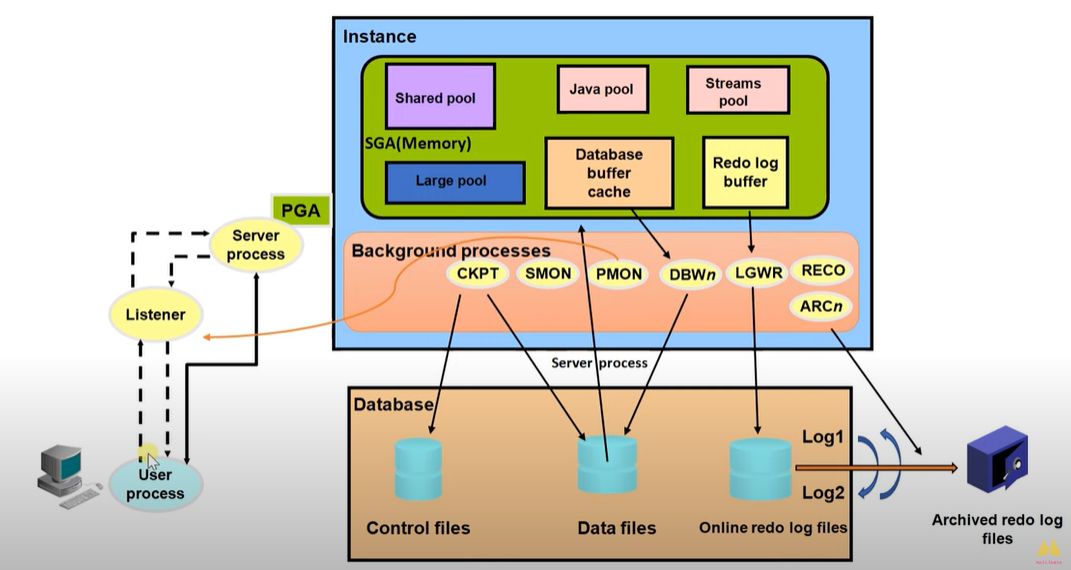
Oracle Database Admin Guide 23ai

# Oracle Database Architecture

## Oracle Database Architecture summary

**Oracle server:**

****

Include:

* + Oracle Instnace:
    - Memory:
      * SGA: System global area: shared memory for all database
      * PGA: Program global area: memory for each program
    - Backgroud process: SMON, PMON, DBWn, LGWR, CKPT
  + Oracle Database
    - Parameter file (spfile, pfile)
    - Controlfile
    - Datafile
    - Online redo log files, archivelog file
    - Password file

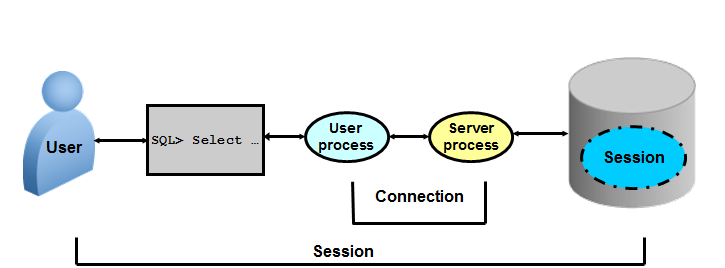
### Oracle Instance:

* + Là tập hợp của các tiến trình ngầm (background processes) và cấu trúc bộ nhớ (memory structure). Một Instance được khởi động để truy cập dữ liệu trong Oracle Database.
  + Định danh bởi: Oracle System Identifier **(SID)**
  + Mỗi khi một Instance được khởi động, một System Global Area (SGA) được cấp phát và các tiến trình ngầm của Oracle cũng sẽ được khởi động. Các tiến trình ngầm này sẽ thực hiện tác vụ vào/ra và quản lý các tiến trình khác của Oracle nhằm cung cấp khả năng chạy song song để thi hành tốt hơn và tin cậy hơn:
    - Pmon
    - Smon
    - DBWR
    - LogWR

### Oracle Database

* + Là một tập hợp các file hệ thống hay còn gọi là các file Database, cung cấp các thông tin về những thiết bị lưu trữ vật lý và thông tin và Database.
  + Định danh bởi: **DB\_NAME** và **DB\_ID**
  + Những file Database được sử dụng để đảm bảo rằng dữ liệu được lưu giữ ở trạng thái nhất quán, và có thể được khôi phục lại tại thời điểm một Instance bị lỗi:
    - Spfile: memory, vị trí control file, các setting
    - Control file: vị trí datafile, redo log thông tin backup, SCN
    - Datafile
    - Redolog file
    - Archivelog file

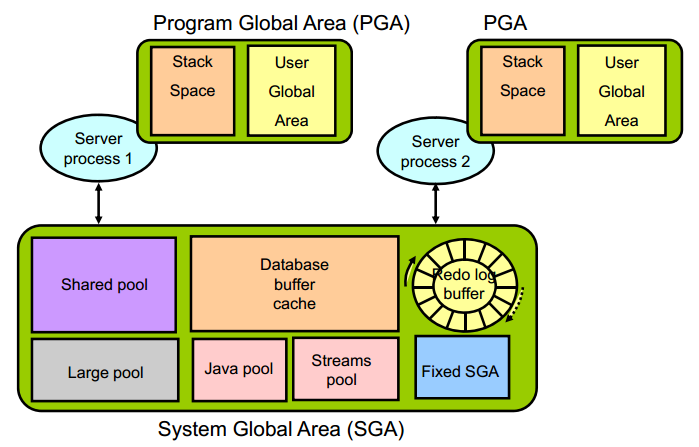
### Config and connect to Oracle database

* + Có 2 dạng cấu hình của database instance:
    - Single instance: 1 instance - 1 database
    - Rac: n instance (recommend 2) – 1 database
    - Multitenant: 1(n) instance – n database
  + Connecting to the Oracle Database
  + Connection: Communication between a user process and an instance
  + Session: Specific connection of a user to an instance through a user process
  + Khi user muốn thao tác với database thì phải tạo ra các yêu cầu (user process), các yêu cầu này được gửi tới server và server sẽ thực hiện các yêu cầu này để thao tác với db.

## Memory Architecture detail

Có hai cấu trúc bộ nhớ cơ bản trong một Instance:

* + System Global Area (SGA):
    - Là vùng bộ nhớ chia sẻ được sử dụng để lưu trữ dữ liệu và các thông tin điều khiển của Oracle server. Được chỉ định khi một Instance được khởi động, và là thành phần cơ bản của một Oralce Instance.
    - The SGA được chia sẻ dung chung bởi server và các tiến trình nền processes
  + Program Global Area (PGA):
    - Là một phần cấu trúc bộ nhớ lưu trữ dành cho 1 User process kết nối tới 1 Instance bao gồm dữ liệu và thông tin điều khiển cho một Server hoặc một Background process.
    - Được khởi tạo Được chỉ định khi một Server Process được khởi động.



