

# Index

## NUMERICS

- 1NF—first normal form, 32
- 2NF—second normal form, 32
- 3NF—third normal form, 32

## A

- ABS() function, 193, 475**
- ABSOLUTE (reserved word), 462, 464**
- ACID test, 254**
- ACOS() function, 193, 475**
- ACTION (reserved word), 462, 464**
- Active Directory procedures**
  - sp\_ActiveDirectory\_Obj, 483
  - sp\_ActiveDirectory\_SCP, 483
- Active Server Pages (ASP), 410**
- ADA (reserved word), 462**
- ADD (reserved word), 460, 462**
- adding records**
  - defaults, inserting, 259–261
  - INSERT statement, 257–258
  - NULL values, inserting, 259–261
  - overview, 256
  - rows from another table, inserting, 261–262
  - SELECT INTO statement, 262
  - SELECT statement, 258
  - stored procedures, managing inserts using, 262–263

- ad-hoc queries, 350
- ADMIN (reserved word), 464**
- ADO (ActiveX Data Objects), 402**
- Adventure Works Cycles database**
  - installation of, 45
  - overview, 44–45
- AFTER (reserved word), 464**
- aggregate functions**
  - AVG() function, 165–166, 208, 209–210
  - COUNT() function, 166, 208–209, 472
  - COUNT\_BIG() function, 208, 472
  - grouping data, 213–228
  - GROUPING() function, 472
  - list of, 472
  - MAX() function, 166–167, 208, 472
  - MIN() function, 166–167, 208, 472
  - overview, 164–165, 207–208
  - statistical functions, 210–212
  - STDEV() function, 208, 212, 472
  - STDEVP() function, 208, 212, 472
  - subqueries, used in, 286
  - SUM() function, 167, 208, 209, 472
  - system-supplied aggregate functions used to
    - summarize column values, list of, 208
  - user-defined aggregate function, 212
  - VAR() function, 208, 210–211, 472
  - VARP() function, 208, 211, 472
- AGGREGATE (reserved word), 464**
- ALIAS (reserved word), 464**

## ALL (reserved word)

---

**ALL (reserved word)**, 460, 462  
**ALLOCATE (reserved word)**, 462, 464  
**ALTER (reserved word)**, 460, 462  
**ALTER statement**, 117, 322  
**ALTER TABLE command**, 457  
altering database objects, 322  
**AND operator**, 141, 319  
**AND (reserved word)**, 460, 462  
**ANSI standard**, 2  
**ANY (reserved word)**, 460, 462  
application integration, 444–447  
application of normalization rules, 34–42  
application programming interface (API), 401  
application programming models  
    budget constraints, 400  
    categories of, 398–399  
    client/server database solutions, 401–404  
    currentness of data, 400  
    delivery time constraints, 400  
    design patterns, 397  
    desktop database applications, 400–401  
    infrastructure investment, 399  
    location of users, 399  
    multi-system integrated solutions, 411–412  
    multi-tier Web service solutions, 410–411  
    n-tier component solutions, 405–409  
    overview, 397–398  
    security requirements, 400  
    selection of, 398–399  
    storage space, 400  
    system availability requirements, 400  
    users accessing data, number of, 399  
    Web server application, 409–410  
**APP\_NAME() function**, 200, 480  
**ARE (reserved word)**, 462, 464  
arguments, 161  
**ARRAY (reserved word)**, 464  
**AS (reserved word)**, 460, 462  
**ASC (reserved word)**, 460, 462  
**ASCII() function**, 181–184, 478  
**ASIN() function**, 193, 475  
**ASP (Active Server Pages)**, 410  
**ASSERTION (reserved word)**, 462, 464

**AT (reserved word)**, 462, 465  
**ATAN() function**, 193, 475  
**ATN2() function**, 193, 475  
**AUTHORIZATION (reserved word)**, 460, 462  
auto-commit transactions, 254  
**AVG() function**, 165–166, 209–210, 472  
**AVG (reserved word)**, 462

## B

**BACKUP (reserved word)**, 460  
Ballmer, Steve (Microsoft), 24  
batch directives, 66  
**BCNF (Boyce-Codd Normal Form)**, 32–33  
**BEFORE (reserved word)**, 465  
**BEGIN (reserved word)**, 460, 462  
*Beginning SQL Server 2005 Programming*  
    (Viera), 1  
benchmarking subqueries, 291–293  
best practices for subqueries, 291–293  
**BETWEEN operator**, 145  
**BETWEEN (reserved word)**, 460, 462  
biased estimate of variance, 211  
**BINARY (reserved word)**, 465  
**BINARY\_CHECKSUM() function**, 473  
**BIT (reserved word)**, 462, 465  
**BIT\_LENGTH (reserved word)**, 462  
**BLOB (reserved word)**, 465  
**Books Online**  
    OSQL utility, 70–72  
    overview, 49, 69  
    updating, 69  
**Bool, George (logical gates)**, 141  
**BOOLEAN (reserved word)**, 465  
**BOTH (reserved word)**, 462, 465  
Boyce, Raymond F. (relational database design), 31  
**Boyce-Codd Normal Form (BCNF)**, 32–33  
**BREADTH (reserved word)**, 465  
**BREAK (reserved word)**, 460  
breaking normalization rules, 42  
bridge table, 39  
**BROWSE (reserved word)**, 460

**budget constraints, 400**  
**buffer cache, 44**  
**BULK (reserved word), 460**  
**business cases for subqueries, 293–296**  
**Business Intelligence Development Studio, 73, 97–98**  
**business logic, processing, 378–380**  
**BY (reserved word), 460, 462**

**C**

**calculated columns, 135–137**  
**CALL (reserved word), 465**  
**CALLER option, 344**  
**Camel Case naming standard, 112**  
**CASCADE (reserved word), 460, 462**  
**CASCADED (reserved word), 462, 465**  
**CASE command, 382–384, 454–455**  
**CASE (reserved word), 460, 462**  
**CAST() function, 169–171, 473**  
**CAST (reserved word), 462, 465**  
**catalog procedures**

- sp\_column\_privileges, 483
- sp\_column\_privileges\_ex, 483
- sp\_columns, 483
- sp\_columns\_ex, 483
- sp\_databases, 484
- sp\_fkeys, 484
- sp\_pkeys, 484
- sp\_server\_info, 484
- sp\_special\_columns, 484
- sp\_sproc\_columns, 484
- sp\_statistics, 484
- sp\_stored\_procedures, 484
- sp\_table\_privileges, 484
- sp\_table\_privileges\_ex, 484
- sp\_tables, 484

**CATALOG (reserved word), 462, 465**  
**CATALOG\_NAME view, 516**  
**catalogs, managing and populating, 304–313**  
**categories of application programming models, 398–399**  
**CEILING() function, 193, 475**

**CHAR() function, 182, 478**  
**CHAR (reserved word), 462, 465**  
**CHARACTER (reserved word), 463, 465**  
**CHARACTER\_LENGTH (reserved word), 463**  
**CHARACTER\_MAXIMUM\_LENGTH view, 510, 512, 514, 515**  
**CHARACTER\_OCTET\_LENGTH view, 510, 512, 514, 515**  
**CHARACTER\_SET\_CATALOG view, 511, 512, 514, 515**  
**CHARACTER\_SET\_NAME view, 511, 512, 514, 515**  
**CHARACTER\_SET\_SCHEMA view, 511, 512, 514, 515**  
**CHARINDEX() function, 184–185, 478**  
**CHAR\_LENGTH (reserved word), 462**  
**check constraints, 324, 333–334**  
**CHECK (reserved word), 460, 463**  
**CHECK\_CONSTRAINTS views**

- CONSTRAINT\_CATALOG view, 509
- CONSTRAINT\_CLAUSE view, 509
- CONSTRAINT\_NAME view, 509
- CONSTRAINT\_SCHEMA view, 509

**CHECK\_OPTION view, 518**  
**checkpoint, 273–274**  
**CHECKPOINT (reserved word), 460**  
**checksum functions**

- BINARY\_CHECKSUM() function, 473
- CHECKSUM() function, 473
- CHECKSUM\_AGG() function, 473

**CLASS (reserved word), 465**  
**client/server database solutions**

- application programming models, 401–404
- event-handling code, 403–404
- overview, 401–402
- user interface, 402–403

**client/server processes, 42**  
**CLOB (reserved word), 465**  
**CLOSE (reserved word), 460, 463**  
**clustered index, 324**  
**CLUSTERED (reserved word), 460**  
**COALESCE() function, 200, 202–203, 480**  
**COALESCE (reserved word), 460, 463**

**Codd, Edger (relational database design), 31**

**COLLATE (reserved word), 460, 463**

**COLLATION (reserved word), 463, 465**

**COLLATION\_CATALOG view, 511, 512, 514, 515**

**COLLATION\_NAME view, 511, 512, 514, 515**

**COLLATIONPROPERTY() function, 200, 480**

**COLLATION\_SCHEMA view, 511, 512, 514, 515**

**COL\_LENGTH() function, 194, 476**

**COL\_NAME() function, 194, 476**

**column aliasing, 134–135**

**COLUMN (reserved word), 460, 463**

**COLUMN\_DEFAULT view, 510, 514**

**COLUMN\_DOMAIN\_USAGE views**

COLUMN\_NAME view, 510

DOMAIN\_CATALOG view, 509, 510

DOMAIN\_NAME view, 509

DOMAIN\_SCHEMA view, 509

TABLE\_CATALOG view, 509

TABLE\_NAME view, 510

TABLE\_SCHEMA view, 510

**COLUMN\_NAME view, 510, 511, 513, 514, 518**

**COLUMN\_PRIVILEGES views**

COLUMN\_NAME view, 510

GRANTEE view, 510

GRANTOR view, 510

IS\_GRANTABLE view, 510

PRIVILEGE\_TYPE view, 510

TABLE\_CATALOG view, 510

TABLE\_NAME view, 510

TABLE\_SCHEMA view, 510

**COLUMNPROPERTY() function, 194, 476**

**columns**

choosing, 128–132

overview, 5, 8

**COLUMNS views**

CHARACTER\_MAXIMUM\_LENGTH view, 510

CHARACTER\_OCTET\_LENGTH view, 510

CHARACTER\_SET\_CATALOG view, 511

CHARACTER\_SET\_NAME view, 511

CHARACTER\_SET\_SCHEMA view, 511

COLLATION\_CATALOG view, 511

COLLATION\_NAME view, 511

COLLATION\_SCHEMA view, 511

COLUMN\_DEFAULT view, 510

DATA\_TYPE view, 510

DATETIME\_PRECISION view, 511

DOMAIN\_CATALOG view, 511

DOMAIN\_NAME view, 511

DOMAIN\_SCHEMA view, 511

IS\_NULLABLE view, 510

NUMERIC\_PRECISION view, 510

NUMERIC\_PRECISION\_RADIX view, 510

NUMERIC\_SCALE view, 511

ORDINAL\_POSITION view, 510

**command syntax reference**

ALTER TABLE command, 457

CASE command, 454–455

COMPUTE BY clause, 454

COMPUTE clause, 454

CREATE DATABASE command, 457

CREATE DEFAULT command, 457

CREATE PARTITION FUNCTION command, 459

CREATE PARTITION SCHEME command, 459

CREATE PROCEDURE command, 457

CREATE RULE command, 458

CREATE SCHEMA command, 458–459

CREATE TABLE command, 458

CREATE TRIGGER command, 458

CREATE VIEW command, 458

DECLARE @ local\_variable command, 455–456

DELETE command, 455

EXCEPT command, 453

FOR clause, 454

FROM command, 451–452

GROUP BY command, 452–453

HAVING command, 453

INSERT command, 455

INTERSECT command, 453

LIKE operator, 456–457

object reference, 449

ODBC reserved words, 462–464

OPTION clause, 454

ORDER BY command, 454

reserved words, 459–466

script comment conventions, 459

SELECT command, 450–451

- SELECT INTO command, 451
- SET operator, 456
- TOP command, 451
- UNION command, 453
- UPDATE command, 455
- WHERE command, 452
- WITH command, 450
- command-line scripting, 99–101**
- comments**
  - comment block, 459
  - in-line comment, 459
  - overview, 112–114
  - single-line comment, 459
- COMMIT (reserved word), 460, 463**
- comparison operators, 140–148**
- COMPLETION (reserved word), 465**
- complexity, hiding, 364–365**
- component transaction management, 409**
- COMPUTE BY clause, 228, 454**
- COMPUTE clause, 225–228, 454**
- COMPUTE (reserved word), 460**
- conceptual design, 27–28**
- conditional logic, 380–382**
- configuration variables**
  - @@DATEFIRST, 467
  - @@DBTS, 467
  - @@ERROR, 167–168
  - @@LANGID, 467
  - @@LANGUAGE, 468
  - @@LOCK\_TIMEOUT, 468
  - @@MAX\_CONNECTIONS, 468
  - @@MAX\_PRECISION, 468
  - @@MICROSOFTVERSION, 468
  - @@NESTLEVEL, 468
  - @@OPTIONS, 468–469
  - overview, 167
  - @@REMSERVER, 469
  - @@SERVERNAME, 469
  - @@SERVICENAME, 168, 469
  - @@TEXTSIZE, 469
  - @@TOTAL\_ERRORS, 168
  - @@TOTAL\_READ, 168
  - @@VERSION, 169, 469
- Configure SQL XML Support in IIS option, 49**
- CONNECT (reserved word), 463, 465**
- connection pooling, 408**
- CONNECTION (reserved word), 463, 465**
- connections, 16**
- @@CONNECTIONS, 204, 470**
- Consensus Soundex (Russell & Odell), 302**
- CONSTRAINT (reserved word), 461, 463**
- CONSTRAINT\_CATALOG view, 509, 511, 512, 513, 517**
- CONSTRAINT\_CLAUSE view, 509**
- CONSTRAINT\_COLUMN views**
  - COLUMN\_NAME view, 511
  - CONSTRAINT\_CATALOG view, 511
  - CONSTRAINT\_NAME view, 511
  - CONSTRAINT\_SCHEMA view, 511
  - TABLE\_CATALOG view, 511
  - TABLE\_NAME view, 511
  - TABLE\_SCHEMA view, 511
- CONSTRAINT\_COLUMN\_USAGE view, 511**
- CONSTRAINT\_NAME view, 509, 511, 512, 513, 517**
- constraints, 331–336**
- CONSTRAINTS (reserved word), 463, 465**
- CONSTRAINT\_SCHEMA view, 509, 511, 512, 513, 517**
- CONSTRAINT\_TABLE\_USAGE views**
  - CONSTRAINT\_CATALOG view, 511
  - CONSTRAINT\_NAME view, 511
  - CONSTRAINT\_SCHEMA view, 511
  - TABLE\_CATALOG view, 511
  - TABLE\_NAME view, 511
  - TABLE\_SCHEMA view, 511
- CONSTRAINT\_TYPE view, 517**
- CONSTRUCTOR (reserved word), 465**
- CONTAINS predicate, 314–317**
- CONTAINS (reserved word), 461**
- CONTAINSTABLE predicate, 317–318**
- CONTAINSTABLE (reserved word), 461**
- CONTINUE (reserved word), 461, 463**
- conversion functions**
  - CAST() function, 169–171, 473
  - CONVERT() function, 161, 172–174, 473

