



Index

Symbols

- & substitution variable 38
- && substitution variable 36, 38
- + outer join 154
- . CONCAT character 36
- /* comment start 18
- */ comment end 18
- ANSI/ISO comment 18
- @ command 28
- \ ESCAPE character 26, 37, 65, 314

Numerics

- 1-to-many relationships. *See*
one-to-many relationships
- 2 objects selected from *N* objects 302–303
- 2000, year of 77–79

A

- ACCEPT command 38, 50, 316
- ad hoc queries
 - alternative methods for 135, 228
 - anticipating errors 217
 - combining into one report 46, 68
 - coordinating commands 173
 - costs of unasked xiii
 - debugging tools 217
 - elegance in 291
 - future vision xiii
 - generalizing 28–32
 - SELECT command and 107
 - simplifying 107, 195, 296

- standards 1, 10–21
 - documentation 18–19
 - formatting 20
 - modularizing 12–17
 - structured SQL 10–11
 - tools available for xiv, 174
 - problems with xiv–xv
 - See also* performance
- ADD_MONTHS. *See* date functions
- address hierarchies 109–115
 - performance comparisons 258–259
 - PL/SQL approach 112–114
 - SQL approach 110–111
- address_hierarchy function. *See* PL/SQL functions
- age calculated 76
- aliases
 - COMPUTE command and 184
 - created with AS 124, 125
 - FROM clause subqueries and 250
 - ORDER BY clause and 124, 126
- ALL_CATALOG. *See* SYS views
- ampersands
 - changing substitution character 37
 - distinguishing from substitution variables 26, 37, 65, 66
- anonymous PL/SQL blocks. *See* PL/SQL blocks, anonymous
- arguments
 - blank spaces and 28–29
 - operating system and 31–32
 - passing to a query 28–32
 - query construction and 29–31

- sources of errors 28
- specifying 28–29
- AS. *See* aliases
- ASCII character set
 - sorting order 122
- ASCII flat files
 - producing 62–63, 212, 214, 242
- averages
 - requiring two reports 46
- AVG. *See* group functions
- B**
- BETWEEN. *See* operators
- bind variables
 - assigning values to 71
 - computing a median 246, 286
 - computing a percentage 272
 - declaring 71
 - examples of 70, 200, 204
 - in top *N* reports 274
 - report titles and 70–71
 - restricting report results 304
 - sampling the *N*th row 235, 282
 - selecting a random sample 241
- blank lines
 - creating for preprinted forms 190
 - eliminating 41, 57, 208
- blanks, trailing
 - suppressing 63
- block reports 186–189
 - aligning data values 188
 - custom headings 206
 - described 186
 - eliminating blank lines 208
 - general format 54
 - labeling data columns 187
 - partially blocking 188–189
- bottom title. *See* BTITLE command
- BREAK command
 - clearing 313
 - double-spacing and 39
 - frequency distributions and 174
 - implicitly defining a user variable 52
 - ROW 48
 - SKIP 19
 - SKIP PAGE 48, 190
 - suppressing redundant information 44, 57, 192
- BTITLE command
 - bind variables and 70
 - examples of 62
 - forcing queries to a new page 68
 - printer escape sequences and 64
- business days
 - described 86
 - due dates and 86
- business populations
 - defining with SQL segments 16
 - See also* WHERE clause
- C**
- carriage returns 186
- cartesian product 291
 - examples of 105, 199, 302, 303
- character data
 - checking for numeric values 102–103
 - locating an embedded string 129
- character functions
 - ASCII 210
 - CHR 66, 100
 - INSTR 22, 129, 210, 211
 - LOWER 25
 - LPAD 48, 188
 - LTRIM 329
 - REPLACE 100
 - RPAD 214
 - RTRIM 74
 - SUBSTR 22, 128, 210
 - UPPER 42, 73
- check constraints. *See* constraints
- child nodes
 - described 140
- CHR. *See* character functions
- CLEAR command 313
- clears.sql. *See* SQL programs
- COLSEP. *See* SET command
- COLUMN command
 - clearing 313
 - examples of 20, 22
 - FOLD_AFTER 48
 - HEADING 98
 - LONG datatypes and 168
 - NEW_VALUE 23, 25, 52

