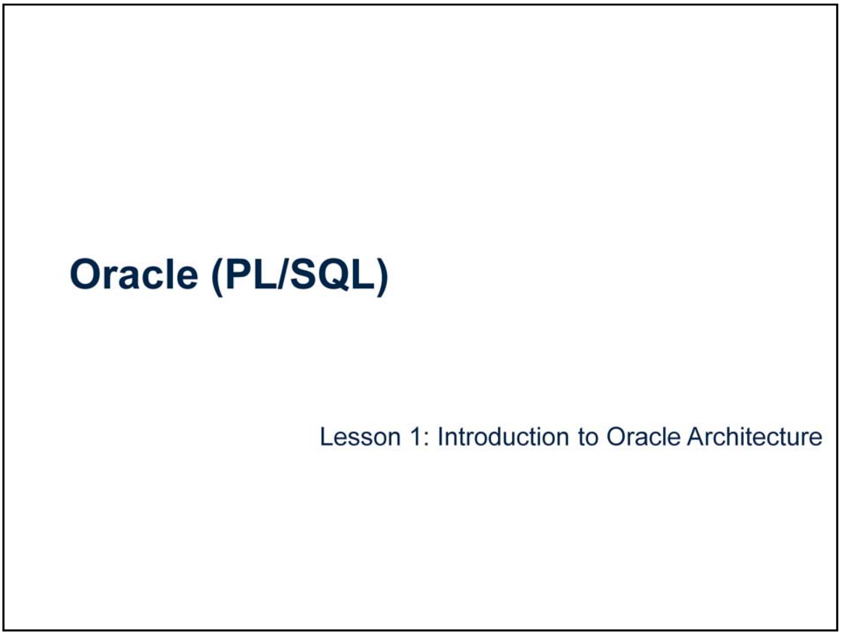
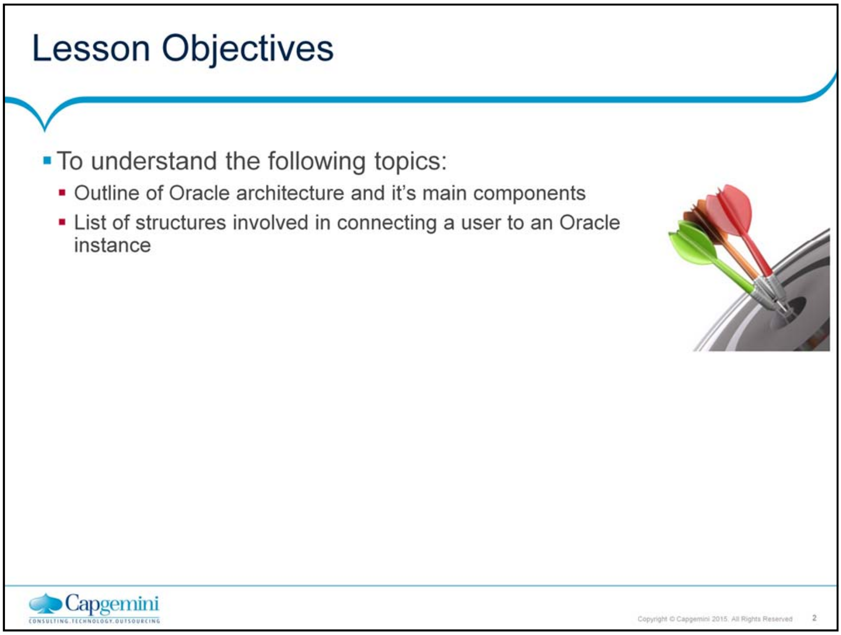


Oracle (PL/SQL)

Lesson 1: Introduction to Oracle Architecture

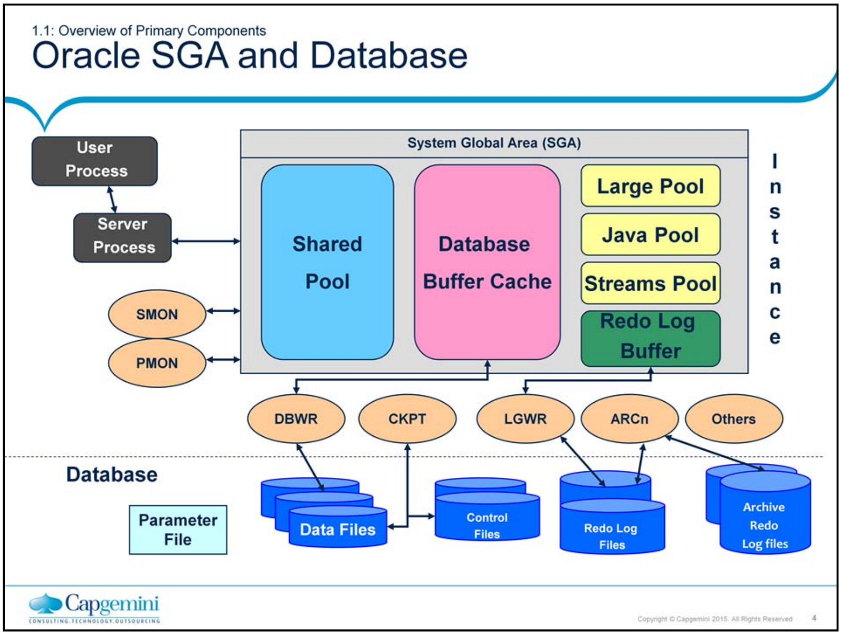


Lesson Objectives

- To understand the following topics:
  - Outline of Oracle architecture and it’s main components
  - List of structures involved in connecting a user to an Oracle instance

<u>Introduction to Oracle Database</u>:

- Oracle is an Object Relational Database Management System (ORDBMS).
- Oracle uses:
    - Relational Data Model as well as Object Relational Data Model to store its database, and
    - SQL (commonly abbreviated as Structured Query Language) to process the stored data.
- Oracle database is designed with improvements in the areas such as:
    - database performance
    - ease of management
    - scalability
    - security
    - availability, etc.
- The following features makes Oracle very powerful:
    - usage of XMLType, which is a new data type that lets you store native XML documents directly in the database
    - support of multimedia and large objects
    - support for Oracle Streams, which are a generic mechanism for sharing data that can be used as the basis of many processes including messaging, replication, and warehouse ETL processes



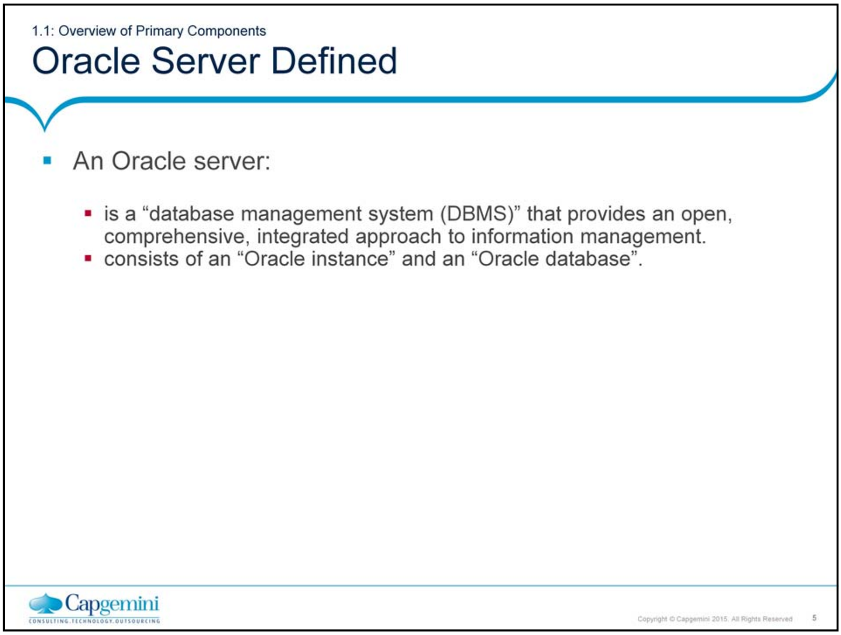
1.1: Overview of Primary Components

Oracle SGA and Database

Overview of Primary Components in Oracle Architecture:

The Oracle Server consists of an “Oracle instance” and an “Oracle database”.

1. **Oracle instance**: An “Oracle instance” is the combination of the “background processes” and “memory structures”.
   - The instance must be started to access the data in the database. Every time an instance is started, a “System Global Area (SGA)” is allocated, and “Oracle background processes” are started.
   - Background processes perform functions on behalf of the invoking process. The background processes perform input / output (I/O), and monitor other Oracle processes to provide increased parallelism for “better performance” and “reliability”.
2. **Oracle database**: An Oracle database consists of “Operating System files”, also known as “database files” that provide the actual physical storage for database information. The database files are used to ensure that the data is kept consistent and can be recovered in the event of a failure of the instance.
3. **Other key files**: Non-database files are used to configure the instance, authenticate privileged users, and recover the database in the event of a disk failure. Eg: Parameter File, Archive files
4. **User and server processes**: The “user processes” and “server processes” are the primary processes involved when a SQL statement is executed. However, other processes may help the server complete the processing of the SQL statement.



1.1: Overview of Primary Components

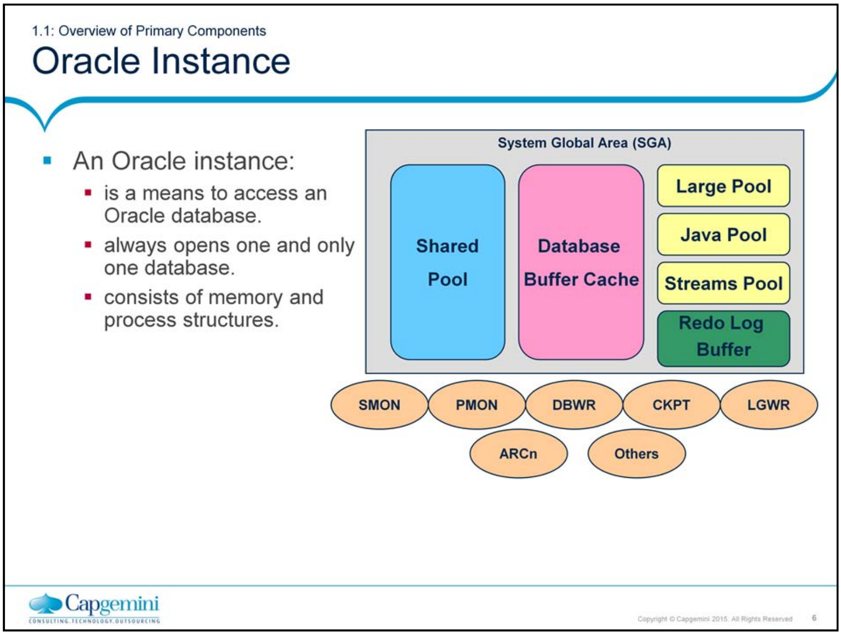
Oracle Server Defined

- An Oracle server:
  - is a “database management system (DBMS)” that provides an open, comprehensive, integrated approach to information management.
  - consists of an “Oracle instance” and an “Oracle database”.

5

<u>Oracle Server</u>:

- The Oracle Server can run on a number of different computers in one of the following environments:
  - ➢ Client-Application Server-Server
  - ➢ Client-Server
  - ➢ Host-Based
- **Client-Application Server-Server:** (Three-tier) Users access the database from their personal computers (clients) through an application server, which is used for the application’s processing requirements.
- **Client-Server:** (Two-tier) Users access the database from their personal computer (client) over a network, and the database sits on a separate computer (server).
- **Host-Based:** Users are connected directly to the same computer which houses the database.



1.1: Overview of Primary Components

Oracle Instance

- An Oracle instance:
  - is a means to access an Oracle database.
  - always opens one and only one database.
  - consists of memory and process structures.

System Global Area (SGA)

Shared Pool

Database Buffer Cache

Large Pool

Java Pool

Streams Pool

Redo Log Buffer

SMON

PMON

DBWR

CKPT

LGWR

ARCn

Others

Capgemini

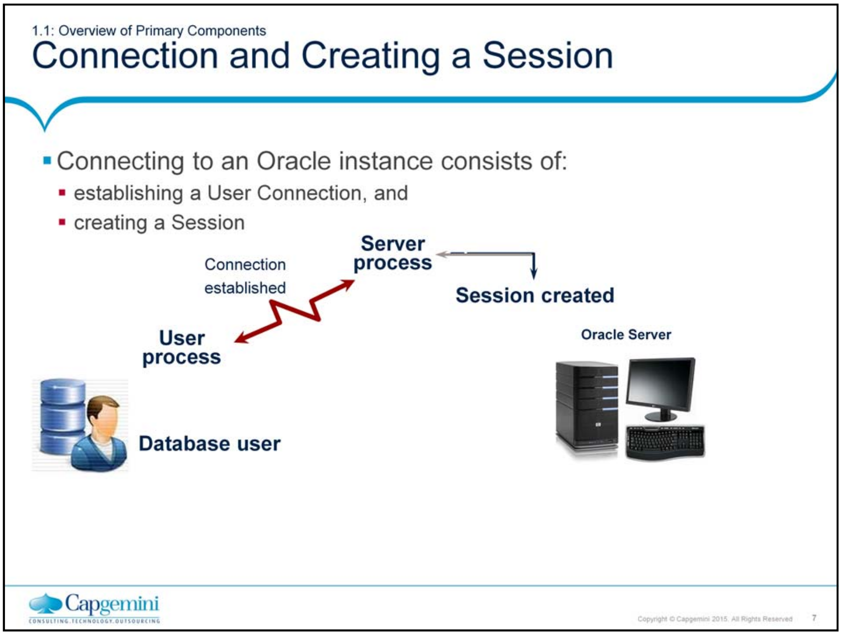
CONSULTING.TECHNOLOGY.OUTSOURCING

6

<u>Oracle Instance</u>:

- When a database is started on a database server (regardless of the type of computer), the Oracle instance starts up. An Oracle instance consists of two components, one is memory structures and two is background processes. The memory structures allocated are termed as the System Global Area (SGA). Along with the allocation of memory the background processes also start.
- This combination of the SGA and the Oracle processes is called an “Oracle instance”.
- The memory and processes of an Oracle instance:
  - ➢ efficiently manage the data of the associated database, and
  - ➢ serve one or multiple users of the database
- Oracle instance is a means to access an Oracle database.
- Oracle instance always opens “one and only one database”.

