

Managing Storage Space

- ❑ Space is automatically managed by the Oracle Database server. It generates alerts about potential problems and recommends possible solutions.
- ❑ **Oracle Managed Files (OMF)**
 - ❖ No need to bother about the file names and storage requirements. Oracle provides a unique file name.
 - ❖ Eliminates the need for the DBA to directly manage OS files.
 - ❖ Allows operations to be specified in terms of objects and not files.
 - ❖ Reduces the chance of overwriting a file. Avoids accidental corruption of data files.

- ❑ **Free-space management with bitmaps (“locally managed”)**

note: This is known as Automatic Segment Space Management

For example, the following statement creates tablespace mytbs1 with automatic segment-space management:

```
CREATE TABLESPACE mytbs1
DATAFILE '/u01/oracle/data/mytbs01.dbf' SIZE 500M
EXTENT MANAGEMENT LOCAL
SEGMENT SPACE MANAGEMENT AUTO;
```

- ❑ **Automatic data file extension**

So the files can grow automatically based on the amount of data in the files.

- ❑ **Proactive space management (default thresholds and server-generated alerts)**

Managing Storage Space

Availability and Optimization of Free Space in a Data Block

Two types of statements can increase the free space of one or more data blocks:

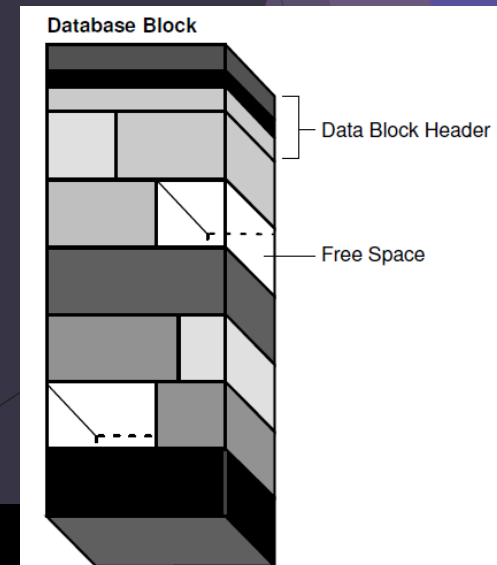
- 1- DELETE statements
- 2- UPDATE statements that update existing values to smaller values

The released space from these types of statements is available for subsequent INSERT statements under the following conditions:

- ❑ If the INSERT statement is in the same transaction and subsequent to the statement that frees space, then the INSERT statement can use the space made available.
- ❑ If the INSERT statement is in a separate transaction from the statement that frees space (perhaps being run by another user), then the INSERT statement can use the space made available only after the other transaction commits and only if the space is needed.

Oracle Database automatically and transparently coalesces the free space of a data block *only* when the following conditions are true:

- An INSERT or UPDATE statement attempts to use a block that contains sufficient free space to contain a new row piece.
- The free space is fragmented so that the row piece cannot be inserted in a contiguous section of the block.



Managing Storage Space

Row Chaining and Migrating

In two circumstances, the data for a row in a table may be too large to fit into a single data block:

In the first case, the row is too large to fit into one data block when it is first inserted. In this case, Oracle stores the data for the row in a chain of data blocks (one or more) reserved for that segment. Row chaining most often occurs with large rows, such as rows that contain a column of datatype LONG or LONG RAW. Row chaining in these cases is unavoidable.

the second case, a row that originally fit into one data block is updated so that the overall row length increases, and the block's free space is already completely filled. In this case, Oracle **migrates** the data for the entire row to a new data block, assuming the entire row can fit in a new block. Oracle preserves the original row piece of a migrated row to point to the new block containing the migrated row. The rowid of a migrated row does not change.

Note: When a row is chained or migrated, I/O performance associated with this row decreases because Oracle must scan more than one data block to retrieve the information for the row.

Managing Storage Space

Types of Segments

❑ A segment is a set of extents allocated for a certain logical structure.

❑ The different types of segments include:

✓ **Table and cluster**

Note: For a partitioned table, each partition has a data segment

✓ **Index**

Note: For a partitioned index, each partition has an index segment.

