# Exploring the Oracle Database Architecture

**Chapter 1**

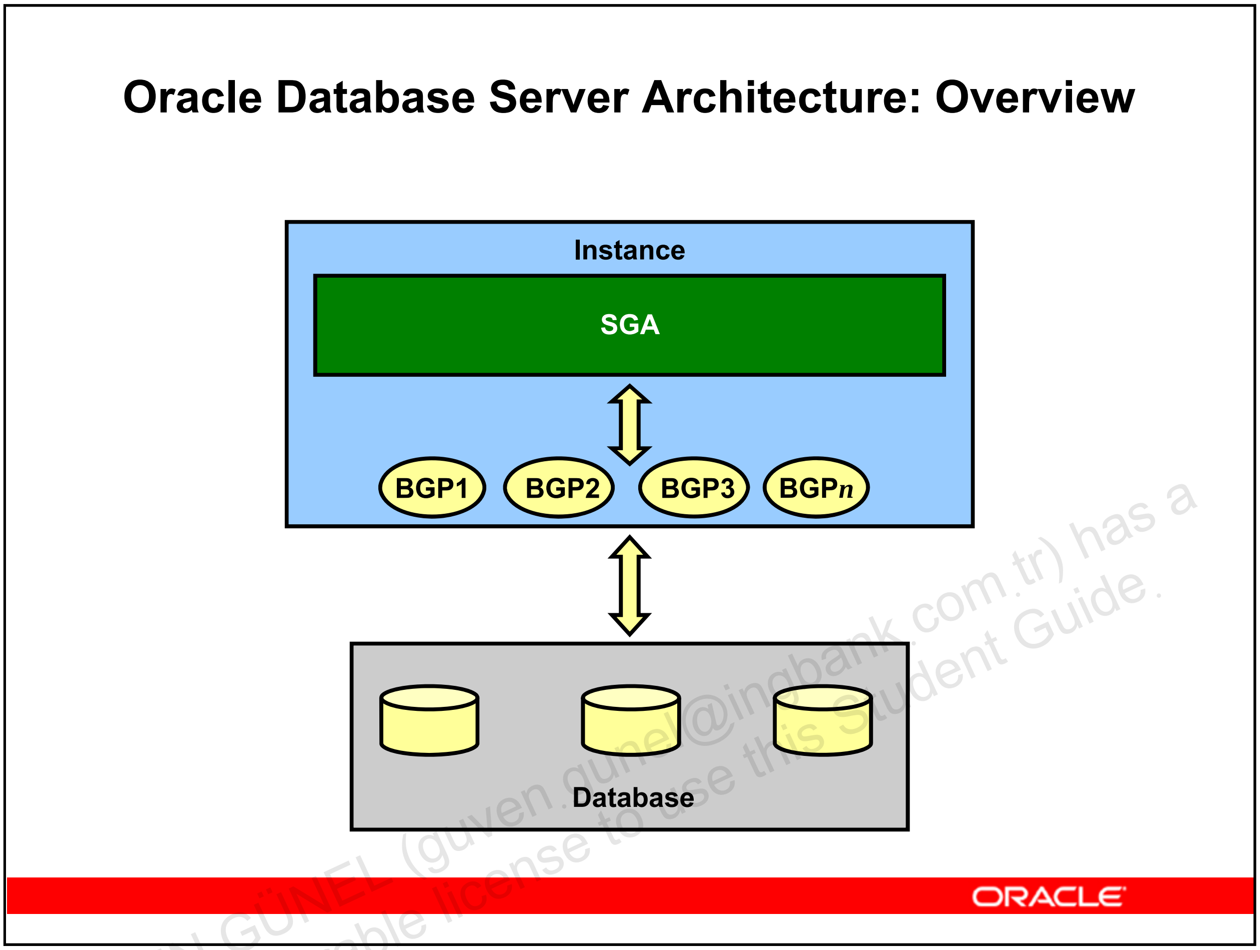
# Exploring the Oracle Database Architecture

1

ORACLE

**Objectives**

This lesson provides an overview of the Oracle Database server architecture. You learn about physical and logical structures and about the various components.




## Oracle Database Server Architecture: Overview

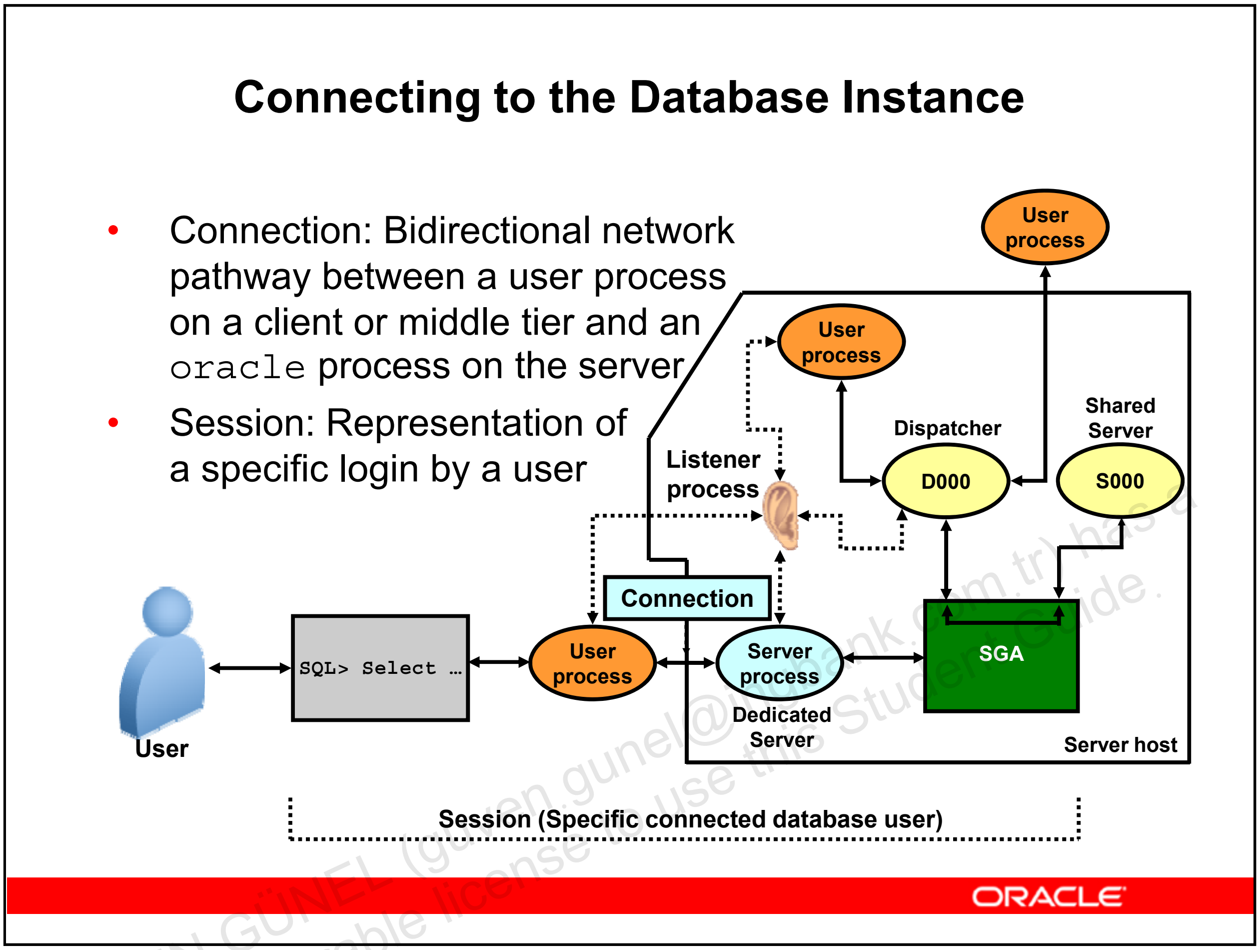
An Oracle Database server consists of an Oracle Database and one or more Oracle Database instances. An instance consists of memory structures and background processes. Every time an instance is started, a shared memory area called the System Global Area (SGA) is allocated and the background processes are started.