- NEWLINE 186, 188
- NOPRINT 25, 40, 122
- NULL 60, 61, 208
- rounding implicitly 104
- WORD_WRAPPED 19, 56
- column headings. *See* headings
- columns, report
 - alternatives to compressing 54
 - compressing 54, 59
 - creative headings 56
 - default display formats 55, 58
 - default separation 55
 - hiding columns 36, 40
 - LONG default formats 168
- combining reports. *See* SET command, EMBEDDED 68
- comments
 - delimiters 18–19
 - restrictions on 19
 - See also* REMARK
- compound queries
 - described 154
 - See also* UNION, UNION ALL, INTERSECT, and MINUS set operations
- compressing report columns. *See* columns, report
- COMPUTE command
 - AVG 245, 247
 - clearing 313
 - frequency distributions and 174
 - ON REPORT 96, 104
 - SUM 47
- CONCAT. *See* SET command
- concatenation
 - decreasing report column size 56
 - NULL values and 210
 - single quotes and 42
 - suffixing substitution variables 36
- CONNECT BY clause 137, 142
- constraints, integrity 8–9, 326
 - check 9
 - described 326
 - enabled or disabled 326
 - referential 9, 177
 - report of 8
- constrnt.sql. *See* SQL programs
- control characters
 - removing from strings 100
- conventions used in book xvii
- conversion functions
 - CHARTOROWID 235
 - CONVERT 122, 123
 - described 73
 - ROWIDTOCHAR 198, 235
 - TO_CHAR for dates 58, 59, 74, 80, 127
 - TO_CHAR for numbers 97, 186
 - TO_DATE 77, 78
 - TO_NUMBER 102, 128
- CONVERT. *See* conversion functions 122
- correlated subqueries
 - alternatives to 152
 - balanced and unbalanced 151
 - defining report populations 148
 - equivalent PL/SQL functions 13, 152
 - equivalent set operations 142
 - examples of 15, 16, 111, 148–151, 223
 - falling through joins and 222
 - finding nonunique combinations of data columns 293
 - multiplying through joins and 230
 - performance and 309
 - performance comparisons 260–261
 - views and 14
 - See also* subqueries
- COUNT function. *See* group functions
- COUNT(*). *See* group functions.
- counting
 - across two tables 300–301
- CREATE VIEW command
 - examples of 15, 305
 - sorting restrictions 117
- cross-tabulated reports 178–185
 - by days of week 298–299
 - described 178
 - examples of 178
 - identity matrix tables and 181–183, 185
 - percentage distributions in 200
 - performance comparisons 268–269
 - requiring knowledge of data values 181

- rotated tables 181
 - sums and 184–185
 - traditional 180–183
 - cursors, PL/SQL. *See* PL/SQL cursors
- D**
- data warehouses xv
 - database exploration 2–9
 - reports useful for 1
 - datatypes. *See* DATE, VARCHAR2, NUMBER, and LONG datatypes
 - date arithmetic 74–91
 - subtraction 82
 - DATE datatypes
 - date and time components 80
 - default display width 55
 - date functions
 - ADD_MONTHS 75
 - MONTHS_BETWEEN 76
 - NEXT_DAY 84, 85
 - SYSDATE 241, 313, 314
 - TRUNC 82, 84, 116, 298
 - date ranges
 - calculating duration 82–83
 - creating 75
 - recoding into weeks 298
 - dates
 - different centuries 78
 - failing WHERE conditions 80–81
 - GROUP BY and 116
 - stamping reports with date 313
 - stamping reports with time 314
 - time component of 80, 116
 - datestamping reports 313
 - DBA tools and queries 228–231
 - DBMS_OUTPUT package 238
 - debugging queries 218–227
 - DECODE function 92–96
 - assigning descriptions to recoded data 96, 295
 - complexity reached quickly 93
 - computing a median 246
 - computing due dates 254
 - concatenating strings when NULL values possible 210
 - cross-tabulated reports and 268, 298
 - examples of 42, 60, 87, 128, 164, 295
 - NULL values and 60, 61, 146, 210
 - outer joins and 163
 - recoding numeric data 92, 256, 295
 - relational operators and 92–93
 - rotated table reports and 181
 - defdict.sql. *See* SQL programs
 - defheadr.sql. *See* SQL programs
 - DEFINE command 22, 24, 323
 - default editor and 25
 - implicit. *See* user variables
 - DESCRIBE command 4
 - dictionaries, data column 4–5, 322–323
 - described 322
 - report of 4
 - table comments for 42
 - discount function. *See* PL/SQL functions
 - DISTINCT
 - finding its opposite 292–293
 - sort key too long 121
 - double quotes
 - arguments and 29, 31
 - double-spacing a report 36, 39
 - DUAL table
 - compound queries and 237, 246
 - creating header line in flat file 213
 - creating printer escape sequences 66
 - datestamping a report 314
 - implicitly defining a user variable 52
 - due dates
 - algorithm ignoring holidays 87
 - calculating 86–91
 - holidays and 90–91
 - performance comparisons 254–255
 - PL/SQL approach 88–89
 - SQL approach 86–88
 - due_date function. *See* PL/SQL functions
 - due_date1 function. *See* PL/SQL functions
 - dummy queries
 - See* DUAL table
 - duplicates
 - finding 230–231, 279
 - UNION and 126
- E**
- EBCDIC character set
 - sorting order 122
 - ECHO. *See* SET command

