Performance

Objectives

- After completing this lesson, you should be able to:
  - Monitor performance in a CDB and PDBs
  - Manage SGA and PGA limits at the PDB level
  - Manage AWR snapshots at the CDB and PDB levels
  - Run ADDM tasks for CDB and PDB recommendations
  - Manage application shared object statistics
  - Control query DOP involving the `containers()` construct
  - Manage Heat Map and ADO policy declaration in a PDB
  - Manage a CDB fleet
  - Describe use cases for Consolidated Database Replay

Tuning a CDB

- Basic rules:
  - The PDB appears to applications exactly the same as a non-CDB.
  - Some initialization parameters can be set at the PDB level.
  - SQL statements are tuned on a per PDB basis.
    - Common objects statistics are gathered in the application root of the common object.
    - Local objects statistics are gathered in the PDB of the local object.
  - AWR tools run at the CDB and PDB level: ASH / ADDM
  - AWR snapshots can be taken at CDB or PDB level.
  - The cursors in the shared pool are identified by PDB.
  - Instance-wide information is kept in the CDB root container.

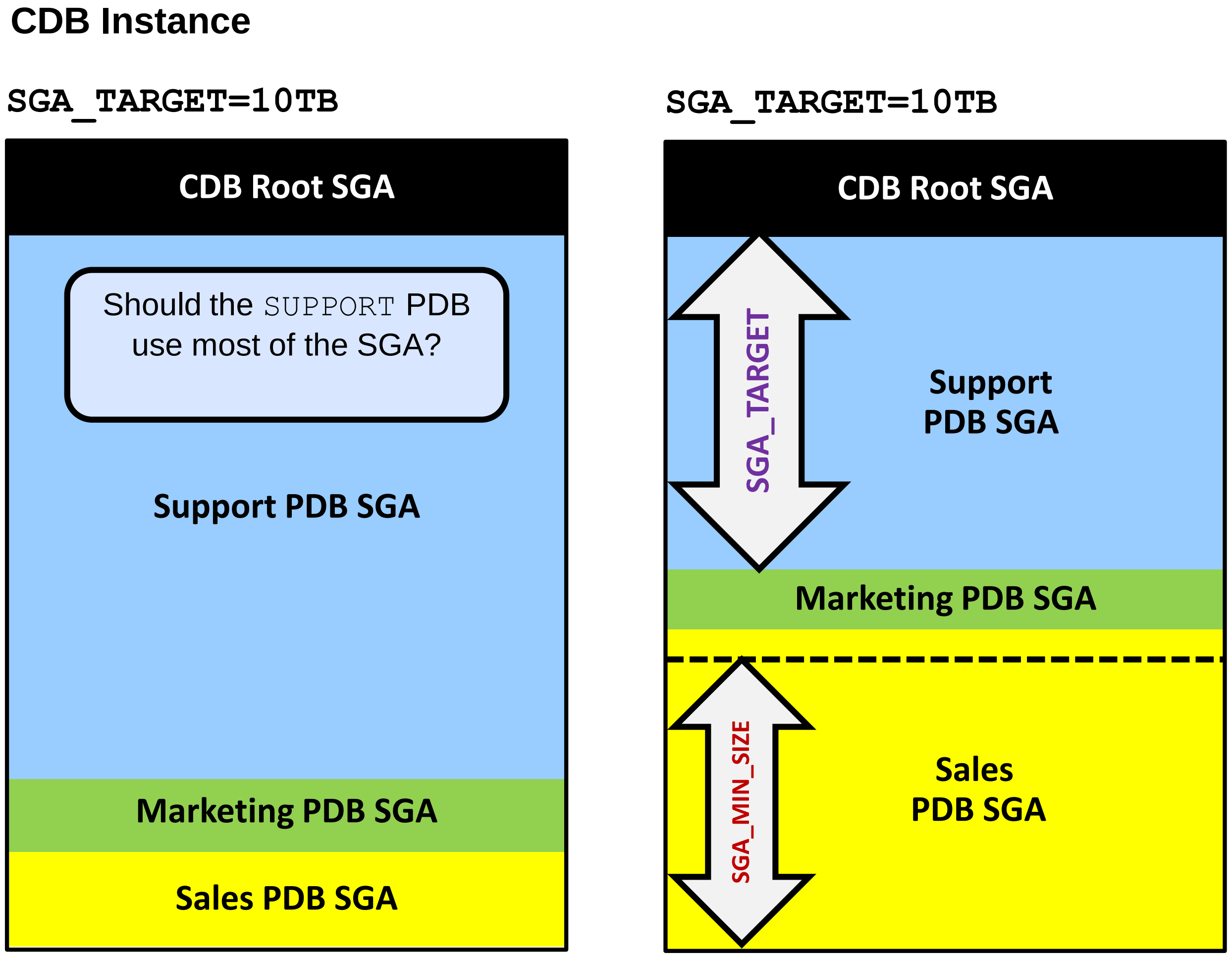
Sizing the CDB

- Areas of concern:
  - Memory (SGA and PGA)
    - Buffer Cache
    - Shared Pool
    - Program Global Area
  - CPU over allocation
  - SQL Tuning

Testing the Estimates

- Consolidated Database Replay tests:
  - Consolidation of servers
  - Scale-up
  - Peak load capacity

Managing SGA for PDBs



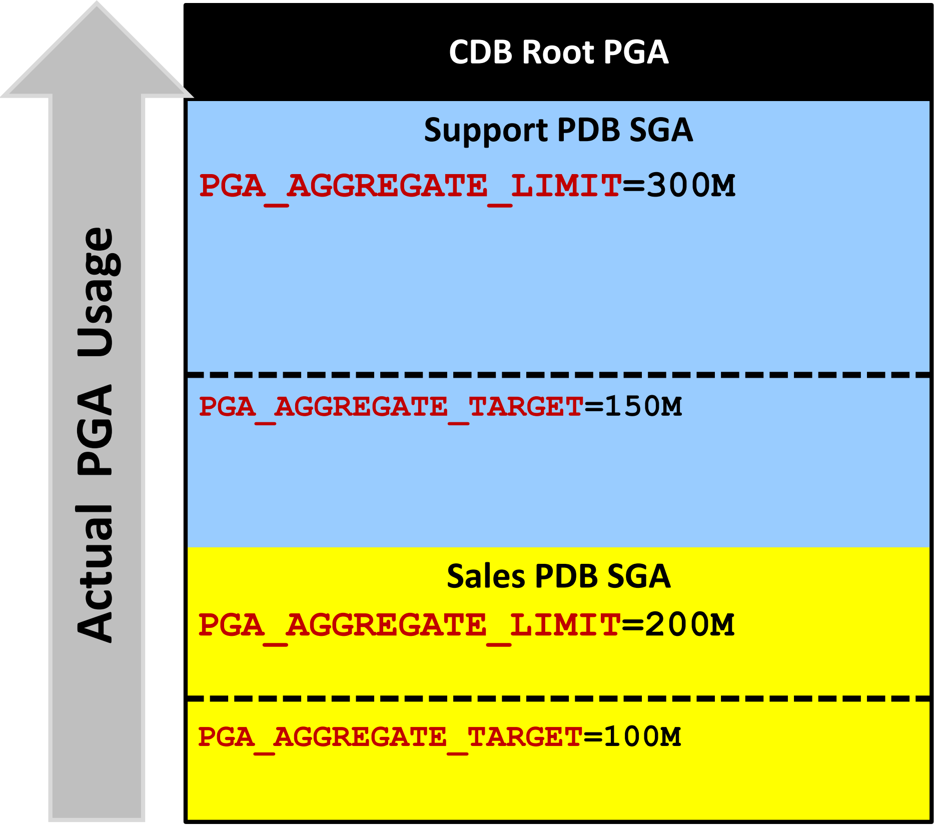
- SGA_TARGET set at the PDB level enforces a hard limit for the PDB's SGA.
- SGA_TARGET at the PDB level provides more SGA for other containers.
- SGA_MIN_SIZE set for a PDB guarantees SGA space for the PDB.
- Parameters at PDB level are:
  - DB_CACHE_SIZE
  - SHARED_POOL_SIZE
- (PDB minimums) cannot be > 50 percent of memory.