We will understand the SGA first in the subsequent slides and then move on to the database



1.1: Overview of Primary Components

Connection and Creating a Session

- Connecting to an Oracle instance consists of:
  - establishing a User Connection, and
  - creating a Session

Server process

Connection established

Session created

User process

Oracle Server

Database user

Copyright © Capgemini 2015. All Rights Reserved 7

Connecting to an Oracle Instance:

- Before users can submit SQL statements to an Oracle database, they must connect to an instance.
  - **Establishing a User Connection**
    - The user starts a tool such as SQL*Plus or runs an application developed by using a tool such as Oracle Forms. This application or tool is executed as a “user process”.
    - Establishing the connection creates a communication pathway between a user process and an Oracle Server. As depicted in the figure on the slide, the user process communicates with a server process. The user process executes on the client machine and server process executes on the server machine and actually executes SQL statements submitted by the system user.
    - On the slide you can see one –to-one correspondence between the user and server process. This kind of connection is called as a “Dedicated Server” connection. Alternatively you can also use “Shared Server” connection. In a Dedicated Server process one user process connects to one server process. In a Shared Server connection the server process is shared amongst more than one user process.

➢**Creating a Session**

- A session is a “specific connection” of a user to an Oracle server.
  - →The session starts when the user is validated by the Oracle server.
  - →The session ends when the user logs out(disconnect) or when there is an abnormal termination (network failure or client machine failure).
- For a given database user, many “concurrent sessions” are possible if the user logs on from many tools, applications, or terminals at the same time. The DBA can limit the number of concurrent sessions.

It the Oracle Server is not running due to whatever reason, the user trying to connect will get “Oracle not available” error.

In one of the earlier lessons, we also talked about using HostString or connect string to connect to Oracle database. Let us understand about this.

On the client exists a tnsnames.ora file. This file is used by Oracle client to connect to oracle server/database. To connect to the database you provide hoststring or connect string thorugh SQL*Plus

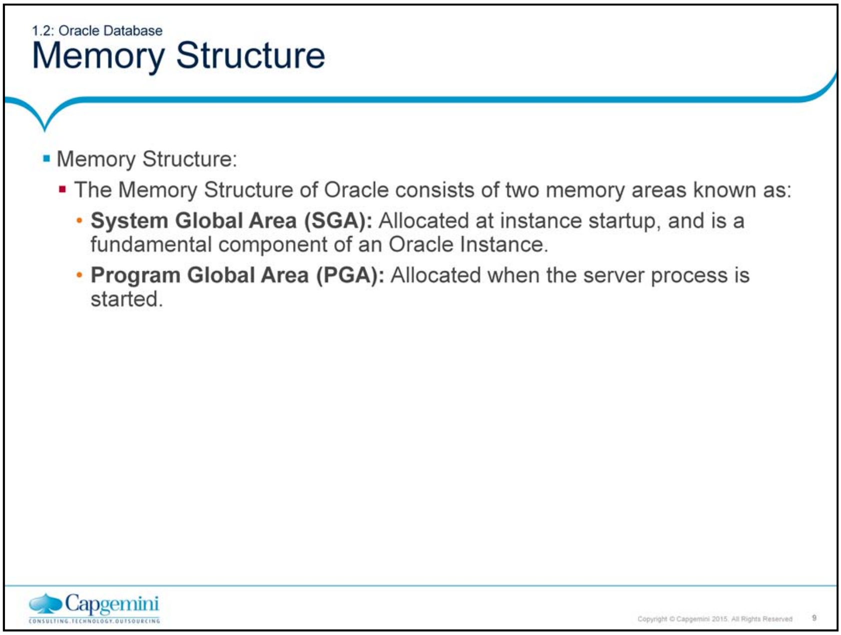
For example : scott/tiger@trgdb

This connect identifier i.e trgdb as in the above example is resolved by Oracle client by looking into the tnsnames.ora file which is placed on the client machine. The tnsnames.ora provides the network resolution for connect identifier used by Oracle clients and applications.

Below is a sample entry of tnsnames.ora file. Address_List and Address specifies the address of the server and port to which the client needs to connect The connect_data specifies the service name i.e sid of the database.

```
TRGDB =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.224.26)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = trgdb)
    )
  )
```

Note: SID is a unique name for the Oracle database instance which is running on the server



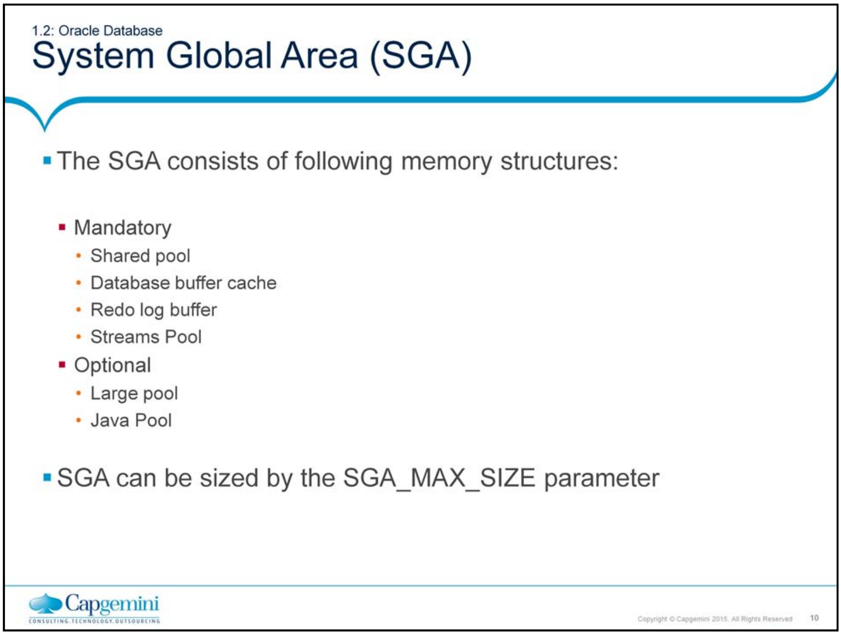
1.2: Oracle Database

Memory Structure

- Memory Structure:
  - The Memory Structure of Oracle consists of two memory areas known as:
    - **System Global Area (SGA):** Allocated at instance startup, and is a fundamental component of an Oracle Instance.
    - **Program Global Area (PGA):** Allocated when the server process is started.

<u>Memory Structure</u>:

- Memory structure of Oracle consists of two memory areas known as:
  - **System Global Area (SGA):** It is allocated at instance startup, and is a fundamental component of an Oracle Instance.
  - **Program Global Area (PGA):** It is allocated when the server process is started.



1.2: Oracle Database

System Global Area (SGA)

- The SGA consists of following memory structures:
  - Mandatory
    - Shared pool
    - Database buffer cache
    - Redo log buffer
    - Streams Pool
  - Optional
    - Large pool
    - Java Pool
- SGA can be sized by the SGA_MAX_SIZE parameter

10

<u>System Global Area (SGA)</u>:

- System Global Area (SGA) is a group of “shared memory structures” that contain data and control information for one Oracle database instance. When “multiple users” are connected to the same instance, the data in the SGA is “shared by all users”. Hence, it is appropriately called “Shared Global Area”.
- Oracle allocates memory for the SGA, when the database instance is started and returns the memory when the instance is shut down. The maximum size of the SGA is determined by SGA_MAX_SIZE initialization parameter in the initInstanceName.ora file or server parameter (SPFILE) file.
- The following statement can be used to view SGA memory allocations:

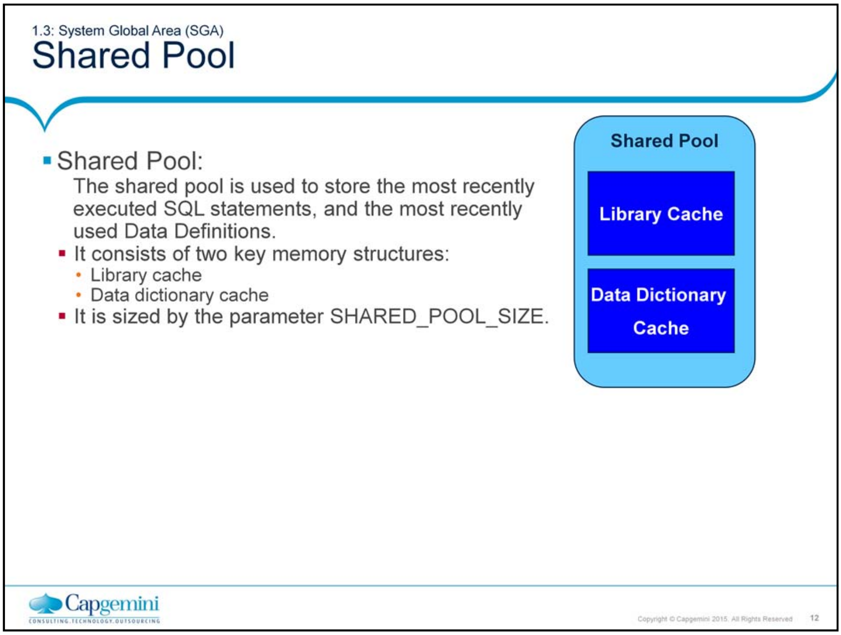
  SHOW SGA;

➢ Total System Global Area	36437964 bytes
➢ Fixed Size	6543794 bytes
➢ Variable Size	19521536 bytes
➢ Database Buffers	16777216 bytes
➢ Redo Buffers	73728 bytes

- Oracle 9i onwards Oracle Server user a Dynamic SGA. Memory structures for the SGA can be made without shutting down the database

System Global Area (SGA) (contd.):

- There are various initialization parameters to affect the amount of memory allocated to SGA.
    - SGA_MAX_SIZE: As mentioned earlier it is used to set the limit on the amount of memory allocated to SGA. A typical value could be 1 GB. If the value for this parameter set in the parameter file is less than the sum of memory allocated for all components within SGA either explicitly in the parameter file or default, then the database ignores the setting for SGA_MAX_SIZE. For optimal performance, the entire SGA should fit in real memory to avoid paging to/from disk by the operating system.
    - Other parameters which influence the SGA size are DB_CACHE_SIZE,LOG_BUFFER,SHARED_POOL etc… As we go along we will cover the parameters in the respective memory structures.
- Memory allocated to SGA is contiguous virtual memory unit termed as granules. Granule size depends on the size of SGA. If the size of SGA is less than 1 GB in total, each granule is 4 MB and it is greater the 1 GB than each granule is 16 MB. These granules are assigned to each memory structure. The actual number of granules assigned to each of these memory components can be determined by querying v$BUFFER_POOL.
- Automatic Shared Memory Management: Prior to Oracle 10g, the DBA had to manually specify SGA component sizes through the initialization parameters, like SHARED_POOL_SIZE, DB_CACHE_SIZE and so on. With Automatic Shared Memory Management, the DBA can specify the total SGA memory available through SGA_TARGET initialization parameter. The Oracle database automatically distributes this memory among various subcomponents to ensure most optimum memory utilization. With automatic memory management, the different SGA components are flexible sized to adapt to SGA available. Once DBA has set this parameter, DBA can forget about the sizes of individual components. No out of memory errors are generated unless the system has actually run out of memory. Manual tuning effort is reduced.
- The SGA_TARGET parameter reflects the total size of the SGA and include the following components:-
    - Fixed SGA and other internal allocations needed by Oracle
    - Log Buffer
    - Shared Pool
    - Database Buffer Cache (Keep and Recycle buffer caches, if specified)
    - Java Pool
    - Streams Pool
- If the value of SGA_TARGET is greater than SGA_MAX_SIZE at startup, then the SGA_MAX_SIZE value is bumped up to accommodate SGA_TARGET. The shared pool, Java pool, large pool and buffer cache are automatically sized in Oracle 10g if SGA_TARGET is set. The other SGA components like Keep/Recycle buffer caches, any additional buffer caches and streams pool is not automatically sized and the DBA must specify their sizes explicitly



1.3: System Global Area (SGA)

Shared Pool

- Shared Pool:

  The shared pool is used to store the most recently executed SQL statements, and the most recently used Data Definitions.
  - It consists of two key memory structures:
    - Library cache
    - Data dictionary cache
  - It is sized by the parameter SHARED_POOL_SIZE.

Shared Pool
Library Cache
Data Dictionary Cache

12

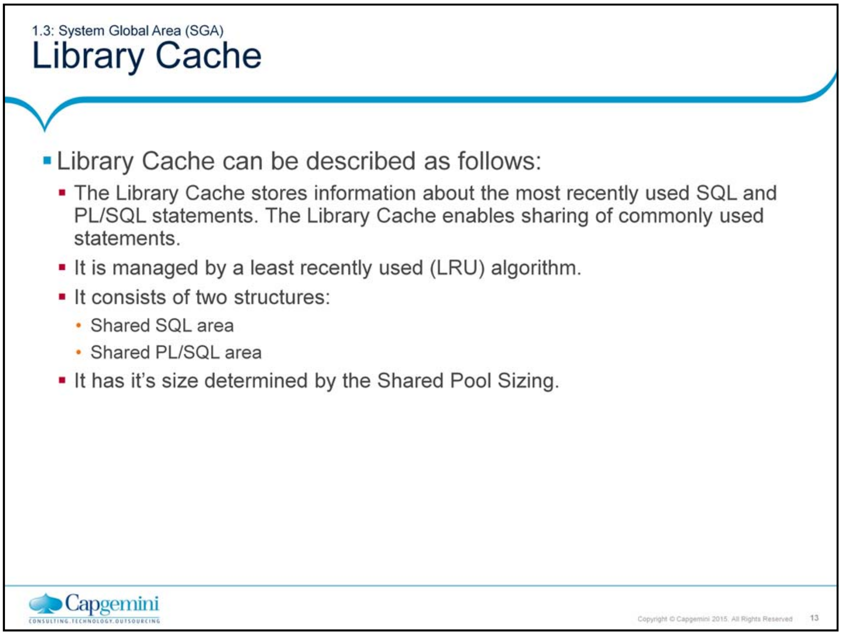
<u>Shared Pool</u>:

The Shared Pool is a memory structure shared by all users. It consists of both fixed and variable structures. The variable component grows and shrinks depending on the demands placed on memory size by users and application programs. The Shared Pool includes the Library Cache and Data Dictionary Cache.

Memory can be allocated to the Shared Pool by the parameter SHARED_POOL_SIZE in the parameter file. The default value is 8 MB on 32 bit platforms and 64MB on 64-bit platforms. Increasing the value of this parameter increases the amount of memory reserved for the shared pool. You can alter the size of the shared pool dynamically with the

ALTER SYSTEM SET SHARED_POOL_SIZE=20M

You cannot exceed the maximum size of the SGA. The shared pool stores the most recently executed SQL statement and used data definitions. This will help is reuse of the same SQL statements executed by the user. This helps in performance improvement.