- EDIT command 25
- editor, setting default 25, 318
- errors
 - ORA-00904 30
 - ORA-00936 29
 - ORA-01452 230
 - ORA-01467 121
 - ORA-01468 154
 - ORA-01719 163
 - ORA-01799 159
- ESCAPE character 26, 37, 65
- escape sequences, printer 64, 65
- EXCEPTION. *See* PL/SQL language elements
- EXECUTE command 70, 71, 132, 241
- EXISTS 141
 - correlated subqueries and 110, 148, 155
- EXPLAIN PLAN
 - hierarchical data relationships in 140
- F**
- falling through joins. *See* joins and outer joins
- FAQ. *See* frequently asked questions
- FEEDBACK. *See* SET command 212
- FETCH. *See* PL/SQL language elements
- fixed-format flat files 214–215
 - See also* ASCII flat files
- FLOOR function. *See* number functions
- FOR LOOP. *See* PL/SQL language elements
- foreign keys. *See* keys, foreign
- format elements
 - D 87
 - DAY 74, 84, 298
 - HH24 83
 - RR 78
 - SSSS 294
 - YYYY 78
- formatting reports 206–211
- formfeed 36, 314
 - disabling to combine reports 46
 - forcing 48–49, 68
- frequency distribution reports 174–177
 - by time interval 294–297
 - including code descriptions 176–177
 - performance comparisons 297, 306
 - returning maximum count 304–306
 - table owners report 319
- frequently asked questions
 - how collected xv
 - implications of focus on xvi–xvii
- FROM clause
 - described 107
 - including subqueries in 200, 245, 250, 288, 301, 302, 304
 - placement of views in 199
- full table scans
 - forcing 119
- functions
 - single-row or group described 73
 - See also* character, conversion, date, group, and number functions
- G**
- get_address_line function. *See* PL/SQL functions
- get_city_line function. *See* PL/SQL functions
- getcolmn.sql. *See* SQL programs
- getinstc.sql. *See* SQL programs
- getrand. *See* PL/SQL procedures 241
- getuser.sql. *See* SQL programs
- GLOBAL_NAME. *See* SYS views
- glogin.sql. *See* SQL*PLUS environment
- grand totals
 - calculating 175, 319
- GROUP BY clause 116–120
 - computing mode 288
 - dates and 116
 - debugging join errors 226, 230
 - described 107, 116
 - finding nonunique combinations of data columns 293
 - frequency distributions and 174
 - index interactions and 118
 - ORDER BY clause and 117–120
 - sampling the *N*th row 234, 282
 - sort key too long 121
- group functions
 - AVG 47, 244
 - COUNT 99, 146, 219
 - COUNT(*) 99

described 73, 244
 MAX 148, 149, 165, 249
 MIN 309
 NULL values and 146
 SUM 99, 157, 180, 184, 299

H**HAVING** clause

computing median 246, 286
 computing mode 249, 288
 debugging join errors 226, 230
 described 107, 116
 finding nonunique combinations of
 data columns 293
 identifying rows multiplying through
 joins 279
 sampling the *N*th row 234, 282
 systematic sampling 284

headings

block reports and 186
 building custom 206, 322
 creating default 20, 316
 fitting column width 56
 suppressing 46, 62
See also COLUMN command 98

HEADSEP. *See* SET commandhiding columns. *See* columns, report

hierarchical queries 136–143

default sorting order 136
 described 136
 inverted 142–143
 LEVEL described 136
 selecting leaves 140–141
 sorting 136–141

hints. *See* optimizer hints

HP LaserJet printers

formatting reports for 64–66

I

identity matrix tables

cross-tabulated reports and 181, 185,
 268
 described 181

if-then-else logic

DECODE function and 92

implicit DEFINE. *See* user variables**IN** operator

NULL values and 145
 outer joins and 163–167, 265–267

INDEX_ASC. *See* optimizer hints

indexes 6–7

forcing sort order 130, 236
 leading edge
 described 7
 effect on performance 324
 examples of 118, 131
 performance and 7, 324
 reports of 6, 324

indexes.sql. *See* SQL programs**INSERT** command

sorting and 130–132
 PL/SQL approach 132–133
 SQL approach 130–132
 sorting implicitly
 UNION approach 131
 sorting restrictions 117

