Oracle Architectural Components

D

ORACLE

Objectives

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

- Describe the Oracle server architecture and its main components
- List the structures involved in connecting a user to an Oracle instance
- List the stages in processing:
  - Queries
  - Data manipulation language (DML) statements
  - Commits

ORACLE

Objectives

This appendix introduces the Oracle server architecture by describing the files, processes, and memory structures involved in establishing a database connection and executing a SQL command.

Oracle Database Architecture: Overview

The Oracle Database consists of two main components:

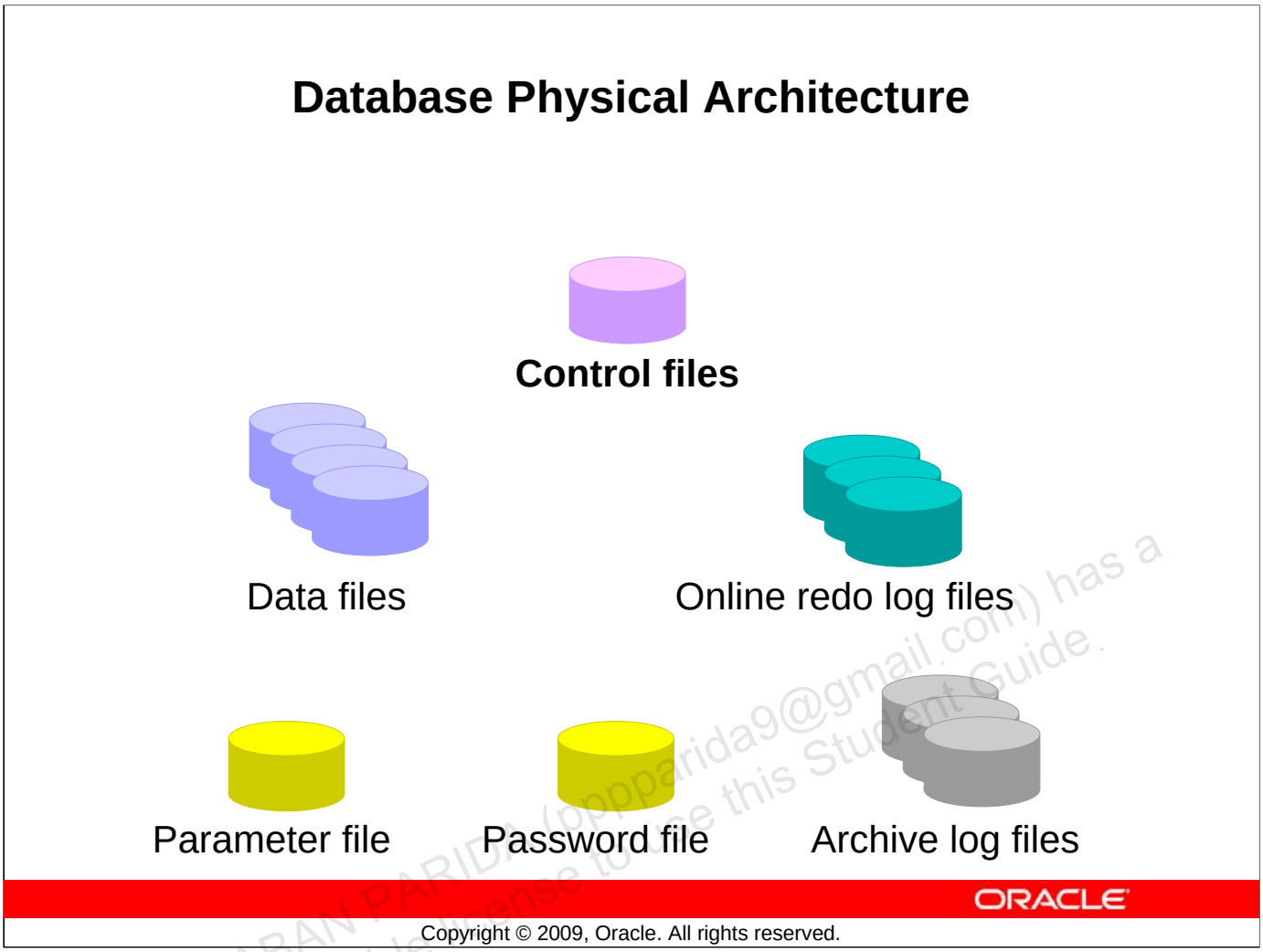
- The database or the physical structures
- The instance or the memory structures

ORACLE

Oracle Database Architecture: Overview

The Oracle Database consists of two main components—the instance and the database itself.

- The database consists of physical files such as:
  - The control file where the database configuration is stored
  - The redo log files that have information required for database recovery
  - The data files where all data is stored
  - The parameter file, which contains the parameters that control the size and properties of an instance
  - The password file, which contains the superuser or SYSDBA password
- The instance consists of the System Global Area (SGA) and the server processes that perform tasks within the database.



Database Physical Architecture

The files that make up an Oracle Database are organized into the following:

- **Control files:** These files contain data about the database itself, called the metadata. These files are critical to the database. Without them, you cannot open the data files to access the data within the database.
- **Data files:** These files contain the data of the database.
- **Online redo log files:** These files allow for instance recovery of the database. If the database were to crash and not lose any data files, the instance will be able to recover the database with the information in these files.

There are other files that are not officially part of the database but are important to the successful running of the database. These are:

- **Parameter file:** The parameter file is used to define how the instance will be configured when it starts up.
- **Password file:** This file allows users to connect remotely to the database and perform administrative tasks.
- **Archive log files:** These files contain an ongoing history of the redo generated by the instance. These files allow for database recovery. By using these files and a backup of the database, it is possible to recover a lost data file.

Control Files

- Contain physical database structure information
- Should be multiplexed to protect against loss
- Are read at mount stage

Control files

ORACLE

Control Files

When you start the instance and mount the database, the control file is read. The entries in the control file specify the physical files that constitute the database.

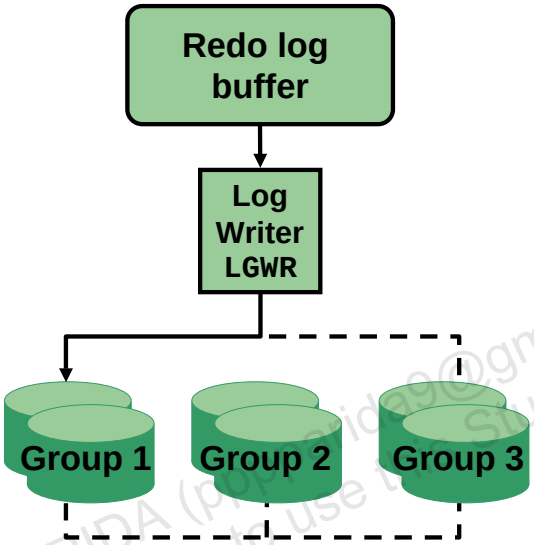
When you add additional files to your database, the control file is automatically updated.

The location of the control files is specified in an initialization parameter.