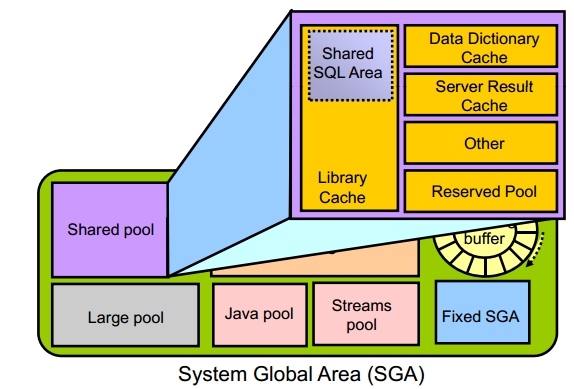
**AMM vs ASMM**

* + **AMM in Oracle 11g:** The 11g release uses AMM and manages all of the SGA AND PGA via the memory\_targetparameter.
  + **ASMM in Oracle10g:** Oracle ASMM was with Oracle 10g and uses two parameters sga\_max\_size for the SGA and pga\_aggregate\_targetfor the PGA.
* **Use ASMM better**

### Kiến trúc SGA

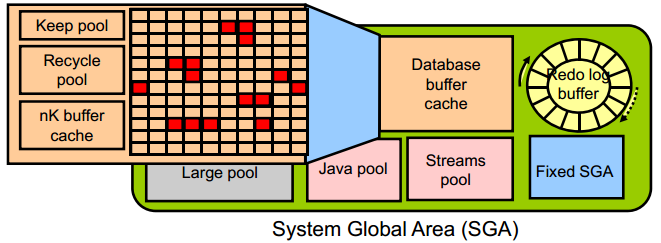
#### Shared pool

* + **Library cache**: Lưu trữ những định nghĩa về những đoạn lệnh SQL và PL/SQL vừa được thực thi gần đây nhất theo thuật giải Least Recently Used (LRU). Library cache bao gồm 2 cấu trúc là Shared SQL area và Shared PL/SQL area. Kích thước của vùng này được xác định bởi Shared pool sizing.
  + **Data dictionary cache**: Thu thập những định nghĩa được dùng gần đây nhất trên cơ sỡ dữ liệu bao gồm các thông tin về Database file, tables, indexes, columns, user, privileges. Trong quá trình phân tích cú pháp đoạn lệnh, Server Process sẽ đọc các thông tin định nghĩa ở Data dictionary cache để lấy tên các đối tượng, xác nhận truy cập. Kích thước của vùng này được xác định bởi Shared pool sizing.
  + **Shared SQL area:** Oracle Database represents each SQL statement that it runs with a shared SQL area (as well as a private SQL area kept in the PGA). When a new SQL statement is parsed, Oracle Database allocates memory from the shared pool to store in the shared SQL area. The size of this memory depends on the complexity of the statement
  + **The server result cache** contains the SQL query result cache and PL/SQL function result cache, which share the same infrastructure. The server result cache contains result sets, not data blocks
  + **The reserved pool** is a memory area in the shared pool that Oracle Database can use to allocate large contiguous chunks of memory



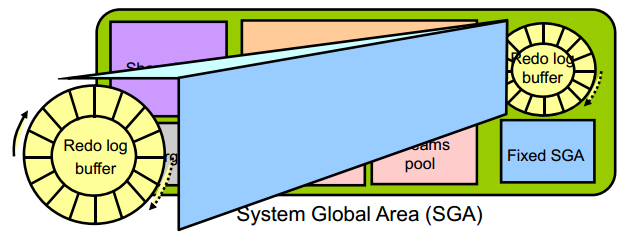
#### ****Database Buffer Cache****

* + Lưu trữ những bản copy của **Block** dữ liệu đã được đọc từ **Data File.** Khi một đoạn SQL được thực thi, thì **Server Process** sẽ đọc các thông tin từ **Database buffer cache** để lấy các **block** dữ liệu cần thiết, điều này giúp cho tốc độ hoạt động của hệ thống sẽ cao hơn vì đọc trên cache sẽ nhanh hơn là đọc trên đĩa cứng . Nếu các block dữ liệu không có trong Database buffer cache thì Server process mới đọc dữ liệu từ data file. Database buffer cache cũng sử dụng thuật giải LRU như ở Shared Pool.
  + The **keep buffer pool** and the **recycle buffer pool** are used for specialized buffer pool tuning. The keep buffer pool is designed to retain buffers in memory longer than the LRU would normally retain them. The recycle buffer pool is designed to flush buffers from memory faster than the LRU normally would.
  + **nK buffer caches,** additional buffer can be configured to hold blocks of a size that is different from the default block size.



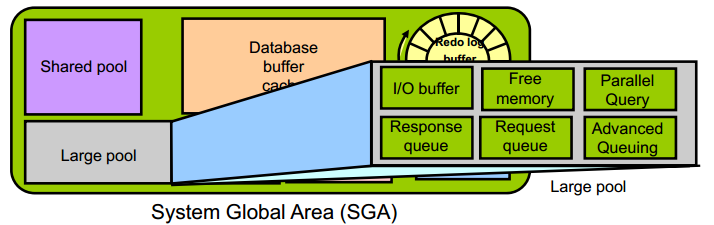
#### Redo log buffer

* + Redo entries contain the information necessary to reconstruct (or redo) changes that are made to the database by DML, DDL, or internal operations. Redo entries are used for database recovery if necessary



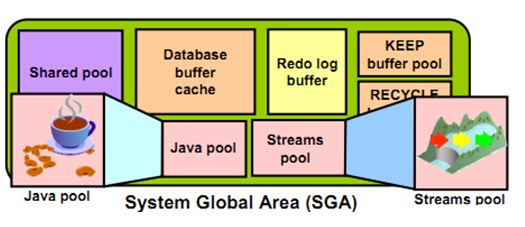
#### ****Large Pool****

* + The database administrator can configure an optional memory area called the large pool to provide large memory allocations for:
    - Session memory for the shared server and the Oracle XA interface (used where transactions interact with multiple databases)
    - I/O server processes
    - Oracle Database backup and restore operations
    - Parallel Query operations
    - Advanced Queuing memory table storage

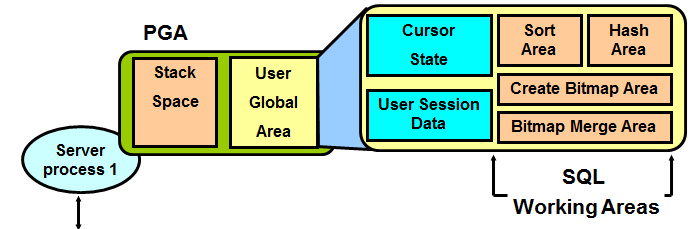


#### Java Pool và Streams Pool

* + Java Pool là vùng dùng cho các Procedure viết bằng Java.
  + Stream Pool dùng cho Oracle Stream.



### Kiến trúc PGA



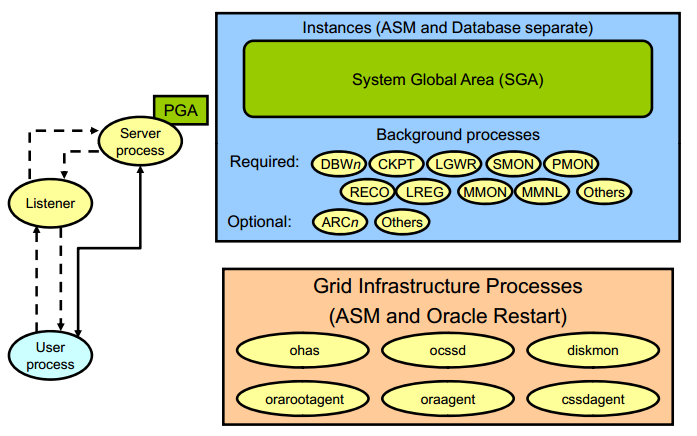
**Program Global Area (PGA)** là vùng nhớ riêng bao gồm dữ liệu và thông tin điều khiển của một Server Process. Mỗi Server Process đều có một PGA riêng biệt. PGA bao gồm:

* + Stack space: lưu trữ các biến và các mảng được xử lý trên PGA
  + User Global Area (UGA):
    - Cursor State: lưu trữ thông tin con trỏ
    - User session data: Lưu trữ thông tin điều khiển của 1 session
    - SQL working Areas: thực thi các câu lệnh truy vấn SQL:
      * Hash area: kết nối giữa các bảng bằng hash
      * Create bitmap area: sử dụng để tạo index bitmap
      * Bitmap merge area: sử dụng để tạo các kế hoạch cho index bitmap
      * Sort area: sử dụng cho các hàm gom nhóm dữ liệu như: ORDER BY and GROUP BY

