Monitoring and Managing Storage

Objectives

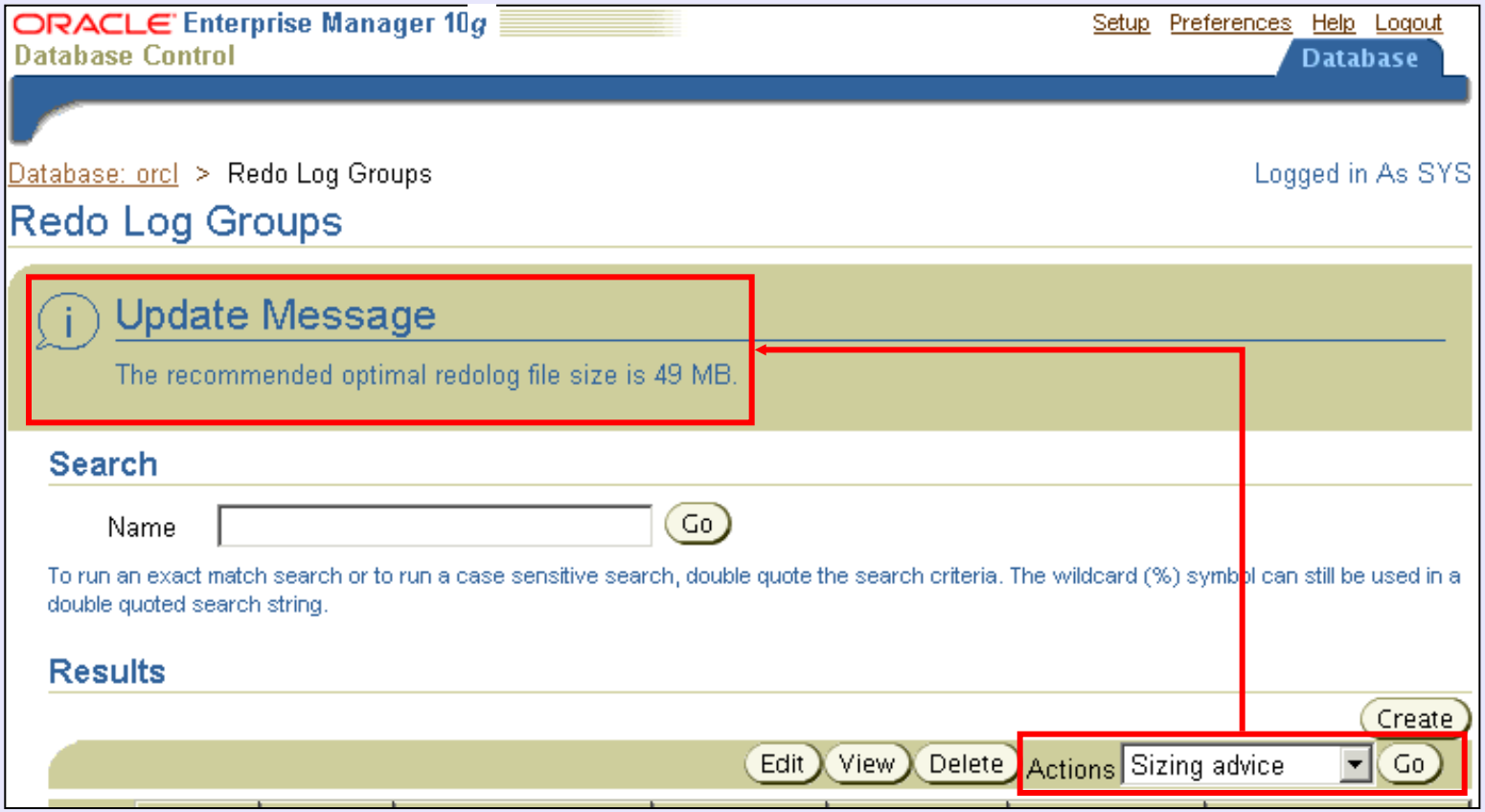
After completing this lesson, you should be able to:

- Tune redo writing and archiving operations
- Issue statements that can be suspended when encountering space condition errors
- Reduce space-related error conditions through proactively managing tablespace space usage
- Reclaim wasted space from tables and indexes using the segment-shrink functionality
- Estimate the size of new tables and indexes
- Use different storage options to improve performance of queries
- Rebuild indexes online

Online Redo Log File Configuration

- Size redo log files to minimize contention.
- Provide enough groups to prevent waiting.
- Store redo log files on separate, fast devices.
- Monitor the redo log file configuration with:
  - `V$LOGFILE`
  - `V$LOG`
  - `V$INSTANCE_RECOVERY`

Redo Logfile Sizing Advisor



Increasing the Performance of Archiving

- Allow the LGWR process to write to a disk different from the one the ARCn process is reading.
- Share the archiving work during a temporary increase in workload:

```
ALTER SYSTEM ARCHIVE LOG ALL
TO <log_archive_dest>
```

- Increase the number of archive processes.
- Change archiving speed:
  - LOG_ARCHIVE_MAX_PROCESSES
  - LOG_ARCHIVE_DEST_n

Resumable Statements

A resumable statement:

- Allows 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

Using Resumable Space Allocation

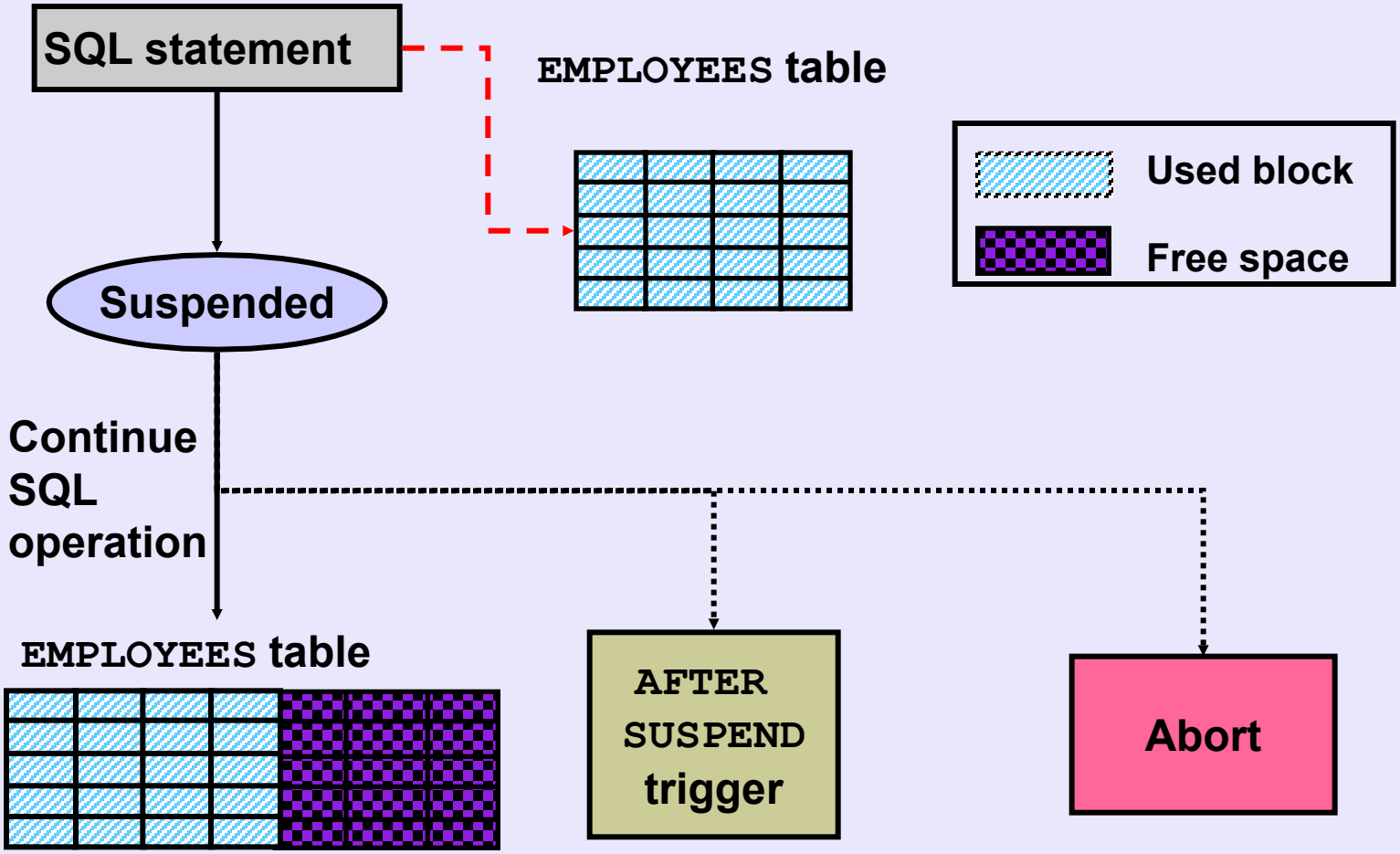
- Queries, DML operations, and certain DDL operations can be resumed if they encounter an out-of-space error.
- A resumable statement can be issued through SQL, PL/SQL, SQL*Loader, or the Oracle Call Interface (OCI).
- To run a resumable statement, you must first enable resumable statements for your session.

```
ALTER SESSION ENABLE RESUMABLE;

INSERT INTO sales_new SELECT * FROM sh.sales;

ALTER SESSION DISABLE RESUMABLE;
```