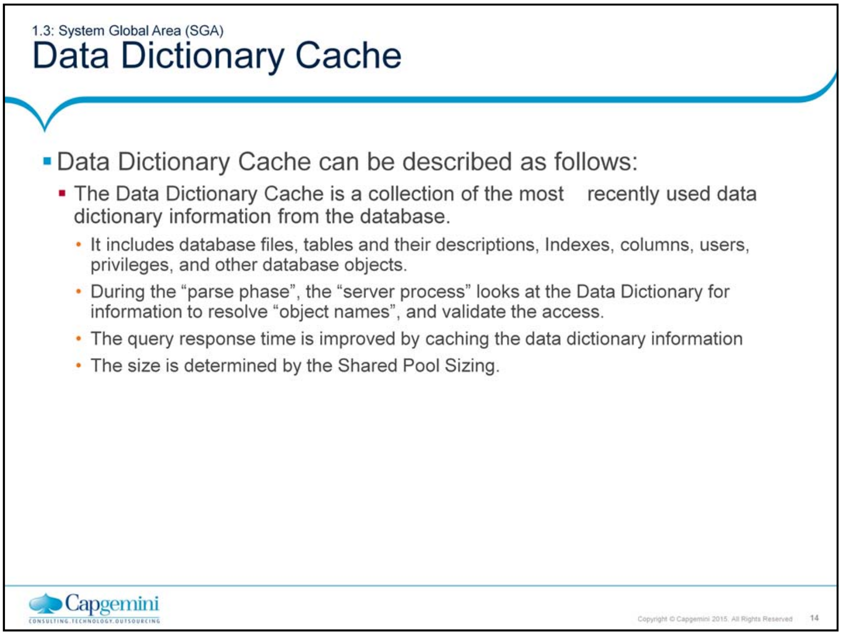
1.3: System Global Area (SGA)

Library Cache

- Library Cache can be described as follows:
  - The Library Cache stores information about the most recently used SQL and PL/SQL statements. The Library Cache enables sharing of commonly used statements.
  - It is managed by a least recently used (LRU) algorithm.
  - It consists of two structures:
    - Shared SQL area
    - Shared PL/SQL area
  - It has it's size determined by the Shared Pool Sizing.

<u>Library Cache</u>:

- Library Cache contains statement text, parsed code, and execution plan. It contains fully parsed or compiled representations of PL SQL blocks (procedures, triggers, etc) and SQL statements. The library cache size is based on the sizing defined for the shared pool. Memory is allocated when a statement is parsed or a program unit is called. If the Shared Pool is too small the size of the library cache is also affected. As a result the statement definitions are continually purged in order to have space to load new SQL and PL/SQL statements into the library cache, which affects performance. The library cache is managed by a least recently used (LRU) algorithm. As the cache is filled, less recently used execution paths and parse trees are removed from the library cache to make room for new entries. If the SQL and PL/SQL statements that are oldest and not reused, eventually age out.
- The library cache consists of two memory structures:
  - ➢ Shared SQL: This structure stores and shares the execution plan and parse tree for SQL statements run against the database. If the user executes an identical SQL statement then it is able to take benefit of the parse information available in the shared SQL area to accelerate the execution. To ensure that statements use Shared SQL area whenever possible, the text, schema, and bind variables should be exactly same.
  - ➢ Shared PL/SQL: This structure stores and shares the most recently used parsed and compiled PL/SQL statements like functions, procedures , packages & triggers.



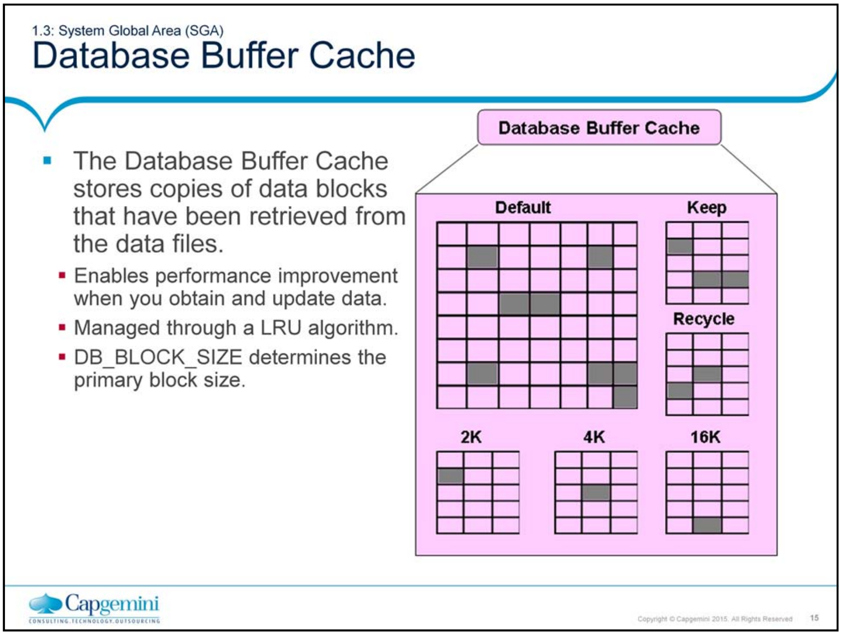
1.3: System Global Area (SGA)

Data Dictionary Cache

- Data Dictionary Cache can be described as follows:
  - The Data Dictionary Cache is a collection of the most recently used data dictionary information from the database.
    - It includes database files, tables and their descriptions, Indexes, columns, users, privileges, and other database objects.
    - During the “parse phase”, the “server process” looks at the Data Dictionary for information to resolve “object names”, and validate the access.
    - The query response time is improved by caching the data dictionary information
    - The size is determined by the Shared Pool Sizing.

Data Dictionary Cache:

- The data dictionary cache is sometimes also referred to as row cache. The overall response time of the query improves by caching information from the data dictionary. Whenever this information is needed for execution of a SQL statement it is read from the database and stored in the cache.
- The size of the data dictionary is dependent on the size of shared pool. If the data dictionary is too small, then the dictionary tables residing on the disk will have to repeatedly queried for information needed which will slow down the response time for queries.



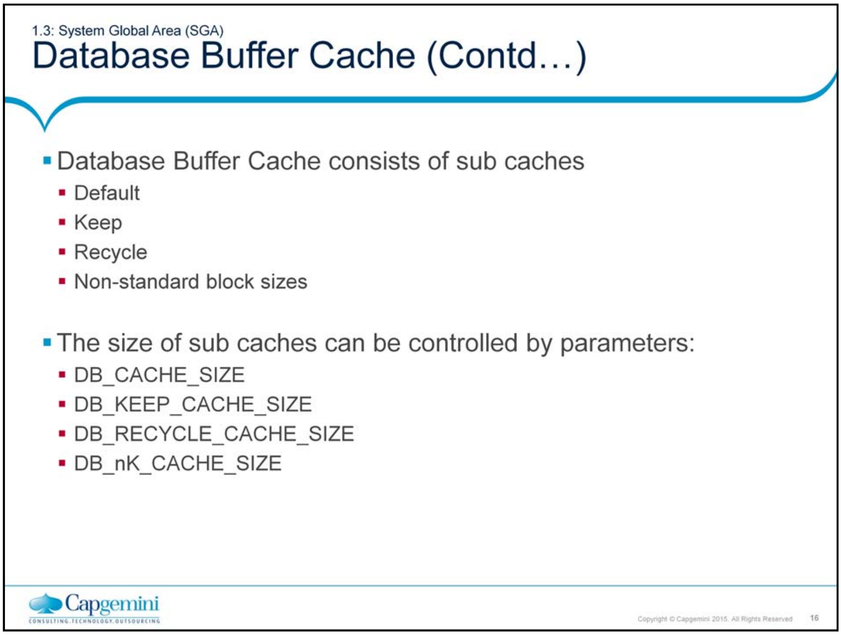
1.3: System Global Area (SGA)

Database Buffer Cache

- The Database Buffer Cache stores copies of data blocks that have been retrieved from the data files.
  - Enables performance improvement when you obtain and update data.
  - Managed through a LRU algorithm.
  - DB_BLOCK_SIZE determines the primary block size.

<u>Database Buffer Cache</u>:

- The Database Buffer cache (DB cache) is a large memory structure which stores copies of actual data blocks retrieved from datafiles for queries and DML commands. Whenever a query is issued the server process first looks in the DB cache to determine if the requested information happens to be already located in memory, if not available in memory the server process retrieves the information from disk and stores it in the cache.
- The data read from the disk is read one block at a time and not a row at a time. Blocks in the DB cache is stored according to the LRU algorithm and are aged out of memory if a buffer cache block is not used in order to provide space for the insertion of newly needed database blocks. Before reading a block from the database the process must find a free buffer. As you can see in the figure on the slide, the DB cache is made of many different pools
- The block size for a database is set when a database is created and is specified by DB_BLOCK_SIZE parameter. The size of each buffer in the default, keep, recycle pool is influenced by this parameter.



1.3: System Global Area (SGA)

Database Buffer Cache (Contd…)

- Database Buffer Cache consists of sub caches
  - Default
  - Keep
  - Recycle
  - Non-standard block sizes

- The size of sub caches can be controlled by parameters:
  - DB_CACHE_SIZE
  - DB_KEEP_CACHE_SIZE
  - DB_RECYCLE_CACHE_SIZE
  - DB_nK_CACHE_SIZE

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

16

Database Buffer Cache:

• The DB cache also consists of independent sub-caches for non-standard block sizes. The size of the DB cache can be controlled by the following parameters:

DB_CACHE_SIZE : Sizes the default buffer pool, it cannot be set to 0. It allows to dynamically change the memory allocated to cache

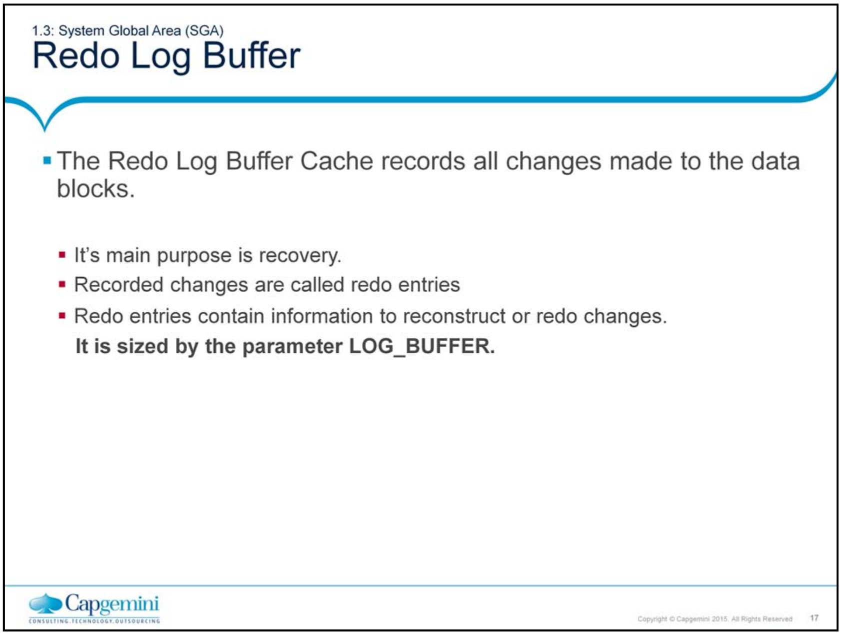
DB_KEEP_CACHE_SIZE: Sizes the KEEP buffer pool. This pool holds on to blocks in memory that are more likely to be reused. For example, table data containing username and passwords.

DB_RECYCLE_CACHE_SIZE: Sizes the RECYCLE buffer pool. This pool stores data that have little chance of being reused. Thus the data blocks are quickly removed from memory when not needed.

DB_nK_CACHE_SIZE: an Oracle database can also be created with non-standard block sizes apart from the standard block sizes. You can create upto 4 non-standard block sizes from 2KB to 32KB. To size the non-standard buffer pools within the DB cache this parameter can be used.

To change the size dynamically you can use

ALTER SYSTEM SET DB_CACHE_SIZE=100M



1.3: System Global Area (SGA)

Redo Log Buffer

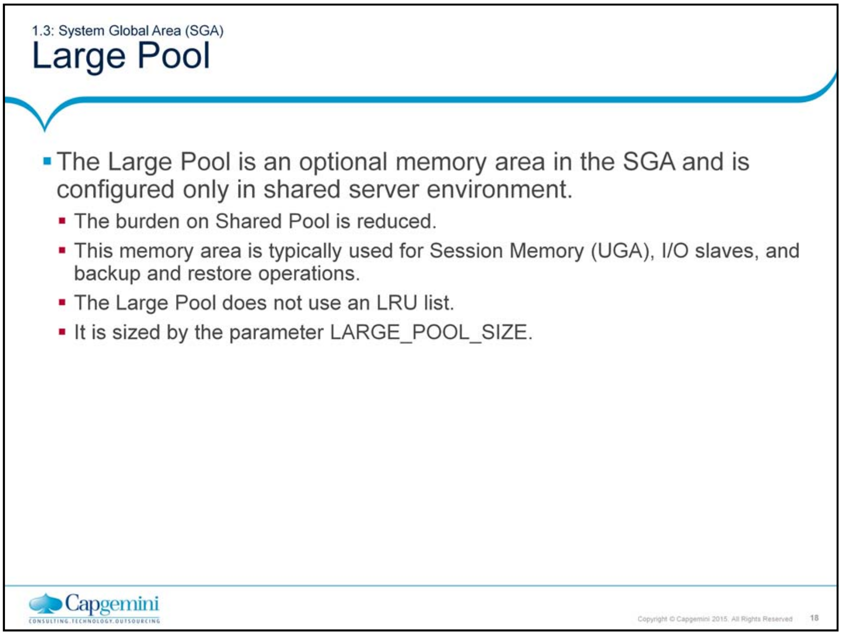
- The Redo Log Buffer Cache records all changes made to the data blocks.
  - It's main purpose is recovery.
  - Recorded changes are called redo entries
  - Redo entries contain information to reconstruct or redo changes.

  It is sized by the parameter LOG_BUFFER.

