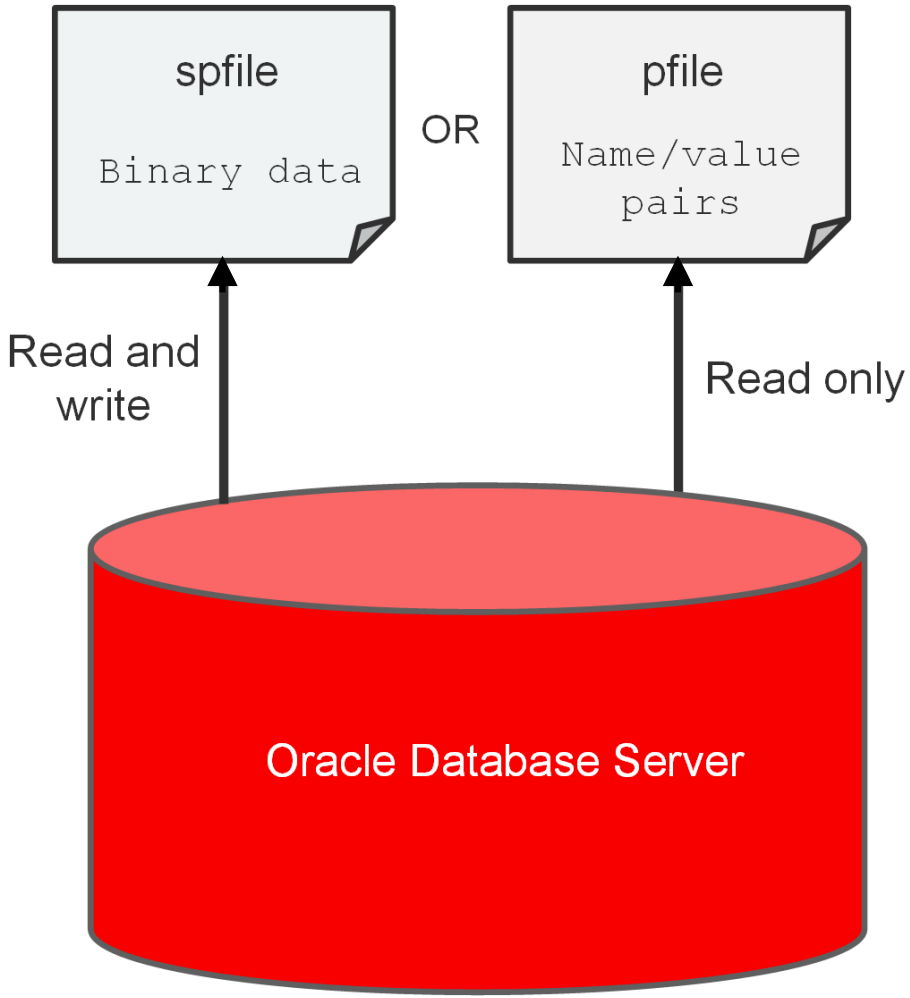
Managing Database Instances

Objectives

- After completing this lesson, you should be able to:
  - Describe initialization parameter files and initialization parameters
  - View and modify initialization parameters in SQL*Plus
  - Start up and shut down Oracle databases
  - Open and close PDBs
  - Work with the Automatic Diagnostic Repository (ADR)
  - Query dynamic performance views

Working with Initialization Parameters



Initialization Parameters

- Initialization parameters (parameters):
  - Set database limits
  - Set database-wide defaults
  - Specify files and directories
  - Affect performance
- Parameters can be of two types: basic or advanced.
  - Tune around 30 basic parameters to get reasonable database performance.
  - Example of a basic parameter: `SGA_TARGET`
  - Example of an advanced parameter: `DB_CACHE_SIZE`
- Derived parameters calculate their values from the values of other parameters.
  - Example: `SESSIONS` is derived from `PROCESSES`.
- Some parameter values or value ranges depend on the host operating system.
  - Example: `DB_BLOCK_SIZE`

