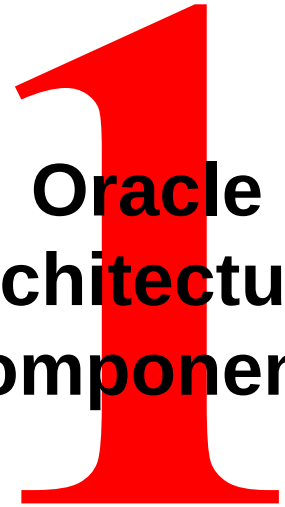


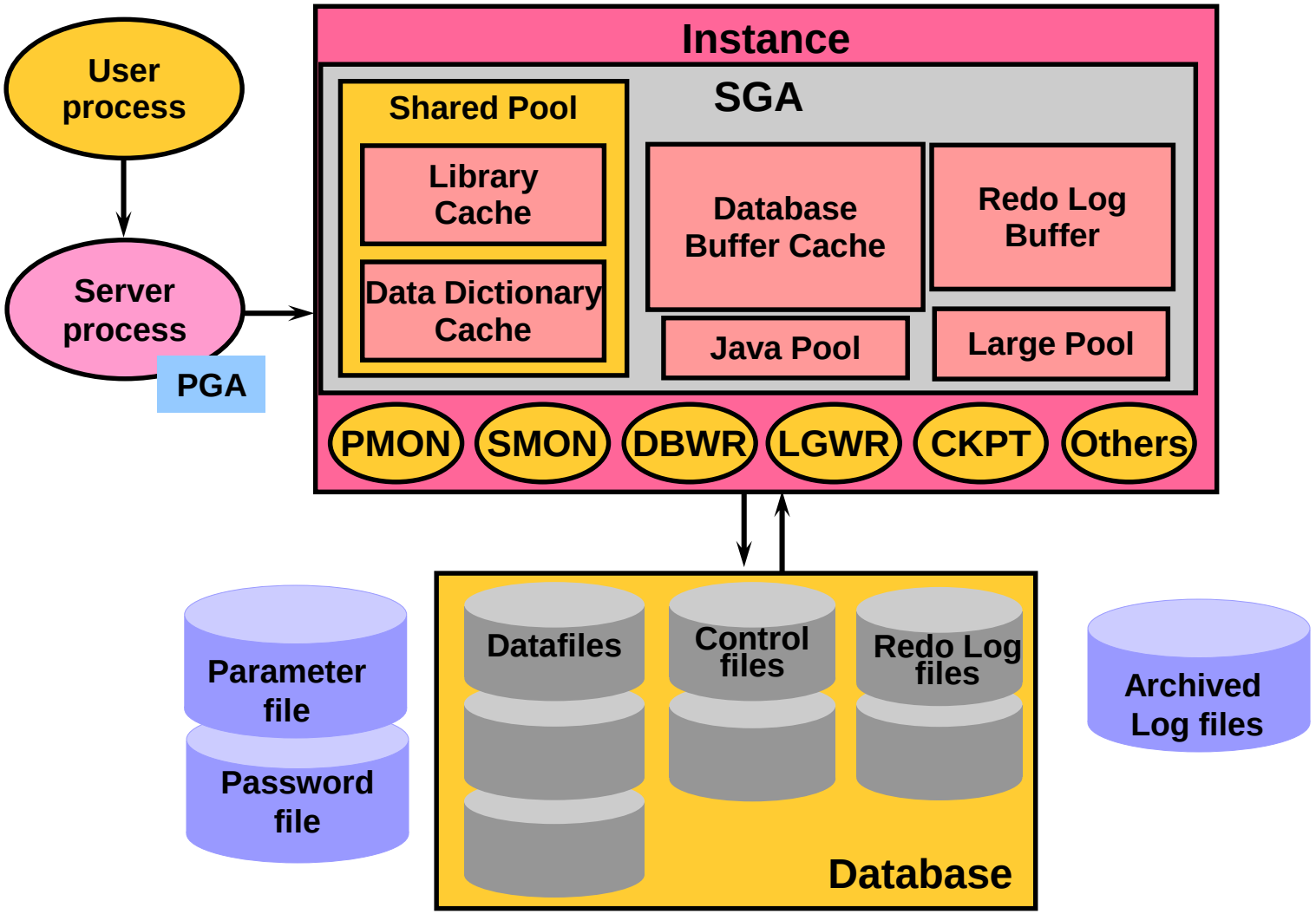
Oracle Architectural Components

Objectives

After completing this lesson, you should be able to do the following:

- Outline the Oracle architecture and its main components
- List the structures involved in connecting a user to an Oracle Instance

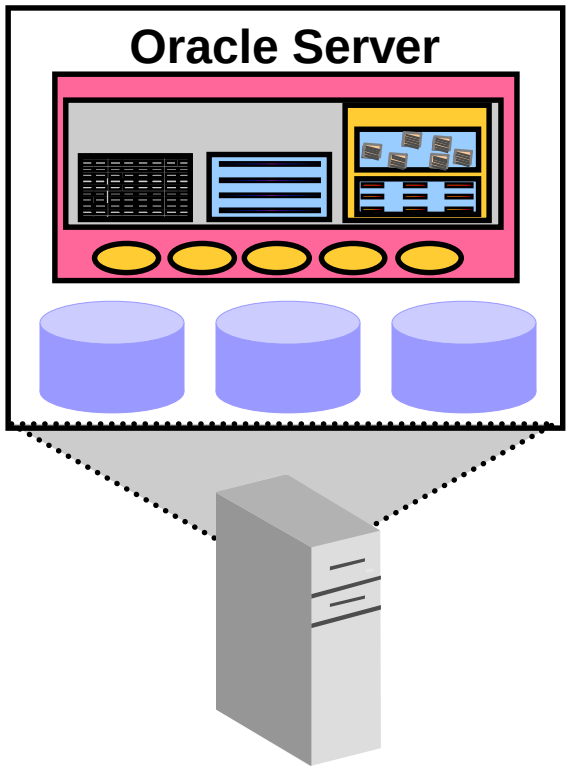
Overview of Primary Components



Oracle Server

An Oracle server:

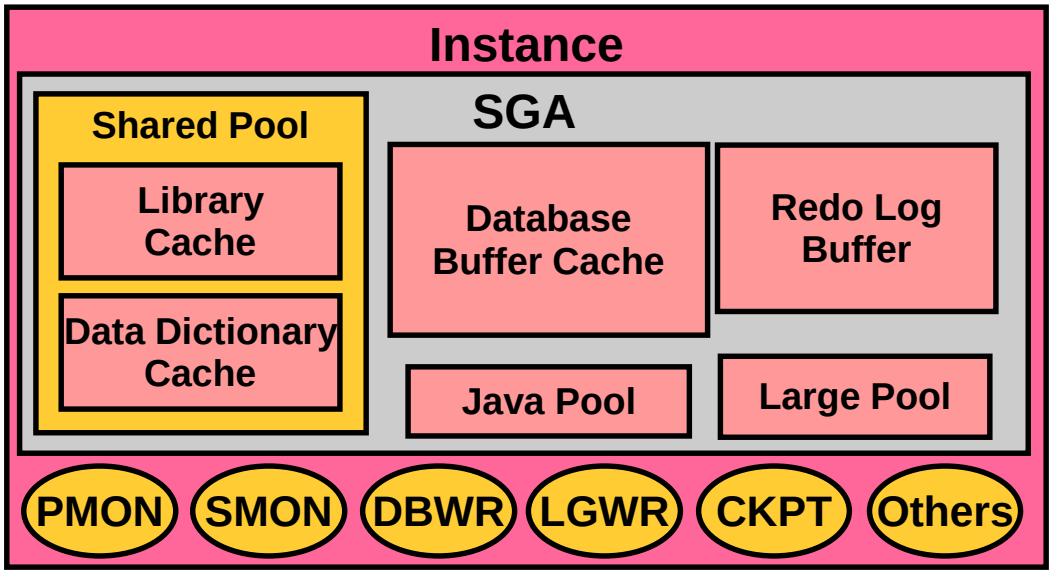
- Is a database management system that provides an open, comprehensive, integrated approach to information management
- **Consists of an Oracle Instance and an Oracle database**



Oracle Instance

An Oracle Instance:

- Is a means to access an Oracle database
- Always opens one and only one database
- Consists of memory and background process structures



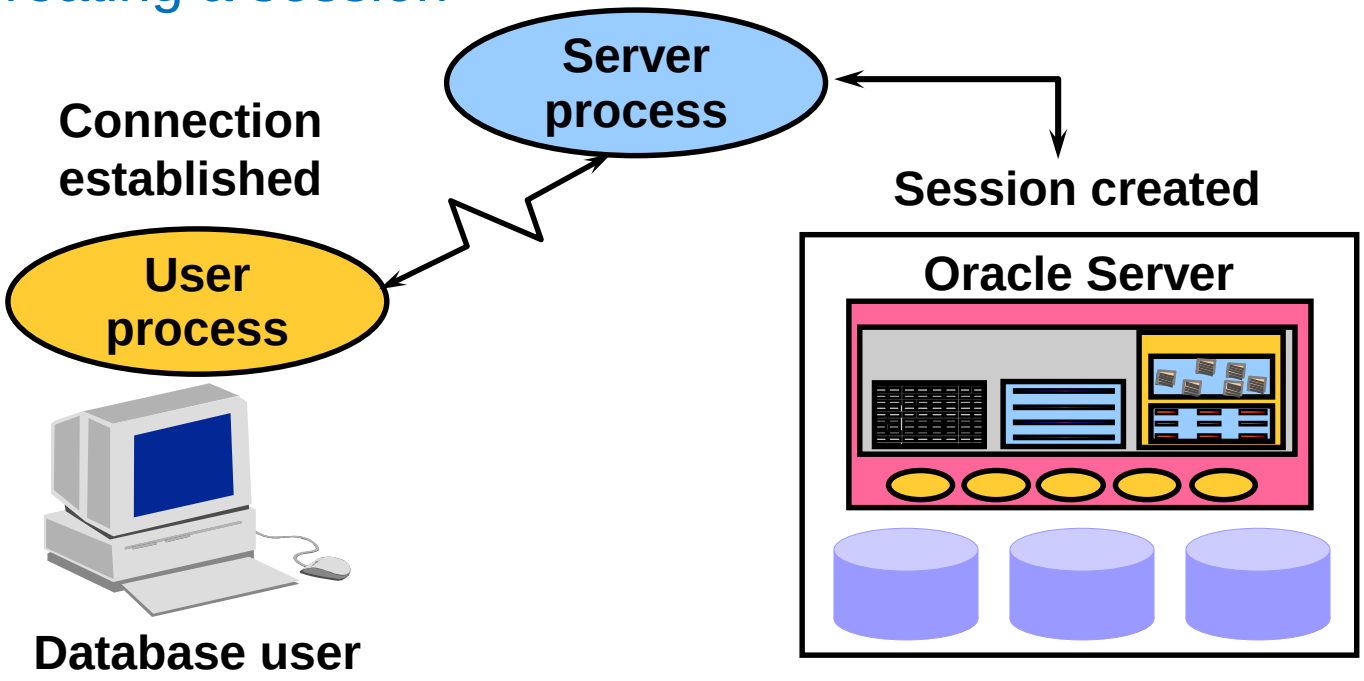
Memory structures

Background process structures

Establishing a Connection and Creating a Session

Connecting to an Oracle Instance:

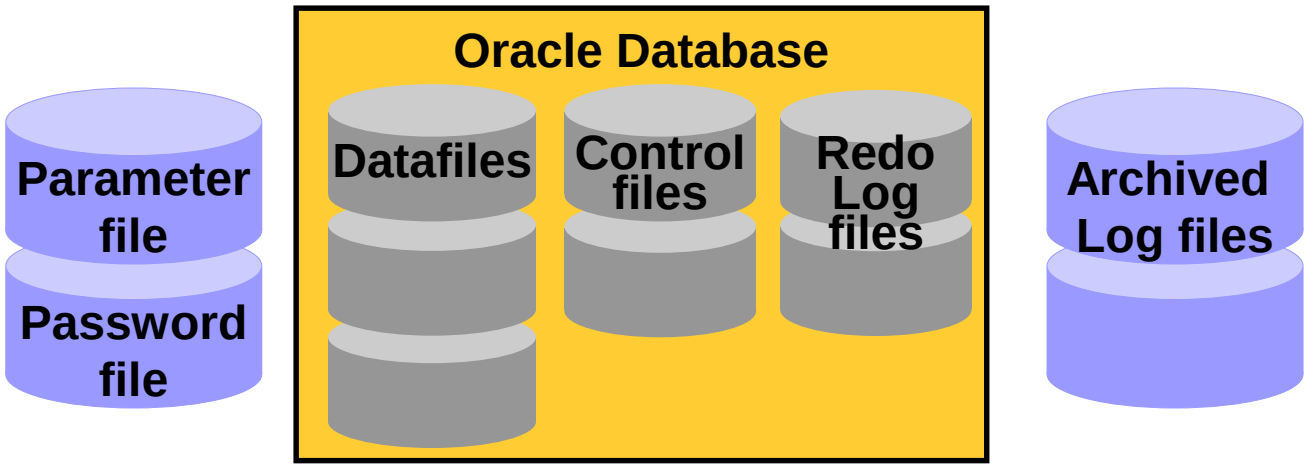
- Establishing a user connection
- Creating a session



Oracle Database

An Oracle database:

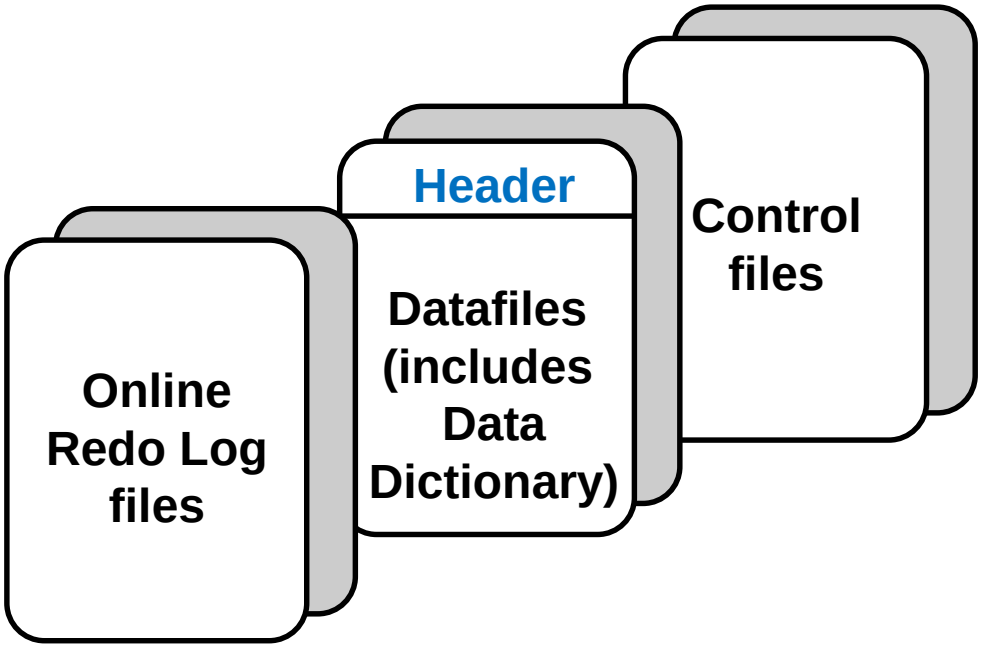
- Is a collection of data that is treated as a unit
- Consists of three file types



Physical Structure

The physical structure includes three types of files:

- Control files
- Datafiles
- Redo log files



Memory Structure

Oracle's memory structure consists of two memory areas known as:

- System Global Area (SGA): Allocated at instance startup, and is a fundamental component of an Oracle Instance
- Program Global Area (PGA): Allocated when the server process is started

System Global Area

- The SGA consists of several memory structures:
  - Shared Pool
  - Database Buffer Cache
  - Redo Log Buffer
  - Other structures (for example, lock and latch management, statistical data)
- There are two additional memory structures that can be configured within the SGA:
  - Large Pool
  - Java Pool

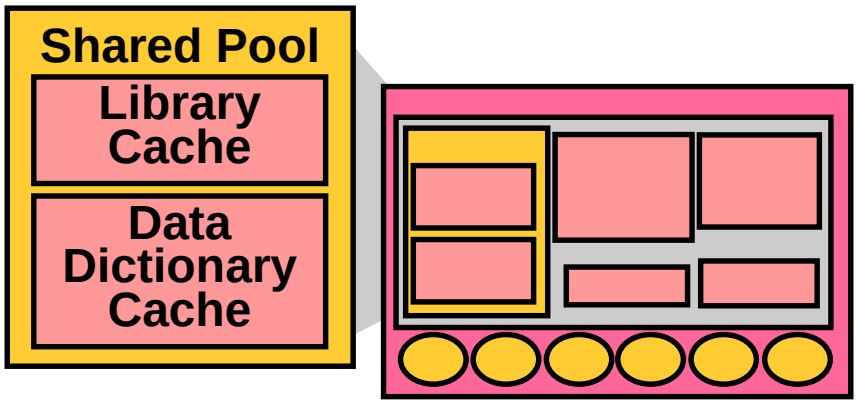
System Global Area

- SGA is dynamic
- Sized by the `SGA_MAX_SIZE` parameter
- Allocated and tracked in granules by SGA components
  - Contiguous virtual memory allocation
  - Granule size based on total estimated `SGA_MAX_SIZE`

Shared Pool

- Used to store:
  - Most recently executed SQL statements
  - Most recently used data definitions
- It consists of two key performance-related memory structures:
  - Library Cache
  - Data Dictionary Cache
- Sized by the parameter SHARED_POOL_SIZE

```
ALTER SYSTEM SET
SHARED_POOL_SIZE = 64M;
```



