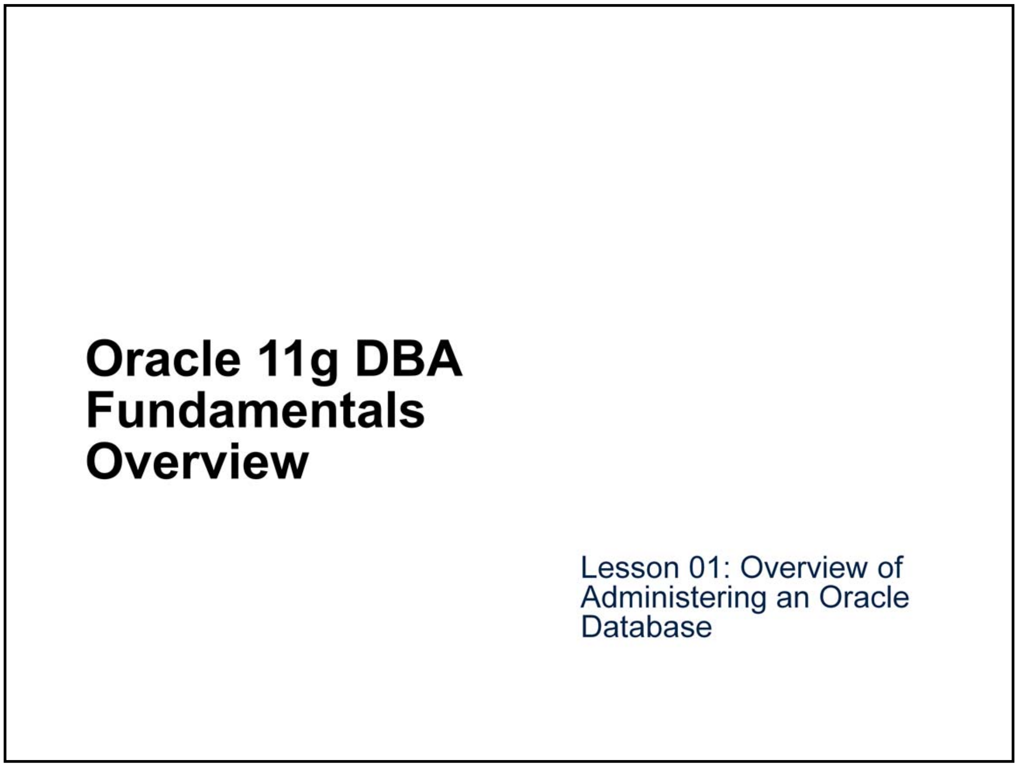
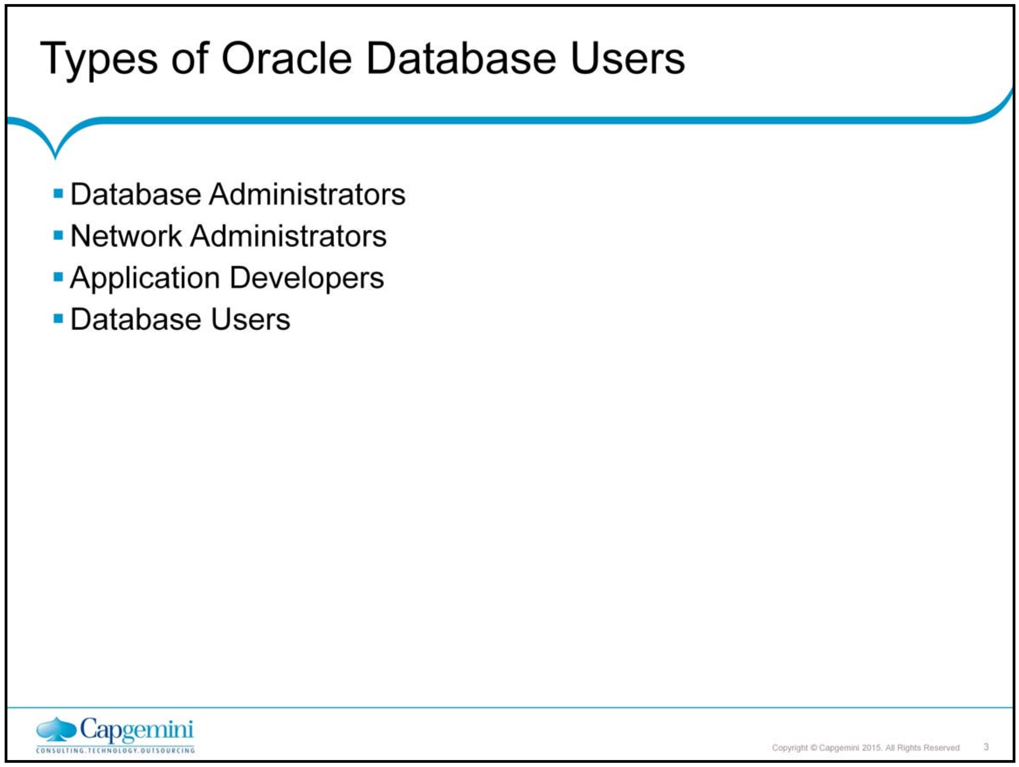

Oracle 11g DBA Fundamentals Overview
Lesson 01: Overview of Administering an Oracle Database

Lesson Objectives

- Types of Oracle Database Users
- Tasks of a Database Administrator
- DBA Security and Privileges
- Tools for Administering the Database
- Review the Oracle Database 11g architecture
- Managing Oracle Database Processes

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved 2


Types of Oracle Database Users
Database Administrators
Network Administrators
Application Developers
Database Users
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
3

Tasks of a Database Administrator

- Task 1: Evaluate the Database Server Hardware
- Task 2: Install the Oracle Database Software
- Task 3: Plan the Database
- Task 4: Create and Open the Database
- Task 5: Back Up the Database
- Task 6: Enroll System Users
- Task 7: Implement the Database Design
- Task 8: Back Up the Fully Functional Database
- Task 9: Tune Database Performance

Note:
- Keep titles concise – for example: Description, Features, Characteristics, Examples
- Ideas should be clean and simple
- No period at the end of bullet points

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved 4

Add the notes here.

DBA Security and Privileges

- Two user accounts are automatically created when Oracle Database is installed:
  - SYS (default password: CHANGE_ON_INSTALL)
  - SYSTEM (default password: MANAGER)

Copyright © Capgemini 2015. All Rights Reserved 5

SYS
When you create an Oracle Database, the user SYS is automatically created and granted the DBA role. All of the base tables and views for the database data dictionary are stored in the schema SYS. These base tables and views are critical for the operation of Oracle Database. To maintain the integrity of the data dictionary, tables in the SYS schema are manipulated only by the database. They should never be modified by any user or database administrator, and no one should create any tables in the schema of user SYS.
SYSTEM
When you create an Oracle Database, the user SYSTEM is also automatically created and granted the DBA role. The SYSTEM username is used to create additional tables and views that display administrative information, and internal tables and views used by various Oracle Database options and tools.

The DBA Role
A predefined DBA role is automatically created with every Oracle Database installation. This role contains most database system privileges. Therefore, the DBA role should be granted only to actual database administrators.

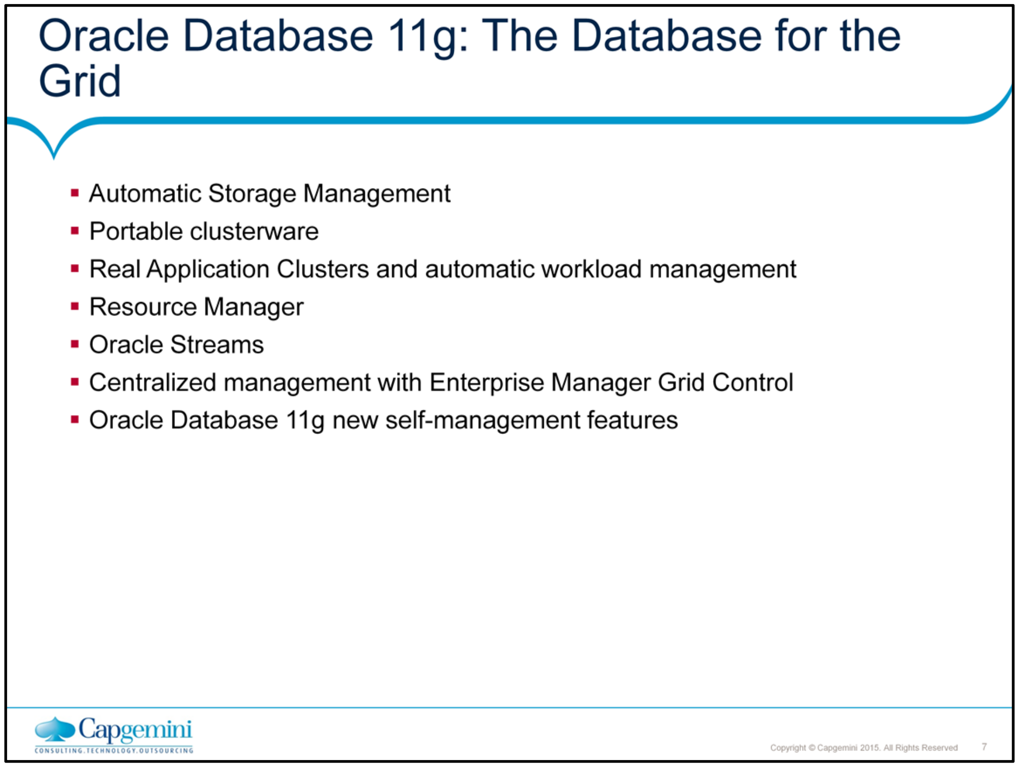
Tools for Administering the Database

- Oracle Universal Installer (OUI)
- Database Configuration Assistant (DBCA)
- Database Upgrade Assistant
- Oracle Net Manager
- Oracle Enterprise Manager

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved 6

Add the notes here.



Oracle Database 11g: The Database for the Grid

Oracle Database 11g is the first database that is designed for grid computing. To summarize, some of the most important features are the following:

Automatic Storage Management (ASM) virtualizes your storage and provides easy provisioning of your database storage.

