

# SQL Commands for Lock, Fragmentation and Index Compression Analysis

Martin Frauendorfer, SAP Active Global Support

[martin.frauendorfer@sap.com](mailto:martin.frauendorfer@sap.com)

November 2010

# Agenda



1. The SQL Command Collection
2. Lock Analysis
3. Fragmentation Analysis
4. Index Compression Analysis

- 1. The SQL Command Collection**
2. Lock Analysis
3. Fragmentation Analysis
4. Index Compression Analysis

## Motivation

- SAP transactions like DBACOCKPIT and Oracle features like AWR reports provide already many monitoring possibilities.

## *Why should we need additional SQL scripts?*

- Areas and details can be covered that are too specific for standard tools
- Quick adaptations to individual needs possible
- Easy delivery, error correction and maintenance
- “Copy & Paste” execution in SQL Command Editor in transaction DBACOCKPIT possible
- Also integrated in RSORASTT / Loadable SAP Support Monitors with more comfortable user interface (SAP Note 1299493).

## Details

- **The SQL Command Collection is attached to SAP Note 1438410 and can be downloaded.**
- **You are free to use it, but SAP is not officially responsible for corrections and maintenance** (if you detect a problem and send me a mail, I will try to eliminate it)
- **No official documentation is provided for the SQL commands, but purpose and results should usually be clear based on the file name (e.g. “Space\_LargestTables.txt” → display the largest tables).**
- **Most of the SQL commands can be customized by settings in the BASIS\_INFO section rather at the top of the command. “-1” for number parameters and “%” for character parameters usually means that no restriction is done.**
- **This presentation intends to explain the possibilities of some of the SQL commands.**
- **Archive with discussed scripts: [SQL Command Script Collection](#)**

# Agenda



1. The SQL Command Collection
2. Lock Analysis
3. Fragmentation Analysis
4. Index Compression Analysis

## Lock Analysis Overview

- **There are some recurrent problems during Oracle lock analysis:**
  - What kind of lock is it – latch, mutex, row cache lock, enqueue, ...?
  - In case of enqueue: What enqueue – TX, TM, HW, ...?
  - In case of TX enqueue: What type: row lock contention, index contention, ITL contention, ...?
  - A shared or an exclusive lock?
  - What is the blocking session?
  - In case of cascaded enqueues: What is the final blocking session?
  - What is the blocking session doing while it holds the lock?
  - How long are typical lock hold times?
  - Why does a DDL statement repeatedly fail with ORA-00054?
  - Which blocks or rows are responsible for TX enqueue waits?
  - What was the reason for lock waits in the recent past?
- **All these questions can be answered using the following scripts...**

## SQL Command: Locks\_CurrentBlockingLocks.txt

### ■ Purpose:

- Determine final blocking session responsible for lock waits

### ■ Features:

- Displays current locks that block other sessions (“blocking locks”), including exact lock type and shared vs. exclusive
- Hierarchical display of lock cascades (no indent of SID → final blocking session)
- Covers many types of locks, not only enqueues
- Displays object, file, block and slot in case of exclusive lock waits, so that the exact row can be identified (e.g. in order to determine the number range object in case of NRIV lock waits)
- Generates a row extraction command in case of exclusive lock waits that can be executed to retrieve the locking record.



## SQL Command: Locks\_CurrentBlockingLocks.txt

### ■ Example:

- Intermittent bad performance
- Oracle sessions show many exclusive lock waits (“enq: TX – row lock contention”) on number range table NRIV
- Locks\_CurrentBlockingLocks.txt was run with the following results:

#### [Locks\\_CurrentBlockingLocks\\_Example](#)

- Session 7048 is the final blocking session
- Executing the command again showed a different final blocking session
- The first level waiters like sessions 8439 or 8398 wait for a lock on DBESTA\_BWPROT and not on NRIV!
- Only the waiters on the next levels (e.g. session 6104 or 8456) wait for an NRIV lock
- The main problem is DBESTA\_BWPROT – the NRIV waits are only a secondary problem (although much more obvious in the session overview)