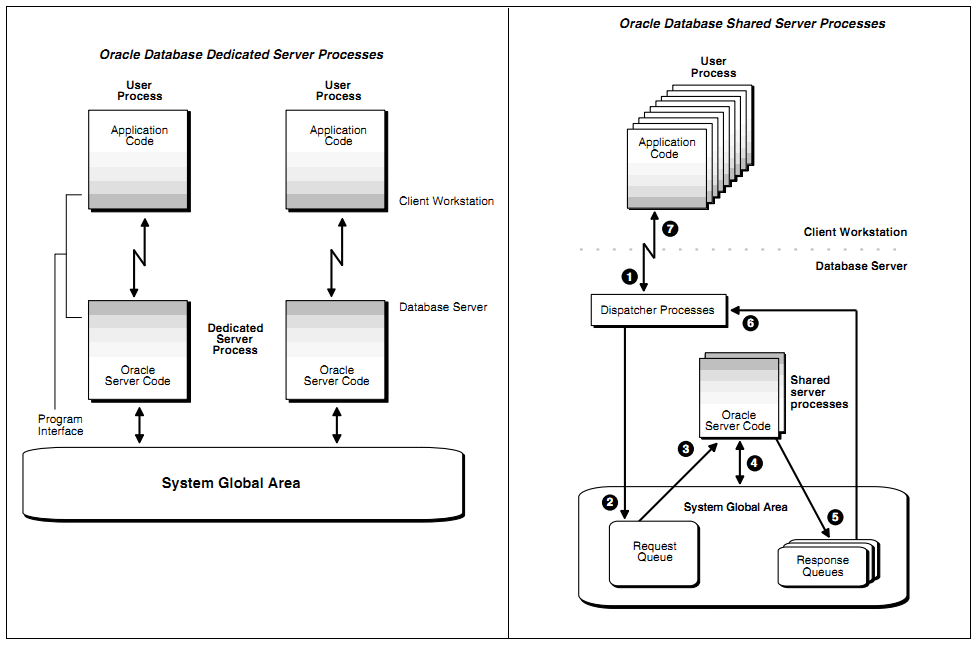
## Processes architecture detail

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

* + User processes that run the application or Oracle tool code
  + Oracle Database processes that run the Oracle Database server code (including server processes and background processes)
    - Server processes created on behalf of each user’s application can perform one or more of the following:
      * Parse and run SQL statements issued through the application.
      * Read necessary data blocks from data files on disk into the shared database buffers of the SGA (if the blocks are not already present in the SGA).
      * Return results in such a way that the application can process the information
    - To maximize performance and accommodate many users, a multiprocess Oracle Database system uses some additional Oracle Database processes called background processes. An Oracle Database instance can have many background processes.
  + Oracle daemons and application processes not specific to a single database
    - Networking listeners
    - Grid Infrastructure daemons



### Oracle Database process mode



#### Dedicated Server Processes mode

Khái niệm:

* Dedicated server process: một server process chỉ phục vụ một user process

Sử dụng khi:

* Thực thi các job lớn (các job rất ít, hoặc không có thời gian chờ đợi...)
* Sử dụng (RMAN) để back up, restore, hoặc recover database

#### Shared Server Processes

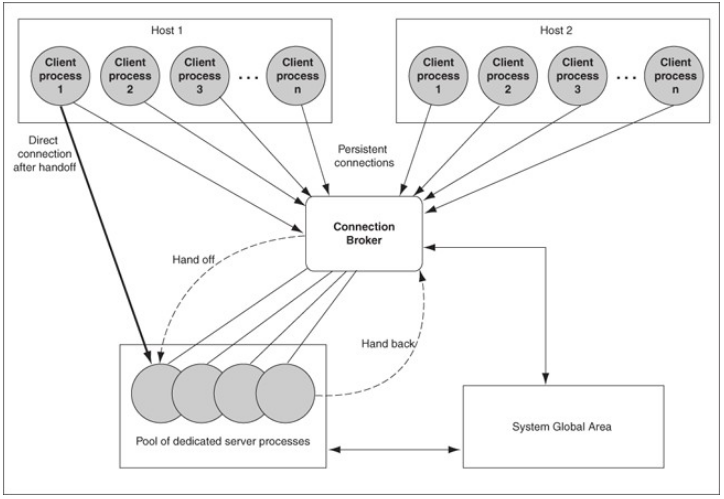
Khái niệm:

* Shared server process: một server process có thê phục vụ nhiều user process
* Sử dụng 1 bộ điều phối để quản lý tài nguyên (**DISPATCHERS**)

Sử dụng khi:

* Có rất nhiều user process kết nối tới DB và các server này tương tự nhau và có thể chờ đợi được (đăng kí vé ....)

#### Database Resident Connection Pooling



Database Resident Connection Pooling (DRCP) provides a connection pool in the database server for typical Web application usage scenarios where the application acquires a database connection, works on it for a relatively short duration, and then releases it. DRCP pools "dedicated" servers. A pooled server is the equivalent of a server foreground process and a database session combined.

DRCP complements middle-tier connection pools that share connections between threads in a middle-tier process. In addition, DRCP enables sharing of database connections across middle-tier processes on the same middle-tier host and even across middle-tier hosts. This results in significant reduction in key database resources needed to support a large number of client connections, thereby reducing the database tier memory footprint and boosting the scalability of both the middle-tier and the database tier. Having a pool of readily available servers also has the additional benefit of reducing the cost of creating and tearing down client connections.

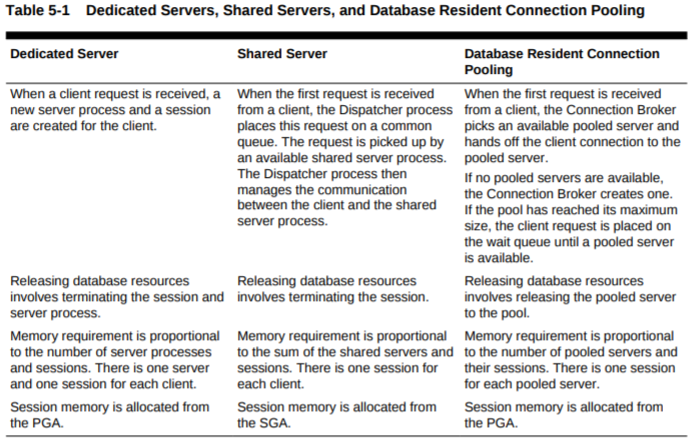
DRCP is especially relevant for architectures with multi-process single threaded application servers (such as PHP/Apache) that cannot perform middle-tier connection pooling. The database can still scale to tens of thousands of simultaneous connections with DRCP.

**When To Use Database Resident Connection Pooling:**

* A large number of client connections need to be supported with minimum memory usage.
* The client applications are similar and can share or reuse sessions.
* Applications are similar if they connect with the same database credentials and use the same schema.
* The client applications acquire a database connection, work on it for a relatively short duration, and then release it.
* Session affinity is not required across client requests.
* There are multiple processes and multiple hosts on the client side.

**Advantages of Database Resident Connection Pooling**

* Enables resource sharing among multiple middle-tier client applications.
* Improves scalability of databases and applications by reducing resource usage.