### **conversion functions (continued)**

CURSOR\_STATUS() function, 474  
overview, 169, 473  
STR() function, 174–175

**CONVERT (reserved word), 461, 463**

**correlated subqueries, 290–291**

**CORRESPONDING (reserved word), 463, 465**

**COS() function, 193, 475**

**COT() function, 193, 475**

**COUNT() function, 166, 208, 208–209, 472**

**COUNT (reserved word), 463**

**COUNT\_BIG() function, 208, 472**

**@@CPU\_BUSY, 204, 470**

**CREATE DATABASE command, 457**

**CREATE DEFAULT command, 457**

**CREATE FUNCTION statement, 344**

**CREATE PARTITION FUNCTION command, 459**

**CREATE PARTITION SCHEME command, 459**

**CREATE PROCEDURE command, 340–343, 457**

**CREATE (reserved word), 461, 463**

**CREATE RULE command, 458**

**CREATE SCHEMA command, 458–459**

**CREATE statement, 117, 322**

**CREATE TABLE command, 325–330, 458**

**CREATE TRIGGER command, 458**

**CREATE VIEW command, 336–339, 458**

**CREATED view, 516**

**cross joins, 243**

**CROSS (reserved word), 461, 463**

**CUBE (reserved word), 465**

**CURRENT (reserved word), 461, 463**

**CURRENT\_DATE (reserved word), 461, 463**

**currentness of data, 400**

**CURRENT\_PATH (reserved word), 465**

**CURRENT\_ROLE (reserved word), 465**

**CURRENT\_TIME (reserved word), 461, 463**

**CURRENT\_TIMESTAMP() function, 200, 480**

**CURRENT\_TIMESTAMP (reserved word), 461, 463**

**CURRENT\_USER() function, 200, 480**

**CURRENT\_USER (reserved word), 461, 463**

**cursor functions and variables**

@@CURSOR\_ROWS, 176  
CURSOR\_STATUS() function, 176

@@FETCH\_STATUS, 176  
overview, 175–176, 469, 474

### **cursor management procedures**

sp\_add\_data\_file\_recover\_suspect\_db, 485  
sp\_addextendedproc, 485  
sp\_addextendedproperty, 485  
sp\_add\_log\_file\_recover\_suspect\_db, 485  
sp\_add\_maintenance\_plan, 489  
sp\_add\_maintenance\_plan\_db, 489  
sp\_add\_maintenance\_plan\_job, 489  
sp\_addmessage, 485  
sp\_addtype, 485  
sp\_addumpdevice, 485  
sp\_altermessage, 485  
sp\_attach\_db, 485  
sp\_attach\_single\_file\_db, 485  
sp\_autostats, 485  
sp\_bindefault, 485  
sp\_bindrule, 485  
sp\_bindsession, 485  
sp\_certify\_removable, 485  
sp\_configure, 485  
sp\_create\_removable, 486  
sp\_createstats, 486  
sp\_cursor\_list, 484  
sp\_cycle\_errorlog, 486  
sp\_datatype\_info, 486  
sp\_dbcmptlevel, 486  
sp\_dboption, 486  
sp\_dbremove, 486  
sp\_delete\_backuphistory, 486  
sp\_delete\_maintenance\_plan, 489  
sp\_delete\_maintenance\_plan\_db, 489  
sp\_delete\_maintenance\_plan\_job, 489  
sp\_depends, 486  
sp\_describe\_cursor, 484  
sp\_describe\_cursor\_columns, 484  
sp\_describe\_cursor\_tables, 484  
sp\_detach\_db, 486  
sp\_dropdevice, 486  
sp\_dropextendedproc, 486  
sp\_dropextendedproperty, 486  
sp\_dropmessage, 486  
sp\_droptype, 486

sp\_executesql, 486  
 sp\_getapplock, 486  
 sp\_getbindtoken, 487  
 sp\_help, 487  
 sp\_helpconstraint, 487  
 sp\_helpdb, 487  
 sp\_helpdevice, 487  
 sp\_helpextendedproc, 487  
 sp\_helpfile, 487  
 sp\_helpfilegroup, 487  
 sp\_helpindex, 487  
 sp\_helplanguage, 487  
 sp\_help\_maintenance\_plan, 489  
 sp\_helpserver, 487  
 sp\_helpsort, 487  
 sp\_helpstats, 487  
 sp\_helpstext, 487  
 sp\_helptrigger, 487  
 sp\_indexoption, 487  
 sp\_invalidate\_textptr, 487  
 sp\_lock, 488  
 sp\_monitor, 488  
 sp\_procoption, 488  
 sp\_recompile, 488  
 sp\_refreshview, 488  
 sp\_releaseapplock, 488  
 sp\_rename, 488  
 sp\_renamedb, 488  
 sp\_resetstatus, 488  
 sp\_serveroption, 488  
 sp\_setnetname, 488  
 sp\_settriggerorder, 488  
 sp\_spaceused, 488  
 sp\_tableoption, 488  
 sp\_unbindefault, 488  
 sp\_unbindrule, 488  
 sp\_updateextendedproperty, 488  
 sp\_updatestats, 488  
 sp\_validname, 488  
 sp\_who, 488

### **CURSOR (reserved word), 461, 463**

#### **cursor variables**

@@CURSOR\_ROWS, 469  
 @@FETCH\_STATUS, 469

### **@@CURSOR\_ROWS, 176, 469**

#### **cursors**

creating, 297–298  
 FETCH NEXT statement, 297–298  
 INSENSITIVE option, 297  
 management procedures, 484  
 navigating, 297–298  
 overview, 296  
 rowset *versus* cursor operations, 296–297

### **CURSOR\_STATUS() function, 176, 474**

### **CYCLE (reserved word), 465**

## **D**

### **DAO (Data Access Objects), 401**

### **data caching, 443–444**

#### **Data Control Language (DCL)**

DENY statement, 123, 347  
 GRANT statement, 123, 346–347  
 overview, 123–124  
 REVOKE statement, 123, 347

#### **Data Definition Language (DDL)**

ALTER statement, 117, 322  
 CREATE statement, 117, 322  
 DROP statement, 117, 322  
 objects, altering, 121–122  
 objects, dropping, 122–123  
 overview, 117–118  
 scripting practices, 120–123  
 stored procedure, creation of, 118–119  
 table, creation of, 117–118  
 trigger, creation of, 119  
 user-defined function, creation of, 119–120  
 view, creation of, 118

### **data, information transformed into, 33–34**

#### **data integrity**

connections, 16  
 described, 16  
 locks, 16–18  
 transactions, 16

#### **Data Manipulation Language (DML)**

Camel Case naming standard, 112  
 comments, 112–114  
 Hungarian Notation naming standard, 112

## Data Manipulation Language (DML) (continued)

---

### Data Manipulation Language (DML) (continued)

- in-line comments, 113–114
- layers, queries having, 108
- long names naming standard, 112
- lower-case, delimited naming standard, 112
- naming conventions, 110–112
- overview, 107
- Pascal Case naming standard, 112
- query syntax, 109–110
- row-based operations, 109
- script generation, 115
- script management, 116
- SELECT statement, 108
- set-based operations, 109
- templates, 114–115
- version control, 116–117

### data objects

- CALLER option, 344
- check constraint, 333–334
- constraints, 331–336
- CREATE FUNCTION statement, 344
- CREATE PROCEDURE statement, 340–343
- CREATE TABLE statement, 325–330
- CREATE VIEW statement, 336–339
- defaults, 329–330
- DENY statement, 347
- EXECUTE AS option, 344
- foreign key constraint, 334–336
- GRANT statement, 346–347
- IDENTITY property, 326–329
- indexed views, 339–340
- naming, 322–325
- nullability, 325–326
- overriding constraints, 336
- OWNER option, 344
- parameters, 341–342
- primary key constraint, 331–333
- REVOKE statement, 347
- SCHEMABINDING option, 338
- securing, 344–347
- SELF option, 344
- unique constraint, 333
- UniquelIdentifier type, 330–331

- USER\_NAME option, 344
- WITH CHECK option, 338
- WITH ENCRYPTION option, 338–339, 342
- WITH RECOMPILE statement, 343

### DATA (reserved word), 465

#### data retrieval

- SELECT statement, 128–156
- storage and retrieval, 127–128

#### data transactions

- ACID test, 254
- adding records, 256–263
- auto-commit transactions, 254
- checkpoint, 273–274
- consistency of, 254
- criteria for, 254
- defaults, inserting, 259–261
- DELETE statement, 269–270
- durability of, 254
- explicit transactions, 254, 273–276
- filtering updates, 264–266
- implicit transactions, 254
- INSERT statement, 257–258
- isolation of, 254
- logged operations, 256
- modifying records, 263–269
- multiple tables, deleting records based on, 271–272
- multiple tables, updating rows based on, 266–267
- NULL values, inserting, 259–261
- overview, 253–254
- removing records, 269–273
- rows from another table, inserting, 261–262
- SELECT INTO statement, 262
- SELECT statement, 258
- stored procedures, managing inserts using, 262–263
- stored procedures, updating records using, 267–269
- transaction logs, 255–256
- TRUNCATE TABLE statement, 272–273
- types of, 254
- unity of, 254

- UPDATE statement, 264–269
- views, updating using, 267
- WHERE clause, 270
- Data Transformation Services (DTS), 256**
- data types, 8–13**
- database engine procedures, 485–488**
- database maintenance procedures, 489**
- database management system (DBMS)**
  - described, 4
  - hierarchical database management systems (HDBMS), 4
  - network database management systems (NDBMS), 4
  - object-oriented database management systems (ODBMS), 4
  - object-relational database management systems (ORDBMS), 4–5
  - relational database management systems (RDBMS), 4
- database objects**
  - altering, 322
  - creating, 322
  - dropping, 322
  - overview, 321–322
- DATABASE (reserved word), 461**
- Database Tuning Advisor, 73**
- database..object, 449**
- DATABASEPROPERTY() function, 194, 476**
- DATABASEPROPERTYEX() function, 194, 476**
- database.schema.object, 449**
- DATALength() function, 200, 203–204, 480**
- datasets, 418–423**
- DATA\_TYPE view, 510, 512, 514, 515**
- date and time, altering, 456**
- date functions**
  - DATEADD() function, 176–178, 474
  - DATEDIFF() function, 178–179, 474
  - DATENAME() function, 180, 474
  - DATEPART() function, 180, 474
  - DAY() function, 181, 474
  - GETDATE() function, 180–181, 474
  - GETUTCDATE() function, 180–181, 474
  - MONTH() function, 181, 474
  - overview, 176
  - YEAR() function, 181, 474
- DATE (reserved word), 463, 465**
- DATEADD() function, 176–178, 474**
- DATEDIFF() function, 178–179, 474**
- @@DATEFIRST, 467**
- DATENAME() function, 180, 474**
- DATEPART() function, 180, 474**
- DATETIME\_PRECISION view, 511, 512, 514, 515**
- DAY() function, 181, 474**
- DAY (reserved word), 463, 465**
- DBCC (reserved word), 461**
- DB\_ID() function, 195, 476**
- DBMS. See Database Management System (DBMS)**
- DB\_NAME() function, 195, 476**
- @@DBTS, 467**
- DCL. See Data Control Language (DCL)**
- DDL. See Data Definition Language (DDL)**
- DEALLOCATE (reserved word), 461, 463**
- DEC (reserved word), 463, 465**
- DECIMAL (reserved word), 463, 465**
- DECLARE @ local\_variable command, 455–456**
- DECLARE (reserved word), 461, 463**
- default query view, 419**
- DEFAULT (reserved word), 461, 463**
- DEFAULT\_CHARACTER\_SET\_CATALOG view, 516**
- DEFAULT\_CHARACTER\_SET\_NAME view, 516**
- DEFAULT\_CHARACTER\_SET\_SCHEMA view, 516**
- defaults**
  - inserting, 259–261
  - naming, 324
  - overview, 329–330
- DEFERRABLE (reserved word), 464, 466**
- DEFERRED (reserved word), 464, 466**
- DEGREES() function, 193, 475**
- DELETE command, 269–270, 455**
- Delete procedure, 374**
- DELETE (reserved word), 461, 464**
- DELETE\_RULE view, 514**
- delivery time constraints, 400**
- DENSE\_RANK() function, 197–198, 477**
- DENY command, 123, 347**

