Managing Storage Space

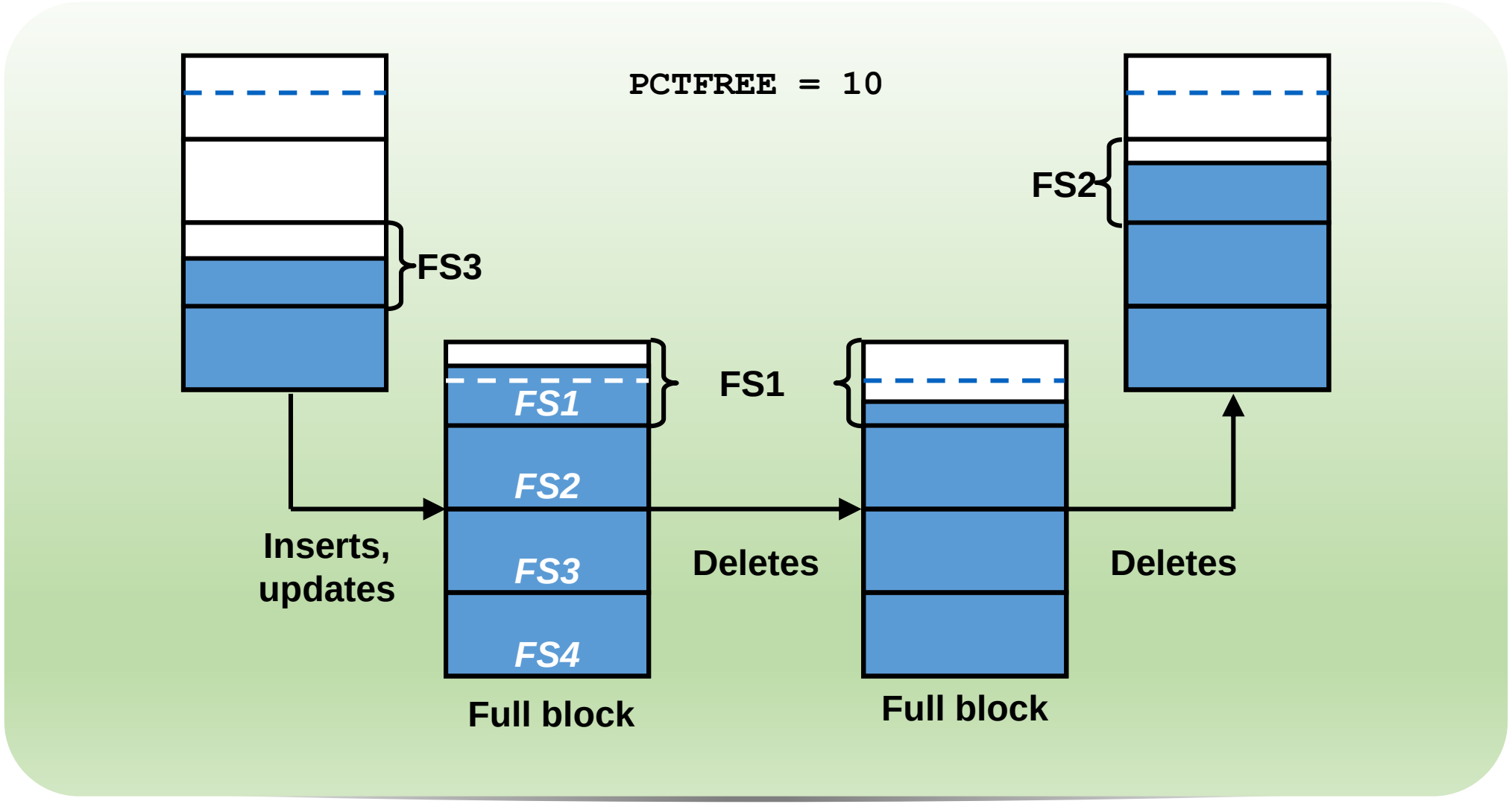
Objectives

- After completing this lesson, you should be able to:
  - Describe how the Oracle Database server automatically manages space
  - Save space by using compression
  - Proactively monitor and manage tablespace space usage
  - Describe segment creation in the Oracle database
  - Control deferred segment creation
  - Reclaim wasted space from tables and indexes by using the segment shrink functionality
  - Manage resumable space allocation

