Oracle Advanced Compression Frequently Asked Questions

July 2025, Version 23ai

Public

Purpose Statement

This document provides an overview of features and enhancements included in release 23ai. It is intended solely to help you assess the business benefits of upgrading to 23ai and planning for the implementation and upgrade of the product features described.

Disclaimer

This document in any form, software, or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement, nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described in this document remains at the sole discretion of Oracle. Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

Table of Contents

What are the Features of Advanced Compression?

Advanced Compression includes a comprehensive set of compression features designed to maximize resource utilization and reduce costs by enabling compression for structured tables (Advanced Row Compression), indexes (Advanced Index Compression), unstructured data (Advanced LOB Deduplication and Compression), database backups/exports (RMAN and Data Pump compression) and network compression (Data Guard redo log transport and Advanced Network Compression).

Advanced Compression also includes an Optimization for Flashback Time Travel (FTT) History Tables, which reduces the storage requirements for Flashback Time Travel history tables when using Flashback Time Travel to track transactional changes.

Automatic Data Optimization, included with Advanced Compression, provides declarative syntax to automatically move and compress data and indexes based on usage statistics collected by Heat Map (a feature of Oracle Database Enterprise Edition).

What are the Benefits of Advanced Row Compression?

Advanced Row Compression uses a compression algorithm specifically designed to work with OLTP (and data warehouse) applications. The algorithm works by eliminating duplicate values within a database block, even across multiple columns.

Compressed blocks contain a structure called a symbol table that maintains compression metadata. When a block is compressed, duplicate values are eliminated by first adding a single copy of the duplicate value to the symbol table. Each duplicate value is then replaced by a short reference to the appropriate entry in the symbol table.

Through this innovative design, compressed data is self-contained within the database block, as the metadata used to translate compressed data into its original state is stored in the block header. When compared with competing compression algorithms that maintain a global database symbol table, Oracle's approach typically provides query performance benefits by not introducing additional IO when accessing compressed data.

The compression ratio achieved in each environment depends on the data being compressed, specifically the cardinality of the data. In general, organizations can expect to reduce their storage space consumption by a factor of up to 2x to 4x by using Advanced Row Compression. That is, the amount of space consumed by uncompressed data will be two to four times larger than that of the compressed data.

However, the benefits of Advanced Row Compression go beyond just on-disk storage savings. One significant advantage is Oracle's ability to read compressed blocks directly without uncompressing the blocks. This helps improve query performance due to the reduction in IO, and the reduction in system calls related to the IO operations. Further, the buffer cache becomes more efficient by storing more data without having to add memory.

Why not just use Storage-Based Compression and Deduplication?

An alternative to initiating compression, at the database-level, is to use deduplication and compression at the storage-level to address capacity utilization concerns. With a storage-based solution, duplicate blocks are eliminated from incoming data streams, then many storage systems can analyze what remaining data can be compressed as a secondary function. The combination of deduplication and compression can be used to minimize the amount of data stored on the storage system.

With deduplication and compression enabled at the storage-level, the two operations are executed with no knowledge of the use or value of incoming data. Without true application awareness, the deduplication and compression functions are not applied strategically to the incoming data but are reducing data by any means necessary to minimize use of limited storage resources.

Without awareness of the data value and use, duplicate data is deduplicated, and any compressible data is compressed, without regard for rehydration or decompression impacts on data retrieval. If there is duplicate, or compressible data, that should remain in duplicate or uncompressed form for speed of access, the deduplication and compression functions must be turned off. This is also true for compression-only appliances that may be used to minimize bandwidth usage between remote sites.

Because an effective storage deduplication/compression strategy requires fully hydrated and uncompressed data to be sent to the storage device, the data either would have to be not compressed or would have to be uncompressed before going to storage. If the database is not compressed, then all the advantages of compression (reduced storage, reduced IO, reduced memory use in the buffer cache etc.) will not be available on the database. If data and indexes must be uncompressed before storage, then additional CPU and IO overhead will be required to perform the decompression.

An additional consideration when evaluating storage-based solutions is to consider the impact on network traffic and available bandwidth. Where Advanced Compression is driven by the database and can reduce the amount of network traffic between server and storage, an effective storage deduplication or compression strategy requires fully hydrated and uncompressed data to be sent to the storage device.

Why use Database Compression if Disk is Cheap?

Enterprises are experiencing an explosion in the volume of data required to effectively run their businesses. This trend in data growth can be attributed to several key factors. Recent changes in the regulatory landscape, such as Sarbanes-Oxley and HIPAA, are contributing to this trend by mandating that enterprises retain large amounts of information for long periods.

Mass distribution of rich and multimedia content over the Internet, made possible through advancements in broadband technologies, also contributes to the growth in overall data volume. Various estimates indicate that data volume is almost doubling every 2-3 years.