## DENY (reserved word)

---

**DENY (reserved word)**, 461  
**DEPTH (reserved word)**, 466  
**DEREF (reserved word)**, 466  
**derived columns**, 135–137  
**derived table, creating**, 287  
**DESC (reserved word)**, 461, 464  
**DESCRIBE (reserved word)**, 464, 466  
**DESCRIPTOR (reserved word)**, 464, 466  
**design patterns**, 397  
**desktop database applications**, 400–401  
**DESTROY (reserved word)**, 466  
**DESTRUCTOR (reserved word)**, 466  
**deterministic functions**, 161–162  
**DETERMINISTIC (reserved word)**, 466  
**DIAGNOSTICS (reserved word)**, 464, 466  
**diagram pane in Query Designer tool**, 53  
**DICTIONARY (reserved word)**, 466  
**DIFFERENCE() function**, 304, 478  
**DISCONNECT (reserved word)**, 464, 466  
**DISK (reserved word)**, 461  
**DISTINCT (reserved word)**, 461, 464  
**distributed queries procedures**

- sp\_addlinkedserver, 489
- sp\_addlinkedsrvlogin, 489
- sp\_catalogs, 489
- sp\_column\_privileges\_ex, 489
- sp\_columns\_ex, 489
- sp\_droplinkedsrvlogin, 489
- sp\_foreignkeys, 489
- sp\_indexes, 490
- sp\_linkedservers, 490
- sp\_primarykeys, 490
- sp\_serveroption, 490
- sp\_table\_privileges\_ex, 490
- sp\_tables\_ex, 490

**DISTRIBUTED (reserved word)**, 461  
**Distributed Transaction Coordinator (DTC)**, 409  
**DML**. See **Data Manipulation Language (DML)**  
**docking window placement in SQL Server Management Studio**, 77–80  
**DOMAIN (reserved word)**, 464, 466  
**DOMAIN\_CATALOG view**, 509, 510, 511, 512, 514

**DOMAIN\_CONSTRAINTS views**

- CONSTRAINT\_CATALOG view, 512
- CONSTRAINT\_NAME view, 512
- CONSTRAINT\_SCHEMA view, 512
- DOMAIN\_CATALOG view, 512
- DOMAIN\_NAME view, 512
- DOMAIN\_SCHEMA view, 512
- INITIALLY\_DEFERRED view, 512
- IS\_DEFERRABLE view, 512

**DOMAIN\_DEFAULT view**, 512  
**DOMAIN\_NAME view**, 509, 511, 512, 514  
**DOMAINS views**

- CHARACTER\_MAXIMUM\_LENGTH view, 512
- CHARACTER\_OCTET\_LENGTH view, 512
- CHARACTER\_SET\_CATALOG view, 512
- CHARACTER\_SET\_NAME view, 512
- CHARACTER\_SET\_SCHEMA view, 512
- COLLATION\_CATALOG view, 512
- COLLATION\_NAME view, 512
- COLLATION\_SCHEMA view, 512
- DATA\_TYPE view, 512
- DATETIME\_PRECISION view, 512
- DOMAIN\_CATALOG view, 512
- DOMAIN\_DEFAULT view, 512
- DOMAIN\_NAME view, 512
- DOMAIN\_SCHEMA view, 512
- NUMERIC\_PRECISION view, 512
- NUMERIC\_PRECISION\_RADIX view, 512
- NUMERIC\_SCALE view, 512

**DOMAIN\_SCHEMA view**, 509, 511, 512, 514  
**double quotes**, 313  
**DOUBLE (reserved word)**, 461, 464  
**down arrow icon in SQL Server Management Studio**, 75  
**DROP (reserved word)**, 461, 464  
**DROP statement**, 117, 322  
**dropping database objects**, 322  
**DTC (Distributed Transaction Coordinator)**, 409  
**DTD\_IDENTIFIER view**, 516  
**DTS (Data Transformation Services)**, 256  
**DUMMY (reserved word)**, 461  
**DUMP (reserved word)**, 461  
**DYNAMIC (reserved word)**, 466

**E**

**EACH** (reserved word), 466  
**Edit menu in Query Analyzer**, 64  
**efficiency and primary keys**, 6  
**Ellison, Larry (Oracle)**, 19  
**ELSE** (reserved word), 461, 464  
**embedded SELECT statements**, 280–283  
**ENCRYPT()** function, 480  
**END** (reserved word), 461  
**END-EXEC** (reserved word), 464, 466  
**Enterprise Manager**  
   opening, 50  
   overview, 50–51  
   SQL expression, modifying, 58–62  
**Entity Relationship Diagram (ERD)**, 15  
**EQUALS** (reserved word), 466  
**ERRLVL** (reserved word), 461  
**@@ERROR**, 167–168, 204, 470  
**errors, handling and raising**, 374–378  
**ESCAPE** (reserved word), 461, 464  
**event-handling code**, 403–404  
**EVERY** (reserved word), 466  
**EXCEPT** command, 453  
**EXCEPT** (reserved word), 460, 464  
**EXCEPTION** (reserved word), 464, 466  
**exclusive locks**, 18  
**EXEC** (reserved word), 460, 462  
**EXECUTE AS** option, 344  
**EXECUTE** (reserved word), 460, 462  
**executing query with Query Designer tool**, 56  
**execution plan**, 43  
**EXISTES** (reserved word), 462  
**EXISTS()** function, 288–289  
**EXISTS** (reserved word), 460  
**EXIT** (reserved word), 460  
**EXP()** function, 193, 475  
**explicit transactions**, 254, 273–276  
**expression, joining on an**, 244–245  
**eXtensible Markup Language (XML)**  
   overview, 412  
   procedures, 508  
**EXTERNAL** (reserved word), 462, 466

**external systems and extended procedures**

xp\_cmdshell, 490  
xp\_enumgroups, 490  
xp\_findnextmsg, 490  
xp\_grantlogin, 490  
xp\_logevent, 490  
xp\_loginconfig, 490  
xp\_logininfo, 490  
xp\_msver, 490  
xp\_revokelogin, 490  
xp\_sprintf, 491  
xp\_sqlmaint, 491  
xp\_sscanf, 491

**EXTERNAL\_LANGUAGE** view, 516

**EXTERNAL\_NAME** view, 516

**EXTRACT** (reserved word), 462

**F**

**FALSE** (reserved word), 462, 466

**federated views**, 360–363

**federating data**, 249–250

**FETCH NEXT** statement, 297–298

**FETCH** (reserved word), 460, 462

**@@FETCH\_STATUS**, 176, 469

**fields, adding**, 430–431

**fifth normal form**, 33

**File menu in Query Analyzer**, 64

**FILE** (reserved word), 460

**file-based database products compared to relational database management system (RDBMS)**, 26

**FILE\_ID()** function, 195, 476

**FILE\_NAME()** function, 195, 476

**FILLFACTOR** (reserved word), 460

**filtering rows**, 137

**filtering techniques, extended**, 144–145

**filtering updates**, 264–266

**FIND** (reserved word), 464

**first normal form — 1NF**, 32

**FIRST** (reserved word), 462, 466

**FLOAT** (reserved word), 462, 466

**FLOOR()** function, 193, 475

- fn\_Get\_SQL() function, 480**
- fn\_HelpCollations() function, 480**
- fn\_helpcollations() function, 200**
- fn\_listextendedproperty() function, 195**
- fn\_ServerSharedDrives() function, 480**
- fn\_serversharedrives() function, 200**
- fn\_trace\_geteventinfo() function, 199, 478**
- fn\_trace\_getfilterinfo() function, 199, 478**
- fn\_trace\_getinfo() function, 199, 478**
- fn\_trace\_gettable() function, 199, 478**
- fn\_VirtualFileStats() function, 480**
- fn\_virtualfilestats() function, 201, 482**
- FOR clause, 454**
- FOR (reserved word), 460, 462**
- foreign key constraints**
  - naming, 324
  - overview, 334–336
- foreign keys, 30**
- FOREIGN (reserved word), 460, 462**
- format properties, setting, 431**
- FORMATMESSAGE() function, 201, 480**
- FORTRAN (reserved word), 462**
- FOUND (reserved word), 462, 464**
- fourth normal form, 32–33**
- FREE (reserved word), 464**
- free-form text indexing, 301**
- FREETEXT predicate, 318**
- FREETEXT (reserved word), 460**
- FREETEXTTABLE predicate, 319–320**
- FREETEXTTABLE (reserved word), 460**
- FROM clause, joining tables in, 234–235**
- FROM command, 451–452**
- FROM (reserved word), 460, 462**
- full joins, 241–243**
- full outer joins, 241–243**
- FULL (reserved word), 460, 462**
- full-text index queries**
  - AND operator, 319
  - catalogs, managing and populating, 304–313
  - CONTAINS predicate, 314–317
  - CONTAINSTABLE predicate, 317–318
  - double quotes, 313
  - FREETEXT predicate, 318
- FREETEXTTABLE predicate, 319–320**
  - logical operators, 319
  - Microsoft Search Service, 302–304
  - NEAR operator, 319
  - OR operator, 319
  - overview, 313
  - ranked results, 317
  - single quotes, 313
  - SQL Server 2000, 305–306
  - SQL Server 2005, 307–313
  - weighting values, 317
- full-text index/search procedures**
  - sp\_fulltext\_catalog, 491
  - sp\_fulltext\_column, 491
  - sp\_fulltext\_database, 491
  - sp\_fulltext\_service, 491
  - sp\_fulltext\_table, 491
  - sp\_help\_fulltext\_catalogs, 491
  - sp\_help\_fulltext\_catalogs\_cursor, 491
  - sp\_help\_fulltext\_columns, 491
  - sp\_help\_fulltext\_columns\_cursor, 491
  - sp\_help\_fulltext\_tables, 491
  - sp\_help\_fulltext\_tables\_cursor, 491
- full-text query expressions**
  - AND operator, 319
  - CONTAINS predicate, 314–317
  - CONTAINSTABLE predicate, 317–318
  - double quotes, 313
  - FREETEXT predicate, 318
  - FREETEXTTABLE predicate, 319–320
  - logical operators, 319
  - NEAR operator, 319
  - OR operator, 319
  - overview, 313
  - ranked results, 317
  - single quotes, 313
  - weighting values, 317
- FULLTEXTCATALOGPROPERTY() function, 195, 477**
- FULLTEXTSERVICEPROPERTY() function, 195, 477**
- FUNCTION (reserved word), 460**

**functions**

- aggregate functions, 164–167
- arguments, 161
- categories of, 160
- conversion functions, 169–181
- CONVERT () function, 161
- cursor functions and variables, 175–176
- date functions, 176–181
- deterministic functions, 161–162
- image/text functions, 192
- mathematical functions, 192–193
- metadata functions, 193–196
- naming, 324
- nested functions, 164
- nondeterministic functions, 161–162
- overview, 160–161
- queries, used in, 163
- ranking functions, 196–199
- security functions, 199–200
- SELECT statement used to assign variables, 163
- SET statement used to assign variables, 162–163
- string manipulation functions, 181–192
- system functions and variables, 200–204
- system statistical functions and variables, 204
- variables used with, 162–163

**future reserved words, 464–466**

**G**

- Gates, Bill (Microsoft), 23**
- GENERAL (reserved word), 464**
- GET (reserved word), 462, 464**
- GETANSINULL() function, 201, 480**
- GETDATE() function, 180–181, 474**
- GETUTCDATE() function, 180–181, 474**
- GLOBAL (reserved word), 462, 464**
- GO (reserved word), 462, 464**
- GOTO (reserved word), 460, 462**
- GRANT command, 123, 346–347**
- GRANT (reserved word), 460, 462**
- GRANTEE view, 510, 517**
- GRANTOR view, 510, 517**

- graphical query designer, 88–93**
- grid pane in Query Designer tool, 53**
- GROUP BY clause, 213–217, 221, 452–453**
- GROUP (reserved word), 460, 462**
- grouped aggregates**
  - COMPUTE BY clause, 228
  - COMPUTE clause, 225–228
  - GROUP BY clause, 213–217, 221
  - GROUPING() function, 223–225
  - HAVING clause, 218–219
  - subgrouping, 221
  - subtotals, 219–220
  - totals, 219–220
  - WITH CUBE operator, 222–223
  - WITH ROLLUP operator, 221–222
- grouping data, 213–228**
- GROUPING() function, 223–225, 472**
- GROUPING (reserved word), 464**
- groups, adding, 436–442**

**H**

- HAS\_DBACCESS() function, 199, 478**
- HAVING clause, 218–219, 286, 453**
- HAVING (reserved word), 460, 462**
- hierarchical database management systems (HDBMS), 4**
- HOLDLOCK (reserved word), 460**
- HOST (reserved word), 464**
- HOST\_ID() function, 201, 481**
- HOST\_NAME() function, 201, 481**
- HOURL (reserved word), 462, 464**
- Hungarian Notation naming standard, 112**