Redo Log Buffer :

- The Redo Log buffer memory stores images of all changes made to database data blocks. It is a circular buffer that is used over and over. As the buffer fills the redo entries are flushed out of the memory.
- A data block typically stores several rows and change in any value of the row will require the redo entry to be created. The redo entries contain information necessary to recreate the data prior to the change which was done by INSERT,DELETE,UPDATE,CREATE, ALTER, or DROP
- To size the Redo Log Buffer you can use

  ALTER SYSTEM SET LOG_BUFFER=95M



1.3: System Global Area (SGA)

Large Pool

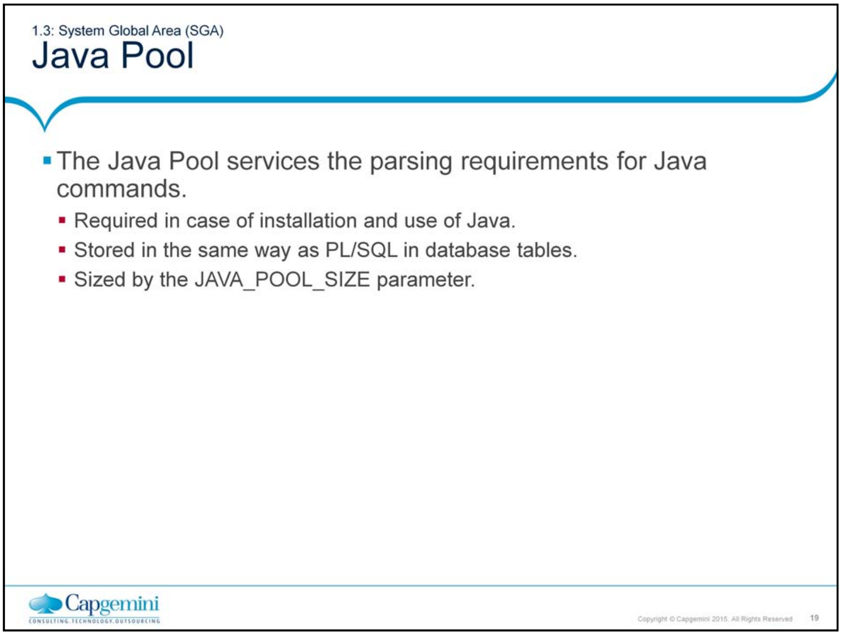
- The Large Pool is an optional memory area in the SGA and is configured only in shared server environment.
  - The burden on Shared Pool is reduced.
  - This memory area is typically used for Session Memory (UGA), I/O slaves, and backup and restore operations.
  - The Large Pool does not use an LRU list.
  - It is sized by the parameter LARGE_POOL_SIZE.

Large Pool:

- The Large Pool is an optional memory area that reduces the burden on the Shared Pool. Whenever users connect through a Shared Server process, Oracle will need to allocate space in the Shared Pool for storing information about the user processes and server processed they are connected to. Hence memory for Large Pool can be allocate in this case so that Shared Pool will not have to give memory for storing this additional information.
- If the Large Pool is allocated then it is used for:

  - Allocating space for session memory requirements from User Global Area (UGA) where the Shared Server environment is configured. (In dedicated server environment the UGA is part of PGA which is covered later in the lesson)

  - Backup and Restore operations by the Recovery Manager
- The LARGE_POOL_SIZE is not a dynamic parameter and it does not use LRU algorithm to manage memory.



1.3: System Global Area (SGA)

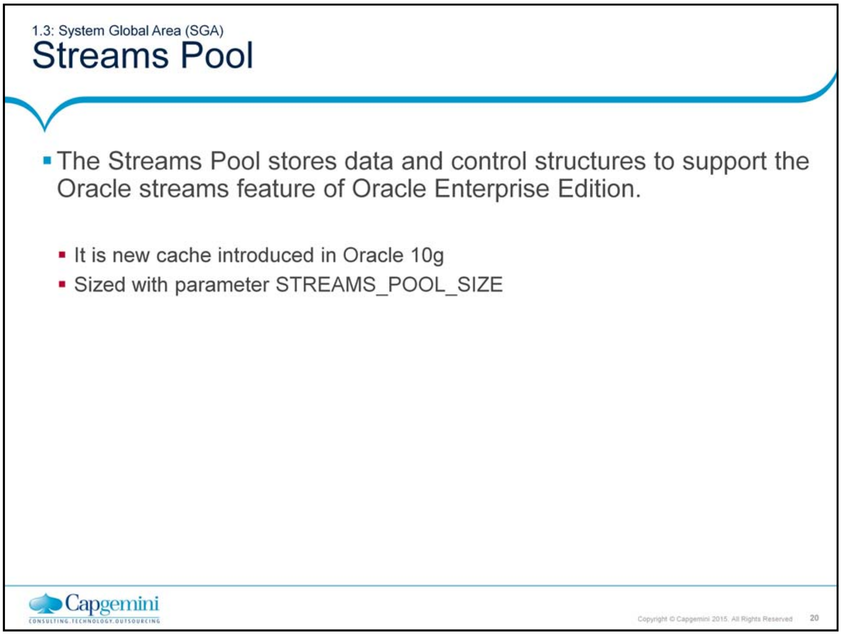
Java Pool

- The Java Pool services the parsing requirements for Java commands.
  - Required in case of installation and use of Java.
  - Stored in the same way as PL/SQL in database tables.
  - Sized by the JAVA_POOL_SIZE parameter.

19

Java Pool:

- The Java Pool is an optional setting and is required if the database has Oracle Java installed and is using Oracle Java Virtual Machine (JVM). The Java Pool even of not defined is set to default size of 24 MB. The Java Pool is used to parse Java commands and store data associated with Java commands. This is similar to storing SQL and PL/SQL code in Shared pool.



1.3: System Global Area (SGA)

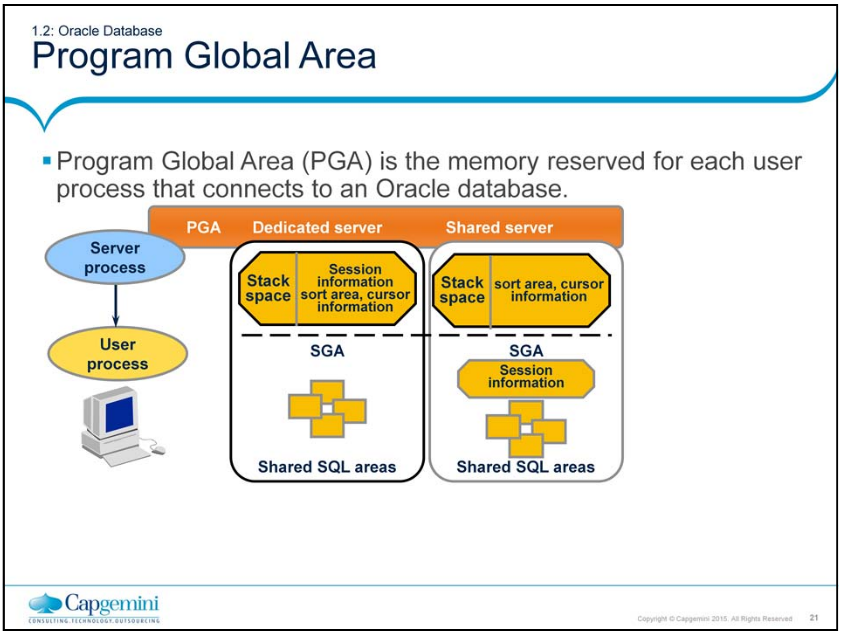
Streams Pool

- The Streams Pool stores data and control structures to support the Oracle streams feature of Oracle Enterprise Edition.
  - It is new cache introduced in Oracle 10g
  - Sized with parameter STREAMS_POOL_SIZE

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Streams Pool:

Oracle Streams manage sharing of data and events in a distributed environment. If the streams pool is set to 0 then the memory for streams operation is allocated from Shared Pool memory up to 10%



1.2: Oracle Database

Program Global Area

- Program Global Area (PGA) is the memory reserved for each user process that connects to an Oracle database.

21

<u>Program Global Area (PGA)</u>:

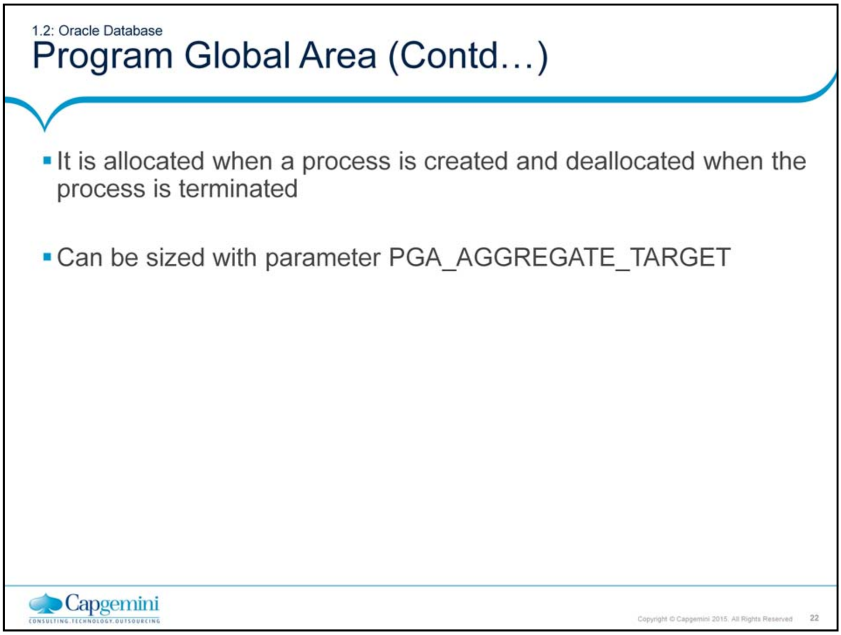
- The Program Global Area is part of memory allocated outside the SGA. It is sometimes also termed as Process Global Area.
- The PGA stores data and control information related to Server Process.
- Unlike the SGA, the PGA is an area that is stored only by one process.
- The PGA contains the following

  - Stack Space: Contains Session variables

  - Cursor State: Indicates the stage in the processing of the SQL statements that are currently used by the session

  - Session Information: Includes privileges for user, session variables, performance statistics for the session

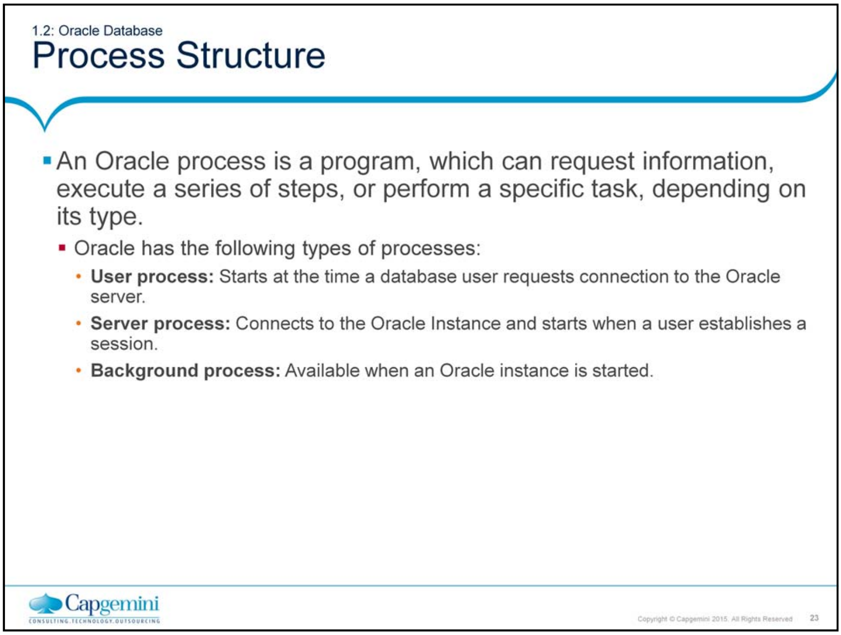
  - Sort Area: Used for storing sorting information required to process the SQL statements
- In a shared server configuration some of these structures are stored in the SGA. As mentioned earlier in a shared server environment multiple user processes share one server process. If a large pool exists within SGA then Large Pool is used for storage else it is stored in Shared Pool.



1.2: Oracle Database

Program Global Area (Contd…)

- It is allocated when a process is created and deallocated when the process is terminated
- Can be sized with parameter PGA_AGGREGATE_TARGET



1.2: Oracle Database

Process Structure

- An Oracle process is a program, which can request information, execute a series of steps, or perform a specific task, depending on its type.
  - Oracle has the following types of processes:
    - **User process:** Starts at the time a database user requests connection to the Oracle server.
    - **Server process:** Connects to the Oracle Instance and starts when a user establishes a session.
    - **Background process:** Available when an Oracle instance is started.

Capgemini CONSULTING.TECHNOLOGY.OUTSOURCING

23

<u>Process structure</u>:

Oracle uses the following types of processes:

- **User Process**

  A user process is a program that requests interaction with the Oracle server.

  - ➢ It must first establish a connection.
  - ➢ It does not interact directly with the Oracle server.

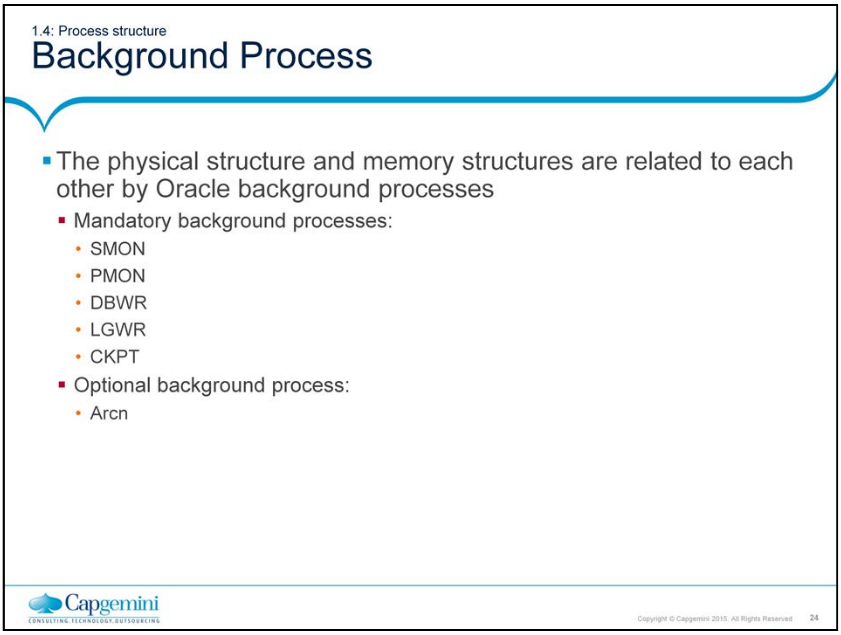
- **Server Process**

  A server process is a program that directly interacts with the Oracle server.

  - ➢ It fulfills calls generated and returns results.
  - ➢ It can be dedicated or shared server.