Space Management Features

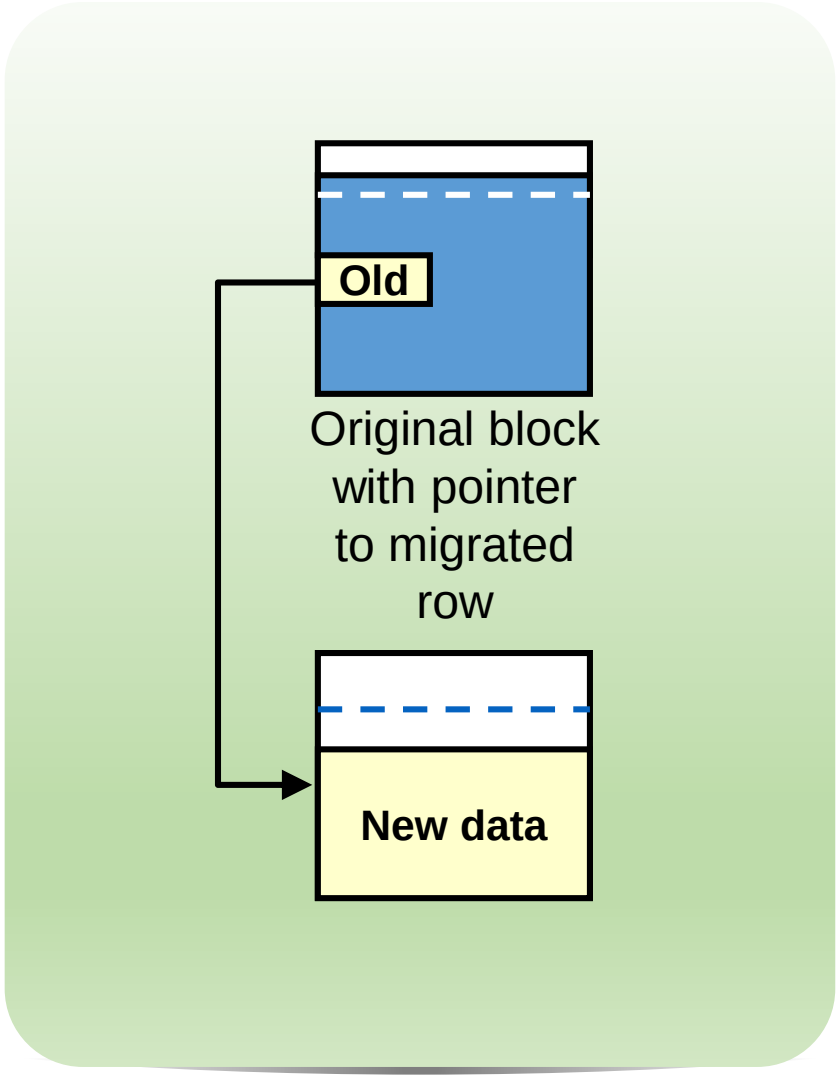
- Oracle Managed Files (OMF)
- Free-space management with bitmaps (“locally managed”) and automatic data file extension
- Proactive space management (default thresholds and server-generated alerts)
- Space reclamation (shrinking segments, online table redefinition)
- Capacity planning (growth reports)