## SQL Command: Locks\_LockedObjects.txt

### ■ Purpose:

- Helpful to understand why DDL commands fail with ORA-00054

### ■ Features:

- Displays all current object DML locks (“enq: TM – contention”)
- Both shared and exclusive locks
- Both blocking (BLOCK > 0) and non-blocking locks (BLOCK = 0)
- Includes time (in seconds) a lock was already held (CTIME\_S)

### ■ BASIS\_INFO:

- INSTANCE\_NUMBER (individual instance number, -1 for current instance, -2 for all RAC instances)
- OBJECT\_NAME (individual object name, pattern or ,%' for all objects)
- SESSION\_ID (individual locking session or -1 for all sessions)

## SQL Command: Locks\_LockedObjects.txt

### ■ Example:

- ALTER INDEX REBUILD ONLINE on index BALDAT~0 permanently failed with ORA-00054 (not directly but after some time, controlled by EVENT 10629).
- Locks\_LockedObjects.txt was executed for BALDAT and showed that many different sessions hold locks at the same time, but usually no single lock is held for longer than 10 seconds (CTIME\_S <= 10).
- As a consequence it was decided to use event 10626 instead of 10629 for rebuilding this index.
- This blocked DML operations on the table for about 10 seconds, but that was acceptable and the rebuild finished successfully.
- Details: [Locks\\_LockedObjects\\_Example](#)

## SQL Command: Locks\_LockHolderActivities.txt

### ■ Purpose:

- Analysis of lock holder activity in critical path (usually between DML operation and next COMMIT)

### ■ Features:

- Aggregates lock holder activities based on ASH samples
- If no ASH sample for lock holder exists, it is counted as INACTIVE (happens e.g. in case of ABAP processing between DML and COMMIT)
- Also counts the total and average number of waiting sessions

### ■ BASIS\_INFO:

- DBID (individual database ID, -1 for current database ID)
- INSTANCE\_NUMBER (individual instance number, -1 for current instance, -2 for all RAC instances)
- BEGIN\_DATE, END\_DATE (individual ASH aggregation time frame)
- BEGIN\_SNAP\_ID, END\_SNAP\_ID (individual ASH aggregation snapshots)
- SQL\_ID (individual waiter SQL\_ID)
- EVENT (individual waiter event)

## SQL Command: Locks\_LockHolderActivities.txt

### ■ Example:

- Performance of SELECT FOR UPDATE on TSP01 significantly impacted by “enq: TX – row lock contention” (99.97 % of runtime).
- Locks\_LockHolderActivities.txt shows that lock holder mainly executes a SELECT on table KNVH
- This SELECT is unnecessarily expensive because of an incomplete WHERE clause.
- By tuning it also the lock waits will be minimized.
- Details: [Locks\\_LockHolderActivities\\_Example](#)

## SQL Command: Locks\_TopBlocks.txt

### ■ Purpose:

- Determine records that are responsible for lock waits (can be useful input for further analysis on application side)

### ■ Features:

- Aggregates accessed blocks based on ASH samples
- Aggregation on record level not possible with Oracle 10g, with Oracle 11g the script Locks\_TopRows\_11g.txt allows aggregation on record level

### ■ BASIS\_INFO:

- DBID (individual database ID, -1 for current database ID)
- INSTANCE\_NUMBER (individual instance number, -1 for current instance, -2 for all RAC instances)
- BEGIN\_DATE, END\_DATE (individual ASH aggregation time frame)
- BEGIN\_SNAP\_ID, END\_SNAP\_ID (individual ASH aggregation snapshots)
- SQL\_ID (individual waiter SQL\_ID)
- EVENT (individual waiter event)
- NUM\_RECORDS (maximum number of result set records)

## SQL Command: Locks\_TopBlocks.txt

### ■ Example:

- SELECT FOR UPDATE on NRIV is blocked with “enq: TX – row lock contention”
- Locks\_TopBlocks.txt returns two “hot” blocks
- Only the customer specific number range object ZPWPVOR appears in these blocks with a significant number range level
- This number range object is not buffered, number range buffering would minimize the lock waits
- Details: [Locks\\_TopBlocks\\_Example](#)