Modifying Initialization Parameters

- Modify parameters to set capacity limits or improve performance.
  - Use EM Express or SQL*Plus (`ALTER SESSION` or `ALTER SYSTEM`).
- Query `V$PARAMETER` for an initialization parameter to learn whether you can make:
  - Session-level changes (`ISSES_MODIFIABLE` column)
  - System-level changes (`ISSYS_MODIFIABLE` column)
  - PDB-level changes (`ISPDB_MODIFIABLE` column)
- Use the `SCOPE` clause with the `ALTER SYSTEM` command to tell the system where to update the system-level parameter:
  - `MEMORY`
  - `SPFILE`
  - `BOTH`
- Use the `DEFERRED` keyword to set or modify the value of the parameter for future sessions that connect to the database.

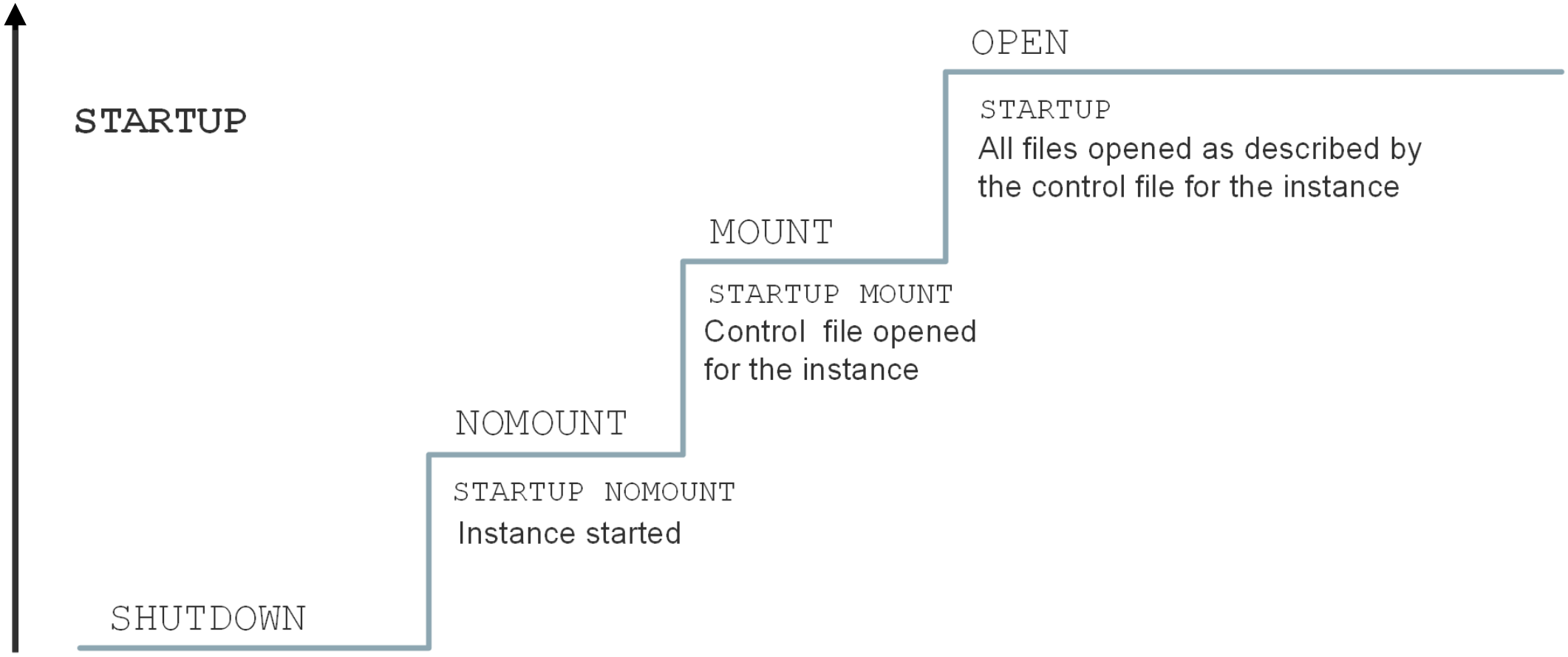
Viewing Initialization Parameters

- Ways to view initialization parameters in SQL*Plus:
  - Issue the `SHOW PARAMETER` command.
    - Example: Find out about all the parameters whose names contain the word “para.”

```
SQL> SHOW PARAMETER para
```