To protect against failure of the database due to the loss of the control file, you should multiplex the control file on at least three different physical devices. By specifying multiple files through the initialization parameter, you enable the Oracle Database server to maintain multiple copies of the control file.

Redo Log Files

- Record changes to the database
- Should be multiplexed to protect against loss



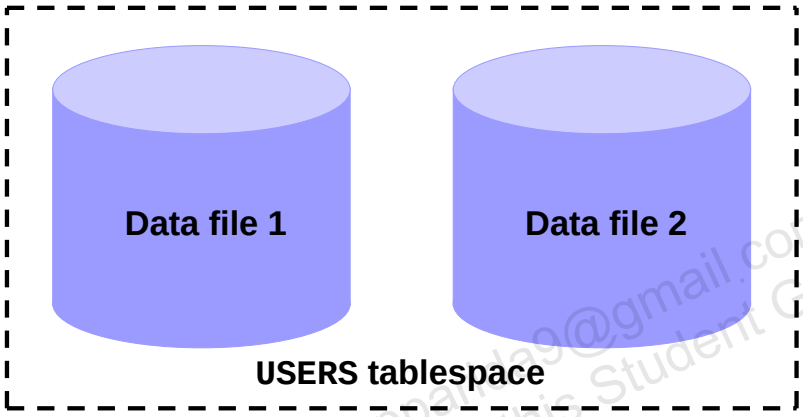
Redo Log Files

Redo log files are used to record changes to the database as a result of transactions and internal Oracle Database server actions. They protect the database from loss of integrity due to system failures caused by power outages, disk failures, and so on. Redo log files should be multiplexed to ensure that the information stored in them is not lost in the event of a disk failure.

The redo log consists of groups of redo log files. A group consists of a redo log file and its multiplexed copies. Each identical copy is said to be a member of that group and each group is identified by a number. The log writer (`LGWR`) process writes redo records from the redo log buffer to a redo log group until the file is filled or a log switch operation is requested. Then it switches and writes to the files in the next group. The redo log groups are used in a circular fashion.

Tablespaces and Data Files

- Tablespaces consist of one or more data files.
- Data files belong to only one tablespace.



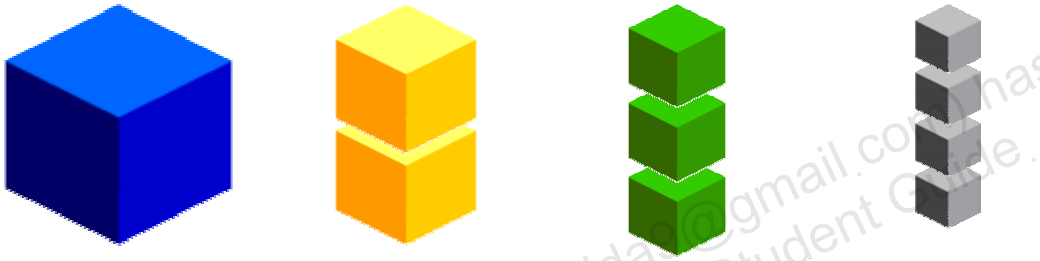
Tablespaces and Data Files

A database is divided into logical storage units called tablespaces, which can be used to group related logical structures. Each database is logically divided into one or more tablespaces. One or more data files are explicitly created for each tablespace to physically store the data of all logical structures in a tablespace.

Note: You can also create `bigfile` tablespaces, which are tablespaces with a single, but very large (up to 4-G blocks) data file. Traditional `smallfile` tablespaces (which are the default) can contain multiple data files, but the files cannot be as large. For more information about `bigfile` tablespaces, refer to the *Database Administrator's Guide*.

Segments, Extents, and Blocks

- Segments exist within a tablespace.
- Segments consist of a collection of extents.
- Extents are a collection of data blocks.
- Data blocks are mapped to OS blocks.



ORACLE

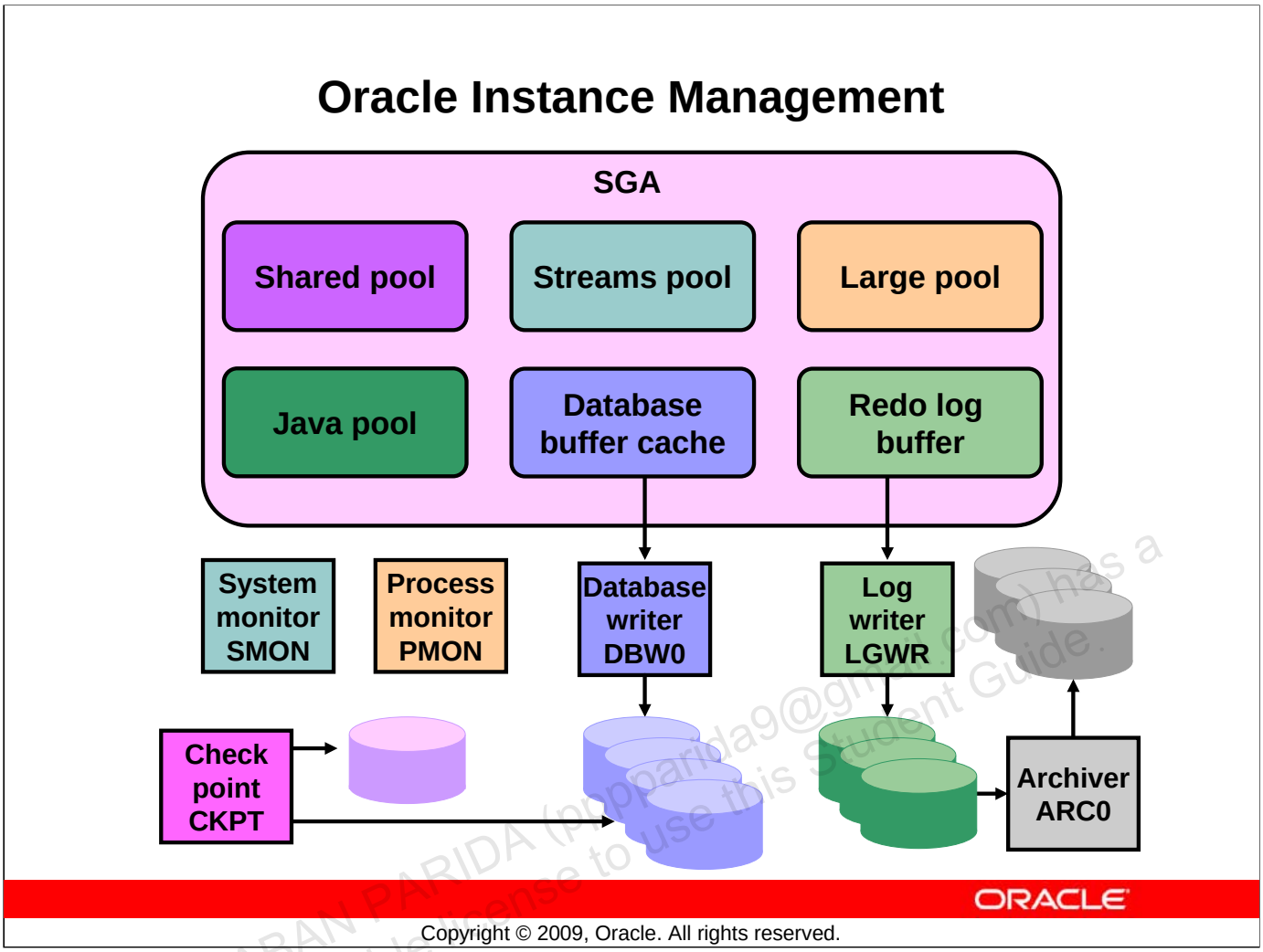
Segments, Extents, and Blocks

Database objects such as tables and indexes are stored in tablespaces as segments. Each segment contains one or more extents. An extent consists of contiguous data blocks, which means that each extent can exist only in one data file. Data blocks are the smallest unit of I/O in the database.