- **Background Processes**

  The relationship between the “physical structures” and “memory structures” is maintained and enforced by background processes in Oracle.



1.4: Process structure

Background Process

- The physical structure and memory structures are related to each other by Oracle background processes
  - Mandatory background processes:
    - SMON
    - PMON
    - DBWR
    - LGWR
    - CKPT
  - Optional background process:
    - Arcn

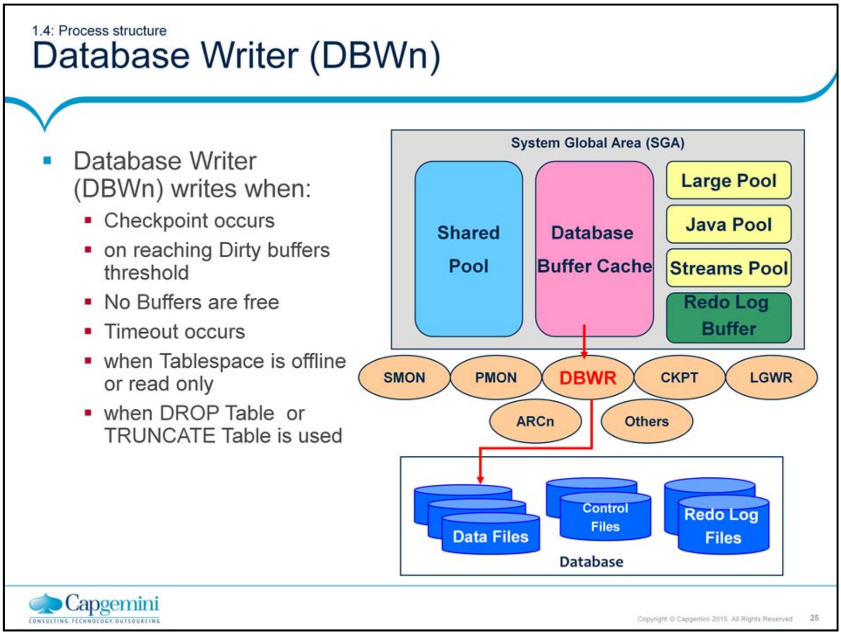
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

24

Background Process:

The Oracle architecture has five mandatory background process as mentioned on the slide.

Apart from that Oracle also has optional process which are available only if the option is being used in the Database. ARCn is the most common optional background process



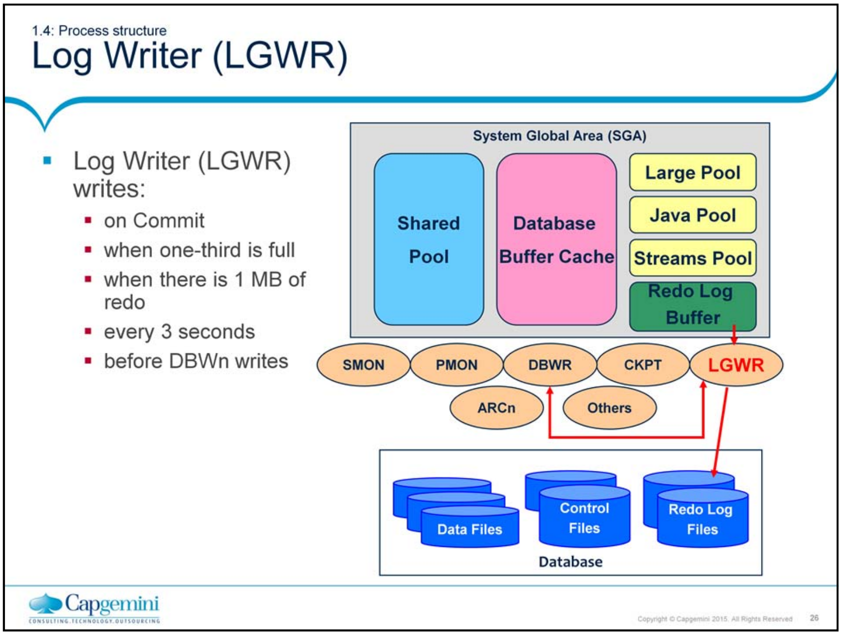
1.4: Process structure

Database Writer (DBWn)

- Database Writer (DBWn) writes when:
  - Checkpoint occurs
  - on reaching Dirty buffers threshold
  - No Buffers are free
  - Timeout occurs
  - when Tablespace is offline or read only
  - when DROP Table or TRUNCATE Table is used

<u>Database Writer</u>:

- The server process records changes to rollback and data blocks in the Buffer Cache. Database Writer (DBWn) writes the dirty buffers from the Database Buffer Cache to the Data files. To ensure that sufficient number of free buffers are available in DB cache is also taken care by DBWn. Free buffers are buffers that can be overwritten when server processes need to read in blocks from the data files.
- Database performance is improved because server processes make changes only in the Buffer Cache.
- DBWn delays writing to the data files until one of the following events occur:
  - ➢ System reaching incremental or normal Checkpoint
  - ➢ The number of dirty buffers reaching a threshold value
  - ➢ A process scanning a specified number of blocks, during the scan for free buffers, and cannot find any.
  - ➢ System placing a normal or temporary tablespace offline.
  - ➢ System placing a tablespace in read-only mode.
  - ➢ System is dropping or truncating a table.



1.4: Process structure

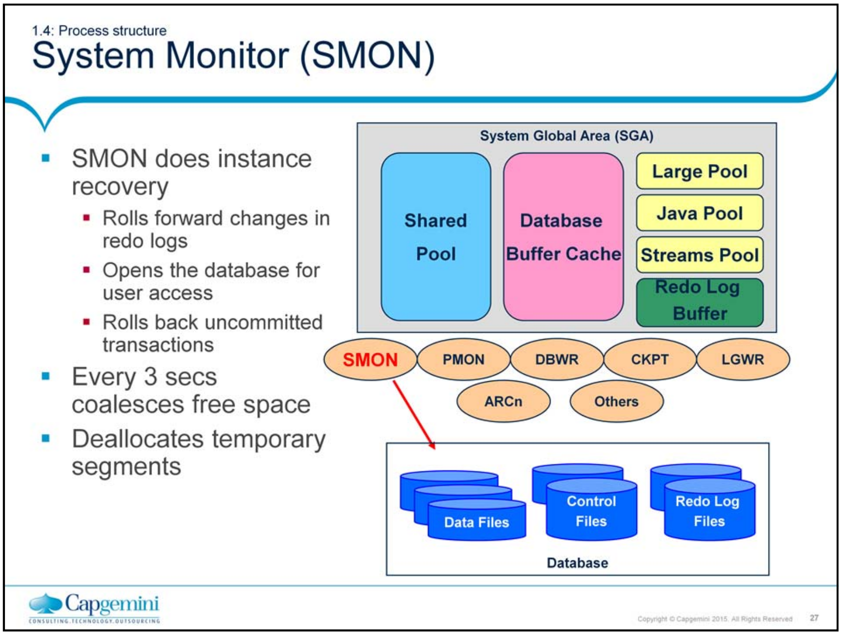
Log Writer (LGWR)

- Log Writer (LGWR) writes:
  - on Commit
  - when one-third is full
  - when there is 1 MB of redo
  - every 3 seconds
  - before DBWn writes

<u>LOG Writer</u>

- The Log Writer (LGWR) performs sequential writes from the Redo Log Buffer Cache to the Redo Log File under the following situations:
  - ➢ when a transaction commits
  - ➢ when the Redo Log Buffer Cache is one-third full
  - ➢ when there is more than one megabyte of change records in the Redo Log Buffer Cache
  - ➢ every 3 seconds
  - ➢ before DBWn writes modified blocks in the DB Cache to the data files
- Since the redo is needed for recovery, LGWR confirms the commit only after the redo is written to disk.
- LGWR can also call on DBWn to write to the data files.

Note: DBWn does not write to the online redo logs.



1.4: Process structure

System Monitor (SMON)

- SMON does instance recovery
  - Rolls forward changes in redo logs
  - Opens the database for user access
  - Rolls back uncommitted transactions
- Every 3 secs coalesces free space
- Deallocates temporary segments

System Global Area (SGA)

Shared Pool

Database Buffer Cache

Large Pool

Java Pool

Streams Pool

Redo Log Buffer

SMON PMON DBWR CKPT LGWR

ARCn Others

Data Files

Control Files

Redo Log Files

Database

Capgemini CONSULTING.TECHNOLOGY.OUTSOURCING

27

System Monitor (SMON):

- When the Oracle instance fails due to whatever reason, any information in the SGA which has not been written to disk is lost. SMON automatically performs instance recovery when the database is reopened. Whenever the instance recovery happens the following tasks are carried out:

  - Rolling forward to recover data that has not been recorded in the data files but has been recorded in the online redo logs. Due to SGA failure data is not written to disk. During this process SMON reads the redo files and applies changes from the redo logs to data blocks. Since all the committed transactions are have been written to redo logs this process completely recovers transactions.

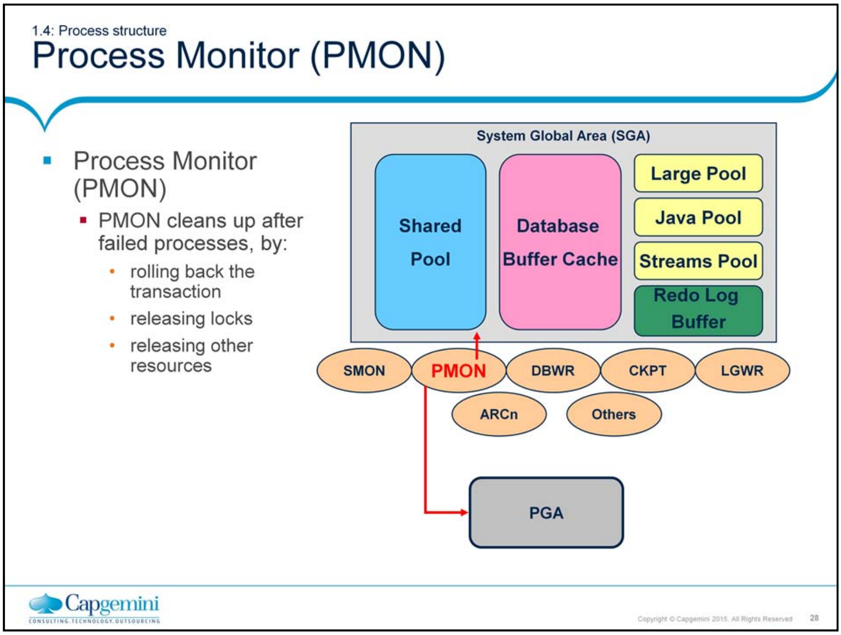
  - Opens the database so that users can log on and any data that is not locked by unrecovered transactions is available immediately

  - Rolling back uncommitted transactions. They are rolled back by SMON or by individual server processes as they access locked data.

- Apart from instance recover which is the main function of SMON, it also performs some maintenance tasks

  - It combines adjacent free space in the data files

  - It deallocates temporary segments to return them as free space data files.



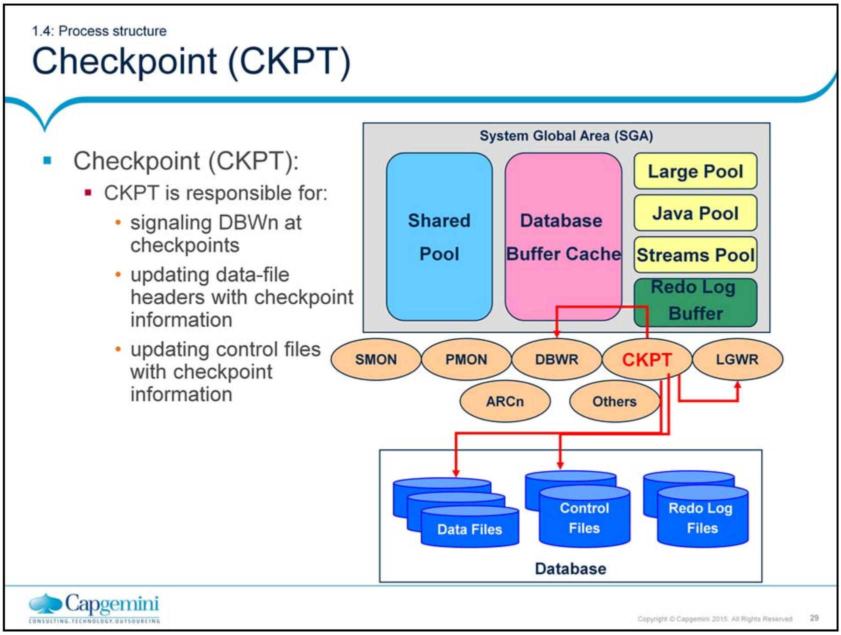
1.4: Process structure

Process Monitor (PMON)

- Process Monitor (PMON)
  - PMON cleans up after failed processes, by:
    - rolling back the transaction
    - releasing locks
    - releasing other resources

Process Monitor:

- The PMON background process cleans up after the failed processes, by:
  - rolling back the current transaction of the user
  - releasing all currently held table or row locks
  - freeing other resources that are currently reserved by the user



1.4: Process structure

Checkpoint (CKPT)

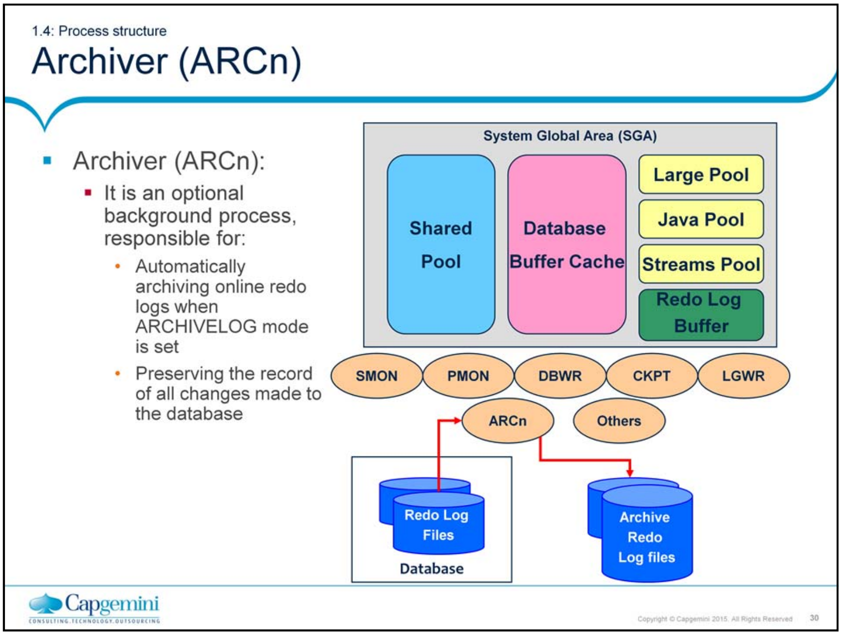
- Checkpoint (CKPT):
  - CKPT is responsible for:
    - signaling DBWn at checkpoints
    - updating data-file headers with checkpoint information
    - updating control files with checkpoint information

Capgemini CONSULTING.TECHNOLOGY.OUTSOURCING

29

Checkpoint (CKPT):

- An event called as a Checkpoint occurs when the Oracle background process DBWn writes all the modified Database Buffers in the SGA, including both committed and uncommitted data, to the data files.
- Checkpoints are implemented for the following reasons:
  - ➢ Checkpoints ensure that memory data blocks, which frequently change, are regularly written to data files.
    - Because of the Least Recently Used (LRU) algorithm of DBWn, a data block that changes frequently might never qualify as the least recently used block, and thus might never be written to disk if checkpoints do not occur.
  - ➢ In case, instance recovery is required, the redo log entries before the Checkpoint no longer need to be applied to the data files, since all database changes up to the Checkpoint have been recorded in the data files. Hence, checkpoints are useful because they can expedite instance recovery.
- **Note:** CKPT does not write data blocks to disk, or redo blocks to the online redo logs.