Library Cache

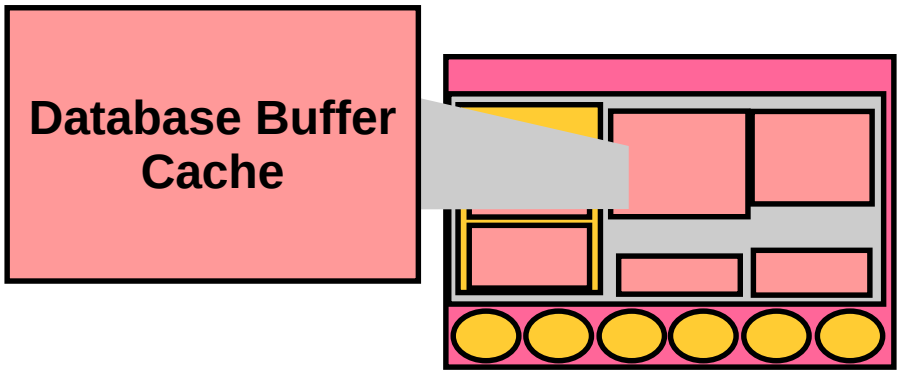
- Stores information about the most recently used SQL and PL/SQL statements
- Enables the sharing of commonly used statements
- Is managed by a least recently used (LRU) algorithm
- Consists of two structures:
  - Shared SQL area
  - Shared PL/SQL area
- Size determined by the Shared Pool sizing

Data Dictionary Cache

- A collection of the most recently used definitions in the database
- Includes information about database files, tables, indexes, columns, users, privileges, and other database objects
- During the parse phase, the server process looks at the data dictionary for information to resolve object names and validate access
- Caching data dictionary information into memory improves response time on queries and DML
- Size determined by the Shared Pool sizing

Database Buffer Cache

- Stores copies of data blocks that have been retrieved from the datafiles
- Enables great performance gains when you obtain and update data
- Managed through an LRU algorithm
- `DB_BLOCK_SIZE` determines primary block size



Database Buffer Cache

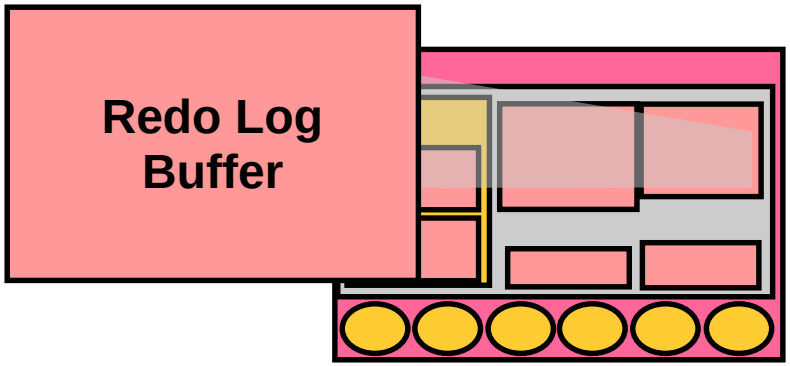
- Consists of independent sub-caches:
  - DB_CACHE_SIZE
  - DB_KEEP_CACHE_SIZE
  - DB_RECYCLE_CACHE_SIZE
- Can be dynamically resized

```
ALTER SYSTEM SET DB_CACHE_SIZE = 96M;
```

- DB_CACHE_ADVICE set to gather statistics for predicting different cache size behavior
- Statistics displayed by V$DB_CACHE_ADVICE

Redo Log Buffer

- Records all changes made to the database data blocks
- Primary purpose is recovery
- Changes recorded within are called redo entries
- Redo entries contain information to reconstruct or redo changes
- Size defined by `LOG_BUFFER`



Large Pool

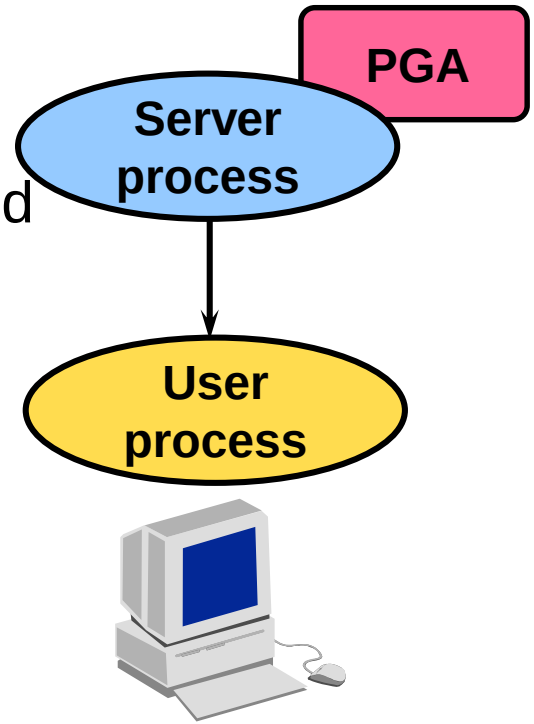
- An optional area of memory in the SGA
- Relieves the burden placed on the Shared Pool
- Used for:
  - Session memory (UGA) for the Shared Server
  - I/O server processes
  - Backup and restore operations or RMAN
  - Parallel execution message buffers
    - `PARALLEL_AUTOMATIC_TUNING` set to `TRUE`
- Does not use an LRU list
- Sized by `LARGE_POOL_SIZE`

Java Pool

- Services parsing requirements for Java commands
- Required if installing and using Java
- Sized by `JAVA_POOL_SIZE` parameter

Program Global Area

- Memory reserved for each user process connecting to an Oracle database
- Allocated when a process is created
- Deallocated when the process is terminated
- Used by only one process