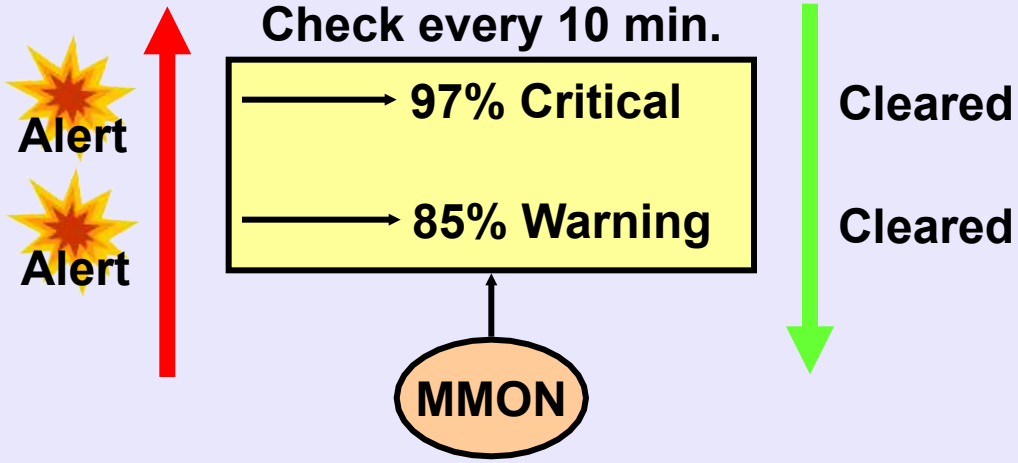
Resuming Suspended Statements



Proactive Tablespace Monitoring Overview

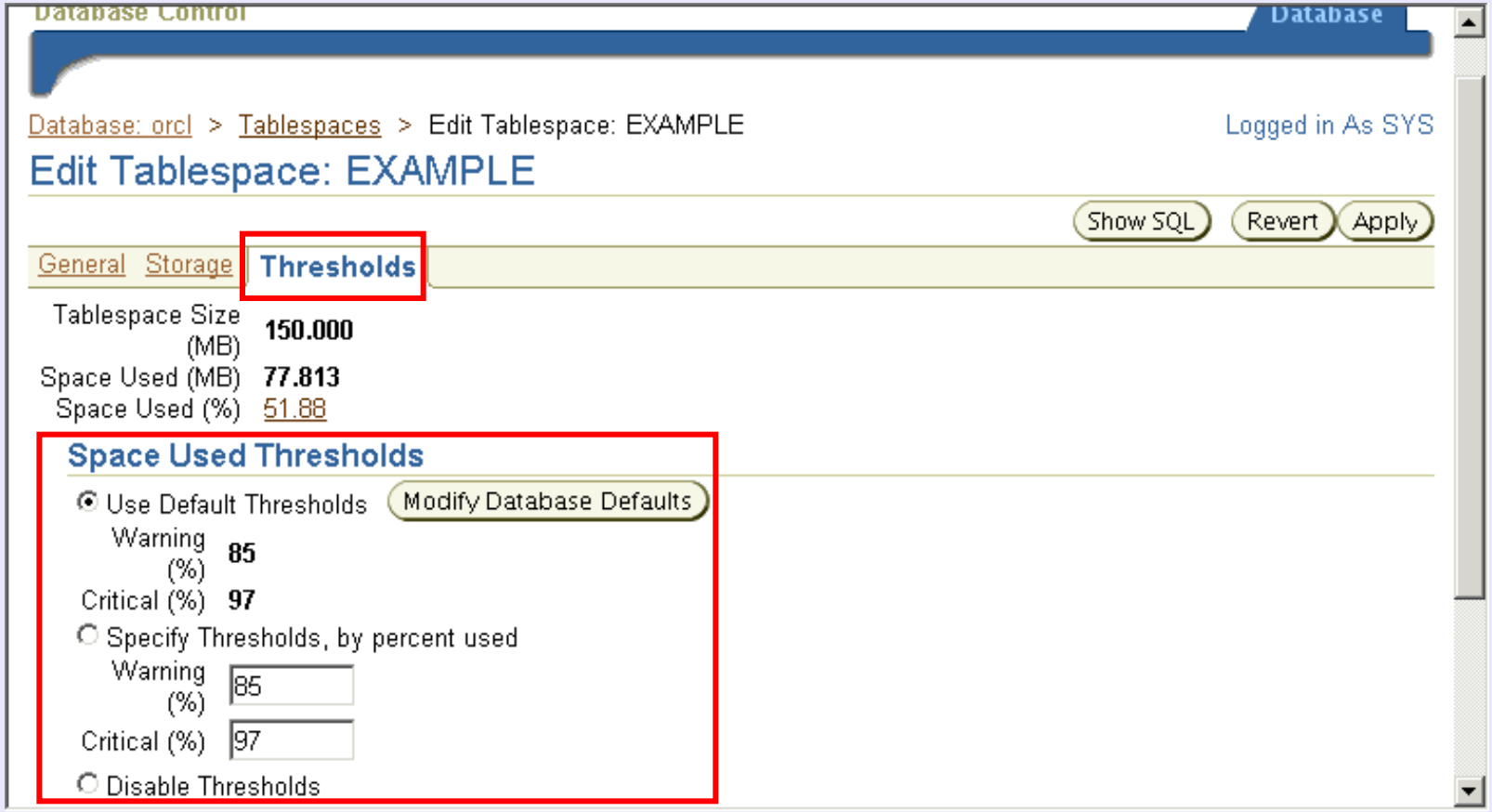
- Server-generated alerts inform you that:
  - Tablespaces are running low on available space
  - Segments are running out of space
- Data gathering and reporting provides:
  - Historical tablespace disk space usage
  - Segment growth trend analysis