Managing PGA for PDBs

CDB Instance

PGA_AGGREGATE_LIMIT=1TB

PGA_AGGREGATE_TARGET=500GB



Actual PGA Usage

CDB Root PGA

Support PDB SGA

PGA_AGGREGATE_LIMIT=300M

PGA_AGGREGATE_TARGET=150M

Sales PDB SGA

PGA_AGGREGATE_LIMIT=200M

PGA_AGGREGATE_TARGET=100M

Instance PGA_AGGREGATE_LIMIT

- No more PGA can be allocated.
- Call or session of the largest PGA users is terminated.

Instance PGA_AGGREGATE_TARGET

- All sessions must use TEMP rather than PGA.
- PDB PGA_AGGREGATE_LIMIT
- PDB PGA_AGGREGATE_TARGET
- These parameters set the same behavior at the PDB level.

Monitoring PDB Memory Usage

- Monitor memory usage before and after configuring PDB memory parameters.

```
V$RSRCPDBMETRIC /
V$RSRCPDBMETRIC_HISTORY /
V$RSRC_PDB
     SGA_BYTES
     PGA_BYTES
     SHARED_POOL_BYTES
     BUFFER_CACHE_BYTES
```

- Monitor per PDB history statistics.

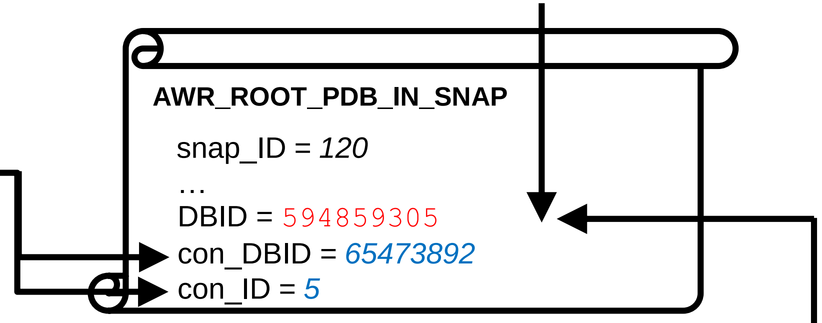
```
V$RSRC_PDB_HISTORY
```

AWR and ADDM Behavior

- AWR snapshots are created to collect statistics:

```
SQL> CONNECT / as sysdba
SQL> exec DBMS_WORKLOAD_REPOSITORY.CREATE_SNAPSHOT (FLUSH_LEVEL => 'TYPICAL', DBID => 594859305)
```

  - Collects statistics at the PDB level
  - Collects statistics for each PDB opened
- ADDM runs at the CDB level only:
  - Recommendations at the CDB level and PDB level



AWR_ROOT_PDB_IN_SNAP

snap_ID = *120*

…

DBID = 594859305

con_DBID = *65473892*

con_ID = *5*

```
SQL> CONNECT / AS SYSDBA
SQL> var task_name VARCHAR2(60)
SQL> DECLARE
       taskid NUMBER;
     BEGIN
       dbms_advisor.create_task('ADDM',taskid,:task_name);
       dbms_advisor.set_task_parameter(:task_name, 'START_SNAPSHOT', 97);
       dbms_advisor.set_task_parameter(:task_name, 'END_SNAPSHOT', 119);
       dbms_advisor.set_task_parameter(:task_name, 'DB_ID', 594859305);
       dbms_advisor.execute_task(:task_name);
     END;
/
```

PDB-Level Snapshot Views

- CDB-level AWR views

AWR_ROOT_PDB_IN_SNAP

snap_ID = *120*
…
DBID = 594859305
con_DBID = *65473892*
con_ID = *5*

AWR_ROOT_RSRC_PDB_METRIC

snap_ID = *120*
…
DBID = *594859305*
con_DBID = *65473892*
con_ID = *5*

- PDB-level AWR views

AWR_PDB_PARAMETER

snap_ID = *120*
DBID = *594859305*
parameter_name = *db_performance_profile*
value = *PROF_HIGH*
con_DBID = *65473892*
con_ID = *5*

AWR_PDB_RSRC_PDB_METRIC

snap_ID = *120*
DBID = *594859305*
pga_used = *491030*
cpu_consumed_time = *58018*
con_DBID = *65473892*
con_ID = *5*

AWR_PDB_SQL_SUMMARY

snap_ID = *120*
DBID = *594859305*
total_sql= *419*
total_sql_mem = *15252584*
Single_use_sql_mem = *473848*
con_DBID = *65473892*
con_ID = *5*

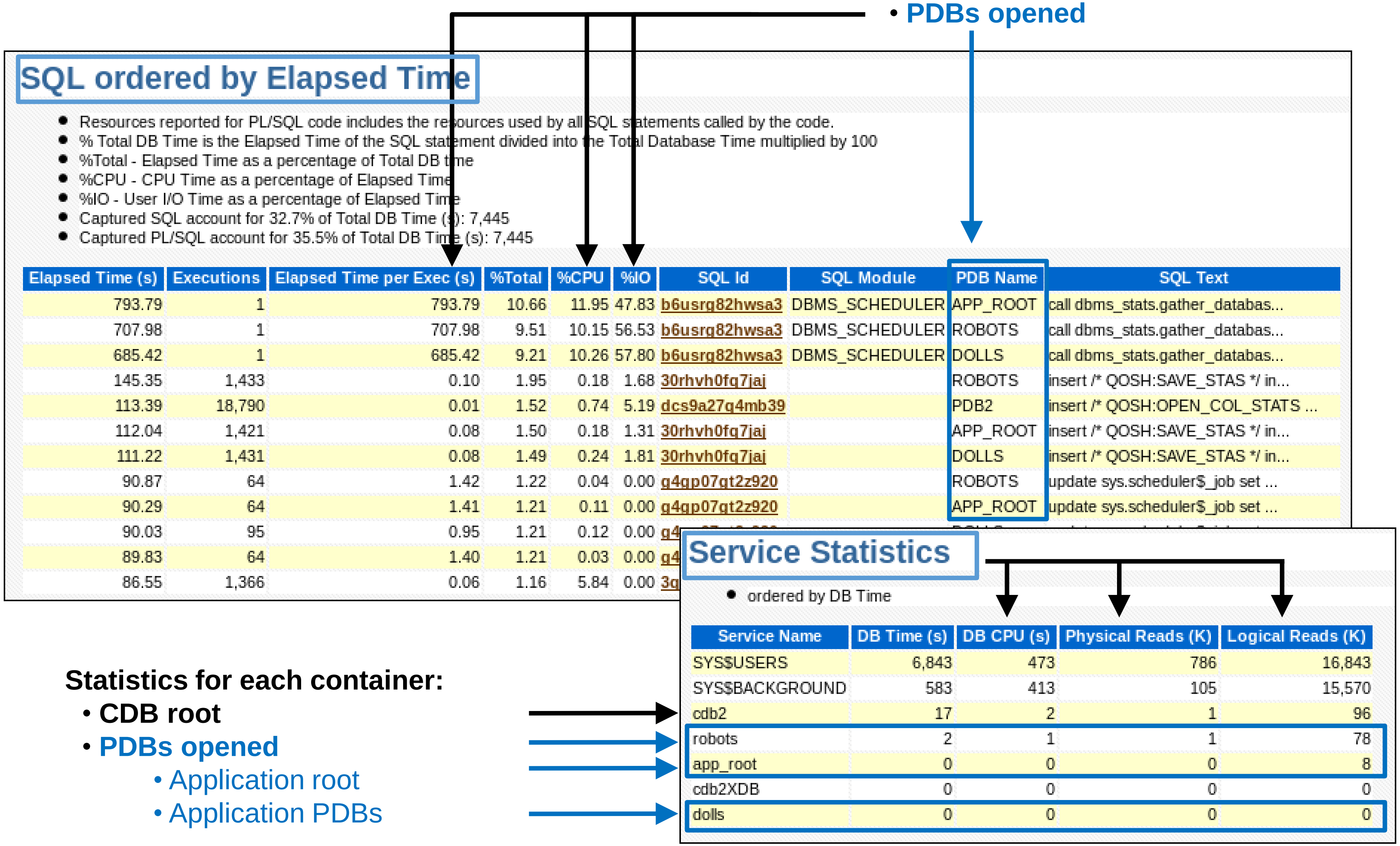
- List of all PDB-level AWR views

```
SQL> SELECT view_name FROM dba_views WHERE  view_name LIKE '%AWR%_PDB%';
```