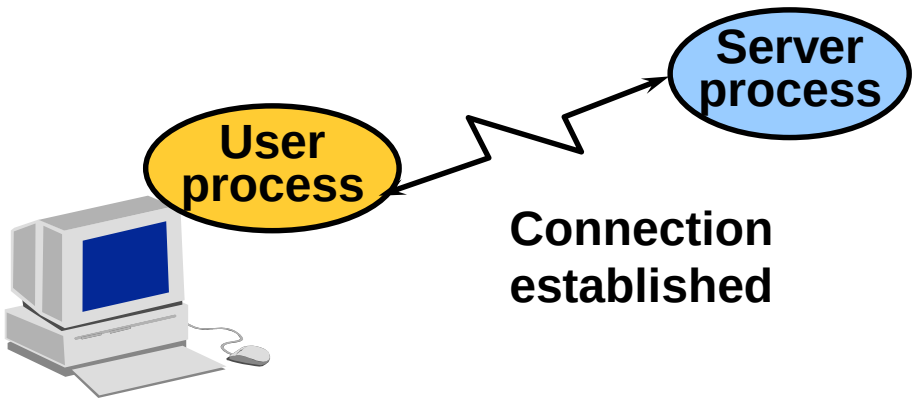
Process Structure

Oracle takes advantage of various types of processes:

- User process: Started at the time a database user requests connection to the Oracle server
- Server process: Connects to the Oracle Instance and is started when a user establishes a session
- Background processes: Started when an Oracle Instance is started

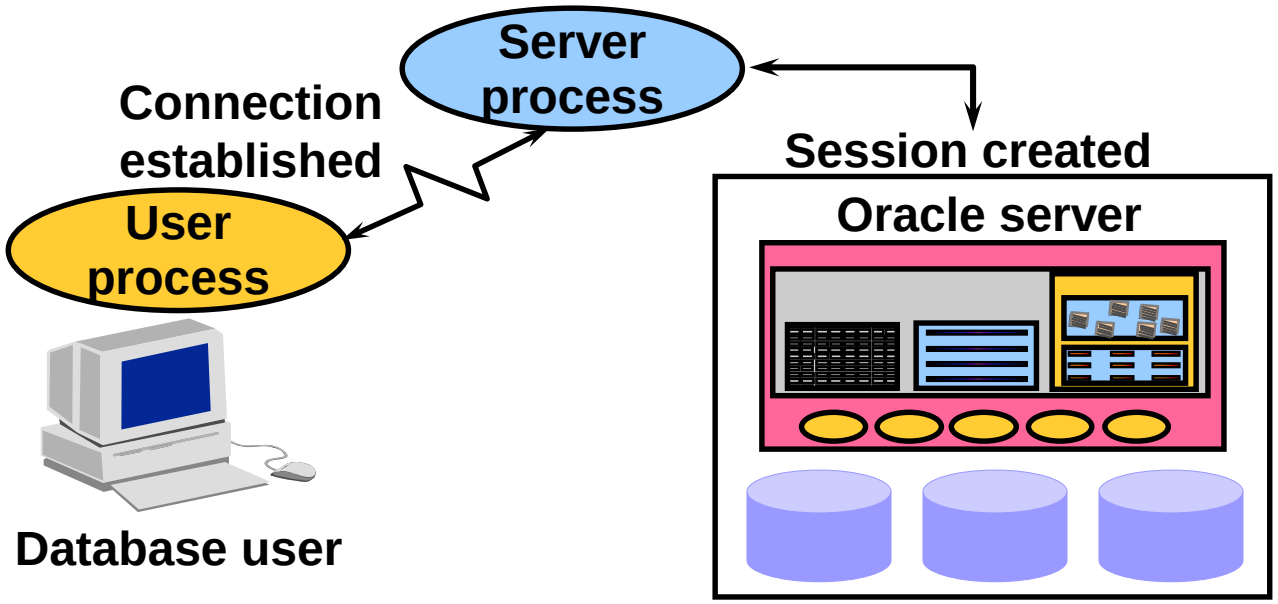
User Process

- A program that requests interaction with the Oracle server
- Must first establish a connection
- Does not interact directly with the Oracle server



Server Process

- A program that directly interacts with the Oracle server
- Fulfills calls generated and returns results
- Can be Dedicated or Shared Server