INSTR. *See* character functions

integers

identifying with MOD 102

integrity constraints. *See* constraints

interquartile range statistic 233

INTERSECT set operation 154

described 154

is_pos_integer function. *See* PL/SQL functions**J**

joins

criteria cannot be predetermined 16
 debugging interactions in 227
 detecting errors in 218–221
 equivalent set operations 154
 errors falling through 109, 144, 220
 detecting 228–229
 identifying source of 222–224
 performance comparisons
 276–278
 types of problems 223
 errors multiplying through 109, 220
 detecting 226
 identifying source 226–227, 230
 performance comparisons
 279–281

- types of errors 226
 - nonequi 194, 203, 270
 - self-join described 194
 - source of most query problems 218
 - stepping through 220
 - types of errors 220
 - See also* outer joins
- K**
- keys 6–7
 - error of duplicate 230, 279
 - foreign 9
 - finding violations 228–229
 - primary 6
 - in simulations 254
 - unique 6
- L**
- landcomp.sql. *See* SQL programs
- landscape print orientation 66, 315
- leaf nodes
 - described 140
 - displayed graphically 140
 - examples of 140
- left-justifying numeric data 97–98
- LEVEL. *See* pseudocolumns
- LIKE. *See* operators
- line numbers
 - default action of queries 68
- line size
 - alternatives to compressing 54
 - compressing report to fit 54–59, 328
 - computing minimum necessary 54, 328–329
 - examples of 56
 - settings 26
- linear congruential method
 - generating random numbers 240
- linefeed
 - forcing 323
- linesize.sql. *See* SQL programs
- literals
 - forcing sort order 126
- login.sql 318
 - defining path names 24, 318
 - examples of 24
 - setting default editor 25, 318
 - See also* SQL*PLUS environment
- LONG datatypes 55, 168–169
 - default formats 168, 329
 - formatting 168, 328
 - size limits 168
 - sort key too long 121
- LONG system variable. *See* SET command
- LONGCHUNKSIZE. *See* SET command
- LOWER. *See* character functions
- LPAD. *See* character functions 188
- M**
- mail merges 212–213
- master-detail reports 192–193
 - described 192
 - examples of 192
- MAX. *See* group functions
- max_address function. *See* PL/SQL functions
- max_regseq function. *See* PL/SQL functions
- median statistic 233, 244–246
 - compared to mean 244
 - described 244
 - performance comparisons 286
- MIN. *See* group functions
- MINUS set operation 154
 - described 154
 - detecting join errors 229, 276
 - identifying leaves 142
- MOD. *See* number functions
- mode statistic 249–251
 - described 249
 - performance comparisons 286–289
- MONTHS_BETWEEN. *See* date functions
- multiplying through joins. *See* joins
- N**
- name_now function. *See* PL/SQL functions
- National Language Support
 - DATE datatypes and 55
- negative numbers. *See* number functions, SIGN
- new page. *See* formfeed
- new_sort_key. *See* PL/SQL functions

- NEW_VALUE. *See* COLUMN command
- NEWLINE. *See* COLUMN command 186
- NEXT_DAY. *See* date functions
- NLS. *See* National Language Support
- NLS_DATE_FORMAT 78, 329
RR format 79
- NLS_SESSION_PARAMETERS. *See* SYS
views
- NLS_TERRITORY 84, 298
- nodes. *See* root, child, parent, and leaf
nodes
- NOPRINT. *See* COLUMN command
- NOT EXISTS. *See* EXISTS
- NULL values 144–147
!= (not equal) and 144
avoiding display of 208
changing display of 60
concatenation and 210
counting 99, 146–147
DECODE function and 60, 61
described 144
IN problems 145
joins and 7
NVL function and 60
outer joins and 157
preprinted forms and 190
referential constraints and 9
sorting order and 175
unique keys and 6
- NUMBER datatypes 5
left-justifying report data 97
precision 5, 322
scale 5
- number functions
FLOOR 76, 82
MIN 309
MOD 86, 102, 238, 239, 284
ROUND 104, 199, 201
SIGN 92, 102, 210, 256, 295
TRUNC 86, 129, 240, 245
- NUMWIDTH. *See* SET command
- NVL function 60, 146, 157, 180, 210
- O**
- one-to-many relationships
described 192
finding incomplete 228–229
- online_duration. *See* PL/SQL functions
- operators
!= (not equal) and NULL values 144
BETWEEN 299
DECODE and relational 92–93
LIKE 80
See also IN and OR
- optimizer hints
FULL 120
INDEX_ASC 130, 131
- optimizer. *See* rule-based optimizer
- OR operator 160
outer joins and 163–167, 265–267
- ORA ERRORS. *See* errors
- Oracle instance
including in SQL prompt 22, 25
- Oracle7
release versions used xvi
- ORDER BY clause 121–133
aliases and 124
coordinating with BREAK 44
DESC 40
described 107, 121
EBCDIC sorting 122
GROUP BY clause and 117
hierarchical queries and 137
position notation 124
redundant information and 44
restricted from subqueries 117
ROWNUM and 202, 236
running total reports and 194
sequential numbering and 128
sort key too long 121
substitution variables and 124
UNION set operations and 126–127
- outer joins 156–167
as a debugging tool 228, 276
equivalent set operations 135, 155,
160
errors falling through 156–158
examples of 115, 158, 160, 176
expressions and 158
full outer join described 154
IN or OR operators and 163–167,
265–267
NULL values and 157, 176
performance comparisons 262–263