Block Space Management

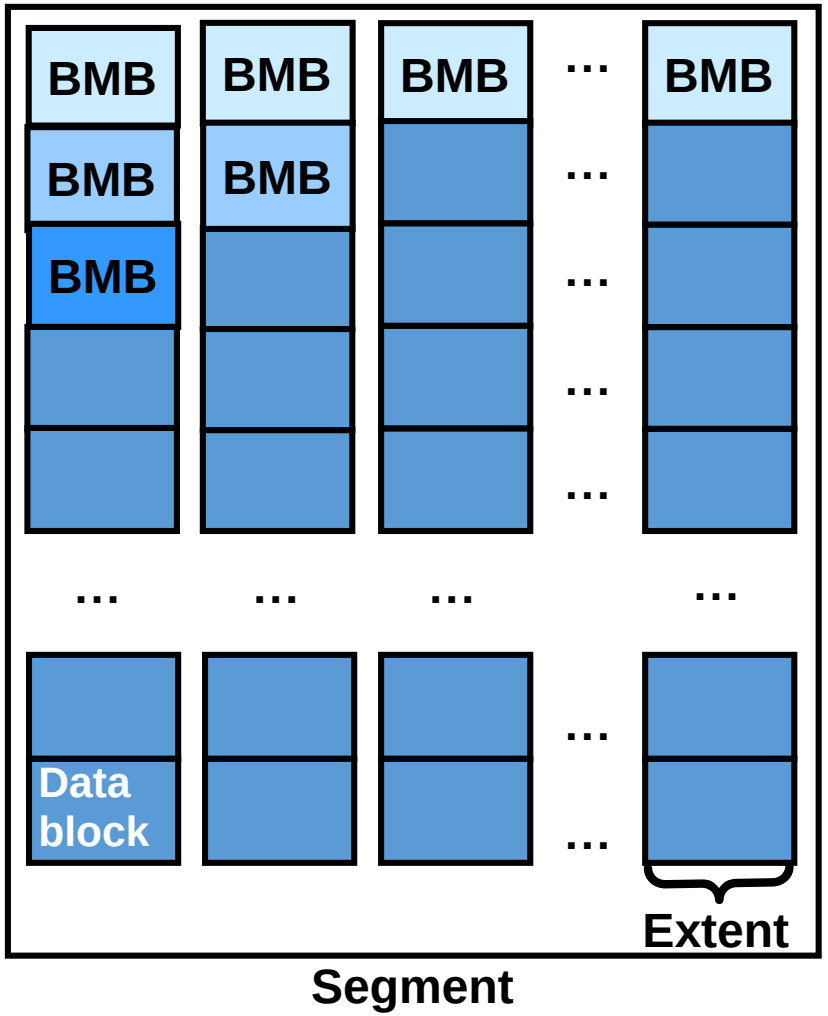


Row Chaining and Migration

- On update: Row length increases, exceeding the available free space in the block.
- Data needs to be stored in a new block.
- Original physical identifier of row (`ROWID`) is preserved.
- The Oracle Database server needs to read two blocks to retrieve data.
- Segment Advisor finds segments containing the migrated rows.
- There is automatic coalescing of fragmented free space inside the block.



Free Space Management Within Segments



- Tracked by bitmaps in segments
- Benefits:
  - More flexible space utilization
  - Runtime adjustment
  - Multiple process search of bitmap blocks (BMBs)

Types of Segments

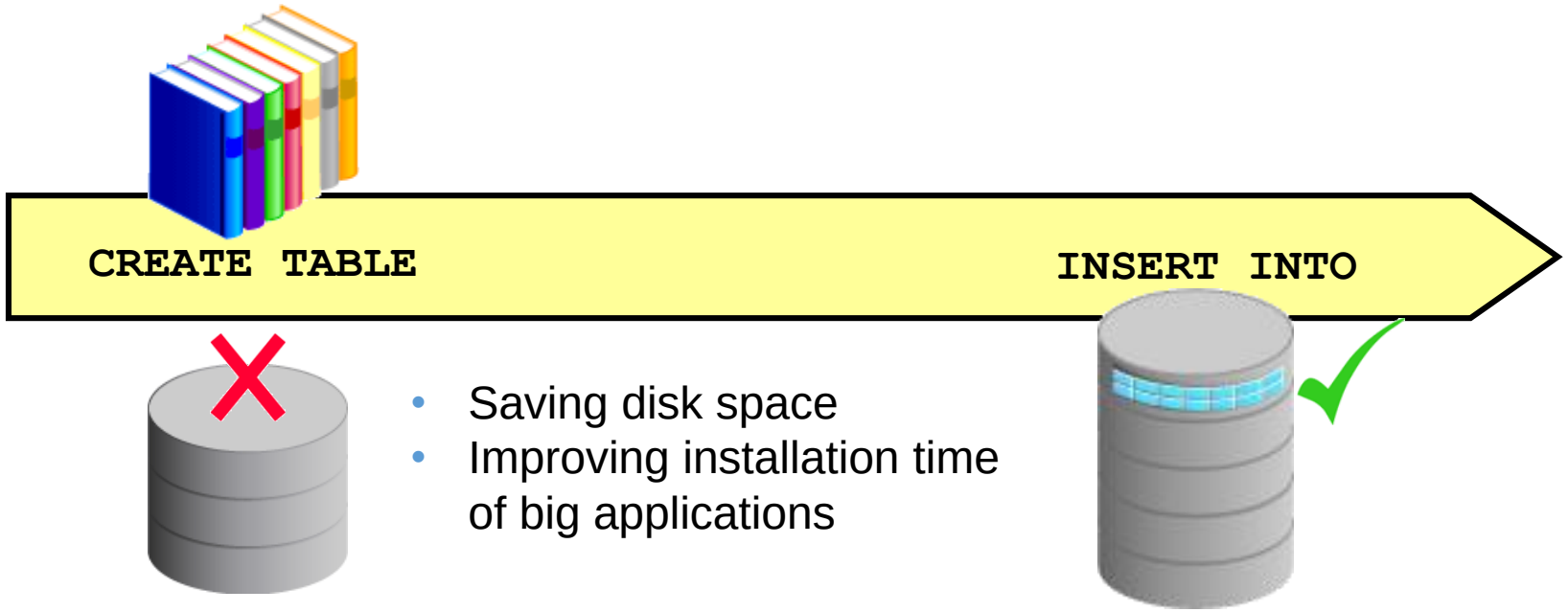
- A segment is a set of extents allocated for a certain logical structure.
- The different types of segments include:
  - Table and cluster
  - Index
  - Undo
  - Temporary
- Segments are dynamically allocated by the Oracle Database server.

