table, 116–120
  - schemas, 293
  - stored procedures, 204–205
  - symmetric keys, 316
  - XML data, 334–335

- delimited identifiers, 21–22
  - delivery phase of database design, 131
  - Deny() method, 379
  - DENY statement, 301–302
  - denying permissions, 301–302
  - dependencies, 135
  - designing and testing on development
    - hardware, 393
  - designing databases. *See* database design
  - designing relationships, 136
  - Detach and Attach method of transferring data, 107–108
  - DetachDatabase() method, 379
  - DetachedDatabaseInfo() method, 379
  - deterministic functions, 234
  - Developer edition, 11
  - development hardware, 393
  - Difference() function, 249
  - disk I/O operations, 256
  - DML triggers 209–213
  - documentation, 131, 391–392
  - downloads for Visual Studio 2005, 351
  - DPAPI (data protection API), 308
  - DROP ASSEMBLY statement, 350
  - DROP ASYMMETRIC KEY statement, 315
  - DROP DATABASE statement, 140
  - DROP FUNCTION statement, 233–234
  - DROP INDEX statement, 266
  - DROP LOGIN statement, 290
  - DROP ROLE statement, 303
  - DROP SCHEMA statement, 293
  - DROP SYMMETRIC KEY statement, 316
  - DROP\_EXISTING option (CREATE INDEX statement), 262
  - dropping. *See* deleting
- **E** ●
- editing. *See* modifying
  - editions of SQL Server, 11, 394
  - enabling Common Language Runtime (CLR), 345–347
  - encryption
    - asymmetric keys, 312–315
    - catalog views, 313
    - certificates, 312–314, 316–317
    - cryptographic properties, 313–314
    - data protection API (DPAPI), 308
    - database level, 308, 310–312
    - database master key, 310–312
    - decryption, 309–311
    - defined, 307
    - service master key, 308–310
    - session key, 308
    - SQL Server 2005 level, 308–309
    - step-by-step example, 317–318
    - symmetric keys, 312–313, 315–316
    - Windows level, 308
  - entering data, 11–12
  - Enterprise edition, 11
  - entities, 127–130
  - EnumAvailableMedia() method, 379
  - EnumCollations() method, 379
  - EnumDatabaseMirrorWitnessRoles() method, 379
  - EnumDetachedDatabaseFiles() method, 379
  - EnumDetachedLogFiles() method, 379
  - EnumDirectories() method, 379
  - EnumErrorLogs() method, 379
  - EnumLocks() method, 379
  - EnumMembers() method, 379
  - EnumObjectPermissions() method, 379
  - EnumPerformanceCounters() method, 379
  - EnumProcesses() method, 379
  - EnumServerAttributes() method, 379
  - EnumServerPermissions() method, 380
  - EnumStartupProcedures() method, 380
  - EnumWindowsDomainGroups() method, 380
  - EnumWindowsGroupInfo() method, 380
  - EnumWindowsUserInfo() method, 380
  - equality (=) comparison operator, 56
  - Equals() method, 380
  - ERD (entity-relationship diagram), 127–130
  - error handling. *See* handling errors
  - @@ERROR system function, 245, 267, 269–272
  - ERROR\_LINE function, 276–278
  - ERROR\_MESSAGE function, 276
  - ERROR\_NUMBER function, 276
  - ERROR\_PROCEDURE function, 276–279
  - errors
    - causes of, 267–268
    - severity of, 268
    - transactions, 122
  - ERROR\_SEVERITY function, 276–277
  - ERROR\_STATE function, 276–277
  - ESCAPE keyword, 60
  - escaping wildcard characters, 60
  - establishing hardware requirements, 394
  - estimated execution plan of a query, 35–36
  - evaluation phase of database design, 127
  - event notifications, 212
  - exact numeric data types, 20, 143–145
  - exclusive inner join, 86
  - EXEC statement, 199
  - EXECUTE permission, 298
  - EXECUTE statement, 199

executing  
 queries, 31  
 stored procedures, 199, 202–204  
 triggers, 211  
 execution plan of a query, 35–36  
 EXISTS keyword, 94–96  
 Exp() function, 240  
 Express edition, 11  
 extended stored procedures, 199

## • F •

@@Fetch\_Status function, 238  
 Filegroup\_Id() function, 242  
 Filegroup\_Name() function, 242  
 Filegroupproperty() function, 242  
 File\_Id() function, 242  
 File\_Name() function, 242  
 Fileproperty() function, 243  
 FILLFACTOR option (CREATE INDEX statement), 261  
 filtering stored procedures, 195–197  
 first normal form, 132–134  
 flexibility of Transact-SQL, 1  
 float data type, 145–146  
 Floor() function, 240  
 fn\_Helpcollations() function, 245  
 ::fn\_Servershareddrives()  
 function, 245  
 fn\_Trace\_Geteventinfo function, 248  
 fn\_Trace\_Getfilterinfo function, 248  
 fn\_Trace\_Getinfo() function, 248  
 fn\_Trace\_Gettable() function, 248  
 ::fn\_Virtualfilestats() function, 252  
 ::fn\_Virtualservernodes()  
 function, 245  
 FOR XML clause, 323, 336–340  
 fora (forums), 396  
 foreign key  
 defined, 136  
 joins, 71  
 REFERENCES constraint, 164  
 table relationships, 152–154, 158, 160–161  
 Formatmessage() function, 245  
 FREETEXT keyword, 60  
 Freetexttable() function, 247  
 FROM clause  
 CREATE VIEW statement, 183  
 DELETE statement, 117–119  
 joins, 75–76  
 SELECT statement, 52–55  
 full outer join, 84  
 full-text index, 259