AWR Report

Statistics for each container:

- CDB root
- PDBs opened



SQL ordered by Elapsed Time

- Resources reported for PL/SQL code includes the resources used by all SQL statements called by the code.
- % Total DB Time is the Elapsed Time of the SQL statement divided into the Total Database Time multiplied by 100
- %Total - Elapsed Time as a percentage of Total DB time
- %CPU - CPU Time as a percentage of Elapsed Time
- %IO - User I/O Time as a percentage of Elapsed Time
- Captured SQL account for 32.7% of Total DB Time (s): 7,445
- Captured PL/SQL account for 35.5% of Total DB Time (s): 7,445

Elapsed Time (s)	Executions	Elapsed Time per Exec (s)	%Total	%CPU	%IO	SQL Id	SQL Module	PDB Name	SQL Text
793.79	1	793.79	10.66	11.95	47.83	b6usrg82hwsa3	DBMS_SCHEDULER	APP_ROOT	call dbms_stats.gather_databas...
707.98	1	707.98	9.51	10.15	56.53	b6usrg82hwsa3	DBMS_SCHEDULER	ROBOTS	call dbms_stats.gather_databas...
685.42	1	685.42	9.21	10.26	57.80	b6usrg82hwsa3	DBMS_SCHEDULER	DOLLS	call dbms_stats.gather_databas...
145.35	1,433	0.10	1.95	0.18	1.68	30rhvh0fq7jaj		ROBOTS	insert /* QOSH:SAVE_STAS */ in...
113.39	18,790	0.01	1.52	0.74	5.19	dcs9a27q4mb39		PDB2	insert /* QOSH:OPEN_COL_STATS ...
112.04	1,421	0.08	1.50	0.18	1.31	30rhvh0fq7jaj		APP_ROOT	insert /* QOSH:SAVE_STAS */ in...
111.22	1,431	0.08	1.49	0.24	1.81	30rhvh0fq7jaj		DOLLS	insert /* QOSH:SAVE_STAS */ in...
90.87	64	1.42	1.22	0.04	0.00	g4gp07gt2z920		ROBOTS	update sys.scheduler$_job set ...
90.29	64	1.41	1.21	0.11	0.00	g4gp07gt2z920		APP_ROOT	update sys.scheduler$_job set ...
90.03	95	0.95	1.21	0.12	0.00	g4			
89.83	64	1.40	1.21	0.03	0.00	g4			
86.55	1,366	0.06	1.16	5.84	0.00	3q			

Service Statistics

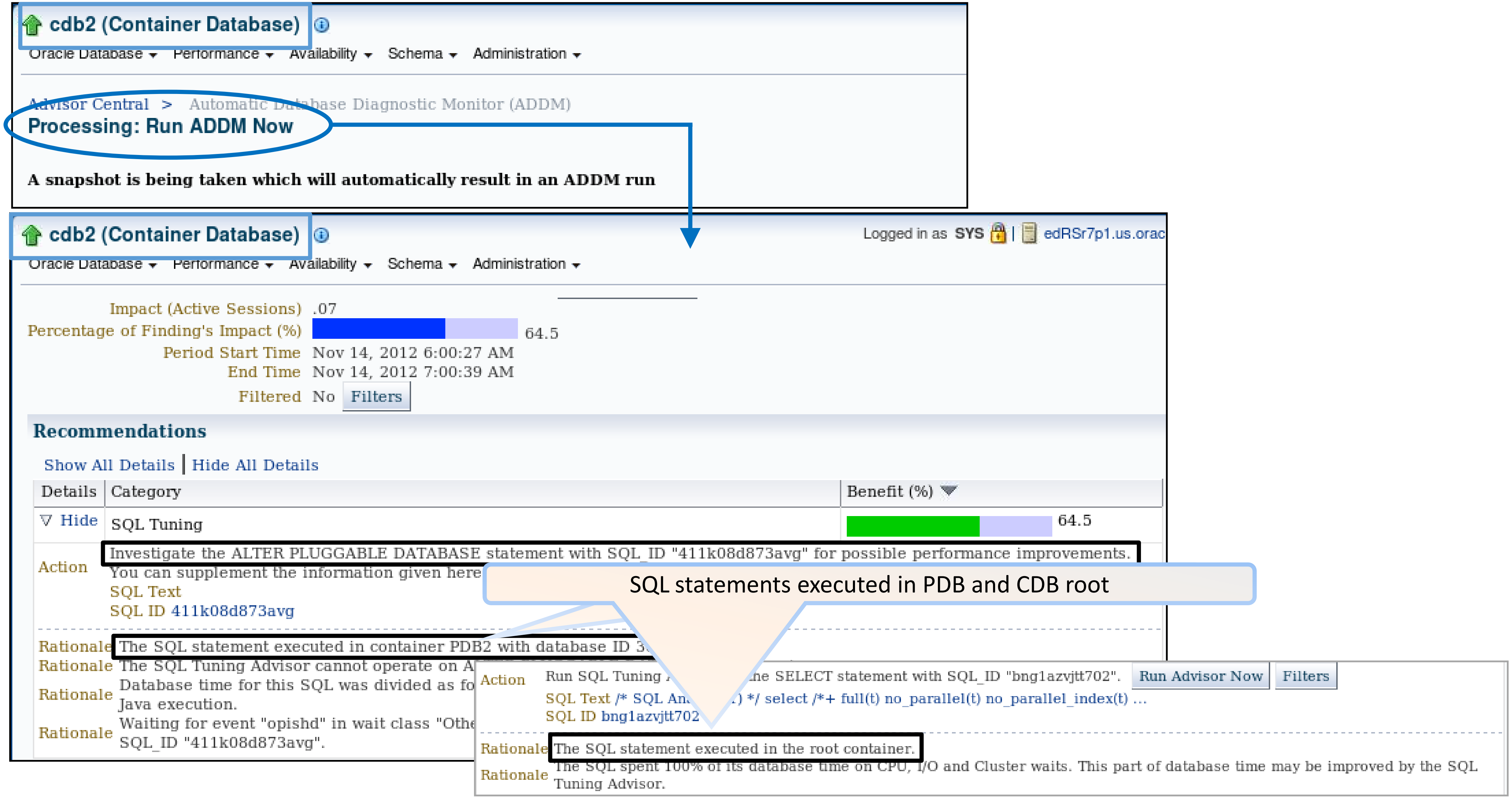
- ordered by DB Time

Service Name	DB Time (s)	DB CPU (s)	Physical Reads (K)	Logical Reads (K)
SYS$USERS	6,843	473	786	16,843
SYS$BACKGROUND	583	413	105	15,570
cdb2	17	2	1	96
robots	2	1	1	78
app_root	0	0	0	8
cdb2XDB	0	0	0	0
dolls	0	0	0	0

Statistics for each container:

- CDB root
- PDBs opened
  - Application root
  - Application PDBs

ADDM Tasks: At the CDB Level Only



cdb2 (Container Database)
Oracle Database ▾ Performance ▾ Availability ▾ Schema ▾ Administration ▾

Advisor Central > Automatic Database Diagnostic Monitor (ADDM)