Allocating Extents

- Searching the data file's bitmap for the required number of adjacent free blocks
- Sizing extents with storage clauses:
  - `UNIFORM`
  - `AUTOALLOCATE`
- Viewing the extent map
- Obtaining deallocation advice

Understanding Deferred Segment Creation

- `DEFERRED_SEGMENT_CREATION = TRUE` is the default.
- Deferred segment is the default for tables, indexes, and partitions.
- Segment creation takes place as follows:
  - Table creation > Data dictionary operation
  - DML > Segment creation



Controlling Deferred Segment Creation

- With the DEFERRED_SEGMENT_CREATION parameter:
  - Initialization parameter file
  - ALTER SESSION command
  - ALTER SYSTEM command
- With the SEGMENT CREATION clause:
  - IMMEDIATE
  - DEFERRED (default)

```
CREATE TABLE SEG_TAB3(C1 number, C2 number)
   SEGMENT CREATION IMMEDIATE TABLESPACE SEG_TBS;
CREATE TABLE SEG_TAB4(C1 number, C2 number)
   SEGMENT CREATION DEFERRED;
```

Restrictions and Exceptions

- Segment creation on demand is not for the following:
  - IOTs, clustered tables, or other special tables
  - Tables in dictionary-managed tablespaces

Space-Saving Features

- No segments for unusable indexes and index partitions
- Creating an index without a segment:

```
CREATE INDEX test_i1 ON seg_test(c) UNUSABLE
```

- Removing any allocated space for an index:

```
ALTER INDEX test_i UNUSABLE
```

- Creating the segment for an index:

```
ALTER INDEX test_i REBUILD
```

Private Temporary Tables

```
USER_PRIVATE_TEMP_TABLES
```

- Private Temporary Tables (PTTs) exist only for the session that creates them.
  - You can create a PTT with the `CREATE PRIVATE TEMPORARY TABLE` statement.
  - The table name must start with `ORA$PTT_` :

```
PRIVATE_TEMP_TABLE_PREFIX = ORA$PTT_
```

```
SQL> CREATE PRIVATE TEMPORARY TABLE ORA$PTT_mine (c1 DATE, … c3 NUMBER(10,2));
```

  - The `CREATE PRIVATE TEMPORARY TABLE` statement does not commit a transaction.
  - Two concurrent sessions may have a PTT with the same name but different shape.

  - PTT definition and contents are automatically dropped at the end of a session or transaction.

```
SQL> CREATE PRIVATE TEMPORARY TABLE ORA$PTT_mine (c1 DATE …)
            ON COMMIT PRESERVE DEFINITION;
```

```
SQL> DROP TABLE ORA$PTT_mine;
```

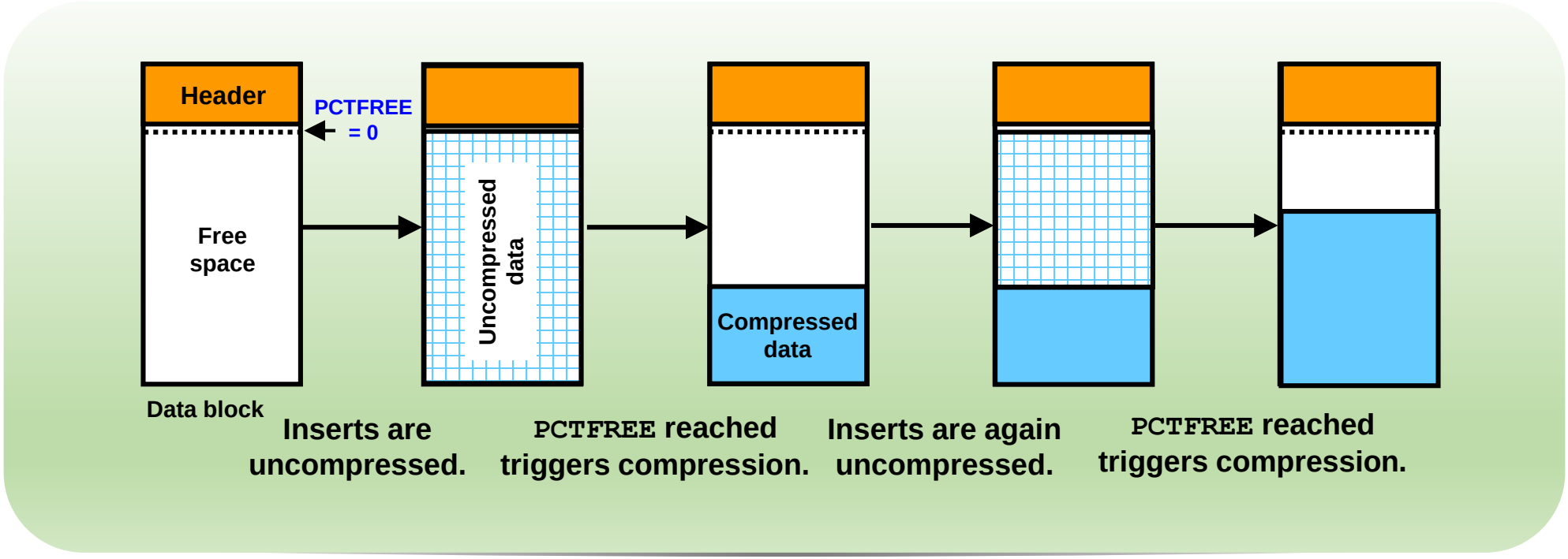
Table Compression: Overview

- Reducing storage costs by compressing all data:
  - Basic compression for direct-path insert operations: 10x
  - Advanced row compression for all DML operations: 2–4x