This growth in data volume presents a daunting management challenge for organizations. First and foremost are the spiraling storage costs. Even though the cost of storage has been declining dramatically in the last few years, the enormous growth in the volume of data that needs to be retained online makes storage one of the biggest cost elements of most IT budgets. In addition, application scalability and query performance must continue to meet the demands of the business – even as data volumes grow.

Advanced Compression helps organizations cope with these challenges. Innovations in Oracle compression technologies help organizations reduce the resources and costs of managing large data volumes. Another important benefit is in query performance. A major bottleneck for many systems is IO bandwidth. Advanced Compression can help alleviate that bottleneck in several cases by reducing the amount of data that needs to be transferred across IO channel and further boost performance through improved memory efficiencies.

Who uses Advanced Compression?

Financial, government, education, healthcare, utilities, insurance, retail, banking, manufacturing, transportation, military and more use Oracle Compression around the globe.

How do I Enable Compression, Online, for Existing Tables?

These are some of the recommended approaches to enabling Advanced Row Compression, on existing tables:

Online Redefinition (DBMS_REDEFINITION)

This approach will enable Advanced Row Compression for future DML and compress existing data. Using DBMS_REDEFINITION keeps the table online for both read/write activity during the migration. Run in parallel for best performance.

For more detailed information, please see the Oracle Database documentation.

ALTER TABLE … MOVE ONLINE

This approach will enable Advanced Row Compression for future DML and will compress existing data. ALTER TABLE ... MOVE ONLINE allows DML operations to continue to run uninterrupted on the table being moved. Run in parallel for best performance.

The use of “ONLINE” requires Advanced Compression.

For more detailed information, please see the Oracle Database documentation.

What Compression Ratio can I Expect for my Tables, Indexes and LOBS?

The compression ratio achieved in each environment depends on the nature of the data being compressed; specifically, the cardinality of the data.

In general, organizations can typically reduce their storage space consumption by a factor of up to 2x to 4x when using Advanced Row Compression. A 2x compression ratio represents approximately up to a 50% reduction in the storage footprint.

Can Partitioned Tables be Compressed?

Yes. Compression can be enabled at a tablespace, table, or partition level.

Can Compression be Set as the Default for a Tablespace?

Yes. Both Advanced Row and Advanced Index compression can be enabled as the default when any new table is created, or a table is rebuilt.

Do All Tables Need to be Compressed?

No – You can pick-and-choose which tables you want to compress, and you do not have to compress all tables/partitions.

Should Compression be Enabled by Default at the Tablespace-Level?

For custom applications, we recommend compressing at the tablespace-level, but users should consider turning off compression on very high traffic tables, such as tables used as queues.

For commercial packaged applications, where typically the number of objects can be very large, the recommended approach is object selection instead of exclusion. Often the largest tables and indexes consume most of the database space. Compressing those objects, while excluding high traffic objects like tables used as queues, will give most of the compression benefits. Other objects can be compressed over time as needed.

Can the Compression Ratio be Estimated Before Enabling Compression on an Existing Table?

Yes. Compression Advisor (DBMS_COMPRESSION) is a free PL/SQL package used to estimate potential compression ratios for tables, indexes, and LOBs based on analysis of a sample of data. It provides a good estimate of the actual compression ratio that may be obtained after implementing Advanced Compression.

Compression Advisor is included with Oracle Database Enterprise Edition 11g Release 2 and above.

Can SecureFiles LOB Segments be Compressed?

Yes. The Advanced LOB Compression and Advanced LOB Deduplication features of Advanced Compression are intended to reduce the amount of storage required for SecureFiles LOB segments.

Can Index-Organized Tables (IOTs) be Compressed?

Yes. Index-Organized Tables (IOT's) are essentially indexes, so they cannot be compressed with Advanced Row Compression, but can be compressed using Advanced Index Compression (LOW) or Prefix Compression.

Should all Tables be Compressed?

Yes. The general recommendation is to compress all the user tables (and indexes) in the database with one exception: if the table is used as a queue, i.e., if rows are inserted into the table, then later most, or all, the rows are deleted and more rows are inserted then deleted, then you should not compress the table.

Should Specially Created Test Data and Test Applications be Used to Test Advanced Compression?

No. The best test environment for testing each Advanced Compression feature is where you can most closely duplicate the production environment (using your actual data and applications) – this will provide the most realistic (pre- and post- compression) performance and functionality comparisons.

What is the CPU Overhead Associated with Advanced Compression?

Although CPU overhead is minimal (5% or more typically), implementing Advanced Compression is ideal on systems with available CPU cycles, as compression will have additional, although minor, overhead for some DML operations.

Testing with your own data, and applications, is the best way to determine the actual CPU overhead associated with compression.

What data Types are not Supported by Advanced Row Compression?

Advanced Row Compression is NOT supported for use with tables that have LONG data types.

Will a Larger Block Size Improve the Compression Ratio?

It is possible a larger block size will have a better compression ratio if the larger block has more duplicate data on the block. However, larger blocks do not always ensure higher compression ratios.