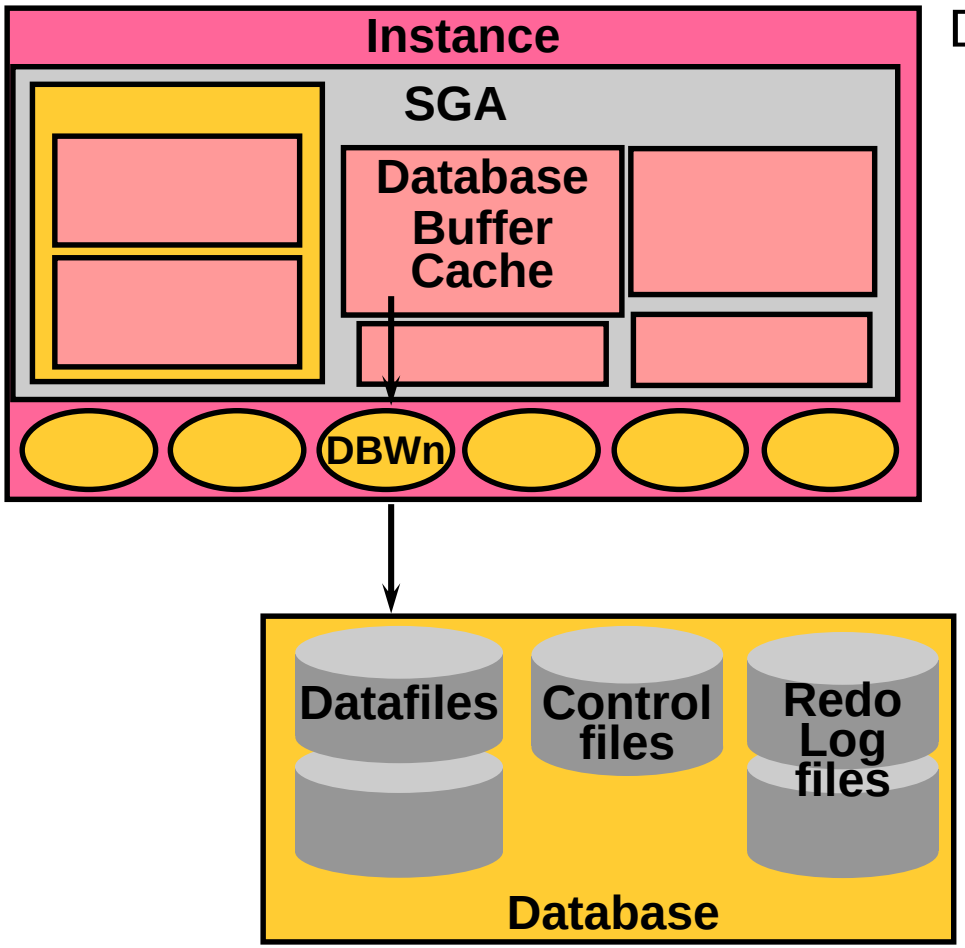


Background Processes

Maintains and enforces relationships between physical and memory structures

- Mandatory background processes:
  - DBWn PMON CKPT
  - LGWR SMON
- Optional background processes:
  - ARCn LMDn RECO
  - CJQ0 LMON Snnn
  - Dnnn Pnnn
  - LCKn QMNn

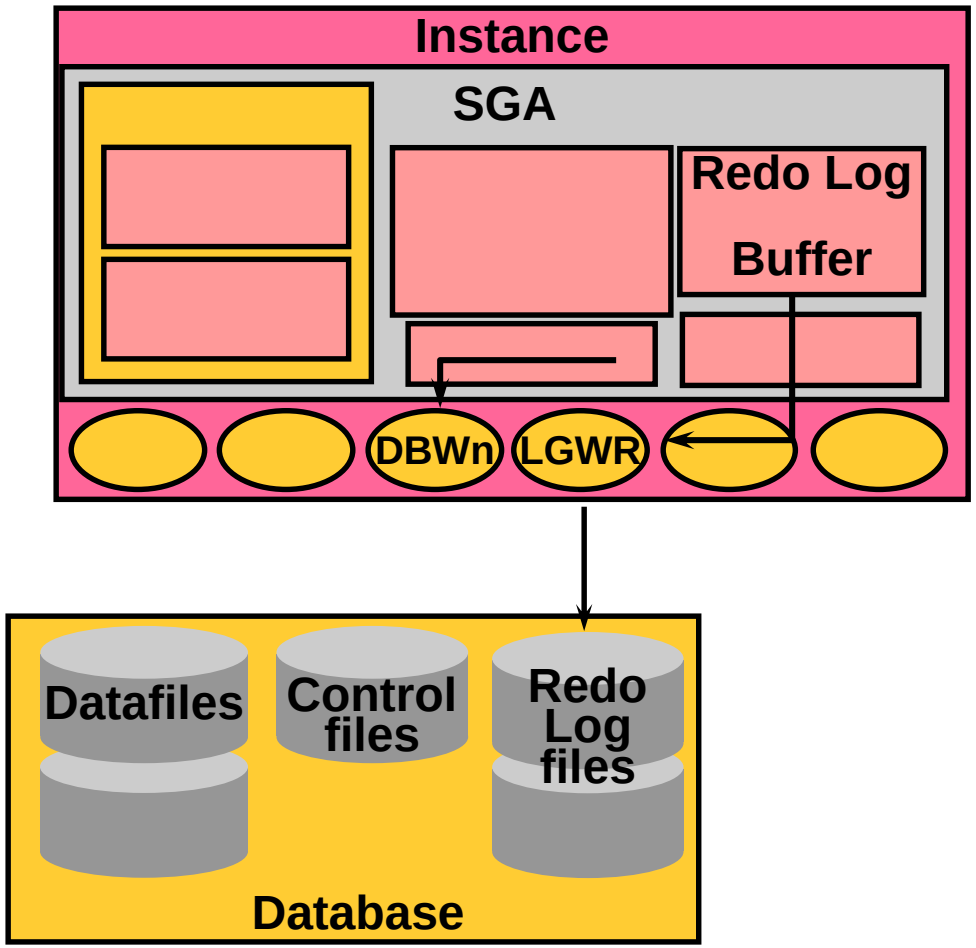
Database Writer (DBWn)



DBWn writes when:

- Checkpoint occurs
- Dirty buffers reach threshold
- There are no free buffers
- Timeout occurs
- RAC ping request is made
- Tablespace `OFFLINE`
- Tablespace `READ ONLY`
- Table `DROP` or `TRUNCATE`
- Tablespace `BEGIN BACKUP`