1.4: Process structure

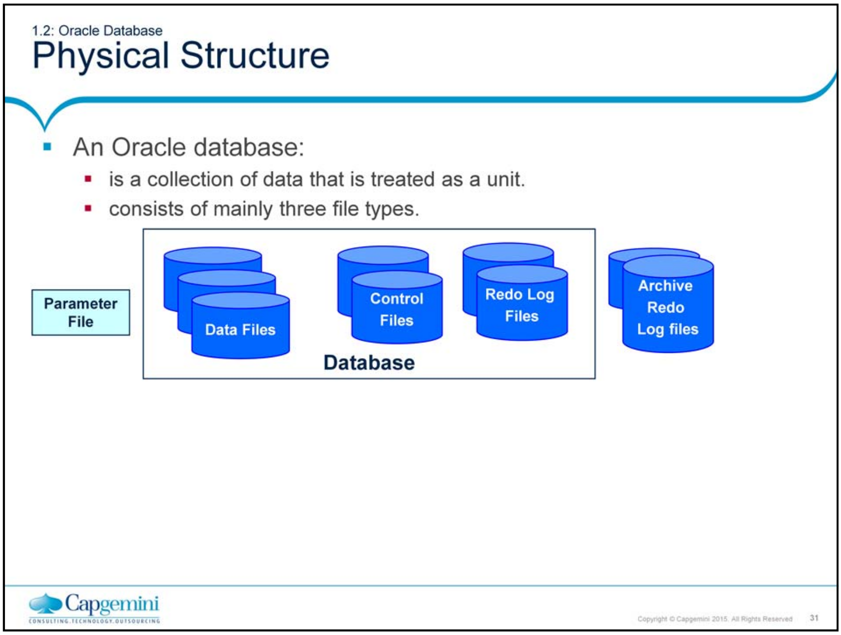
Archiver (ARCn)

- Archiver (ARCn):
  - It is an optional background process, responsible for:
    - Automatically archiving online redo logs when ARCHIVELOG mode is set
    - Preserving the record of all changes made to the database

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

<u>Archiver (ARCn)</u>

- Archiver (ARCn) is an optional background process. It is responsible for:
  - ➢ Automatically archiving online redo logs when ARCHIVELOG mode is set
  - ➢ Preserving the record of all changes made to the database
- ARCn is crucial for recovering a database after the loss of a disk. As online redo log files are 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, such that all 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.
- 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.
  - ➢ 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 is recovered in case there is an instance crash.



1.2: Oracle Database

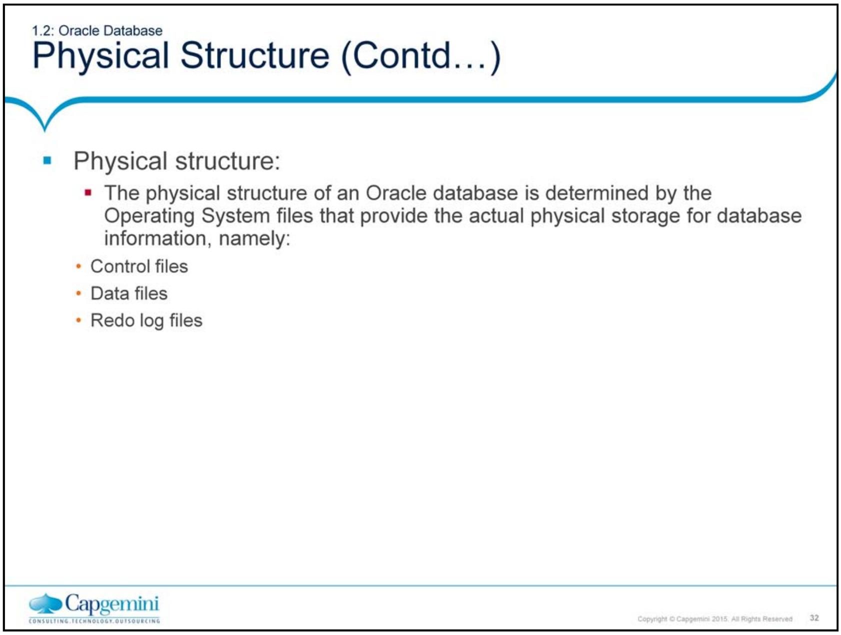
Physical Structure

- An Oracle database:
  - is a collection of data that is treated as a unit.
  - consists of mainly three file types.

31

Oracle Database:

- An Oracle database has a “logical structure” and a “physical structure”.



1.2: Oracle Database

Physical Structure (Contd…)

- Physical structure:
  - The physical structure of an Oracle database is determined by the Operating System files that provide the actual physical storage for database information, namely:
  - Control files
  - Data files
  - Redo log files

32

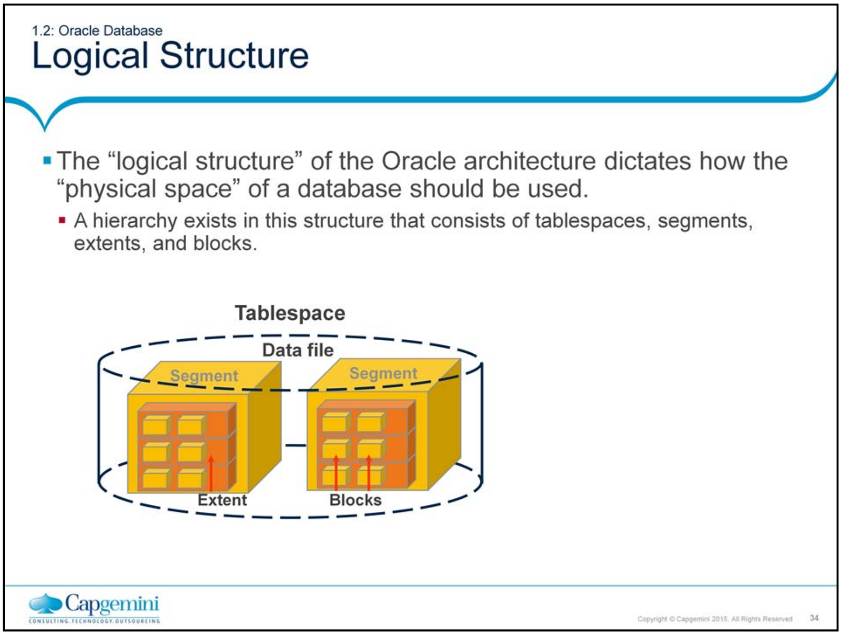
- **Physical structure**:
  - The physical structure of the database is the set of “operating system files” in the database.
  - An Oracle database consists of three file types.
    - **Control files** that contain information necessary to maintain and verify database integrity.
    - **Data files** that contain the actual data in the database.
    - **Redo logs** that contain a record of changes made to the database to enable recovery of the data in case of failures.
  - However, the physical structure of an Oracle database includes only three types of files: control files, data files, and redo log files.

Other Key File Structures

- The Oracle server uses other files, as well, that are not part of the database.
  - **Parameter file** defines the characteristics of an Oracle instance. For example: The parameter file contains “parameters” that size some of the memory structures in the SGA.

    To start an instance, the Oracle server must read the “initialization parameter file”.

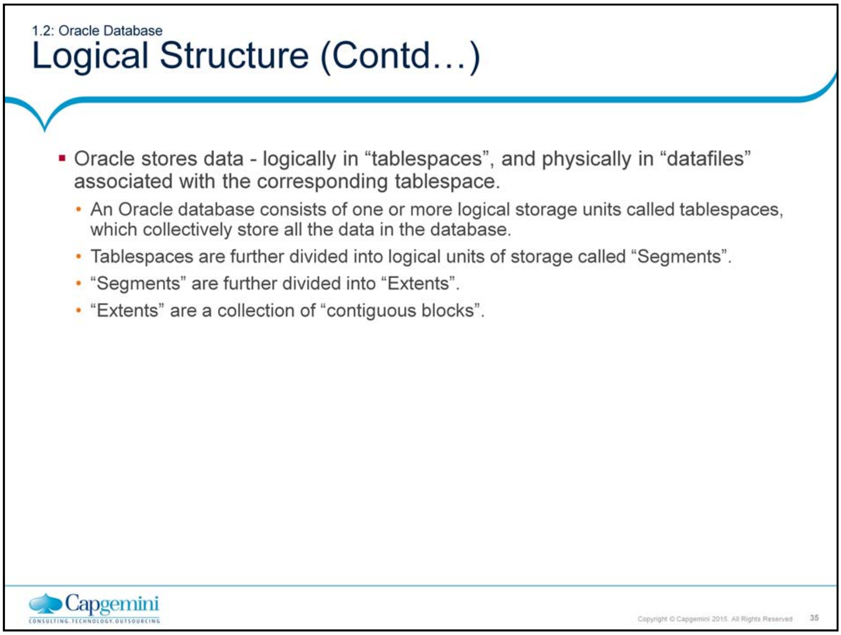
    There are two types of “initialization parameter files”:
    - **Static parameter file**, PFILE, commonly referred to as “initSID.ora”. The PFILE is a text file that can be maintained by using a standard operating system editor. The parameter file is read-only during “instance startup”. If the file is modified, the instance must be shut down, and restarted in order to make the new parameter values effective.
    - **Persistent parameter file**, SPFILE, commonly referred to as “spfileSID.ora”. SPFILE, new to Oracle9i, is a binary file. The file is not meant to be manually modified and must always reside on the “server side”. By default, the file is located in $ORACLE_HOME/dbs, and has a default name in the format of “spfileSID.ora”. Once the file is created it is maintained by the Oracle server. The SPFILE provides the ability to make changes to the database that are persistent across shutdown and startup.
  - **Password file** authenticates the users, who are privileged to start up and shut down an Oracle instance.
  - **Archived redo log files** are offline copies of the redo log files that may be necessary to recover from media failures.



1.2: Oracle Database

Logical Structure

- The "logical structure" of the Oracle architecture dictates how the "physical space" of a database should be used.
  - A hierarchy exists in this structure that consists of tablespaces, segments, extents, and blocks.



1.2: Oracle Database

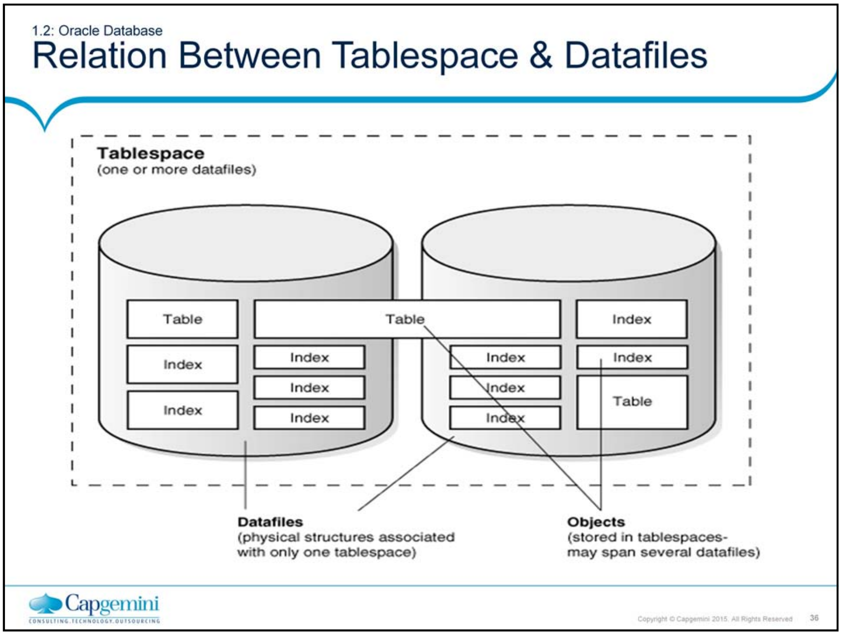
Logical Structure (Contd…)

- Oracle stores data - logically in “tablespaces”, and physically in “datafiles” associated with the corresponding tablespace.
  - An Oracle database consists of one or more logical storage units called tablespaces, which collectively store all the data in the database.
  - Tablespaces are further divided into logical units of storage called “Segments”.
  - “Segments” are further divided into “Extents”.
  - “Extents” are a collection of “contiguous blocks”.

Logical Structure:

- Oracle stores data - logically in “tablespaces”, and physically in “datafiles” associated with the corresponding tablespace.
- An Oracle database consists of one or more logical storage units called as “tablespaces”, which collectively store all the data of the database.
- Each “tablespace” in an Oracle database consists of one or more files called “datafiles”, which are physical structures that conform to the operating system in which Oracle is running.
  - A database's data is collectively stored in the datafiles that constitute each tablespace of the database.

  For example: The simplest Oracle database will have one tablespace and one datafile.
- Tablespaces are divided into logical units of storage called “segments”.
- Segments are further divided into “Extents”.
- Extents are a collection of “contiguous blocks”.