Compression Method	Compression Ratio	CPU Overhead	`CREATE` and `ALTER TABLE` Syntax	Typical Applications
Basic table compression	High	Minimal	`COMPRESS [BASIC]`	DSS
Advanced row compression	High	Minimal	`ROW STORE COMPRESS ADVANCED`	OLTP, DSS

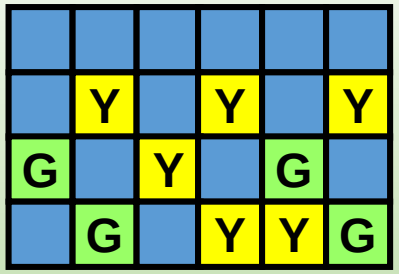
Compression for Direct-Path Insert Operations

- Is enabled with `CREATE TABLE ... COMPRESS BASIC`
- Is recommended for bulk loading data warehouses
- Maximizes contiguous free space in blocks

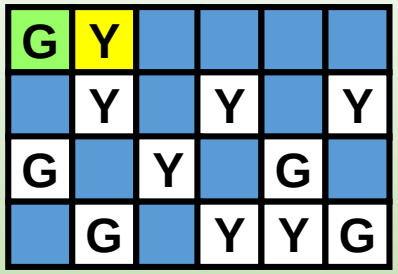


Advanced Row Compression for DML Operations

- Is enabled with `CREATE TABLE … ROW STORE COMPRESS ADVANCED`
- Is recommended for active OLTP environments



Uncompressed block



OLTP compression with the symbol table at the beginning of the block

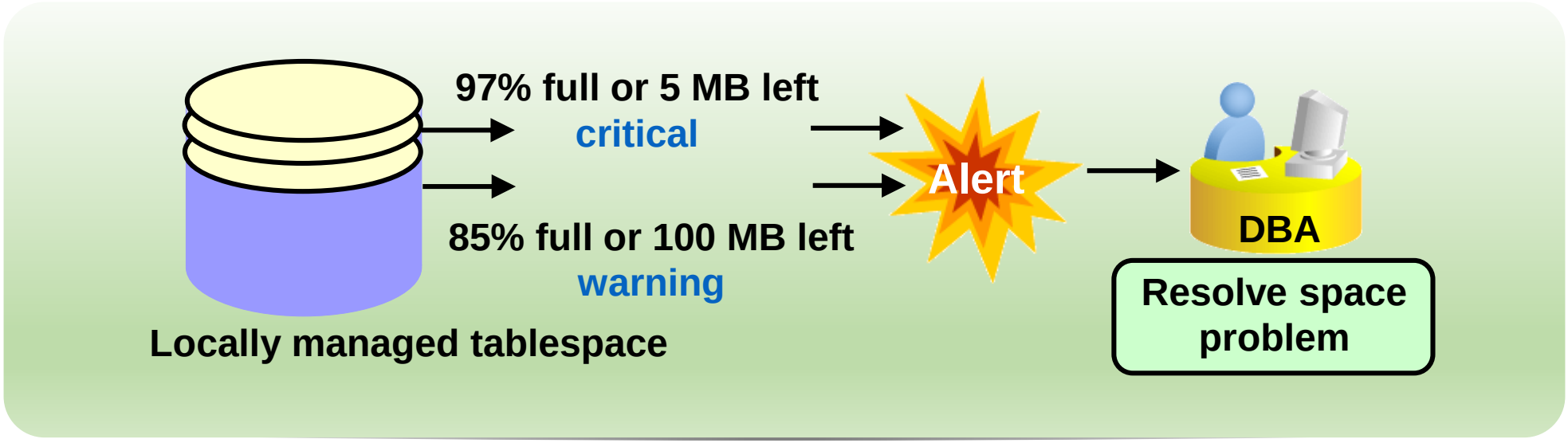
Specifying Table Compression

- You can specify table compression for:
  - An entire heap-organized table
  - A partitioned table (each partition can have a different type or level of compression)
  - The storage of a nested table
- You cannot:
  - Specify basic and advanced row compression on tables with more than 255 columns
  - Drop a column if a table is compressed for direct loads, but you can drop it if the table is advance row compressed