# Agenda



1. The SQL Command Collection
2. Lock Analysis
3. **Fragmentation Analysis**
4. Index Compression Analysis



## Fragmentation Analysis Overview

- **Indexes and tables that contain a significant amount of unused space have various disadvantages:**
  - Increased amount of I/O
  - Increased buffer pool utilization
  - Increased replacement of other segments from buffer pool
  - Decreased performance
  - Larger database size
  - Longer backup runtime
  - Longer administration tasks like statistics creation or structure validation
- **Therefore it is useful to check for significantly fragmented segments in order to reorganize or rebuild them.**
- **The following SQL commands can be used to determine fragmentation quickly based on CBO statistics or the Oracle Segment Advisor evaluations.**

## SQL Command: `Space_SegmentFragmentation_CBOStatistics.txt`

### ■ Purpose:

- Determine unused space and storage quality based on existing CBO statistics

### ■ Features:

- Threshold values for unused space and storage quality can be specified
- Analysis can be restricted to particularly large or I/O intensive segments (focus on space or performance)
- Modification information since last statistic run can be taken into account in order to minimize inaccurate results due to outdated CBO statistics
- Restriction information can be displayed in case the CBO statistics are not be reliable for a certain reason (modified CBO statistics, LONG or LONG RAW columns, LOB columns, locked statistics, stale statistics, statistics older than segment, row length of 0, small segment, cluster table, missing statistics, bitmap index, IOT, leaf blocks = 0, IOT index, compressed index)

### ■ Prerequisites:

- Regular BRCONNECT statistic run must be scheduled (recommendation: daily)

## SQL Command: `Space_SegmentFragmentation_CBOSTatistics.txt`

- **BASIS\_INFO (subset of possible settings):**
  - SEGMENT\_TYPE (INDEX, TABLE or ALL)
  - MIN\_SIZE\_MB (minimum segment size to be considered)
  - NUM\_RECORDS (number of output records)
  - QUALITY\_THRESHOLD[\_2] (maximum storage quality to be considered)
  - UNUSED\_MB\_THRESHOLD[\_2] (minimum unused space to be considered)
  - NUM\_LARGEST\_SEGMENTS (only the specified number of largest segments are considered)
  - NUM\_IO\_READ\_SEGMENTS (only the specified number of segments that are top in I/O reads are considered)
  - SORT\_BY (sort order of the output, QUALITY → storage quality, UNUSED → unused space, SEGMENT → segment name)
  - INCLUDE\_TAB\_MODIFICATIONS ('X' to include DML operations since last statistics run for more precise calculation)
  - RESTRICTION\_INFO ('X' to display reliability restrictions)

## SQL Command: `Space_SegmentFragmentation_CBOStatistics.txt`

### ■ Output columns (subset):

- PART\_NAME: Name of partition (if segment is partitioned)
- SIZE\_POS: Position of segment in list of largest segments (e.g. "1" → largest segment in the system)
- IO\_POS: Position of segment in list of segments with highest number of disk reads according to segment statistics
- UNUSED\_MB: Amount of unused space inside the segment
- QUALITY\_%: Storage quality in percent
- SEG\_GROSS\_MB: Real size of segment (DBA\_SEGMENTS)
- SEG\_NET\_MB: SEG\_GROSS\_MB reduced by estimation of space required for administration purposes like block headers and PCTFREE
- DATA\_GROSS\_MB: Amount of allocated space based on CBO statistics (roughly  $BLOCKS * 8192$  byte for table and  $LEAF_BLOCKS * 8192$  byte for indexes)
- DATA\_NET\_MB: Amount of used space based on CBO statistics (roughly  $NUM\_ROWS * AVG\_ROW\_LEN$ ); for technical reasons the reliability restrictions are added to this column if configured

