

# 1

## Core Concepts and Tools of the Oracle Database

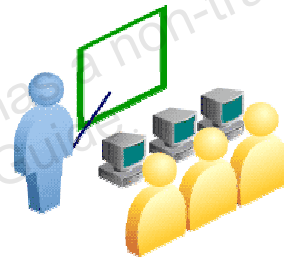
ORACLE

Copyright © 2009, Oracle. All rights reserved.

# Objectives

After completing this lesson, you should be able to:

- Describe the core concepts of the Oracle Database architecture with Automatic Storage Management (ASM)
- Use configuration and management DBA tools
- Describe the technical course environment



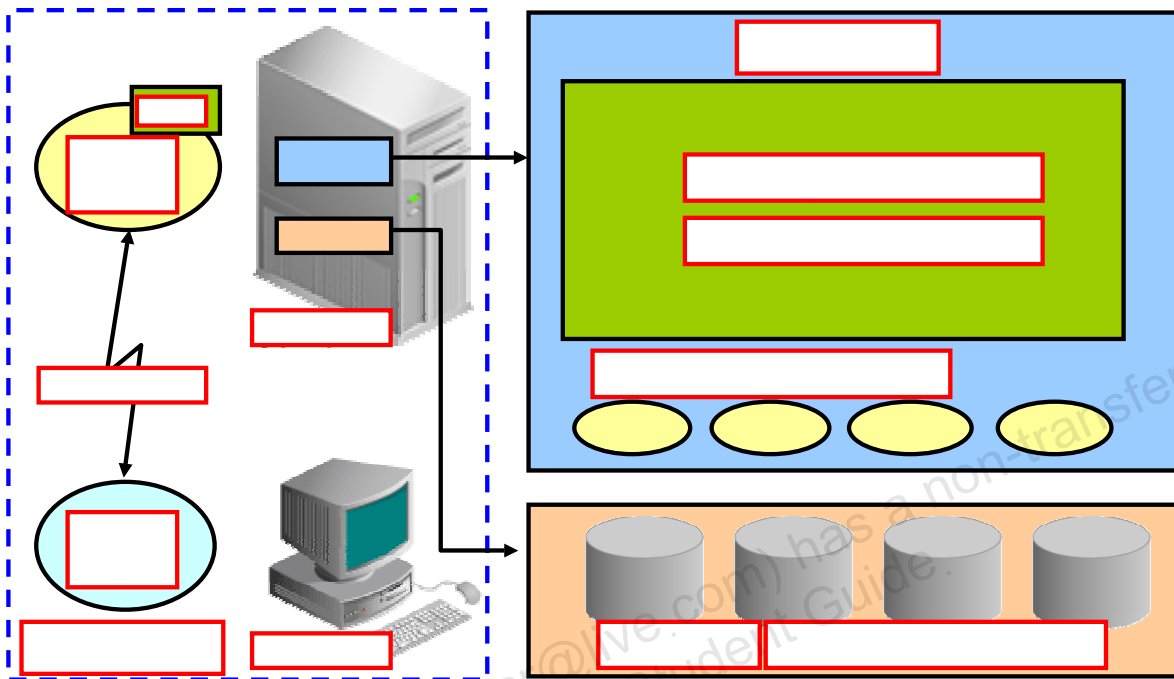
ORACLE

Copyright © 2009, Oracle. All rights reserved.

## Objectives

This lesson reviews the Oracle Database architecture with ASM and provides an overview of the technical environment for this course. In this way, core concepts and DBA tools are introduced.

## Naming the Core Components of an Oracle Database Server



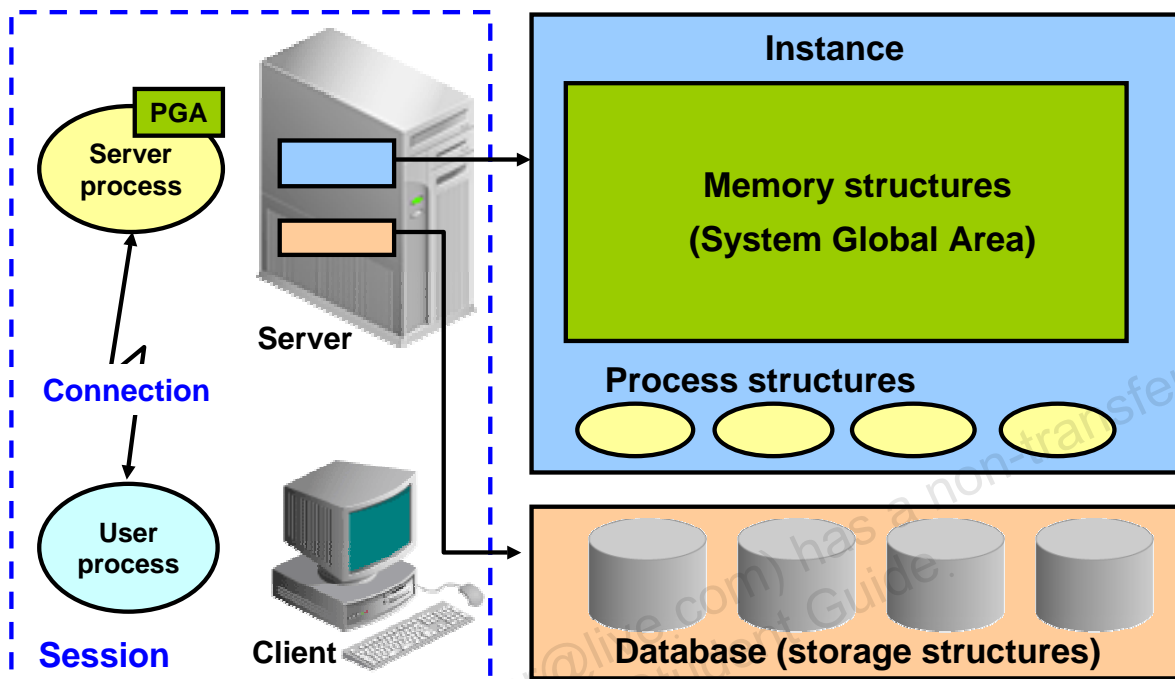
Copyright © 2009, Oracle. All rights reserved.