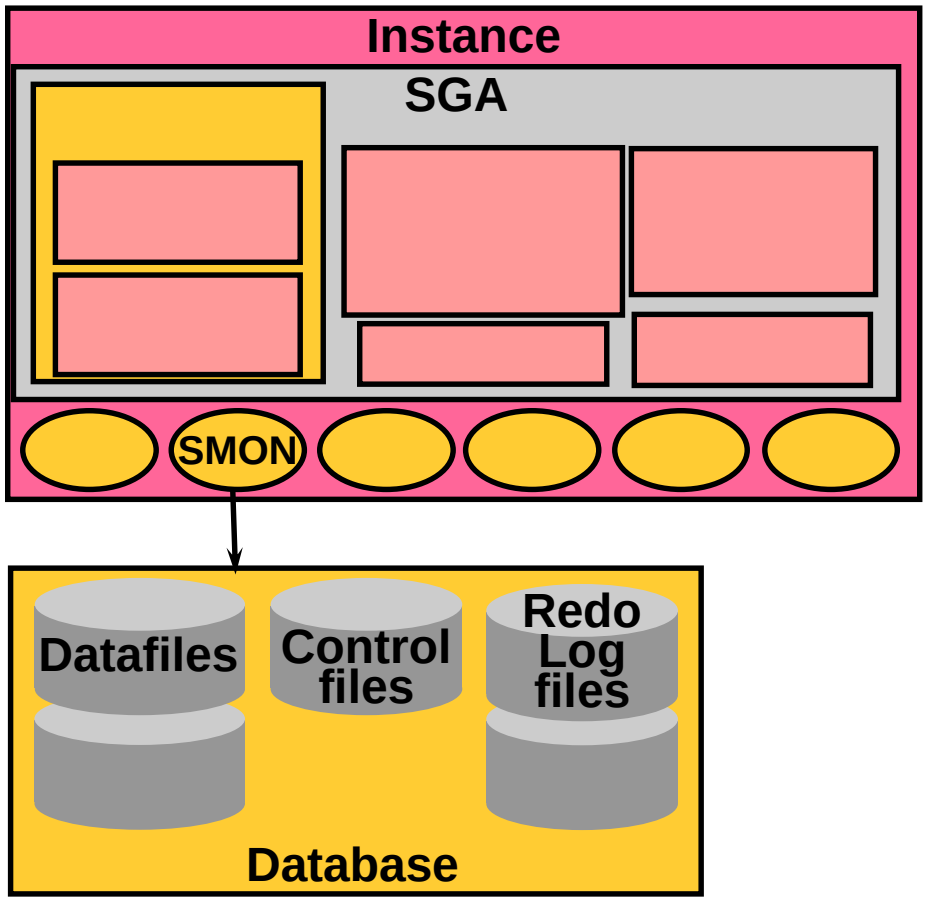
Log Writer (LGWR)



LGWR writes:

- At commit
- When one-third full
- When there is 1 MB of redo
- Every three seconds
- Before DBWn writes

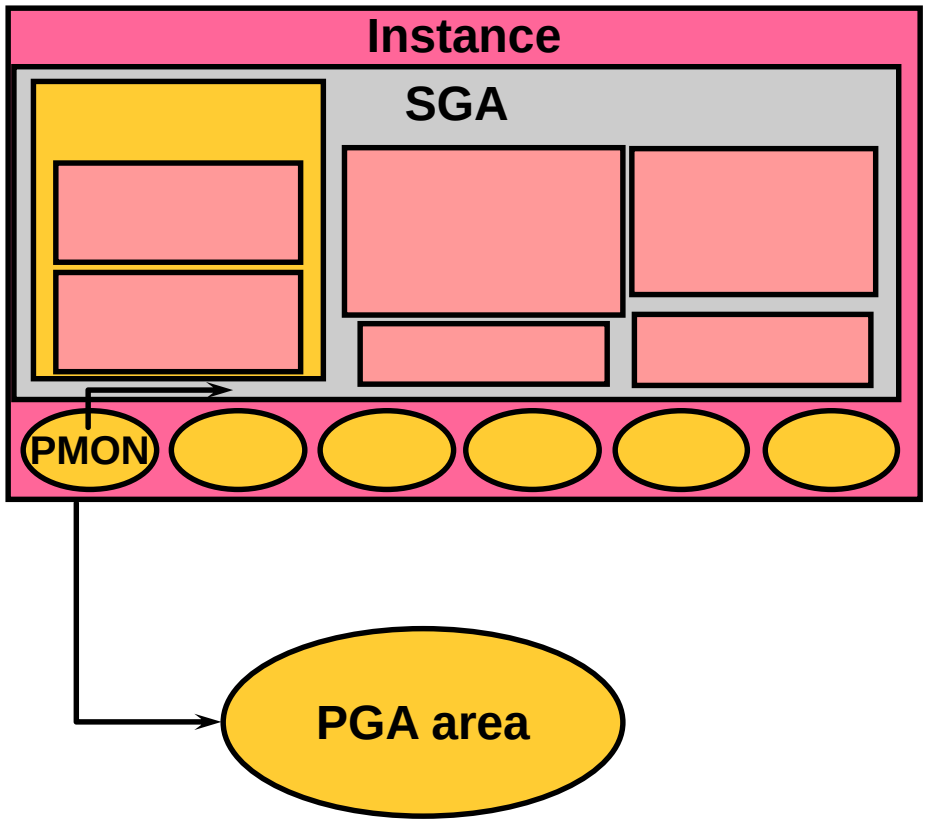
System Monitor (SMON)



Responsibilities:

- Instance recovery
  - Rolls forward changes in redo logs
  - Opens database for user access
  - Rolls back uncommitted transactions
- Coalesces free space
- Deallocates temporary segments

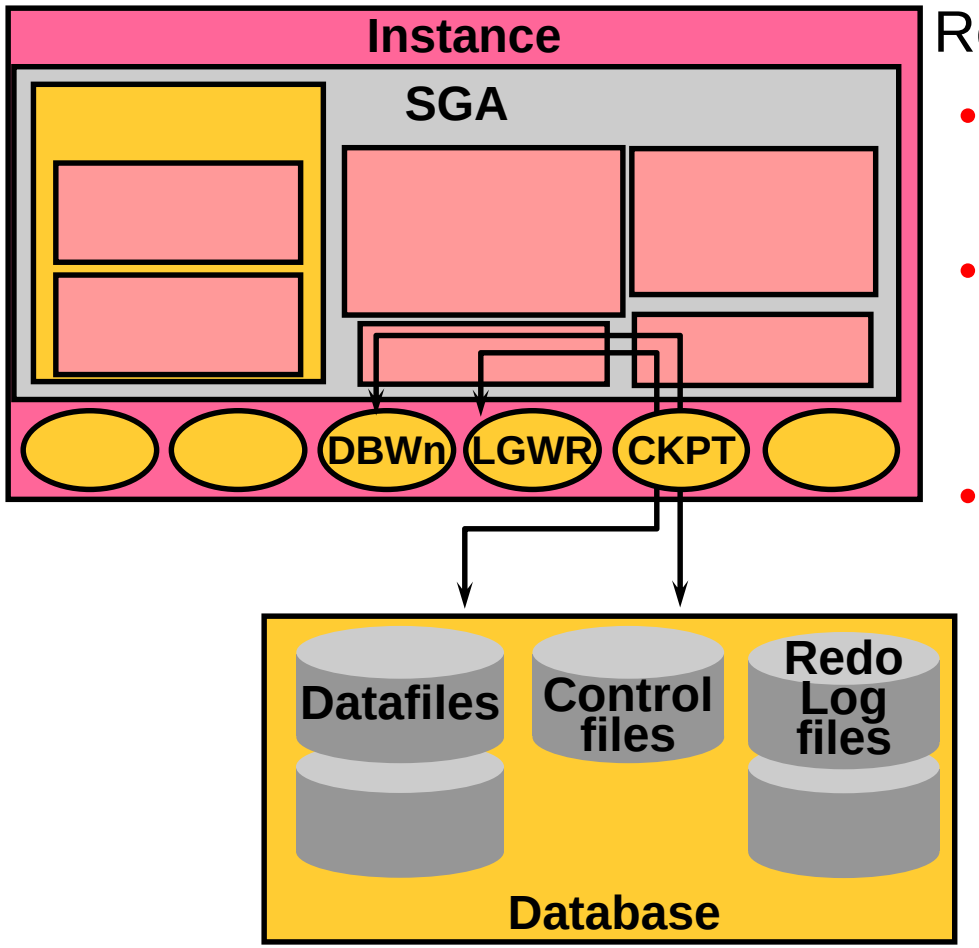
Process Monitor (PMON)