Processing: Run ADDM Now

A snapshot is being taken which will automatically result in an ADDM run

cdb2 (Container Database)
Logged in as SYS | edRSr7p1.us.orac
Oracle Database ▾ Performance ▾ Availability ▾ Schema ▾ Administration ▾

Impact (Active Sessions) .07
Percentage of Finding's Impact (%) 64.5
Period Start Time Nov 14, 2012 6:00:27 AM
End Time Nov 14, 2012 7:00:39 AM
Filtered No Filters

Recommendations

Show All Details | Hide All Details

Details	Category	Benefit (%)
▽ Hide	SQL Tuning	64.5

Action: Investigate the ALTER PLUGGABLE DATABASE statement with SQL_ID "411k08d873avg" for possible performance improvements.
You can supplement the information given here
SQL Text
SQL ID 411k08d873avg

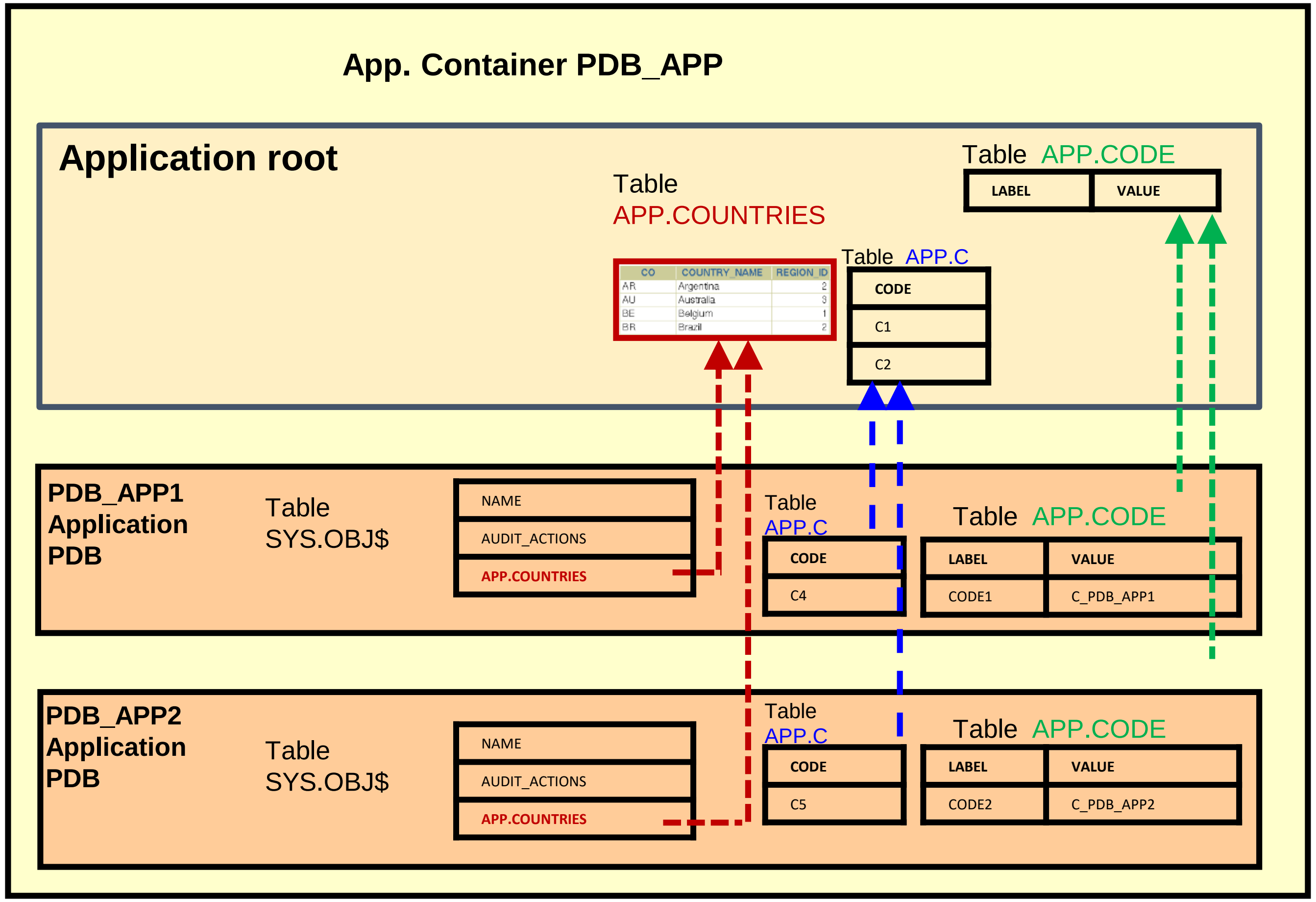
Rationale: The SQL statement executed in container PDB2 with database ID 3
Rationale: The SQL Tuning Advisor cannot operate on A
Rationale: Database time for this SQL was divided as fo
Java execution.
Rationale: Waiting for event "opishd" in wait class "Othe
SQL_ID "411k08d873avg".

SQL statements executed in PDB and CDB root

Action: Run SQL Tuning A … the SELECT statement with SQL_ID "bng1azvjtt702". Run Advisor Now Filters
SQL Text /* SQL An…) */ select /*+ full(t) no_parallel(t) no_parallel_index(t) ...
SQL ID bng1azvjtt702

Rationale: The SQL statement executed in the root container.
Rationale: The SQL spent 100% of its database time on CPU, I/O and Cluster waits. This part of database time may be improved by the SQL Tuning Advisor.

Basic Rules: Statistics for Common Objects



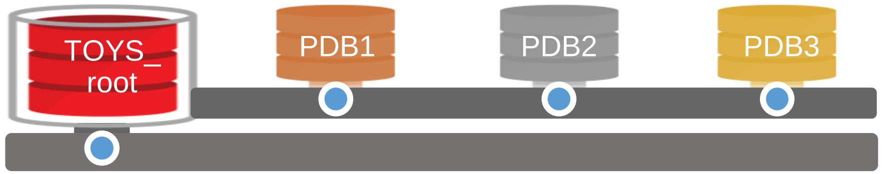
CO	COUNTRY_NAME	REGION_ID
AR	Argentina	2
AU	Australia	3
BE	Belgium	1
BR	Brazil	2

- ➢ Statistics for common data-linked objects are gathered in the application root.
- ➢ Statistics for common metadata-linked objects are gathered in the application PDB.
- ➢ Statistics for common extended data-linked objects are gathered both in the application root and application PDB.
- ➢ Statistics for local objects are gathered in the application PDB.

Controlling the Degree of Parallelism of Queries

```
SQL> CONNECT toys_app@toys_root
```

```
SQL> SELECT sum(revenue), year
     FROM   CONTAINERS(sales_data)
     WHERE  year = 2014 GROUP BY year;
```



➔ Queries using the `containers()` construct execute in parallel by default.