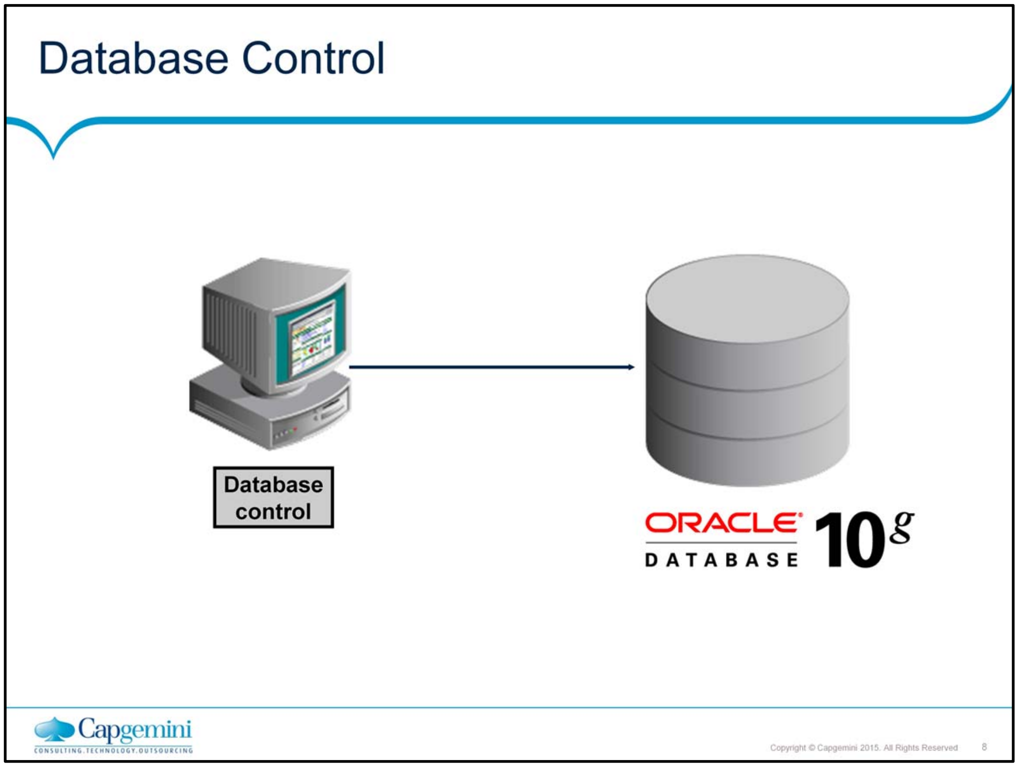
Oracle Database 11g offers portable clusterware that runs on all platforms.

Oracle Database 11g offers automatic workload management for services within a RAC database.

Oracle Database 11g provides additional mappings for consumer groups based on user host machine, application, OS username, or service.

Oracle Streams can stream data between databases, nodes, or blade farms in a grid. It provides a unified framework for information sharing, combining message queuing, replication, events, and data warehouse loading into a single technology.

Enterprise Manager Grid Control provides a single tool that can monitor and manage not only every Oracle software element (Oracle Application Server 11g and Oracle Database 11g) in your grid but also Web applications via Application Performance Management (APM), hosts, storage devices, and server load balancers.



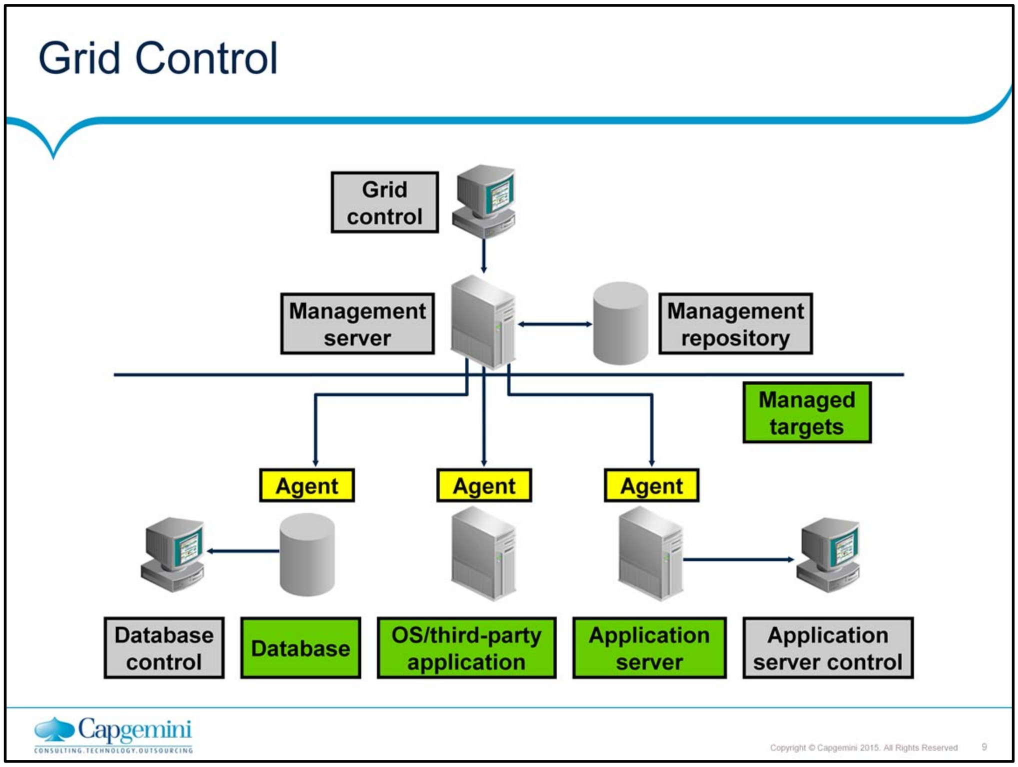
Database Control

Oracle Database 11g ships with Oracle Enterprise Manager’s Database Control. Database Control is a web-enabled control console that the database administrator can use for:

- Performance monitoring
- Managing proactive alerts
- Controlling maintenance wizards and advisors
- User and database object administration
- Database backup and recovery
- Storage management

and much more.

Each Oracle Database 11g you create will have its own Database Control. You will be using Enterprise Manager Database Control in this course to manage the database on your classroom PC.

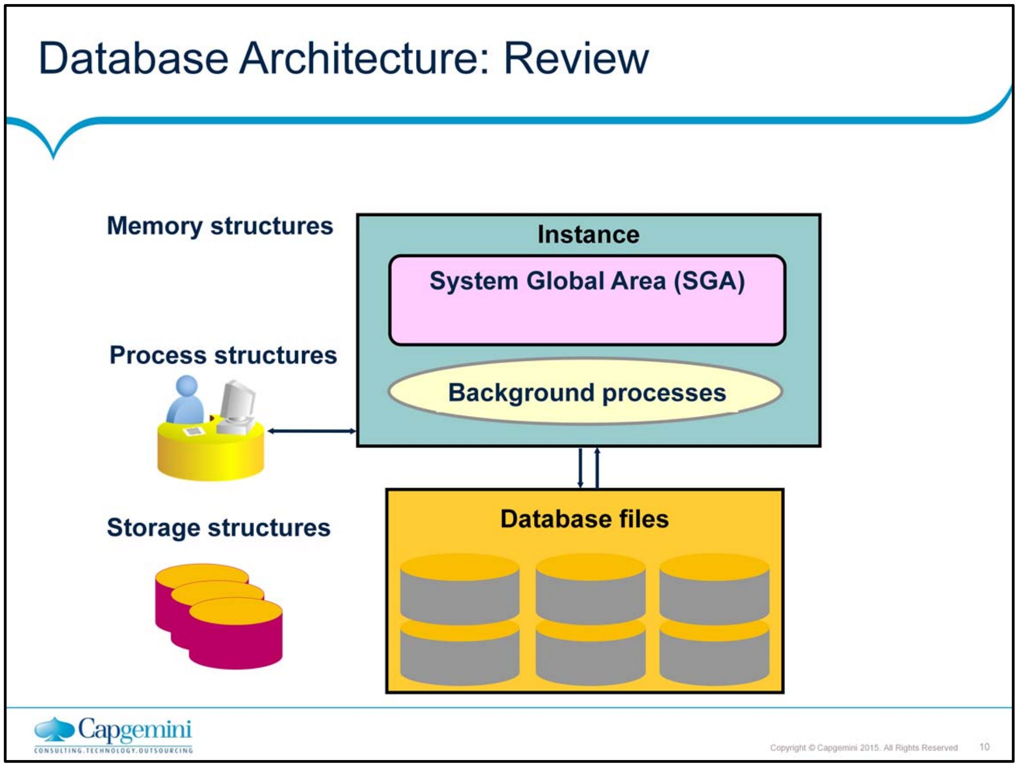


Grid Control

Database Control's capabilities can be extended and integrated with the rest of your systems using Oracle Enterprise Manager's Grid Control. The architecture of the Grid Control framework provides a high level of flexibility and functionality. You can easily customize Enterprise Manager to suit the monitoring and administrative needs of your environment.

The typical Enterprise Manager framework configuration consists of the following functional areas:

- Managed targets
- Management services
- Web-based grid control
- Database control
- Application server control



Database Architecture: Review

The following pages are a basic review of the Oracle database architecture. In this course, you enhance your knowledge of Oracle's database structures, processes, and utilities.