When the database requests a set of data blocks from the operating system (OS), the OS maps this to the actual OS block on the storage device. Because of this, you need not be aware of the physical address of any of the data in your database. This also means that a data file can be striped and or mirrored on several disks.

The size of the data block can be set at database creation time. The default size of 8 K is adequate for most databases. If your database supports a data warehouse application that has large tables and indexes, then a larger block may be beneficial. If your database supports a transactional application where reads and write are very random, then specifying a smaller block size may be beneficial. The maximum block size is dependent on your OS. The minimum block size is 2 K and should rarely (if ever) be used.

You can have tablespaces with different block sizes. Generally, this should be used only to support transportable tablespaces. For details, refer to the *Database Administrator's Guide*.



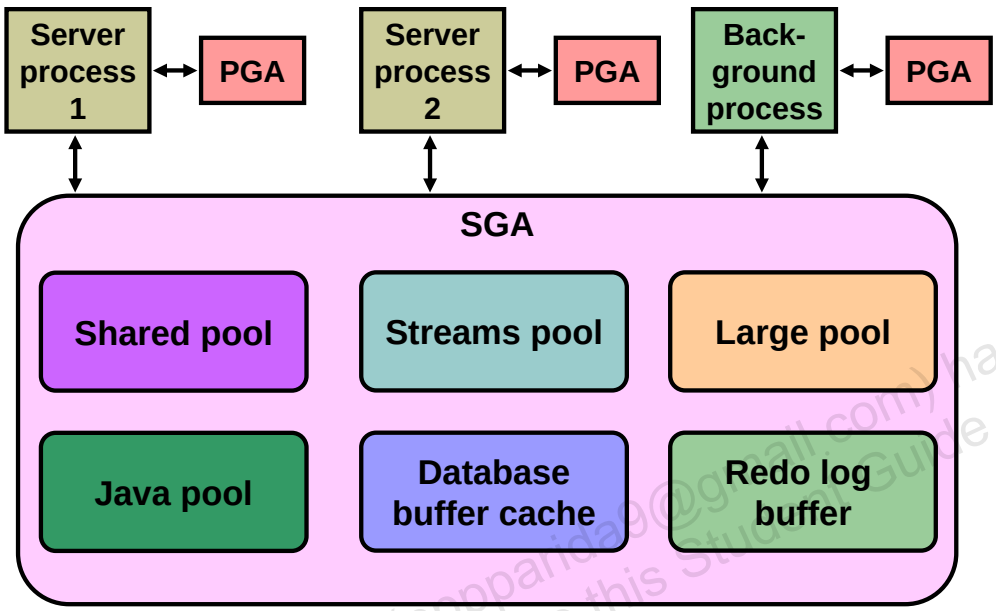
Oracle Instance Management

An Oracle Database server consists of an Oracle Database and an Oracle instance. An Oracle instance consists of memory buffers known as the System Global Area (SGA) and background processes.

The instance is idle (nonexistent) until it is started. When the instance is started, an initialization parameter file is read and the instance is configured accordingly.

After the instance is started and the database is opened, users can access the database.

Oracle Memory Structures



Oracle Memory Structures

The basic memory structures associated with an Oracle instance include:

- **System Global Area (SGA):** Shared by all server and background processes
- **Program Global Area (PGA):** Private to each server and background process; there is one PGA for each process

The System Global Area (SGA) is a shared memory area that contains data and control information for the instance.

The SGA consists of the following data structures:

- **Database buffer cache:** Caches blocks of data retrieved from the database
- **Redo log buffer:** Caches redo information (used for instance recovery) until it can be written to the physical redo log files stored on disk
- **Shared pool:** Caches various constructs that can be shared among users
- **Large pool:** Is an optional area used for buffering large I/O requests
- **Java pool:** Is used for all session-specific Java code and data within the Java Virtual Machine (JVM)
- **Streams pool:** Is used by Oracle Streams

When you start the instance by using Enterprise Manager or SQL*Plus, the memory allocated for the SGA is displayed.

Oracle Memory Structures (continued)

With the dynamic SGA infrastructure, the size of the database buffer cache, the shared pool, the large pool, the Java pool, and the Streams pool can be changed without shutting down the instance.

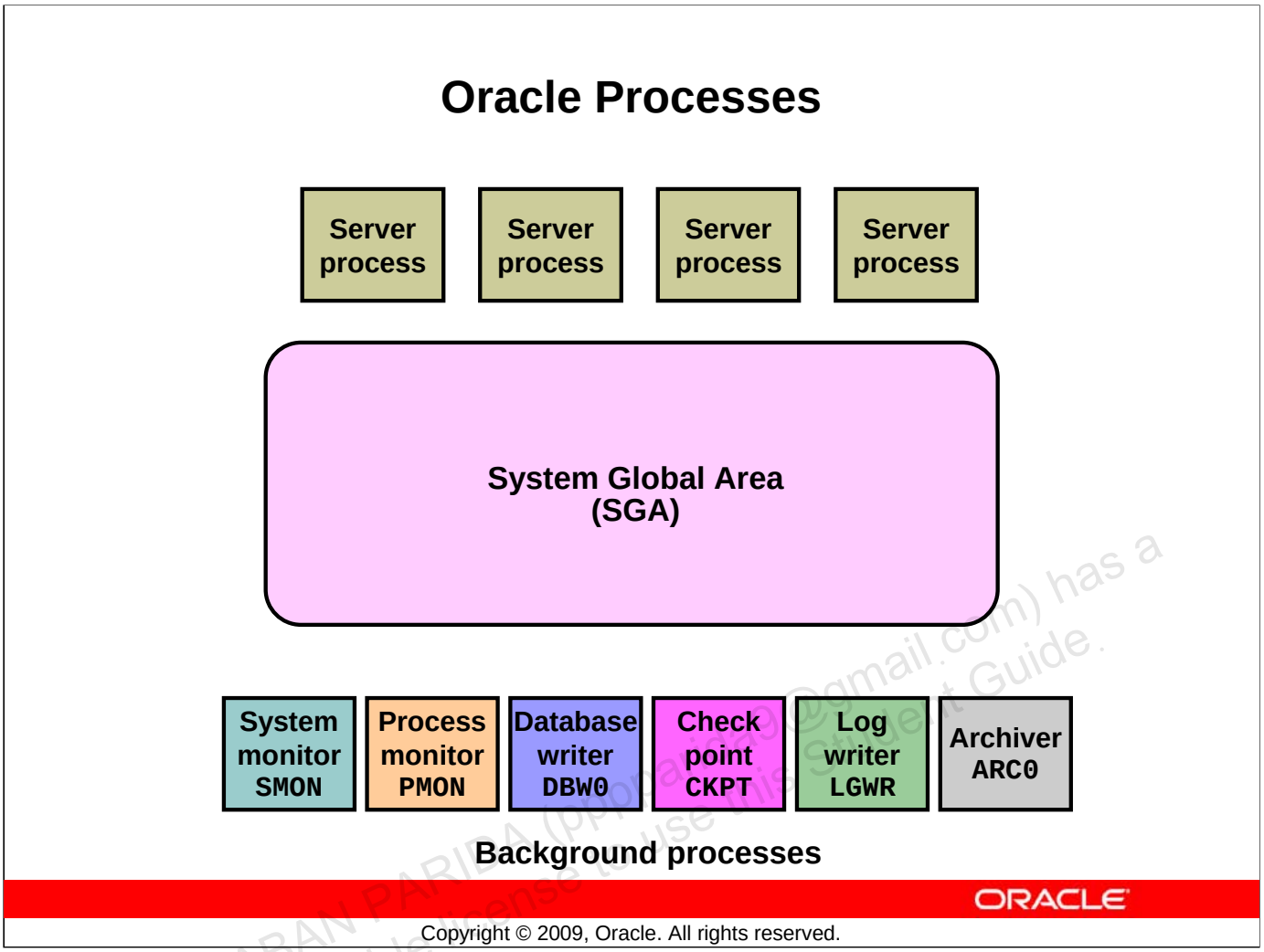
The preconfigured database has been pretuned with adequate settings for the memory parameters. However, as your database usage expands, you may find it necessary to alter the settings of the memory parameters.