Using Compression Advisor

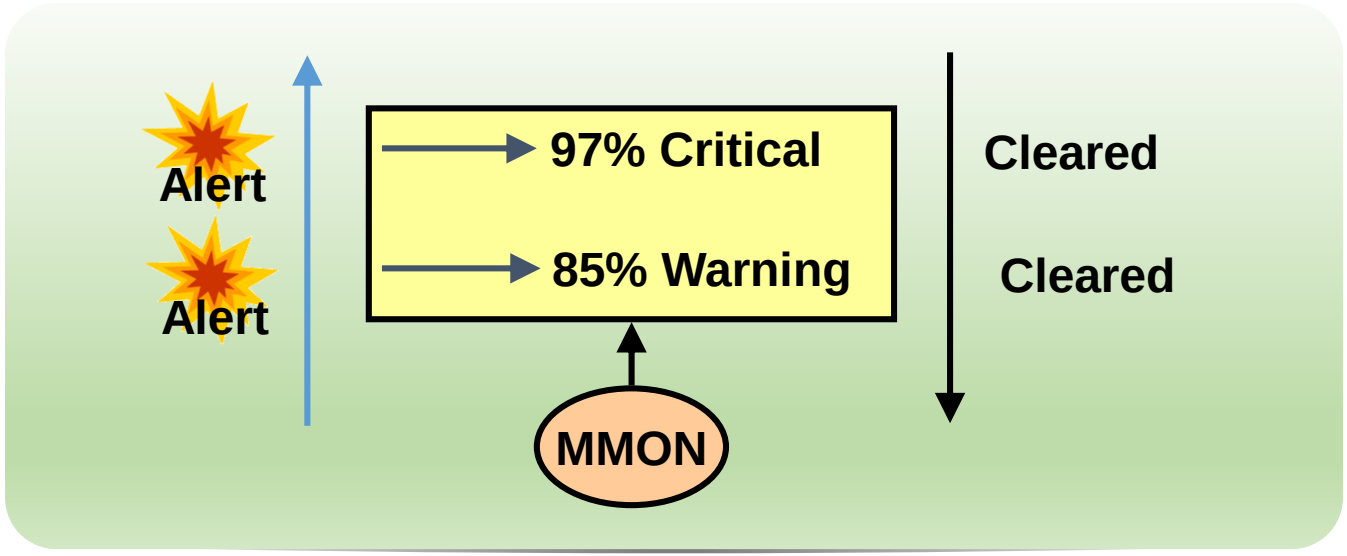
- Analyzes objects to give an estimate of space savings for different compression methods
- Helps in deciding the correct compression level for an application
- Recommends various strategies for compression
  - Picks the right compression algorithm for a particular data set
  - Sorts on a particular column for increasing the compression ratio
  - Presents tradeoffs between different compression algorithms

Resolving Space Usage Issues



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

Monitoring Tablespace Space Usage

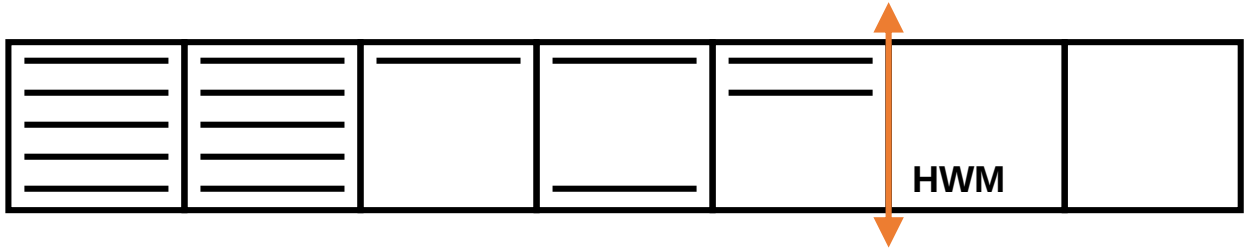


- Read-only and offline tablespaces: Do not set up alerts.
- Temporary tablespace: Threshold corresponds to space currently used by sessions.
- Undo tablespace: Threshold corresponds to space used by active and unexpired extents.
- Auto-extensible files: Threshold is based on the maximum file size.

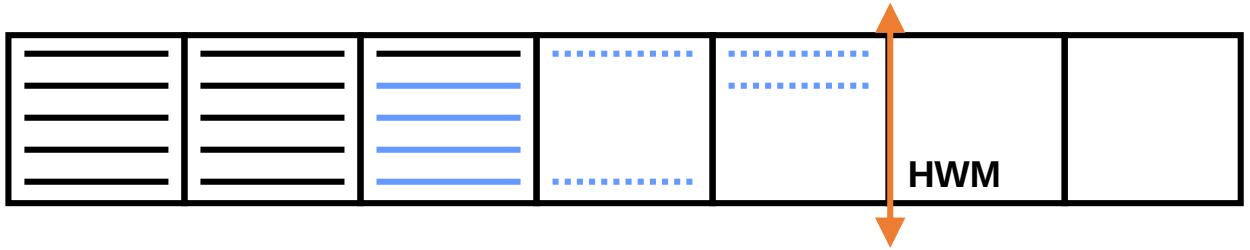
Reclaiming Space by Shrinking Segments

- Shrink is an online and in-place operation.
- It is applicable only to segments residing in ASSM tablespaces.
- Candidate segment types:
  - Heap-organized tables and index-organized tables
  - Indexes
  - Partitions and subpartitions
  - Materialized views and materialized view logs