  - Query the following views:
    - `V$PARAMETER`
    - `V$PARAMETER2`
    - `V$SPPARAMETER`
    - `V$SYSTEM_PARAMETER`
    - `V$SYSTEM_PARAMETER2`

Starting the Oracle Database Instance



Shutting Down an Oracle Database Instance

- Sometimes you need to shut down the database instance (for example, to change a static parameter or patch the database server).
- Use the `SHUTDOWN` command to shut down the database instance in various modes: `ABORT`, `IMMEDIATE`, `TRANSACTIONAL`, and `NORMAL`.

	ABORT	**IMMEDIATE**	**TRANSACTIONAL**	**NORMAL**
Allows new connections	No	No	No	No
Waits until current sessions end	No	No	No	Yes
Waits until current transactions end	No	No	Yes	Yes
Forces a checkpoint and closes files	No	Yes	Yes	Yes

Comparing SHUTDOWN Modes

On the way down:

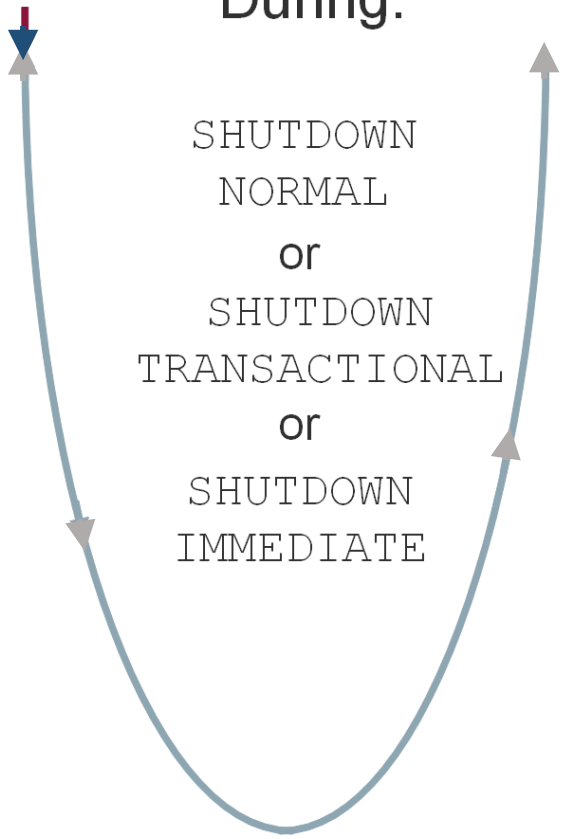
- Uncommitted changes rolled back, for IMMEDIATE
- Database buffer cache written to data files
- Resources released

During:

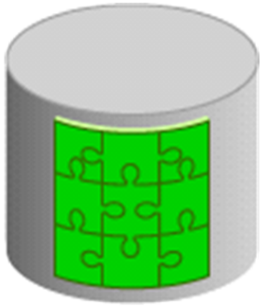
SHUTDOWN NORMAL
or
SHUTDOWN TRANSACTIONAL
or
SHUTDOWN IMMEDIATE

On the way up:

- No instance recovery



Consistent database



Comparing SHUTDOWN Modes

On the way down:

- Modified buffers not written to data files
- Uncommitted changes not rolled back

During:

SHUTDOWN ABORT
or
Instance failure
or
STARTUP FORCE

Inconsistent database

On the way up:

- Online redo log files used to reapply changes
- Undo segments used to roll back uncommitted changes
- Resources released

