

# Index

## ■ A

- Advanced Index Compression, 110
- Asynchronous global index, 149–150
- Automatic SQL tuning
  - architecture, 212
  - disabling and enabling, 216–218
  - SQL script, 216
  - verification, 213
  - viewing, 213–215
- Automatic workload repository (AWR), 211

## ■ B

- Bitmap indexes, 10, 14
  - advantages, 75
  - benefits, 64
  - BILLING\_FACT table, 69, 71
  - bitmap join indexes
    - benefits, 71
    - geography dimension (GEO\_ID), 71
    - index creation, 72
  - composite bitmap index, 59
  - data dictionary, 73–74
  - DML mechanism, 69
  - drawbacks and limitations, 58, 75
  - GENDER column, 58, 60
  - index creation, 61
  - on index-organized table, 63
  - index partitions, 70
  - join indexes
    - billing fact *vs.* geography dimension, 72
  - maintenance operations, 61
  - OLTP, 58
  - partitioned bitmap index, 62
  - performance implications, 60
  - query performance, 68
  - rule of thumb, 59
  - star schema, 57
  - star transformation, 66

- Bitmap join indexes, 11, 14
- B-tree cluster index, 11, 17
- B-tree indexes, 10, 119
  - displaying index code, 35
  - dropping index, 36
  - foreign key constraints
    - data dictionary views, 48
    - index implementation, 47
  - index creation, 33
  - inherit storage
    - parameters, 32
  - leaf nodes and branch blocks, 26
  - manageability features, 29
  - naming standards, 32
  - primary key constraint, 37
    - advantages, 39
    - ALTER TABLE, 38
    - CREATE TABLE, 38
    - drop index, 40
    - viewing index details, 40
  - reporting, indexes, 34
  - retrieve table data, 28
  - ROWID and
    - column value, 24
  - separate tablespaces, 31
  - size estimation, 30
  - table layout, 26
  - technical aspects, 24
  - unique key constraint
    - advantages, 44
    - ALTER TABLE, 42
    - CREATE TABLE, 43
    - dropping, index, 46
    - implementation, 42
    - index creation, 44

## ■ C

- Clustering factor, 168
- Coalescing indexes, 199–200

## ■ INDEX

- Composite indexes, 112
  - advantages, 114
  - choosing keys, 115
  - index scan, 116
  - index skip scan feature, 113
  - low cardinality column, 114
- Cost-based optimizer, 169
- Covering index, 7

## ■ D

- Data definition language (DDL)
  - CREATE TABLE statement, 82, 84
  - INCLUDING clause, 83
- Data manipulation language (DML), 58, 68–69
- Data warehouse environment, 153
- DBA\_AUTOTASK\_CLIENT, 217
- DBMS\_AUTO\_TASK\_ADMIN.DISABLE, 217
- DBMS\_METADATA package, 206–207, 209
- DBMS\_SQLTUNE.SCRIPT\_TUNING\_TASK function, 216
- DBMS\_SQLTUNE.SELECT\_CURSOR\_CACHE function, 224
- DBMS\_SQLTUNE.SELECT\_WORKLOAD\_REPOSITORY function, 223
- DBMS\_STATS package, 185, 187
- Descending indexes, 10, 96–97
- DML. *See* Data manipulation language (DML)
- Domain indexes, 11, 17, 122

## ■ E

- Extended data types, 50–52

## ■ F

- Fast full index scan, 176
- Foreign keys, 46
- Function-based indexes, 11, 15, 99

## ■ G

- Globally partitioned indexes, 11, 16, 132
  - architecture, 131
  - drawbacks, 140
  - hash partitioned index, 135
  - range partitioned index, 132
  - rebuilding indexes, 147
- Global partial indexes, 137