- replacing regular joins with 224
 - rules ensuring accuracy 157
 - subqueries and 159–162
 - owncol.sql. *See* SQL programs
 - owners.sql. *See* SQL programs
- P**
- page numbers
 - default action of queries 68
 - how incremented 68
 - sequencing in multiple reports 68
 - page size
 - paper size and PAGESIZE 48
 - settings 26, 36
 - See also* SET command 36
 - parent nodes
 - described 140
 - examples of 140
 - path names
 - advantages of defining 318
 - examples of 18
 - login.sql and 24, 318
 - pausing screen displays. *See* screen displays
 - percentage distribution reports 199–201
 - performance comparisons 272–273
 - PL/SQL approach 200
 - performance
 - criteria used in simulations 253
 - improving 307–311
 - indexes and 324
 - leading edge of index and 7
 - non-equi joins and 203
 - placement of view in FROM 199
 - ROWID and 114
 - running total reports and 194
 - tradeoffs with developer's time 228
 - tradeoffs with elegance 291, 296
 - performance comparisons
 - address hierarchies 258–259
 - correlated subqueries 260–261, 310
 - cross-tabulated reports 268–269
 - due dates 254–255
 - errors falling through joins 276–278
 - errors multiplying through joins 279–281
 - finding nonunique data columns 293
 - frequency distribution reports 296, 306
 - median statistic 286
 - mode statistic 286–289
 - outer joins to subqueries 262–263
 - outer joins with IN or OR 265–267
 - percentage distribution reports 272–273
 - recoding numeric data 256–257
 - running total reports 270–271
 - selecting the *N*th row 282–283
 - systematic samples 284–285
 - top *N* reports 274–275
 - pitch, compressed 66
 - PL/SQL blocks, anonymous
 - computing a median 246
 - producing a top *N* report 204
 - sampling the *N*th row 235
 - selecting a random sample 241
 - PL/SQL cursors 70
 - coordinating with main query 196, 197
 - sorting an INSERT 132
 - PL/SQL functions
 - address_hierarchy 112, 258
 - discount 93
 - due_date 88, 254
 - due_date1 90
 - get_address_line 209
 - get_city_line 189
 - is_pos_integer 102
 - max_address 111, 227
 - max_regseq 152, 162
 - name_now 13
 - new_sort_key 138
 - online_duration 296
 - running_fee 195
 - running_fee3 198
 - sales_category 95
 - simplifying queries with 14, 17, 94, 112, 162, 195, 296
 - sponsor_exist 166
 - time_cat 296
 - PL/SQL language elements
 - %ROWTYPE 139
 - CLOSE 13
 - CURSOR 13