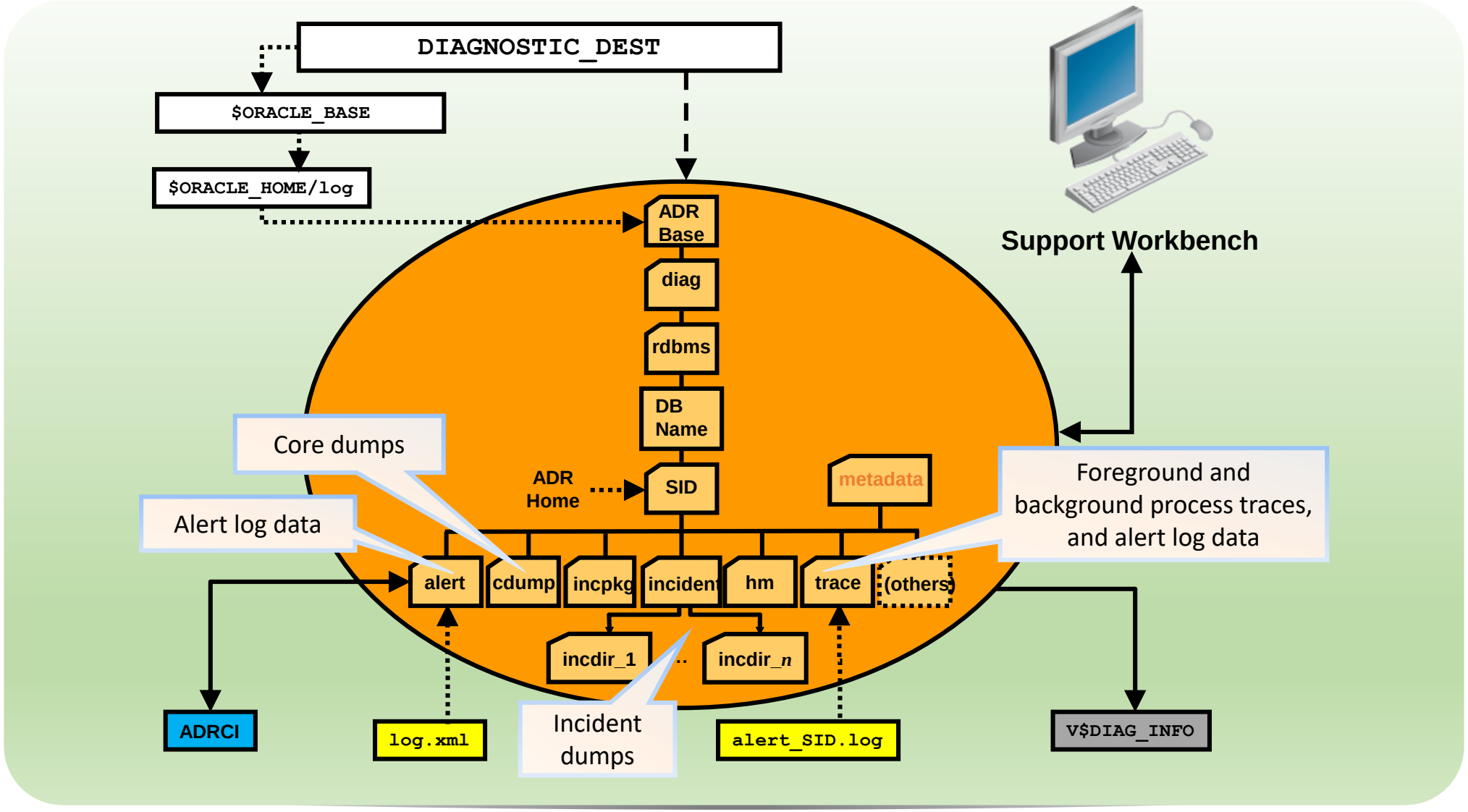
Opening and Closing PDBs

- Open/close a PDB to open/close its data files.
- A PDB has four open modes:
  - `READ WRITE` (the PDB is fully started/opened)
  - `READ ONLY`
  - `MIGRATE`
  - `MOUNTED` (the PDB is shut down/closed)
- Use the `ALTER PLUGGABLE DATABASE` command or `STARTUP` and `SHUTDOWN` commands to open and close PDBs.
  - Example: `SQL> ALTER PLUGGABLE DATABASE PDB1 OPEN;`
- The `ALTER PLUGGABLE DATABASE` command lets you change from any open mode to another.
- To use the `STARTUP` command, the PDB must be in `MOUNTED` mode.