The SGA contains data and control information for one Oracle Database instance.

The background processes consolidate functions that would otherwise be handled by multiple Oracle Database server programs running for each user process. They may asynchronously perform input/output (I/O) and monitor other Oracle Database processes to provide increased parallelism for better performance and reliability.

The database consists of physical files and logical structures discussed later in this lesson. Because the physical and logical structures are separate, the physical storage of data can be managed without affecting access to the logical storage structures.

**Note:** Oracle Real Application Clusters (Oracle RAC) comprises two or more Oracle Database instances running on multiple clustered computers that communicates with each other by means of an interconnect and access the same Oracle Database.




## Connecting to the Database Instance

When users connect to an Oracle Database server, they are connected to an Oracle Database instance. The database instance services those users by allocating other memory areas in addition to the SGA, and starting other processes in addition to the Oracle Database background processes:

- User processes sometimes called client or foreground processes are created to run the software code of an application program. Most environments have separate machines for the client processes. A user process also manages communication with a corresponding server process through a program interface.
- Oracle Database server creates server processes to handle requests from connected user processes. A server process communicates with the user process and interacts with the instance and the database to carry out requests from the associated user process.

An Oracle Database instance can be configured to vary the number of user processes for each server process. In a dedicated server configuration, a server process handles requests for a single user process.

A shared server configuration enables many user processes to share a small number of shared server processes, minimizing the number of server processes and maximizing the use of available system resources. One or more dispatcher processes are then used to queue user process requests in the SGA and dequeue shared server responses.

The Oracle Database server runs a listener that is responsible for handling network connections. The application connects to the listener that creates a dedicated server process or handles the connection to a dispatcher.

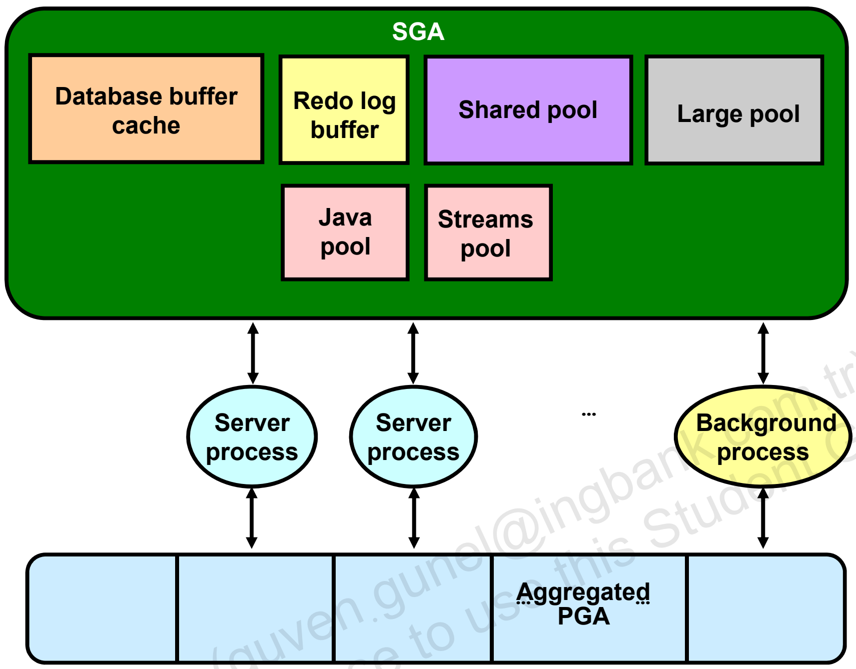
Connections and sessions are closely related to user processes, but are very different in meaning:

- A connection is a communication pathway between a user process and an Oracle Database instance. A communication pathway is established by using available interprocess communication mechanisms (on a computer that runs both the user process and Oracle Database) or network software (when different computers run the database application and Oracle Database, and communicate through a network).
- A session represents the state of a current database user login to the database instance. For example, when a user starts SQL*Plus, the user must provide a valid database username and password, and then a session is established for that user. A session lasts from the time a user connects until the user disconnects or exits the database application.

**Note:** Multiple sessions can be created and exist concurrently for a single Oracle database user using the same username. For example, a user with the username/password of `HR/HR` can connect to the same Oracle Database instance several times.

## Oracle Database Memory Structures: Overview

Oracle Database Memory Structures: Overview



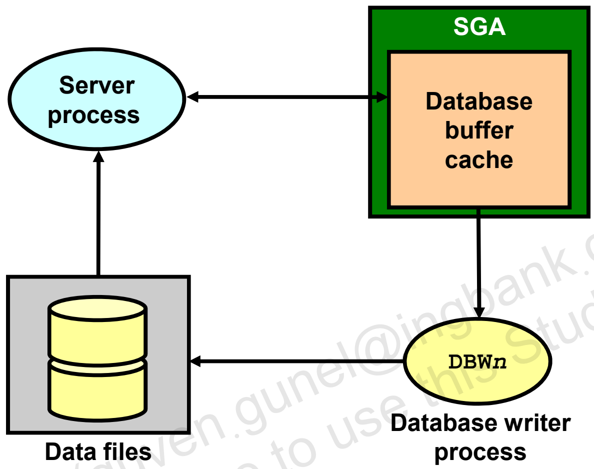

Oracle Database Memory Structures: Overview

Oracle Database allocates memory structures for various purposes. For example, memory stores the program code that is run, data that is shared among users, and private data areas for each connected user. Two basic memory structures are associated with an instance:

- **System Global Area (SGA):** The SGA is shared by all server and background processes. The SGA includes the following data structures:
  - **Database buffer cache:** Caches blocks of data retrieved from the database files
  - **Redo log buffer:** Caches recovery information before writing it to the physical files
  - **Shared pool:** Caches various constructs that can be shared among sessions
  - **Large pool:** Optional area used for certain operations, such as Oracle backup and recovery operations, and I/O server processes
  - **Java pool:** Used for session-specific Java code and data in the Java Virtual Machine (JVM)
  - **Streams pool:** Used by Oracle Streams to store information about the capture and apply processes
- **Program Global Areas (PGA):** Memory regions that contain data and control information about a server or background process. A PGA is suballocated from the aggregated PGA area.

## Database Buffer Cache

- Is a part of the SGA
- Holds copies of data blocks that are read from data files
- Is shared by all concurrent processes






### Database Buffer Cache

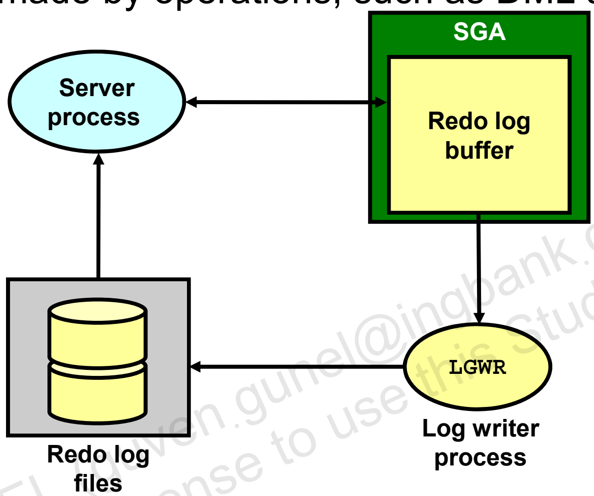
The database buffer cache is the portion of the SGA that holds copies of data blocks that are read from data files. All users concurrently connected to the instance share access to the database buffer cache.

The first time an Oracle Database server process requires a particular piece of data, it searches for the data in the database buffer cache. If the process finds the data already in the cache (a cache hit), it can read the data directly from memory. If the process cannot find the data in the cache (a cache miss), it must copy the data block from a data file on disk into a buffer in the cache before accessing the data. Accessing data through a cache hit is faster than data access through a cache miss.