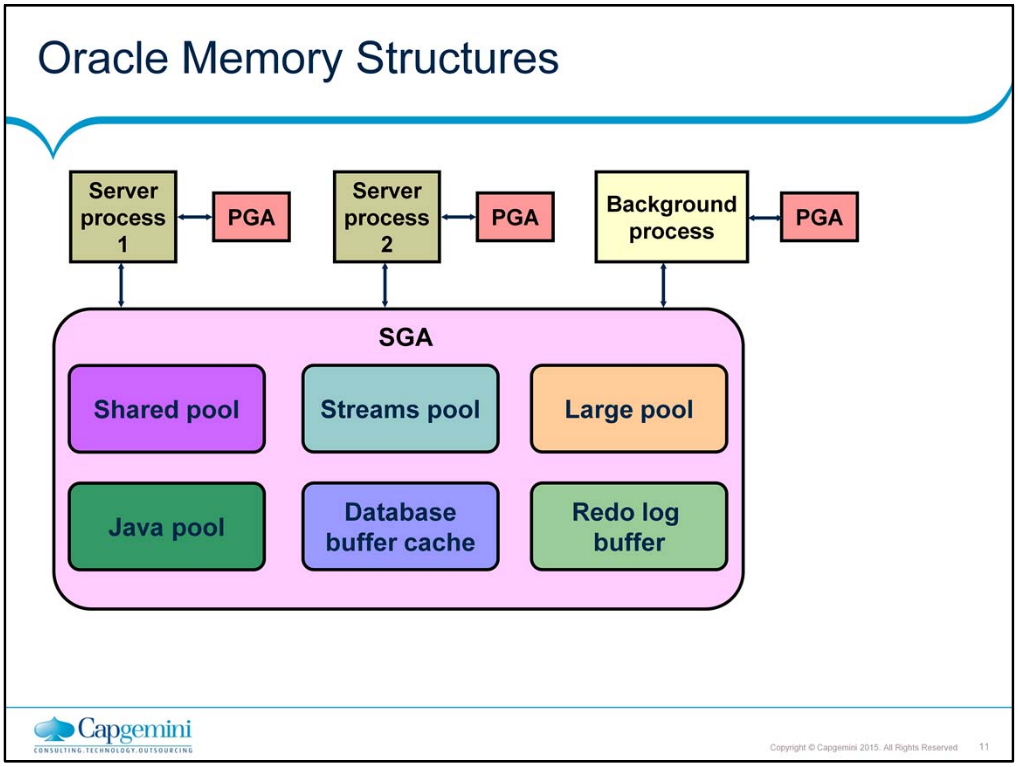
Each running Oracle database is associated with an Oracle instance. When a database is started on a database server, the Oracle software allocates a shared memory area called the System Global Area (SGA) and starts several Oracle background processes. This combination of the SGA and the Oracle processes is called an Oracle instance.

After starting an 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.

Multiple instances can execute concurrently on the same computer, each accessing its own physical database.

You can look at the Oracle database architecture as various interrelated structural components.

An Oracle database uses memory structures and processes to manage and access the database. All memory structures exist in the main memory of the computers that constitute the database server. Processes are jobs that work in the memory of these computers. A process is defined as a "thread of control" or a mechanism in an operating system that can run a series of steps.



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 SGA is a memory area that contains data and control information for the instance.
- The SGA includes 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 the disk
  - Shared pool: Caches various constructs that can be shared among users
  - Large pool: Is an 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: 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 amount of memory allocated for the SGA is displayed.

Oracle Memory Structures Full Notes Page

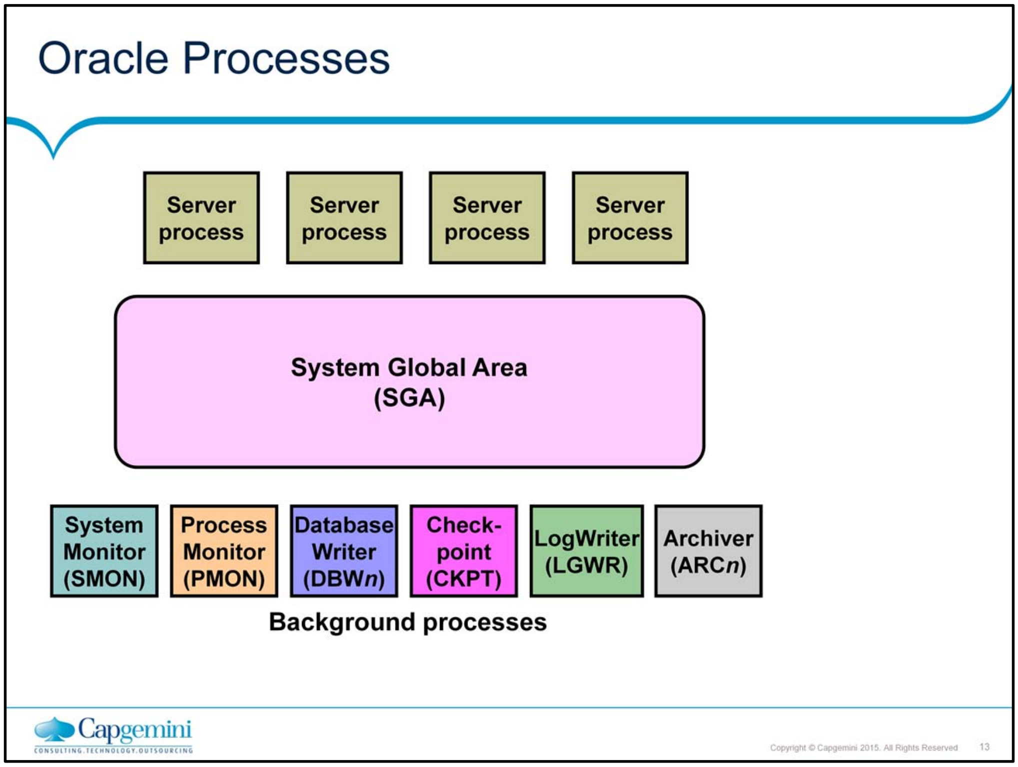
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved 12

Oracle Memory Structures (continued)

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 created 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.

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 changes without shutting down the instance.

The Oracle database uses initialization parameters to create and configure memory structures. For example, the SGA_TARGET parameter specifies the total amount of space available to the SGA. If you set SGA_TARGET to 0, Automatic Shared Memory Management is disabled.



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 the commands issued by the application. The Oracle server 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 perform other required tasks. Which background processes are present depends on 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

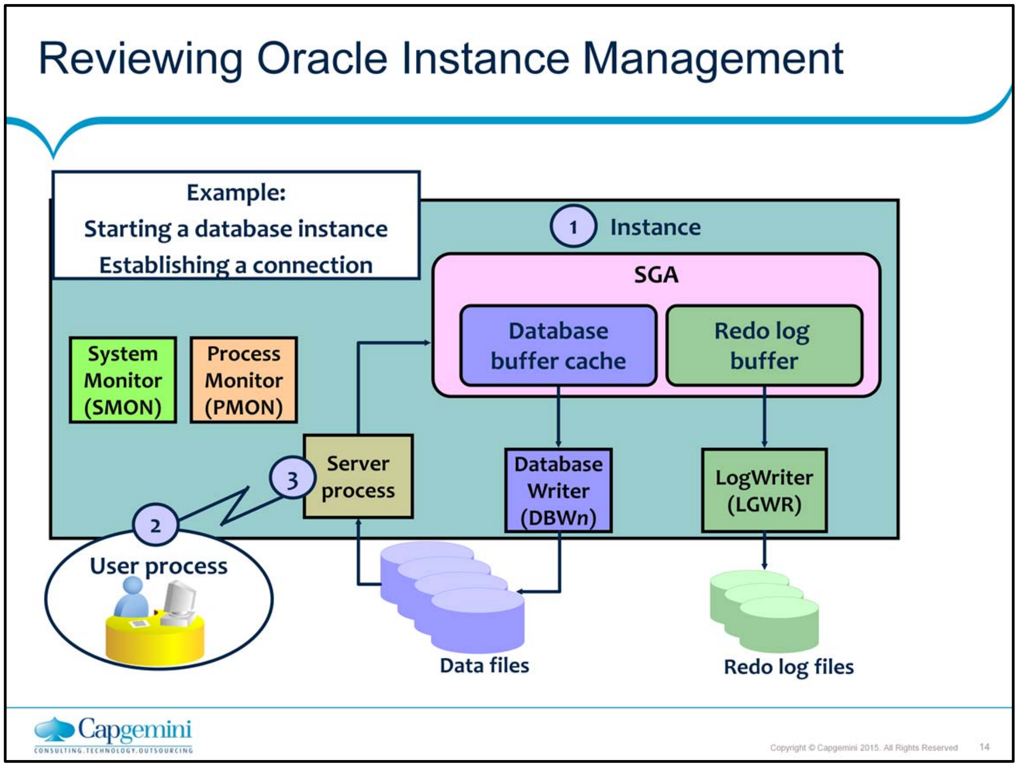
Process Monitor (PMON): Performs process cleanup when a user process fails

Database Writer (DBWn): Writes modified blocks from the database buffer cache to the data files on the disk

Checkpoint (CKPT): Updates all the data files and control files of the database to indicate the most recent checkpoint

LogWriter (LGWR): Writes redo log entries to the disk

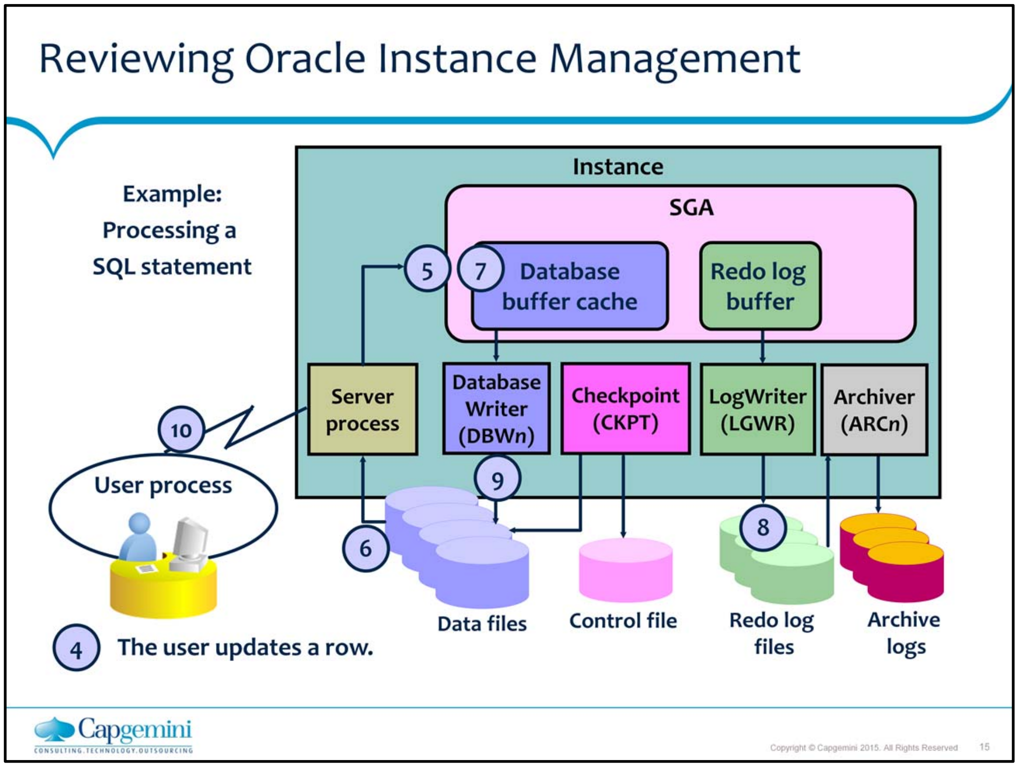
Archiver (ARCn): Copies redo log files to an archival storage when a log switch occurs



Reviewing Oracle Instance Management

The following example describes the most basic level of operations that the Oracle database performs. It illustrates an Oracle configuration where the user and associated server processes are on separate computers (connected through a network).