**I**

- IBM DB2, 19**
- IDENT\_CURRENT() function, 201, 481**
- IDENT\_INCR() function, 201, 481**
- identity, 37**
- @@IDENTITY, 204, 470**
- IDENTITY() function, 201, 481**
- identity keys, 30**

**IDENTITY property, 326–329**

**IDENTITY (reserved word), 460, 462**

**IDENTITYCOL (reserved word), 460**

**IDENTITY\_INSERT (reserved word), 460**

**IDENT\_SEED() function, 201, 481**

**@@IDLE, 204, 470**

**IF (reserved word), 460**

**IF statement, 380–382**

**IGNORE (reserved word), 464**

**image/text functions**

overview, 192

PATINDEX() function, 475

TEXTPTR() function, 475

TEXTVALID() function, 475

**IMMEDIATE (reserved word), 462, 465**

**implicit transactions, 254**

**Import and Export Data shortcut, 49**

**IN() function, 145–146, 287–288**

**IN (reserved word), 460, 462**

**INCLUDE (reserved word), 462**

**INDEX (reserved word), 460, 462**

**INDEX\_COL() function, 195, 477**

**indexed views, 339–340**

**INDEXKEY\_PROPERTY() function, 195, 477**

**INDEXPROPERTY() function, 195, 477**

**INDICATOR (reserved word), 462, 465**

**information schema views reference**

CHECK\_CONSTRAINTS view, 509

COLUMN\_DOMAIN\_USAGE view, 509–510

COLUMN\_PRIVILEGES view, 510

COLUMNS view, 510–511

CONSTRAINT\_COLUMN\_USAGE view, 511

CONSTRAINT\_TABLE\_USAGE view, 511

DOMAIN\_CONSTRAINTS view, 512

DOMAINS view, 512

KEY\_COLUMN\_USAGE view, 513

PARAMETERS view, 513

REFERENTIAL\_CONSTRAINTS view, 513–514

ROUTINE\_COLUMNS view, 514

ROUTINES view, 515–516

SCHEMATA view, 516

TABLE\_CONSTRAINTS view, 517

TABLE\_PRIVILEGES view, 517

TABLES view, 517

VIEW\_COLUMN\_USAGE view, 518

VIEWS view, 518

VIEW\_TABLE\_USAGE view, 518

**information transformed into data, 33–34**

**Informix, 19–20**

**infrastructure investment, 399**

**INITIALIZE (reserved word), 465**

**INITIALLY (reserved word), 463, 465**

**INITIALLY\_DEFERRED view, 512, 517**

**in-line comments, 113–114, 459**

**inline table-valued functions, 390–391**

**INNER JOIN statement, 235–236**

**inner join subqueries, 283–284**

**INNER (reserved word), 461, 463**

**INOUT (reserved word), 465**

**INPUT (reserved word), 463, 465**

**INSENSITIVE option, 297**

**INSENSITIVE (reserved word), 463**

**INSERT command, 257–258, 455**

**Insert procedure, 371–373**

**INSERT (reserved word), 461, 463**

**installation of Adventure Works Cycles  
database, 45**

**INT (reserved word), 463, 465**

**INTEGER (reserved word), 463, 465**

**intent locks, 18**

**INTERSECT command, 453**

**INTERSECT (reserved word), 461, 463**

**INTERVAL (reserved word), 463, 465**

**INTERVAL\_PRECISION view, 515**

**INTERVAL\_TYPE view, 515**

**INTO (reserved word), 461, 463**

**@@IO\_BUSY, 204, 470**

**IS (reserved word), 461, 463**

**ISDATE() function, 201, 481**

**IS\_DEFERRABLE view, 512, 517**

**IS\_DETERMINISTIC view, 516**

**IS\_GRANTABLE view, 510, 517**

**IS\_IMPLICITLY\_INVOCABLE view, 516**

**IS\_MEMBER() function, 200, 478**

**ISNULL() function, 201, 481**

**IS\_NULLABLE view, 510, 514**

**IS\_NULL\_CALL** view, 516  
**ISNUMERIC()** function, 201, 481  
**ISOLATION** (reserved word), 463, 465  
**IS\_SRVROLEMEMBER()** function, 200, 478  
**IS\_UPDATABLE** view, 518  
**IS\_USER\_DEFINED\_CAST** view, 516  
**ITERATE** (reserved word), 465

## J

**JET Database Engine**, 20  
**JOIN** (reserved word), 461, 463  
 join table, 39  
 joins  
   cross joins, 243  
   expression, joining on an, 244–245  
   FROM clause, joining tables in, 234–235  
   full joins, 241–243  
   full outer joins, 241–243  
   INNER JOIN statement, 235–236  
   legacy outer joins, 239  
   multicolumn joins, 239–240  
   multi-table joins, 245–246  
   non-equijoins, 240–241  
   outer joins, 236–239  
   overview, 233  
   predicate, filtering records in join clause using, 243–244  
   SELECT statement and, 244–245  
   special-purpose join operations, 241–246  
   WHERE clause, joining tables in, 233–234

## K

**KEY** (reserved word), 461, 463  
**KEY\_COLUMN\_USAGE** views  
   **CONSTRAINT\_CATALOG** view, 513  
   **CONSTRAINT\_NAME** view, 513  
   **CONSTRAINT\_SCHEMA** view, 513  
   **ORDINAL\_POSITION** view, 513  
   **TABLE\_CATALOG** view, 513  
   **TABLE\_NAME** view, 513  
   **TABLE\_SCHEMA** view, 513  
**KILL** (reserved word), 461

## L

**@@LANGID**, 467  
**@@LANGUAGE**, 468  
**LANGUAGE** (reserved word), 463, 465  
**LARGE** (reserved word), 465  
**LAST** (reserved word), 463, 465  
**LAST ALTERED** view, 516  
**LATERAL** (reserved word), 465  
 layers, queries having, 108  
**LEADING** (reserved word), 463, 465  
**LEFT()** function, 185–186, 478  
**LEFT** (reserved word), 461, 463  
 legacy outer joins, 239  
**LEN()** function, 185, 478  
**LESS** (reserved word), 465  
**LEVEL** (reserved word), 463, 465  
**LIKE** operator, 456–457  
**LIKE** (reserved word), 461, 463  
**LIMIT** (reserved word), 465  
**LINENO** (reserved word), 461  
 linked server connection, 362  
 linking tables, 15  
**LOAD** (reserved word), 461  
**LOCAL** (reserved word), 463, 465  
**LOCALTIME** (reserved word), 465  
**LOCALTIMESTAMP** (reserved word), 465  
 location of users, 399  
**LOCATOR** (reserved word), 465  
 locking function, 456  
 locking options, 393–395  
 locks  
   described, 16  
   exclusive locks, 18  
   intent locks, 18  
   shared locks, 17  
   update locks, 17  
**@@LOCK\_TIMEOUT**, 468  
**LOG()** function, 193, 475  
**LOG10()** function, 475  
 log shipping procedures  
   sp\_add\_log\_shipping\_alert\_job, 492  
   sp\_add\_log\_shipping\_primary\_database, 492  
   sp\_add\_log\_shipping\_primary\_secondary, 492

### **log shipping procedures (continued)**

sp\_add\_log\_shipping\_secondary\_primary, 492  
sp\_change\_log\_shipping\_primary\_database, 492  
sp\_change\_log\_shipping\_secondary\_database, 492  
sp\_change\_log\_shipping\_secondary\_primary, 492  
sp\_cleanup\_log\_shipping\_history, 492  
sp\_delete\_log\_shipping\_alert\_job, 492  
sp\_delete\_log\_shipping\_primary\_database, 492  
sp\_delete\_log\_shipping\_primary\_secondary, 492  
sp\_delete\_log\_shipping\_secondary\_database, 492  
sp\_delete\_log\_shipping\_secondary\_primary, 492  
sp\_help\_log\_shipping\_alert\_job, 492  
sp\_help\_log\_shipping\_monitor\_primary, 492  
sp\_help\_log\_shipping\_monitor\_secondary, 492  
sp\_help\_log\_shipping\_primary\_database, 492  
sp\_help\_log\_shipping\_primary\_secondary, 492  
sp\_help\_log\_shipping\_secondary\_database, 492  
sp\_help\_log\_shipping\_secondary\_primary, 492  
sp\_refresh\_log\_shipping\_monitor, 492  
sp\_resolve\_logins, 492

**logged operations, 256**

**logical comparisons, 141**

**logical design, 28**

**logical gates, 141**

**logical operators, 319**

**long names naming standard, 112**

**looping, 384–386**

**LOWER() function, 188–190, 478**

**LOWER (reserved word), 463**

**lower-case, delimited naming standard, 112**

**LTRIM() function, 190–191, 479**

## **M**

### **mail procedures**

sp\_send\_dbmail, 493  
sysmail\_add\_account\_sp, 493  
sysmail\_add\_principalprofile\_sp, 493  
sysmail\_add\_profileaccount\_sp, 493  
sysmail\_add\_profile\_sp, 493

sysmail\_configure\_sp, 493  
sysmail\_delete\_account\_sp, 493  
sysmail\_delete\_principalprofile\_sp, 493  
sysmail\_delete\_profileaccount\_sp, 493  
sysmail\_delete\_profile\_sp, 493  
sysmail\_help\_account\_sp, 493  
sysmail\_help\_configure\_sp, 493  
sysmail\_help\_principalprofile\_sp, 493  
sysmail\_help\_profileaccount\_sp, 493  
sysmail\_help\_profile\_sp, 493  
sysmail\_start\_sp, 493  
sysmail\_stop\_sp, 493  
sysmail\_update\_account\_sp, 493  
sysmail\_update\_principalprofile\_sp, 493  
sysmail\_update\_profileaccount\_sp, 493  
sysmail\_update\_profile\_sp, 493  
xp\_readmail, 493  
xp\_sendmail, 493

**many-to-many relationships, 15**

**MAP (reserved word), 465**

**MATCH (reserved word), 463, 465**

**MATCH\_OPTION view, 514**

### **mathematical functions**

ABS() function, 193, 475  
ACOS() function, 193, 475  
ASIN() function, 193, 475  
ATAN() function, 193, 475  
ATN2() function, 193, 475  
CEILING() function, 193, 475  
COS() function, 193, 475  
COT() function, 193, 475  
DEGREES() function, 193, 475  
EXP() function, 193, 475  
FLOOR() function, 193, 475  
LOG() function, 193, 475  
LOG10() function, 475  
overview, 192  
PI() function, 193, 476  
POWER() function, 193, 476  
RADIANS() function, 193, 476  
RAND() function, 193, 476  
ROUND() function, 193, 476  
SIGN() function, 193, 476

- SIN() function, 193, 476
- SQRT() function, 193, 476
- SQUARE() function, 193, 476
- TAN() function, 193, 476
- MAX() function, 166–167, 208, 472**
- (max) option, 12–13**
- MAX (reserved word), 463**
- @@MAX\_CONNECTIONS, 468**
- MAX\_DYNAMIC\_RESULT\_SETS view, 516**
- MAXIMUM\_CARDINALITY view, 516**
- @@MAX\_PRECISION, 468**
- metadata functions**
  - COL\_LENGTH() function, 194, 476
  - COL\_NAME() function, 194, 476
  - COLUMNPROPERTY() function, 194, 476
  - DATABASEPROPERTY() function, 194, 476
  - DATABASEPROPERTYEX() function, 194, 476
  - DB\_ID() function, 195, 476
  - DB\_NAME() function, 195, 476
  - FILE\_ID() function, 195, 476
  - FILE\_NAME() function, 195, 476
  - fn\_listextendedproperty() function, 195
  - FULLTEXTCATALOGPROPERTY() function, 195, 477
  - FULLTEXTSERVICEPROPERTY() function, 195, 477
  - INDEX\_COL() function, 195, 477
  - INDEXKEY\_PROPERTY() function, 195, 477
  - INDEXPROPERTY() function, 195, 477
  - OBJECT\_ID() function, 195, 477
  - OBJECT\_NAME() function, 195, 477
  - OBJECTPROPERTY() function, 196, 477
  - overview, 193–194
- method call, 409**
- Microsoft, 23, 24**
- Microsoft Access**
  - overview, 20, 401
  - views created in, 351–353
- Microsoft Search Service**
  - DIFFERENCE() function, 304
  - overview, 302
  - SOUNDEX() function, 302–304
- Microsoft Solutions Framework (MSF), 412**
- Microsoft SQL Server, 18–19**
- Microsoft Visual SourceSafe, 116**
- @@MICROSOFTVERSION, 468**
- middle-tier component, 406–408**
- MIN() function, 166–167, 208, 472**
- MIN (reserved word), 463**
- MINUTE (reserved word), 463, 465**
- miscellaneous functions, 456**
- MODIFIES (reserved word), 465**
- MODIFY (reserved word), 465**
- modifying data through views, 365–366**
- modifying records**
  - filtering updates, 264–266
  - multiple tables, updating rows based on, 266–267
  - overview, 263
  - stored procedures, updating records using, 267–269
  - UPDATE statement, 264–269
  - views, updating using, 267
- MODULE (reserved word), 463, 465**
- MODULE\_CATALOG view, 515**
- MODULE\_NAME view, 515**
- MODULE\_SCHEMA view, 515**
- MONTH() function, 181, 474**
- MONTH (reserved word), 463, 465**
- MSF (Microsoft Solutions Framework), 398**
- multicolumn joins, 239–240**
- multiple associations, 39–40**
- multiple tables**
  - deleting records based on, 271–272
  - joins, 245–246
  - queries, 233–251
  - updating rows based on, 266–267
- multi-statement table-valued functions, 391–393**
- multi-system integrated solutions**
  - application programming models, 411–412
  - overview, 411
  - project management challenges, 412
  - system integration and data exchange, 412
- multi-table queries**
  - joins, 233–246
  - subqueries, 233
  - union queries, 247–251