The buffers in the cache are managed by a complex algorithm that uses a combination of least recently used (LRU) lists and touch count. The `DBWn` (Database Writers) processes are responsible for writing modified (dirty) buffers in the database buffer cache to disk when necessary.

## Redo Log Buffer

- Is a circular buffer in the SGA (based on the number of CPUs)
- Contains redo entries that have the information to redo changes made by operations, such as DML and DDL








### Redo Log Buffer

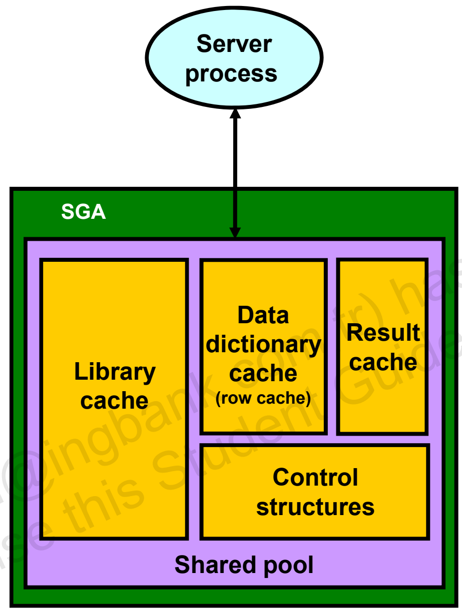
The redo log buffer is a circular buffer in the SGA that holds information about changes made to the database. This information is stored in redo entries. Redo entries contain the information necessary to reconstruct (or redo) changes that are made to the database by `INSERT`, `UPDATE`, `DELETE`, `CREATE`, `ALTER`, or `DROP` operations. Redo entries are used for database recovery, if necessary.

Redo entries are copied by Oracle Database server processes from the user's memory space to the redo log buffer in the SGA. The redo entries take up continuous, sequential space in the buffer. The `LGWR` (log writer) background process writes the redo log buffer to the active redo log file (or group of files) on disk. `LGWR` is a background process that is capable of asynchronous I/O.

**Note:** Depending on the number of CPUs on your system, there may be more than one redo log buffer. They are automatically allocated.

# Shared Pool

- Is part of the SGA
- Contains:
  - Library cache
    - Shared parts of SQL and PL/SQL statements
  - Data dictionary cache
  - Result cache:
    - SQL queries
    - PL/SQL functions
  - Control structures
    - Locks






## Shared Pool

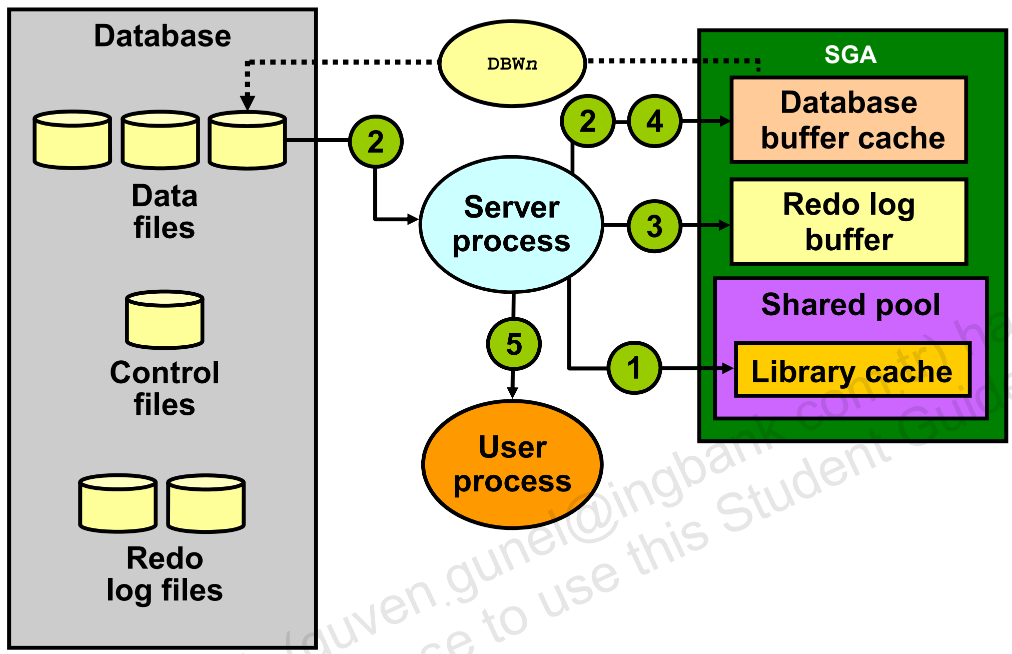
The shared pool portion of the SGA contains the following main parts:

- The library cache includes the sharable parts of SQL statements, PL/SQL procedures and packages. It also contains control structures such as locks.
- The data dictionary is a collection of database tables containing reference information about the database. The data dictionary is accessed so often by Oracle Database that two special locations in memory are designated to hold dictionary data. One area is called the data dictionary cache, also known as the row cache, and the other area is called the library cache. All Oracle Database server processes share these two caches for access to data dictionary information.
- The result cache is composed of the SQL query result cache and the PL/SQL function result cache. This cache is used to store results of SQL queries or PL/SQL functions to speed up their future executions.
- Control structures are essentially lock structures.