1.2: Oracle Database

Relation Between Tablespace & Datafiles

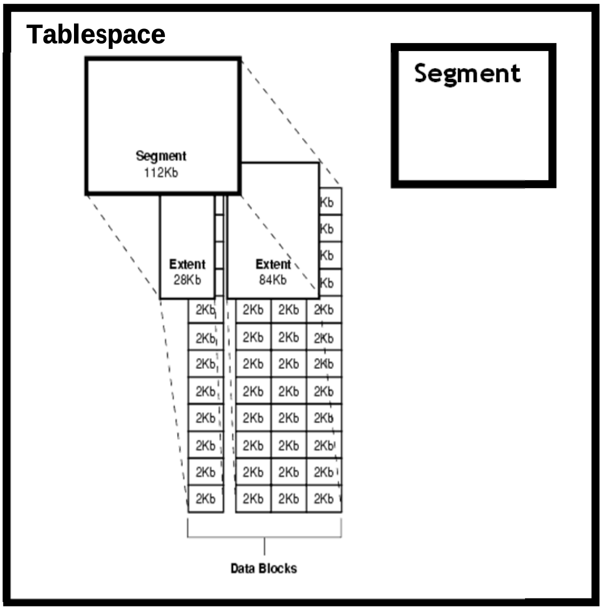
Datafiles
(physical structures associated with only one tablespace)

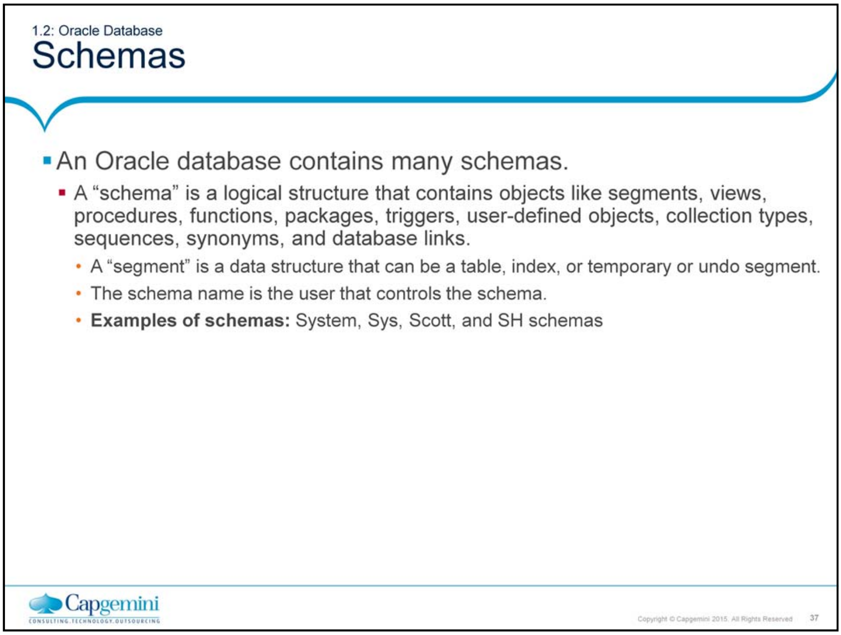
Objects
(stored in tablespaces-may span several datafiles)

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

36

<u>Logical components of an Oracle Database</u>:

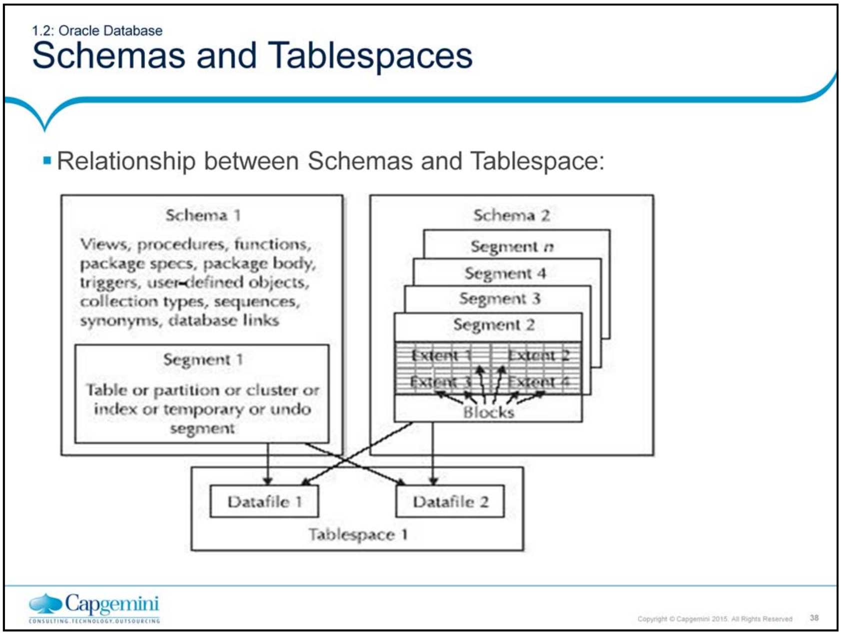




1.2: Oracle Database

Schemas

- An Oracle database contains many schemas.
  - A “schema” is a logical structure that contains objects like segments, views, procedures, functions, packages, triggers, user-defined objects, collection types, sequences, synonyms, and database links.
    - A “segment” is a data structure that can be a table, index, or temporary or undo segment.
    - The schema name is the user that controls the schema.
    - **Examples of schemas:** System, Sys, Scott, and SH schemas



1.2: Oracle Database

Schemas and Tablespaces

- Relationship between Schemas and Tablespace:

Schema 1
Views, procedures, functions, package specs, package body, triggers, user-defined objects, collection types, sequences, synonyms, database links
Segment 1
Table or partition or cluster or index or temporary or undo segment
Schema 2
Segment n
Segment 4
Segment 3
Segment 2
Extent 1
Extent 2
Extent 3
Extent 4
Blocks
Datafile 1
Datafile 2
Tablespace 1

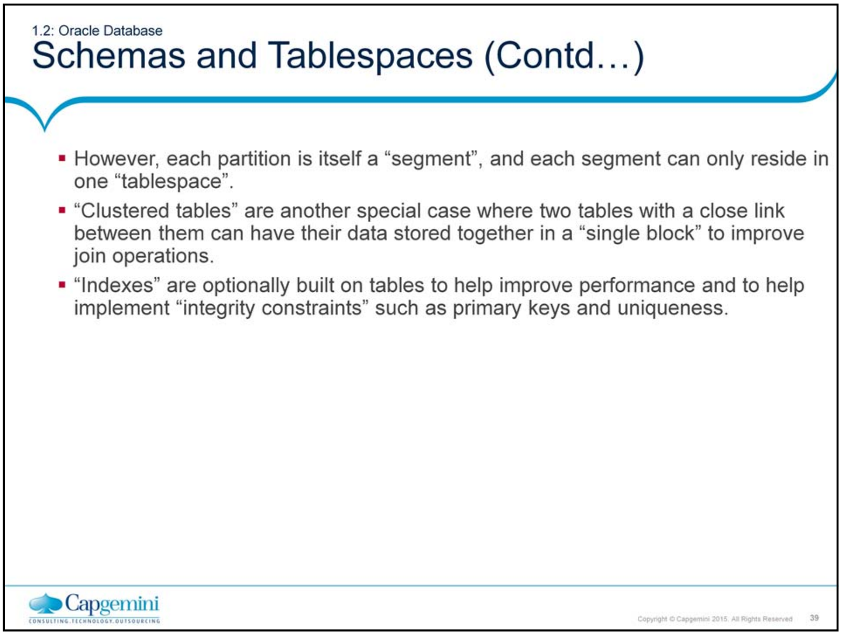
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

38

<u>Relationship between Schemas and Tablespace</u>:

- As shown in the slide, a “schema” can have many “segments” and many “segment types”.
- Each segment is a single instance of a table, partition, cluster, index, or temporary or undo segment.

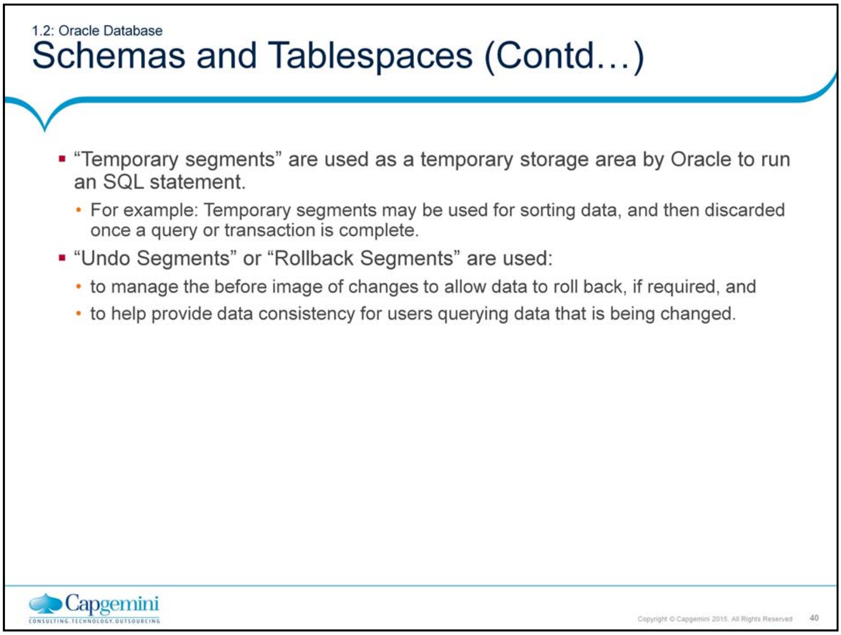
  For example: A table with two indexes is implemented as three segments in the schema.
- A “Segment” is further broken down into “Extents”, which are a collection of contiguous “data blocks”.
- As data is added to Oracle, it will first fill the blocks in the allocated Extents. Once these extents are full, new extents can be added to the segment as long as it is allowed by the available space.



1.2: Oracle Database

Schemas and Tablespaces (Contd…)

- However, each partition is itself a “segment”, and each segment can only reside in one “tablespace”.
- “Clustered tables” are another special case where two tables with a close link between them can have their data stored together in a “single block” to improve join operations.
- “Indexes” are optionally built on tables to help improve performance and to help implement “integrity constraints” such as primary keys and uniqueness.

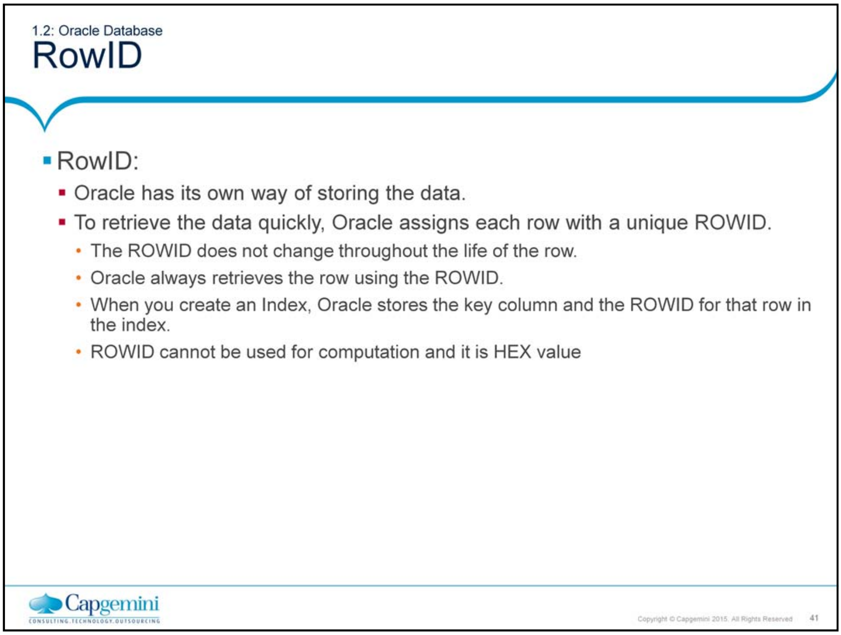


1.2: Oracle Database

Schemas and Tablespaces (Contd…)

- “Temporary segments” are used as a temporary storage area by Oracle to run an SQL statement.
  - For example: Temporary segments may be used for sorting data, and then discarded once a query or transaction is complete.
- “Undo Segments” or “Rollback Segments” are used:
  - to manage the before image of changes to allow data to roll back, if required, and
  - to help provide data consistency for users querying data that is being changed.

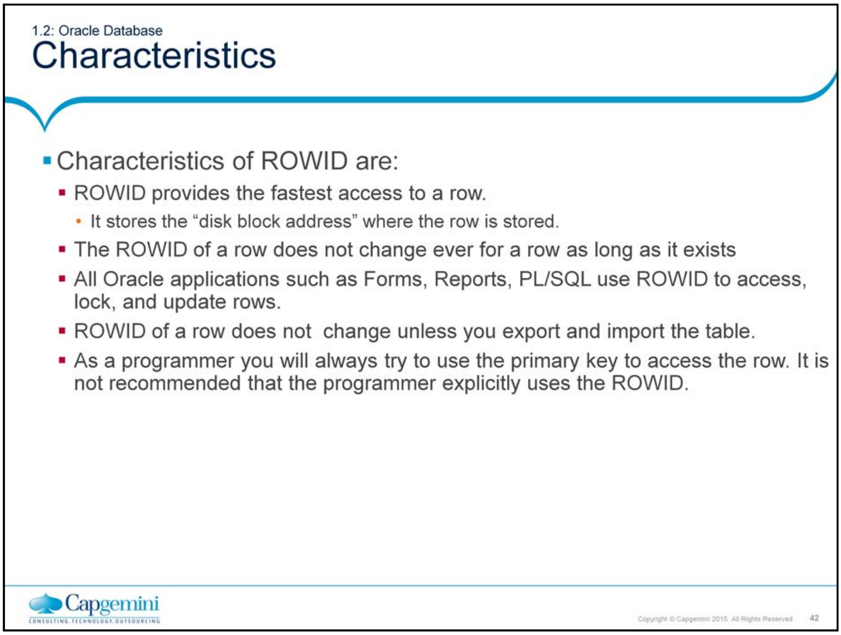
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING



1.2: Oracle Database

RowID

- RowID:
  - Oracle has its own way of storing the data.
  - To retrieve the data quickly, Oracle assigns each row with a unique ROWID.
    - The ROWID does not change throughout the life of the row.
    - Oracle always retrieves the row using the ROWID.
    - When you create an Index, Oracle stores the key column and the ROWID for that row in the index.
    - ROWID cannot be used for computation and it is HEX value

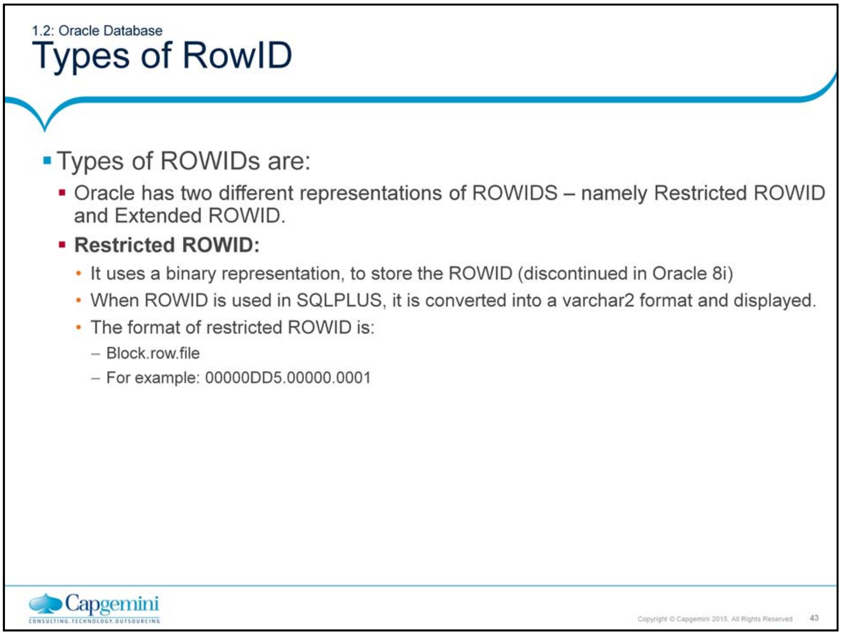


1.2: Oracle Database

Characteristics

- Characteristics of ROWID are:
  - ROWID provides the fastest access to a row.
    - It stores the “disk block address” where the row is stored.
  - The ROWID of a row does not change ever for a row as long as it exists
  - All Oracle applications such as Forms, Reports, PL/SQL use ROWID to access, lock, and update rows.
  - ROWID of a row does not change unless you export and import the table.
  - As a programmer you will always try to use the primary key to access the row. It is not recommended that the programmer explicitly uses the ROWID.