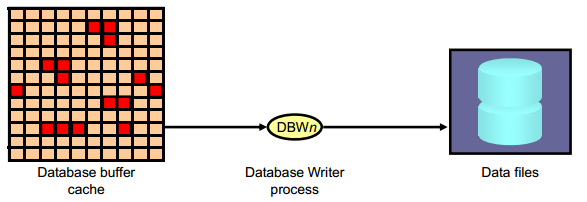


### Background process

Background process (các tiến trình nền) thực hiện các chức năng thay cho lời gọi tiến trình xử lý tương ứng. Nó điều khiển vào ra, cung cấp các cơ chế xử lý song song nâng cao hiệu quả và độ tin cậy. Tùy theo từng cấu hình mà Oracle instance có các Background process như:

* + Database Writer process (DBWn)
  + Log Writer process (LGWR)
  + Checkpoint process (CKPT)
  + System monitor process (SMON)
  + Process monitor process (PMON)
  + Recoverer process (RECO)
  + Listener registration process (LREG)
  + Manageability monitor process (MMON)
  + Manageability monitor lite process (MMNL)
  + Job queue coordinator (CJQ0)
  + Job slave processes (Jnnn)
  + Archiver processes (ARCn)
  + Queue monitor processes (QMNn)

#### Database Writer (DBWn)



The Database Writer process (DBWn) writes the contents of buffers to data files.

* + The DBWn processes are responsible for writing modified (dirty) buffers in the database buffer cache to disk.
  + The *DB\_WRITER\_PROCESSES* initialization parameter specifies the number of DBWn processes. The maximum number of Database Writer processes is 100. If it is not specified by the user during startup, Oracle Database determines how to set *DB\_WRITER\_PROCESSES* based on the number of CPUs and processor groups.

The DBWn process writes dirty buffers to disk under the following conditions:

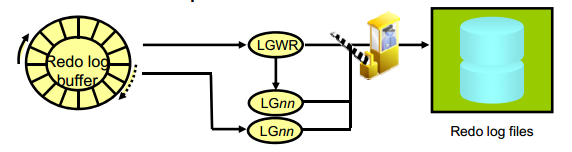
* + When a server process cannot find a clean reusable buffer after scanning a threshold

number of buffers, it signals DBWn to write. DBWn writes dirty buffers to disk asynchronously while performing other processing.

* + DBWn writesbuffers to advance the checkpoint, which is the position in the redo thread (log) from which instance recovery begins. This log position is determined by the oldest dirty buffer in the buffer cache.

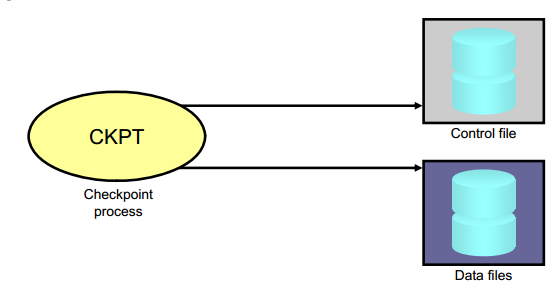
In all cases, DBWn performs batched (multiblock) writes to improve efficiency. The number of blocks written in a multiblock write varies by operating system

#### Log Writer Process (LGWR)



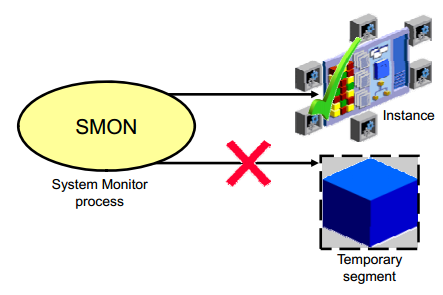
* + The Log Writer process (LGWR) is responsible for redo log buffer management by writing the redo log buffer entries to a redo log file on disk. LGWR writes all redo entries that have been copied into the buffer since the last time it wrote.
  + LGWR starts and coordinates multiple helper processes that concurrently perform some of the work. LGWR handles the operations that are very fast, or must be coordinated, and delegates operations to the LGnn that could benefit from concurrent operations, primarily writing the redo from the log buffer to the redo log file and posting the completed write to the foreground process that is waiting.
  + Writes the redo log buffer to a redo log file on disk
    - When a user process commits a transaction
    - When an online redo log switch occurs
    - When the redo log buffer is one-third full or contains 1 MB of buffered data
    - Before a DBWn process writes modified buffers to disk
    - When three seconds have passed since the last write
  + Serves as coordinator of LGnn processes and ensures correct order for operations that must be ordered

#### Checkpoint Process (CKPT)



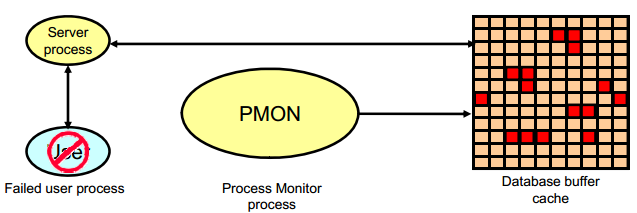
* + A checkpoint is a data structure that defines a system change number (SCN) in the redo thread of a database. Checkpoints are recorded in the control file and in each data file header. They are a crucial element of recovery.
  + When a checkpoint occurs, Oracle Database must update the headers of all data files to record the details of the checkpoint. This is done by the CKPT process. And Signals DBWn to write blocks to disk

#### System Monitor Process (SMON)



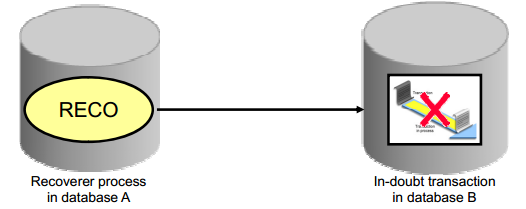
* + The System Monitor process (SMON) performs recovery at instance startup if necessary. SMON is also responsible for cleaning up temporary segments that are no longer in use. If any terminated transactions were skipped during instance recovery because of file-read or offline errors, SMON recovers them when the tablespace or file is brought back online

#### Process Monitor Process (PMON)



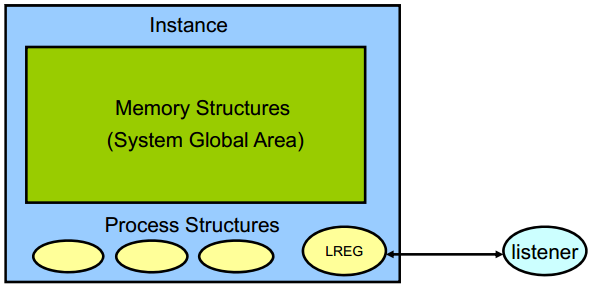
* + The Process Monitor process (PMON) performs process recovery when a user process fails. PMON is responsible for cleaning up the database buffer cache and freeing resources that the user process was using
  + PMON periodically checks the status of dispatcher and server processes, and restarts any that have stopped running (but not any that Oracle Database has terminated intentionally).