➔The query DOP used is 4: sum (app. root + opened application PDBs).

```
SQL> ALTER SESSION SET containers_parallel_degree = 12;
```

```
SQL> SELECT sum(revenue), year FROM CONTAINERS(sales_data)
     WHERE  year =  2014  GROUP  BY year;
```

➔The query DOP that is used for each statement by using the `containers()` construct is now 12.

```
SQL> SELECT /*+ CONTAINERS(DEFAULT_PDB_HINT=PARALLEL 8*/ sum(revenue), year
     FROM   CONTAINERS(sales_data)
     WHERE  year =  2014  GROUP  BY year;
```

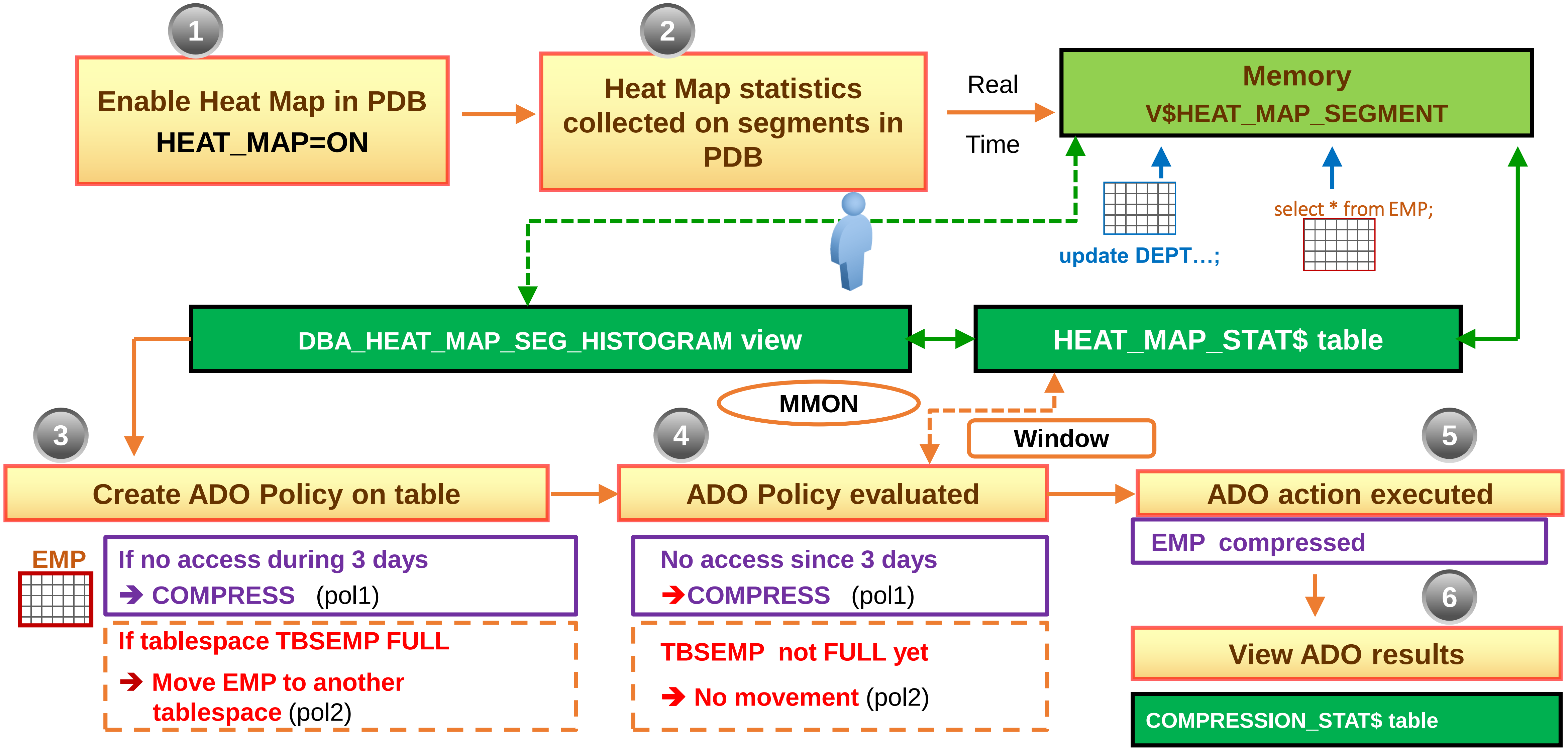
➔The query DOP that is used for the statement is now 8.

Heat Map and ADO Support

- Oracle Database CDBs support ADO and Heat Map statistics.
  - ADO policies automatically compress data in objects in PDBs.
  - ADO policies automatically move segments in PDBs to other tablespaces in the same PDB when necessary.
  - ADO is dependent on Heat Map statistics collection and does not work unless Heat Map is enabled.

```
HEAT_MAP = ON
```

Managing Heat Map and ADO Policies in PDB



1

Enable Heat Map in PDB

HEAT_MAP=ON

2

Heat Map statistics collected on segments in PDB

Real Time

Memory

V$HEAT_MAP_SEGMENT

update DEPT…;

select * from EMP;

DBA_HEAT_MAP_SEG_HISTOGRAM view

HEAT_MAP_STAT$ table

MMON

Window

3

Create ADO Policy on table

EMP

If no access during 3 days

➔ COMPRESS (pol1)

If tablespace TBSEMP FULL

➔ Move EMP to another tablespace (pol2)

4

ADO Policy evaluated

No access since 3 days

➔COMPRESS (pol1)

TBSEMP not FULL yet

➔ No movement (pol2)