**Note:** In general, any item in the shared pool remains until it is flushed according to a modified LRU algorithm.

## Processing a DML Statement: Example

Processing a DML Statement: Example

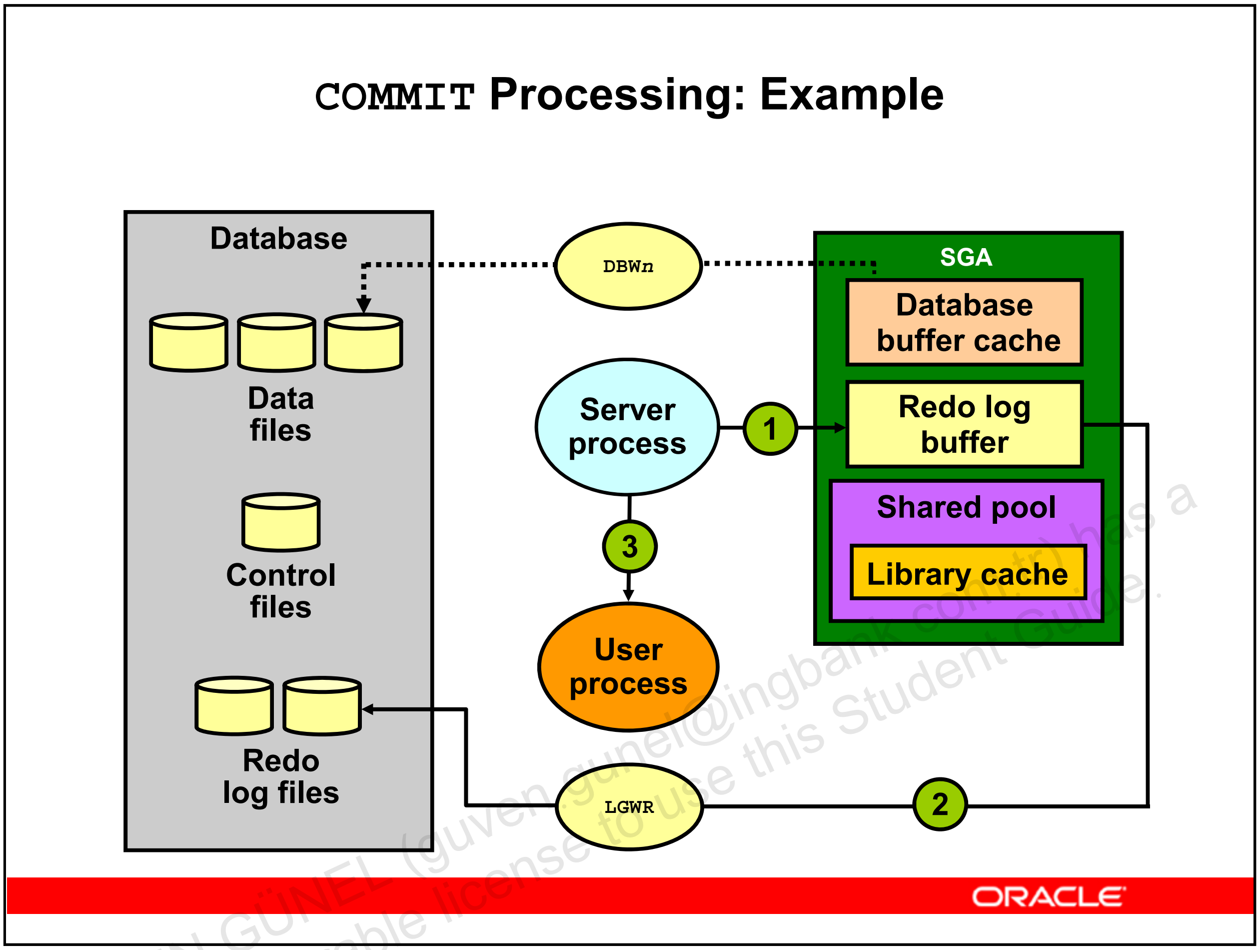



**Processing a DML Statement: Example**

The steps involved in executing a data manipulation language (DML) statement are:

1. The server process receives the statement and checks the library cache for any shared SQL area that contains a similar SQL statement. If a shared SQL area is found, the server process checks the user's access privileges to the requested data, and the existing shared SQL area is used to process the statement. If not, a new shared SQL area is allocated for the statement, so that it can be parsed and processed.
2. If the data and undo segment blocks are not already in the buffer cache, the server process reads them from the data files into the buffer cache. The server process locks the rows that are to be modified.
3. The server process records the changes to be made to the data buffers as well as the undo changes. These changes are written to the redo log buffer before the in-memory data and undo buffers are modified. This is called write-ahead logging.
4. The undo segment buffers contain values of the data before it is modified. The undo buffers are used to store the before image of the data so that the DML statements can be rolled back, if necessary. The data buffers record the new values of the data.
5. The user gets the feedback from the DML operation (such as how many rows were affected by the operation).

**Note:** Any changed blocks in the buffer cache are marked as dirty buffers; that is, the buffers are not the same as the corresponding blocks on the disk. These buffers are not immediately written to disk by the `DBWn` processes.





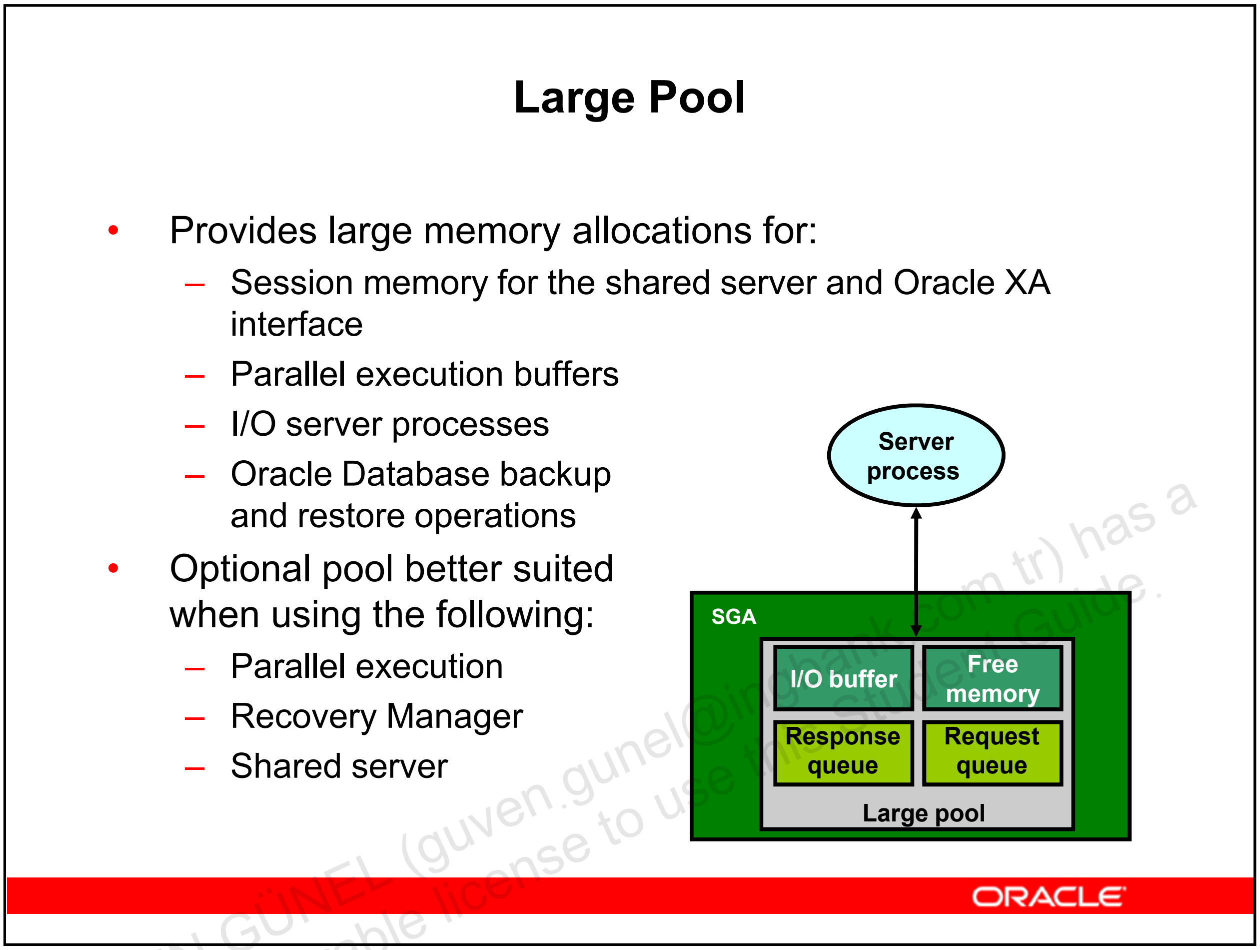

## COMMIT Processing: Example

When COMMIT is issued, the following steps are performed:

1. The server process places a commit record, along with the system change number (SCN), in the redo log buffer. The SCN is a number monotonically incremented and is unique within the database. It is used by Oracle Database 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 Oracle Database to perform consistency checks without depending on the date and time of the operating system.
2. The LGWR background process 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, Oracle Database can guarantee that the changes are not lost even if there is an instance failure.
3. If modified blocks are still in the SGA, and if no other session is modifying them, then the database removes lock-related transaction information from the blocks. This process is known as commit cleanout.
4. The server process provides feedback to the user process about the completion of the transaction.

**Note:** If not done already, DBWn eventually writes the actual changes back to disk based on its own internal timing mechanism.

## Large Pool




### Large Pool

You can configure an optional memory area called the large pool to provide large memory allocations for:

- Session memory for the shared server, the Oracle XA interface (used where transactions interact with more than one database), or parallel execution buffers
- I/O server processes
- Oracle Database backup and restore operations

By allocating the above memory components from the large pool, Oracle Database can use the shared pool primarily for caching the shared part of SQL and PL/SQL constructs. The shared pool was originally designed to store SQL and PL/SQL constructs. Using the large pool avoids fragmentation issues associated with having large and small allocations sharing the same memory area. Unlike the shared pool, the large pool does not have an LRU list.

You should consider configuring a large pool if your instance uses any of the following:

- **Parallel execution:** Parallel query uses shared pool memory to cache parallel execution message buffers.
- **Recovery Manager:** Recovery Manager uses the shared pool to cache I/O buffers during backup and restore operations.