Fulltextcatalogproperty() function, 243  
 Fulltextserviceproperty() function, 243  
 functions (system)  
 Abs(), 239  
 Acos(), 239  
 aggregate functions, 236  
 App\_Name(), 244  
 Ascii(), 249  
 Asin(), 239  
 Atan(), 240  
 Atn2(), 240  
 avg(), 236, 253–254  
 binary\_checksum(), 236  
 Cast(), 96–100, 244  
 Ceiling(), 240  
 Char(), 249  
 Charindex(), 249  
 checksum(), 236  
 checksum\_agg(), 236  
 Coalesce(), 244  
 Collation\_Property(), 244  
 Col\_Length(), 242  
 Col\_Name(), 242  
 ColumnProperty(), 242  
 Columns\_Update(), 244  
 configuration functions, 237  
 @@Connections, 251  
 Containstable(), 247  
 Convert(), 96–100, 244  
 Cos(), 240  
 Cot(), 240  
 count(), 236  
 count\_big(), 236  
 @@Cpu\_Busy, 251  
 Current\_TimeStamp, 245  
 Current\_User, 245, 304, 306  
 cursor functions, 238  
 @@Cursor\_Rows, 238  
 @@Cursor\_Status, 238  
 DatabaseProperty(), 242  
 Datalength, 245  
 date and time functions, 238–239  
 Dateadd(), 238  
 Datediff(), 238  
 @@DateFirst function, 237  
 Datename(), 238  
 Day(), 239  
 Db\_Id(), 242  
 Db\_Name(), 242  
 @@Dbts, 237  
 defined, 227, 234  
 Degrees(), 240  
 deterministic, 234  
 Difference(), 249

- @@ERROR, 245, 267, 269–272
- ERROR\_LINE, 276–278
- ERROR\_MESSAGE, 276
- ERROR\_NUMBER, 276
- ERROR\_PROCEDURE, 276–279
- ERROR\_SEVERITY, 276–277
- ERROR\_STATE, 276–277
- Exp(), 240
- @@Fetch\_Status, 238
- Filegroup\_Id(), 242
- Filegroup\_Name(), 242
- Filegroupproperty(), 242
- File\_Id(), 242
- File\_Name(), 242
- Fileproperty(), 243
- Floor(), 240
- fn\_Helpcollations(), 245
- ::fn\_Servershareddrives(), 245
- fn\_Trace\_Geteventinfo, 248
- fn\_Trace\_Getfilterinfo, 248
- fn\_Trace\_Getinfo(), 248
- fn\_Trace\_Gettable(), 248
- ::fn\_Virtualfilestats(), 252
- ::fn\_Virtualservernodes(), 245
- Formatmessage(), 245
- Freetexttable(), 247
- Fulltextcatalogproperty(), 243
- Fulltextserviceproperty(), 243
- Getansinull(), 245
- Getdate(), 239
- Getutcdate(), 239
- Has\_Dbaccess(), 248
- Has\_Perms\_By\_Name, 304–305
- Host\_id(), 245
- Host\_name(), 245
- 'id, 237
- Ident\_Current(), 245
- Ident\_Incr(), 245
- @@Identity, 246
- Ident\_Seed(), 245
- @@Idle, 252
- image functions, 253**
- Index\_Col(), 243
- Indexkey\_Property(), 243
- Indexproperty(), 243
- @@Io\_Busy, 252
- Isdate(), 246
- IS\_MEMBER, 304
- Is\_Member(), 248
- IsNull(), 246
- Isnumeric(), 246
- IS\_SRVROLEMEMBER, 304
- Is\_Srvrolemember(), 248
- @@LangId, 237
- @@Language, 237
- Left(), 249
- Len(), 249
- @@Lock\_Timeout, 237
- Log(), 240
- Log10(), 240
- Lower(), 250
- Ltrim(), 250
- mathematical functions, 239–241**
- max(), 236
- @@Max\_Connections, 237
- @@Max\_Precision, 237
- metadata functions, 242–244**
- min(), 236
- Month(), 239
- Nchar(), 250
- @@NestLevel, 237
- Newid(), 246
- non-deterministic, 234**
- Nullif(), 246
- Object\_Id(), 243
- Object\_Name(), 243
- Objectproperty(), 243
- Objectpropertyex(), 243
- Opendatasource(), 247
- Openquery(), 247
- Openrowset(), 248
- Openxml(), 248
- @@Options, 237
- @@Packet\_Errors, 252
- @@Pack\_Received, 252
- @@Pack\_Sent, 252
- Parsename(), 246
- Patindex(), 250, 253
- Permissions(), 246, 304
- Pi(), 241
- Power(), 241
- @@Procid(), 244
- Quotename(), 250
- Radians(), 241
- Rand(), 241
- @@RemServer, 237
- Replace(), 250
- Replicate(), 250
- Reverse(), 250
- Right(), 250
- Round(), 241
- @@Rowcount(), 246
- Rowcount\_big(), 246
- rowset functions, 247–248**
- Rtrim(), 250
- SCHEMA\_ID, 304
- SCHEMA\_NAME, 304
- Scope\_Identity(), 246