Working with the Automatic Diagnostic Repository

- The Automatic Diagnostic Repository (ADR):
  - Is a file-based repository outside the database
  - Is a system-wide central tracing and logging repository
  - Stores database diagnostic data such as:
    - Traces
    - Alert log
    - Health monitor reports

Automatic Diagnostic Repository



Viewing the Alert Log

- The alert log file is a chronological log of messages about the database instance and database, such as:
  - Any nondefault initialization parameters used at startup
  - All internal errors (`ORA-600`), block corruption errors (`ORA-1578`), and deadlock errors (`ORA-60`) that occurred
  - Administrative operations, such as the SQL statements `CREATE`, `ALTER`, `DROP DATABASE`, and `TABLESPACE`, and the Enterprise Manager or SQL*Plus statements `STARTUP`, `SHUTDOWN`, `ARCHIVE LOG`, and `RECOVER`
  - Several messages and errors relating to the functions of shared server and dispatcher processes
  - Errors during the automatic refresh of a materialized view
- Query `V$DIAG_INFO` to find the location of the alert log.
  - The path to `alert_SID.log` corresponds to the Diag Trace entry.
  - The path to `log.xml` corresponds to the Diag Alert entry.
- You can view the alert log in a text editor or in ADRCI.

Using Trace Files

- Trace files contain:
  - Error information (contact Oracle Support Services if an internal error occurs)
  - Information that can provide guidance for tuning applications or an instance
- Each server and background process can write to an associated trace file.
- Trace file names for background processes are named after their processes.
  - Exception: Trace files generated by job queue processes
- Oracle Database includes an advanced fault diagnosability infrastructure for preventing, detecting, diagnosing, and resolving problems.
- When a critical error occurs:
  - An incident number is assigned to the error
  - Diagnostic data for the error (such as trace files) is immediately captured and tagged with the incident number
  - Data is stored in the ADR
- ADR files can be automatically purged by setting retention policy parameters.

Administering the DDL Log File

- Enable the capture of certain DDL statements to a DDL log file by setting `ENABLE_DDL_LOGGING` to `TRUE`.
- The DDL log contains one log record for each DDL statement.
- Two DDL logs containing the same information:
  - XML DDL log: `log.xml` written to `$ORACLE_BASE/diag/rdbms/<dbname>/<SID>/log/ddl`
  - Text DDL: `ddl_<sid>.log` written to `$ORACLE_BASE/diag/rdbms/<dbname>/<SID>/log`
- Example:

```
$ more ddl_orcl.log
Thu Nov 15 08:35:47 2016
diag_adl:drop user app_user
```