## SQL Command: Space\_SegmentFragmentation\_CBOStatistics.txt

### ■ Example:

- Analysis of both tables and indexes
- Segments are listed in the following cases:
  - High amount of unused space (QUALITY\_THRESHOLD = 60, UNUSED\_MB\_THRESHOLD = 100)
  - Bad storage quality (QUALITY\_THRESHOLD\_2 = 20, UNUSED\_MB\_THRESHOLD\_2 = 1)
- Restriction information is displayed at the end of each row (RESTRICTION\_INFO = 'X')

[Segments SegmentFragmentation CBOStatistics Example](#)

## SQL Command: `Space_SegmentFragmentation_SegmentAdvisor.txt`

### ■ Purpose:

- Determine unused space and storage quality based on the results of the Oracle Segment Advisor runs

### ■ Features:

- Threshold values for unused space and storage quality can be specified
- Analysis can be restricted to particularly large or I/O intensive segments (focus on space or performance)
- In contrast to the CBO based fragmentation check no restriction information exists, because Segment Advisor results should generally be reliable
- Very quick, because information is already available in DBA\_ADVISOR views

### ■ Prerequisites:

- Oracle Automatic Segment Advisor needs to be scheduled regularly (e.g. once a week or once a month) or manual Segment Advisor runs need to be performed

## SQL Command: `Space_SegmentFragmentation_SegmentAdvisor.txt`

### ■ **BASIS\_INFO:**

- OWNER (Pattern for segment owner)
- SEGMENT\_NAME (pattern for segment name)
- SEGMENT\_TYPE (INDEX, TABLE or ALL)
- MIN\_SIZE\_MB (minimum segment size to be considered)
- NUM\_RECORDS (number of output records)
- QUALITY\_THRESHOLD (maximum storage quality to be considered)
- UNUSED\_MB\_THRESHOLD (minimum unused space to be considered)
- NUM\_LARGEST\_SEGMENTS (only the specified number of largest segments are considered)
- NUM\_IO\_READ\_SEGMENTS (only the specified number of segments that are top in I/O reads are considered)
- SORT\_BY (sort order of the output, QUALITY → storage quality, UNUSED → unused space, SEGMENT → segment name)

## SQL Command: `Space_SegmentFragmentation_SegmentAdvisor.txt`

### ■ Output columns:

- ANALYSIS\_TIME: Timestamp of most recent Segment Advisor analysis of the segment
- SEGMENT\_NAME: Name of segment
- PARTITION\_NAME. Name of partition (if segment is partitioned)
- SIZE\_POS: Position of segment in list of largest segments (e.g. "1" → largest segment in the system)
- IO\_POS: Position of segment in list of segments with highest number of disk reads according to segment statistics
- UNUSED\_MB: Amount of unused space inside the segment
- QUALITY\_%: Storage quality in percent
- GROSS\_MB: Allocated space according to Segment Advisor
- NET\_MB: Used space according to Segment Advisor



## SQL Command: `Space_SegmentFragmentation_SegmentAdvisor.txt`

### ■ Example:

- Analysis of tables
- Tables are listed if storage quality is below 70 % (`QUALITY_THRESHOLD = 70`) and amount of unused space is at least 100 MB (`UNUSED_MB_THRESHOLD = 100`)

[Space\\_SegmentFragmentation\\_SegmentAdvisor\\_Example](#)

1. The SQL Command Collection
2. Lock Analysis
3. Fragmentation Analysis
4. **Index Compression Analysis**

## Index Compression Overview

- **Being able to compress indexes has many advantages:**
  - Less space consumption on disk
  - Less space consumption in buffer pool
  - Less I/O activities
  - Better performance due to less disk reads and buffer gets
  - Less CPU consumption (because the small compression related CPU overhead is smaller than the CPU savings due to less disk reads and buffer gets)
- **The main problem of index compression is that an optimal amount of leading index columns (prefix) has to be found that can be compressed**
- **An exact analysis (e.g. via `VALIDATE STRUCTURE` or provided PL/SQL scripts) needs to scan a lot of data and runs a long time**
- **The following SQL command can be used to determine an acceptable index compression prefix and the expected minimum space gain quickly based on CBO statistics.**