Shrinking Segments

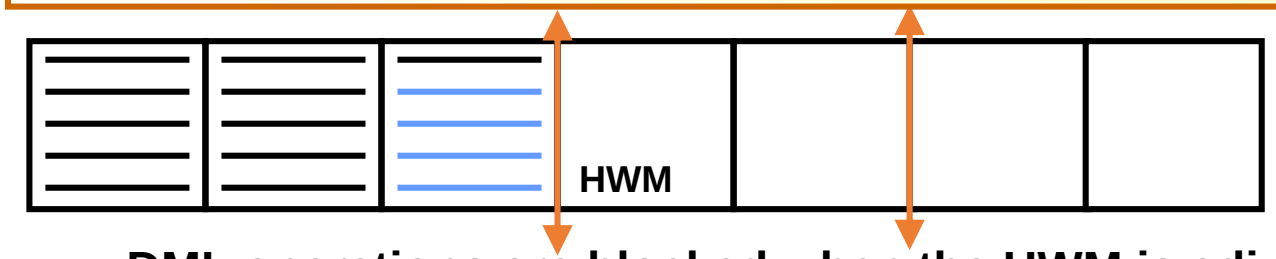


1 `ALTER TABLE employees SHRINK SPACE COMPACT;`



DML operations and queries can be issued during compaction.

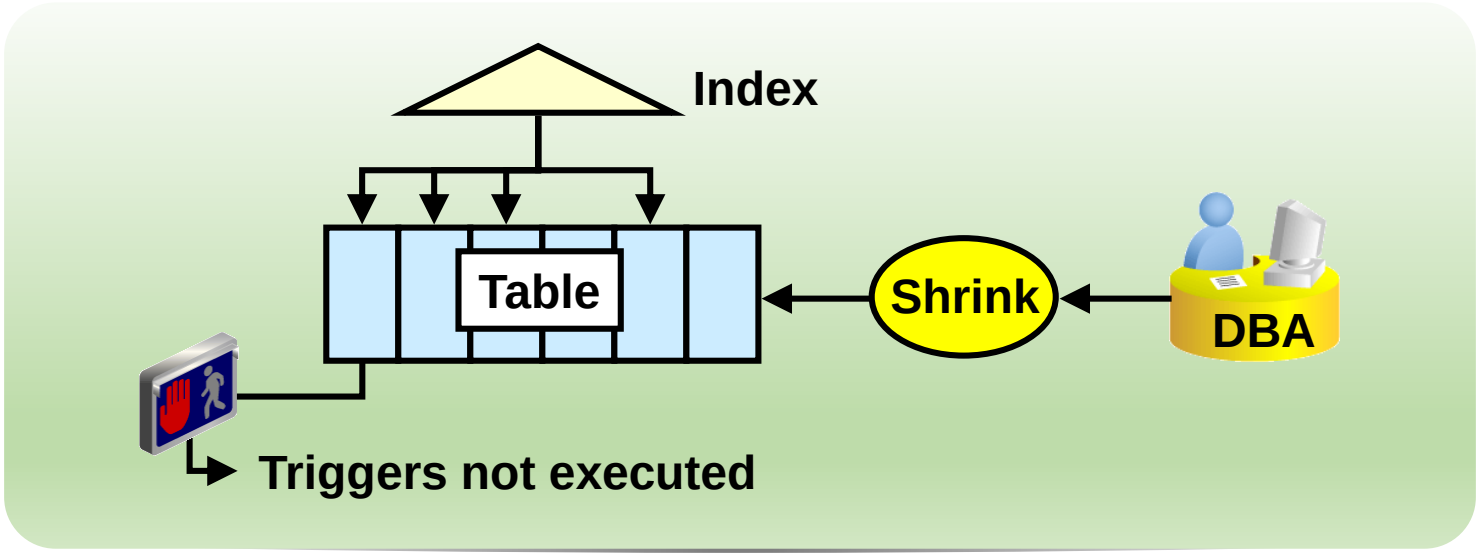
2 `ALTER TABLE employees SHRINK SPACE;`



DML operations are blocked when the HWM is adjusted.

Results of a Shrink Operation

- Improved performance and space utilization
- Indexes maintained
- Triggers not executed
- Number of migrated rows may be reduced
- Rebuilding secondary indexes on IOTs recommended