Oracle provides alerts and advisors to identify memory-sizing problems and to help you determine appropriate values for memory parameters.

A Program Global Area (PGA) is a memory region that contains data and control information for each server process. A server process services a client's requests. Each server process has its own private PGA area that is created when the server process is started. Access to it is exclusive to that server process, and is read and written only by the Oracle code acting on behalf of it.

The amount of PGA memory used and its content depends on whether the instance is configured in shared server mode. Generally, the PGA contains the following:

- **Private SQL area:** Contains data such as bind information and run-time memory structures. Each session that issues a SQL statement has a private SQL area.
- **Session memory:** Is memory allocated to hold session variables and other information related to the session



Oracle Processes

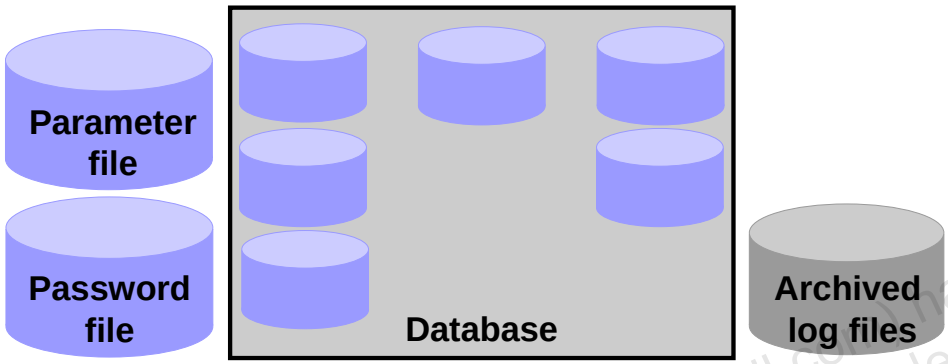
When you invoke an application program or an Oracle tool such as Enterprise Manager, the Oracle server creates a server process to execute commands issued by the application.

Oracle also creates a set of background processes for an instance that interact with each other and with the operating system to manage the memory structures, asynchronously perform I/O to write data to disk, and do general housekeeping.

Which background processes are present depends upon the features that are being used in the database. The most common background processes are the following:

- **System monitor (SMON):** Performs crash recovery when the instance is started following a failure
- **Process monitor (PMON):** Performs process cleanup when a user process fails
- **Database writer (DBW*n*):** Writes modified blocks from the database buffer cache to the files on disk
- **Checkpoint (CKPT):** Signals DBW*n* at checkpoints and updates all the data files and control files of the database to indicate the most recent checkpoint
- **Log writer (LGWR):** Writes redo log entries to disk
- **Archiver (ARC*n*):** Copies the redo log files to archival storage when the log files are full or a log switch occurs

Other Key Physical Structures



Other Key Files

The Oracle server also uses other files that are not part of the database:

- The parameter file defines the characteristics of an Oracle instance. For example, it contains parameters that size some of the memory structures in the SGA.
- The password file authenticates which users are permitted to start up and shut down an Oracle instance.
- Archived redo log files are offline copies of the redo log files that may be necessary for recovery from media failures.