## SQL Command: `Space_IndexCompression.txt`

### ■ Purpose:

- Conservative index compression analysis based on CBO statistics

### ■ Features:

- Threshold value for minimum amount of saved data space
- Analysis can be restricted to particularly large or I/O intensive segments (focus on space or performance)
- Generation of Oracle or BRSPACE index rebuild command for compression activation possible

### ■ Restrictions:

- Result can't be 100 % precise because correlation of leading index columns is not visible in CBO statistics, script assumes worst case scenario of uncorrelated columns, in about 20 % of all indexes this can lead to a suggested prefix length that is smaller than the optimal prefix length

### ■ Prerequisites:

- Regular BRCONNECT statistic run must be scheduled (recommendation: daily)

## SQL Command: Space\_IndexCompression.txt

- **BASIS\_INFO (subset of possible settings):**
  - NUM\_RECORDS (maximum number of returned rows)
  - MIN\_SAVE\_PCT (minimum percentage of saved space; attention: The percent calculation is based on the data area only, fix components like block headers or ROWIDs are not considered)
  - NUM\_LARGEST\_INDEXES (only the specified number of largest indexes are considered)
  - NUM\_IO\_READ\_INDEXES (only the specified number of indexes that are top in I/O reads are considered)
  - SORT\_BY (SPACE → unused space, SEGMENT → segment name)
  - REBUILD\_COMMAND ('X' → Generation of Oracle rebuild command, 'B' → Generation of BRSPACE rebuild command)
  - ONLINE ('X' for ONLINE option in REBUILD command)
  - NOLOGGING ('X' for ONLINE option in REBUILD command)
  - PCTFREE (PCTFREE value in REBUILD command)
  - PARALLEL (parallel degree in REBUILD command; attention: In case of Oracle rebuild command a manual NOPARALLEL is required afterwards)

## SQL Command: `Space_IndexCompression.txt`

### ■ Output columns (subset):

- `SIZE_POS`: Position of index in list of largest indexes (e.g. “1” → largest index in the system)
- `IO_POS`: Position of index in list of indexes with highest number of disk reads according to segment statistics
- `COL`: Number of index columns
- `COMP`: Currently used compression prefix (empty if no compression is in use)
- `NOCOMP_MB`: Data size without compression
- `COMP_MB`: Data size with suggested compression
- `SAVING_MB`: Saved space with suggested compression
- `SAVING_%`: Percentage of saved data space with suggested compression
- `COMP_REC`: Compression recommendation (prefix length) or – if `REBUILD_COMMAND` is used – Oracle / `BRSPACE` rebuild command

## SQL Command: Space\_IndexCompression.txt

### ■ Example:

- Display of the 50 indexes with the highest expected space savings
- First example provides only the suggested prefix length
- In second example (taken on another system) BRSPACE rebuild commands are generated

[Space IndexCompression Example](#)

### ■ BRSPACE Reorganizations and Rebuilds

- If you want to use the script algorithm during normal BRSPACE reorganizations and rebuilds, you can specify the option “-ic sap\_proc” as of BRSPACE version 7.20 (3) (SAP Note 1464156) .

**Thank you!**



No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, Clear Enterprise, SAP BusinessObjects Explorer and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP France in the United States and in other countries.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

The information in this document is proprietary to SAP. No part of this document may be reproduced, copied, or transmitted in any form or for any purpose without the express prior written permission of SAP AG.

This document is a preliminary version and not subject to your license agreement or any other agreement with SAP. This document contains only intended strategies, developments, and functionalities of the SAP® product and is not intended to be binding upon SAP to any particular course of business, product strategy, and/or development. Please note that this document is subject to change and may be changed by SAP at any time without notice.

SAP assumes no responsibility for errors or omissions in this document. SAP does not warrant the accuracy or completeness of the information, text, graphics, links, or other items contained within this material. This document is provided without a warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials. This limitation shall not apply in cases of intent or gross negligence.

The statutory liability for personal injury and defective products is not affected. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third-party Web pages nor provide any warranty whatsoever relating to third-party Web pages.