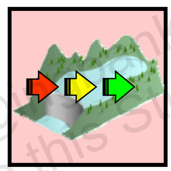
- **Shared server:** In a shared server architecture, the session memory for each client process is included in the shared pool.

## Java Pool and Streams Pool

# Java Pool and Streams Pool

- Java pool memory is used in server memory for all session-specific Java code and data in the JVM.
- Streams pool memory is used exclusively by Oracle Streams to:
  - Store buffered queue messages
  - Provide memory for Oracle Streams processes

Java pool



Streams pool

### Java Pool and Streams Pool

Java pool memory is used for all session-specific Java code and data in the JVM. Java pool memory is used in different ways, depending on the mode in which Oracle Database runs.

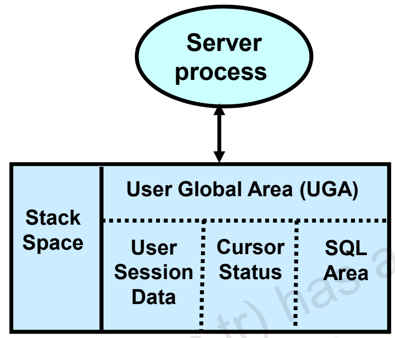
Oracle Streams enables the propagation and management of data, transactions and events in a data stream either within a database, or from one database to another. The Streams pool is used exclusively by Oracle Streams. The Streams pool stores buffered queue messages, and it provides memory for Oracle Streams capture and apply processes.

**Note:** A detailed discussion of Java programming and Oracle Streams is beyond the scope of this course.

## Program Global Area (PGA)

# Program Global Area (PGA)

- PGA is a memory area that contains:
  - Session information
  - Cursor information
  - SQL execution work areas:
    - Sort area
    - Hash join area
    - Bitmap merge area
    - Bitmap create area
- Work area size influences SQL performance.
- Work areas can be automatically or manually managed.




ORACLE



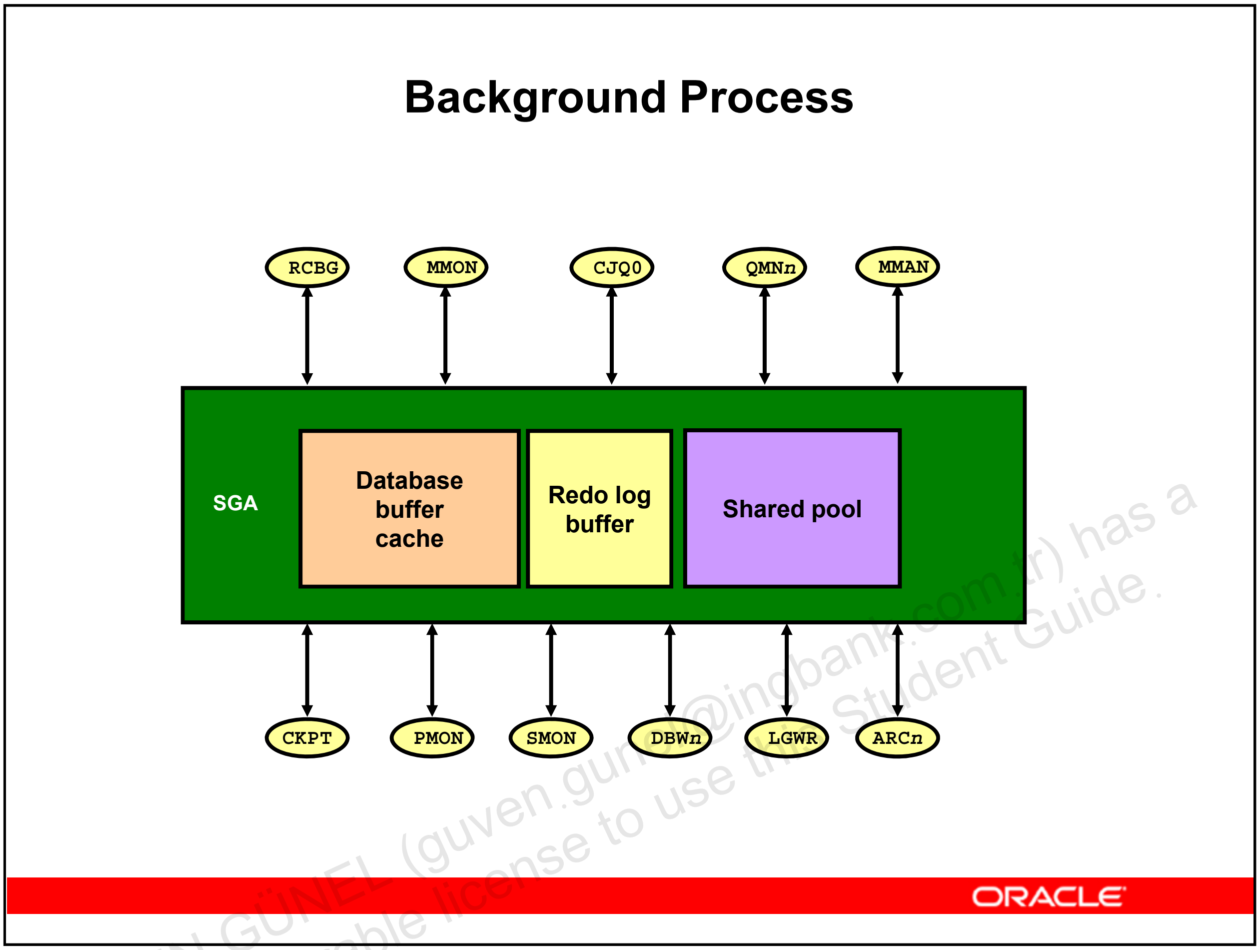


### Program Global Area (PGA)

The PGA can be compared to a temporary countertop workspace used by a file clerk (the server process) to perform a function on behalf of a customer (client process). The clerk clears a section of the countertop, uses the workspace to store details about the customer's request, and then gives up the space when the work is done.

Generally, the PGA memory is divided into the following areas:

- Session memory is the memory allocated to hold a session's variables (logon information) and other information related to the session. For a shared server, the session memory is shared and not private.
- Cursors are handles to private memory structures of specific SQL statements
- SQL work areas are allocated to support memory-intensive operators, such as the ones listed in the slide. Generally, bigger work areas can significantly improve the performance of a particular operator at the cost of higher memory consumption.