### Naming the Core Components of an Oracle Database Server

The following are a few sample questions to get you started by naming the core components:

1. The two main components of a basic Oracle Database system: \_\_\_\_\_ and \_\_\_\_\_.
2. The Instance consists of \_\_\_\_\_ and \_\_\_\_\_ processes.
3. The three major structures in the Oracle Database server architecture are: \_\_\_\_\_, \_\_\_\_\_, and \_\_\_\_\_.
4. A session is a connection between the \_\_\_\_\_ and the \_\_\_\_\_.

# Oracle Database Server Architecture Overview



ORACLE

Copyright © 2009, Oracle. All rights reserved.

## Oracle Database Server Architecture

There are three major structures in the Oracle Database server architecture: memory structures, process structures, and storage structures. A basic Oracle database system consists of an Oracle database and a database instance.

The database consists of both physical structures and logical structures. Because the physical and logical structures are separate, the physical storage of data can be managed without affecting access to logical storage structures.

The instance consists of memory structures and background processes associated with that instance. Every time an instance is started, a shared memory area called the System Global Area (SGA) is allocated and the background processes are started. Processes are jobs that work in the memory of computers. A process is defined as a “thread of control” or a mechanism in an operating system that can run a series of steps. After starting a database instance, the Oracle software associates the instance with a specific database. This is called *mounting the database*. The database is then ready to be opened, which makes it accessible to authorized users.

**Note:** The Oracle Automatic Storage Management (ASM) uses the concept of an instance for the memory and process components, but is not associated with a specific database.

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

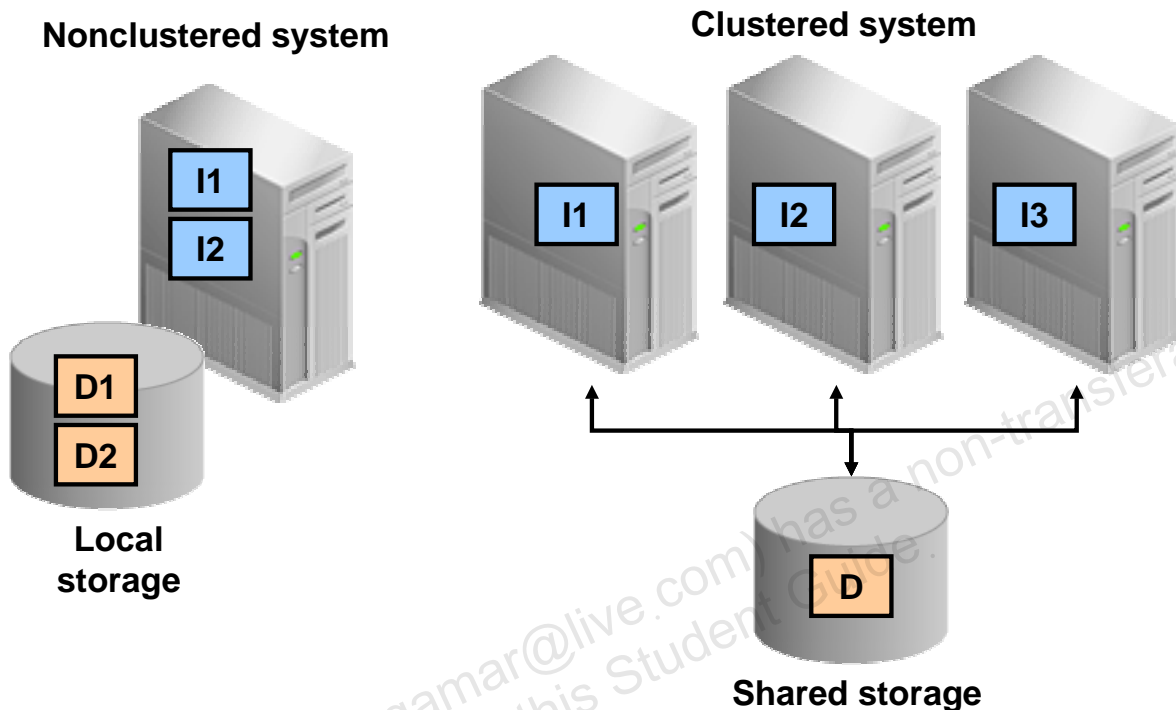
## Oracle Database Architecture Overview (continued)

A *connection* is a communication pathway between a user process and an Oracle Database instance. A communication pathway is established 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 user login to the database instance. For example, when a user starts SQL\*Plus, the user must provide a valid 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.

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.