#### Recoverer Process (RECO)



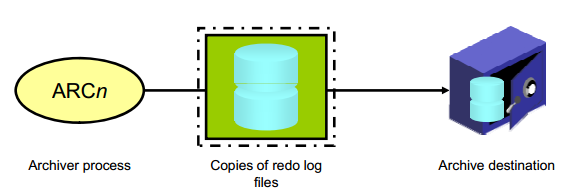
* + The Recoverer process (RECO) is a background process that is used with the distribute database configuration that automatically resolves failures involving distributed transactions
  + Automatically connects to other databases involved in indoubt distributed transactions
  + Automatically resolves all in-doubt transactions
  + Removes any rows that correspond to in-doubt transactions

#### Listener Registration Process (LREG)



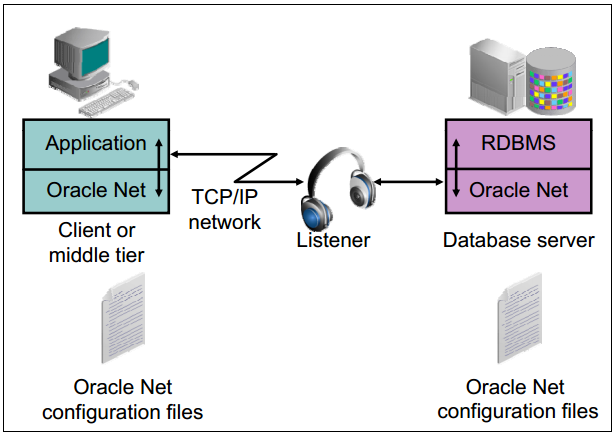
* + The Listener Registration process, LREG, registers information about the database instance and dispatcher processes with the Oracle Net Listener. LREG provides the listener with the following information:
    - Names of the database services
    - Name of the database instance associated with the services and its current and maximum load
    - Service handlers available for the instance, including their type, protocol addresses, and current and maximum load

#### Archiver Processes (ARC*n*)



* + The Archiver processes (ARCn) copy redo log files to a designated storage device after a log switch has occurred. ARCn processes are present only when the database is in ARCHIVELOG mode and automatic archiving is enabled.
  + Can collect transaction redo data and transmit that data to standby destinations

### Oracle Networking process



Oracle Net Services enables network connections from a client or middle-tier application to the Oracle server. After a network session is established, Oracle Net acts as the data courier for both the client application and the database server. It is responsible for establishing and maintaining the connection between the client application and database server, as well as exchanging messages between them. Oracle Net (or something that simulates Oracle Net, such as Java Database Connectivity) is located on each computer that needs to talk to the database server.

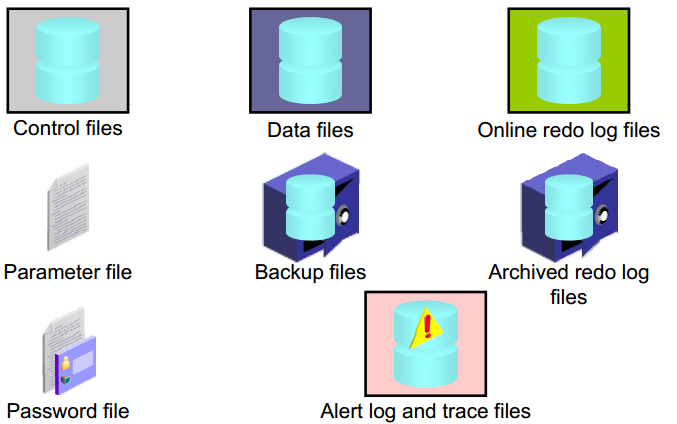
On the client computer, Oracle Net is a background component for application connections to the database.

On the database server, Oracle Net includes an active process called Oracle Net Listener, which is responsible for coordinating connections between the database and external applications.

To make a client or middle-tier connection, Oracle Net requires the client to know the:

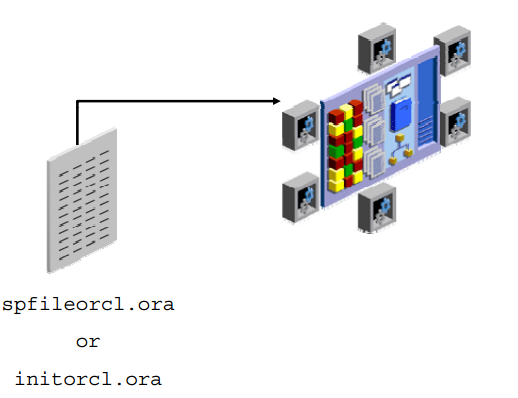
* Host where the listener is running
* Port that the listener is monitoring
* Protocol that the listener is using
* Name of the service that the listener is handling

## Oracle Database Storage physical detail



### Parameter file

* + File chứa những thông tin cấu hình của instance và địa chỉ của controlfile.
    - Ví dụ như: Các vùng nhớ khởi tạo được cấp phát bao nhiêu GB, các control\_file nằm ở đâu, bao nhiêu process, session v.v…
  + Parameter có 2 loại là SPFile (file Binary) và Pfile (file clear text), nhiệm vụ của 2 file này là giống nhau. Mặc định, lúc khởi động DB, Oracle sẽ ưu tiên sử dụng spfile trước, nếu không có thì sẽ chuyển sang dùng pfile (pfile có thể lưu trữ địa chỉ của spfile)



When you start the instance, an initialization parameter file is read. There are two types of parameter files.

* Server parameter file (SPFILE): This is the preferred type of initialization parameter file. It is a binary file that can be written to and read by the database server and must not be edited manually. It resides on the server on which the Oracle instance is executing; it is persistent across shutdown and startup. The default name of this file, which is automatically sought at startup, is spfile<SID>.ora.
* Text initialization parameter file: This type of initialization parameter file can be read by the database server, but it is not written to by the server. The initialization parameter settings must be set and changed manually by using a text editor so that they are persistent across shutdown and startup. The default name of this file (which is automatically sought at startup if an SPFILE is not found) is init<SID>.ora.

In the majority of cases, it is necessary to set and tune only the 30 or so basic parameters to get reasonable performance from the database. In rare situations, modification of the advanced parameters may be needed to achieve optimal performance. There are more than 300 advanced parameters.

* **CONTROL\_FILES** **parameter**: Specifies one or more control file names. Oracle strongly recommends that you multiplex and mirror control files. Range of values: from one to eight file names (with path names). Default value: OS dependent.
* **DB\_FILES** **parameter**: Specifies the maximum number of database files that can be opened for this database. Range of values: OS dependent. Default value: 200.
* **PROCESSES parameter**: Specifies the maximum number of OS user processes that can simultaneously connect to an Oracle server. This value should allow for all background processes and user processes. Range of values: from 6 to an OS-dependent value. Default value: Dynamic and dependent on the number of CPUs.
* **DB BLOCK SIZE \_BLOCK\_SIZE parameter**: Specifies the size (in bytes) of an Oracle database block. This value is set at database creation and cannot be subsequently changed. This specifies the standard block size for the database. All tablespaces will use this size by default. Range of values: 2048 to 32768 (OS-dependent). Default value: 8192.
* **DB\_CACHE\_SIZE parameter**: Specifies the size of the default buffer pool. Range of values: At least 4 MB times the number of CPUs (smaller values are automatically rounded up to this value). Default value: 0 if SGA\_TARGET is set, otherwise the larger of 48 MB or (4 MB\*CPU\_COUNT)
* **PGA\_AGGREGATE\_TARGET parameter**: Specifies the amount of Program Global Area (PGA) memory available to all server processes attached to the instance. This memory does not reside in the System Global Area (SGA). The database uses this parameter as a target amount of PGA memory to use. When setting this parameter, subtract the SGA from the total memory on the system that is available to the Oracle instance. The minimum value is 10 MB and the maximum value is (4096 GB – 1). The default is 10 MB or 20% of the size of the SGA, whichever is greater.
* **SHARED\_POOL\_SIZE parameter**: Specifies the size of the shared pool in bytes. The shared pool contains objects such as shared cursors, stored procedures, control structures, and parallel execution message buffers. Range of values: OS-dependent. Default value: 0 if SGA\_TARGET is set, otherwise 128 MB if 64-bit; 48 MB if 32-bit.
* **UNDO\_MANAGEMENT parameter**: Specifies the undo space management mode that the system should use. When set to AUTO, the instance is started in automatic undo management mode. Otherwise, it is started in rollback undo mode. In rollback undo mode, undo space is allocated as rollback segments. In automatic undo mode, undo space is allocated as undo tablespaces. Range of values: AUTO or MANUAL. If the UNDO MANAGEMENT parameter is omitted when the instance is started the default value AUTO is used.