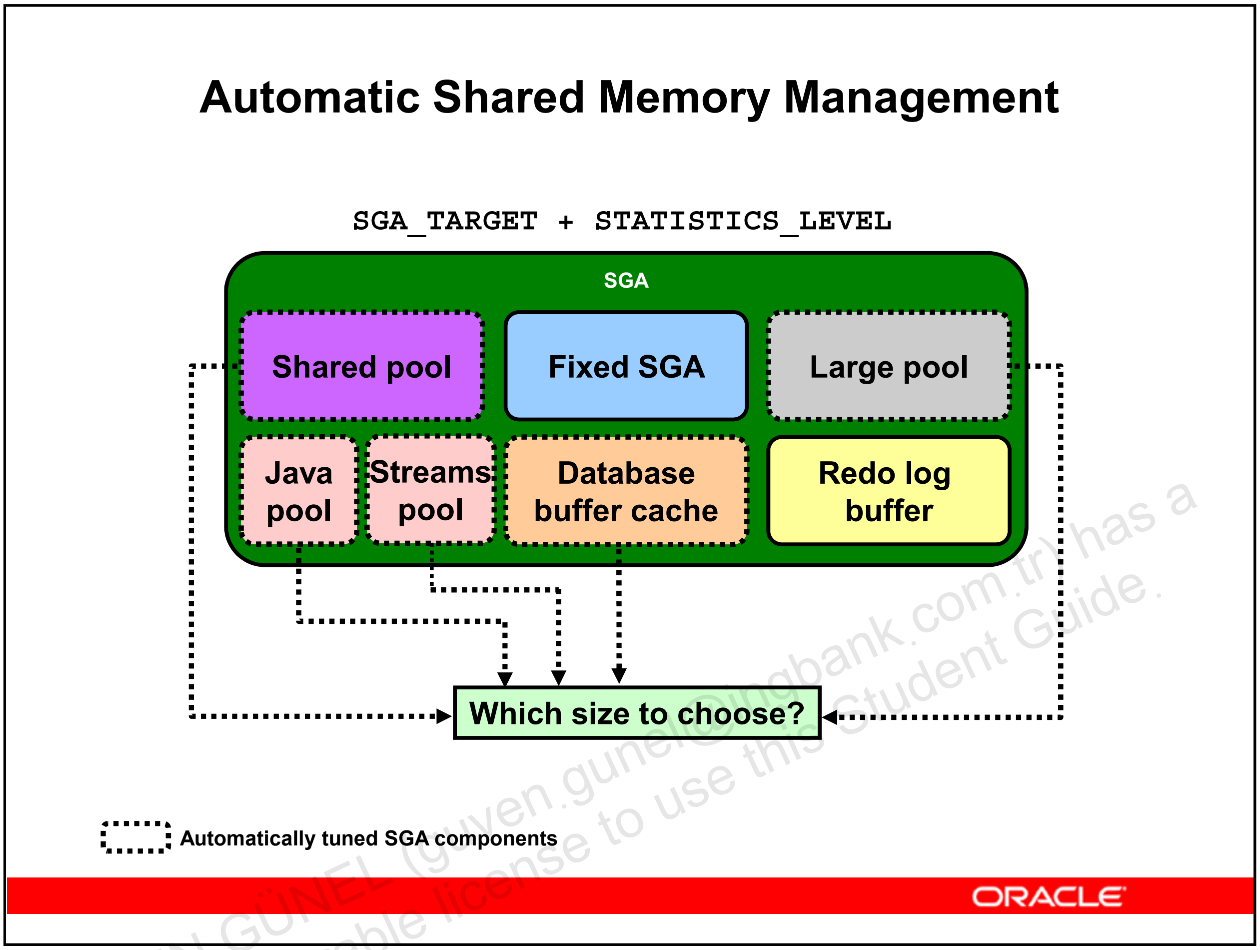



### Background Process

The background processes commonly seen in non-RAC, non-ASM environments can include the following:

- **Database writer process (`DBWn`):** Asynchronously writes modified (dirty) buffers in the database buffer cache to disk
- **Log writer process (`LGWR`):** Writes the recovery information called redo information in the log buffer to a redo log file on disk
- **Checkpoint process (`CKPT`):** Records checkpoint information in control files and each data file header
- **System Monitor process (`SMON`):** Performs recovery at instance startup and cleans up unused temporary segments
- **Process monitor process (`PMON`):** Performs process recovery when a user process fails
- **Result cache background process (`RCBG`):** Used to maintain the result cache in the shared pool
- **Job queue process (`CJQ0`):** Runs user jobs used in batch processing through the Scheduler

- **Archiver processes (`ARCn`):** Copies redo log files to a designated storage device after a log switch has occurred
- **Queue monitor processes (`QMNn`):** Monitors the Oracle Streams message queues
- **Manageability monitoring process (`MMON`):** Performs manageability-related background tasks
- **Memory Manager background process (`MMAN`):** Used to manage SGA and PGA memory components automatically



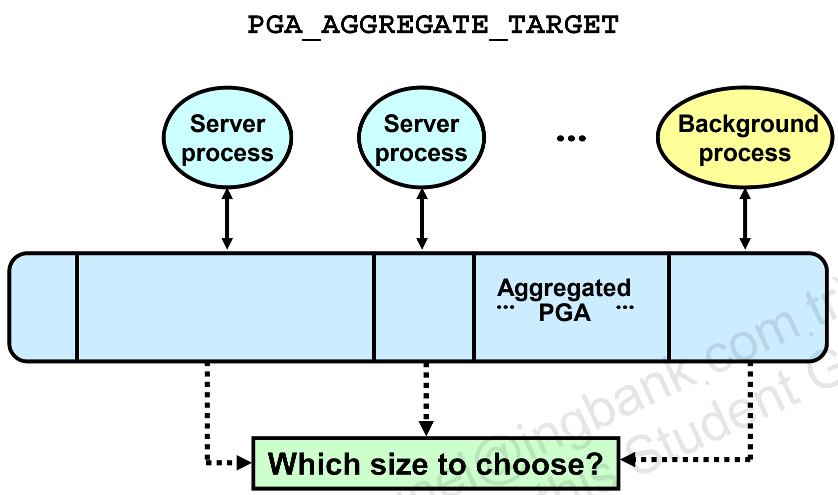

## Automatic Shared Memory Management

You can use the Automatic Shared Memory Management (ASMM) feature to enable the database to automatically determine the size of each of these memory components within the limits of the total SGA size.

The system uses an SGA size parameter (`SGA_TARGET`) that includes all the memory in the SGA, including all the automatically sized components, manually sized components, and any internal allocations during startup. ASMM simplifies the configuration of the SGA by enabling you to specify a total memory amount to be used for all SGA components. The Oracle Database then periodically redistributes memory between the automatically tuned components, according to workload requirements.

**Note:** You must set `STATISTICS_LEVEL` to `TYPICAL` or `ALL` to use ASMM.

## Automated SQL Execution Memory Management






### Automated SQL Execution Memory Management

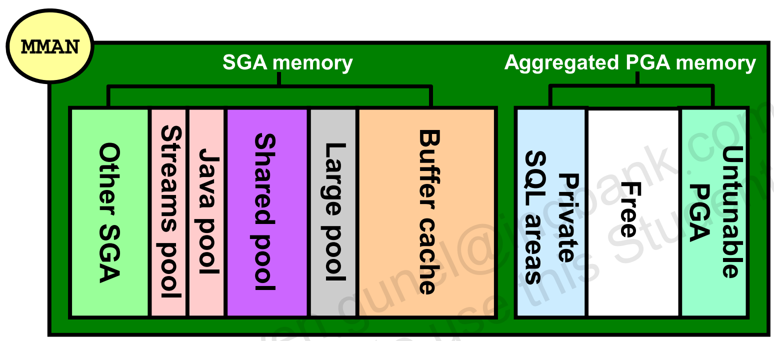
This feature provides an automatic mode for allocating memory to working areas in the PGA. You can use the PGA_AGGREGATE_TARGET parameter to specify the total amount of memory that should be allocated to the PGA areas of the instance's sessions. In automatic mode, working areas that are used by memory-intensive operators (sorts and hash joins) can be automatically and dynamically adjusted.

This feature offers several performance and scalability benefits for decision support system (DSS) workloads and mixed workloads with complex queries. The overall system performance is maximized, and the available memory is allocated more efficiently among queries to optimize both throughput and response time. In particular, the savings that are gained from improved use of memory translate to better throughput at high loads.

**Note:** In earlier releases of Oracle Database server, you had to manually specify the maximum work area size for each type of SQL operator, such as sort or hash join. This proved to be very difficult because the workload changes constantly. Although the current release of Oracle Database supports this manual PGA memory management method that might be useful for specific sessions, it is recommended that you leave automatic PGA memory management enabled.

# Automatic Memory Management

- Sizing of each memory component is vital for SQL execution performance.
- It is difficult to manually size each component.
- Automatic memory management automates memory allocation of each SGA component and aggregated PGA.