functions (system) (continued)  
 security functions, 248–249, 304–305  
 @@Servername, 237  
 SERVERPROPERTY(), 31  
 Server\_property(), 246  
 @@Servicename, 237  
 Sessionproperty(), 246  
 Session\_User(), 247, 304  
 SETUSER, 305  
 Sign(), 241  
 Sin(), 241  
 Soundex(), 250  
 Space(), 251  
 Sql\_Variant\_Property(), 244  
 Sqrt(), 241  
 Square(), 241  
 Stats\_Date(), 247  
 stdev(), 236  
 Str(), 251  
 string functions, 249–251  
 Stuff(), 251  
 Substring(), 251  
 sum(), 236, 253–254  
 Suser\_Id, 305  
 Suser\_Name, 305  
 Suser\_Sid(), 249, 305  
 Suser\_Sname(), 249, 305  
 sys.fn\_builtin\_permissions, 305–306  
 system statistical functions, 251–252  
 SYSTEM\_USER, 305  
 System\_User, 247  
 Tan(), 241  
 text functions, 253  
 Textptr(), 253  
 @@Textsize, 237  
 Textvalid(), 253  
 @@Timeticks, 252  
 @@Total\_Errors, 252  
 @@Total\_Read, 252  
 @@Total\_Write, 252  
 @@Trancount, 247  
 Typeproperty(), 244  
 Unicode(), 251  
 Update(), 247  
 Upper(), 251  
 User(), 249  
 USER\_ID, 305  
 User\_Id(), 249  
 User\_Name(), 247, 249, 305  
 var(), 236  
 varp(), 236  
 @@Version, 237

viewing list of available system functions,  
 234–235  
 Year(), 239  
 functions (user-defined)  
 Common Language Runtime (CLR), 227–228  
 CREATE FUNCTION permission, 298  
 creating, 227–233  
 defined, 227  
 deleting, 233–234  
 differences from stored procedures,  
 199, 228  
 dropping, 233–234  
 modifying, 233  
 scalar-valued function, 228  
 table-valued function, 228  
 Transact-SQL, 227–228

## • G •

GetActiveDBConnectionCount()  
 method, 380  
 Getansinull() function, 245  
 Getdate() function, 239  
 GetDefaultInitFields() method, 380  
 GetHashCode() method, 380  
 GetPropertyNames() method, 380  
 GetSmoObject() method, 380  
 GetType() method, 380  
 Getutcdate() function, 239  
 Globally Unique Identifier (GUID), 149  
 GO keyword, 201  
 goals for programming, 9–10  
 Google, 398  
 GRANT ALL PRIVILEGES statement, 298  
 GRANT ALL statement, 298  
 Grant() method, 380  
 GRANT statement, 297, 300–301  
 greater than (>) comparison operator, 56  
 greater than or equal to (>=) comparison  
 operator, 56  
 GROUP BY clause in a SELECT statement,  
 52, 65–67  
 GUID (Globally Unique Identifier), 149

## • H •

handling errors  
 @@ERROR system function, 267, 269–272  
 RAISERROR statement, 272–274  
 Transact-SQL (T-SQL), 268  
 TRY . . . CATCH construct, 267, 271–272,  
 274–280  
 hardware requirements, 394

Has\_Dbaccess() function, 248  
 Has\_Perms\_By\_Name function, 304–305  
 HAVING clause in a SELECT statement, 52, 67  
 help, 37, 40  
 Host\_id() function, 245  
 Host\_name() function, 245

## I

'id function, 237  
 Ident\_Current() function, 245  
 identifiers  
   defined, 21, 129  
   delimited, 21–22  
   regular, 21–23  
 Ident\_Incr() function, 245  
 identity column, 113–116, 151–152  
 @@Identity function, 246  
 identity keyword, 151  
 Ident\_Seed() function, 245  
 @@Idle function, 252  
 IGNORE\_DUP\_KEY option (CREATE INDEX statement), 262  
 Image data type, 148  
 image functions, 253  
 implementation phase, 130–131  
 improving performance, 393  
 INCLUDE keyword, 264  
 included columns in indexes, 264  
 Index\_Col() function, 243  
 indexes  
   binary tree (B-tree), 257  
   clustered, 257–260  
   creating, 192, 255, 260–261  
   data rows, 257–258  
   defined, 255  
   deleting, 266  
   dropping, 266  
   full-text, 259  
   included columns, 264  
   index nodes, 257  
   index rows, 257  
   inserts, 192  
   I/O operations, 256  
   maximum number of bytes in key columns, 262–264  
   modifying, 264–265  
   nonclustered, 257–258, 260  
   primary key, 258  
   unique, 258  
   updates, 192  
   uses of, 255–256  
   views, 192, 259  
   XML data, 259, 335–336  
 Indexkey\_Property() function, 243  
 Indexproperty() function, 243  
 inequality (<> or !=) comparison operator, 56  
 Initialize() method, 380  
 inner join  
   creating, 73–84  
   defined, 70  
   exclusive inner join, 86  
 INSERT permission, 298, 301  
 INSERT statement, 111–116  
 inserting rows into tables, 111–116  
 inserts  
   indexes, 192  
   INSERT statement, 111–116  
   views, 191–192  
   XML data, 333–334  
 installing  
   sample databases, 42–44  
   SQL Server Books Online, 40, 395  
   SQL Server 2005, 2, 10, 15  
   Visual Studio 2005, 351  
 instances of databases, 27  
 INSTEAD OF trigger, 209–210, 213, 220–223  
 int data type, 144  
 Integration Services Developer Center Web site, 397  
 intermediate node, 257  
 I/O operations, 256  
 @@Io\_Busy function, 252  
 Isdate() function, 246  
 IsDetachedPrimaryFile() method, 380  
 IS\_MEMBER function, 304  
 Is\_Member() function, 248  
 Isnull() function, 246  
 Isnumeric() function, 246  
 IS\_SRVROLEMEMBER function, 304  
 Is\_Srvrolemember() function, 248  
 IsWindowsGroupMember() method, 380