**multi-valued columns, 40–41**

**MySQL, 20**

## N

**NAMES (reserved word), 463, 465**

### **naming**

check constraints, 324

clustered index, 324

conventions, 110–112

default, 324

foreign key constraints, 324

functions, 324

non-clustered index, 324

overview, 322–325

primary key constraints, 324

stored procedures, 324

tables, 324

triggers, 324

views, 324

**NATIONAL (reserved word), 461, 463**

**NATURAL (reserved word), 463, 465**

**navigating cursors, 297–298**

**NCHAR() function, 184, 479**

**NCHAR (reserved word), 463, 466**

**NCLOB (reserved word), 466**

**NEAR operator, 319**

**nested functions, 164**

**nested loop inner join, 291**

**@@NESTLEVEL, 468**

**.NET Framework incorporated with SQL Server 2005, 3–4**

**network database management systems (NDBMS), 4**

**new features of SQL Server 2005, 3**

**NEW (reserved word), 466**

**NEWID() function, 201, 481**

**NEXT (reserved word), 464, 466**

**NO (reserved word), 464, 466**

**NOCHECK (reserved word), 461**

**non-clustered index, 324**

**NONCLUSTERED (reserved word), 461**

**nondeterministic functions, 161–162**

**NONE (reserved word), 464, 466**

**non-equi joins, 240–241**

### **normalization rules**

application of, 34–42

Boyce-Codd Normal Form (BCNF), 32–33

breaking, 42

bridge table, 39

described, 30–32

fifth normal form, 33

first normal form — 1NF, 32

fourth normal form, 32–33

identity, 37

join table, 39

multiple associations, 39–40

multi-valued columns, 40–41

Object Role Modeling (ORM), 33

planning ahead, 37–38

questioning, 42

second normal form — 2NF, 32

surrogate key, 37

third normal form — 3NF, 32

Unified Modeling Language (UML), 33

**NOT EXISTS() function, 289**

**NOT operator, 142–143**

**NOT (reserved word), 461, 464**

**notification services procedures, 493–494**

**NSAdministrationHistory, 493**

**NSDiagnosticDeliveryChannel, 493**

**NSDiagnosticEventClass, 493**

**NSDiagnosticEventProvider, 494**

**NSDiagnosticFailedNotifications, 494**

**NSDiagnosticNotificationClass, 494**

**NSDiagnosticSubscriptionClass, 494**

**NSEventBatchDetails, 494**

**NSEventBeginBatch, 494**

**NSEventFlushBatch, 494**

**NSEventSubmitBatch, 494**

**NSEventWrite, 494**

**NSExecuteRuleFiring, 494**

**NSNotificationBatchDetails, 494**

**NSPrepareRuleFiring, 494**

**NSQuantumDetails, 494**

**NSQuantumExecutionTime, 494**

**NSQuantumFailures**, 494  
**NSQuantumList**, 494  
**NSQuantumPerformance**, 494  
**NSQuantumsSkipped**, 494  
**NSScheduledSubscriptionDetails**, 494  
**NSScheduledSubscriptionList**, 494  
**NSSetQuantumClock**, 494  
**NSSetQuantumClockDate**, 494  
**NSSnapshotApplications**, 494  
**NSSnapshotDeliveryChannels**, 494  
**NSSnapshotEvents**, 494  
**NSSnapshotProviders**, 494  
**NSSnapshotSubscriptions**, 494  
**NSVacuum**, 494  
**n-tier component solutions**  
   component transaction management, 409  
   middle-tier component, 406–408  
   overview, 405  
   presentation layer, 408  
   resource pooling, 408–409  
   server-side SQL objects, 405–406  
**NTILE(n) function**, 199, 477  
**NULL (reserved word)**, 461, 464  
**null value**  
   inserting, 259–261  
   overview, 143–144  
**nullability**, 13, 325–326  
**NULLIF() function**, 201, 481  
**NULLIF (reserved word)**, 461, 464  
**NUMERIC (reserved word)**, 464, 466  
**NUMERIC\_PRECISION view**, 510, 512, 514, 515  
**NUMERIC\_PRECISION\_RADIX view**, 510, 512, 514, 515  
**NUMERIC\_SCALE view**, 511, 512, 514, 515

## O

**object browser in Query Analyzer**, 69  
**Object Explorer in SQL Server Management Studio**, 83–84  
**object reference**, 449  
**OBJECT (reserved word)**, 466  
**Object Role Modeling (ORM)**, 33, 398

**OBJECT\_ID() function**, 195, 477  
**OBJECT\_NAME() function**, 195, 477  
**object-oriented database management systems (ODBMS)**, 4  
**object-pooling**, 409  
**OBJECTPROPERTY() function**, 196, 477  
**object-relational database management systems (ORDBMS)**, 4–5  
**objects**  
   altering, 121–122  
   dropping, 122–123  
   overview, 449  
**OCTET-LENGTH (reserved word)**, 464  
**ODBC (Open Database Connectivity)**  
   overview, 402  
   reserved words in, 462–464  
**Odell, Margaret (Consensus Soundex)**, 302  
**OF (reserved word)**, 461, 464  
**OFF (reserved word)**, 461  
**OFFSETS (reserved word)**, 461  
**OLD (reserved word)**, 466  
**OLE automation procedures**  
   sp\_OACreate, 495  
   sp\_OADestroy, 495  
   sp\_OAGetErrorInfo, 495  
   sp\_OAGetProperty, 495  
   sp\_OAMethod, 495  
   sp\_OASetProperty, 495  
   sp\_OAStop, 495  
**ON (reserved word)**, 461, 464  
**one-to-exactly-one relationships**, 15  
**one-to-many relationships**, 14  
**one-to-one relationships**, 14  
**one-to-zero relationships**, 14  
**ONLY (reserved word)**, 464, 466  
**Open Database Connectivity (ODBC)**  
   overview, 402  
   reserved words in, 462–464  
**OPEN (reserved word)**, 461, 464  
**OPENDATASOURCE (reserved word)**, 461  
**opening**  
   Enterprise Manager, 50  
   SQL Server Management Studio, 74

**OPENQUERY (reserved word)**, 461  
**OPENROWSET (reserved word)**, 461  
**OPENXML (reserved word)**, 461  
**OPERATION (reserved word)**, 466  
operator precedence, 146–148  
**OPTION clause**, 454  
**OPTION (reserved word)**, 461, 464  
**@@OPTIONS**, 468–469  
**OR operator**, 142, 319  
**OR (reserved word)**, 461, 464  
Oracle, 19  
**ORDBMS (object relational database management systems)**, 4–5  
**ORDER BY clause**, 150–152, 454  
**ORDER (reserved word)**, 461, 464  
**ORDINALITY (reserved word)**, 466  
**ORDINAL\_POSITION view**, 510, 513, 514  
**ORM (Object Role Modeling)**, 33, 398  
**OSQL utility**, 70–72  
**OUT (reserved word)**, 466  
outer join subqueries, 284–285  
outer joins, 236–239  
**OUTER (reserved word)**, 461, 464  
**OUTPUT (reserved word)**, 464, 466  
**OVER (reserved word)**, 461  
**OVERLAPS (reserved word)**, 462  
overriding constraints, 336  
**OWNER option**, 344

## P

**@@PACKET\_ERRORS**, 204, 471  
**@@PACK\_RECEIVED**, 204, 470  
**@@PACK\_SENT**, 204, 470  
**PAD (reserved word)**, 462, 466  
parameter lookup list, 424–425  
**PARAMETER (reserved word)**, 466  
parameterized queries, 420–423  
parameterized views, stored procedures as, 366–367  
parameters, 341–342  
**PARAMETERS (reserved word)**, 466  
**PARAMETERS view**, 513

**PARAMETER\_STYLE view**, 516  
parent-child relationships, 14  
parentheses, use of, 148–150  
**PARSENAME() function**, 201, 481  
**PARTIAL (reserved word)**, 462, 466  
partitioned views, 357–360  
partitioning data, 249–251  
Pascal Case naming standard, 112  
**PASCAL (reserved word)**, 462  
**PATH (reserved word)**, 466  
**PATINDEX() function**, 185, 475, 479  
**PERCENT (reserved word)**, 460  
**PERCENT statement**, 155–156  
performance and, 250–251  
**PERMISSIONS() function**, 201, 481  
physical design, 28–29  
**PI() function**, 193, 476  
**PLAN (reserved word)**, 460  
**PL/SQL (Procedure Language/Structured Query Language)**, 19  
**POSITION (reserved word)**, 462  
**POSTFIX (reserved word)**, 466  
**POWER() function**, 193, 476  
**PRECISION (reserved word)**, 460, 462  
predicate, filtering records in join clause using, 243–244  
**PREFIX (reserved word)**, 466  
**PREORDER (reserved word)**, 466  
**PREPARE (reserved word)**, 462, 466  
presentation layer and n-tier component solutions, 408  
**PRESERVE (reserved word)**, 462, 464  
primary key constraints  
  naming, 324  
  overview, 331–333  
primary keys, 6–7, 29–30  
**PRIMARY (reserved word)**, 460, 462  
**PRINT (reserved word)**, 460  
**PRIOR (reserved word)**, 462, 464  
**PRIVILEGES (reserved word)**, 462, 464  
**PRIVILEGE\_TYPE view**, 510, 517  
**PROC (reserved word)**, 460  
procedure cache, 44

**Procedure Language/Structured Query Language (PL/SQL), 19****PROCEDURE (reserved word), 460, 462****Professional SQL Server 2000 Programming (Viera), 1, 304****Professional SQL Server 2005 Programming (Viera), 1, 304****Professional SQL Server Reporting Services (Wiley Publishing, Inc.), 414****Profiler, 49, 73****profiler procedures**

- sp\_trace\_create, 495
- sp\_trace\_generateevent, 495
- sp\_trace\_setevent, 495
- sp\_trace\_setfilter, 495
- sp\_trace\_setstatus, 495

**programming objects**

- business logic, processing, 378–380
- CASE statement, 382–384
- complexity, hiding, 364–365
- conditional logic, 380–382
- creating views, 351–356
- Delete procedure, 374
- errors, handling and raising, 374–378
- federated views, 360–363
- IF statement, 380–382
- inline table-valued functions, 390–391
- Insert procedure, 371–373
- locking options, 393–395
- looping, 384–386
- Microsoft Access, creating views in, 351–353
- modifying data through views, 365–366
- multi-statement table-valued functions, 391–393
- parameterized views, stored procedures as, 366–367
- partitioned views, 357–360
- record maintenance, 370
- rows, ordering, 356–357
- scalar functions, 386–390
- securing data, 363
- SQL Script, creating views in, 354–356
- SQL Server 2000, error handling in, 374–377
- SQL Server 2005, error handling in, 378

- stored procedures, 366–386
- transaction management, 393–395
- Update procedure, 373–374
- using parameters, 367–369
- values, returning, 369–370
- views, 350–366
- virtual tables, 350
- Visual Studio, creating views in, 353–354
- project management challenges, 412**
- projects in SQL Server Management Studio, 82–83**
- PUBLIC (reserved word), 460, 462**
- PWDCONPARE() function, 481**
- PWDENCRYPT() function, 482**

**Q****queries**

- functions, used in, 163
- SQL Server Reporting Services, 418–423
- writing, 84–86

**Query Analyzer**

- Edit menu, 64
- File menu, 64
- launching, 64
- object browser, 69
- overview, 49, 63–64
- Query menu, 64
- SQL script and batch conventions, 65–68
- Tools menu, 64

**Query Designer tool**

- diagram pane, 53
- executing query, 56
- grid pane, 53
- how it works, 52–63
- overview, 52–53
- SQL expression, modifying, 58–62
- SQL pane, 53
- table, viewing, 58
- toolbar options, 57–58

**query execution, 456****query processing**

- buffer cache, 44
- described, 42–44

## query processing (continued)

- execution plan, 43
- procedure cache, 44

## query syntax, 109–110

## questioning normalization rules, 42

## QUOTENAME() function, 192, 479

# R

## RADIANS() function, 193, 476

## RAID device, 357–359

## RAISERROR (reserved word), 460

## RAND() function, 193, 476

## RANK() function, 197–198, 477

## ranked results, 317

## ranking functions

- DENSE\_RANK() function, 197–198, 477
- NTILE(n) function, 199, 477
- overview, 196
- RANK() function, 197–198, 477
- ROW\_NUMBER() function, 197, 477

## RDBMS. See relational database management systems (RDBMS)

## RDO (Remote Data Objects), 402

## READ COMMITTED locking option, 394

## READ (reserved word), 460, 462

## READ UNCOMMITTED locking option, 394

## READS (reserved word), 464

## READTEXT (reserved word), 460

## REAL (reserved word), 462, 464

## RECONFIGURE (reserved word), 460

## record maintenance, 370

## RECURSIVE (reserved word), 464

## REF (reserved word), 464

## REFERENCES (reserved word), 460, 462

## REFERENCING (reserved word), 464

## REFERENTIAL\_CONSTRAINTS views

- CONSTRAINT\_CATALOG view, 513
- CONSTRAINT\_NAME view, 513
- CONSTRAINT\_SCHEMA view, 513
- DELETE\_RULE view, 514
- MATCH\_OPTION view, 514
- UNIQUE\_CONSTRAINT\_CATALOG view, 513
- UNIQUE\_CONSTRAINT\_NAME view, 513