MEMORY_TARGET + STATISTICS_LEVEL

ORACLE

## Automatic Memory Management

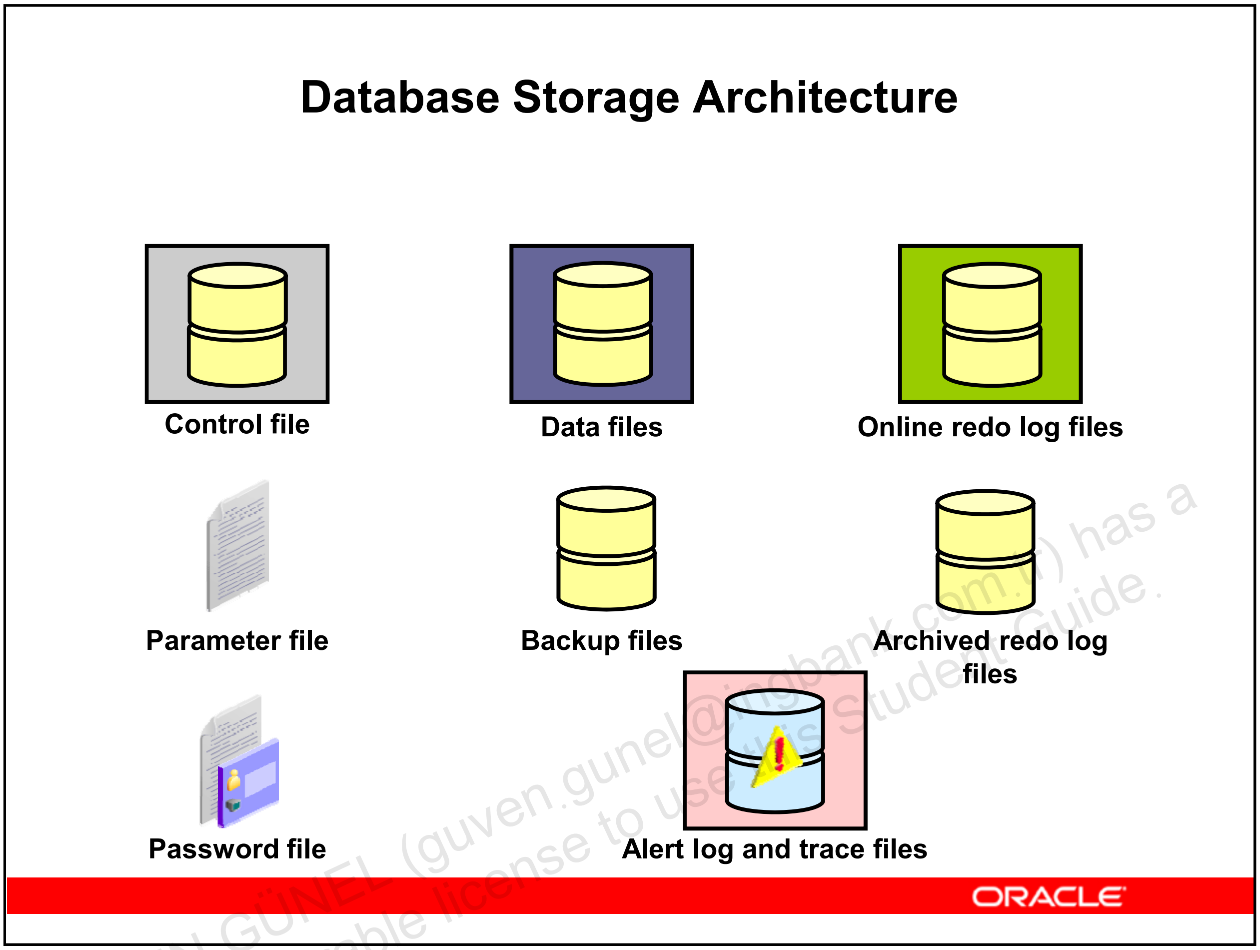
As seen already, the size of the various memory areas of the instance directly impacts the speed of SQL processing. Depending on the database workload, it is difficult to size those components manually.

With Automatic Memory Management, the system automatically adapts the size of each memory's components to your workload memory needs.

Set your `MEMORY_TARGET` initialization parameter for the database instance and the `MMAN` background process automatically tunes to the target memory size, redistributing memory as needed between the internal components of the SGA and between the SGA and the aggregated PGAs.

The Automatic Shared Memory Management feature uses the SGA memory broker that is implemented by two background processes Manageability Monitor (MMON) and Memory Manager (MMAN). Statistics and memory advisory data are periodically captured in memory by MMON. MMAN coordinates the sizing of the memory components according to MMON decisions.

**Note:** Currently, this mechanism is only implemented on Linux, Solaris, HP-UX, AIX, and Windows.




## Database Storage Architecture

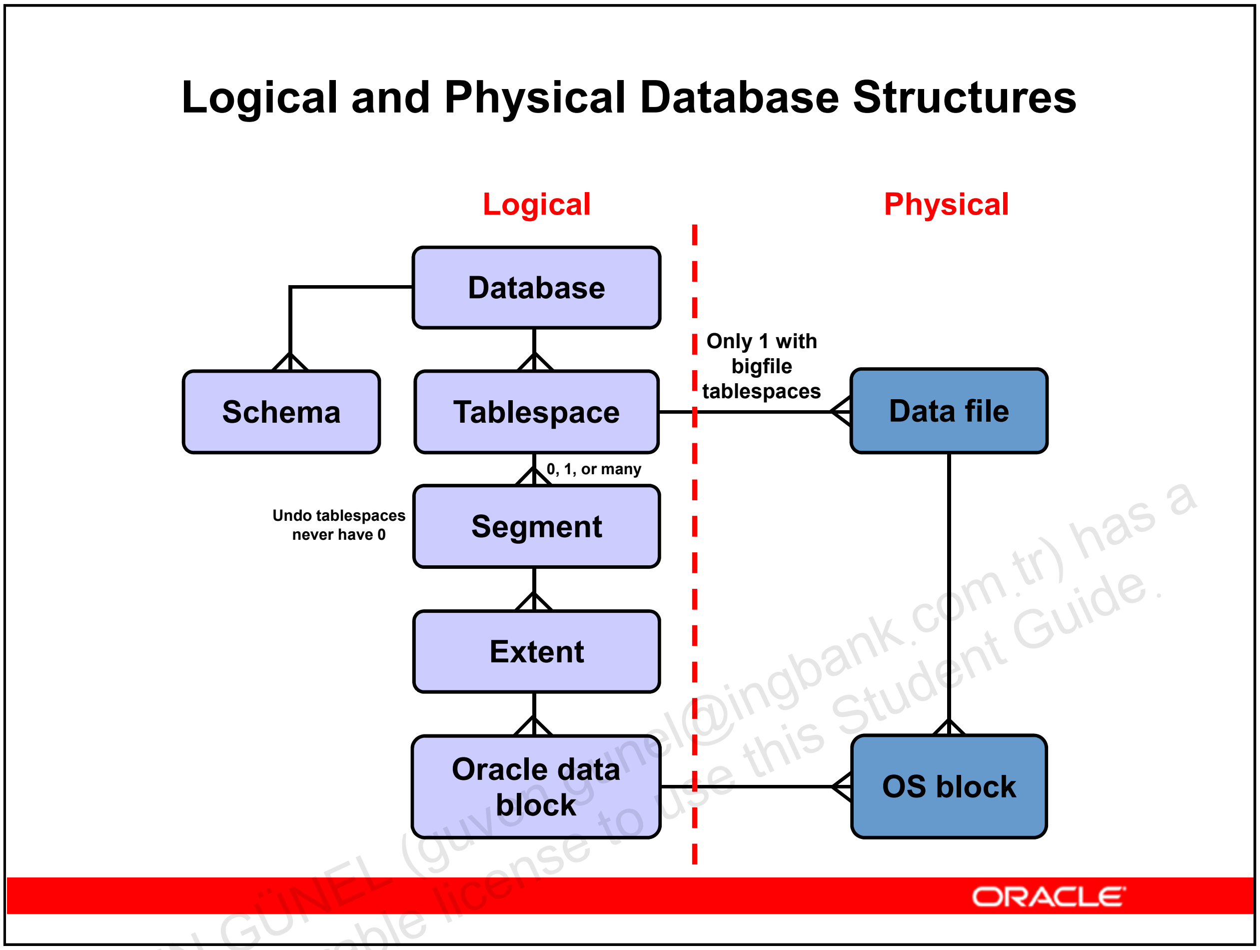
The files that constitute an Oracle database are organized into the following:

- **Control file:** Contain data about the database itself (that is, physical database structure information). These files are critical to the database. Without them, you cannot open data files to access the data in the database.
- **Data files:** Contain the user or application data of the database, as well as metadata and the data dictionary
- **Online redo log files:** Allow for instance recovery of the database. If the database server crashes and does not lose any data files, the instance can recover the database with the information in these files.