# Instance-Database Configurations

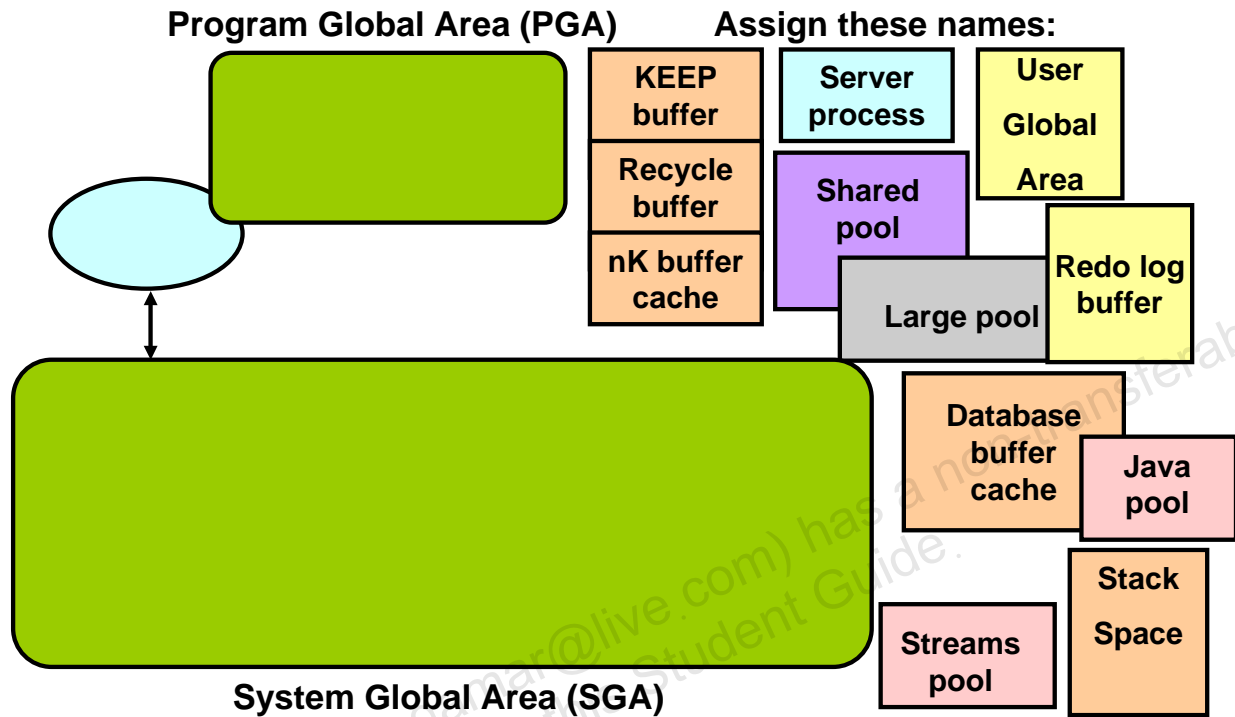


Copyright © 2009, Oracle. All rights reserved.

## Instance-Database Configurations

Each database instance is associated with one and only one database. If there are multiple databases on the same server, there is a separate and distinct database instance for each database. A database instance cannot be shared. A Real Application Clusters (RAC) database usually has multiple instances on separate servers for the same shared database. In this model, the same database is associated with each RAC instance, which preserves the requirement that at most only one database be associated with an instance.

# Naming the Memory Structures of an Oracle Database Instance



Copyright © 2009, Oracle. All rights reserved.

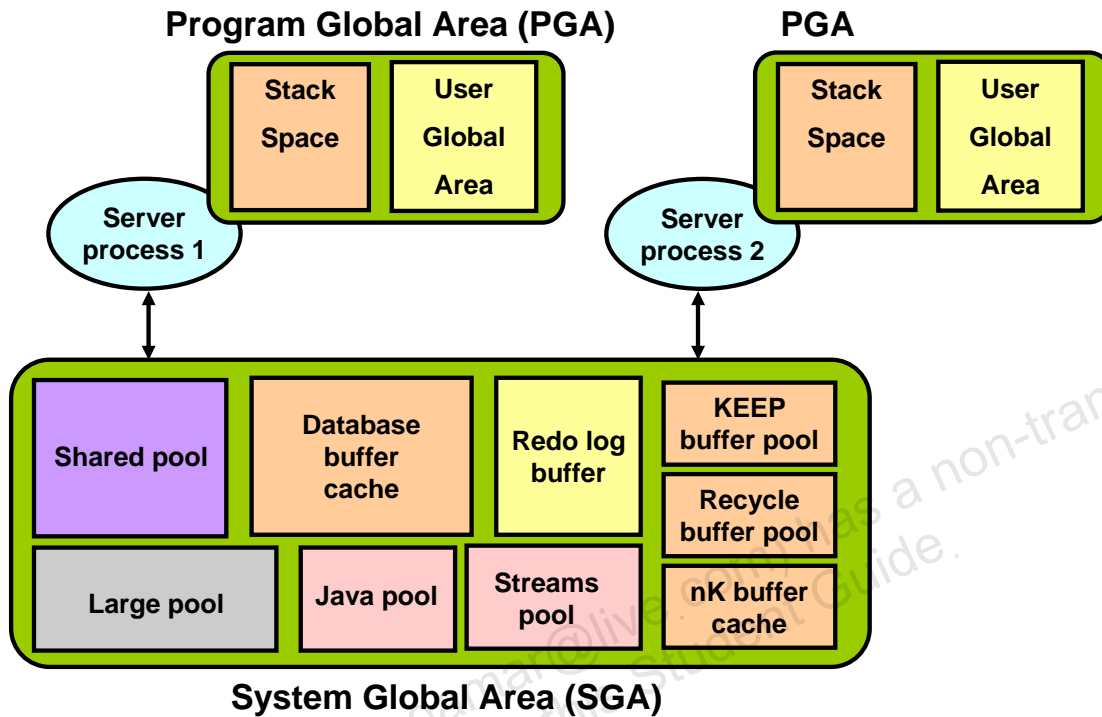
## Naming the Memory Structures of an Oracle Database Instance

To get you started:

1. Which are the components of the PGA: \_\_\_\_\_ and \_\_\_\_\_.
2. Name the main components of the SGA:

- \_\_\_\_\_
- \_\_\_\_\_
- \_\_\_\_\_
- \_\_\_\_\_
- \_\_\_\_\_
- \_\_\_\_\_
- \_\_\_\_\_
- \_\_\_\_\_
- \_\_\_\_\_

# Oracle Database Memory Structures



Copyright © 2009, Oracle. All rights reserved.

## Oracle Database Memory Structures

Oracle Database creates and uses memory structures for various purposes. For example, memory stores program code being 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):** Group of shared memory structures, known as SGA components, that contain data and control information for one Oracle Database instance. The SGA is shared by all server and background processes. Examples of data stored in the SGA include cached data blocks and shared SQL areas.
- **Program Global Areas (PGA):** Memory regions that contain data and control information for a server or background process. A PGA is nonshared memory created by Oracle Database when a server or background process is started. Access to the PGA is exclusive to the server process. Each server process and background process has its own PGA.



## Oracle Database Memory Structures (continued)

The SGA is the memory area that contains data and control information for the instance. The SGA includes the following data structures:

- **Shared pool:** Caches various constructs that can be shared among users
- **Database buffer cache:** Caches blocks of data retrieved from the database
- **KEEP buffer pool:** A specialized type of database buffer cache that is tuned to retain blocks of data in memory for long periods of time
- **Recycle buffer pool:** A specialized type of database buffer cache that is tuned to recycle or remove block from memory quickly
- **nK buffer cache:** One of several specialized database buffer caches designed to hold block sizes different than the default database block size
- **Redo log buffer:** Caches redo information (used for instance recovery) until it can be written to the physical redo log files stored on the disk
- **Large pool:** Optional area that provides large memory allocations for certain large processes, such as Oracle backup and recovery operations, and I/O server processes
- **Java pool:** Used for all session-specific Java code and data in the Java Virtual Machine (JVM)
- **Streams pool:** Used by Oracle Streams to store information required by capture and apply

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

A Program Global Area (PGA) is a memory region that contains data and control information for each server process. An Oracle server process services a client's requests. Each server process has its own private PGA that is allocated when the server process is started. Access to the PGA is exclusive to that server process, and the PGA is read and written only by the Oracle code acting on its behalf. The PGA is divided into two major areas: stack space and the user global area (UGA).