UNIQUE\_CONSTRAINT\_SCHEMA view, 513

UPDATE\_RULE view, 514

## relational database management systems (RDBMS)

data integrity, 16–18

described, 5

file-based database products compared, 26

overview, 4

relationships, 13–15

tables, 5–13

## relationships

Entity Relationship Diagram (ERD), 15

linking tables, 15

many-to-many relationships, 15

one-to-exactly-one relationships, 15

one-to-many relationships, 14

one-to-one relationships, 14

one-to-zero relationships, 14

overview, 13–14, 29

parent-child relationships, 14

## RELATIVE (reserved word), 462, 464

## Remote Data Objects (RDO), 402

## removing records

DELETE statement, 269–270

multiple tables, deleting records based on, 271–272

overview, 269

TRUNCATE TABLE statement, 272–273

WHERE clause, 270

## @@REMSERVER, 469

## REPEATABLE READ locking option, 394

## REPLACE() function, 191, 479

## REPLICATE() function, 191, 479

## replication procedures, 496–501

## REPLICATION (reserved word), 460

## Report Designer in SQL Server Reporting Services, 414–418

## report layout, 426–430

## Report Manager, 73, 443

## report parameters, configuring, 431–436

## Reporting Services, 49

## reserved words

future reserved words, 464–466

list of, 460–461

- ODBC reserved words, 462–464
- overview, 459–460
- resource pooling, 408–409**
- RESTORE (reserved word), 460**
- RESTRICT (reserved word), 460, 462**
- RESULT (reserved word), 464**
- results, sorting, 150–152**
- RETURN (reserved word), 460**
- RETURNS (reserved word), 465**
- REVERSE() function, 191–192, 479**
- REVOKE command, 123, 347**
- REVOKE (reserved word), 460, 462**
- RIGHT() function, 185–186**
- RIGHT (reserved word), 460, 462**
- ROLE (reserved word), 465**
- ROLLBACK (reserved word), 460, 462**
- ROLLUP (reserved word), 465**
- ROUND() function, 193, 476**
- ROUTINE (reserved word), 465**
- ROUTINE\_BODY view, 516**
- ROUTINE\_CATALOG view, 515**
- ROUTINE\_COLUMNS views**
  - CHARACTER\_MAXIMUM\_LENGTH view, 514
  - CHARACTER\_OCTET\_LENGTH view, 514
  - CHARACTER\_SET\_CATALOG view, 514
  - CHARACTER\_SET\_NAME view, 514
  - CHARACTER\_SET\_SCHEMA view, 514
  - COLLATION\_CATALOG view, 514
  - COLLATION\_NAME view, 514
  - COLLATION\_SCHEMA view, 514
  - COLUMN\_DEFAULT view, 514
  - COLUMN\_NAME view, 514
  - DATA\_TYPE view, 514
  - DATETIME\_PRECISION view, 514
  - DOMAIN\_CATALOG view, 514
  - DOMAIN\_NAME view, 514
  - DOMAIN\_SCHEMA view, 514
  - IS\_NULLABLE view, 514
  - NUMERIC\_PRECISION view, 514
  - NUMERIC\_PRECISION\_RADIX view, 514
  - NUMERIC\_SCALE view, 514
  - ORDINAL\_POSITION view, 514
  - TABLE\_CATALOG view, 514
  - TABLE\_NAME view, 514
  - TABLE\_SCHEMA view, 514
- ROUTINE\_DEFINITION view, 516**
- ROUTINE\_NAME view, 515**
- ROUTINES views**
  - CHARACTER\_MAXIMUM\_LENGTH view, 515
  - CHARACTER\_OCTET\_LENGTH view, 515
  - CHARACTER\_SET\_CATALOG view, 515
  - CHARACTER\_SET\_NAME view, 515
  - CHARACTER\_SET\_SCHEMA view, 515
  - COLLATION\_CATALOG view, 515
  - COLLATION\_NAME view, 515
  - COLLATION\_SCHEMA view, 515
  - CREATED view, 516
  - DATA\_TYPE view, 515
  - DATETIME\_PRECISION view, 515
  - DTD\_IDENTIFIER view, 516
  - EXTERNAL\_LANGUAGE view, 516
  - EXTERNAL\_NAME view, 516
  - INTERVAL\_PRECISION view, 515
  - INTERVAL\_TYPE view, 515
  - IS\_DETERMINISTIC view, 516
  - IS\_IMPLICITLY\_INVOCABLE view, 516
  - IS\_NULL\_CALL view, 516
  - IS\_USER\_DEFINED\_CAST view, 516
  - LAST\_ALTERED view, 516
  - MAX\_DYNAMIC\_RESULT\_SETS view, 516
  - MAXIMUM\_CARDINALITY view, 516
  - MODULE\_CATALOG view, 515
  - MODULE\_NAME view, 515
  - MODULE\_SCHEMA view, 515
  - NUMERIC\_PRECISION view, 515
  - NUMERIC\_PRECISION\_RADIX view, 515
  - NUMERIC\_SCALE view, 515
  - PARAMETER\_STYLE view, 516
  - ROUTINE\_BODY view, 516
  - ROUTINE\_CATALOG view, 515
  - ROUTINE\_DEFINITION view, 516
  - ROUTINE\_NAME view, 515
  - ROUTINE\_SCHEMA view, 515
  - ROUTINE\_TYPE view, 515
  - SCHEMA\_LEVEL\_ROUTINE view, 516
  - SCOPE\_CATALOG view, 516

### **ROUTINES views (continued)**

- SCOPE\_NAME view, 516
- SCOPE\_SCHEMA view, 516
- SPECIFIC\_CATALOG view, 515
- SPECIFIC\_NAME view, 515
- SPECIFIC\_SCHEMA view, 515
- SQL\_DATA\_ACCESS view, 516
- SQL\_PATH view, 516
- TYPE\_UDT\_CATALOG view, 515
- TYPE\_UDT\_NAME view, 516
- TYPE\_UDT\_SCHEMA view, 516
- UDT\_CATALOG view, 515
- UDT\_NAME view, 515
- UDT\_SCHEMA view, 515

**ROUTINE\_SCHEMA view, 515**

**ROUTINE\_TYPE view, 515**

**ROW (reserved word), 465**

**row-based operations, 109**

**@@ROWCOUNT, 204, 470**

**ROWCOUNT (reserved word), 460**

**ROWCOUNT\_BIG() function, 201, 482**

**ROWGUIDCOL (reserved word), 460**

**ROW\_NUMBER() function, 197, 477**

### **rows**

- from another table, inserting, 261–262
- ordering, 356–357
- overview, 5

**ROWS (reserved word), 462, 465**

**rowset versus cursor operations, 296–297**

**RTRIM() function, 190–191, 479**

**RULE (reserved word), 460**

**Russell, Robert (*Consensus Soundex*), 302**

## **S**

**SAVE (reserved word), 460**

**SAVEPOINT (reserved word), 465**

**scalar expressions, 280–283**

**scalar functions, 386–390**

**scaling considerations, 250–251**

**SCHEMA (reserved word), 460, 462**

**SCHEMABINDING option, 338**

**SCHEMA\_LEVEL\_ROUTINE view, 516**

**SCHEMA\_NAME view, 516**

**schema.object, 449**

**SCHEMA\_OWNER view, 516**

### **SCHEMATA views**

- CATALOG\_NAME view, 516

- DEFAULT\_CHARACTER\_SET\_CATALOG view, 516

- DEFAULT\_CHARACTER\_SET\_NAME view, 516

- DEFAULT\_CHARACTER\_SET\_SCHEMA view, 516

- SCHEMA\_NAME view, 516

- SCHEMA\_OWNER view, 516

**SCOPE (reserved word), 465**

**SCOPE\_CATALOG view, 516**

**SCOPE\_IDENTITY() function, 202, 482**

**SCOPE\_NAME view, 516**

**SCOPE\_SCHEMA view, 516**

### **script comment conventions**

- comment block, 459

- in-line comment, 459

- single-line comment, 459

**script generation, 115**

**script management, 116**

**scripting options in SQL Server Management Studio, 87–88**

**scripting practices, 121–123**

**SCROLL (reserved word), 462, 465**

**SEARCH (reserved word), 465**

**second normal form — 2NF, 32**

**SECOND (reserved word), 462, 465**

**SECTION (reserved word), 462, 465**

**securing data, 363**

### **security functions**

- fn\_trace\_geteventinfo() function, 199, 478

- fn\_trace\_getfilterinfo() function, 199, 478

- fn\_trace\_getinfo() function, 199, 478

- fn\_trace\_gettable() function, 199, 478

- HAS\_DBACCESS() function, 199, 478

- IS\_MEMBER() function, 200, 478

- IS\_SRVROLEMEMBER() function, 200, 478

- overview, 199, 478

- SUSER\_SID() function, 200, 478

- SUSER\_SNAME() function, 200, 478

- USER\_ID() function, 478

- USER\_id() function, 200

- USER\_NAME() function, 200

**security procedures, 501–504**

- security requirements in application programming models, 400**
- SELECT command**
  - AND operator, 141
  - BETWEEN operator, 145
  - calculated columns, 135–137
  - clauses in, 128
  - column aliasing, 134–135
  - columns, choosing, 128–132
  - comparison operators, 140–148
  - derived columns, 135–137
  - filtering rows, 137
  - filtering techniques, extended, 144–145
  - IN () function, 145–146
  - joins and, 244–245
  - logical comparisons, 141
  - NOT operator, 142–143
  - Null value, 143–144
  - operator precedence, 146–148
  - OR operator, 142
  - ORDER BY clause, 150–152
  - overview, 108, 128, 258, 450–451
  - parentheses, use of, 148–150
  - PERCENT statement, 155–156
  - results, sorting, 150–152
  - SQL Server 2005 schemas, 133–134
  - top values, 152–154
  - variables, used to assign, 163
  - WITH TIES statement, 154–155
  - WHERE clause, 138–148
- SELECT INTO command, 262, 451**
- SELECT (reserved word), 460, 462**
- SELF option, 344**
- semantics, 26–27**
- SEQUEL (Structured English Query Language), 105**
- SEQUENCE (reserved word), 465**
- SERIALIZABLE locking option, 394**
- Server Network Utility, 49**
- server.database..object, 449**
- server.database.schema.object, 449**
- @@SERVICENAME, 469**
- server...object, 449**
- SERVERPROPERTY() function, 202, 482**
- server..schema.object, 449**
- server-side SQL objects, 405–406**
- Service Manager, 49**
- @@SERVICENAME, 168, 469**
- SESSION (reserved word), 462, 465**
- SESSIONPROPERTY() function, 202, 482**
- SESSION\_USER() function, 202, 482**
- SESSION\_USER (reserved word), 461, 462**
- SET operator**
  - overview, 456
  - variables, used to assign, 162–163
- SET (reserved word), 461, 463**
- set-based operations, 109**
- SETS (reserved word), 465**
- SETUSER (reserved word), 461**
- shared locks, 17**
- SHUTDOWN (reserved word), 461**
- SIGN() function, 193, 476**
- SIN() function, 193, 476**
- single quotes, 313**
- single-line comment, 459**
- SIZE (reserved word), 463, 465**
- SMALLINT (reserved word), 463, 465**
- SOAP (Simple Object Application Protocol), 411**
- SOME (reserved word), 461, 463**
- SOUNDEX() function, 302–304, 479**
- SPACE() function, 191, 479**
- SPACE (reserved word), 463, 465**
- sp\_ActiveDirectory\_Obj, 483**
- sp\_ActiveDirectory\_SCP, 483**
- sp\_add\_data\_file\_recover\_suspect\_db, 485**
- sp\_addextendedproc, 485**
- sp\_addextendedproperty, 485**
- sp\_addlinkedserver, 489**
- sp\_addlinkedsrvlogin, 489**
- sp\_add\_log\_file\_recover\_suspect\_db, 485**
- sp\_add\_log\_shipping\_alert\_job, 492**
- sp\_add\_log\_shipping\_primary\_database, 492**
- sp\_add\_log\_shipping\_primary\_secondary, 492**
- sp\_add\_log\_shipping\_secondary\_primary, 492**
- sp\_add\_maintenance\_plan, 489**
- sp\_add\_maintenance\_plan\_db, 489**