Test with your own data to determine if larger block sizes will have an impact on your compression ratio.

Doesn't Basic Compression have the Same Benefits as Advanced Row Compression?

No. Basic Table Compression only compresses data that was loaded using bulk load operations.

Advanced Row Compression, a feature of Advanced Compression, allows data to be compressed during bulk loads, as well as during all types of data manipulation operations, including conventional DML such as INSERT and UPDATE.

Advanced Row Compression is well suitable for both Data Warehouse and transactional/OLTP environments. Basic Table Compression is best suited for applications with no, or very little data modification.

How is Advanced Compression RMAN Backup Compression Different from RMAN Basic Compression?

Basic Compression is included with RMAN.

Advanced Compression provides three levels of RMAN Compression: LOW, MEDIUM, and HIGH. The amount of storage savings increases from LOW to HIGH, while potentially consuming more CPU resources.

- **LOW** -- Fastest compression and best suited when backup is CPU constrained
- **MEDIUM** -- Balance between CPU usage and compression ratio
- **HIGH** -- Best compression ratio and highest CPU utilization. Best suited when backup is constrained by network or IO

What are Advanced LOB Compression and Advanced LOB Deduplication?

SecureFiles, a feature in Oracle Database, offers a ‘best-of-both-worlds’ architecture for storing unstructured content, such as documents, spreadsheets, and XML files. SecureFiles is specifically engineered to deliver high performance for file data comparable to that of traditional file systems while retaining the advantages of the Oracle Database. Advanced Compression has two storage optimization features that can be leveraged with SecureFiles.

The first feature, Advanced LOB Deduplication eliminates duplicate copies of SecureFiles LOB data.

The Second Feature, Advanced LOB Compression, utilizes industry standard compression algorithms to further minimize the storage requirements of SecureFiles LOB data.

There are three levels of Advanced LOB compression available: LOW, MEDIUM, and HIGH.

- **LOW** -- Optimized for high performance and maintains up to 80% of the compression achieved through MEDIUM, while utilizing 3x less CPU
- **MEDIUM** – The default which typically provides good compression with a modest CPU overhead of up to 3%-5%
- **HIGH** -- The highest storage savings but incurs the most CPU overhead

How does Advanced Row Compression Work?

Advanced Row Compression uses a unique compression algorithm specifically designed to work with OLTP/DW applications. The algorithm works by eliminating duplicate values within a database block, even across multiple columns.

Compressed blocks contain a structure called a symbol table that maintains compression metadata. When a block is compressed, duplicate values are eliminated by first adding a single copy of the duplicate value to the symbol table. Each duplicate value is then replaced by a short reference to the appropriate entry in the symbol table. Through this innovative design, compressed data is self-contained within the database block as the metadata used to translate compressed data into its original state is stored in the block.

When compared with competing compression algorithms that maintain a global database symbol table, Oracle’s unique approach provides query performance benefits by not introducing additional IO when accessing compressed data.

What Optimizations has Oracle Done to Reduce Compression Overhead?

Advanced Row Compression typically has no adverse impact on read operations. There is additional work performed while writing data, making it impossible to eliminate performance overhead for write operations. However, Oracle has put in a significant amount of work to minimize this overhead for Advanced Row Compression where possible.

Oracle compresses blocks in batch mode rather than compressing data every time a write operation takes place. A newly initialized block remains uncompressed until data in the block reaches an internally controlled threshold. When a transaction causes the data in the block to reach this threshold, all contents of the block are compressed.

Subsequently, as more data is added to the block and the threshold is again reached, the entire block is recompressed to achieve the highest level of compression. This process repeats until Oracle determines that the block can no longer benefit from further compression. Only transactions that trigger the compression of the block will experience the slight compression overhead.

Does Data, and Indexes, Stay Compressed in Memory?

Yes. Oracle Database can read compressed blocks directly without having to first uncompress the block. Therefore, there is typically no significant performance degradation for accessing compressed data.

In fact, in many cases query performance improves due to the reduction in IO since Oracle will have to access fewer blocks. Further, the buffer cache will become more efficient by storing more data without having to add memory.

Do chained rows get compressed?

Yes. Before Oracle Database 12*c*, blocks containing many types of chained rows could not be compressed. This limitation was removed starting with Oracle Database 12*c* Release 2.

Does Advanced Compression include Information Lifecycle Management (ILM) capabilities?

Yes. Information Lifecycle Management is the practice of applying policies for the effective management of information throughout its useful life. This includes every phase of a “row” from its beginning to its end, and consists of the policies, processes, practices, and tools used to align the business value of information with the most appropriate and cost-effective IT infrastructure from the time information is created through its final disposition.

The Advanced Compression feature, Automatic Data Optimization (ADO), is used to create Information Lifecycle Management policies to automate compression tiering, and storage tiering, as part of an ILM solution.

For more detailed information, please see the Oracle Database documentation.

Connect with us

Call +**1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

blogs.oracle.com facebook.com/oracle twitter.com/oracle