✓ **Undo**

Oracle Database maintains information to reverse changes made to the database.

This information consists of records of the actions of transactions, collectively known as undo.

Undo is stored in undo segments in an undo tablespace

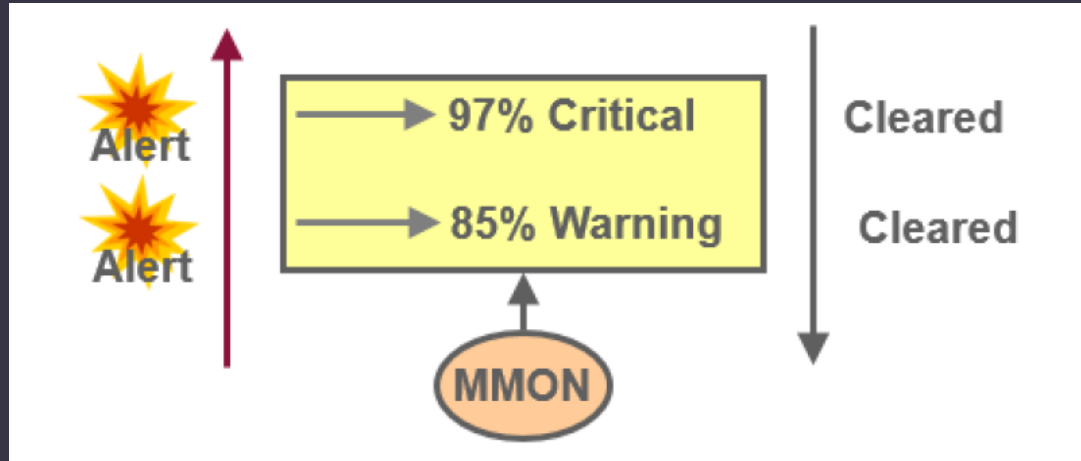
✓ **Temporary**

A temporary segment is created by the Oracle Database server when a SQL statement needs a temporary database area to complete execution.

When the statement finishes execution, the extents in the temporary segment are returned to the system for future use

Managing Storage Space

Monitoring Tablespace Space Usage



The database server tracks space utilization while performing regular space management activities. This information is aggregated by the **MMON** process. An alert is triggered when the threshold for a tablespace has been reached or cleared.

The **DBMS_SERVER_ALERT** package enables you to configure the Oracle Database server to issue an alert when a threshold for a specified server metric has been violated.

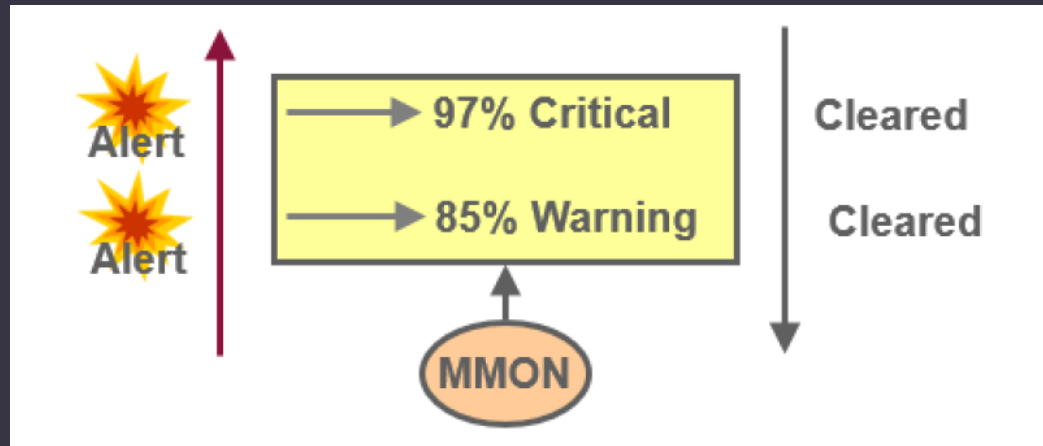
You can configure both warning and critical thresholds for a large number of predefined metrics.

in this course we will focus on **tablespace_pct_full** metric

if a warning threshold is reached, the server generates a severity level 5 alert. If a critical threshold is reached, the server generates a severity level 1 alert.