## • I •

JOIN keyword, 84  
 joins  
   cross join, 70, 88–89  
   defined, 47, 70  
   FROM clause, 75–76  
   inner join, 70, 73–84, 86  
   outer join, 70, 84–88  
   SELECT statement, 73–89  
   subqueries, 91–94  
   uses of, 70–73  
   views, 185–188  
   WHERE clause, 75–76

## • K •

### keys

- defined, 136
- foreign key, 71, 136, 152–154, 158, 160–161, 164
- primary key, 70–71, 136, 151–154, 157–159, 258
- surrogate keys, 113

### keywords

- AS, 65, 77
- BETWEEN, 60–61
- CONTAINS, 60–62
- ESCAPE, 60
- EXISTS, 94–96
- FREETEXT, 60
- GO, 201
- identity, 151
- JOIN, 84
- LIKE, 60, 62–64
- PRIMARY KEY, 151–152

KillAllProcesses() method, 381

KillDatabase() method, 381

KillProcess() method, 381

## • L •

@LangId function, 237

@Language function, 237

leaf node, 257

Left() function, 249

left outer join, 84, 87

Len() function, 249

less than (<) comparison operator, 56

less than or equal to (<=) comparison operator, 56

libraries, 342–343

LIKE keyword, 60, 62–64

listening to clients' needs, 391

@Lock\_Timeout function, 237

Log() function, 240

Log10() function, 240

LogFile object, 18

logical operators, 57–59

Login object, 17

### logins

- adding, 288–290
- associating with users, 295–296
- creating, 288–290
- defined, 287
- deleting, 290
- dropping, 290

modifying, 290

security, 13

viewing, 287

viewing information about, 287–288

Lower() function, 250

Ltrim() function, 250

## • M •

maintenance phase of database design, 131

malicious code, 13

managing services, 37–38

manifest, 343

mantissa, 146

many-to-many relationship, 82–83, 129–130, 393

master database, 19

### master keys

database master key, 310–312

service master key, 308–310

mathematical functions, 239–241

MAX aggregate, 65

max() function, 236

@@Max\_Connections function, 237

MAXDOP option (CREATE INDEX statement), 262

maximum number of bytes in key columns, 262–264

@@Max\_Precision function, 237

metadata functions, 242–244

### methods

Alter(), 378

AttachDatabase(), 378

CompareUrn(), 378

CryptProtectData, 308

CryptUnprotectData, 308

DeleteBackupHistory(), 378

Deny(), 379

DetachDatabase(), 379

DetachedDatabaseInfo(), 379

EnumAvailableMedia(), 379

EnumCollations(), 379

EnumDatabaseMirrorWitnessRoles(), 379

EnumDetachedDatabaseFiles(), 379

EnumDetachedLogFiles(), 379

EnumDirectories(), 379

EnumErrorLogs(), 379

EnumLocks(), 379

EnumMembers(), 379

EnumObjectPermissions(), 379

EnumPerformanceCounters(), 379

EnumProcesses(), 379

- EnumServerAttributes(), 379
  - EnumServerPermissions(), 380
  - EnumStartupProcedures(), 380
  - EnumWindowsDomainGroups(), 380
  - EnumWindowsGroupInfo(), 380
  - EnumWindowsUserInfo(), 380
  - Equals(), 380
  - GetActiveDBCConnectionCount(), 380
  - GetDefaultInitFields(), 380
  - GetHashCode(), 380
  - GetPropertyNames(), 380
  - GetSmoObject(), 380
  - GetType(), 380
  - Grant(), 380
  - Initialize(), 380
  - IsDetachedPrimaryFile(), 380
  - IsWindowsGroupMember(), 380
  - KillAllProcesses(), 381
  - KillDatabase(), 381
  - KillProcess(), 381
  - PingSqlServerVersion(), 381
  - ReadErrorLog(), 381
  - ReferenceEquals(), 381
  - Refresh(), 381
  - Revoke(), 381
  - SetDefaultInitFields(), 381
  - ToString(), 381
  - Microsoft
    - fora (forums), 396
    - .NET Framework, 341–343
    - SQL Server 2005 Web Site, 397
  - Microsoft Intermediate Language (MSIL), 344
  - Microsoft SQL Server 2000, 9
  - Microsoft SQL Server 2005
    - editions, 1, 11
    - installing, 2, 10, 15
  - MIN aggregate, 65
  - min() function, 236
  - mixed-mode authentication, 289
  - model database, 19
  - modifying
    - asymmetric keys, 315
    - columns, 161–162
    - database master key, 311
    - databases, 139
    - functions, 233
    - indexes, 264–265
    - logins, 290
    - roles, 303
    - service master key, 309
    - stored procedures, 204–206
    - tables, 161–162
    - users, 296
  - money data type, 144
  - Month() function, 239
  - msdb database, 19
  - MSIL (Microsoft Intermediate Language), 344
- **N** •
- namespaces, 342–343, 374
  - naming
    - case sensitivity, 142
    - columns (tables), 142–143
    - projects, 363
    - queries, 30
    - stored procedures, 203–204
    - tables, 142–143, 156–157
    - views, 189–190
  - naming conventions, 21–23, 142–143
  - navigating objects, 51
  - Nchar() function, 250
  - nchar(n) data type, 147
  - nested queries, 91–96
  - nested TRY...CATCH statements, 279–280
  - @NestLevel function, 237
  - .NET Framework, 341–343
  - network protocols, 37–38
  - Newid() function, 246
  - newsgroups, 396
  - nonclustered index, 257–258, 260
  - non-deterministic functions, 234
  - non-Unicode character data types, 146–147
  - normal forms, 131–136
  - Northwind database, 20, 44
  - NOT logical operator, 58
  - NOT NULL constraint, 163–168
  - ntext data type, 147
  - Nullif() function, 246
  - numeric data type, 144
  - numeric data types
    - approximate, 20, 145–146
    - exact, 20, 143–145
  - nvarchar(n) data type, 147
- **O** •
- Object Explorer pane, 27–28, 50
  - object models, 374–375
  - Object\_Id() function, 243
  - Object\_Name() function, 243
  - Objectproperty() function, 243
  - Objectpropertyex() function, 243
  - objects. *See* SMO (Server Management Objects)