Processing a SQL Statement

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

ORACLE

Components Used to Process SQL

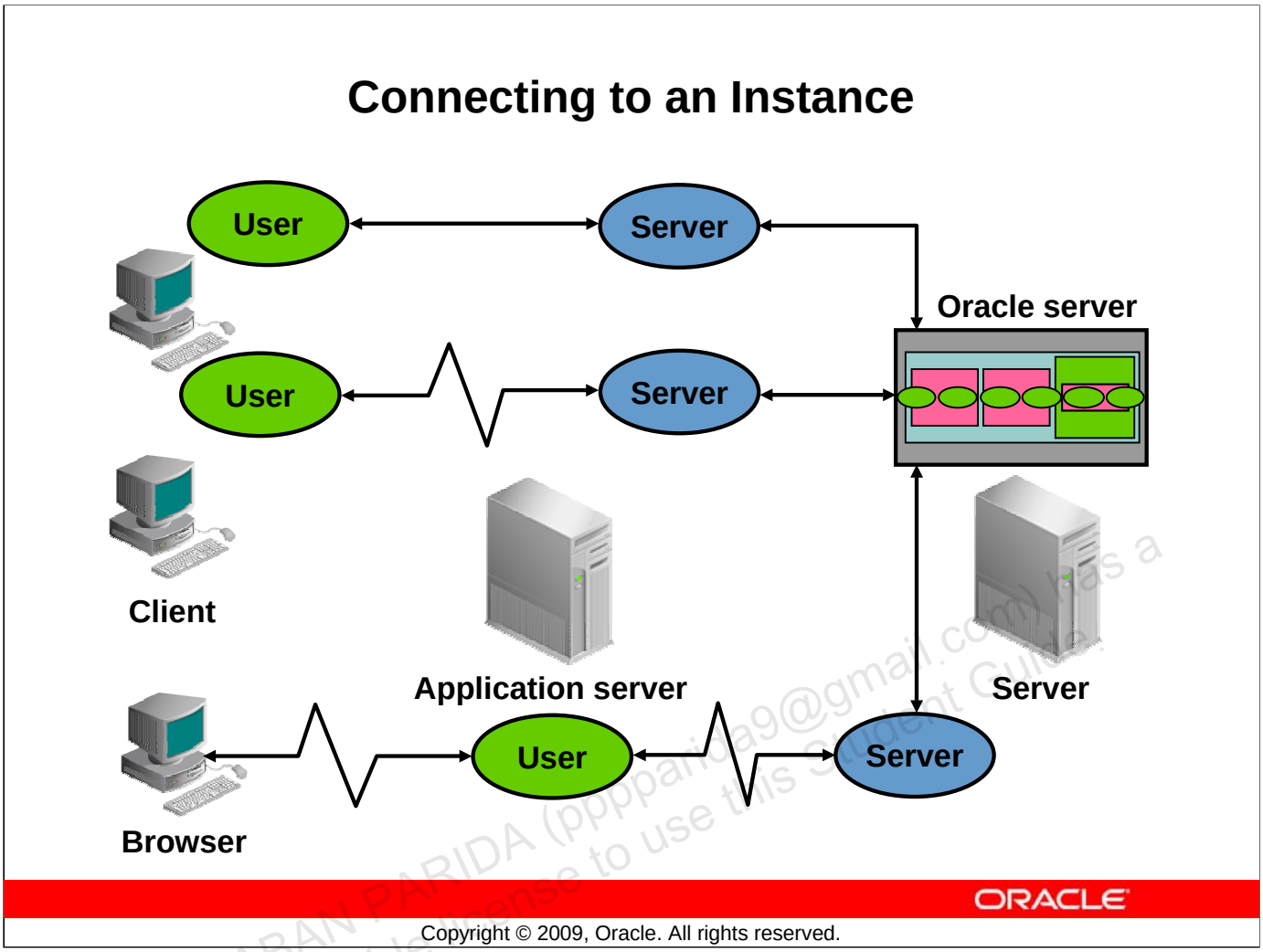
Not all the components of an Oracle instance are used to process SQL statements. The user and server processes are used to connect a user to an Oracle instance. These processes are not part of the Oracle instance, but are required to process a SQL statement.

Some of the background processes, SGA structures, and database files are used to process SQL statements. Depending on the type of SQL statement, different components are used:

- Queries require additional processing to return rows to the user.
- Data manipulation language (DML) statements require additional processing to log the changes made to the data.
- Commit processing ensures that the modified data in a transaction can be recovered.

Some required background processes do not directly participate in processing a SQL statement, but are used to improve performance and to recover the database.

The optional background process, ARC0, is used to ensure that a production database can be recovered.



Processes Used to Connect to an Instance

Before users can submit SQL statements to the Oracle server, they must connect to an instance.

The user starts a tool such as SQL Developer or runs an application developed using a tool such as Oracle Forms. This application or tool is executed in a *user process*.

In the most basic configuration, when a user logs on to the Oracle server, a process is created on the computer running the Oracle server. This process is called a server process. The server process communicates with the Oracle instance on behalf of the user process that runs on the client. The server process executes SQL statements on behalf of the user.

Connection

A connection is a communication pathway between a user process and an Oracle server. A database user can connect to an Oracle server in one of three ways:

- The user logs on to the operating system running the Oracle instance and starts an application or tool that accesses the database on that system. The communication pathway is established using the interprocess communication mechanisms available on the host operating system.

Processes Used to Connect to an Instance (continued)

Connection (continued)

- The user starts the application or tool on a local computer and connects over a network to the computer running the Oracle instance. In this configuration, called client/server, network software is used to communicate between the user and the Oracle server.
- In a three-tiered connection, the user's computer communicates over the network to an application or a network server, which is connected through a network to the machine running the Oracle instance. For example, the user runs a browser on a network computer to use an application residing on an NT server that retrieves data from an Oracle Database running on a UNIX host.

Sessions

A session is a specific connection of a user to an Oracle server. The session starts when the user is validated by the Oracle server, and it ends when the user logs out or when there is an abnormal termination. For a given database user, many concurrent sessions are possible if the user logs on from many tools, applications, or terminals at the same time. Except for some specialized database administration tools, starting a database session requires that the Oracle server be available for use.

Note: The type of connection explained here, where there is a one-to-one correspondence between a user and server process, is called a *dedicated server connection*.

Processing a Query

- Parse:
  - Search for identical statement.
  - Check syntax, object names, and privileges.
  - Lock objects used during parse.
  - Create and store execution plan.
- Execute: Identify rows selected.
- Fetch: Return rows to user process.

ORACLE

Query Processing Steps

Queries are different from other types of SQL statements because, if successful, they return data as results. Whereas other statements simply return success or failure, a query can return one row or thousands of rows.

There are three main stages in the processing of a query:

- Parse
- Execute
- Fetch

During the *parse* stage, the SQL statement is passed from the user process to the server process, and a parsed representation of the SQL statement is loaded into a shared SQL area.

During parse, the server process performs the following functions:

- Searches for an existing copy of the SQL statement in the shared pool
- Validates the SQL statement by checking its syntax
- Performs data dictionary lookups to validate table and column definitions

The execute fetch executes the statement using the best optimizer approach and the fetch retrieves the rows back to the user.

Shared Pool

- The library cache contains the SQL statement text, parsed code, and execution plan.
- The data dictionary cache contains table, column, and other object definitions and privileges.
- The shared pool is sized by `SHARED_POOL_SIZE`.

Shared Pool Components

During the parse stage, the server process uses the area in the SGA known as the shared pool to compile the SQL statement. The shared pool has two primary components:

- Library cache
- Data dictionary cache

Library Cache

The library cache stores information about the most recently used SQL statements in a memory structure called a shared SQL area. The shared SQL area contains:

- The text of the SQL statement
- The parse tree, which is a compiled version of the statement
- The execution plan, with steps to be taken when executing the statement

The optimizer is the function in the Oracle server that determines the optimal execution plan.

If a SQL statement is reexecuted and a shared SQL area already contains the execution plan for the statement, then the server process does not need to parse the statement. The library cache improves the performance of applications that reuse SQL statements by reducing parse time and memory requirements. If the SQL statement is not reused, it is eventually aged out of the library cache.

Shared Pool Components (continued)

Data Dictionary Cache

The data dictionary cache, also known as the dictionary cache or row cache, is a collection of the most recently used definitions in the database. It includes information about database files, tables, indexes, columns, users, privileges, and other database objects.