5

ADO action executed

EMP compressed

6

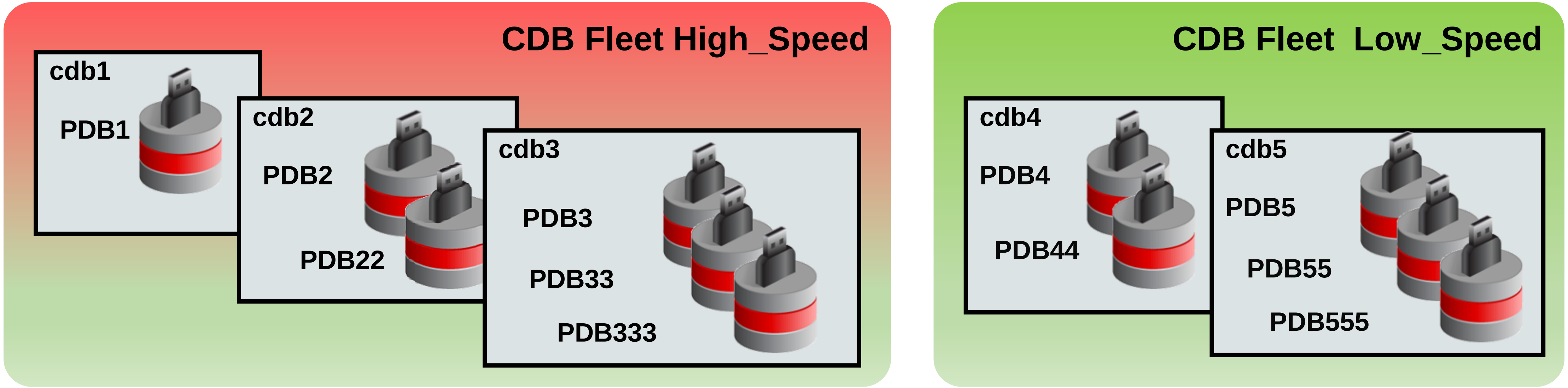
View ADO results

COMPRESSION_STAT$ table

19c

CDB Fleet

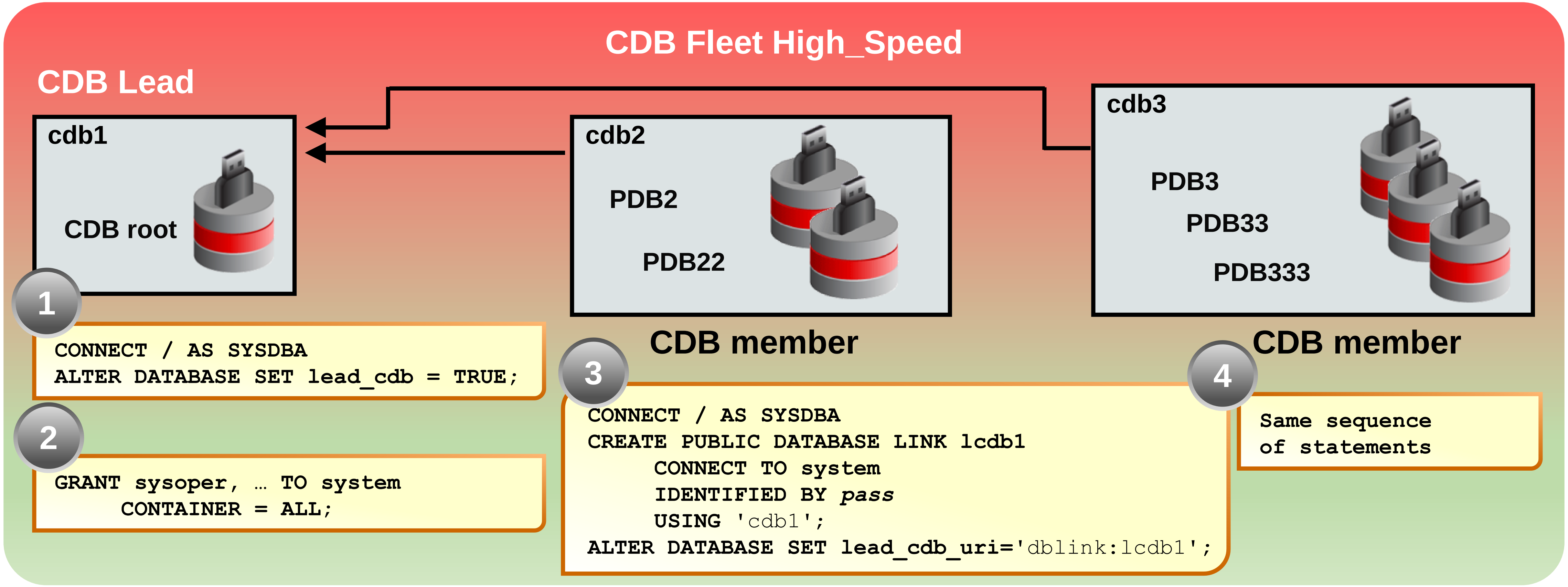
- A CDB fleet is a collection of different CDBs that can be managed as one logical CDB:
  - To provide the underlying infrastructure for massive scalability and centralized management of many CDBs
  - To provision more than the maximum number of PDBs for an application



  - To manage appropriate server resources for PDBs, such as CPU, memory, I/O rate, and storage systems

CDB Lead and CDB Members

- The CDB lead in a fleet is the CDB from which you perform operations across the fleet.
- The CDB members of the fleet link to the CDB lead through a database link.



1

```
CONNECT / AS SYSDBA
ALTER DATABASE SET lead_cdb = TRUE;
```

2

```
GRANT sysoper, … TO system
      CONTAINER = ALL;
```

3

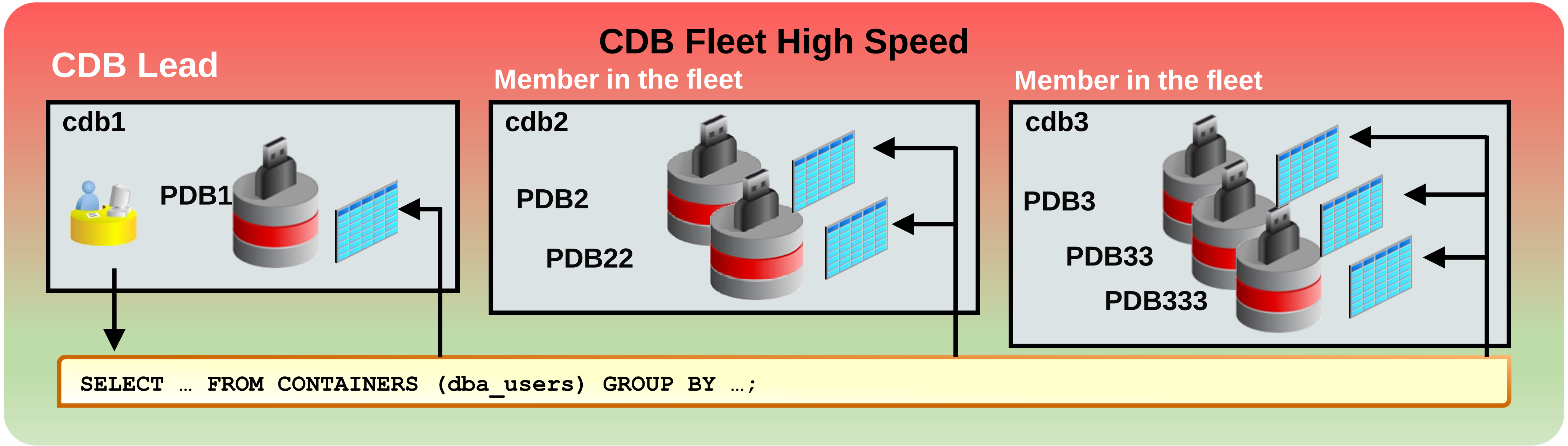
```
CONNECT / AS SYSDBA
CREATE PUBLIC DATABASE LINK lcdb1
      CONNECT TO system
      IDENTIFIED BY pass
      USING 'cdb1';
ALTER DATABASE SET lead_cdb_uri='dblink:lcdb1';
```

4

Same sequence of statements

Use Cases

- Monitoring and collecting diagnostic information across CDBs from the lead CDB
- Querying Oracle-supplied objects, such as DBA views, in different PDBs across the CDB fleet



```
SELECT … FROM CONTAINERS (dba_users) GROUP BY …;
```

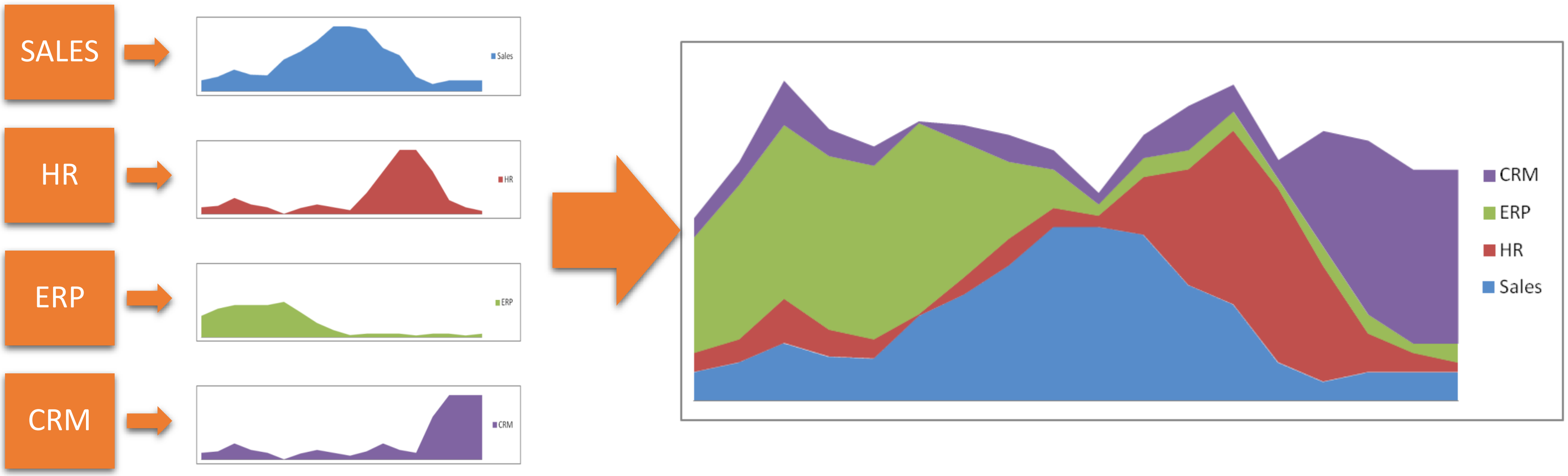
- Serving as a central location where you can view information about and the status of all the PDBs across multiple CDBs

Consolidated Database Replay Use Cases

- Perform realistic load testing for scenarios including:
  - Consolidation of servers
  - Workload scale-up
  - Test peak load capacity

Use Cases: Source Workloads

Workload: Each application has distinct peaks at different times of the workday.