Querying Dynamic Performance Views

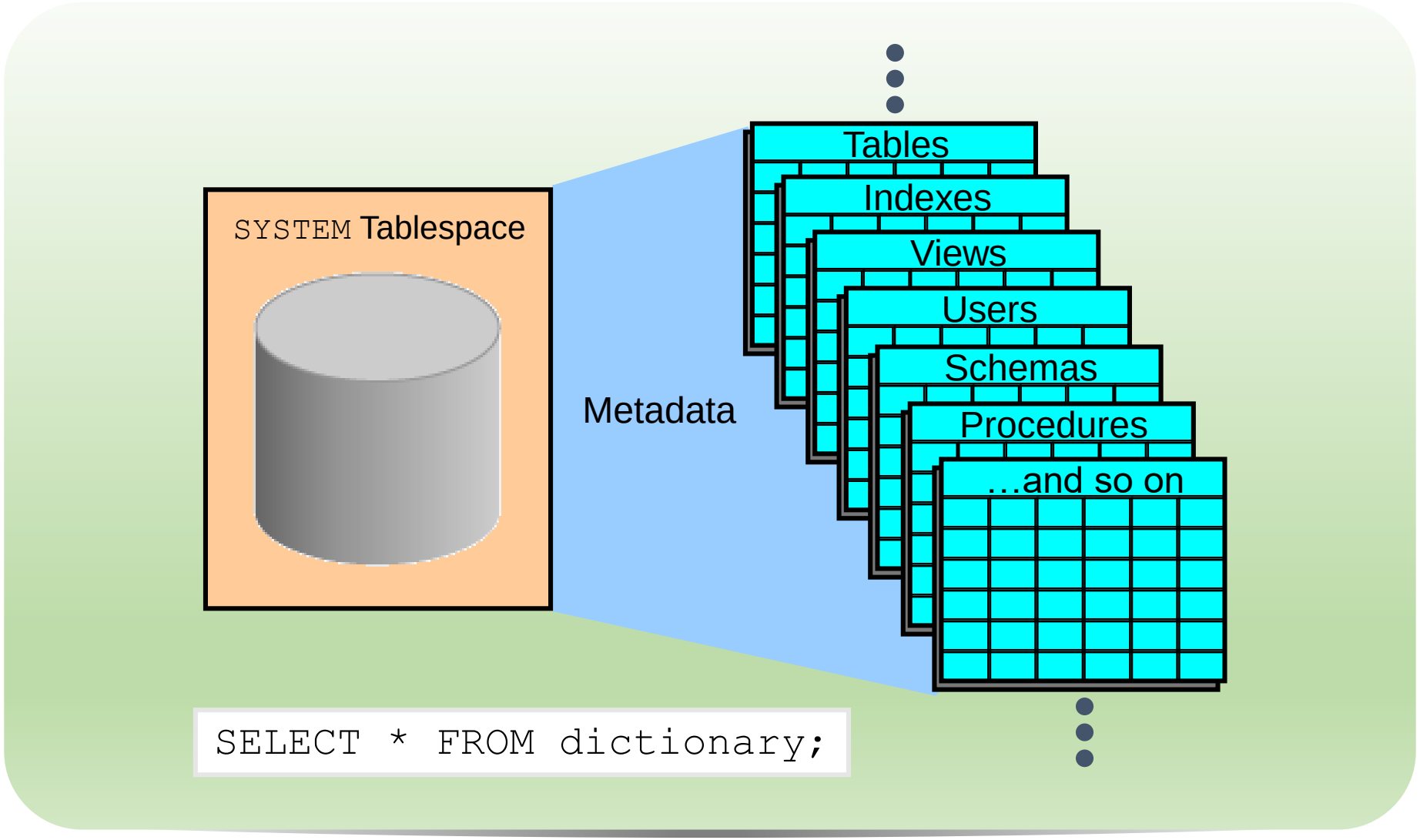
- Dynamic performance views provide access to information about the changing states of instance memory structures:
  - Sessions, file states, and locks
  - Progress of jobs and tasks
  - Backup status, memory usage, and allocation
  - System and session parameters
  - SQL execution
  - Statistics and metrics
- Dynamic performance views start with the prefix `V$`.
- Example query: Which current sessions have logged in from the EDXX9P1 computer on the last day?

```
SQL> SELECT * FROM V$SESSION
  2  WHERE machine = 'EDXX9P1'
  3  AND logon_time > SYSDATE - 1;
```