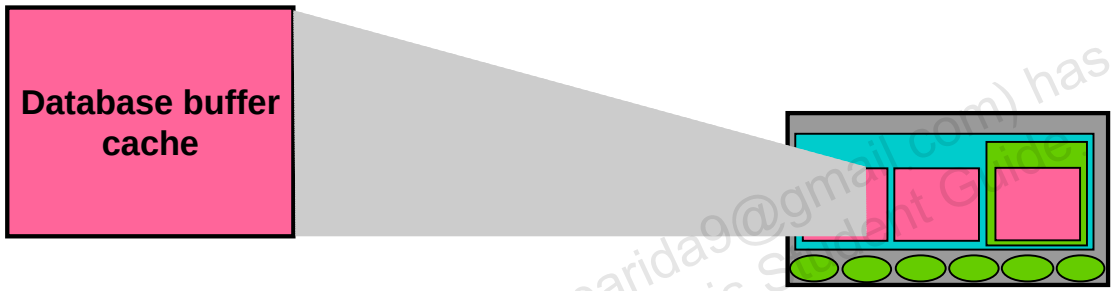
During the parse phase, the server process looks for the information in the dictionary cache to resolve the object names specified in the SQL statement and to validate the access privileges. If necessary, the server process initiates the loading of this information from the data files.

Sizing the Shared Pool

The size of the shared pool is specified by the `SHARED_POOL_SIZE` initialization parameter.

Database Buffer Cache

- It stores the most recently used blocks.
- The size of a buffer is based on DB_BLOCK_SIZE.
- The number of buffers is defined by DB_BLOCK_BUFFERS.



Functions of the Database Buffer Cache

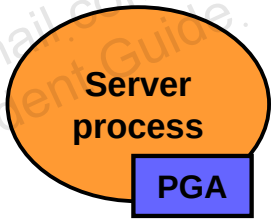
When a query is processed, the server process looks in the database buffer cache for any blocks it needs. If the block is not found in the database buffer cache, the server process reads the block from the data file and places a copy in the buffer cache. Because subsequent requests for the same block may find the block in memory, the requests may not require physical reads. The Oracle server uses a least recently used algorithm to age out buffers that have not been accessed recently to make room for new blocks in the buffer cache.

Sizing the Database Buffer Cache

The size of each buffer in the buffer cache is equal to the size of an Oracle block, and it is specified by the DB_BLOCK_SIZE parameter. The number of buffers is equal to the value of the DB_BLOCK_BUFFERS parameter.

Program Global Area (PGA)

- Is not shared
- Is writable only by the server process
- Contains:
  - Sort area
  - Session information
  - Cursor state
  - Stack space



ORACLE

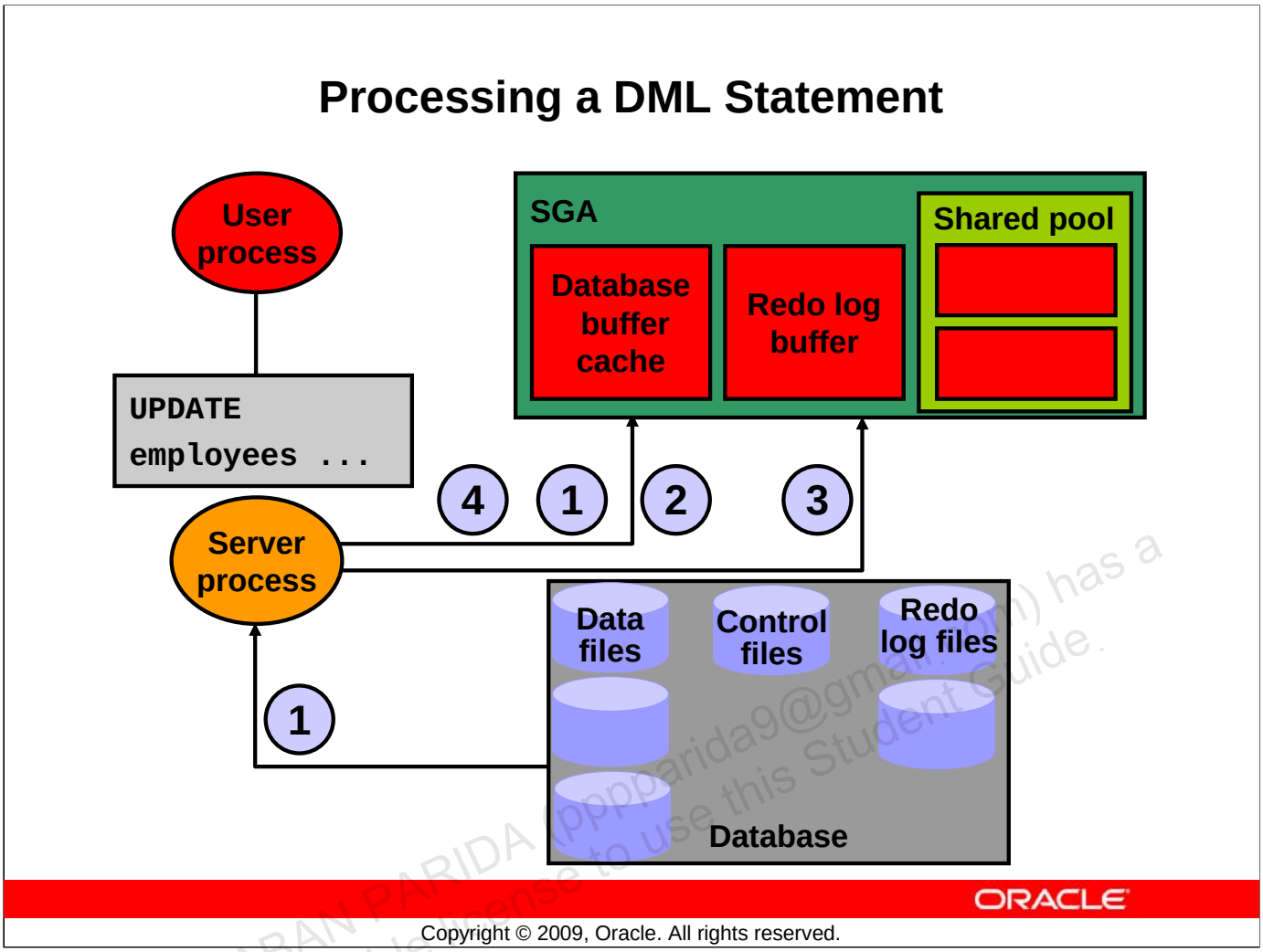
Program Global Area: Components

A Program Global Area (PGA) is a memory region that contains data and control information for a server process. It is a nonshared memory created by Oracle when a server process is started. Access to it is exclusive to that server process, and is read and written only by the Oracle server code acting on behalf of it. The PGA memory allocated by each server process attached to an Oracle instance is referred to as the aggregated PGA memory allocated by the instance.

In a dedicated server configuration, the PGA of the server includes the following components:

- **Sort area:** Is used for any sorts that may be required to process the SQL statement
- **Session information:** Includes user privileges and performance statistics for the session
- **Cursor state:** Indicates the stage in the processing of the SQL statements that are currently used by the session
- **Stack space:** Contains other session variables

The PGA is allocated when a process is created and deallocated when the process is terminated.



DML Processing Steps

A data manipulation language (DML) statement requires only two phases of processing:

- Parse is the same as the parse phase used for processing a query.
- Execute requires additional processing to make data changes.