**sp\_add\_maintenance\_plan\_job**, 489  
**sp\_addmessage**, 485  
**sp\_addtype**, 485  
**sp\_addumpdevice**, 485  
**sp\_altermessage**, 485  
**sp\_attach\_db**, 485  
**sp\_attach\_single\_file\_db**, 485  
**sp\_autostats**, 485  
**sp\_bindefault**, 485  
**sp\_bindrule**, 485  
**sp\_bindsession**, 485  
**sp\_catalogs**, 489  
**sp\_certify\_removable**, 485  
**sp\_change\_log\_shipping\_primary\_database**, 492  
**sp\_change\_log\_shipping\_secondary\_database**, 492  
**sp\_change\_log\_shipping\_secondary\_primary**, 492  
**sp\_cleanup\_log\_shipping\_history**, 492  
**sp\_column\_privileges**, 483  
**sp\_column\_privileges\_ex**, 483, 489  
**sp\_columns**, 483  
**sp\_columns\_ex**, 483, 489  
**sp\_configure**, 485  
**sp\_create\_removable**, 486  
**sp\_createstats**, 486  
**sp\_cursor\_list**, 484  
**sp\_cycle\_errorlog**, 486  
**sp\_databases**, 484  
**sp\_datatype\_info**, 486  
**sp\_dbcmptlevel**, 486  
**sp\_dboption**, 486  
**sp\_dbremove**, 486  
**sp\_delete\_backuphistory**, 486  
**sp\_delete\_log\_shipping\_alert\_job**, 492  
**sp\_delete\_log\_shipping\_primary\_database**, 492  
**sp\_delete\_log\_shipping\_primary\_secondary**, 492  
**sp\_delete\_log\_shipping\_secondary\_database**, 492  
**sp\_delete\_log\_shipping\_secondary\_primary**, 492  
**sp\_delete\_maintenance\_plan**, 489  
**sp\_delete\_maintenance\_plan\_db**, 489  
**sp\_delete\_maintenance\_plan\_job**, 489  
**sp\_depends**, 486  
**sp\_describe\_cursor**, 484  
**sp\_describe\_cursor\_columns**, 484  
**sp\_describe\_cursor\_tables**, 484  
**sp\_detach\_db**, 486  
**sp\_dropdevice**, 486  
**sp\_dropextendedproc**, 486  
**sp\_dropextendedproperty**, 486  
**sp\_droplinkedsrvlogin**, 489  
**sp\_dropmessage**, 486  
**sp\_droptype**, 486  
special-purpose join operations, 241–246  
SPECIFIC (reserved word), 465  
SPECIFIC\_CATALOG view, 515  
SPECIFIC\_NAME view, 515  
SPECIFIC\_SCHEMA view, 515  
SPECIFICTYPE (reserved word), 465  
**sp\_executesql**, 486  
**sp\_fkeys**, 484  
**sp\_foreignkeys**, 489  
**sp\_fulltext\_catalog**, 491  
**sp\_fulltext\_column**, 491  
**sp\_fulltext\_database**, 491  
**sp\_fulltext\_service**, 491  
**sp\_fulltext\_table**, 491  
**sp\_getapplock**, 486  
**sp\_getbindtoken**, 487  
**sp\_help**, 487  
**sp\_helpconstraint**, 487  
**sp\_helpdb**, 487  
**sp\_helpdevice**, 487  
**sp\_helpextendedproc**, 487  
**sp\_helpfile**, 487  
**sp\_helpfilegroup**, 487  
**sp\_help\_fulltext\_catalogs**, 491  
**sp\_help\_fulltext\_catalogs\_cursor**, 491  
**sp\_help\_fulltext\_columns**, 491  
**sp\_help\_fulltext\_columns\_cursor**, 491  
**sp\_help\_fulltext\_tables**, 491  
**sp\_help\_fulltext\_tables\_cursor**, 491  
**sp\_helpindex**, 487  
**sp\_helplanguage**, 487  
**sp\_help\_log\_shipping\_alert\_job**, 492

- sp\_help\_log\_shipping\_monitor\_primary, 492**
- sp\_help\_log\_shipping\_monitor\_secondary, 492**
- sp\_help\_log\_shipping\_primary\_database, 492**
- sp\_help\_log\_shipping\_primary\_secondary, 492**
- sp\_help\_log\_shipping\_secondary\_database, 492**
- sp\_help\_log\_shipping\_secondary\_primary, 492**
- sp\_help\_maintenance\_plan, 489**
- sp\_helpserver, 487**
- sp\_helpsort, 487**
- sp\_helpstats, 487**
- sp\_helptext, 487**
- sp\_helptrigger, 487**
- sp\_indexes, 490**
- sp\_indexoption, 487**
- sp\_invalidate\_textptr, 487**
- sp\_linkedservers, 490**
- sp\_lock, 488**
- sp\_monitor, 488**
- sp\_OACreate, 495**
- sp\_OADestroy, 495**
- sp\_OAGetErrorInfo, 495**
- sp\_OAGetProperty, 495**
- sp\_OAMethod, 495**
- sp\_OASetProperty, 495**
- sp\_OAStop, 495**
- sp\_pkeys, 484**
- sp\_primarykeys, 490**
- sp\_procoption, 488**
- sp\_recompile, 488**
- sp\_refresh\_log\_shipping\_monitor, 492**
- sp\_refreshview, 488**
- sp\_releaseapplock, 488**
- sp\_rename, 488**
- sp\_renamedb, 488**
- sp\_resetstatus, 488**
- sp\_resolve\_logins, 492**
- sp\_send\_dbmail, 493**
- sp\_server\_info, 484**
- sp\_serveroption, 488, 490**
- sp\_setnetname, 488**
- sp\_settriggerorder, 488**
- sp\_spaceused, 488**
- sp\_special\_columns, 484**
- sp\_sproc\_columns, 484**
- sp\_statistics, 484**
- sp\_stored\_procedures, 484**
- sp\_tableoption, 488**
- sp\_table\_privileges, 484**
- sp\_table\_privileges\_ex, 484, 490**
- sp\_tables, 484**
- sp\_tables\_ex, 490**
- sp\_trace\_create, 495**
- sp\_trace\_generateevent, 495**
- sp\_trace\_setevent, 495**
- sp\_trace\_setfilter, 495**
- sp\_trace\_setstatus, 495**
- sp\_unbindefault, 488**
- sp\_unbindrule, 488**
- sp\_updateextendedproperty, 488**
- sp\_updatestats, 488**
- sp\_validname, 488**
- sp\_who, 488**
- sp\_xml\_preparedocument, 508**
- sp\_xml\_removedocument, 508**
- SQL Agent procedures, 504–507**
- SQL Computer Manager, 74**
- SQL expression, modifying, 58–62**
- SQL language**
  - Data Manipulation Language (DML), 107–117
  - nature of SQL and, 106–107
  - origins of, 105–106
- SQL pane in Query Designer tool, 53**
- SQL (reserved word), 463, 465**
- SQL Script, creating views in, 354–356**
- SQL Server**
  - client/server processes, 42
  - conceptual design, 27–28
  - described, 23–25
  - foreign keys, 30
  - identity keys, 30
  - logical design, 28
  - physical design, 28–29
  - primary keys, 29–30
  - query processing, 42–44
  - relationships, 29
  - semantics, 26–27

### **SQL Server (continued)**

- surrogate keys, 30
- terminology, changing, 27–29
- unique identifiers, 30
- who is using, 23–25

### **SQL Server 2000**

- catalogs, managing and populating, 305–306
- described, 25
- Enterprise Edition, 25
- error handling in, 374–377
- full-text index queries, 305–306
- Standard Edition, 25
- templates folder path, 115

### **SQL Server 2005**

- catalogs, managing and populating, 307–313
- described, 25–26
- error handling in, 378
- full-text index queries, 307–313
- (max) option, 12–13
- .NET Framework incorporated with, 3–4
- new features, 3
- schemas, 133–134
- templates folder path, 115
- WITH command, 450

### **SQL Server Books Online, 74**

#### **SQL Server Management Studio**

- docking window placement, 77–80
- down arrow icon, 75
- graphical query designer, 88–93
- Object Explorer, 83–84
- opening, 74
- overview, 74
- projects, 82–83
- queries, writing, 84–86
- scripting options, 87–88
- templates, 94–96
- thumbtack icon, 75
- tool windows, 75–81
- X icon, 75

#### **SQL Server Reporting Services**

- application integration, 444–447
- data caching, 443–444
- datasets, 418–423

- default query view, 419
- features of, 413–414
- fields, adding, 430–431
- format properties, setting, 431
- groups, adding, 436–442
- overview, 413
- parameter lookup list, 424–425
- parameterized queries, 420–423
- queries, 418–423
- Report Designer, 414–418
- report layout, 426–430
- Report Manager, 443
- report parameters, configuring, 431–436
- viewing report with Report Manager, 443

#### **SQL-92 settings, 456**

#### **SQLCA (reserved word), 463**

#### **SQLCMD command-line utility, 99–101**

#### **SQLCODE (reserved word), 463**

#### **SQL\_DATA\_ACCESS view, 516**

#### **SQLERROR (reserved word), 463**

#### **SQL EXCEPTION (reserved word), 465**

#### **SQL\_PATH view, 516**

#### **SQLSTATE (reserved word), 463, 465**

#### **SQLWARNING (reserved word), 463, 465**

#### **SQRT() function, 193, 476**

#### **SQUARE() function, 193, 476**

#### **START (reserved word), 465**

#### **STATE (reserved word), 465**

#### **STATEMENT (reserved word), 465**

#### **STATIC (reserved word), 465**

#### **statistical functions, 210–212**

#### **statistics, displaying, 456**

#### **STATISTICS (reserved word), 461**

#### **STATS\_DATE() function, 202, 482**

#### **STDEV() function, 208, 212, 472**

#### **STDEV() function, 208, 212, 472**

#### **storage and retrieval, 127–128**

#### **storage space, 400**

#### **stored procedures**

- active directory procedures, 483
- business logic, processing, 378–380
- CASE statement, 382–384
- catalog procedures, 483–484

conditional logic, 380–382  
 creation of, 118–119  
 cursor management procedures, 484  
 database engine procedures, 485–488  
 database maintenance procedures, 489  
 Delete procedure, 374  
 distributed queries procedures, 489–490  
 errors, handling and raising, 374–378  
 external systems and extended procedures, 490–491  
 full-text index/search procedures, 491  
 IF statement, 380–382  
 Insert procedure, 371–373  
 insets managed using, 262–263  
 log shipping procedures, 492  
 looping, 384–386  
 mail procedures, 492–493  
 naming, 324  
 notification services procedures, 493–494  
 OLE automation procedures, 495  
 profiler procedures, 495  
 record maintenance, 370  
 replication procedures, 496–501  
 security procedures, 501–504  
 SQL Agent procedures, 504–507  
 SQL Server 2000, error handling in, 374–377  
 SQL Server 2005, error handling in, 378  
 Update procedure, 373–374  
 updating records using, 267–269  
 using parameters, 367–369  
 values, returning, 369–370  
 XML procedures, 508

### **STR() function, 174–175, 479**

#### **string manipulation functions**

ASCII() function, 181–184, 478  
 CHAR() function, 182, 478  
 CHARINDEX() function, 184–185, 478  
 DIFFERENCE() function, 478  
 LEFT() function, 185–186, 478  
 LEN() function, 185, 478  
 LOWER() function, 188–190, 478  
 LTRIM() function, 190–191, 479  
 NCHAR() function, 184, 479  
 overview, 181

PATINDEX() function, 185, 479  
 QUOTENAME() function, 192, 479  
 REPLACE() function, 191, 479  
 REPLICATE() function, 191, 479  
 REVERSE() function, 191–192, 479  
 RIGHT() function, 185–186  
 RTRIM() function, 190–191, 479  
 SOUNDEX() function, 479  
 SPACE() function, 191, 479  
 STR() function, 479  
 STUFF() function, 192, 479  
 SUBSTRING() function, 186–188, 479  
 UNICODE() function, 184, 479  
 UPPER() function, 188–190, 479

### **STRUCTURE (reserved word), 465**

#### **Structured English Query Language (SEQUEL), 105**

### **STUFF() function, 192, 479**

#### **subgrouping, 221**

#### **subqueries**

aggregate functions used in, 286  
 benchmarking, 291–293  
 best practices, 291–293  
 business cases for, 293–296  
 consulting billing time, 294–296  
 correlated subqueries, 290–291  
 derived table, creating, 287  
 embedded SELECT statements, 280–283  
 EXISTS() function, 288–289  
 HAVING clause, 286  
 IN() function, 287–288  
 inner join subqueries, 283–284  
 nested loop inner join, 291  
 NOT EXISTS() function, 289  
 outer join subqueries, 284–285  
 overview, 233, 279  
 scalar expressions, 280–283  
 table aliasing, 286  
 top sales by territory, 293  
 unshipped product orders, 294