## ■ H

- Hash Cluster indexes, 11, 17
- Heap-organized tables, 78

## ■ I, J

- IMCUs. *See* In-memory column compression units (IMCUs)
- Indexed virtual column, 11, 15
- Indexes
  - accessing, index, 6–7
  - bitmap indexes, 10, 14
  - bitmap join indexes, 11, 14
  - B-tree cluster, 11, 17
  - B-tree indexes, 10–11
    - descending indexes, 13
    - index-organized table (IOT), 12
    - key compressed index, 13
    - reverse key indexes, 12
    - unique indexes, 12
  - concepts, 9
  - definition, 1
  - domain index, 11, 17
    - on foreign key columns, 18
  - function-based indexes, 11, 15
  - global partitioned index, 11, 16
  - guidelines, 20
  - Hash cluster, 11, 17
  - ignoring index, 8–9
  - indexed virtual column, 11, 15
  - invisible index, 16
  - local partitioned index, 11, 16
  - multiple column indexes, 19
  - performance with index, 4–6
  - performance without index, 2–4
  - on primary key and unique key columns, 18
  - recommendations, 18
  - ROWID, 1
  - types and features, 10
  - virtual index, 16
- Index extended columns
  - B-tree index creation, 52
  - function-based index solution, 55–56
  - virtual column solution, 52–54
- Index maintenance
  - coalescing indexes, 199–200
  - data pump, 209
  - DBMS\_METADATA package, 206–207
  - DBMS\_STATS package, 185, 187
  - dropping, 209–210
  - index creation
    - avoiding redo generation, 204
    - compressing indexes, 205
    - larger block sizes, 205
    - multiple options, 205
    - parallelism, 203–204
  - index rebuilding
    - automatic index, 199
    - clustering factor, 198

- DBA, 199
  - deleted space, index, 198
  - inefficient, 198
  - Oracle B-tree indexes, 198
  - shrink command, 199
- METHOD\_OPT parameter, 187
- moving tables and indexes, 202
- optimal index statistics, 185
- optimizer statistics, 185
- Oracle DBAs, 185
- SESSION\_TRANSFORM procedure, 207–208
- SET\_FILTER procedure, 208
- shrinking indexes, 200–201
- unusable indexes
  - database, 188–189
  - data load, 189
  - direct path load, 188
  - example, 188
  - local index, 189
  - oracle, 189
  - Oracle Database 12c, 189
  - partition index, 190
  - ROWID, 188
  - SKIP\_UNUSABLE\_INDEXES
    - parameter, 191–193
    - status, existing index, 189
- Index-organized table
  - bitmap indexes, 63
- Index-organized table (IOT), 10, 12
  - advantages, 78
  - ALTER TABLE, 88–89
  - challenges, 79
  - database information, 90
  - data definition language (DDL), 79
  - heap-organized tables, 78
  - in OLTP, 79
  - key compression, 84
  - limitations, 78
  - overflow segment, 81
  - secondary indexes, 85
  - structure, 77
- Index scans, 158
  - index fast full scan, 163
  - index full scan, 162
  - index range scan, 159
  - index skip scan, 161
  - index unique scan, 158
- INDEX\_STATS view
  - benefits, 193, 195
  - problems, 196–197
- In-memory compression units (IMCUs), 237
- In-memory (IM) column store
  - advantage, 238
  - columnar format, 236

- ENABLED/DISABLED, 240–243
- INMEMORY\_SIZE parameter, 239
- MEMCOMPRESS clause, 249
- PRIORITY clause, 248
- processing, 238
- query
  - data dictionary, 246–247
  - execution plan, view, 244–245
  - performance differences, 248
  - session-level statistics, 245–246
- RAC, 250
  - row store format, 235
- SGA, 239
- Invisible indexes, 11, 16, 93
  - database maintenance, 96
  - index creation, 94
  - optimizer, 95
  - uses, 94
  - visibility status, 95

■ **K**

- Key-compressed indexes, 10, 106
  - features, 106
  - ORDER\_MODE and ORDER\_STATUS, 108
  - storage, 109
  - uses, 106
- Key compression, 84–85

■ **L**

- Locally partitioned index, 11, 16, 126, 146
  - architecture, 127
  - partition-level requirements, 128
  - prefixed and non-prefixed options, 128
  - simplest form, 127

■ **M**

- METHOD\_OPT parameter, 187, 210
- Multi-column index, 114

■ **N**

- NOLOGGING option, 204
- Number of distinct values (NDV), 103

■ **O**

- Online transaction processing systems (OLTP), 58, 79, 91, 140
- Optimizer access path
  - full table scan, 157
  - index access path, 157

Oracle Database 12c, 189  
 Oracle Database Cartridge Developer's Guide, 123  
 Oracle query optimizer, 25