- one-to-many relationship, 82–83, 129
- one-to-one relationship, 129
- ONLINE option (CREATE INDEX statement), 260–262
- OPEN MASTER KEY statement, 312
- OpenDataSource() function, 247
- opening
  - database master key, 312
  - Query Designer, 33
  - SQL Server Configuration Manager, 108
- Openquery() function, 247
- Openrowset() function, 248
- Openxml() function, 248
- operators
  - comparison operators, 56–58
  - logical operators, 57–59
- @@Options function, 237
- OR logical operator, 58
- ORDER BY clause
  - CREATE VIEW statement, 188–190
  - SELECT statement, 52, 64–65
- outer join
  - creating, 84–88
  - defined, 70
  - full outer join, 84
  - left outer join, 84, 87
  - right outer join, 84, 87
- Owner object, 18

## • p •

- @@Packet\_Errors function, 252
- @@Pack\_Received function, 252
- @@Pack\_Sent function, 252
- PAD\_INDEX option (CREATE INDEX statement), 261
- ParseName() function, 246
- Patindex() function, 250, 253
- pending changes, 122
- percent sign (%) wildcard character, 63
- performance improvements, 393
- permissions
  - ALTER SETTINGS, 346
  - assigning, 297–300
  - BACKUP DATABASE, 298
  - BACKUP LOG, 298
  - CREATE DATABASE, 298–299
  - CREATE DEFAULT, 298
  - CREATE FUNCTION, 298
  - CREATE PROCEDURE, 298
  - CREATE RULE, 298
  - CREATE TABLE, 298
  - CREATE VIEW, 298

- DELETE, 298
- denying, 301–302
- EXECUTE, 298
- granting, 297–300
- INSERT, 298, 301
- REFERENCES, 298
- revoking, 301
- roles, 302–304
- SELECT, 298–300
- stored procedures, 200
- UPDATE, 298
- Permissions() function, 246, 304
- Pi() function, 241
- PingSqlServerVersion() method, 381
- primary key
  - defined, 136
  - indexes, 258
  - joins, 70–71
  - PRIMARY KEY constraint, 70–71, 164
  - table relationships, 151–154, 157–159
- PRIMARY KEY keyword, 151–152
- privileges, 298, 302
- procedures. *See* stored procedures
- @@Procid() function, 244
- programming goals, 9–10
- programming skills
  - budgeting time, 392
  - choosing the right edition of SQL Server, 394
  - designing and testing on development hardware, 393
  - documenting projects, 391–392
  - establishing the correct hardware requirements, 394
  - handling many-to-many relationships, 393
  - improving performance, 393
  - listening to clients' needs, 391
  - testing, 393–394
  - thinking about long-term needs, 392
  - understanding relations, 392
- properties
  - databases, 137–138
  - Server object, 376–378
- public fora and newsgroups, 396
- pubs database, 20, 44

## • q •

- queries
  - Analysis Services Data Mining Expressions (DMX query), 30
  - Analysis Services Multidimensional Expressions (MDX) query, 30

- Analysis Services XMLA query, 30
- creating, 30–31, 33–35
- database engine queries, 30
- estimated execution plan, 35–36
- executing, 31
- naming, 30
- nested queries, 91–96
- renaming, 30
- SQL Server Mobile query, 30
- subqueries, 91–96
- XML data, 330–332
- Query Designer (SQL Server Management Studio)
  - joins, 77–83
  - opening, 33
  - query creation, 33–35
- Query Designer (Visual Studio 2005), 352, 356, 360–362
- Query Pane, 30
- Quotename() function, 250

## • R •

- Radians() function, 241
- RAISERROR statement, 272–274
- Rand() function, 241
- ReadErrorLog() method, 381
- real data type, 145
- RECONFIGURE statement, 345–346
- recursion, 102–104
- ReferenceEquals() method, 381
- REFERENCES constraint, 164
- REFERENCES permission, 298
- referential integrity, 71, 162
- Refresh() method, 381
- regenerating service master key, 309
- Registered Servers pane, 27–28
- regular identifiers, 21–23
- relations between tables
  - defined, 15, 69
  - importance of understanding, 392
  - referential integrity, 71
- relationships
  - creating, 151–154, 157–161
  - designing, 136
  - entity-relationship diagram, 129–130
  - many-to-many, 82–83, 129–130, 393
  - one-to-many, 82–83, 129
  - one-to-one, 129
- remote connections, 285
- removing. *See* deleting
- @RemServer function, 237