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

The Oracle database uses initialization parameters to create and manage memory structures. The simplest way to manage memory is to allow the database to automatically manage and tune it for you. To do so (on most platforms), you only have to set a target memory size initialization parameter (MEMORY\_TARGET) and a maximum memory size initialization parameter (MEMORY\_MAX\_TARGET).

# Process Architecture

- User process
  - Is the application or tool that connects to the Oracle Database
- Database processes
  - Server process: Connects to the Oracle instance and is started when a user establishes a session
  - Background processes: Are started when an Oracle instance is started
- Daemon/Application processes
  - Networking listeners
  - Grid Infrastructure daemons

ORACLE

Copyright © 2009, Oracle. All rights reserved.

## Process Architecture

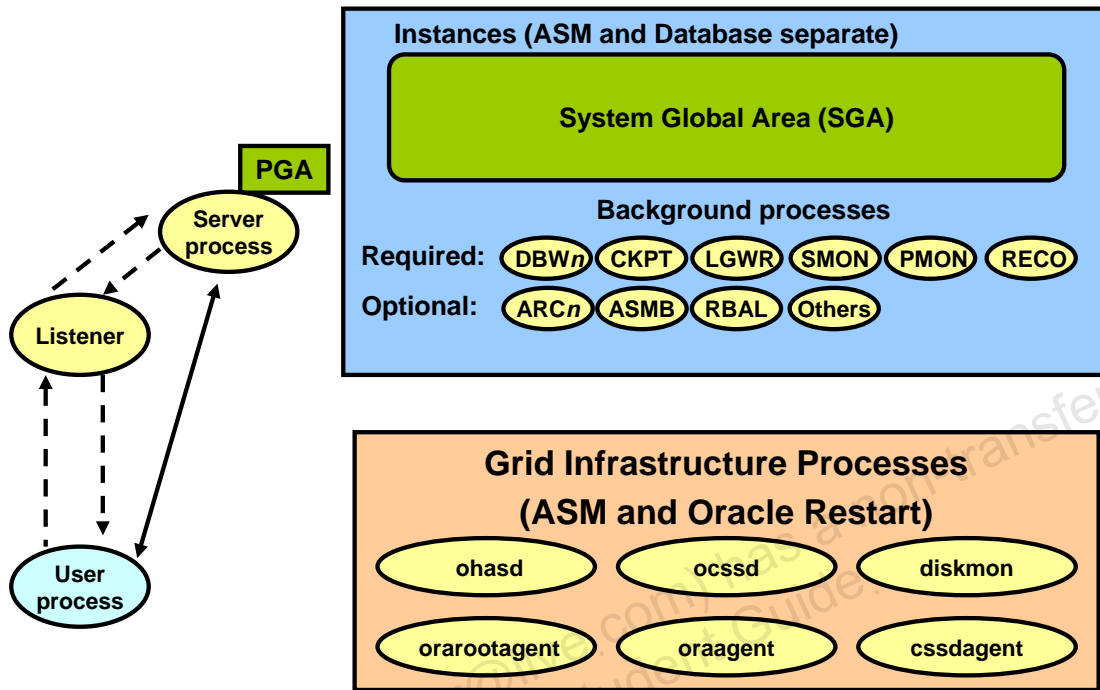
The processes in an Oracle Database system can be divided into three major groups:

- User processes that run the application or Oracle tool code
- Oracle Database processes that run the Oracle database server code (including server processes and background processes)
- Oracle daemons and application processes not specific to a single database

When a user runs an application program or an Oracle tool such as SQL\*Plus, the term *user process* is used to refer to the user's application. The user process may or may not be on the database server machine. Oracle Database also creates a *server process* to execute the commands issued by the user process. In addition, the Oracle server also has 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 perform other required tasks. The process structure varies for different Oracle Database configurations, depending on the operating system and the choice of Oracle Database options. The code for connected users can be configured as a dedicated server or a shared server.

- **Dedicated server:** For each session, the database application is run by a user process that is served by a dedicated server process that executes Oracle database server code.
- **Shared server:** Eliminates the need for a dedicated server process for each connection. A dispatcher directs multiple incoming network session requests to a pool of shared server processes. A shared server process serves any client request.

# Process Structures



ORACLE

Copyright © 2009, Oracle. All rights reserved.

## Process Structures

### Server Processes

Oracle Database creates server processes to handle the requests of user processes connected to the instance. The user process represents the application or tool that connects to the Oracle database. It may be on the same machine as the Oracle database or it may exist on a remote client and utilize a network to reach the Oracle database. The user process first communicates with a listener process that creates a server process in a dedicated environment.

Server processes created on behalf of each user's application can perform one or more of the following:

- Parse and run SQL statements issued through the application
- Read necessary data blocks from data files on disk into the shared database buffers of the SGA (if the blocks are not already present in the SGA)
- Return results in such a way that the application can process the information

### Background Processes

To maximize performance and accommodate many users, a multiprocess Oracle Database system uses some additional Oracle Database processes called *background processes*. An Oracle Database instance can have many background processes.

## Process Structures (continued)

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

- Database writer process (DBW $n$ )
- Log writer process (LGWR)
- Checkpoint process (CKPT)
- System monitor process (SMON)
- Process monitor process (PMON)
- Recoverer process (RECO)
- Job queue coordinator (CJQ0)
- Job slave processes (J $nnn$ )
- Archiver processes (ARC $n$ )
- Queue monitor processes (QMN $n$ )

Other background processes may be found in more advanced configurations such as RAC. See the V\$BGPROCESS view for more information on the background processes.

Some background processes are created automatically when an instance is started, whereas others are started as required.

Other process structures are not specific to a single database, but rather can be shared among many databases on the same server. The Grid Infrastructure and networking processes fall into this category.

Oracle Grid Infrastructure processes on Linux and Unix systems include the following:

- ohasd: Oracle High Availability Service daemon that is responsible to starting Oracle Clusterware processes
- ocssd: Cluster Synchronization Service daemon
- diskmon: Disk Monitor daemon that is responsible for input and output fencing for HP Oracle Exadata Storage Server.
- cssdagent: Starts, stops and check the status of the CSS daemon, ocssd.
- oraagent: Extend clusterware to support Oracle-specific requirements and complex resources
- orarootagent: A specialized Oracle agent process that that helps manage resources owned by root, such as the network.