### Control Files

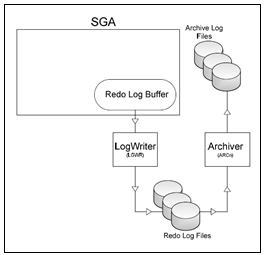
* + Mỗi Oracle database đều có ít nhất một control file. Control file chứa các mục thông tin quy định cấu trúc vật lý của database như:
    - Tên của database.
    - Tên và nơi lưu trữ các datafiles, redo log files.
    - Các thông tin về backup của database
  + Control file cũng được sử dụng đến khi thực hiện khôi phục lại dữ liệu.
  + The control file is critical to the database. Without the control file, the database cannot be opened

### Datafiles

* + Mỗi một Oracle database đều có thể có một hay nhiều datafiles. Các database datafiles chứa toàn bộ dữ liệu trong database. Các dữ liệu thuộc cấu trúc logic của database như tables hay indexes đều được lưu trữ dưới dạng vật lý trong các datafiles của database.
  + Một số tính chất của datafiles:
    - Mỗi datafile chỉ có thể được sử dụng trong một database.
    - Datafiles cũng còn có một số tính chất cho phép tự động mở rộng kích thước mỗi khi database hết chỗ lưu trữ dữ liệu.
    - Một hay nhiều datafiles tạo nên một đơn vị lưu trữ logic của database gọi là tablespace. Một datafile chỉ thuộc về một tablespace.

### Redo Log Files - Archivelog

* + REDO LOG file là một phần quan trọng của quá trình Oracle recovery. Được sử dụng để recover database khi bị crash.
  + Chức năng chính của redo log là ghi lại tất cả các thay đổi đối với dữ liệu trong database. Redo log files được sử dụng để bảo vệ database khỏi những sự cố. Oracle cho phép sử dụng cùng một lúc nhiều redo log gọi là multiplexed redo log để cùng lưu trữ các bản sao của redo log trên các ổ đĩa khác nhau
  + Khi mỗi Oracle block thay đổi, Oracle tạo ra các vector thay đổi cho nó. Mỗi vetor này được cọi là REDO entry hoặc REDO records. Những sự thay đổi này sẽ được server process ghi vào redo log buffer trên PGA. Redo log buffer ghi xuống online redo log vào 1 thời điểm nào đó bởi LGWR
  + Redolog thường gồm ít nhất 2 đơn vị, ghi lại những thao tác tác động lên DB, khi 1 redolog đầy sẽ switch sang redo log bên cạnh. Khi switch sang 1 redolog đã có dữ liệu, dữ liệu này sẽ được ghi xuống archivelog để lưu trữ

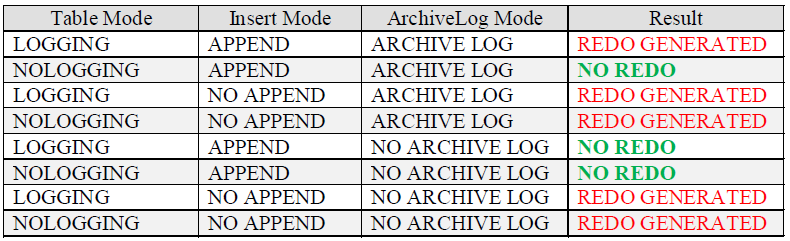


**Logging và nologging**

Tuy việc tạo nên REDO LOG file là một phần quan trọng của quá trình Oracle recovery nhưng vẫn có khả năng có những redo log file được tạo nên từ những thao tác không mong muốn. Điều quan trọng nhất với dữ liệu là không được phép đặt nó trong trình trạng không thể recovery. Nhưng điều này đồng nghĩa với việc không thể sử dụng các option để rút ngắn thời gian hoặc gia tăng hiệu năng của hệ thống

Oracle cung cấp cho người sử dụng khả năng giới hạn ghi log vào bảng và indexes bằng NOLOGGING mode.

* + NOLOGGING có tác động mạnh đến việc recovery.
  + Đặc điểm chính cần chú ý:
    - NOLOGGING được thiết kế để insert 1 lượng dữ liệu lớn(bulk data)
    - Nếu dữ liệu không được ghi log thì nó không thể được recover, dữ liệu nên được backup sau khi chỉnh sửa.
  + Các trường hợp:



Trong một số trường hợp khi ở chế độ NOLOGGING những câu lệnh vẫn ghi log:

* + CREATE TABLE ... AS SELECT
  + CREATE INDEX.
  + UPDATE/INSERRT/DELETE

**Archived redo log files**:

* + Contain an ongoing history of the data changes (redo) that are generated by the instance. Using these files and a backup of the database, you can recover a lost data file. That is, archive logs enable the recovery of restored data files.

**Backup files**:

* + Are use for database recovery. You typically restore a backup file when a media failure or user error has damaged or deleted the original file

### Password file:

Allows users using the SYSDBA, SYSOPER, SYSBACKUP, SYSDG, SYSKM, and SYSASM roles to connect remotely to the instance and perform administrative tasks

### Log file – Trace file

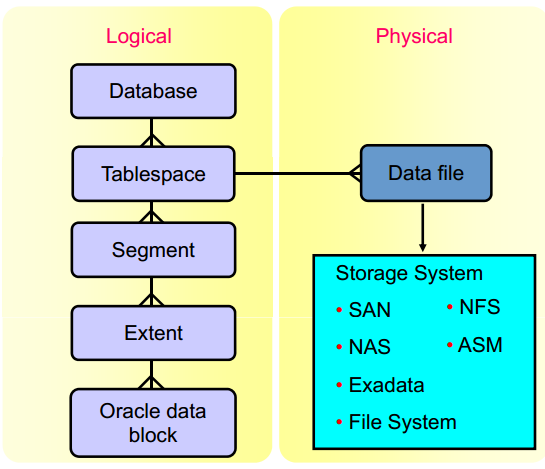
**Trace files**:

* + Each server and backg p round process can write to an associated trace file. When an internal error is detected by a process, the process dumps information about the error to its trace file. Some of the information written to a trace file is intended for the database administrator, whereas other information is for Oracle Support Services.

**Alert log file**:

* + These are special trace entries. The alert log of a database is a chronological log of messages and errors. Oracle recommends that you review the alert log periodically.

## Oracle Database Storage Logical detail



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

### Tablespaces

Một database có thể được phân chia về mặt logic thành các đơn vị gọi là các tablespaces. Tablespaces thường bao gồm một nhóm các thành phần (logic) có quan hệ logic với nhau. Mỗi tablespace có thể được tạo nên, bởi một hoặc nhiều datafiles (physic). Kích thước của database cũng có thể xác định được bằng tổng kích thước của các tablespaces của nó.