- Replace() function, 250
- Replicate() function, 250
- Reporting Services Web site, 398
- requirements phase, 127
- resource database, 19, 194
- RESTORE SERVICE MASTER KEY statement, 310
- restoring service master key, 310
- Reverse() function, 250
- Revoke() method, 381
- REVOKE statement, 301
- revoking permissions, 301
- Right() function, 250
- right outer join, 84, 87
- Role object, 18
- roles
  - application roles, 303–304
  - assigning, 302–303
  - creating, 303–304
  - database roles, 302–303
  - defined, 302
  - deleting, 303
  - dropping, 303
  - modifying, 303
- ROLLBACK TRANSACTION statement, 122
- root node, 257
- Round() function, 241
- @@Rowcount() function, 246
- Rowcount\_big() function, 246
- rows
  - deleting, 116–120
  - identifiers, 129
  - inserting, 111–116
  - table scan, 256, 258
  - timestamps, 148
- rowset functions, 247–248
- Rtrim() function, 250
- rules. *See* business rules
- running. *See* executing

## • S •

- sample databases
  - AdventureWorks database, 19, 43
  - installing, 42–44
  - Northwind database, 20, 44
  - pubs database, 20, 44
- scalar-valued function, 228
- Schema object, 18
- SCHEMA\_ID function, 304
- SCHEMA\_NAME function, 304

- schemas
  - changing ownership of, 293
  - creating, 292–293
  - default schema, 292
  - defined, 290–291
  - deleting, 293
  - dropping, 293
  - separation from users, 290–291
  - XML schema collections, 326–330
- Scope\_Identity() function, 246
- searching
  - SQL Server Books Online, 42
  - stored procedures, 195–197
- second normal form, 134–135
- security
  - catalog views, 286
  - Common Language Runtime (CLR), 344
  - configuring security settings, 284–286
  - encryption, 307–317
  - importance of, 10, 12–13
  - logins, 13
  - malicious code, 13
  - permissions, 297–302
  - privileges, 298, 302
  - SQL injection attacks, 13, 200, 207
  - SQL Server Surface Area Configuration tool, 284–286
  - stored procedures, 200, 207
  - suggested security measures, 284
  - users, 13
  - viewing information about, 286
  - views, 190–191
  - Windows Authentication, 28
- security functions, 248–249, 304–305
- SELECT permission, 298–300
- SELECT statement
  - asterisk (\*) wildcard, 54
  - forms of, 52
  - FROM clause, 52–55
  - GROUP BY clause, 52, 65–67
  - HAVING clause, 52, 67
  - joins, 73–89
  - optional clauses, 52
  - ORDER BY clause, 52, 64–65
  - square brackets [ ] , 52
  - uses of, 10, 47
  - WHERE clause, 52, 55–64
- selecting
  - servers, 106–107
  - templates, 32
- sequence counter, 113
- Server Explorer (Visual Studio 2005), 352–356
- Server Management Objects (SMO). *See* SMO (Server Management Objects)
- Server object, 17, 375–381
- server-level security catalog views, 286
- @@Servername function, 237
- SERVERPROPERTY() function, 31
- Server\_property() function, 246
- servers
  - connecting to, 27–29
  - selecting, 106–107
- service master key, 308–310
- @@Servicename function, 237
- services
  - managing, 37–38
  - SQL Server Agent service, 108–110
- session key, 308
- Sessionproperty() function, 246
- Session\_User function, 247, 304
- SetDefaultInitFields() method, 381
- SETUSER function, 305
- severity of errors, 268
- Sign() function, 241
- Sin() function, 241
- smalldatetime data type, 146
- smallint data type, 144
- smallmoney data type, 145
- SMO (Server Management Objects)
  - Certificate object, 18
  - Compatibility Level object, 18
  - Database object, 17
  - Default object, 18
  - DefaultFileGroup object, 18
  - DefaultSchema object, 18
  - defined, 17–18
  - LogFile object, 18
  - Login object, 17
  - namespaces, 374
  - navigating, 51
  - object model, 374–375
  - Owner object, 18
  - Role object, 18
  - Schema object, 18
  - Server object, 17, 375–381
  - SMO applications, 381–388
  - SqlAssembly object, 18
  - SQL-DMO (SQL Server Distributed Management Objects), 373
  - StoredProcedure object, 18
  - Table object, 17–19
  - Trigger object, 18
  - User object, 17
  - View object, 18
  - Visual Basic.NET, 373
  - Visual C#, 373
- Solution Explorer (Visual Studio 2005), 362–364