## Adding Process Names

- |   |                                   |
|---|-----------------------------------|
| 1. The _____ process writes the dirty buffers to the data files.                                  | A. Checkpoint process (CKPT)      |
| 2. The _____ process writes the redo entries to the online redo log files.                        | B. System monitor process (SMON)  |
| 3. The _____ process writes checkpoint information in the control file and each data file header. | C. Recoverer process (RECO)       |
| 4. The _____ process performs recovery on instance startup.                                       | D. Log writer process (LGWR)      |
| 5. The _____ process performs process recovery when a user process fails.                         | E. Archiver processes (ARCn)      |
| 6. The _____ process resolves in-doubt distributed transactions.                                  | F. Process monitor process (PMON) |
| 7. The _____ processes copy redo log files to a designated storage device.                        | G. Database writer process (DBWn) |

ORACLE

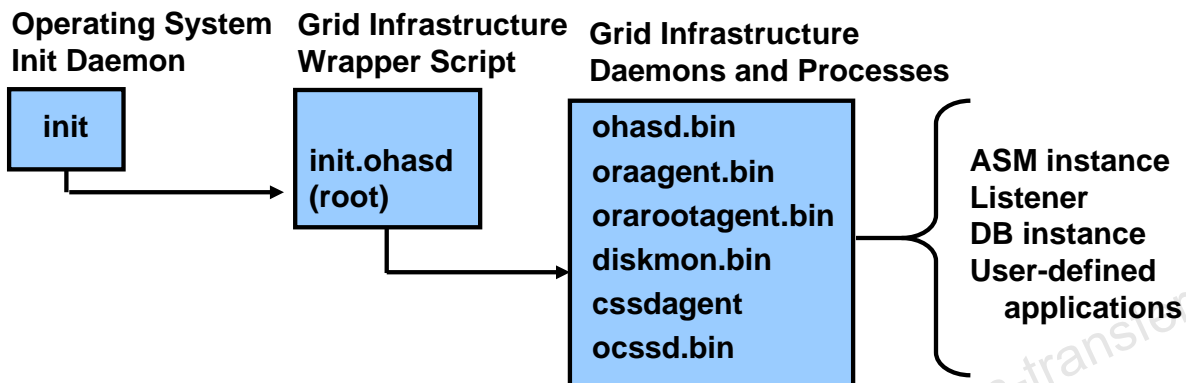
Copyright © 2009, Oracle. All rights reserved.

### Adding Process Names

After you fill in the process names, see “Appendix A Solutions” for possible answers.

# Process Startup Sequence

- Oracle Grid Infrastructure is started by the OS init daemon.



- Oracle Grid Infrastructure installation modifies the `/etc/inittab` file to ensure startup every time the machine is started in the corresponding run level.

```
# cat /etc/inittab
..
h1:35:respawn:/etc/init.d/init.ohasd run >/dev/null 2>&1 </dev/null
```

ORACLE

Copyright © 2009, Oracle. All rights reserved.

## Process Startup Sequence

During the installation of Oracle Grid Infrastructure, entries are placed in the operating system `/etc/inittab` file to start a wrapper script. The wrapper script is responsible for setting up environment variables and then starting the Oracle Grid Infrastructure daemons and processes.

When a command is used to stop Oracle Grid Infrastructure, the daemons will be stopped, but the wrapper script process will remain running.

The format of the UNIX `/etc/inittab` file is as follows:

```
id : run levels : action : process with parameters
```

The wrapper script is started with the `respawn` action so it will be restarted whenever it is terminated.

Some of the Oracle Grid Infrastructures daemons will be running under the root user with real time priority, and others will be running under the Grid Infrastructure owner with user-mode priorities after they are started. On a Windows platform, operating system services are used instead of wrapper initialization scripts and the daemons are executable binaries.

**Note:** It is not supported to execute the wrapper script directly.

## Database Storage Architecture



**Control files**



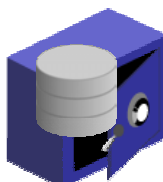
**Data files**



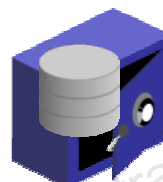
**Online redo log files**



**Parameter file**



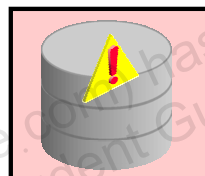
**Backup files**



**Archived redo log files**



**Password file**



**Alert log and trace files**

ORACLE

Copyright © 2009, Oracle. All rights reserved.

## Database Storage Architecture

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

- **Control files:** 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. It can also contain metadata related to backups.
- **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 to 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 instance 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.

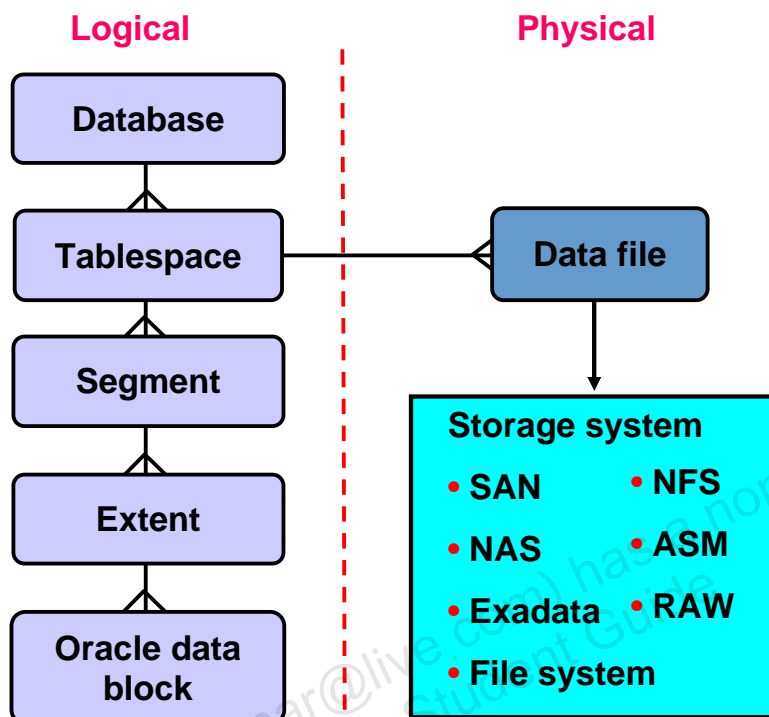
## Database Storage Architecture (continued)

- **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 database administrator, 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. Oracle recommends that you review the alert log periodically.