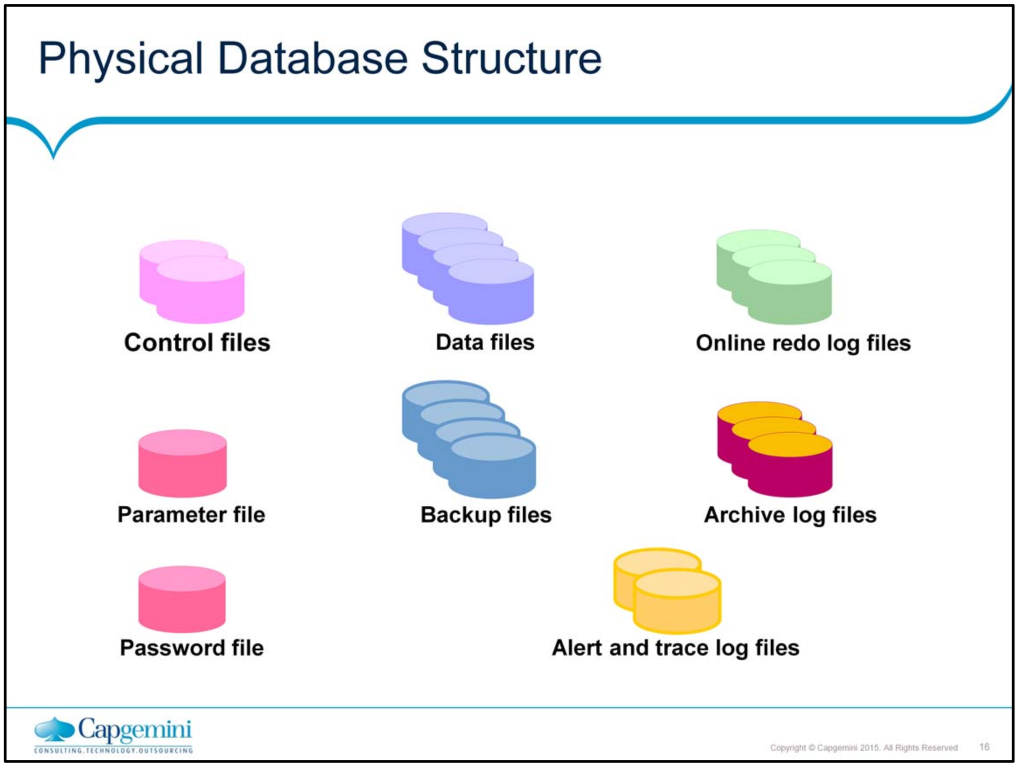
1. An instance has started on the computer running Oracle (often called the host or database server).
2. A computer running an application (a local computer or client workstation) runs the application in a user process. The client application attempts to establish a connection to the instance by using the Oracle Net Services driver.
3. The instance detects the connection request from the application and connects to a server process on behalf of the user process.



Reviewing Oracle Instance Management (continued)

4. The user updates a row.
5. The server process receives the statement and checks whether it is already in the shared pool of the SGA. If a shared SQL area is found, the server process checks the user's access privileges to the requested data, and the previously existing shared SQL area is used to process the statement. If the statement is not in the shared pool, then a new shared SQL area is allocated for the statement, so that it can be parsed and processed.
6. The server process retrieves any necessary data values from the actual data file (table) or from data blocks that are stored in the SGA.
7. The server process modifies the table data in the SGA.
8. When the transaction is committed, the LGWR process immediately records the transaction in the redo log file.
9. The DBWn process writes modified blocks to the disk when doing so is efficient.
10. The server process sends a success or error message across the network to the application.

Throughout this entire procedure, the other background processes run, watching for conditions that require intervention.



Physical Database Structure

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 within the database.

Data files: Contain the user or application data of the database

Online redo log files: Allow for instance recovery of the database. If the database crashes and does not lose any data files, then 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 users 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.

Archive 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.

Physical Database Structure Full Notes Page

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved 17

Physical Database Structure (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 files: Also known as alert logs, these are special trace files. The alert log of a database is a chronological log of messages and errors. Oracle recommends reviewing these files.

Oracle Managed Files (OMF)

- Specify file operations in terms of database objects rather than file names.

Parameter	Description
DB_CREATE_FILE_DEST	Defines the location of the default file system directory for data files and temporary files
DB_CREATE_ONLINE_LOG_DEST_n	Defines the location for redo log files and control file creation
DB_RECOVERY_FILE_DEST	Defines the location for RMAN backups

Exam:

```
SQL> ALTER SYSTEM SET DB_CREATE_FILE_DEST = '/u01/oradata';
SQL> CREATE TABLESPACE tbs_1;
```

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved 18

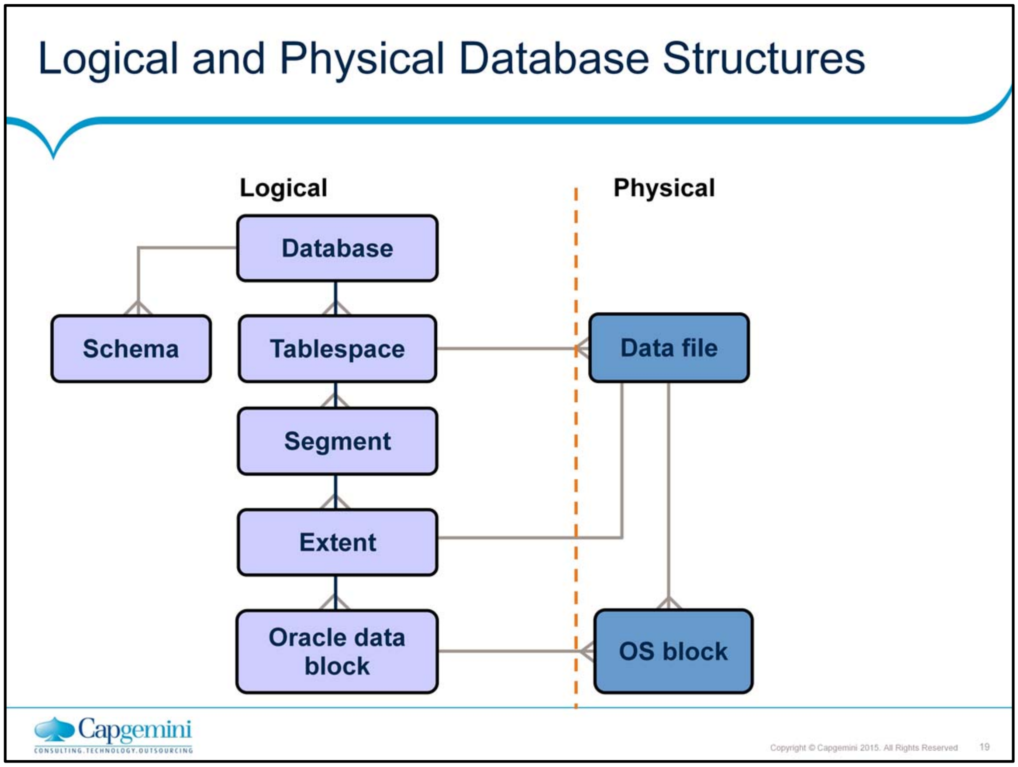
Oracle Managed Files (OMF)

Oracle Managed Files (OMF) eliminate the need for you to directly manage the operating system files that make up an Oracle database. You specify operations in terms of database objects rather than file names. The database internally uses standard file system interfaces to create and delete files as needed for the following database structures:

- Tablespaces
- Redo log files
- Control files
- Archived logs
- Block change tracking files
- Flashback logs
- RMAN backups

A database can have a mixture of Oracle-managed and unmanaged files. The file system directory specified by either of these parameters must already exist: the database does not create it. The directory must also have permissions to allow the database to create the files in it.

The example shows that after DB_CREATE_FILE_DEST is set, the DATAFILE clause can be omitted from a CREATE TABLESPACE statement. The data file is created in the location specified by DB_CREATE_FILE_DEST.



Logical and Physical Database Structures

An Oracle database is a collection of data that is treated as a unit. The general purpose of a database is to store and retrieve related information. 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 application data and an additional one for application indexes.

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 data file, the tablespace has a temporary file.

Logical and Physical Database Structures (continued)

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 such structures 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 called 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 allocated for a certain logical structure. For example, the different types of segments include:

***Data segments*:** Each nonclustered, non-index-organized table has a data segment. 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 by the database administrator 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.