DML Execute Phase

To execute a DML statement:

- If the data and rollback blocks are not already in the buffer cache, the server process reads them from the data files into the buffer cache.
- The server process places locks on the rows that are to be modified.
- In the redo log buffer, the server process records the changes to be made to the rollback and data blocks.
- The rollback block changes record the values of the data before it is modified. The rollback block is used to store the “before image” of the data, so that the DML statements can be rolled back if necessary.
- The data block changes record the new values of the data.

DML Processing Steps (continued)

DML Execute Phase (continued)

The server process records the “before image” to the rollback block and updates the data block. Both of these changes are done in the database buffer cache. Any changed blocks in the buffer cache are marked as dirty buffers (that is, buffers that are not the same as the corresponding blocks on the disk).

The processing of a `DELETE` or `INSERT` command uses similar steps. The “before image” for a `DELETE` contains the column values in the deleted row, and the “before image” of an `INSERT` contains the row location information.

Because the changes made to the blocks are only recorded in memory structures and are not written immediately to disk, a computer failure that causes the loss of the SGA can also lose these changes.

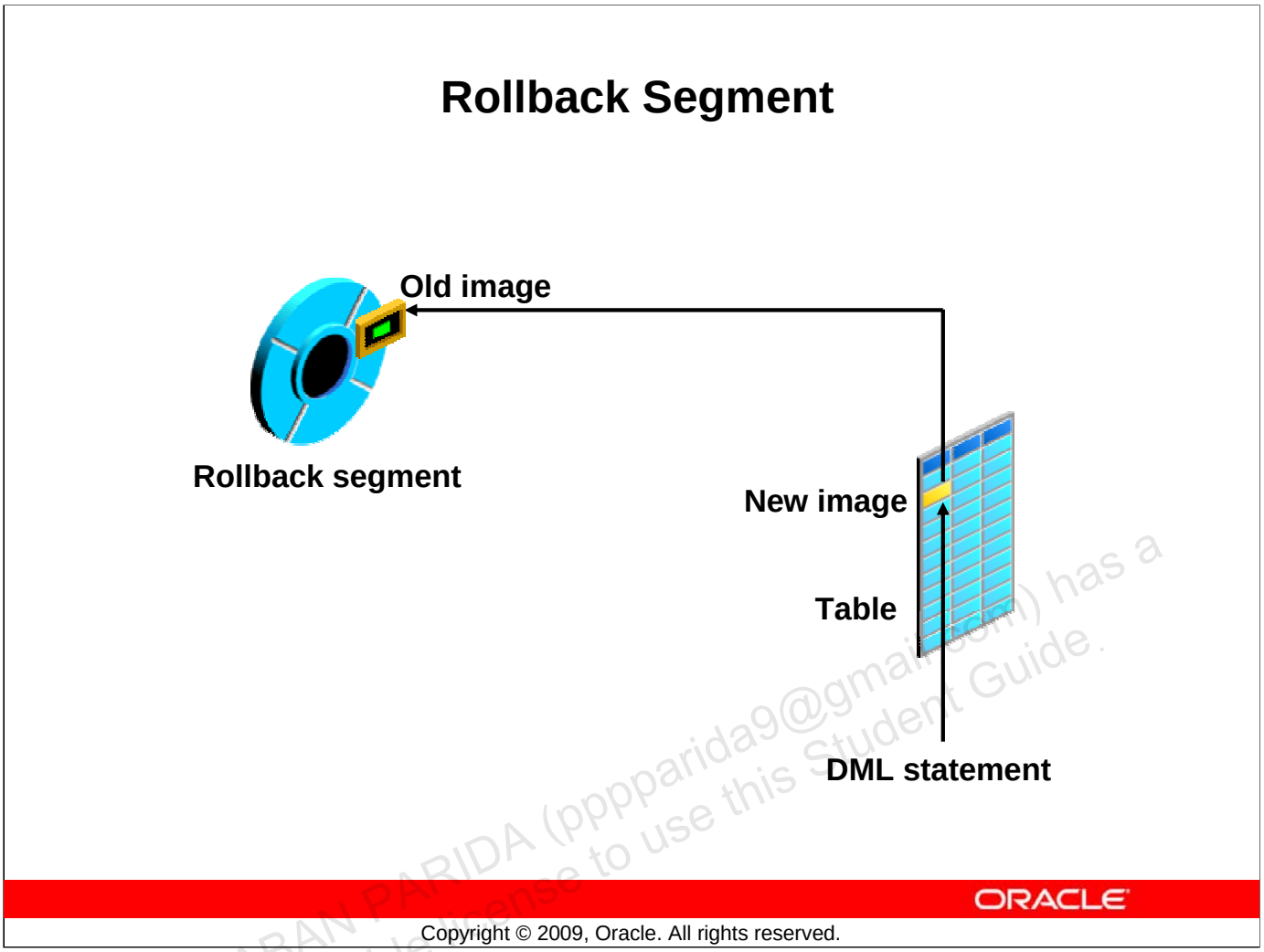
Redo Log Buffer

- Has its size defined by LOG_BUFFER
- Records changes made through the instance
- Is used sequentially
- Is a circular buffer

Redo Log Buffer: Characteristics

The server process records most of the changes made to data file blocks in the redo log buffer, which is a part of the SGA. The redo log buffer has the following characteristics:

- Its size in bytes is defined by the LOG_BUFFER parameter.
- It records the block that is changed, the location of the change, and the new value in a redo entry. A redo entry makes no distinction between the type of block that is changed; it only records which bytes are changed in the block.
- The redo log buffer is used sequentially, and changes made by one transaction may be interleaved with changes made by other transactions.
- It is a circular buffer that is reused after it is filled, but only after all the old redo entries are recorded in the redo log files.