**Note:** Parameter, password, alert, and trace files are covered in other lessons.



# Logical and Physical Database Structures



ORACLE

Copyright © 2009, Oracle. All rights reserved.

## Logical and Physical Database Structures

The database has logical structures and physical structures.

### Databases, Tablespaces, and Data Files

The relationship among databases, tablespaces, and data files is illustrated in the slide. Each database is logically divided into two or more tablespaces. One or more data files are explicitly created for each tablespace to physically store the data of all segments in a tablespace. If it is a TEMPORARY tablespace, it has a temporary file instead of a data file. A tablespace's data file can be physically stored on any supported storage technology.

### Tablespaces

A database is divided into logical storage units called *tablespaces*, which group related logical structures or data files together. For example, tablespaces commonly group all of an application's segments to simplify some administrative operations.

### 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 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.

## Logical and Physical Database Structures (continued)

### Extents

The next level of logical database space is an *extent*. An extent is a specific number of contiguous Oracle data blocks (obtained in a single allocation) that are used to store a specific type of information. Oracle data blocks in an extent are logically contiguous but can be physically spread out on disk because of RAID striping and file system implementations.

### Segments

The level of logical database storage above an extent is called a *segment*. A segment is a set of extents allocated for a certain logical structure. For example:

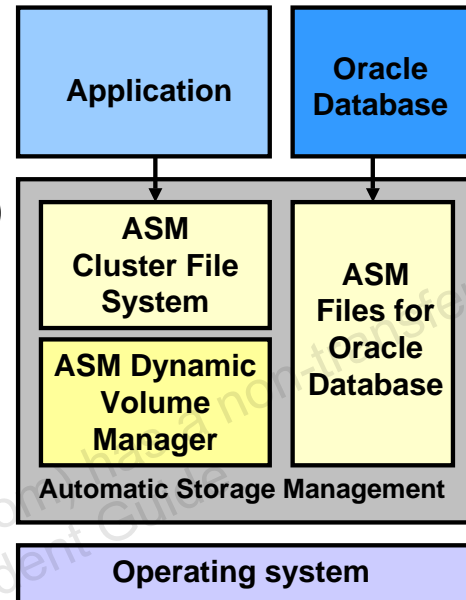
- **Data segments:** Each nonclustered, non-index-organized table has a data segment, with the exception of external tables, global temporary tables, and partitioned tables in which each table has one or more segments. All of the table's data is stored in the extents of its data segment. For a partitioned 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 database-wide.

**Note:** There are other types of segments not listed above. There are also schema objects such as views, packages, triggers, etc. that are not considered segments even though they are database objects. A segment owns its respective disk space allocation. The other objects exist as rows stored in a system metadata segment.

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, and they can come from different data files belonging to the same tablespace.

# Automatic Storage Management

- Is a portable and high-performance cluster file system
- Manages Oracle database files
- Manages application files with ASM Cluster File System (ACFS)
- Spreads data across disks to balance load
- Mirrors data in case of failures
- Solves storage-management challenges



ORACLE

Copyright © 2009, Oracle. All rights reserved.

## Automatic Storage Management

Automatic Storage Management (ASM) provides vertical integration of the file system and the volume manager for Oracle database files. ASM can provide management for single symmetric multiprocessing (SMP) machines or across multiple nodes of a cluster for Oracle Real Application Clusters (RAC) support.

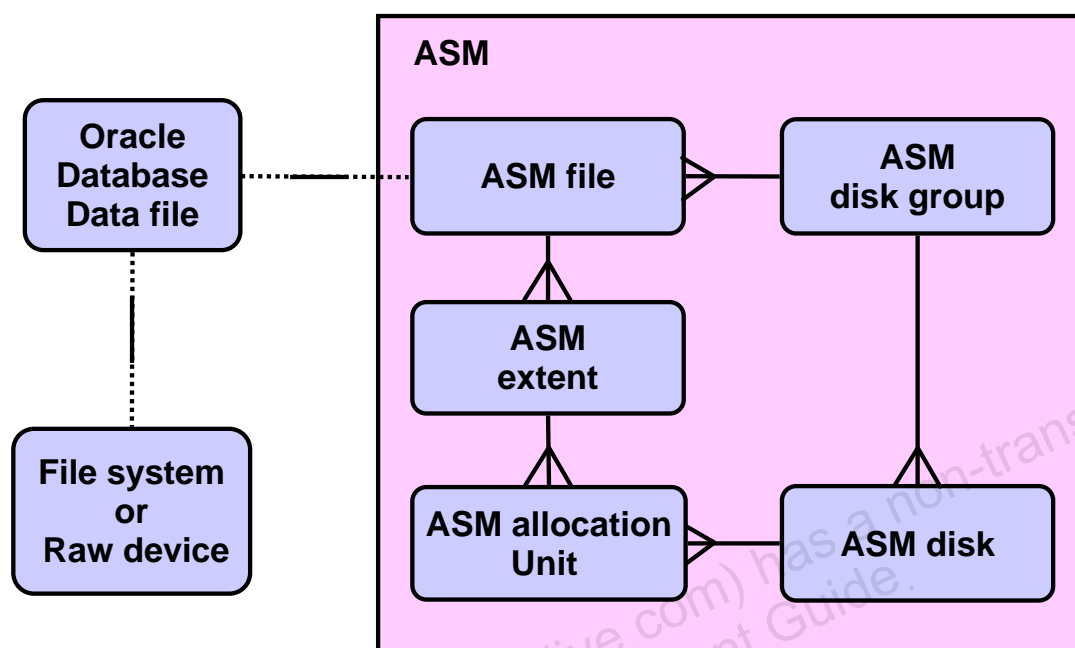
Oracle ASM Cluster File System (ACFS) is a multi-platform, scalable file system, and storage management technology that extends ASM functionality to support application files outside of the Oracle Database such as executables, reports, BFILEs, video, audio, text, images, and other general-purpose application file data.

ASM distributes input/output (I/O) load across all available resources to optimize performance while removing the need for manual I/O tuning. ASM helps DBAs manage a dynamic database environment by enabling them to increase the database size without having to shut down the database to adjust storage allocation.