Tablespace Space Usage Monitoring

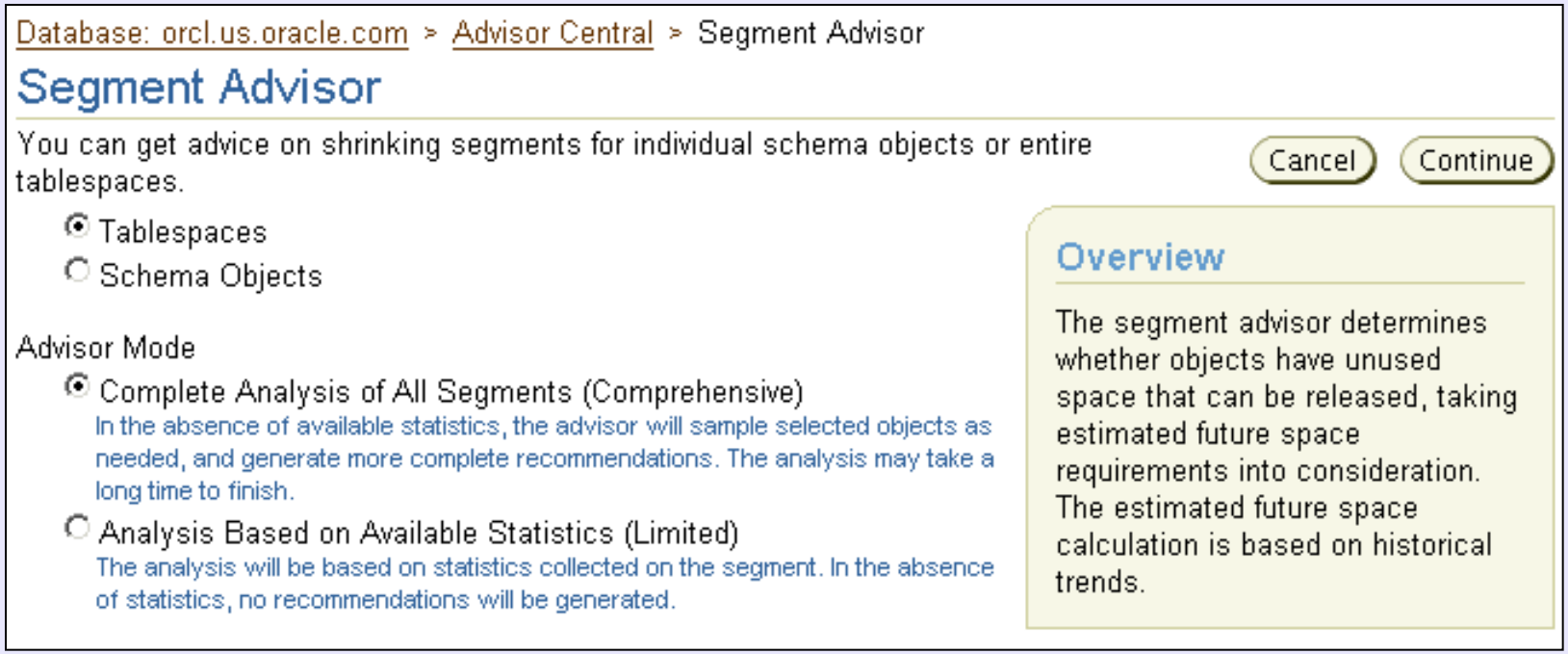


- **Read-only and offline tablespaces**: Do not set up alerts.
- **Temporary tablespace**: Threshold corresponds to space currently used by sessions.
- **Undo tablespace**: Threshold corresponds to space used by active and unexpired extents.
- **Tablespaces that can extend automatically**: Threshold is based on the maximum file size.