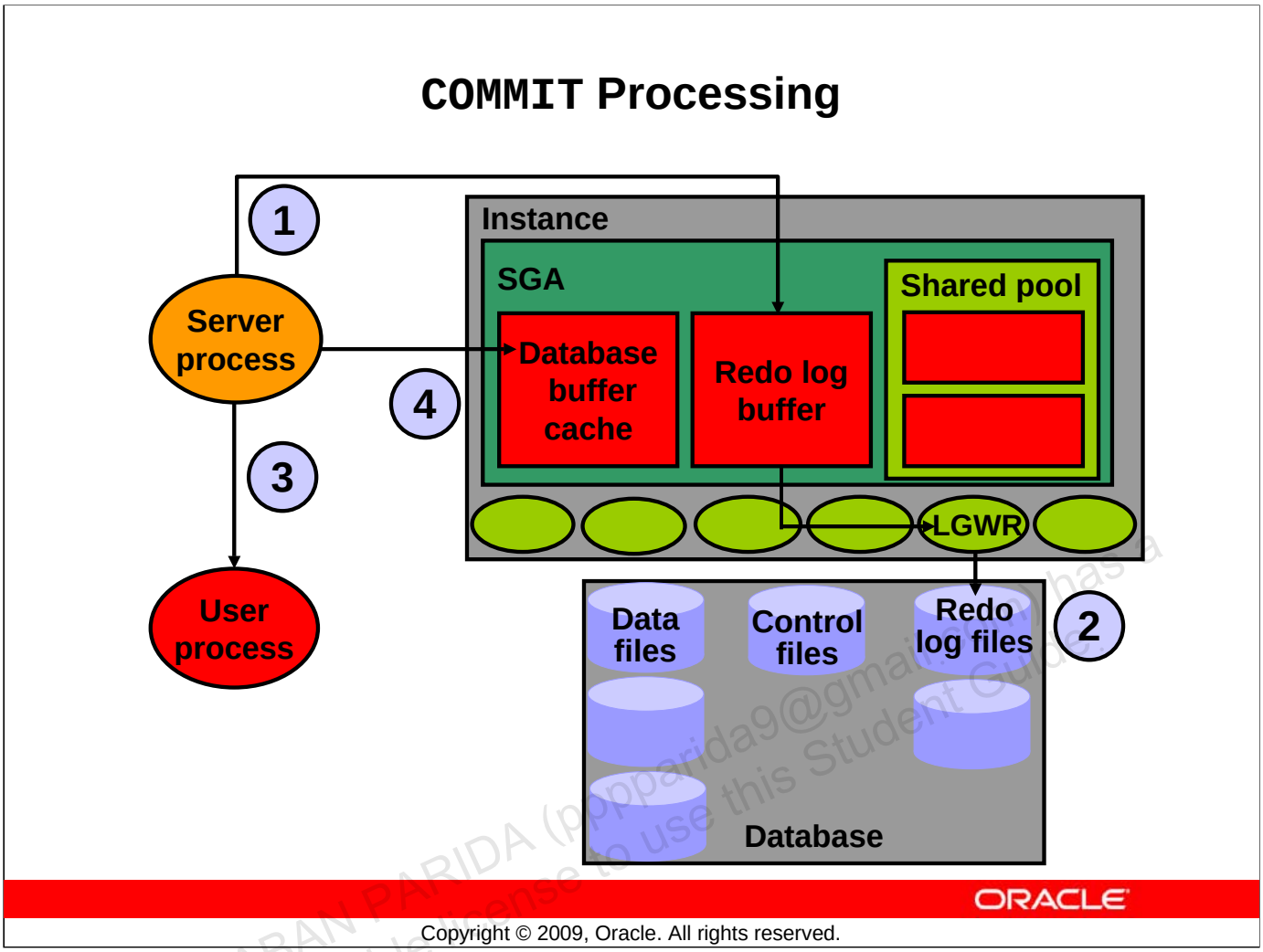
Rollback Segment

Before making a change, the server process saves the old data value in a rollback segment. This "before image" is used to:

- Undo the changes if the transaction is rolled back
- Provide read consistency by ensuring that other transactions do not see uncommitted changes made by the DML statement
- Recover the database to a consistent state in case of failures

Rollback segments, such as tables and indexes, exist in data files, and rollback blocks are brought into the database buffer cache as required. Rollback segments are created by the DBA.

Changes to rollback segments are recorded in the redo log buffer.



Fast COMMIT

The Oracle server uses a fast COMMIT mechanism that guarantees that the committed changes can be recovered in case of instance failure.

System Change Number

Whenever a transaction commits, the Oracle server assigns a commit system change number (SCN) to the transaction. The SCN is monotonically incremented and is unique within the database. It is used by the Oracle server as an internal time stamp to synchronize data and to provide read consistency when data is retrieved from the data files. Using the SCN enables the Oracle server to perform consistency checks without depending on the date and time of the operating system.

Steps in Processing COMMITs

When a COMMIT is issued, the following steps are performed:

- The server process places a commit record, along with the SCN, in the redo log buffer.
- LGWR performs a contiguous write of all the redo log buffer entries up to and including the commit record to the redo log files. After this point, the Oracle server can guarantee that the changes will not be lost even if there is an instance failure.

Fast COMMIT (continued)

Steps in Processing COMMITs (continued)

- The user is informed that the COMMIT is complete.
- The server process records information to indicate that the transaction is complete and that resource locks can be released.

Flushing of the dirty buffers to the data file is performed independently by DBW0 and can occur either before or after the commit.

Advantages of the Fast COMMIT

The fast COMMIT mechanism ensures data recovery by writing changes to the redo log buffer instead of the data files. It has the following advantages:

- Sequential writes to the log files are faster than writing to different blocks in the data file.
- Only the minimal information that is necessary to record changes is written to the log files; writing to the data files would require whole blocks of data to be written.
- If multiple transactions request to commit at the same time, the instance piggybacks redo log records into a single write.
- Unless the redo log buffer is particularly full, only one synchronous write is required per transaction. If piggybacking occurs, there can be less than one synchronous write per transaction.
- Because the redo log buffer may be flushed before the COMMIT, the size of the transaction does not affect the amount of time needed for an actual COMMIT operation.

Note: Rolling back a transaction does not trigger LGWR to write to disk. The Oracle server always rolls back uncommitted changes when recovering from failures. If there is a failure after a rollback, before the rollback entries are recorded on disk, the absence of a commit record is sufficient to ensure that the changes made by the transaction are rolled back.

Summary

In this appendix, you should have learned how to:

- Identify database files: data files, control files, and online redo logs
- Describe SGA memory structures: DB buffer cache, shared SQL pool, and redo log buffer
- Explain primary background processes: `DBW0, LGWR, CKPT, PMON, SMON`, and `ARC0`
- List SQL processing steps: parse, execute, and fetch

ORACLE

Summary

Oracle Database Files

The Oracle Database includes the following files:

- **Control files:** Contain information required to verify the integrity of the database, including the names of the other files in the database (The control files are usually mirrored.)
- **Data files:** Contain the data in the database, including tables, indexes, rollback segments, and temporary segments
- **Online redo logs:** Contain the changes made to the data files (Online redo logs are used for recovery and are usually mirrored.)

Other files commonly used with the database include:

- **Parameter file:** Defines the characteristics of an Oracle instance
- **Password file:** Authenticates privileged database users
- **Archived redo logs:** Are backups of the online redo logs

Summary (continued)

SGA Memory Structures

The System Global Area (SGA) has three primary structures:

- **Shared pool:** Stores the most recently executed SQL statements and the most recently used data from the data dictionary
- **Database buffer cache:** Stores the most recently used data
- **Redo log buffer:** Records changes made to the database using the instance

Background Processes

A production Oracle instance includes the following processes:

- **Database writer (`DBW0`):** Writes changed data to the data files
- **Log writer (`LGWR`):** Records changes to the data files in the online redo log files
- **System monitor (`SMON`):** Checks for consistency and initiates recovery of the database when the database is opened
- **Process monitor (`PMON`):** Cleans up the resources if one of the processes fails
- **Checkpoint process (`CKPT`):** Updates the database status information after a checkpoint
- **Archiver (`ARC0`):** Backs up the online redo log to ensure recovery after a media failure (This process is optional, but is usually included in a production instance.)

Depending on its configuration, the instance may also include other processes.

SQL Statement Processing Steps

The steps used to process a SQL statement include:

- **Parse:** Compiles the SQL statement
- **Execute:** Identifies selected rows or applies DML changes to the data
- **Fetch:** Returns the rows queried by a `SELECT` statement