- DECLARE 205
 - EXCEPTION 102, 103
 - FETCH 13, 114
 - FOR LOOP 132, 204, 243, 284
 - IF . . . ELSIF 93, 95
 - OPEN 13
 - REF CURSOR 70
 - TYPE 70
 - PL/SQL packages
 - DBMS_OUTPUT 238, 284
 - PL/SQL procedures
 - random number generator 240
 - populations, report
 - central importance in queries 217
 - correlated subqueries and 148
 - defined by WHERE clause 109, 218
 - examples of 218
 - multiple populations 174
 - testing for accuracy of 218
 - See also* joins
 - portrait print orientation 315
 - position notation
 - in ORDER BY clause 124
 - sorting compound queries 126
 - positive numbers. *See* number functions, SIGN
 - precision. *See* NUMBER datatypes
 - preprinted forms 190–191
 - aligning data values 190
 - forcing a formfeed 48
 - NULL values and 190
 - primary keys. *See* keys, primary
 - PRINT command 70, 71
 - printer.sql. *See* SQL programs
 - PROMPT command 38, 50, 124
 - creating a blank line 51
 - prompting users
 - concatenating two responses 52
 - converting mixed-case responses 52
 - customizing prompts 50–51
 - disabling VERIFY 316
 - examples of 28, 316, 317, 324, 329
 - substitution variables and 38
 - verifying accuracy of response 52
 - pseudocolumns
 - LEVEL 136
 - ROWNUM 202, 234, 238, 241
 - computing a median 246
 - selecting two objects from *N* objects 303
 - See also* ROWID 197
- Q**
- quotes. *See* single or double quotes
- R**
- random number generators 240
 - seeding 241
 - random samples 240–242
 - random number generators and 240
 - READ ONLY. *See* SET TRANSACTION command
 - recoding numeric data 95–96
 - assigning meaningful codes 95
 - performance comparisons 256–257
 - time durations 294
 - record separators
 - See also* SET command, RECSEPCHAR and RECSEP
 - record separators
 - described 41
 - redundant report information 36
 - ORDER BY clause and 44
 - suppressing 44
 - examples of 57, 192
 - REFCURSOR. *See* VARIABLE command
 - referential constraints. *See* constraints
 - REMARK command 19
 - REPLACE. *See* character functions
 - report columns. *See* columns, report
 - reports
 - datestamping 314
 - other application software and 212–215, 315
 - screen displays of 314
 - spooling to a file 315
 - right-padding
 - in DAY format 74
 - root nodes
 - described 140
 - rotated tables
 - described 181

- ROUND. *See* number functions
- rounded values
 - double-rounding 105
 - See also* ROUND function and COL-UMN command
 - sums of 104–105
 - two approaches 104
- ROWID 108
 - as a debugging tool 228, 231, 276, 279
 - finding nonunique combinations of data columns 293
 - optimizing performance 114
 - outer joins and 160
 - running total reports and 196
 - sampling the Nth row 235, 282
 - SELECT clause and 108
- ROWNUM. *See* pseudocolumns 202
- RTRIM function. *See* character functions
- rule-based optimizer 199, 253
- running total reports 194–198
 - generalizing 196–198
 - performance comparisons 270–271
 - PL/SQL approach 195–198
 - SQL approach 194–195
- running_fee function. *See* PL/SQL functions
- running_fee3 function. *See* PL/SQL functions
- runtime responses
 - retrieving other information with 52–53
 - specifying sort order 124–125
 - See also* prompting users
- S**
- sales_category function. *See* PL/SQL functions
- samples 234–242
 - selecting the Nth row 234–237
 - from sorted data 236–237
 - performance comparisons 282–283
 - PL/SQL approach 235
 - SQL approach 234
 - uses by query writers 234
 - See also* random; systematic samples
- scale. *See* NUMBER datatypes
- screen displays
 - pausing 26, 36, 314
 - settings for 21
 - See also* SET command
- screen size 314
- screen.sql. *See* SQL programs
- SELECT clause 108
 - described 107
 - including ROWID 108
- SELECT command
 - clauses described 107
 - mandatory clauses 107
 - See also* SELECT, FROM, WHERE, GROUP BY, HAVING, CONNECT BY, and ORDER BY clauses
- sequential numbering schemes
 - sorting and 128–129
- SERVEROUTPUT. *See* SET command
- SET command
 - COLSEP 55, 214
 - CONCAT 36
 - DEFINE 37
 - DOC 315
 - ECHO 62, 212, 315
 - EMBEDDED 46, 47, 68
 - ESCAPE 26, 37, 65, 66
 - FEEDBACK 47, 62, 212, 315
 - HEADING 46, 48, 206, 322
 - HEADSEP 56, 57, 58
 - LINESIZE 27, 48, 54, 212
 - ASCII files and 63
 - LONG 55, 168, 169, 329
 - LONGCHUNKSIZE 55, 168, 329
 - NEWPAGE 48, 62
 - forcing a formfeed 49, 68
 - NULL 60, 61, 208
 - NUMWIDTH 55, 58
 - PAGESIZE 27, 48, 212
 - SQL.LNO and 68
 - suppressing other system variables 62
 - PAUSE 24, 26, 315
 - RECSEP 41, 57, 58
 - RECSEPCHAR 41
 - SCAN 37