### **SUBSTRING() function, 186–188, 479**

#### **SUBSTRING (reserved word), 463**

#### **subtotals, 219–220**

## SUM() function

---

- SUM() function, 167, 208, 209, 472**
- SUM (reserved word), 463**
- surrogate keys, 30, 37**
- SUSER\_SID() function, 200, 478**
- SUSER\_SNAME() function, 200, 478**
- Sybase SQLAnywhere, 20**
- sysmail\_add\_account\_sp, 493**
- sysmail\_add\_principalprofile\_sp, 493**
- sysmail\_add\_profileaccount\_sp, 493**
- sysmail\_add\_profile\_sp, 493**
- sysmail\_configure\_sp, 493**
- sysmail\_delete\_account\_sp, 493**
- sysmail\_delete\_principalprofile\_sp, 493**
- sysmail\_delete\_profileaccount\_sp, 493**
- sysmail\_delete\_profile\_sp, 493**
- sysmail\_help\_account\_sp, 493**
- sysmail\_help\_configure\_sp, 493**
- sysmail\_help\_principalprofile\_sp, 493**
- sysmail\_help\_profileaccount\_sp, 493**
- sysmail\_help\_profile\_sp, 493**
- sysmail\_start\_sp, 493**
- sysmail\_stop\_sp, 493**
- sysmail\_update\_account\_sp, 493**
- sysmail\_update\_principalprofile\_sp, 493**
- sysmail\_update\_profileaccount\_sp, 493**
- sysmail\_update\_profile\_sp, 493**
- system availability requirements, 400**
- system functions and variables**
  - APP\_NAME() function, 200, 480
  - COALESCE() function, 200, 202–203, 480
  - COLLATIONPROPERTY() function, 200, 480
  - CURRENT\_TIMESTAMP() function, 200, 480
  - CURRENT\_USER() function, 200, 480
  - DATALENGTH() function, 200, 203–204, 480
  - ENCRYPT() function, 480
  - @@ERROR, 204, 470
  - fn\_Get\_SQL() function, 480
  - fn\_HelpCollations() function, 480
  - fn\_helpcollations() function, 200
  - fn\_ServerSharedDrives() function, 480
  - fn\_serversharedrives() function, 200
  - fn\_VirtualFileStats() function, 480
  - fn\_virtualfilestats() function, 201
  - FORMATMESSAGE() function, 201, 480
  - GETANSINULL() function, 201, 480
  - HOST\_ID() function, 201, 481
  - HOST\_NAME() function, 201, 481
  - IDENT\_CURRENT() function, 201, 481
  - IDENT\_INCR() function, 201, 481
  - @@IDENTITY, 204, 470
  - IDENTITY() function, 201, 481
  - IDENT\_SEED() function, 201, 481
  - ISDATE() function, 201, 481
  - ISNULL() function, 201, 481
  - ISNUMERIC() function, 201, 481
  - NEWID() function, 201, 481
  - NULLIF() function, 201, 481
  - overview, 200, 471–482, 480–482
  - PARSENAME() function, 201, 481
  - PERMISSIONS() function, 201, 481
  - PWDCONPARE() function, 481
  - PWDENCRYPT() function, 482
  - @@ROWCOUNT, 204, 470
  - ROWCOUNT\_BIG() function, 201, 482
  - SCOPE\_IDENTITY() function, 202, 482
  - SERVERPROPERTY() function, 202, 482
  - SESSIONPROPERTY() function, 202, 482
  - SESSION\_USER() function, 202, 482
  - STATS\_DATE() function, 202, 482
  - SYSTEM\_USER() function, 202, 482
  - @@TRANCOUNT, 204, 470
  - USER\_NAME() function, 202, 482
- system global variables, 467–471**
- system integration and data exchange, 412**
- system statistical functions and variables**
  - @@CONNECTIONS, 204, 470
  - @@CPU\_BUSY, 204, 470
  - fn\_virtualfilestats() function, 482
  - @@IDLE, 204, 470
  - @@IO\_BUSY, 204, 470
  - overview, 204, 470–471, 482
  - @@PACKET\_ERRORS, 204, 471
  - @@PACK\_RECEIVED, 204, 470
  - @@PACK\_SENT, 204, 470
  - @@TIMETICKS, 204, 471
  - @@TOTAL\_ERRORS, 204, 471

@@TOTAL\_READ, 204, 471

@@TOTAL\_WRITE, 204, 471

**system-supplied aggregate functions used to summarize column values, list of, 208**

**SYSTEM\_USER() function, 202, 482**

**SYSTEM\_USER (reserved word), 461, 463**

## T

**table aliasing, 286**

**TABLE (reserved word), 461, 463**

**TABLE\_CATALOG view, 509, 510, 511, 513, 514, 517, 518**

**TABLE\_CONSTRAINTS views**

CONSTRAINT\_CATALOG view, 517

CONSTRAINT\_NAME view, 517

CONSTRAINT\_SCHEMA view, 517

CONSTRAINT\_TYPE view, 517

INITIALLY\_DEFERRED view, 517

IS\_DEFERRABLE view, 517

TABLE\_CATALOG view, 517

TABLE\_NAME view, 517

TABLE\_SCHEMA view, 517

**TABLE\_NAME view, 510, 511, 513, 514, 517, 518**

**TABLE\_PRIVILEGES views**

GRANTEE view, 517

GRANTOR view, 517

IS\_GRANTABLE view, 517

PRIVILEGE\_TYPE view, 517

TABLE\_CATALOG view, 517

TABLE\_SCHEMA view, 517

**tables**

columns, 5, 8

creation of, 117–118

described, 5–6

naming, 324

nullability, 13

primary keys, 6–7

rows, 5

viewing, 58

**TABLES views**

TABLE\_CATALOG view, 517

TABLE\_NAME view, 517

TABLE\_SCHEMA view, 517

TABLE\_TYPE view, 517

**TABLE\_SCHEMA view, 510, 511, 513, 514, 517, 518**

**TAN() function, 193, 476**

**templates, 94–96, 114–115**

**templates folder path, 115**

**TEMPORARY (reserved word), 463, 465**

**TERMINATE (reserved word), 465**

**terminology, changing, 27–29**

**TEXTPTR() function, 475**

**@@TEXTSIZE, 469**

**TEXTSIZE (reserved word), 461**

**TEXTVALID() function, 475**

**THAN (reserved word), 465**

**THEN (reserved word), 461, 463**

**third normal form — 3NF, 32**

**thumbtack icon in SQL Server Management Studio, 75**

**TIME (reserved word), 463, 466**

**TIMESTAMP (reserved word), 463, 466**

**@@TIMETICKS, 204, 471**

**TIMEZONE\_HOUR (reserved word), 463, 466**

**TIMEZONE\_MINUTE (reserved word), 463, 466**

**TO (reserved word), 461, 463**

**tool windows in SQL Server Management Studio, 75–81**

**toolbar options in Query Designer tool, 57–58**

**tools for SQL Server 2000**

Books Online, 49

Configure SQL XML Support in IIS option, 49

Enterprise Manager, 49, 50–51

Import and Export Data shortcut, 49

list of, 49

Profiler, 49

Query Analyzer, 49

Reporting Services, 49

Server Network Utility, 49

Service Manager, 49

**tools for SQL Server 2005**

business Intelligence Development Studio, 73

Database Tuning Advisor, 73

list of, 73–74

Profiler, 73

### tools for SQL Server 2005 (continued)

- Report Manager, 73
- SQL Computer Manager, 74
- SQL Server Books Online, 74
- SQL Server Management Studio, 74–96

**Tools menu in Query Analyzer, 64**

**TOP command, 152–154, 451**

**TOP (reserved word), 461**

**top sales by territory and subqueries, 293**

**top values, 152–154**

**@@TOTAL\_ERRORS, 168, 204, 471**

**@@TOTAL\_READ, 168, 204, 471**

**totals, 219–220**

**@@TOTAL\_WRITE, 204, 471**

**TRAILING (reserved word), 463, 466**

**TRAN (reserved word), 461**

**@@TRANCOUNT, 204, 470**

**transaction management**

- locking options, 393–395
- overview, 393

**TRANSACTION (reserved word), 461, 463**

**transactions**

- handling, 456
- logs, 255–256
- overview, 16

**Transact-SQL**

- ANSI standard, 2
- demise of, premature call for, 3–4
- described, 1–2
- as programming language, 2–3
- as query language, 2–3

**TRANSLATE (reserved word), 463**

**TRANSLATION (reserved word), 463, 466**

**TREAT (reserved word), 466**

**TRIGGER (reserved word), 461**

**triggers**

- creation of, 119
- naming, 324

**TRIM (reserved word), 463**

**TRUE (reserved word), 463, 466**

**TRUNCATE (reserved word), 461**

**TRUNCATE TABLE statement, 272–273**

**TSEQUAL (reserved word), 461**

**TYPE\_UDT\_CATALOG view, 515**

**TYPE\_UDT\_NAME view, 516**

**TYPE\_UDT\_SCHEMA view, 516**

## U

**UDT\_CATALOG view, 515**

**UDT\_NAME view, 515**

**UDT\_SCHEMA view, 515**

**UML (Unified Modeling Language), 33, 398**

**UNDER (reserved word), 466**

**UNICODE() function, 184, 479**

**UNION command, 453**

**union queries**

- federating data, 249–250

- overview, 247–249

- partitioning data, 249–251

- performance and, 250–251

- scaling considerations, 250–251

**UNION (reserved word), 461, 463**

**unique constraint, 333**

**unique identifiers, 30**

**UNIQUE (reserved word), 461, 463**

**UNIQUE\_CONSTRAINT\_CATALOG view, 513**

**UNIQUE\_CONSTRAINT\_NAME view, 513**

**UNIQUE\_CONSTRAINT\_SCHEMA view, 513**

**UniquelIdentifier type, 330–331**

**UNKNOWN (reserved word), 463, 466**

**UNNEST (reserved word), 466**

**unshipped product orders and subqueries, 294**

**UPDATE command, 455**

**update locks, 17**

**Update procedure, 373–374**

**UPDATE (reserved word), 461, 463**

**UPDATE statement, 264–269**

**UPDATE\_RULE view, 514**

**UPDATETEXT (reserved word), 461**

**UPPER() function, 188–190, 479**

**UPPER (reserved word), 464**

**USAGE (reserved word), 464, 466**

**USE (reserved word), 461**

**user interface, 402–403**

**USER (reserved word), 461, 464**

**user-defined aggregate function, 212**  
**user-defined function, creation of, 119–120**  
**user-defined functions**  
 inline table-valued functions, 390–391  
 multi-statement table-valued functions, 391–393  
 overview, 386  
 scalar functions, 386–390

**USER\_ID() function, 478**  
**USER\_id() function, 200**  
**USER\_NAME() function, 200, 202, 482**  
**USER\_NAME option, 344**  
**users accessing data, number of, 399**  
**USING (reserved word), 464, 466**

## V

**VALUE (reserved word), 464, 466**  
**VALUES (reserved word), 461, 464**  
**values, returning, 369–370**  
**VAR() function, 208, 210–211, 472**  
**VARCHAR (reserved word), 464, 466**  
**VARIABLE (reserved word), 466**  
**VARP() function, 208, 211, 472**  
**VARYING (reserved word), 461, 464**  
**@@VERSION, 169, 469**  
**version control, 116–117**  
**Viera, Rob**  
*Beginning SQL Server 2005 Programming*, 1  
*Professional SQL Server 2000 Programming*,  
 1, 304  
*Professional SQL Server 2005 Programming*,  
 1, 304

**VIEW (reserved word), 461, 464**  
**VIEW\_CATALOG view, 518**  
**VIEW\_COLUMN\_USAGE views**  
 COLUMN\_NAME view, 518  
 TABLE\_CATALOG view, 518  
 TABLE\_NAME view, 518  
 TABLE\_SCHEMA view, 518  
 VIEW\_CATALOG view, 518  
 VIEW\_NAME view, 518  
 VIEW\_SCHEMA view, 518

**VIEW\_DEFINITION view, 518**  
**VIEW\_NAME view, 518**

**views**  
 complexity, hiding, 364–365  
 creating views, 351–356  
 creation of, 118  
 federated views, 360–363  
 Microsoft Access, creating views in, 351–353  
 modifying data through views, 365–366  
 naming, 324  
 partitioned views, 357–360  
 rows, ordering, 356–357  
 securing data, 363  
 SQL Script, creating views in, 354–356  
 updating data transactions using, 267  
 views, 350–366  
 virtual tables, 350  
 Visual Studio, creating views in, 353–354

**VIEWS**  
 CHECK\_OPTION view, 518  
 IS\_UPDATABLE view, 518  
 TABLE\_CATALOG view, 518  
 TABLE\_NAME view, 518  
 TABLE\_SCHEMA view, 518  
 VIEW\_DEFINITION view, 518

**VIEW\_SCHEMA view, 518**  
**VIEW\_TABLE\_USAGE views**  
 TABLE\_CATALOG view, 518  
 TABLE\_NAME view, 518  
 TABLE\_SCHEMA view, 518  
 VIEW\_CATALOG view, 518  
 VIEW\_NAME view, 518  
 VIEW\_SCHEMA view, 518

**virtual tables, 350**  
**Visual Studio, creating views in, 353–354**

## W

**WAITFOR (reserved word), 461**  
**Web server application, 409–410**  
**weighting values, 317**  
**WHEN (reserved word), 461, 464**  
**WHENEVER (reserved word), 464, 466**  
**WHERE clause**  
 joining tables in, 233–234  
 overview, 138–148, 270, 452

## WHERE (reserved word)

---

**WHERE (reserved word)**, 461, 464  
**WHILE (reserved word)**, 461  
**Wiley Publishing, Inc. (*Professional SQL Server Reporting Services*)**, 414  
**WITH CHECK option**, 338  
**WITH command**, 450  
**WITH CUBE operator**, 222–223  
**WITH ENCRYPTION option**, 338–339, 342  
**WITH RECOMPILE statement**, 343  
**WITH (reserved word)**, 461, 464  
**WITH ROLLUP operator**, 221–222  
**WITH TIES statement**, 154–155  
**WITHOUT (reserved word)**, 466  
**WORK (reserved word)**, 464, 466  
**WRITE (reserved word)**, 464, 466  
**WRITETEXT (reserved word)**, 461

## X

**X icon in SQL Server Management Studio**, 75  
**XML (eXtensible Markup Language)**  
overview, 412  
procedures, 508

**xp\_cmdshell**, 490  
**xp\_enumgroups**, 490  
**xp\_findnextmsg**, 490  
**xp\_grantlogin**, 490  
**xp\_logevent**, 490  
**xp\_loginconfig**, 490  
**xp\_logininfo**, 490  
**xp\_msver**, 490  
**xp\_readmail**, 493  
**xp\_revokelogin**, 490  
**xp\_sendmail**, 493  
**xp\_sprintf**, 491  
**xp\_sqlmaint**, 491  
**xp\_sscanf**, 491

## Y

**YEAR() function**, 181, 474  
**YEAR (reserved word)**, 464, 466

## Z

**ZONE (reserved word)**, 464, 466