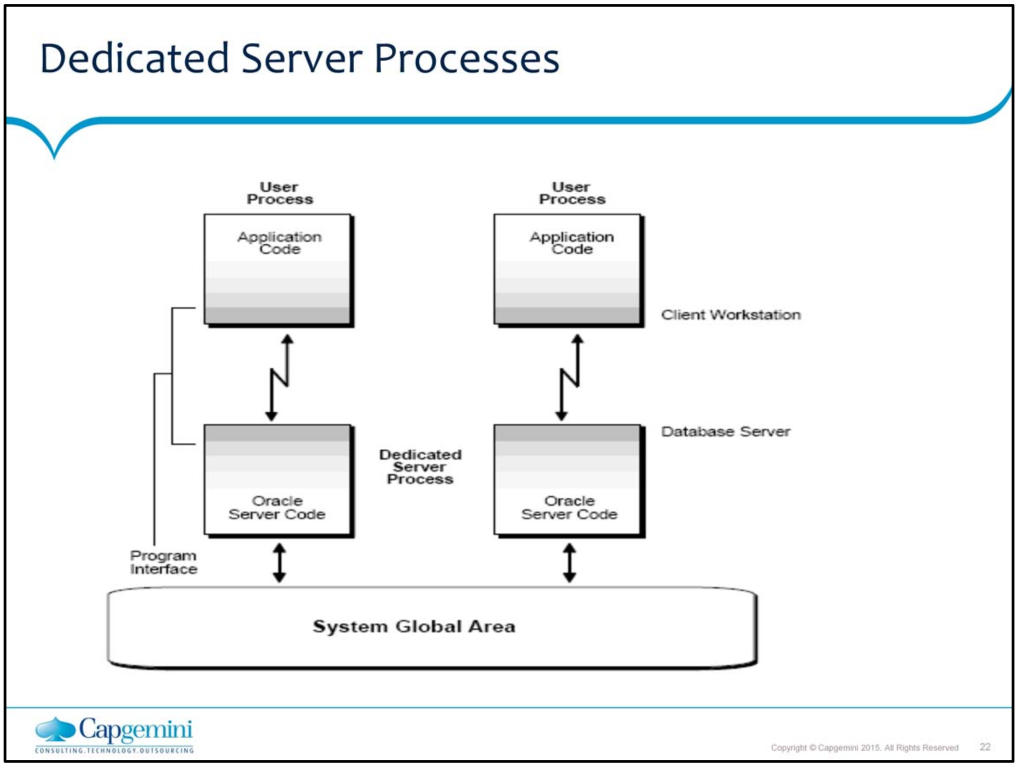
***Temporary segments*:** Temporary segments are created by the Oracle database when a SQL statement needs a temporary work area to complete execution. When the statement finishes execution, the temporary segment's extents are returned to the instance for future use. Specify a default temporary tablespace for every user or a default temporary tablespace, which is used databasewide.

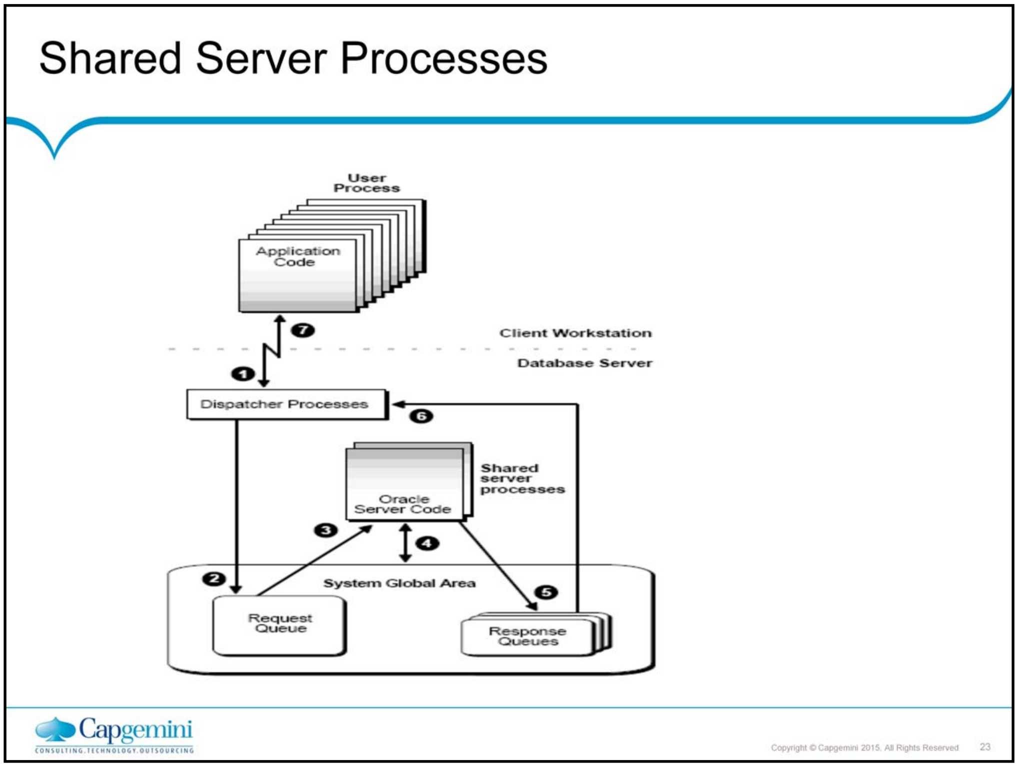
The Oracle database dynamically allocates space. When the existing extents of a segment are full, additional extents are added. Because extents are allocated as needed, the extents of a segment may or may not be contiguous on the disk.

About Dedicated and Shared Server Processes

- A server process can be either of the following:
  - A dedicated server process, which services only one user process
  - A shared server process, which can service multiple user processes

Copyright © Capgemini 2015. All Rights Reserved 21


Dedicated Server Processes
User Process
Application Code
User Process
Application Code
Client Workstation
Database Server
Dedicated Server Process
Oracle Server Code
Oracle Server Code
Program Interface
System Global Area
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
22



Add the notes here.

Configuring Oracle Database for Shared Server

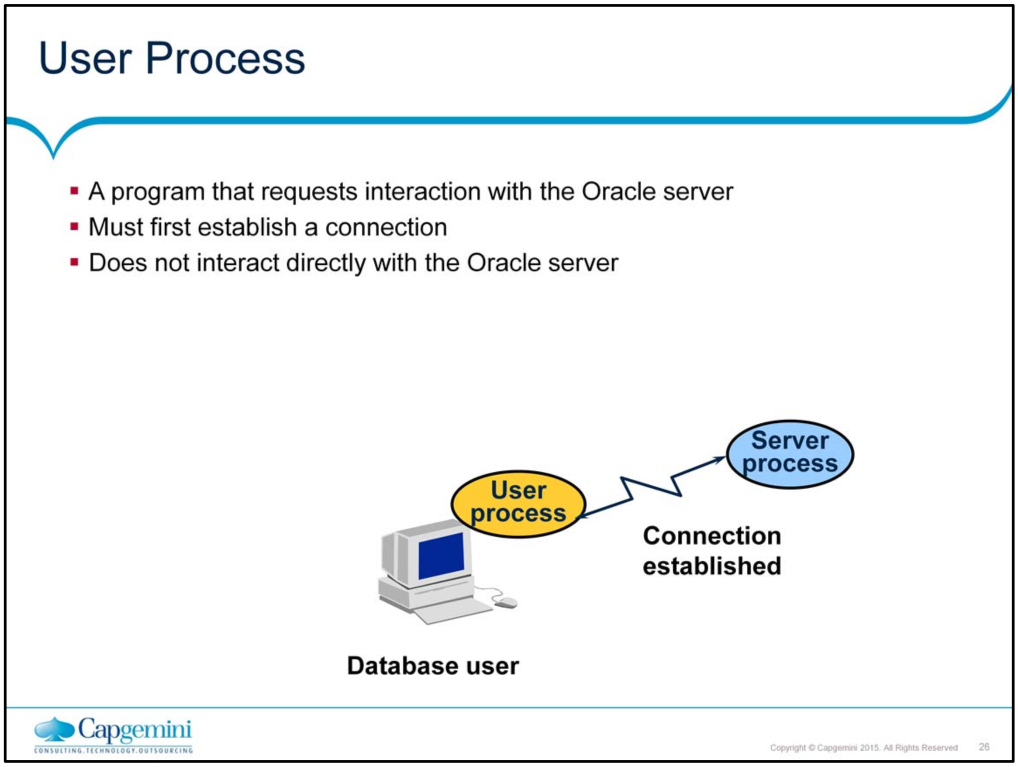
- Initialization Parameters for Shared Server
- Enabling Shared Server
- Configuring Dispatchers
- Monitoring Shared Server

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved 24

Process Structure

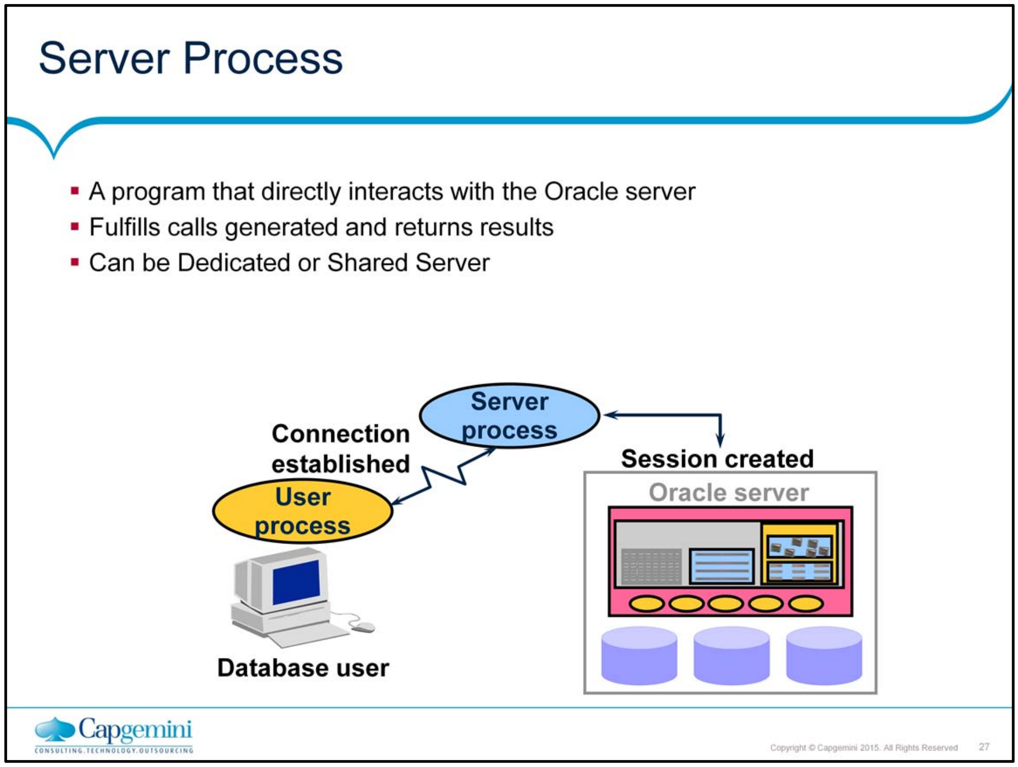
- Oracle takes advantage of various types of processes:
  - User process: Started at the time a database user requests connection to the Oracle server
  - Server process: Connects to the Oracle Instance and is started when a user establishes a session
  - Background processes: Started when an Oracle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