SALES

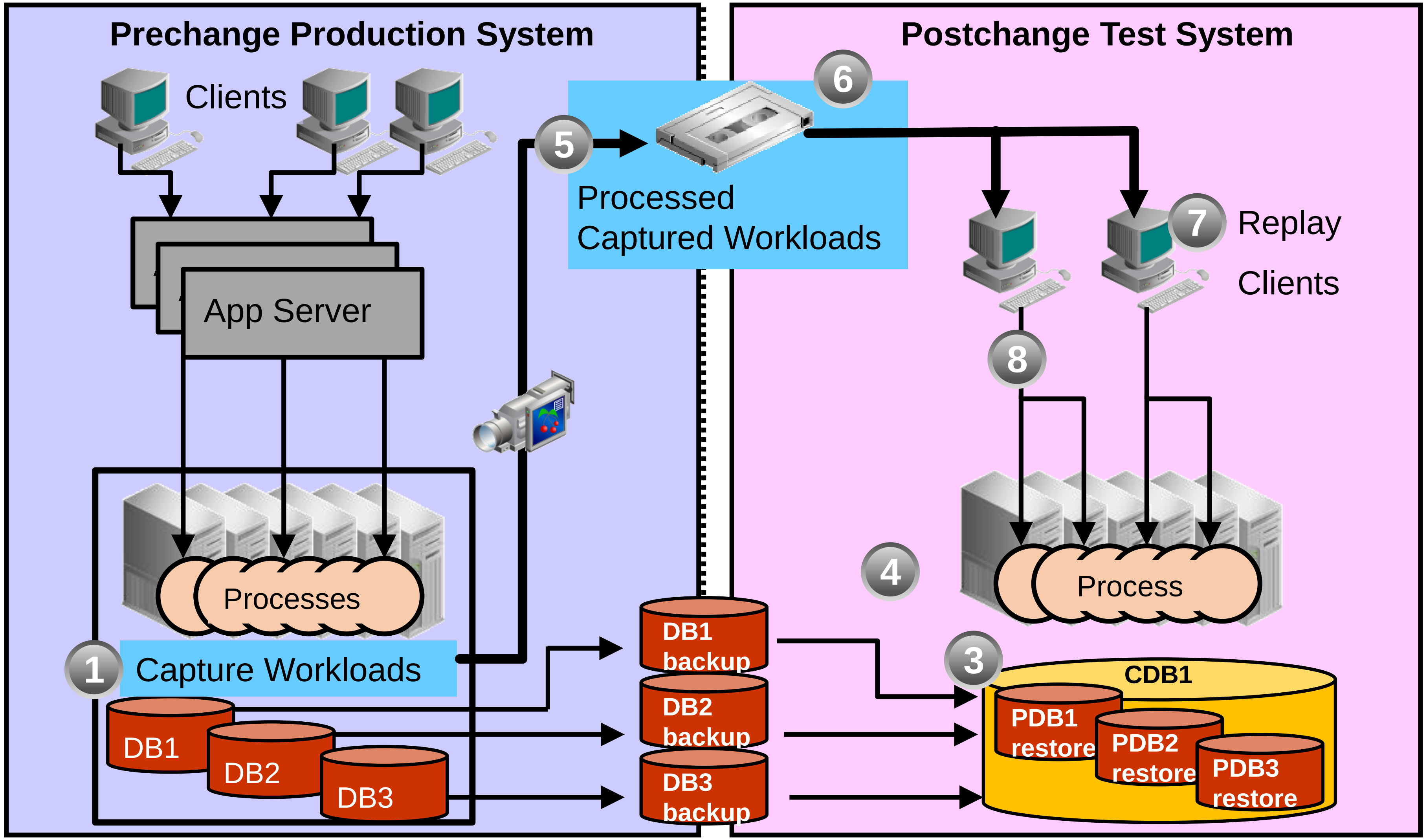
HR

ERP

CRM

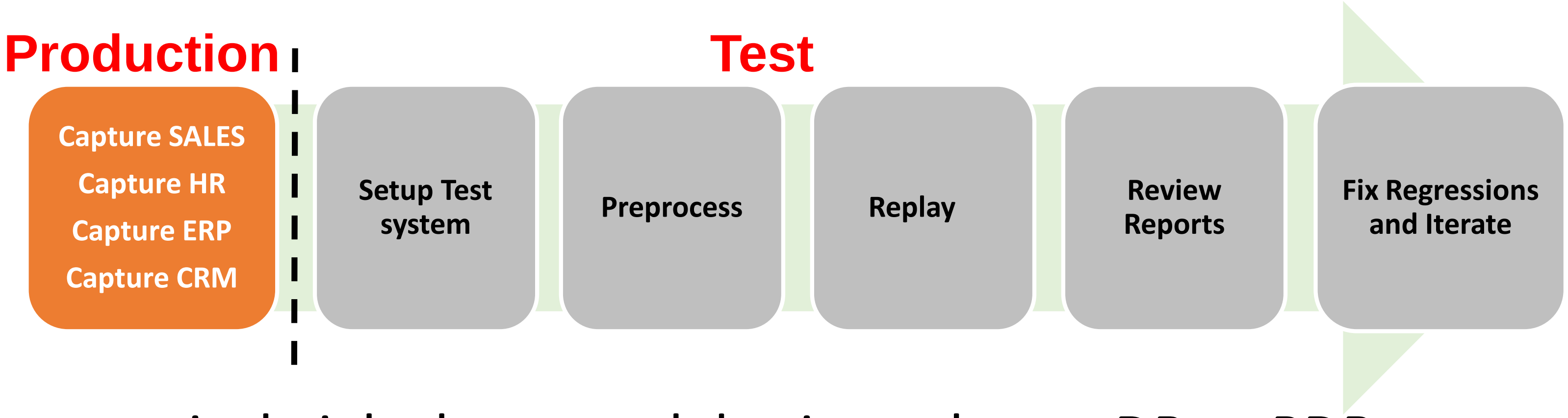
- Each workload captured on different databases and being consolidated is independent of the other.
- Allows multiple workloads captured from different non-CDBs or PDBs to be replayed concurrently in a single CDB.

The Big Picture



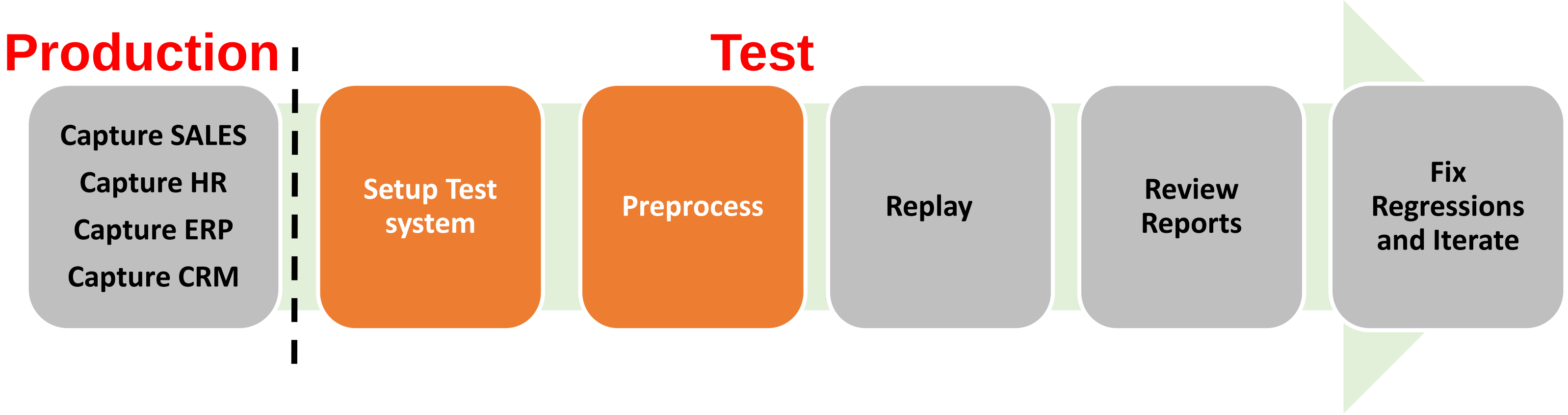
Step 1

- Consolidate multiple non-CDBs or PDBs on a single CDB.