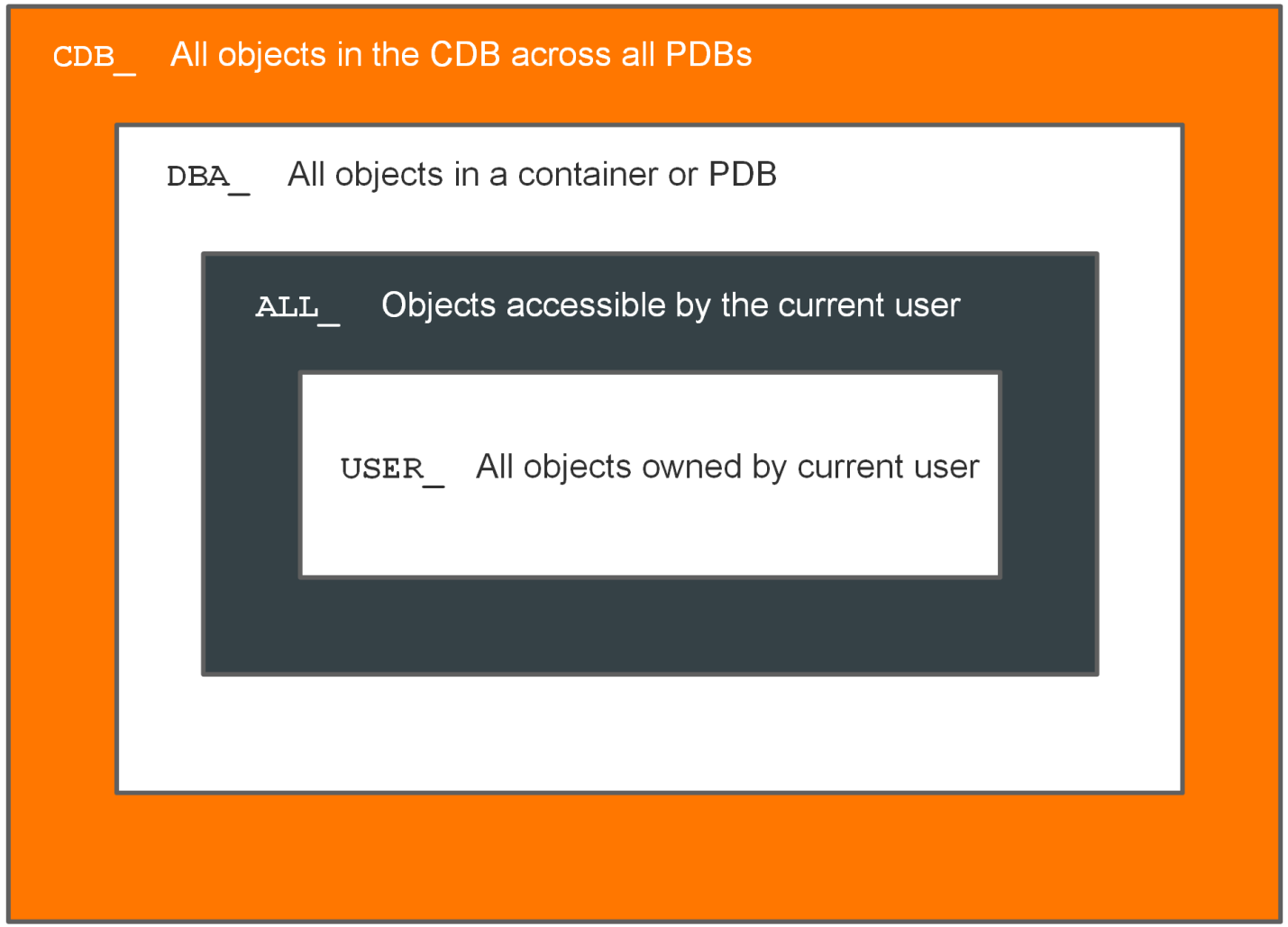
Considerations for Dynamic Performance Views

- These views are owned by the `SYS` user.
- Views provide information depending on the stage (`NOMOUNT`, `MOUNT`, or `OPEN`).
- You can query `V$FIXED_TABLE` to see all the view names.
- These views are often referred to as “v-dollar views.”
- Read consistency is not guaranteed on these views because the data is dynamic.

Data Dictionary: Overview



Querying the Oracle Data Dictionary



Summary

- In this lesson, you should have learned how to:
  - Describe initialization parameter files and initialization parameters
  - View and modify initialization parameters in SQL*Plus
  - Start up and shut down Oracle databases
  - Open and close PDBs
  - Work with the Automatic Diagnostic Repository (ADR)
  - Query dynamic performance views

Practice 7: Overview

- 7-1: Investigating Initialization Parameter Files
- 7-2: Viewing Initialization Parameters by Using SQL*Plus
- 7-3: Modifying Initialization Parameters by Using SQL*Plus
- 7-4: Modifying an Initialization Parameter by Using Enterprise Manager Database Express
- 7-5: Shutting Down and Starting Up the Oracle Database
- 7-6: Viewing Diagnostic Information