Cleans up after failed processes by:

- Rolling back the transaction
- Releasing locks
- Releasing other resources
- Restarting dead dispatchers

Checkpoint (CKPT)

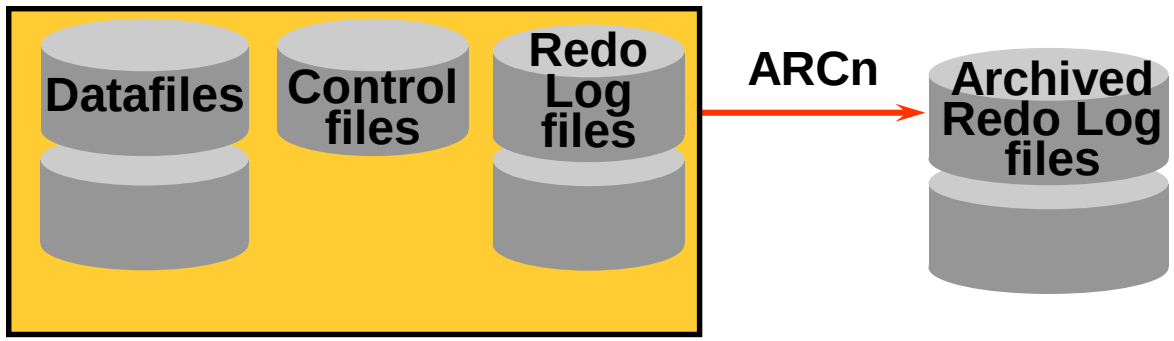


Responsible for:

- Signaling DBWn at checkpoints
- Updating datafile headers with checkpoint information
- Updating control files with checkpoint information

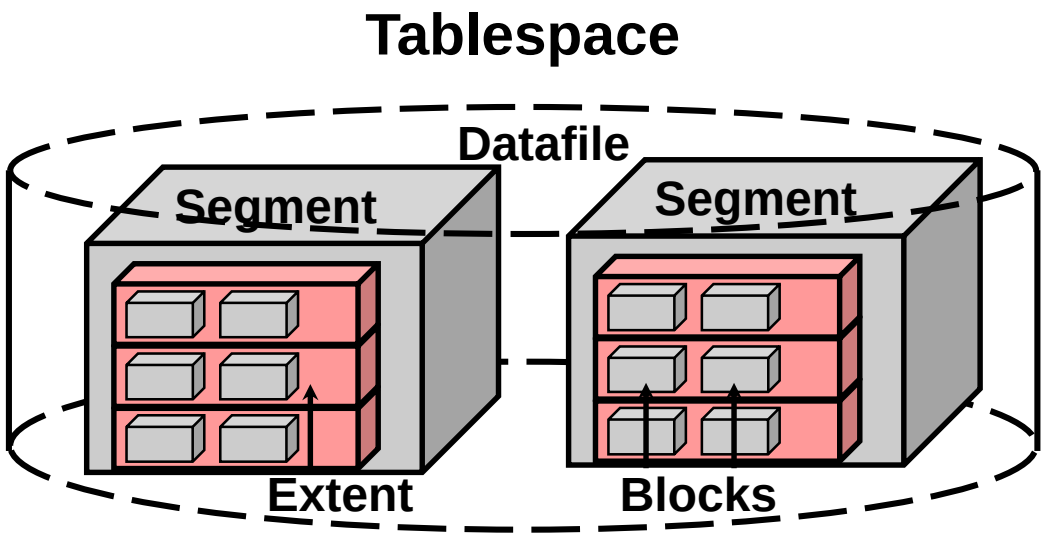
Archiver (ARCn)

- Optional background process
- Automatically archives online redo logs when `ARCHIVELOG` mode is set
- Preserves the record of all changes made to the database



Logical Structure

- Dictates how the physical space of a database is used
- Hierarchy consisting of tablespaces, segments, extents, and blocks



Processing SQL Statements

- Connect to an instance using:
  - User process
  - Server process
- The Oracle server components that are used depend on the type of SQL statement:
  - Queries return rows
  - DML statements log changes
  - Commit ensures transaction recovery
- Some Oracle server components do not participate in SQL statement processing

Summary

In this lesson, you should have learned how to:

- Explain database files: datafiles, control files, online redo logs
- Explain SGA memory structures: Database Buffer Cache, Shared Pool, and Redo Log Buffer
- Explain primary background processes: DBWn, LGWR, CKPT, PMON, SMON
- Explain the use of the background process ARCn
- Identify optional and conditional background processes
- Explain logical hierarchy

Practice 1 Overview

This practice covers the following topics:

- Review of architectural components
- Structures involved in connecting a user to an Oracle Instance