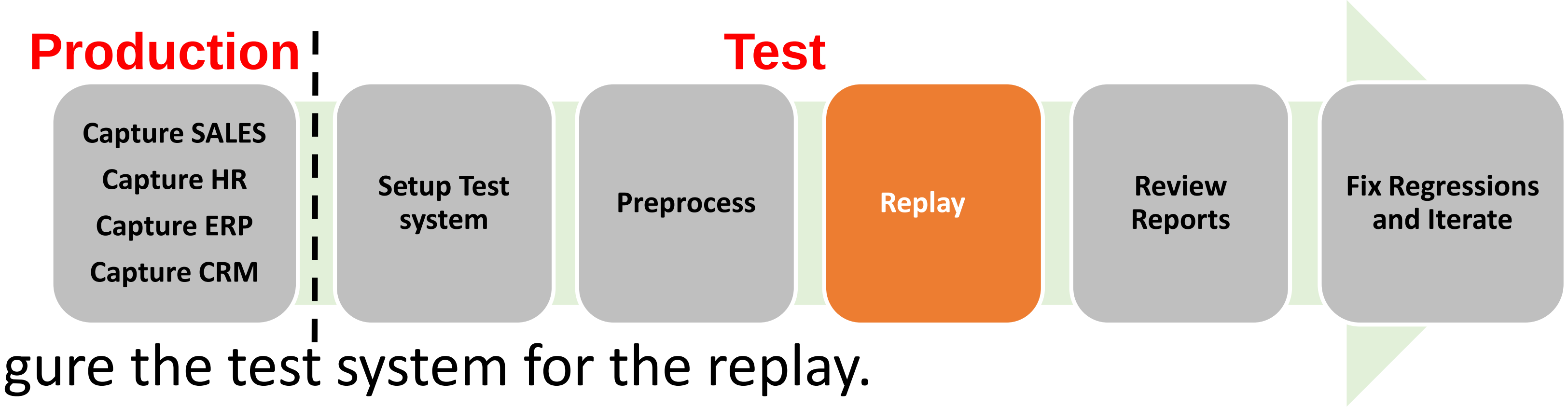
  - Capture a typical eight-hour workday in each non-DB or PDB.
  - Export performance data from each DB (AWR, SQL Tuning Sets).

Step 2



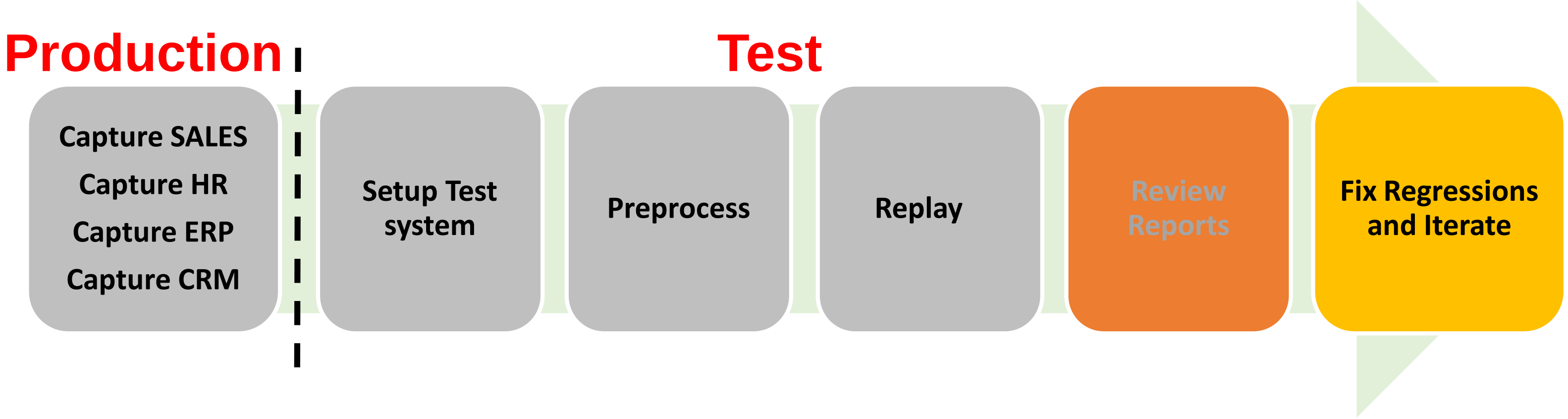
- Move all capture files from production to staging system.
- Restore non-CDBs as PDBs or PDBs to the start conditions for each capture being replayed on the single CDB.
- Process each workload (needed once).
- Replay each workload in isolation on new hardware to obtain baseline and verify suitability.
- Flashback/restore.

Step 3



- Configure the test system for the replay.
  - Add each workload by using `ADD_CAPTURE()`.
  - Initialize consolidated replay and remap all connections.
    - Optionally, remap schema users.
    - Synchronize on “Object_ID”.
  - Calibrate the `wrc` processes.
- Replay the workload on the restored database.
  - Two objects with the same name in different PDBs will be different Object IDs and will not collide during replay.

Step 4



Production

Test

Capture SALES
Capture HR
Capture ERP
Capture CRM

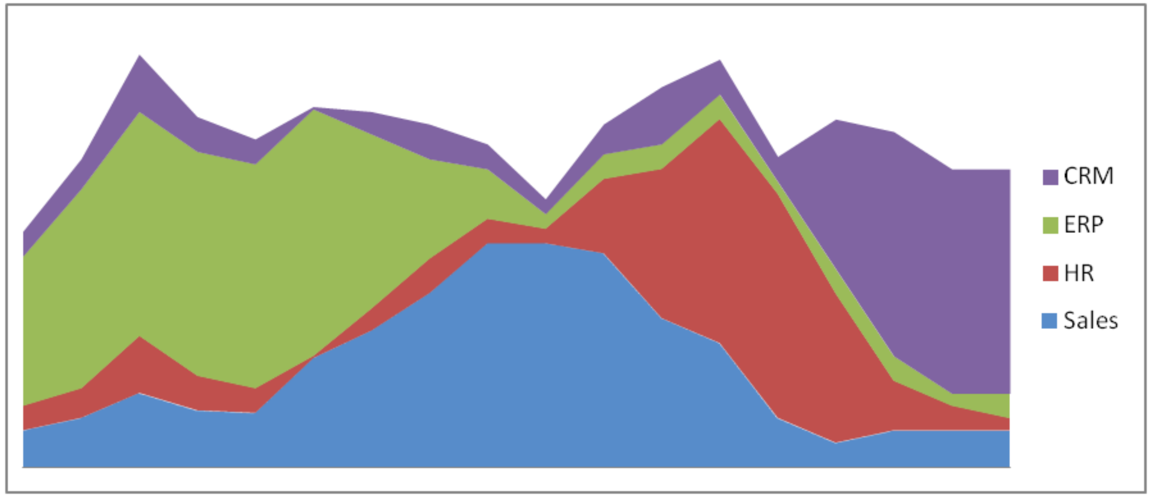
Setup Test system

Preprocess

Replay

Review Reports

Fix Regressions and Iterate



- Verify:
  - Quality of service metrics as perceived by application user per workload
  - Overall capacity
- Only replay can provide definitive answers.

Summary

- In this lesson, you should have learned how to:
  - Monitor performance in a CDB and PDBs
  - Manage SGA and PGA limits at the PDB level
  - Manage AWR snapshots at the CDB and PDB levels
  - Run ADDM tasks for CDB and PDB recommendations
  - Manage application shared object statistics
  - Control query DOP involving the `containers()` construct
  - Manage Heat Map and ADO policy declaration in a PDB
  - Manage a CDB fleet
  - Describe use cases for Consolidated Database Replay

Practice 10: Overview

- 10-1: Monitoring performance at the CDB and PDB levels
- 10-2: Getting performance ADDM recommendations at CDB and PDB levels
- 10-3: Monitoring and tuning SQL executions at the PDB level
- 10-4: Configuring a CDB fleet with its CDB lead and CDB members