ASM can maintain redundant copies of data to provide fault tolerance, or it can be built on top of vendor-supplied storage mechanisms. Data management is done by selecting the desired reliability and performance characteristics for classes of data rather than with human interaction on a per-file basis.

ASM capabilities save the DBA's time by automating manual storage and thereby increasing the administrator's ability to manage more and larger databases with increased efficiency.

# ASM Storage Components



ORACLE

Copyright © 2009, Oracle. All rights reserved.

## ASM Storage Components

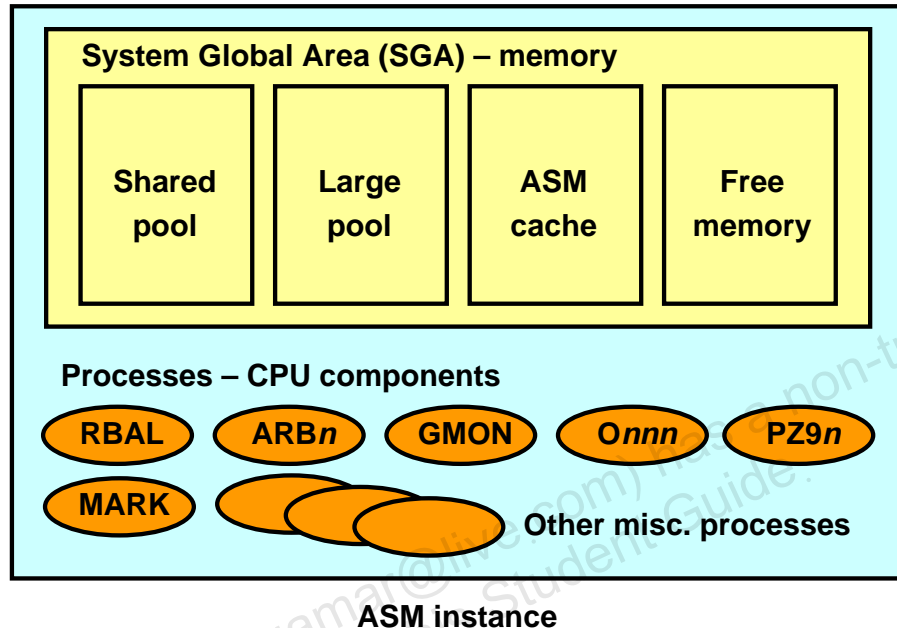
ASM does not eliminate any existing database functionality. Existing databases are able to operate as they always have. New files may be created as ASM files, whereas existing ones are administered in the old way or can be migrated to ASM.

The diagram illustrates the relationships between an Oracle database data file and the ASM storage components. The crow's foot notation represents a one-to-many relationship. An Oracle database data file has a one-to-one relationship with either a file stored on the operating system in a file system or an ASM file.

An Oracle ASM disk group is a collection of one or more Oracle ASM disks managed as a logical unit. The data structures in a disk group are self-contained using some of the space for metadata needs. Oracle ASM disks are the storage devices provisioned to an Oracle ASM disk group and can be physical disk or partitions, a Logical Unit Number (LUN) from a storage array, a logical volume (LV), or a network-attached file. Each ASM disk is divided into many ASM allocation units, the smallest contiguous amount of disk space that ASM allocates. When you create an ASM disk group, you can set the ASM allocation unit size to 1, 2, 4, 8, 16, 32, or 64 MB depending on the disk group compatibility level. One or more ASM allocation units forms an ASM extent. An Oracle ASM extent is the raw storage used to hold the contents of an Oracle ASM file. An Oracle ASM file consists of one or more file extents. Variable extent sizes of 1\*AU size, 4\*AU size, and 16\*AU size are used for supporting very large ASM files.

# ASM Instance

The ASM Instance is the process and memory components for ASM.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

## ASM Instance

Every time ASM or a database is started, a shared memory area called the system global area (SGA) is allocated and Oracle ASM or database background processes are started. The combination of the background processes and the SGA is called an Oracle ASM instance or an Oracle Database instance. The instance represents the CPU and RAM components of a running ASM environment.

The SGA in an ASM instance is different in memory allocation and usage than the SGA in a database instance. The SGA in the ASM instance is divided into four primary areas as follows:

- **Shared Pool:** Used for metadata information
- **Large Pool:** Used for parallel operations
- **ASM Cache:** Used for reading and writing blocks during rebalance operations
- **Free Memory:** Unallocated memory available

The minimum recommended amount of memory for an ASM instance is 256 MB. Automatic memory management is enabled by default on an ASM instance and will dynamically tune the sizes of the individual SGA memory components. The amount of memory that is needed for an ASM instance will depend on the amount of disk space being managed by ASM.

The second part of the ASM instance is the background processes. An ASM instance can have many background processes; not all are always present.

## ASM Components: ASM Instance (continued)

The background processes specific to ASM functionality are covered in the next slide. There are required background processes and optional background processes. Some of these processes may include the following:

- **ARCn:** The archiver processes
- **CKPT:** The checkpoint process
- **DBWn:** The database writer processes
- **DIAG:** The diagnosability process
- **Jnnn:** Job queue processes
- **LGWR:** The log writer process
- **PMON:** The process monitor process
- **PSP0:** The process spawner process
- **QMNn:** The queue monitor processes
- **RECO:** The recoverer process
- **SMON:** The system monitor process
- **VKTM:** The virtual keeper of time process
- **MMAN:** The memory manager process

The above list of processes is not complete. For the ASM instance, these processes will not always perform the same tasks as they would in a database instance. For example, the LGWR process in a database instance is responsible for copying change vectors from the log buffer section of the SGA to the online redo logs on disk. The ASM instance does not contain a log buffer in its SGA, nor does it use online redo logs. The LGWR process in an ASM instance copies logging information to an ASM disk group.

If ASM is clustered, then additional processes related to cluster management will be running in the ASM instance. Some of these processes include the following:

- **LMON:** The global enqueue service monitor process
- **LMDn:** The global enqueue service daemons
- **LMSn:** The global cache service processes
- **LCKn:** The lock processes

For more about managing the ASM instance, see Appendix D and the documentation.