Managing Storage Space

Monitoring Tablespace Space Usage



- ❑ **Read-only and offline tablespaces:** Do not set up alerts.
- ❑ **Temporary tablespace:** Threshold corresponds to space currently used by sessions.
- ❑ **Undo tablespace:** Threshold corresponds to space used by active and unexpired extents.
- ❑ **Auto-extensible files:** Threshold is based on the maximum file size

Managing Storage Space

Monitoring Tablespace Space Usage

DBMS_SERVER_ALERT.SET_THRESHOLD

example

We have tablespace called **TBSALERT**

We need to set warning at 55%

We need to set critical at 70%

```
begin
DBMS_SERVER_ALERT.SET_THRESHOLD(
metrics_id => dbms_server_alert.tablespace_pct_full,
warning_operator => DBMS_SERVER_ALERT.OPERATOR_GE,
warning_value => '55',
critical_operator => DBMS_SERVER_ALERT.OPERATOR_GE,
critical_value => '70',
observation_period => 1,
consecutive_occurrences => 1,
instance_name => 'orcl',
object_type => DBMS_SERVER_ALERT.OBJECT_TYPE_TABLESPACE,
object_name => 'TBSALERT');
end;
```

OPERATOR_GE

A metric value greater than or equal to the threshold value is considered a violation.

https://docs.oracle.com/database/121/ARPLS/d_server_alert.htm#ARPLS68004

Managing Storage Space

Resolving Space Usage Issues



- Resolve space usage issues by:
 - Adding or resizing datafiles
 - Setting `AUTOEXTEND` to `ON`
 - Shrinking objects
 - Reducing `UNDO_RETENTION`
- Check for long-running queries in temporary tablespaces

Managing Storage Space

Reclaiming Wasted Space

Over time, updates and deletes on objects within a tablespace can create pockets of empty space that individually are not large enough to be reused for new data.

This type of empty space is referred to as **fragmented free space**.

Objects with fragmented free space can result in much wasted space, and can impact database performance

The preferred way to defragment and reclaim this space is to perform an **online segment shrink**

Because a shrink operation may cause ROWIDs to change in heap-organized segments, you must enable row movement on the corresponding segment before executing a shrink operation on that segment. Row movement by default is disabled at segment level.

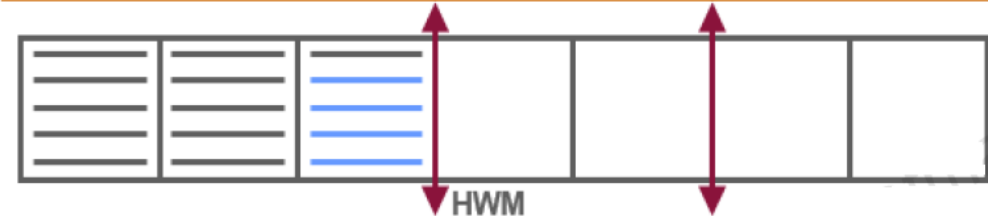


1 `ALTER TABLE employees SHRINK SPACE COMPACT;`



DML operations and queries can be issued during compaction.

2 `ALTER TABLE employees SHRINK SPACE;`



DML operations are blocked when the HWM is adjusted

Managing Storage Space

Segment Advisor

- We use the **Segment Advisor** to identify segments that would benefit from online segment shrink.
- Only segments in locally managed tablespaces with automatic segment space management (ASSM) are eligible.
- **Segment Advisor** can be automatic or manual.
- Running the Segment Advisor Manually with PL/SQL using package **DBMS_ADVISOR**

DBMS_ADVISOR package procedures relevant to the Segment Advisor:

- ☐ CREATE_TASK
- ☐ CREATE_OBJECT
- ☐ SET_TASK_PARAMETER
- ☐ EXECUTE_TASK

For more info

https://docs.oracle.com/database/121/ARPLS/d_advis.htm#ARPLS350