Copyright © Capgemini 2015. All Rights Reserved 25



User Process

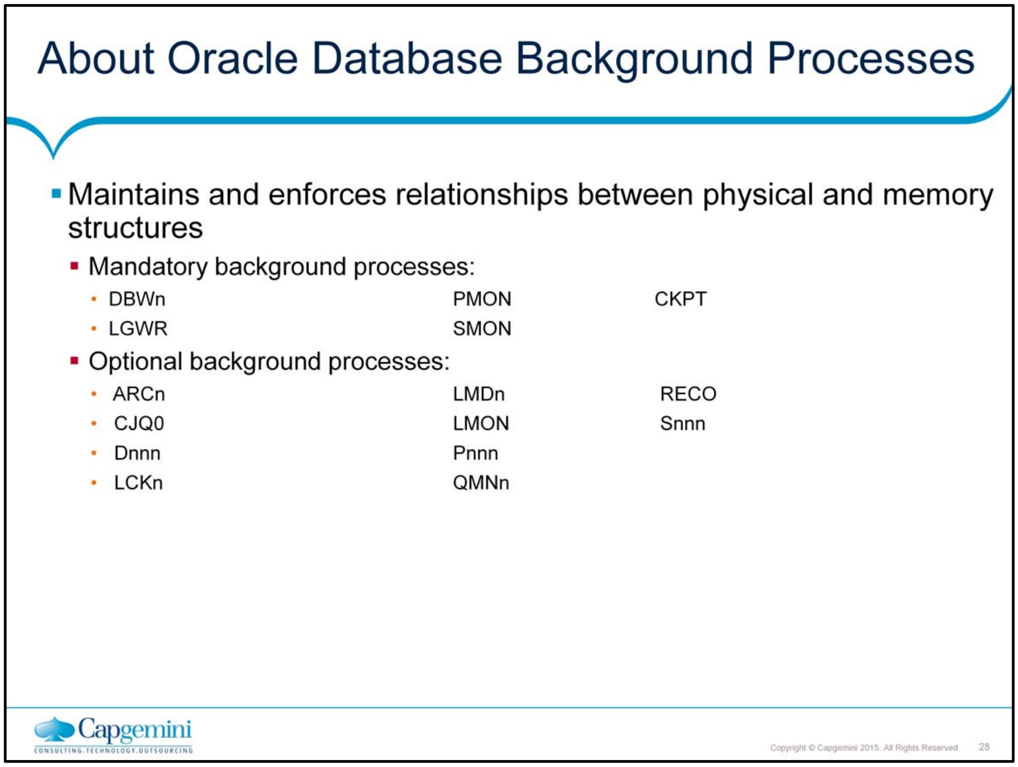
A database user who needs to request information from the database must first make a connection with the Oracle server. The connection is requested using a database interface tool, such as SQL*Plus, and beginning the user process. The user process does not interact directly with the Oracle server. Rather it generates calls through the user program interface (UPI), which creates a session and starts a server process.



Server Process

Once a user has established a connection, a server process is started to handle the user processes requests. A server process can be either a Dedicated Server process or a Shared Server process. In a Dedicated Server environment, the server process handles the request of a single user process. Once a user process disconnects, the server process is terminated. In a Shared Server environment, the server process handles the request of several user processes. The server process communicates with the Oracle server using the Oracle Program Interface (OPI).

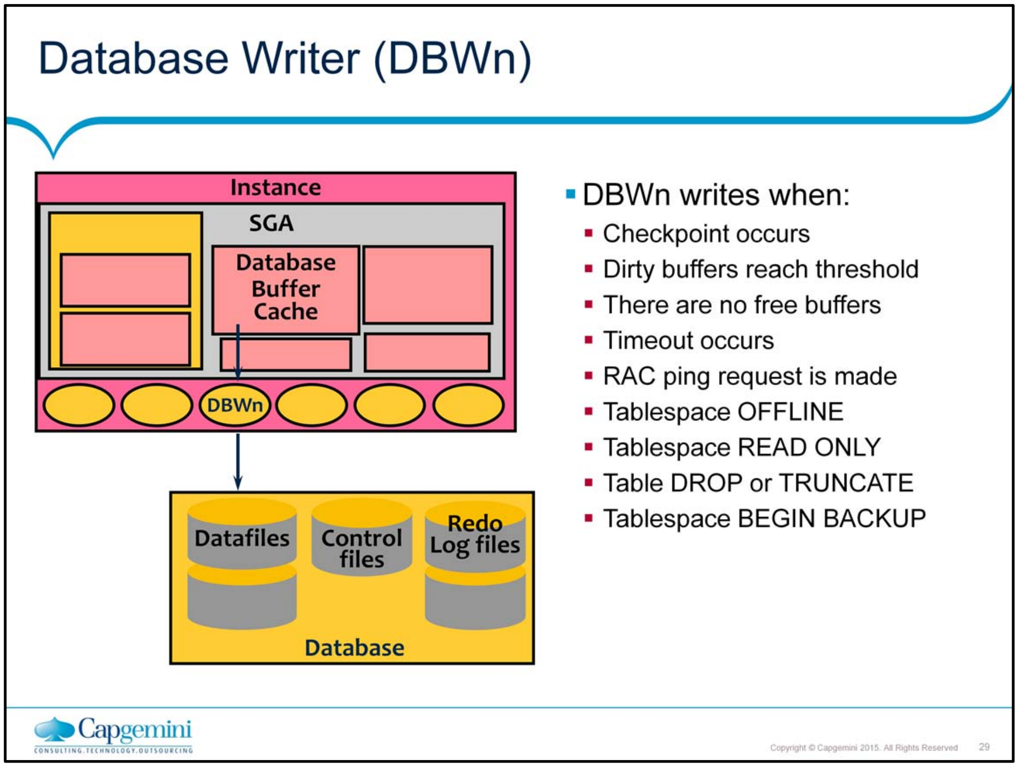
Note: Allocation of server process in a dedicated environment versus a shared environment is covered in further detail in the Oracle9i Database Performance Tuning course.



Background Processes

The Oracle architecture has five mandatory background processes that are discussed further in this lesson. In addition to the mandatory list, Oracle has many optional background process that are started when their option is being used. These optional processes are not within the scope of this course, with the exception of the background process, ARCn. Following is a list of some optional background processes:

RECO: Recoverer
QMNn: Advanced Queuing
ARCn: Archiver
LCKn: RAC Lock Manager–Instance Locks
LMON: RAC DLM Monitor–Global Locks
LMDn: RAC DLM Monitor–Remote Locks
CJQ0: Coordinator Job Queue background process
Dnnn: Dispatcher
Snnn: Shared Server
Pnnn: Parallel Query Slaves

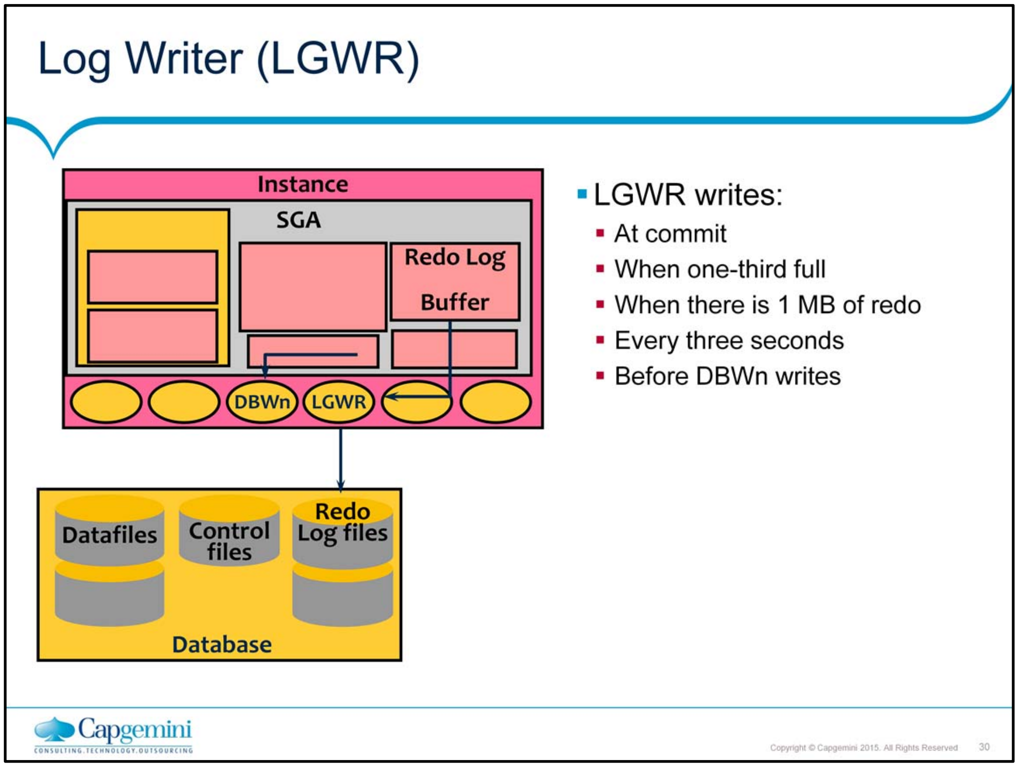


Database Writer (DBWn)

The server process records changes to undo and data blocks in the Database Buffer Cache. DBWn writes the dirty buffers from the Database Buffer Cache to the datafiles. It ensures that a sufficient number of free buffers (buffers that can be overwritten when server processes need to read in blocks from the datafiles) are available in the Database Buffer Cache. Database performance is improved because server processes make changes only in the Database Buffer Cache.