- SERVEROUTPUT 239, 243
- SQLPROMPT 22
- SQLTERMINATOR 19
- TERMOUT 12, 314
- TIME 22
- TIMING 62, 212, 315
- TRIMSPOOL 62, 63, 212
- VERIFY 50, 51, 316
- set operations 154–155
 - equivalent joins 154
 - See also* UNION, UNION ALL, INTERSECT, and MINUS
- SET TRANSACTION 201
- SET TRANSACTION command 46
- shell programs
 - examples of 12, 206
- SIGN function. *See* number functions
- significant digits
 - described 5
- simulations xvii
 - primary keys created in 254
 - procedures described 253
- single quotes
 - arguments and 30, 31, 32
 - concatenating to a string 42
 - substitution variables and 36
 - when valid part of string 42
- single-row functions
 - described 73
 - major types 73
- site profile
 - described 24, 318
- sorting order
 - ASCII character set and 122
 - EBCDIC character set and 122
 - forcing with indexes 130
 - forcing with literals 126, 295
 - hierarchical queries and 136
 - sequential numbering schemes and 128
 - specifying at runtime 124–125
 - UNION and 203, 236, 245
- spacing report columns 36
 - See also* SET command, COLSEP
- sponsor_exist function. *See* PL/SQL functions
- SPOOL command 12, 36, 315
- spooling reports
 - to a file 315
- SQL objects
 - See* SQL segments
- SQL programs
 - clears.sql 12, 21, 313
 - constrnt.sql 8, 326
 - date.sql 12, 313
 - defcnsrt.sql 326
 - defdict.sql 52, 53, 322
 - defheadr.sql 206
 - defindx.sql 324
 - defline.sql 329
 - deftabl.sql 18, 321
 - getcolumn.sql 50, 329
 - getinstc.sql 25
 - getowner.sql 18, 316
 - gettable.sql 317
 - getuser.sql 25
 - indexes.sql 6, 324
 - landcomp.sql 64
 - linesize.sql 54–56, 328
 - login.sql 318
 - owncol.sql 20, 21, 315–316
 - owners.sql 2, 12, 319
 - printer.sql 12, 21, 27, 315
 - screen.sql 12, 21, 26–27, 314
 - screen2.sql 329
 - tabldict.sql 4, 322
 - tables.sql 3, 18, 320
 - time.sql 12, 314
 - xconstrn.sql 326
 - xindexes.sql 324
 - xlinesiz.sql 328
 - xowners.sql 12, 319
 - xtabldic.sql 322
 - xtables.sql 18, 19, 320
- SQL prompt
 - adding a user name 25
 - adding the Oracle instance 25
 - changing 22–23
- SQL segments 15–17
 - described 15
 - examples of 17, 83
 - management of 83
 - report templates as 173, 175, 179
 - uses for 16

- SQL*PLUS commands
 - types of problems solved 35
 - SQL*PLUS commands. *See also* individual commands
 - SQL*PLUS environment 22–27
 - customizing with login.sql 24, 318
 - default with glogin.sql 24, 318
 - reducing interference 21
 - SQL.LNO. *See* system-maintained variables
 - SQL.PNO. *See* system-maintained variables
 - SQL_TRACE 253
 - SQL-92 154, 245, 250
 - standards, query. *See* ad hoc queries
 - START command 13, 24, 28
 - examples of 12
 - passing arguments 28
 - START WITH clause 137, 142
 - statistical analysis
 - inadequacies of SQL 233
 - stepping through joins
 - described 220
 - structured SQL 10–11
 - examples of 11
 - formatting consistently 20
 - shell programs 12
 - subqueries 148–152
 - equivalent PL/SQL functions 148
 - FROM clause 15, 200, 250, 288, 301, 302, 304
 - computing median 245
 - computing mode 250
 - ORDER BY not permitted 117, 130
 - outer joins and 159–162
 - See also* correlated subqueries
 - substitution variables
 - arguments and 29–31
 - changing ampersand designation 37
 - concatenating suffixes to 36
 - delimiters for 23, 25, 36
 - disabling default VERIFY 51
 - disabling SCAN 37
 - ORDER BY clause and 124
 - resetting 38
 - SQL fragments and 17
 - TTITLE command and 71
 - types of 38
 - SUBSTR function. *See* character functions
 - sub-totals
 - computing in reports 179
 - SUM. *See* group functions 157
 - SYS views
 - ALL_CATALOG 19, 319
 - ALL_COL_COMMENTS 323
 - ALL_CONS_COLUMNS 327
 - ALL_CONSTRAINTS 327
 - ALL_IND_COLUMNS 325
 - ALL_INDEXES 325
 - ALL_TAB_COLUMNS 55, 323, 328
 - ALL_TAB_COMMENTS 19, 42, 53, 323
 - GLOBAL_NAME 22, 23
 - NLS_SESSION_PARAMETERS 55, 329
 - SYSDATE. *See* date functions
 - system variables. *See* SET command
 - systematic samples 238–239
 - described 233, 238
 - performance comparisons 284–285
 - PL/SQL approach 238
 - SQL approach 238
 - system-maintained variables
 - SQL.LNO 68
 - SQL.PNO 68, 207
- T**
- tabldict.sql. *See* SQL programs
 - table names
 - prompting for 317
 - report of 3, 19, 320–321
 - table owners 2–3, 319
 - prompting for 316
 - report of 2
 - tables and views
 - ADDRESS 4, 5, 6, 7, 8, 9, 220, 222, 229, 254
 - CADDRESS 6, 9
 - CITYLOC 302
 - CONTACT 304
 - CONTACTS_BY_ID 305
 - FLOWER_SALES 300
 - HOLIDAYS 90
 - LOC 307
 - LOGTIME 294
 - NAME 6, 7, 108, 131, 157, 219, 254, 276, 279