The following additional files are important for the successful running of the database:

- **Parameter file:** Is used to define how the instance is configured when it starts up
- **Password file:** Allows `sysdba`, `sysoper`, and `sysasm` to connect remotely to the database and perform administrative tasks
- **Backup files:** Are used for database recovery. You typically restore a backup file when a media failure or user error has damaged or deleted the original file.

- **Archived redo log files:** Contain an ongoing history of the data changes (redo) that are generated by the instance. Using these files and a backup of the database, you can recover a lost data file. That is, archive logs enable the recovery of restored data files.
- **Trace files:** Each server and background process can write to an associated trace file. When an internal error is detected by a process, the process dumps information about the error to its trace file. Some of the information written to a trace file is intended for the developer, whereas other information is for Oracle Support Services.
- **Alert log file:** These are special trace entries. The alert log of a database is a chronological log of messages and errors. Each instance has one alert log file. It is recommended that you review this periodically.




## Logical and Physical Database Structures

The database has logical structures and physical structures.

### Tablespaces

A database is divided into logical storage units called tablespaces, which group related logical structures together. For example, tablespaces commonly group all of an application's objects to simplify some administrative operations. You may have a tablespace for different applications.

### Databases, Tablespaces, and Data Files

The relationship among databases, tablespaces, and data files is illustrated in the slide. 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. If it is a TEMPORARY tablespace instead of a tablespace containing data, the tablespace has a temporary file.

### Schemas

A schema is a collection of database objects that are owned by a database user. Schema objects are the logical structures that directly refer to the database's data. Schema objects include structures, such as tables, views, sequences, stored procedures, synonyms, indexes, clusters, and database links. In general, schema objects include everything that your application creates in the database.

### Data Blocks

At the finest level of granularity, an Oracle database's data is stored in data blocks. One data block corresponds to a specific number of bytes of physical database space on the disk. A data block size is specified for each tablespace when it is created. A database uses and allocates free database space in Oracle data blocks.

### Extents

The next level of logical database space is an extent. An extent is a specific number of contiguous data blocks (obtained in a single allocation) that are used to store a specific type of information.

### Segments

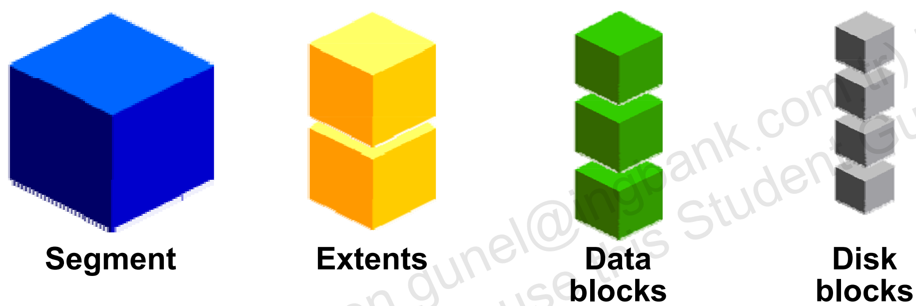
The level of logical database storage above an extent is called a segment. A segment is a set of extents that are allocated for a certain logical structure. Different types of segments include:

- **Data segments:** Each nonclustered, non-index-organized table has a data segment, with the exception of external tables and global temporary tables that have no segments, and partitioned tables in which each table has one or more segments. All of the table's data is stored in the extents of its data segment. For a partitioned table, each partition has a data segment. Each cluster has a data segment. The data of every table in the cluster is stored in the cluster's data segment.
- **Index segments:** Each index has an index segment that stores all of its data. For a partitioned index, each partition has an index segment.
- **Undo segments:** One `UNDO` tablespace is created for each database instance. This tablespace contains numerous undo segments to temporarily store undo information. The information in an undo segment is used to generate read-consistent database information and, during database recovery, to roll back uncommitted transactions for users.
- **Temporary segments:** Temporary segments are created by the Oracle Database when a SQL statement needs a temporary work area to complete execution. When the statement finishes execution, the temporary segment's extents are returned to the instance for future use. Specify either a default temporary tablespace for every user, or a default temporary tablespace that is used across the database.

The Oracle Database dynamically allocates space. When the existing extents of a segment are full, additional extents are added. Because extents are allocated as needed, the extents of a segment may or may not be contiguous on the disk.

# Segments, Extents, and Blocks

- Segments exist in a tablespace.
- Segments are collections of extents.
- Extents are collections of data blocks.
- Data blocks are mapped to disk blocks.




## Segments, Extents, and Blocks

Database objects, such as tables and indexes are stored as segments in tablespaces. 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 an actual file system or disk block on the storage device. Because of this, you do not need to know the physical address of any of the data in your database. This also means that a data file can be striped or mirrored on several disks.

The size of the data block can be set at the time of database creation. The default size of 8 KB is adequate for most databases. If your database supports a data warehouse application that has large tables and indexes, a larger block size may be beneficial.

If your database supports a transactional application in which reads and writes are random, specifying a smaller block size may be beneficial. The maximum block size depends on your OS. The minimum Oracle block size is 2 KB; it should rarely (if ever) be used.

You can have tablespaces with a nonstandard block size. For details, see the *Oracle Database Administrator's Guide*.

## SYSTEM and SYSAUX Tablespaces

- The SYSTEM and SYSAUX tablespaces are mandatory tablespaces that are created at the time of database creation. They must be online.
- The SYSTEM tablespace is used for core functionality (for example, data dictionary tables).
- The auxiliary SYSAUX tablespace is used for additional database components (such as the Enterprise Manager Repository).







### SYSTEM and SYSAUX Tablespaces

Each Oracle Database must contain a SYSTEM tablespace and a SYSAUX tablespace, which are automatically created when the database is created. The system default is to create a smallfile tablespace. You can also create bigfile tablespaces, which enable the Oracle database to manage ultra large files (up to 8 exabytes in size).

A tablespace can be online (accessible) or offline (not accessible). The SYSTEM tablespace is always online when the database is open. It stores tables that support the core functionality of the database, such as the data dictionary tables.

The SYSAUX tablespace is an auxiliary tablespace to the SYSTEM tablespace. The SYSAUX tablespace stores many database components, and it must be online for the correct functioning of all database components.

**Note:** The SYSAUX tablespace may be taken offline for performing tablespace recovery, whereas this is not possible in the case of the SYSTEM tablespace. Neither of them may be made read-only.

## Quiz

The first time an Oracle Database server process requires a particular piece of data, it searches for the data in the:

a. Database Buffer Cache
b. PGA
c. Redo Log Buffer
d. Shared Pool

ORACLE

**Answer: a**

# Quiz

Which of the following is not a database logical structure?

a. Tablespace
b. Data File
c. Schema
d. Segment

ORACLE

**Answer: b**

## Quiz

The `SYSAUX` tablespace is used for core functionality and the `SYSTEM` tablespace is used for additional database components such as the Enterprise Manager Repository.

a. True

b. False

ORACLE

**Answer: b**

# Summary

In this lesson, you should have learned how to:

- List the major architectural components of the Oracle Database server
- Explain memory structures
- Describe background processes
- Correlate logical and physical storage structures

## Practice 1: Overview

This practice covers the following topics:

- Listing the different components of an Oracle Database server
- Looking at some instance and database components directly on your machine