- `sort_in_tempdb` option (`CREATE INDEX statement`), 261
- `Soundex()` function, 250
- `Space()` function, 251
- `sp_addmessage` system stored procedure, 272
- `sp_adduser` system stored procedure, 294
- `sp_configure` system stored procedure, 345–346
- specifying template parameter values, 33
- `sp_setapprole` system stored procedure, 304
- SQL Editor toolbar, 33–34
- SQL injection attacks, 13, 200, 207
- SQL Management Object method of transferring data, 107
- SQL Server Agent service, 108–110
- SQL Server Authentication, 27
- SQL Server Books Online, 10, 39–42, 100, 395–396
- SQL Server Configuration Manager, 37–38, 50, 108
- SQL Server Developer Center Web site, 397
- SQL Server Distributed Management Objects (SQL-DMO), 373
- SQL Server Management Studio
  - Code Editor, 30
  - Copy Database Wizard, 105–110
  - launching, 26–27
  - New Query button, 30
  - Object Explorer pane, 27–28, 50
  - Query Designer, 33–35, 77–83
  - Query Pane, 30
  - Registered Servers pane, 27–28
  - SQL Editor toolbar, 33–34
  - Standard toolbar, 29–30
  - starting, 26–27
  - Summary tab, 28
  - Template Explorer tool, 31–32
  - Transact-SQL (T-SQL), 12, 25–26
  - versions, 26
- SQL Server Management Studio Express, 26
- SQL Server Mobile query, 30
- SQL Server Surface Area Configuration tool, 284–286, 346–347
- SQL Server 2000, 9
- SQL Server 2005
  - editions, 11, 394
  - installing, 2, 10, 15
- SQL Server 2005 level encryption, 308–309
- SQL Server 2005 TechCenter Web site, 397
- SQL Server 2005 Web Site, 397
- `SqlAssembly` object, 18
- SQLCMD utility
  - checking installation status, 36
  - help, 37
  - launching, 36–37
  - starting, 36–37
  - Transact-SQL (T-SQL), 25
- SQL-DMO (SQL Server Distributed Management Objects), 373
- SQL-SMO. *See* SMO (Server Management Objects)
- `sql_variant` data type, 148
- `Sql_Variant_Property()` function, 244
- `Sqrt()` function, 241
- square brackets ([ ] )
  - `SELECT` statement, 52
  - wildcard characters, 63
- `Square()` function, 241
- Standard edition, 11
- starting
  - SQL Server Books Online, 40
  - SQL Server Configuration Manager, 37
  - SQL Server Management Studio, 26–27
  - SQLCMD utility, 36–37
- `STATISTICS_NORECOMPUTE` option (`CREATE INDEX statement`), 262
- `Stats_Date()` function, 247
- `stdev()` function, 236
- stored procedures
  - Common Language Runtime, 198–199
  - `CREATE PROCEDURE` permission, 298
  - creating, 200–201
  - defined, 193
  - deleting, 204–205
  - differences from functions, 199, 228
  - differences from triggers, 211–212
  - dropping, 204–205
  - executing, 199, 202–204
  - extended, 199
  - filtering, 195–197
  - modifying, 204–206
  - naming, 203–204
  - parameters, 200, 206–207
  - permissions, 200
  - searching, 195–197
  - security, 200, 207
  - `sp_addmessage`, 272
  - `sp_adduser`, 294
  - `sp_configure`, 345–346
  - `sp_setapprole`, 304
  - system, 194–197
  - Transact-SQL (T-SQL), 197–198
  - user-defined, 194, 197–199
  - uses of, 193–194, 199–200
  - viewing, 194–195

StoredProcedure object, 18  
 Str() function, 251  
 string data types, 20  
 string functions, 249–251  
 Stuff() function, 251  
 subqueries, 91–96  
 Substring() function, 251  
 SUM aggregate, 65  
 sum() function, 236, 253–254  
 supported libraries, 342–343  
 supported namespaces, 342–343  
 Surface Area Configuration tool,  
   284–286, 346–347  
 surrogate keys, 113  
 Suser\_Id function, 305  
 Suser\_Name function, 305  
 Suser\_Sid() function, 249, 305  
 Suser\_Sname() function, 249, 305  
 symmetric keys, 312–313, 315–316  
 sys.asymmetric\_keys catalog view, 313  
 sys.certificates catalog view, 313–314  
 sys.credentials catalog view, 313  
 sys.crypt\_properties catalog  
   view, 313–314  
 sys.fn\_builtin\_permissions  
   function, 305–306  
 sys.key\_encryptions catalog view, 313  
 sys.symmetric\_keys catalog view, 313  
 system databases, 19, 194  
 system functions. *See* functions (system)  
 system statistical functions, 251–252  
 system stored procedures  
   accessing, 194–195  
   defined, 194  
   filtering, 195–197  
   sp\_admessage, 272  
   sp\_adduser, 294  
   sp\_configure, 345–346  
   sp\_setapprole, 304  
 SYSTEM\_USER function, 305  
 System\_User function, 247

## • T •

table columns. *See* columns  
 table data type, 148  
 Table Designer (Visual Studio 2005),  
   352, 356–360  
 Table object, 17–19  
 table rows. *See* rows  
 table scan, 256, 258  
 table-level constraints, 179–180

tables  
   aliases, 76–77  
   CREATE TABLE permission, 298–299  
   creating, 149–150, 154–158  
   modifying, 161–162  
   naming, 142–143, 156–157  
   referential integrity, 71  
   relations, 15, 69, 392  
   relationships, 136, 151–154, 157–161  
 table-valued function, 228  
 Tan() function, 241  
 tempdb database, 19  
 Template Explorer tool, 31–32  
 templates, 31–33  
 terminating transactions, 122  
 testing  
   database design, 131  
   development hardware, 393  
   importance of, 394  
 text data type, 147  
 text functions, 253  
 Textptr() function, 253  
 @@Textsize function, 237  
 Textvalid() function, 253  
 third normal form, 135–136  
 time and date functions, 238–239  
 time-related data types, 20, 146  
 timestamp data type, 148  
 @@Time\_ticks function, 252  
 tinyint data type, 145  
 ToString() method, 381  
 @@Total\_Errors function, 252  
 @@Total\_Read function, 252  
 @@Total\_Write function, 252  
 @@Tranccount function, 247  
 transactions, 122  
 Transact-SQL (T-SQL)  
   alternatives, 24  
   error-handling, 268  
   flexibility of, 1  
   functions, 227–228  
   limitations of, 13  
   SQL Server Management Studio, 12, 25–26  
   SQLCMD utility, 25  
   stored procedures, 197–198  
 transferring data, 107–108  
 transitive dependency, 135  
 Trigger object, 18  
 triggers  
   AFTER trigger, 209–210, 213–220  
   creating, 209–210, 214–215  
   data integrity, 209, 212  
   DDL triggers, 209–213, 223–224