**Phân loại:**

* + Tablespace SYSTEM
    - Sẽ tự động được tạo khi database tạo.
    - Có trong tất cả các database dùng cho hoạt động của database
    - Chứa thông tin về các data dictionary view, các định nghĩa của store procedure, package và các database trigger.
      * Chứa SYSTEM **Undo segment**.
      * Chứa SYSTEM **rollback segment**
  + Non – System Tablespace
    - Có thể lưu trữ rollback segment, temporary segment, data segment, index segment
    - Giúp cho quản trị database linh hoạt hơn.
  + Temporary tablespace:
    - Tempory Tablespace được sử dụng để chứa các tempory table trong thời gian của các transaction.
    - Temporary tablespaces sử dụng lưu trữ: Intermediate sort results: nếu dữ liệu truy vấn trong PGA quá nhiều thì nó có thể được lưu trữ tại đây và lấy dần ra, Temporary tables and temporary indexes, Temporary LOBs, Temporary B-trees
    - Mặc định, chỉ có Temp tablespace duy nhất được tạo ra khi cài đặt một Oracle mới. Nhưng có thể tạo thêm tablespace bổ sung

### 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. Example:

* Data segments: Each nonclustered, non-index-organized table has a data segment, with the exception of external tables, global temporary tables, and partitioned tables (in which each table has one or more segments). All of the table’s data is stored in the extents of its data segment. For a partitioned table, each partition has a data segment. Each cluster has a data segment. The data of every table in the cluster is stored in the cluster’s data segment.
* Index segments: Each index has an index segment that stores all of its data. For a partitioned index, each partition has an index segment.
* Undo segments: One UNDO tablespace is created for each database instance This tablespace contains numerous undo segments to temporarily store undo information. The information in an undo segment is used to generate read-consistent database information and, during database recovery, to roll back uncommitted transactions for users.
* Temporary segments: Temporary segments are created by the Oracle database when a SQL statement needs a temporary work area to complete execution. When the statement finishes execution, the temporary segment’s extents are returned to the database for future use. Specify either a default temporary tablespace for every user, or a default temporary tablespace that is used database-wide

#### Table

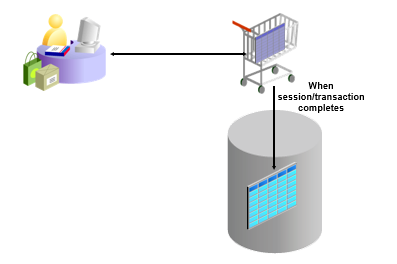
Table là đơn vị lưu trữ dữ liệu của database, nó có thể chứa dữ liệu vĩnh viễn hoặc dữ liệu tạo (Temporary table)

Table là một segment, hoặc là tập hợp của nhiều segment ( nếu table đấy được partition)

#### Temporary Table

Temporary Table là một bảng chứa dữ liệu mà chỉ tồn tại trong thời gian của một giao dịch hoặc phiên. Dữ liệu trong một bảng tạm thời là trong phiên giao dịch, có nghĩa là mỗi phiên chỉ có thể xem và sửa đổi dữ liệu riêng của mình.

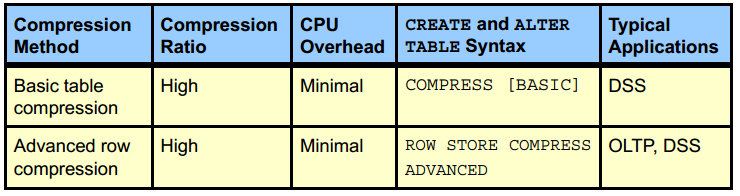
* + Temporary table rất hữu ích trong các ứng dụng một tập hợp kết quả được đếm.
  + Ví dụ: một giỏ mua hàng trong một ứng dụng trực tuyến có thể là một bảng tạm thời. Mỗi mục được đại diện bởi một hàng trong bảng tạm thời. Bởi vì bảng tạm thời được định nghĩa cố định, nên có thể tạo index cho Chỉ số tạo ra trên bảng tạm thời cũng chỉ là tạm thời. Các dữ liệu trong các chỉ số có cùng một phiên, phạm vi giao dịch như các dữ liệu trong bảng tạm thời.



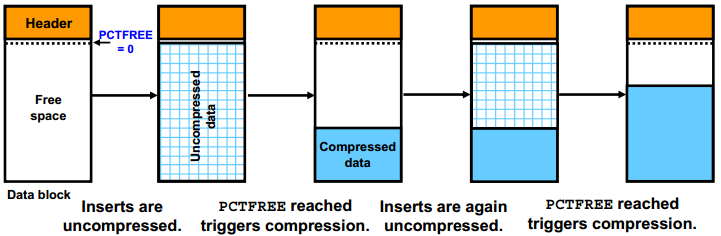
#### Table Compression

Reducing storage costs by compressing all data:

* + Basic compression for direct-path insert operations: 10x
  + Advanced row compression for all DML operations: 2–4x



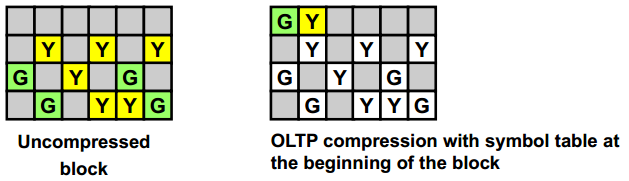
**Basic table compression**



With *COMPRESS* or *COMPRESS BASIC*, you enable basic table compression.

* + The Oracle Database server attempts to compress data during the following direct-pathinsert operations when it is productive to do so:
    - Direct-path SQL\*Loader
    - CREATE TABLE AS SELECT statements
    - Parallel INSERT statements
    - INSERT statements with an APPEND hint
  + **How**: Compression eliminates holes created due to deletions and maximizes contiguous free space in blocks.

**Advanced Row Compression**



With *ROW STORE COMPRESS ADVANCED*, you enable advanced row compression.

* + The Oracle database compresses data during all DML operations on the table. This form of compression is recommended for active OLTP environments.
  + In earlier releases, OLTP table compression was enabled with COMPRESS FOR ALL OPERATIONS and COMPRESS FOR OLTP. This syntax has been deprecated.

**How**: With advanced row compression, duplicate values in the rows and columns in a data block are stored once at the beginning of the block in a symbol table. Duplicate values are replaced with a short reference to the symbol table (as shown in the slide). Thus, information needed to re-create the uncompressed data is stored in the block.

**Specifying Table Compression**

You can specify table compression:

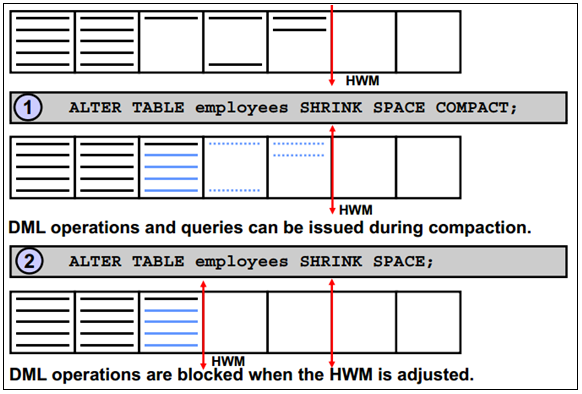
* + For an entire heap-organized table (in the *physical\_properties* clause of *relational\_table* or *object table*)
  + For partitioned tables (Each partition can have a different type or level of compression.)
  + For the storage of a nested table (in the nested\_table\_col\_properties clause)

Table compression has the following restrictions:

* + *ROW STORE COMPRESS ADVANCED* and *COMPRESS BASIC* are not supported for tables with more than 255 columns.
  + You cannot drop a column from a table that is compressed for direct-load operations, although you can set such a column as unused.

**Shrinking Segments**

\*\*\*One way to Improved performance and space utilization



The diagram in the slide describes the two phases of a table shrink operation.

* + Compaction is performed in the first phase. During this phase, rows are moved to the left part of the segment as much as possible. Internally, rows are moved by packets to avoid locking issues.
  + After the rows have been moved, the second phase of the shrink operation is started. During this phase, the high-water mark (HWM) is adjusted and the unused space is released.

The *COMPACT* clause is useful if you have long-running queries that might span the shrink operation and attempt to read from blocks that have been reclaimed. When you specify the *SHRINK SPACE COMPACT* clause, the progress of the shrink operation is saved in the bitmap blocks of the corresponding segment. This means that the next time a shrink operation is executed on the same segment, the Oracle Database server remembers what has been done already. You can then reissue the *SHRINK SPACE* clause without the *COMPACT* clause during off-peak hours to complete the second phase.