# DBA Configuration Tools

Setting up the technical environment for this course included the following tasks and tools:

- Installing and configuring the Oracle Grid Infrastructure for a stand-alone server with the OUI, including:
  - Configuring a listener
  - Creating an ASM instance (+ASM) and configuring the +DATA disk group
  - Configuring Oracle Restart
- Creating and configuring additional ASM disk groups (such as +FRA) with asmca
- Installing the Oracle Database 11g software with OUI
- Creating the orcl Oracle database with dbca

**Note:** These tasks have already been performed for you.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

## DBA Configuration Tools

There are several configuration tools available to assist DBAs with the configuration and management of their Oracle environment. Some of these tools include:

- **Oracle Universal Installer (OUI):** Installs your Oracle software and options. Depending on the product being installed and the options specified during installation, the OUI can automatically call other tools to perform additional tasks, such as the creation of an Oracle database.
- **Oracle ASM Configuration Assistant (ASMCA):** Supports installing and configuring ASM instances, disk groups, volumes, and Oracle Automatic Storage Management Cluster File System (Oracle ACFS)
- **Oracle Database Configuration Assistant (DBCA):** Allows you to create and delete Oracle databases as well as modify database options and manage database templates. This utility is called from the OUI during the installation of the Oracle database software if you chose to create a database as part of that installation.
- **Net Configuration Assistant (NETCA):** Configures listeners and naming methods, which are critical components of an Oracle environment

## DBA Configuration Tools (continued)

**Note:** Oracle Restart is installed and configured as part of an Oracle Grid Infrastructure for stand-alone server installation. Oracle Restart manages Oracle component dependencies and automatically restarts the various Oracle components after a hardware or software failure or whenever your database host computer restarts. If the Oracle Grid Infrastructure installation is done before the installation of the Oracle database software, any databases created will be automatically configured with Oracle Restart.



# Management Framework and Related DBA Tools

The Oracle database management framework includes:

- Database instance
- Listener
- Management interface:
  - Management agent (when using Grid Control)
  - Database Control

Related tools and commands include:  
SQL\*Plus:

```
SQL> startup
SQL> shutdown immediate
```

Listener Control utility:

```
$ lsnrctl status
```

Enterprise Manager Control utility:

```
$ emctl status dbconsole
$ emctl start dbconsole
$ emctl stop dbconsole
```

ORACLE

Copyright © 2009, Oracle. All rights reserved.

## Management Framework and Related DBA Tools

There are three major components of the Oracle database management framework:

- The database instance that is being managed
- A listener that allows connections to the database
- The management interface. This may be either a management agent running on the node where the database server runs (which connects it to Oracle Enterprise Manager Grid Control) or the stand-alone Oracle Enterprise Manager Database Control. This is also referred to as the *Database Console*.

Related tools and commands to start, stop, or check the status are:

- For the database instance, in SQL\*Plus: `startup` and `shutdown immediate`
- For the listener: `lsnrctl status`
- For Enterprise Manager Database Control:
  - `emctl dbconsole status`
  - `emctl dbconsole start`
  - `emctl dbconsole stop`

## Facilitating Database Management with Oracle Restart

- Restarting Oracle components when the host computer restarts or after hardware or software failure
- Monitoring components and restarting them, if needed
- For single-instance environments
- Considering component dependencies:
  - Mounting disk groups and starting the ASM instance before starting the database instance
  - Soft dependency between the database instance and the listener
- Starting Oracle Restart with the `crsctl` utility
- Managing Oracle Restart components with the `srvctl` utility

```
$ srvctl stop database -d orcl -o abort
```

ORACLE

Copyright © 2009, Oracle. All rights reserved.

### Facilitating Database Management with Oracle Restart

- With Oracle Restart, the various Oracle components are automatically restarted after a hardware or software failure or whenever your database host computer restarts.
- Oracle Restart performs periodic checks to monitor the health of these components. If a check operation fails for a component, the component is shut down and restarted.
- Oracle Restart is used in single-instance (nonclustered) environments only. For Oracle Real Application Clusters (Oracle RAC) environments, the functionality to automatically restart components is provided by Oracle Clusterware.
- Oracle Restart ensures that Oracle components are started in the proper order, considering component dependencies. For example, if database files are stored in ASM disk groups, then before starting the database instance, Oracle Restart ensures that the ASM instance is started and the required disk groups are mounted. Likewise, if a component must be shut down, Oracle Restart ensures that dependent components are cleanly shut down first.
- Oracle Restart also manages the soft dependency between database instances and the Oracle Net listener (the listener). When a database instance is started, Oracle Restart attempts to start the listener. If the listener startup fails, the database is still started. If the listener later fails, Oracle Restart does not shut down and restart any database instances.
- You start Oracle Restart with the Clusterware Control (`crsctl`) utility.
- Oracle Restart includes the Server Control (`srvctl`) utility that you use to start and stop Oracle Restart–managed components.

## Facilitating Database Management with Oracle Restart (continued)

**Note:** The `srvctl` utility is located in both the `$ORACLE_HOME/bin` directory for the Grid Infrastructure software and the `$ORACLE_HOME/bin` directory for the Oracle database software. You should use the `srvctl` utility from the Oracle database software when starting the Oracle database. You should use the `srvctl` utility from the Grid Infrastructure software when starting the ASM instance or the listener.

## Quiz

Oracle Restart is installed and configured as part of an Oracle Grid Infrastructure for stand-alone server installation.

1. True
2. False

ORACLE

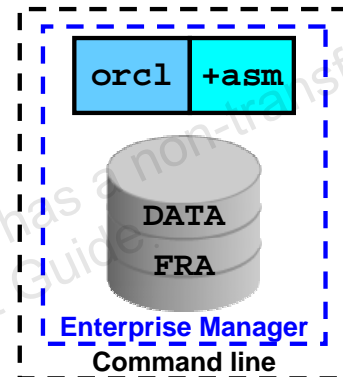
Copyright © 2009, Oracle. All rights reserved.

**Answer: 1**

# Summary

In this lesson, you should have learned how to:

- Describe the core concepts of the Oracle Database architecture with ASM
- Determine which DBA configuration and management tools to use for which task
- Describe the technical course environment



ORACLE

Copyright © 2009, Oracle. All rights reserved.

## Summary

- The diagram in the slide displays the `orcl` and `+asm` instances as rectangles above a database cylinder, which lists the `DATA` and `FRA` disk groups.
- They are surrounded by a blue dashed line, denoting Enterprise Manager as a possible graphical interface.
- All this is surrounded again by a dashed black line, denoting the command line, as a possible character-based interface.

Usman Qamar (usman.qamar@live.com) has a non-transferable  
license to use this Student Guide.