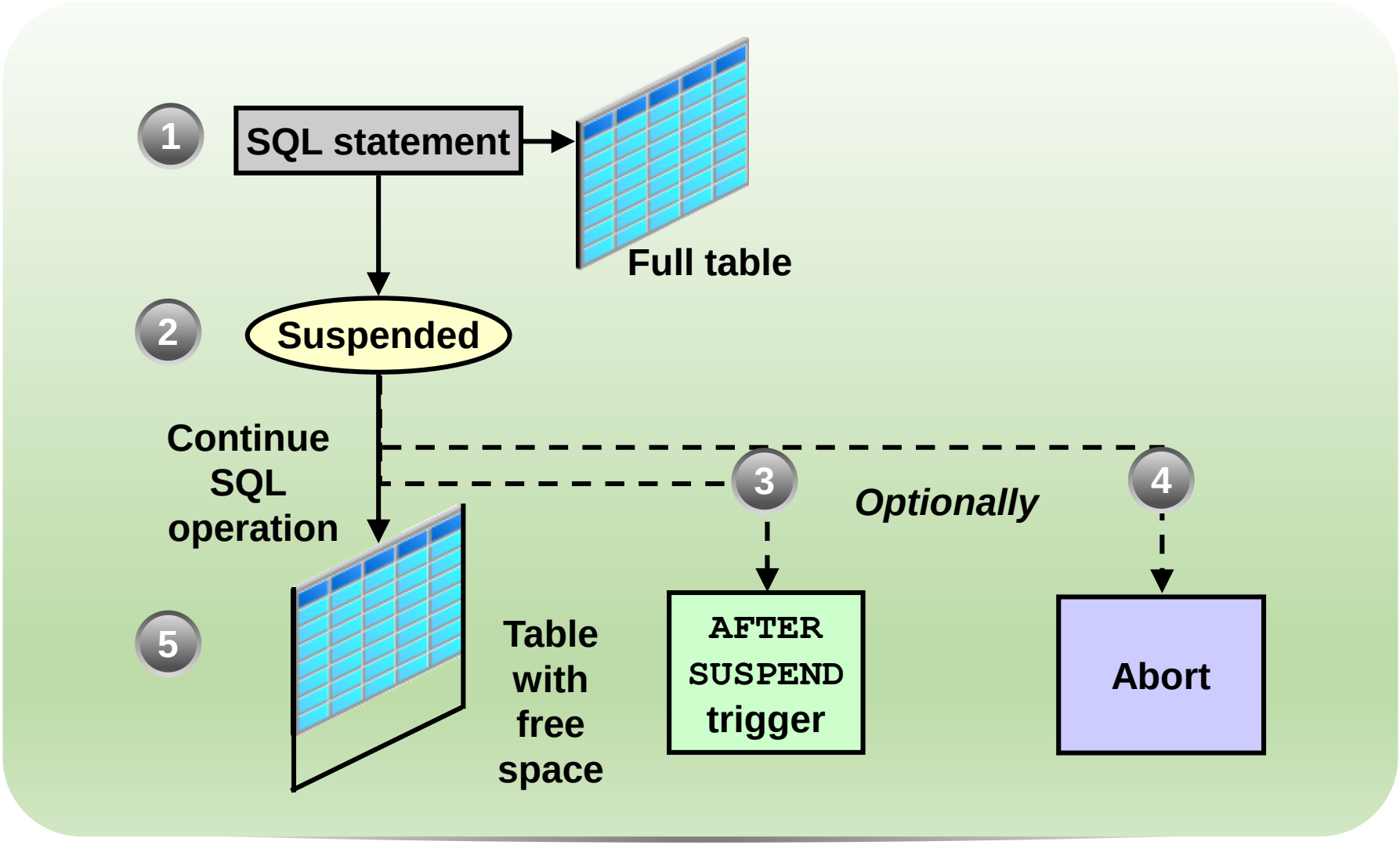
Managing Resumable Space Allocation

- A resumable statement:
  - Enables you to suspend large operations instead of receiving an error
  - Gives you a chance to fix the problem while the operation is suspended, rather than starting over
  - Is suspended for the following conditions:
    - Out of space
    - Maximum extents reached
    - Space quota exceeded
  - Can be suspended and resumed multiple times

Using Resumable Space Allocation

- Queries, DML operations, and certain DDL operations can be resumed if they encounter an out-of-space error.
- A resumable statement can be issued through SQL, PL/SQL, SQL*Loader, and Data Pump utilities, or Oracle Call Interface (OCI).
- A statement executes in resumable mode only if its session has been enabled by one of the following actions:
  - The `RESUMABLE_TIMEOUT` initialization parameter is set to a nonzero value.
  - An `ALTER SESSION ENABLE RESUMABLE` statement is issued.

Resuming Suspended Statements



What Operations Are Resumable?

- The following operations are resumable:
  - Queries: `SELECT` statements that run out of temporary space (for sort areas)
  - DML: `INSERT`, `UPDATE`, and `DELETE` statements
  - The following DDL statements:
    - **`CREATE TABLE ... AS SELECT`**
    - **`CREATE INDEX`**
    - **`ALTER INDEX ... REBUILD`**
    - **`ALTER TABLE ... MOVE PARTITION`**
    - **`ALTER TABLE ... SPLIT PARTITION`**
    - **`ALTER INDEX ... REBUILD PARTITION`**
    - **`ALTER INDEX ... SPLIT PARTITION`**
    - **`CREATE MATERIALIZED VIEW`**

Summary

- In this lesson, you should have learned how to:
  - Describe how the Oracle Database server automatically manages space
  - Save space by using compression
  - Proactively monitor and manage tablespace space usage
  - Describe segment creation in the Oracle database
  - Control deferred segment creation
  - Reclaim wasted space from tables and indexes by using the segment shrink functionality
  - Manage resumable space allocation

Practice 13: Overview

- 13-1: Managing Tablespace Space
- 13-2: Using Compression
- 13-3: Handling Resumable Space Allocation