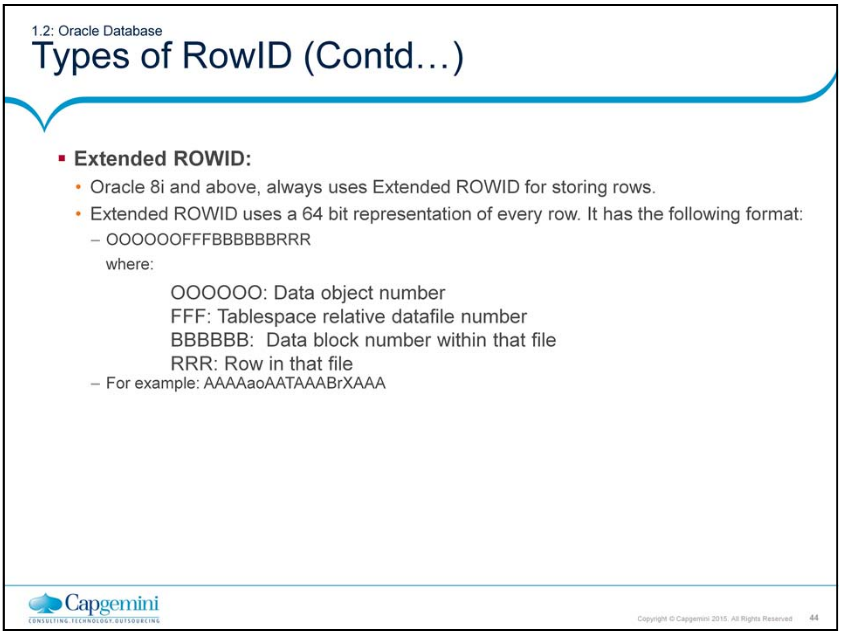
42



1.2: Oracle Database

Types of RowID

- Types of ROWIDs are:
  - Oracle has two different representations of ROWIDS – namely Restricted ROWID and Extended ROWID.
  - **Restricted ROWID:**
    - It uses a binary representation, to store the ROWID (discontinued in Oracle 8i)
    - When ROWID is used in SQLPLUS, it is converted into a varchar2 format and displayed.
    - The format of restricted ROWID is:
      - Block.row.file
      - For example: 00000DD5.00000.0001



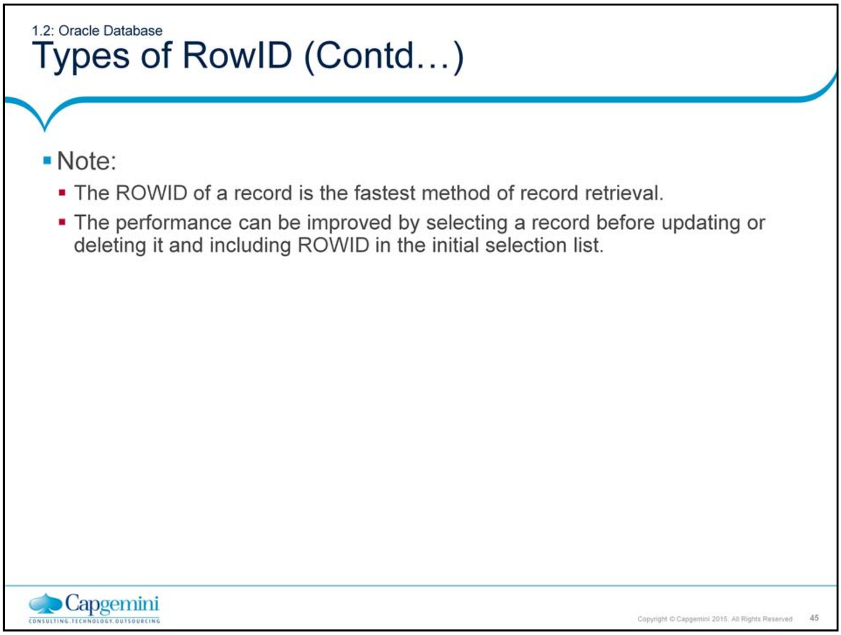
1.2: Oracle Database

Types of RowID (Contd…)

- **Extended ROWID:**
  - Oracle 8i and above, always uses Extended ROWID for storing rows.
  - Extended ROWID uses a 64 bit representation of every row. It has the following format:
    - OOOOOOFFFBBBBBBRRR

      where:

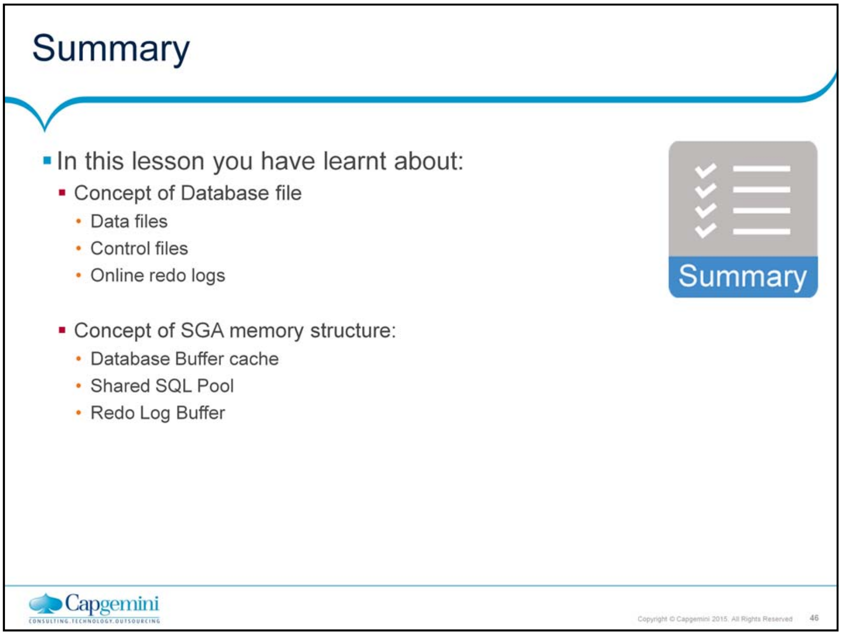
      OOOOOO: Data object number
      FFF: Tablespace relative datafile number
      BBBBBB: Data block number within that file
      RRR: Row in that file
    - For example: AAAAaoAATAAABrXAAA



1.2: Oracle Database

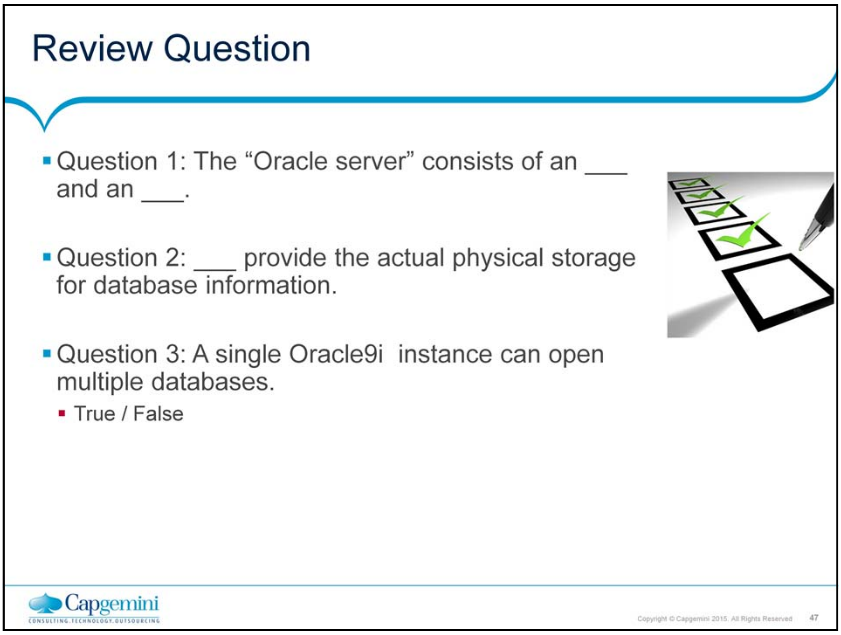
Types of RowID (Contd…)

- Note:
  - The ROWID of a record is the fastest method of record retrieval.
  - The performance can be improved by selecting a record before updating or deleting it and including ROWID in the initial selection list.



Summary

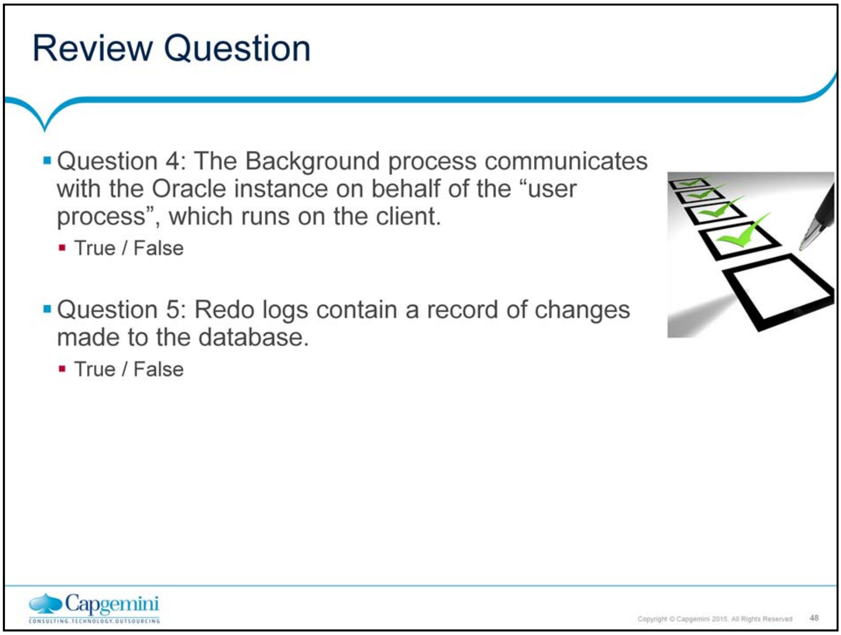
- In this lesson you have learnt about:
  - Concept of Database file
    - Data files
    - Control files
    - Online redo logs
  - Concept of SGA memory structure:
    - Database Buffer cache
    - Shared SQL Pool
    - Redo Log Buffer



Review Question

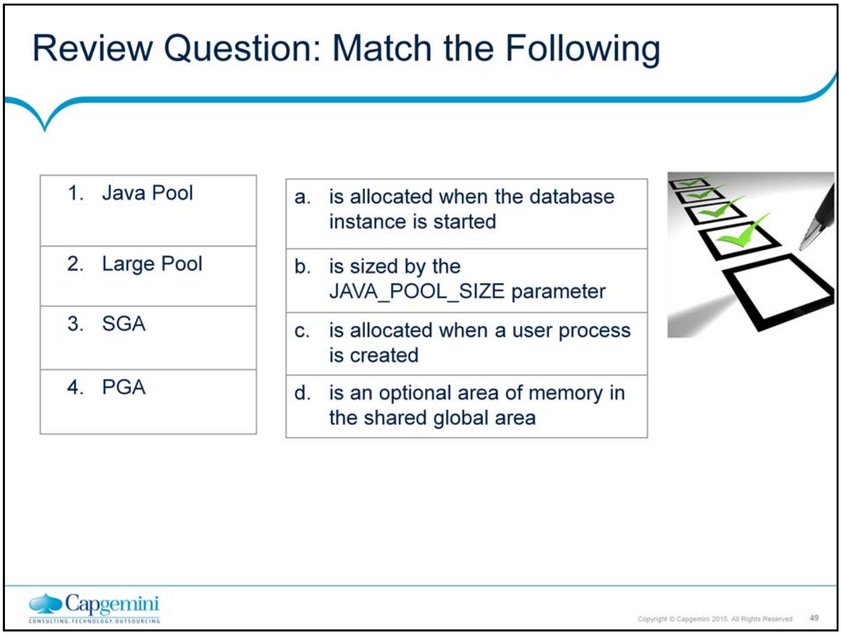
- Question 1: The “Oracle server” consists of an ___ and an ___.
- Question 2: ___ provide the actual physical storage for database information.
- Question 3: A single Oracle9i instance can open multiple databases.
  - True / False

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING



Review Question

- Question 4: The Background process communicates with the Oracle instance on behalf of the “user process”, which runs on the client.
  - True / False
- Question 5: Redo logs contain a record of changes made to the database.
  - True / False



Review Question: Match the Following

1. Java Pool
2. Large Pool
3. SGA
4. PGA

a. is allocated when the database instance is started

b. is sized by the JAVA_POOL_SIZE parameter

c. is allocated when a user process is created

d. is an optional area of memory in the shared global area