debugging, 224–225  
 defined, 11, 209  
 differences from stored procedures,  
   211–212  
 DML triggers, 209–213  
 executing, 211  
 INSTEAD OF trigger, 209–210, 213, 220–223  
 triggering action, 213  
 uses of, 209, 212  
 troubleshooting server connection  
   problems, 29  
 TRY...CATCH construct, 267, 271–272,  
   274–280  
 T-SQL (Transact-SQL). *See* Transact-SQL  
   (T-SQL)  
 typecasting data types, 96–100  
 typed XML, 323–326  
 Typeproperty() function, 244  
 types of data. *See* data types

## • U •

underscore (\_) wildcard character, 63  
 Unicode character data types, 21, 147  
 Unicode() function, 251  
 UNIQUE constraint, 163–164, 170–173  
 unique index, 258  
 uniqueidentifier data type, 149  
 unsupported namespaces, 343  
 untyped XML, 323–326  
 Update() function, 247  
 UPDATE permission, 298  
 UPDATE statement, 120–121  
 updates  
   indexes, 192  
   UPDATE statement, 120–121  
   views, 191–192  
   XML data, 333–334  
 Upper() function, 251  
 USE statement, 53, 149–150  
 User() function, 249  
 User object, 17  
 user-defined functions  
   Common Language Runtime, 227–228  
   CREATE FUNCTION permission, 298  
   creating, 227–233  
   defined, 227  
   deleting, 233–234  
   differences from stored procedures, 199,  
     228  
   dropping, 233–234  
   modifying, 233

  scalar-valued function, 228  
   table-valued function, 228  
   Transact-SQL, 227–228  
 user-defined procedures, 194, 197–199  
 USER\_ID function, 305  
 User\_Id() function, 249  
 User\_Name() function, 247, 249, 305  
 users  
   adding, 293–296  
   associating with logins, 295–296  
   changing the default schema, 292  
   creating, 293–296  
   defined, 287, 290, 293  
   modifying, 296  
   permissions, 297–302  
   privileges, 298  
   roles, 302–304  
   security, 13  
   separation from schemas, 290–291

## • V •

var() function, 236  
 varbinary(n) data type, 148  
 varchar(n) data type, 147  
 variants, 21, 148  
 varp() function, 236  
 @@Version function, 237  
 versions  
   SQL Server Management Studio, 26  
   Visual Studio 2005, 352  
 View Designer (Visual Studio 2005), 352, 356  
 View object, 18  
 viewing  
   databases, 48–50  
   list of system functions, 234–235  
   login information, 288  
   logins, 287  
   security information, 286  
   stored procedures, 194–195  
 views  
   CREATE VIEW permission, 298  
   creating, 181–180  
   defined, 181–182, 193  
   indexed, 192, 259  
   inserts, 191–192  
   joins, 185–188  
   naming, 189–190  
   security, 190–191  
   updates, 191–192  
   uses of, 181–182  
 Visual Basic.NET, 343–344, 373

- Visual C#, 343–344, 373
- Visual Studio 2005
  - Business Intelligence Development Studio (BIDS), 351
  - Database Diagram Designer, 351, 356–357
  - Database Explorer, 352
  - downloads, 351
  - installing, 351
  - projects, 363–372
  - Query Designer, 352, 356, 360–362
  - Server Explorer, 352–356
  - Solution Explorer, 362–364
  - Table Designer, 352, 356–360
  - uses for, 39, 351
  - variants, 26
  - versions, 352
  - View Designer, 352, 356
  - visual database tools, 351–352, 356–362

## • W •

- well-formed XML document, 322–323
- WHERE clause
  - CREATE VIEW statement, 183
  - DELETE statement, 116–120
  - joins, 75–76
  - SELECT statement, 52, 55–64
  - UPDATE statement, 120–121
- wildcard characters
  - \* (asterisk), 54, 60, 63
  - [^] (caret inside square brackets), 63
  - escaping, 60
  - % (percent sign), 63
  - [ ] (square brackets), 63
  - \_ (underscore), 63
- Windows Authentication, 27–28
- Windows Forms project, 364–372
- Windows level encryption, 308
- Workgroup edition, 11

## • X •

- XML
  - deleting, 334–335
  - FOR XML clause, 323, 336–340
  - indexes, 335–336
  - inserts, 333–334
  - queries, 330–332
  - typed XML, 323–326
  - untyped XML, 323–326
  - updates, 333–334
  - uses of, 322
  - well-formed XML document, 322–323
  - xml data type, 13, 15, 21, 149, 322
- XML index, 259
- XML schema collections, 326–330
- XPath (XML Path language), 329
- XQuery language, 323, 330–335
- XSD schema document, 326–328

## • Y •

- Year () function, 239