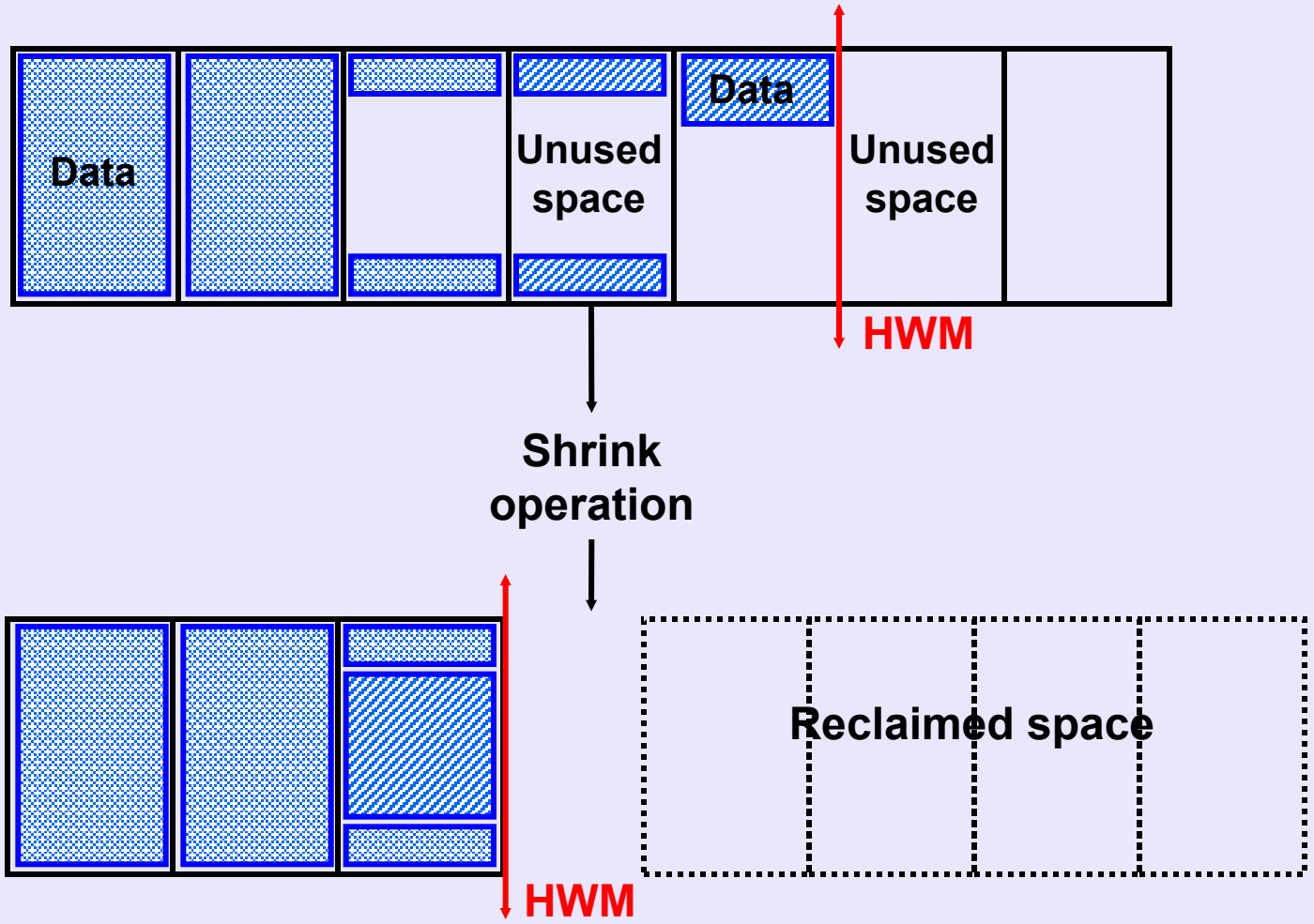
Edit Tablespace Page



Segment Advisor Overview



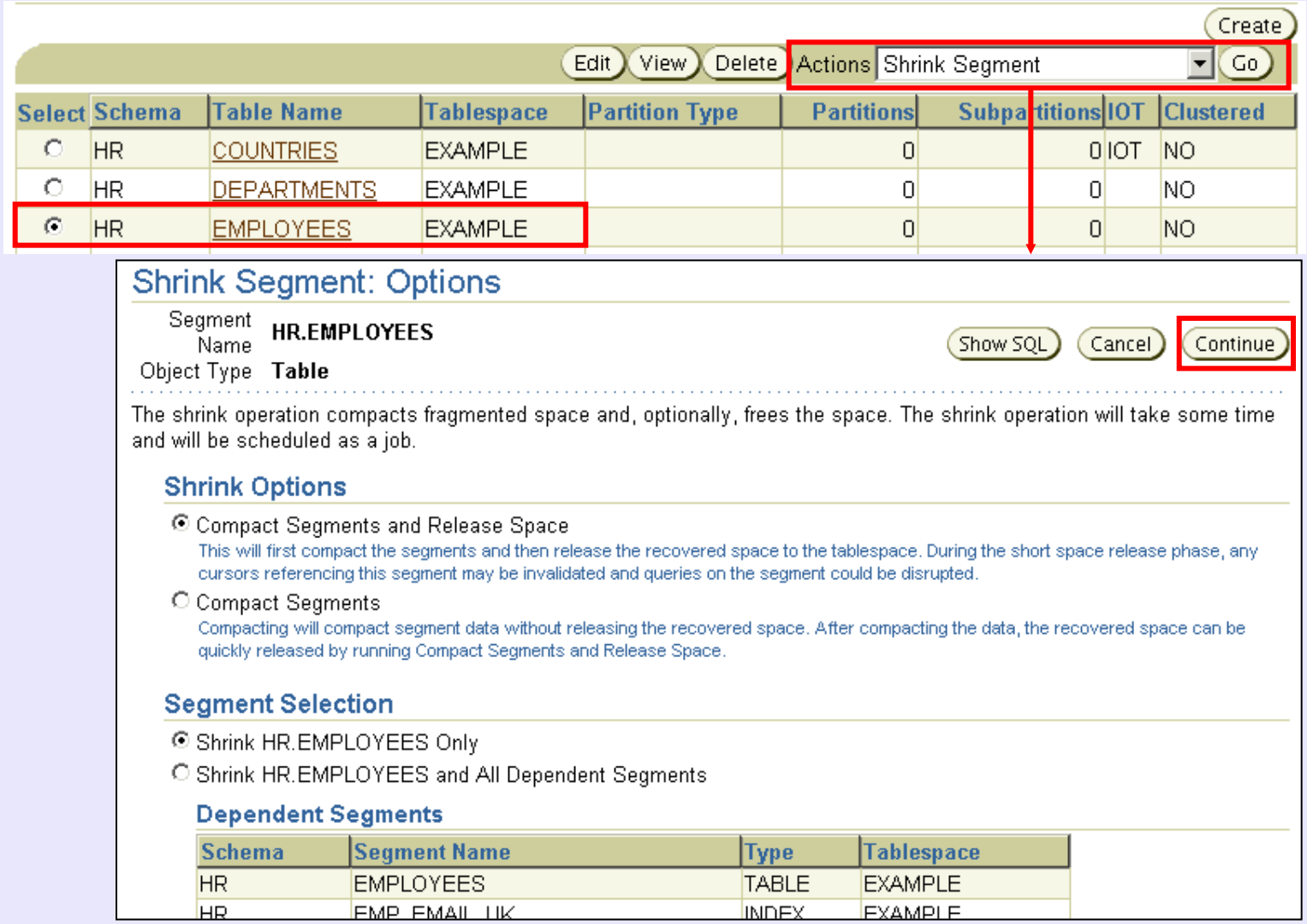
Shrinking Segments: Overview



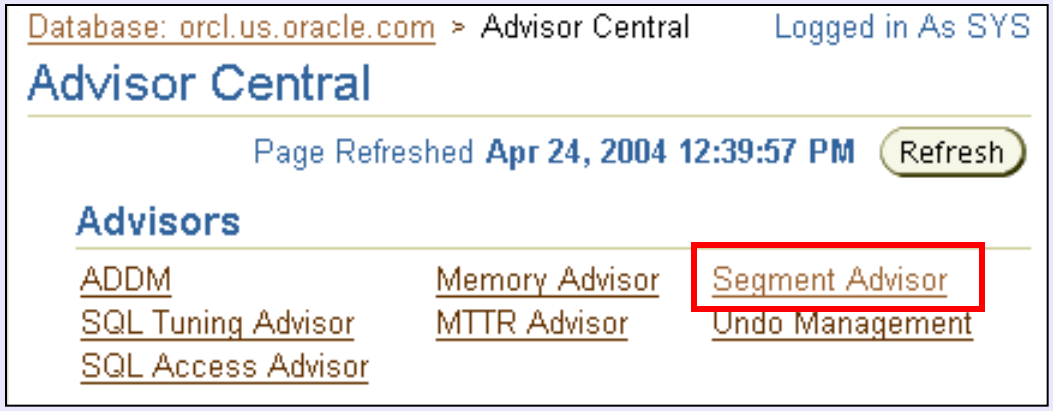
Shrinking Segments: Considerations

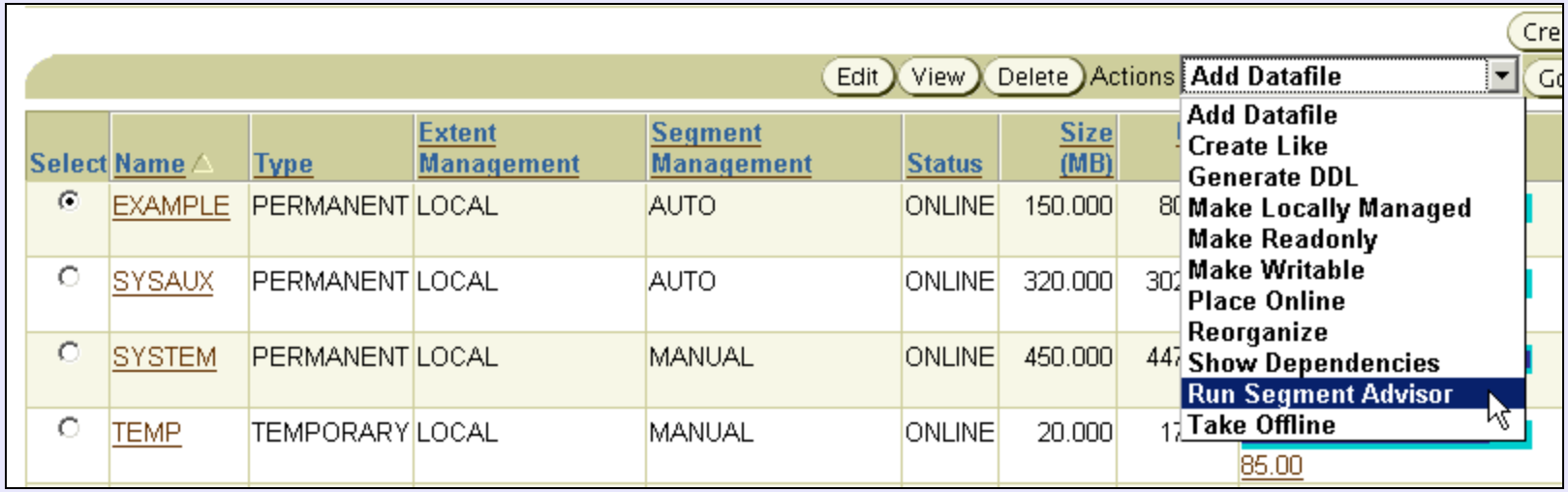
- Online and in-place operation
- Applicable only to segments residing in ASSM tablespaces
- Candidate segment types:
  - Heap-organized tables and index-organized tables
  - Indexes
  - Partitions and subpartitions
  - Materialized views and materialized view logs
- Indexes are maintained.
- Triggers are not fired.