■ **P**

Partitioned indexes  
 advantages, 126, 141  
 as unusable and rebuild, 150  
 data read-only, 153  
 design considerations, 125  
 factors affecting, 125  
 globally partitioned indexes  
   architecture, 131  
   drawbacks, 140  
   hash partitioned index, 135  
   range partitioned index, 132  
   rebuilding indexes, 147  
 index size, 154  
 interval partitioning, 152  
 locally partitioned index, 126  
   advantages, 140  
   architecture, 127  
   partition-level requirements, 128  
   prefixed and non-prefixed options, 128  
   simplest form, 127  
 maintenace activities  
   adding partitions, 142  
   partition drop operation, 146  
   partition exchange, 145  
   partition merge operation, 147  
   partition move operation, 143  
   partition split operation, 144  
   truncated partition, 143  
 maintenance benefits, 126  
 non-partitioned indexes  
   rebuilding indexes, 147  
 partition names, 153  
 performance benefits, 126  
 primary key constraint and unique index, 129  
 status column, 154  
 table-level partition operations, 141  
 types of index, 154  
 Partition split operation, 144  
 Primary key constraint, 37

■ **Q**

Query processing  
 data dictionary, 246–247  
 execution plan, 244–245  
 performance differences, 248  
 session-level statistics, 245–246

■ **R**

RAC. *See* Real application application  
   clusterclusters (RAC)  
 Real application clusters (RAC), 250  
 REPORT\_AUTO\_TUNING\_TASK function, 215  
 Reverse key indexes, 10, 119  
   disadvantages, 120  
   index block contention, 121  
   index creation, 122  
 ROWID, 5  
 Table row's physical address (ROWID), 1, 23

■ **S**

SELECT\_CURSOR\_CACHE function, 224  
 SESSION\_TRANSFORM procedure, 207–208  
 SET\_FILTER procedure, 208  
 SGA. *See* System global area (SGA)  
 Shrinking indexes, 200–201  
 Specialized indexes  
   composite indexes (*see* Composite indexes)  
   descending indexes, 96  
   domain indexes, 122  
   function-based indexes, 98  
     advantages, 98  
     arithmetic expressions, 101  
     case-insensitive searches, 99  
     CASE statement, 100  
     last\_name column, 99  
     limitations, 102  
     statistics, 103  
     UPPER function, 100  
   invisible indexes (*see* “Invisible indexes”)  
   key-compressed indexes (*see* Key-compressed indexes)  
   reverse key index (*see* Reverse key indexes)  
   virtual columns, 104  
   virtual indexes, 117  
 SQL Tuning Advisor  
   adding statements, 228  
   AWR, 219  
   built-in PL/SQL packages, 212  
   DBMS\_SQLTUNE.CAPTURE\_CURSOR\_CACHE\_SQLSET procedure, 224  
   DBMS\_SQLTUNE.SELECT\_WORKLOAD\_REPOSITORY function, 218–219  
   deleting statements, 227  
   displaying, 225, 227  
   dropping, 228  
   high-resource SQL statement information, 222–223  
   memory, 220–221

Oracle query optimizer, 211  
 PROD\_WORKLOAD, 224  
 resource-ranking measures and units, 220  
 running
 

- architecture, 229
- DBMS\_SQLTUNE PL/SQL package, 230
- enterprise manager, 234
- EXECUTE\_TUNING\_TASK procedure, 232
- Snapshot IDs, 231
- SQL Developer, 233
- SQL\_ID, 231
- text, 230
- viewing and dropping, 233

 SQL statements, 211  
 SQL tuning set architecture, 218  
 STS, 212  
 non-SYS users, 220  
 tuning mode, 211  
 SQL tuning set (STS), 212  
 System global area (SGA), 239

■ **T**

Tune index usage
 

- avoid index usage
  - fast full scan, 166
  - NO\_INDEX hint, 165
  - table scan, 166
- cost-based optimizer

- change execution plans, 169
- clustering factor, 168
- demonstration purposes, 166
- INDEX\_ASC hint, 176
- INDEX\_DESC hint, 176
- INDEX hint, 175, 177
- INDEX\_JOIN hint, 177
- INDEX\_SS hint, 177
- leading portion, 174
- multiblock read count, 167
- NOT EQUAL condition, 169
- NULL values, 172
- number of distinct rows, 168
- query functions, 173
- test\_code column, 167
- TRUNC function, 174
- wildcard-based search, 171
- monitor index usage, 163
- optimizer access paths, 157
- PARALLEL\_INDEX hint, 181

■ **U**

Unique indexes, 10

■ **V, W, X, Y, Z**

Very large databases (VLDBs), 153  
 Virtual indexes, 11, 16, 117