DBWn defers writing to the datafiles until one of the following events occurs:

- Incremental or normal checkpoint
- The number of dirty buffers reaches a threshold value
- A process scans a specified number of blocks when scanning for free buffers and cannot find any
- Timeout occurs
- A ping request in Real Application Clusters (RAC) environment
- Placing a normal or temporary tablespace offline
- Placing a tablespace in read-only mode
- Dropping or truncating a table
- ALTER TABLESPACE tablespace name BEGIN BACKUP



Log Writer (LGWR)

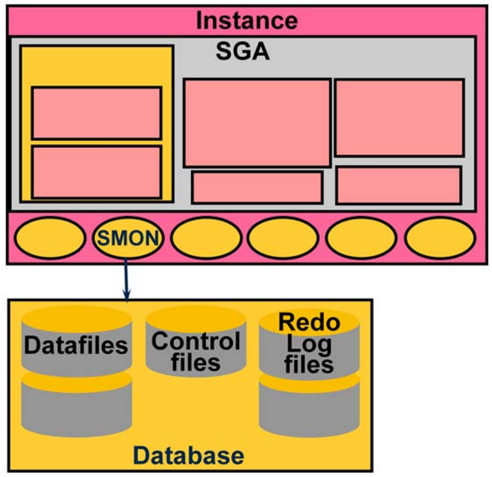
LGWR performs sequential writes from the Redo Log Buffer to the redo log file under the following situations:

- When a transaction commits
- When the Redo Log Buffer is one-third full
- When there is more than 1 MB of changes recorded in the Redo Log Buffer
- Before DBWn writes modified blocks in the Database Buffer Cache to the datafiles
- Every three seconds

Because the redo is needed for recovery, LGWR confirms the commit operation only after the redo is written to disk.

LGWR can also call on DBWn to write to the datafiles.

System Monitor (SMON)



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

Capgemini CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved 31

System Monitor (SMON)

If the Oracle Instance fails, any information in the SGA that has not been written to disk is lost. For example, the failure of the operating system causes an instance failure. After the loss of the instance, the background process SMON automatically performs instance recovery when the database is reopened. Instance recovery consists of the following steps:

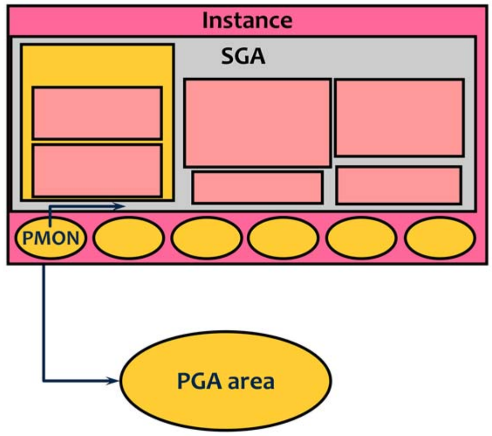
1. Rolling forward to recover data that has not been recorded in the datafiles but that has been recorded in the online redo log. This data has not been written to disk because of the loss of the SGA during instance failure. During this process, SMON reads the redo log files and applies the changes recorded in the redo log to the data blocks. Because all committed transactions have been written to the redo logs, this process completely recovers these transactions.
2. Opening the database so that users can log on. Any data that is not locked by unrecovered transactions is immediately available.
3. Rolling back uncommitted transactions. They are rolled back by SMON or by the individual server processes as they access locked data.

SMON also performs some space maintenance functions:

It combines, or coalesces, adjacent areas of free space in the datafiles.

It deallocates temporary segments to return them as free space in datafiles.

Process Monitor (PMON)



- Cleans up after
- failed processes by:
  - Rolling back the transaction
  - Releasing locks
  - Releasing other resources
  - Restarting dead dispatchers

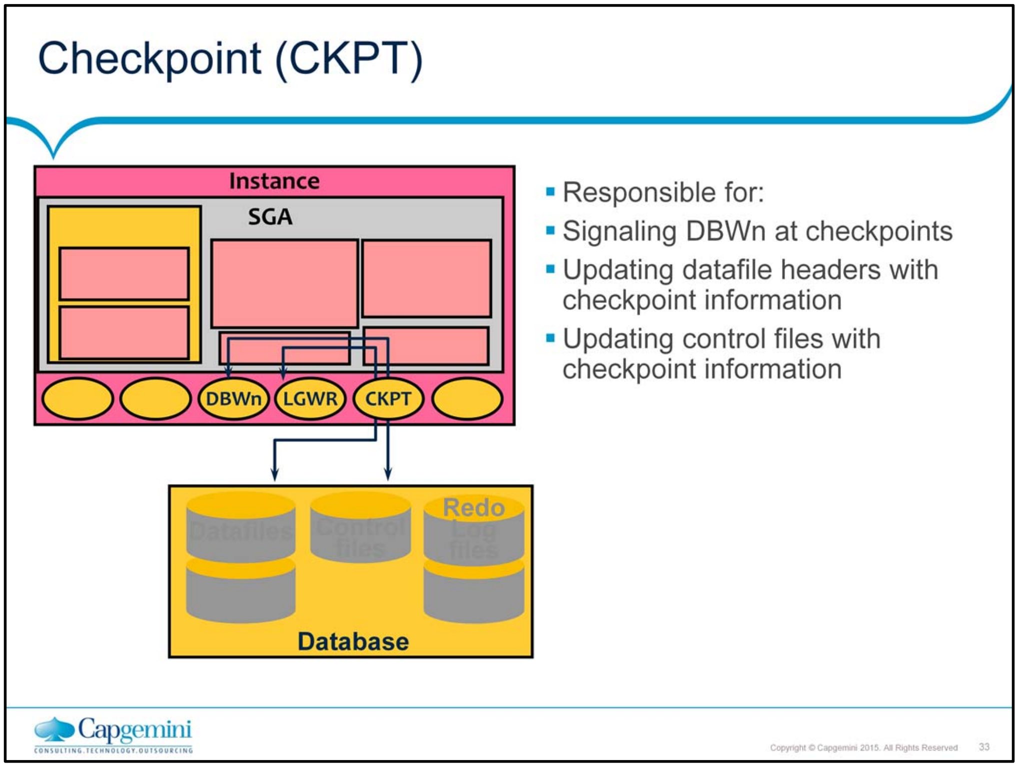
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved 32

Process Monitor (PMON)

The background process PMON cleans up after failed processes by:

- Rolling back the user’s current transaction
- Releasing all currently held table or row locks
- Freeing other resources currently reserved by the user
- Restarts dead dispatchers

Dispatchers are covered in further detail in the Oracle9i Database Administration Fundamentals II course.



Checkpoint (CKPT)

Every three seconds the CKPT process stores data in the control file to identify that place in the redo log file where recovery is to begin, this being called a checkpoint. The purpose of a checkpoint is to ensure that all of the buffers in the Database Buffer Cache that were modified prior to a point in time have been written to the datafiles. This point in time (called the checkpoint position) is where database recovery is to begin in the event of an instance failure. DBWn will already have written all of the buffers in the Database Buffer Cache that were modified prior to that point in time. Prior to Oracle9i, this was done at the end of the redo log. In the event of a log switch CKPT also writes this checkpoint information to the headers of the datafiles.

Checkpoints are initiated for the following reasons:

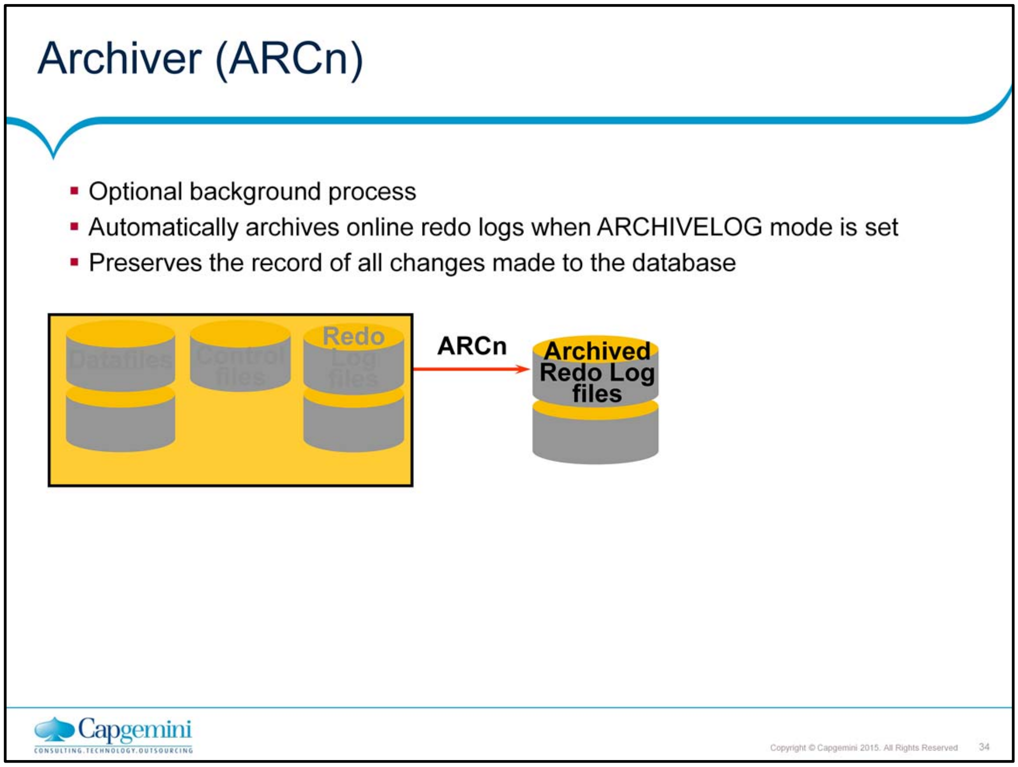
To ensure that modified data blocks in memory are written to disk regularly so that data is not lost in case of a system or database failure

To reduce the time required for instance recovery. Only the redo log entries following the last checkpoint need to be processed for recovery to occur

To ensure that all committed data has been written to the datafiles during shutdown

Checkpoint information written by CKPT includes checkpoint position, system change number, location in the redo log to begin recovery, information about logs, and so on.

Note: CKPT does not write data blocks to disk or redo blocks to the online redo logs.



Archiver (ARCn)

All other background processes are optional, depending on the configuration of the database; however, one of them, ARCn, is crucial to recovering a database after the loss of a disk. As online redo log files get filled, the Oracle server begins writing to the next online redo log file. The process of switching from one redo log to another is called a log switch. The ARCn process initiates backing up, or archiving, of the filled log group at every log switch. It automatically archives the online redo log before the log can be reused, so that all of the changes made to the database are preserved. This enables the DBA to recover the database to the point of failure even if a disk drive is damaged.

Archiving redo log files:

One of the important decisions that a DBA has to make is whether to configure the database to operate in ARCHIVELOG or in NOARCHIVELOG mode.