Database Control and Segment Shrink

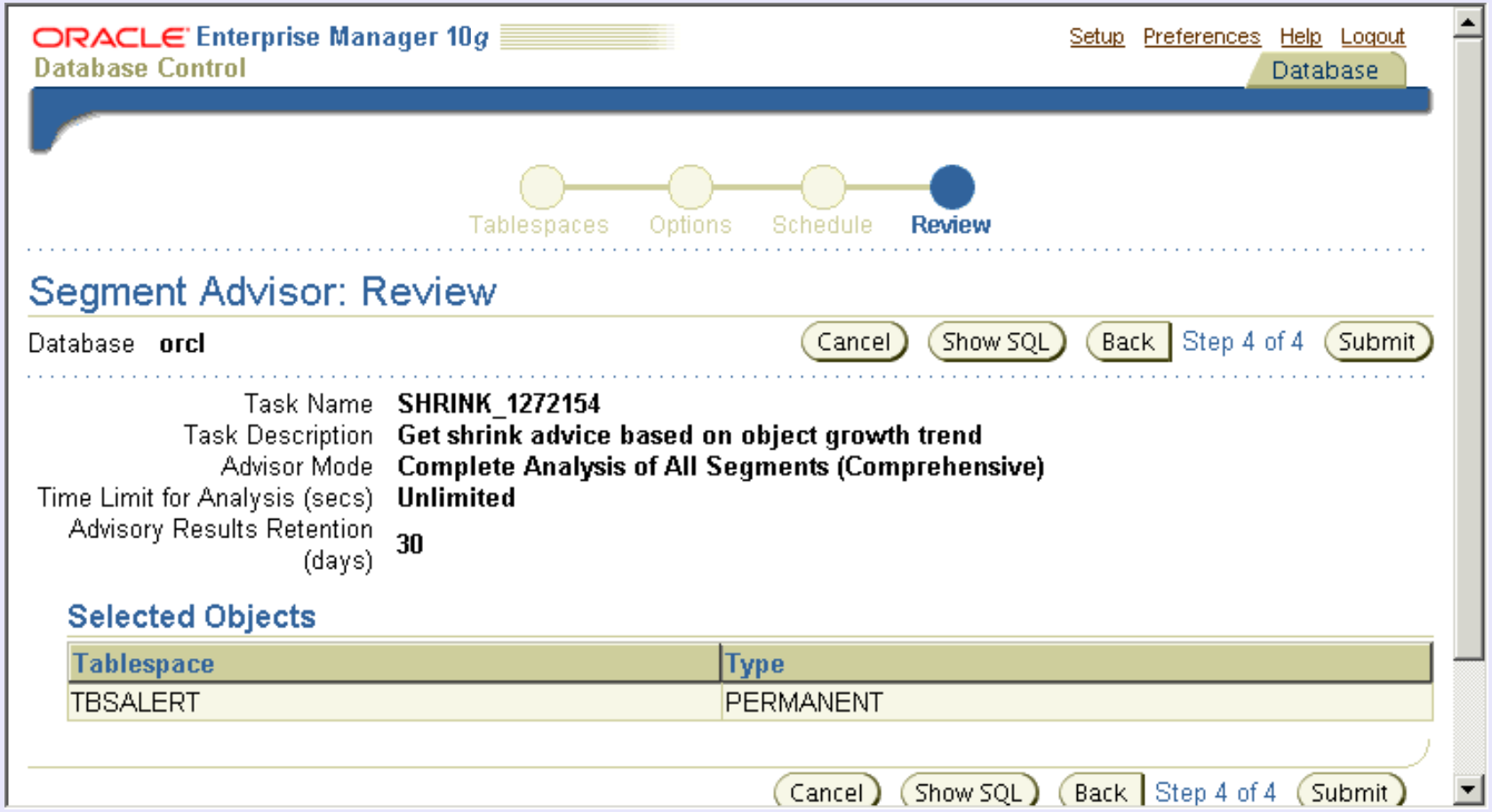


Accessing the Segment Advisor

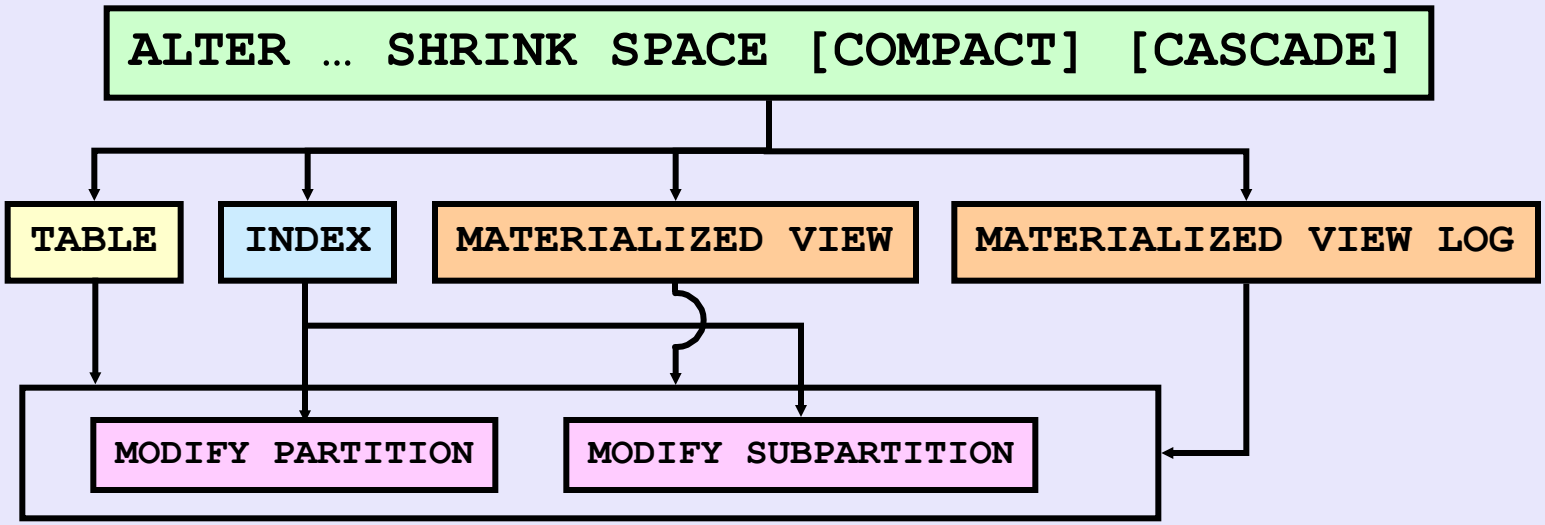




Segment Advisor



Shrinking Segments Using SQL



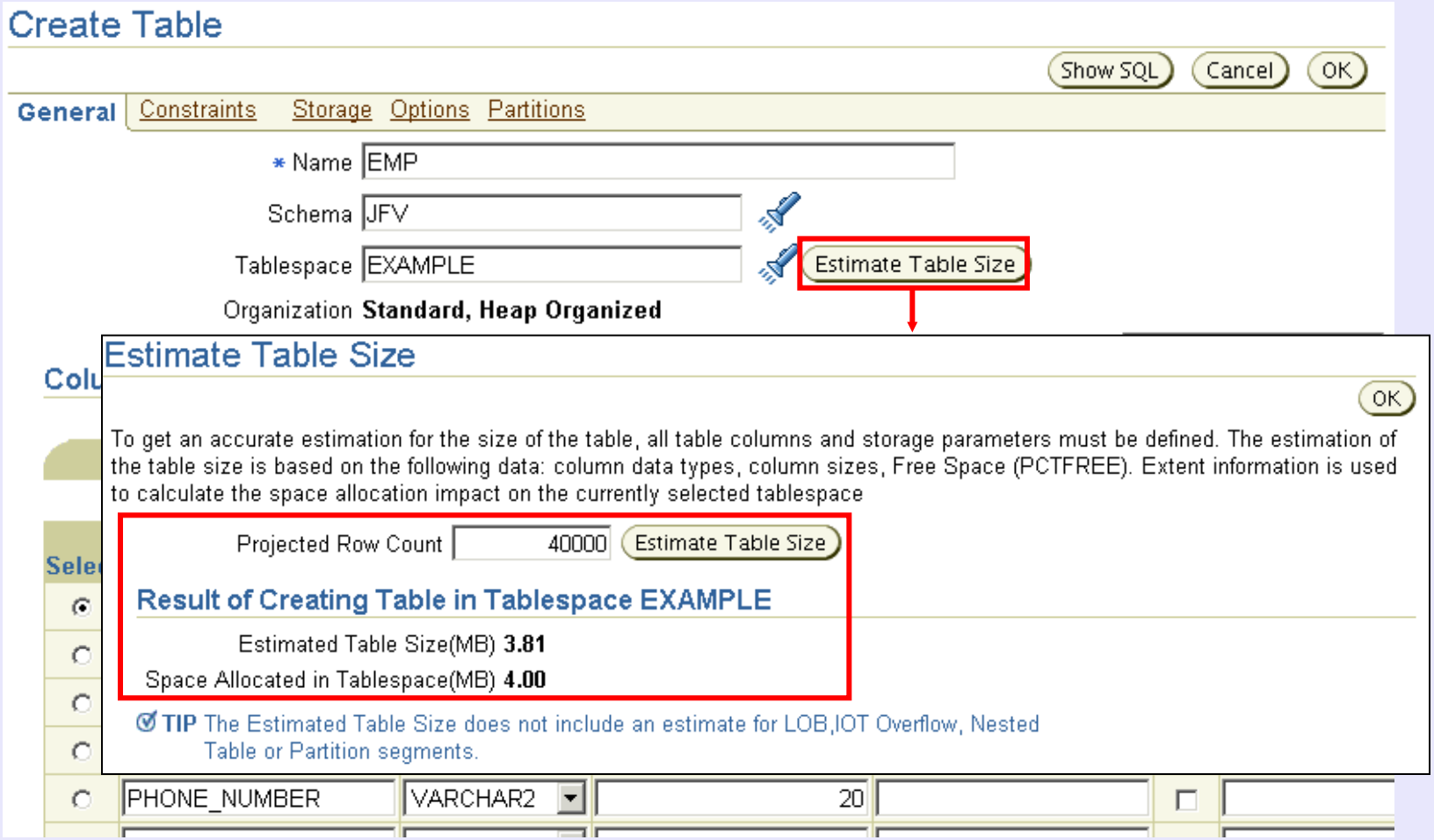

```
ALTER TABLE employees ENABLE ROW MOVEMENT;
```
1

```
ALTER TABLE employees SHRINK SPACE CASCADE;
```
2

Segment Shrink: Execution Considerations

- Use compaction only:
  - To avoid unnecessary cursor invalidation
  - During peak hours
- DML operations and queries can be issued during compaction.
- DML operations are blocked when the HWM is adjusted.

Segment Resource Estimation



Growth Trend Report

- Used by the Segment Advisor
- Space usage statistics are collected into AWR