- OVERTIME* 298
 - PERSON* 222, 254
 - RECENTADDRESS* 15
 - SAMPLE* 242
 - SEMINARS* 77, 80, 82, 87, 89, 100
 - SEXIDEN* 181, 185
 - STUDENT* 157, 160, 194, 195, 254, 276, 279
 - TABL1996* 299
 - TESTS* 99
 - TOTNAME* 199, 200
 - TOTSALES* 105
 - UNDERGRAD* 254
 - tables, data
 - documentation comments 320
 - tables.sql. *See* SQL programs
 - TERMOUT. *See* SET command
 - time
 - component of DATE datatypes 80
 - computing durations 294
 - time_cat. *See* PL/SQL functions
 - timestamping reports 314
 - TIMING. *See* SET command 212
 - titles, suppressing top and bottom 62
 - TKPROF 253
 - TO_CHAR. *See* conversion functions
 - TO_DATE. *See* conversion functions
 - TO_NUMBER. *See* conversion functions
 - top *N* reports 202–204
 - inappropriate use of ROWNUM 202
 - issue of ties 202, 203
 - performance comparisons 274–275
 - PL/SQL approach 204
 - SQL approach 202–204
 - top title. *See* TTITLE command
 - trailing blanks 36
 - TRIMSPOOL. *See* SET command
 - TRUNC date function. *See* date functions
 - TRUNC number function. *See* number functions
 - TTITLE command
 - bind variables and 70
 - COL 207, 322
 - custom headings and 206, 322
 - datestamping reports 314
 - examples of 12, 207
 - printer escape sequences and 64
 - substitution variables and 71
 - tuning. *See* performance
 - two objects selected from *N* objects 302–303
- U**
- UNDEFINE command 38
 - uniform random deviates 240, 242
 - See also* random samples
 - UNION ALL set operation 154, 301
 - described 154
 - UNION set operation
 - creating views with 300
 - datatype correspondence and 126, 127
 - described 154
 - eliminating duplicates 126, 301
 - equivalent joins 154
 - equivalent outer joins 160, 165, 262, 265
 - examples of 161, 165, 203, 236, 245, 247
 - forcing a sort order 203, 236, 237, 245
 - ORDER BY clause and 126–127
 - top *N* reports and 202
 - unique keys. *See* keys, unique
 - unstructured SQL
 - examples of 10
 - UPPER function. *See* character functions
 - user profile
 - described 24, 318
 - user variables
 - default datatype 51
 - defining explicitly 22
 - defining implicitly 23, 25, 38, 52, 66, 70, 324
 - defining several at once 323
 - examples of 321, 322, 324, 326
 - maximum length 51
 - prompting for 50
 - user variables. *See also* DEFINE and UNDEFINE commands
 - utility programs 313–317
 - described 313

V

- VARCHAR2 datatypes 102, 128
- VARIABLE command 70, 71, 201, 205
 - REFCURSOR 70
- views, data
 - percentage distribution reports and 199, 272
 - performance and 14
 - See also* FROM clause subqueries
 - simplifying queries with 14–15

W, X, Y, Z

- weeks
 - defined by NLS_TERRITORY 84
 - grouping by 84–85
 - user defined 84
- WHERE clause 109–115
 - balanced self-joins 195
 - coordinating with PL/SQL cursors 196, 197
 - defining report populations 109, 218
 - described 107
 - symmetry with correlated subqueries 150
- wrapping
 - blank lines and 41