NOARCHIVELOG mode: In NOARCHIVELOG mode, the online redo log files are overwritten each time a log switch occurs. LGWR does not overwrite a redo log group until the checkpoint for that group is complete. This ensures that committed data can be recovered if there is an instance crash. During the instance crash, only the SGA is lost. There is no loss of disks, only memory. For example, an operating system crash causes an instance crash.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved 35

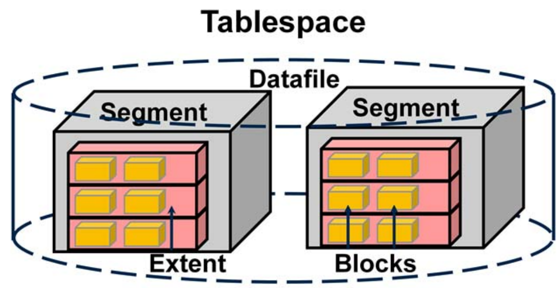
Archiving Redo Log Files (continued)

ARCHIVELOG mode: If the database is configured to run in ARCHIVELOG mode, inactive groups of filled online redo log files must be archived before they can be used again. Since changes made to the database are recorded in the online redo log files, the database administrator can use the physical backup of the datafiles and the archived online redo log files to recover the database without losing any committed data because of any single point of failure, including the loss of a disk. Usually, a production database is configured to run in ARCHIVELOG mode.

Archive log modes are covered in further detail in the Oracle9i Database Administration Fundamentals II course.

Logical Structure

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



Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved 36

Logical Structure

A logical structure hierarchy exists as follows:

- An Oracle database contains at least one tablespace.
- A tablespace contains one or more segments.
- A segment is made up of extents.
- An extent is made up of logical blocks.
- A block is the smallest unit for read and write operations.

The Oracle database architecture includes logical and physical structures that make up the database.

- The physical structure includes the control files, online redo log files, and datafiles that make up the database.
- The logical structure includes tablespaces, segments, extents, and data blocks.

The Oracle server enables fine-grained control of disk space use through tablespace and logical storage structures, including segments, extents, and data blocks.

Logical Structure (continued)

Tablespaces:

The data in an Oracle database are stored in tablespaces.

- An Oracle database can be logically grouped into smaller logical areas of space known as tablespaces.
- A tablespace can belong to only one database at a time.
- Each tablespace consists of one or more operating system files, which are called datafiles.
- A tablespace may contains one or more segments.
- Tablespaces can be brought online while the database is running.
- Except for the SYSTEM tablespace or a tablespace with an active undo segment, tablespaces can be taken offline, leaving the database running.
- Tablespaces can be switched between read-write and read-only status.

Datafiles (Not a logical structure):

- Each tablespace in an Oracle database consists of one or more files called datafiles. These are physical structures that conform with the operating system on which the Oracle server is running.
- A data file can belong to only one tablespace.
- An Oracle server creates a data file for a tablespace by allocating the specified amount of disk space plus a small amount of overhead.
- The database administrator can change the size of a data file after its creation or can specify that a data file should dynamically grow as objects in the tablespace grow.

Segments:

- A segment is the space allocated for a specific logical storage structure within a tablespace.
- A tablespace may consist of one or more segments.
- A segment cannot span tablespaces; however, a segment can span multiple datafiles that belong to the same tablespace.
- Each segment is made up of one or more extents.

Extents:

Space is allocated to a segment by extents.

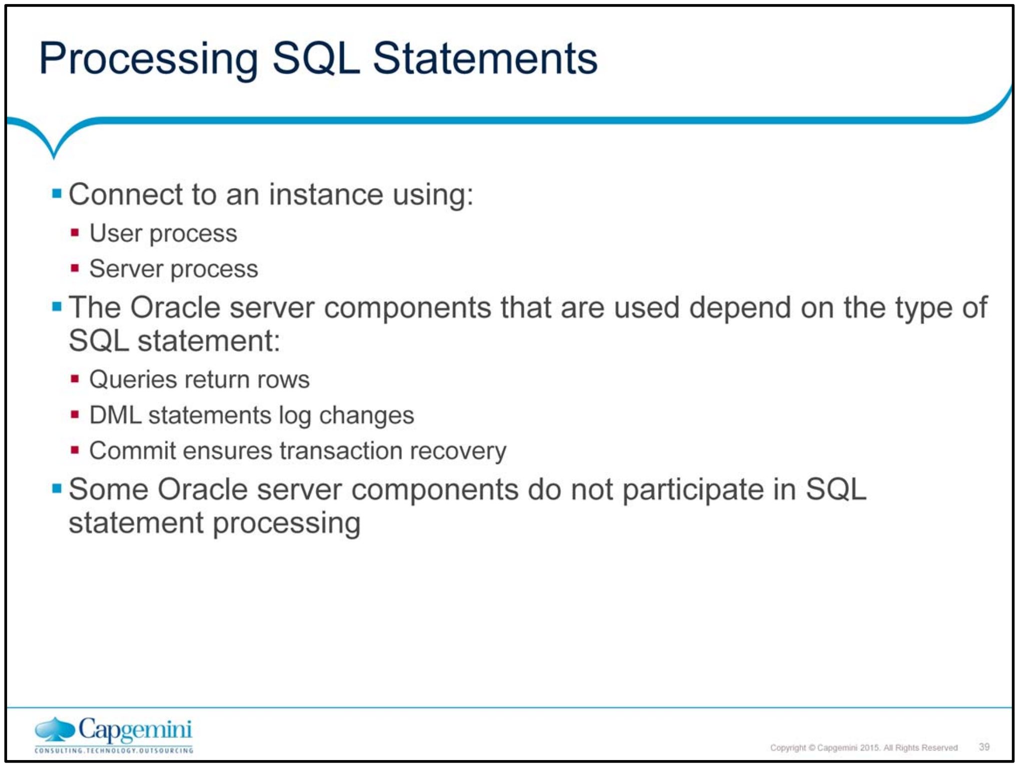
- One or more extents make up a segment.
  - When a segment is created, it consists of at least one extent.
  - As the segment grows, extents get added to the segment.
  - The DBA can manually add extents to a segment.
- An extent is a set of contiguous Oracle blocks.
- An extent cannot span datafiles, and therefore, it must exist in one datafile.

Logical Structure (continued)

Data blocks:

The Oracle server manages the storage space in the datafiles in units called Oracle blocks or data blocks.

- At the finest level of granularity, the data in an Oracle database is stored in data blocks.
- Oracle data blocks are the smallest units of storage that the Oracle server can allocate, read, or write.
- One data block corresponds to one or more operating system blocks allocated from an existing data file.
- The standard data block size for an Oracle database is specified by the DB_BLOCK_SIZE initialization parameter when the database is created.
- The data block size should be a multiple of the operating system block size to avoid unnecessary I/O.
- The maximum data block size is dependent on the operating system.



Processing SQL Statements

- Processing a query:
  - Parse:
    - Search for identical statement
    - Check syntax, object names, and privileges
    - Lock objects used during parse
    - Create and store execution plan
  - Bind: Obtains values for variables
  - Execute: Process statement
  - Fetch: Return rows to user process

Processing SQL Statements (continued)

Processing a DML statement:

Parse: Same as the parse phase used for processing a query.

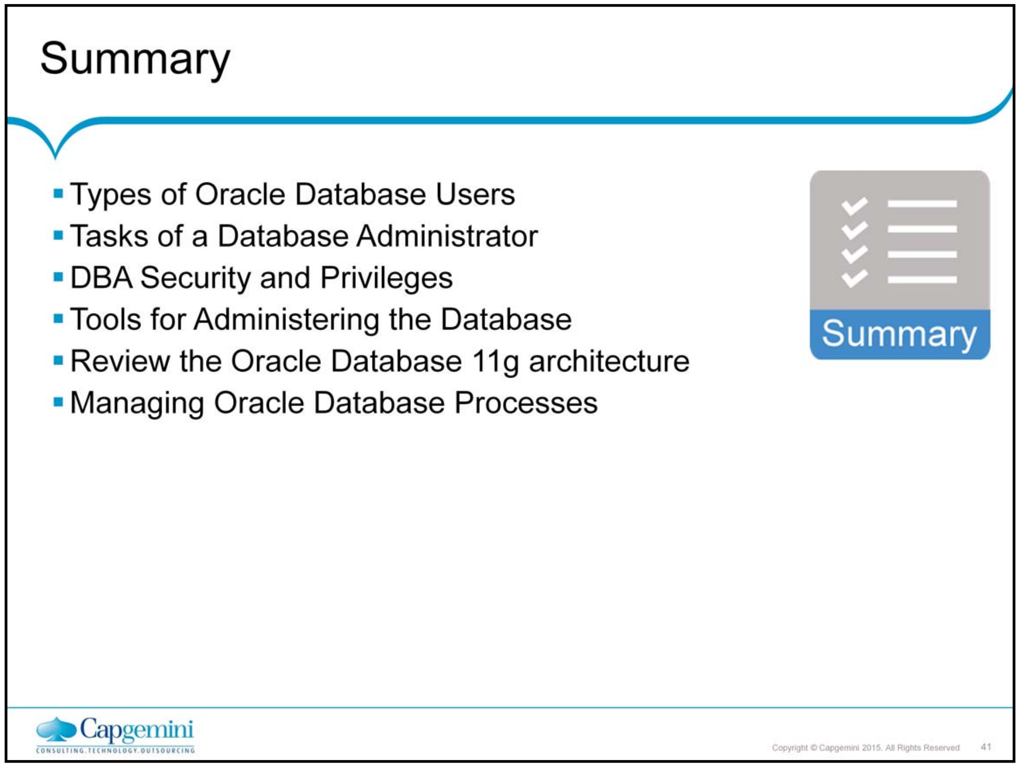
Bind: Same as the bind phase used for processing a query.

Execute:

- If the data and undo blocks are not already in the Database Buffer Cache, the server process reads them from the datafiles into the Database Buffer Cache.
- The server process places locks on the rows that are to be modified. The undo block is used to store the before image of the data, so that the DML statements can be rolled back if necessary.
- The data blocks record the new values of the data.
- The server process records the before image to the undo block and updates the data block. Both of these changes are made in the Database Buffer Cache. Any changed blocks in the Database 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.

Processing a DDL statement:

The execution of DDL (Data Definition Language) statements differs from the execution of DML (Data Manipulation Language) statements and queries, because the success of a DDL statement requires write access to the data dictionary. For these statements, parsing actually includes parsing, data dictionary lookup, and execution. Transaction management, session management, and system management SQL statements are processed using the parse and execute stages. To re-execute them, simply perform another execute.



Add the notes here.