### Blocks and extends

**Data Blocks**

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

**Extents**

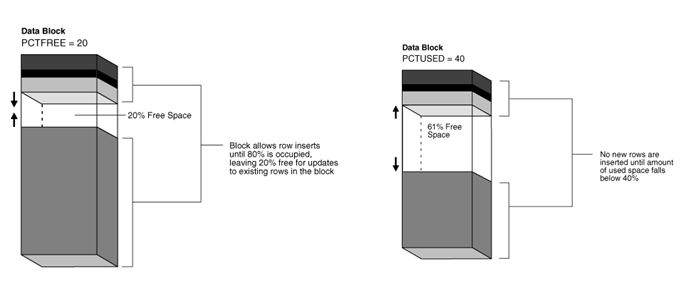
* The next level of logical database space is an extent. An extent is a specific number of contiguous Oracle data blocks (obtained in a single allocation) that are used to store a specific type of information. Oracle data blocks in an extent are logically contiguous but can be physically spread out on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

#### Block Pct used, pct free

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

Thực chất dữ liệu được lưu trữ trong các block

* Freespace: Không gian cấp phát cho việc insert/update trong tương lai, ảnh hưởng bởi giá trị của 2 tham số là PCTUSED và PCTFREE
* Data Row chứa dữ liệu
* Khi create/alter bất kỳ table/index nào, Oracle sẽ sử dụng 2 tham số đề điều khiển không gian
  + PCTFREE: Số % dành riêng cho việc update các dữ liệu đã có trong tương lai
  + PCTUSED: Số % của không gian nhỏ nhất đã được sử dụng cho việc insert data mới, giá trị này xác định khi nào thì các blocks sẽ được đưa trở lại vào trong FREELIST
  + FREELIST: Cấu trúc xác định mà Oracle sử dụng để maitains 1 danh sách các block free hiện có.

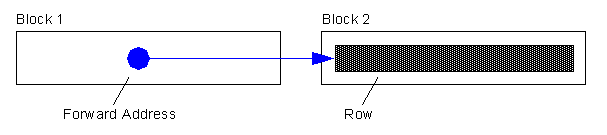


* Tham số PCTFREE xác định số % nhỏ nhất (không gian) của 1 data block được dành riêng cho việc update những row đã có trong block đó.
* Ví dụ:
  + Ta xác định 20% là giá trị của PCTFREE trong câu lệnh CREATE TABLE, thì có nghĩa, 20% của từng data block trong table segment sẽ được dành riêng cho việc update các row đã có bên trong từng block.
  + Block chỉ được phép insert dữ liệu khi dữ liệu khi % block giảm xuống nhỏ hơn PCTUSED

#### Row Migration

Trong trường hợp: 1 row đã được lưu overhead vào trong 1 data block, tuy nhiên, khi insert dữ liệu vào thì free space đã đầy. Vì thế, Oracle sẽ (dịch chuyển) data của toàn bộ row này sang 1 data block mới, với data block cũ, Oracle sẽ giữ lại 1 row piece – cũng chính là rowID để trỏ tới block mới.

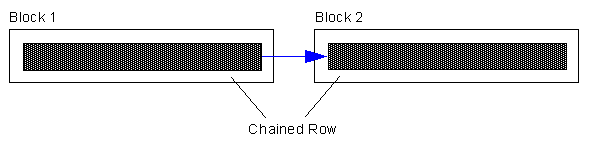
Đối với trường hợp Row Migration, việc Full Table Scan không bị ảnh hưởng (I/O increase), là bởi vì Forward address sẽ bị bỏ qua, tuy nhiên, Row Migration lại làm ảnh hưởng đến việc đọc Index. Bởi vì, Index sẽ nói rằng: “Đến file X, block Y, slot Z…để tìm row này”, và vì thế, khi nhận được thông điệp trên, ta lại mất thêm 1 I/O physical hoặc logical để tìm row này. (Physical I/O, Logical I/O đề cập ở phần khác).



#### Row Chaining

Trong nhiều trường hợp, data cho 1 row quá lớn để lưu trong 1 single data block, do vậy, Oracle sẽ lưu data của row này vào trong 1 hoặc nhiều chained data block (các data block móc nối). Lấy ví dụ, nếu ta sử dụng 1 data block size 4KB cho Database, và cần insert 1 row với size 8KB, Oracle sẽ sử dụng 3 blocks để lưu lại data trong row. Row chaining xảy ra với các trường hợp:

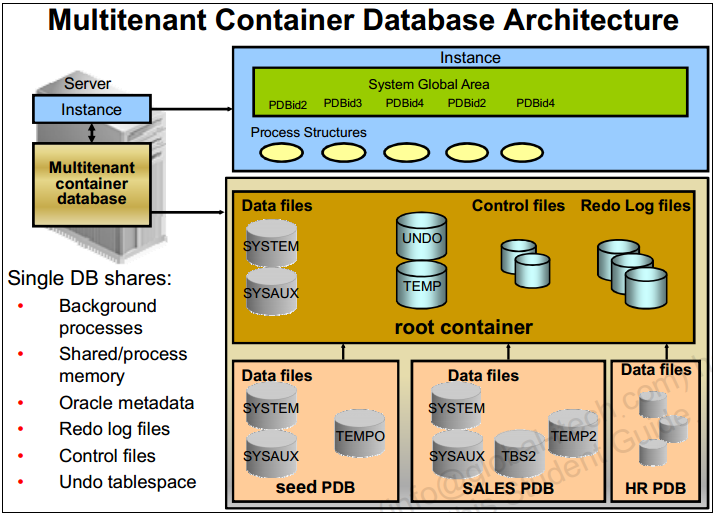
* Rowsize vượt quá data block size
* Table có LONG và LONGRAW columns
* Table có trên 255 column



**Extents**

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

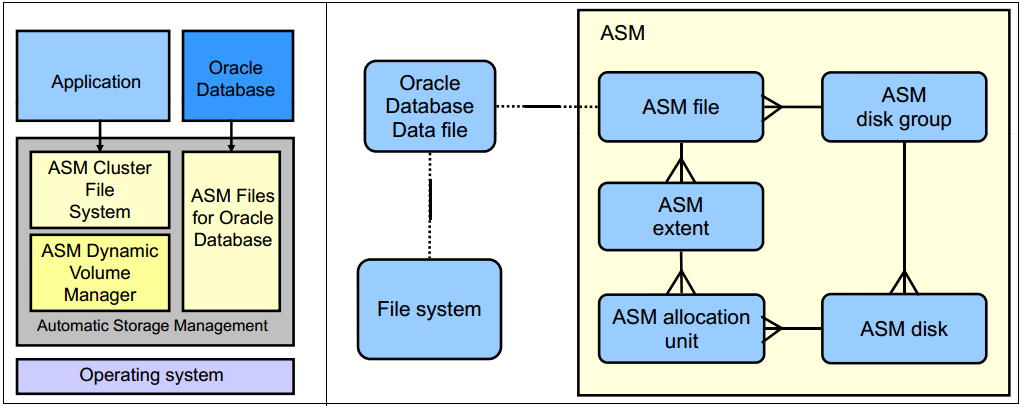
## Multitenant Container Database architecture



At the physical level, the CDB has a database instance and database files, just as a non-CDB does.

* The redo log files are common for the whole CDB. The information it contains is annotated with the identity of the PDB where a change occurs. Oracle GoldenGate is enhanced to understand the format of the redo log for a CDB. All PDBs in a CDB share the ARCHIVELOG mode of the CDB.
* The control files are common for the whole CDB. The control files are updated to reflect any additional tablespace and data files of plugged PDBs.
* The UNDO tablespace is common for all containers.
* A temporary tablespace common to all containers is required. But each PDB can hold its own temporary tablespace for its own local users.
* Each container has its own data dictionary stored in its proper SYSTEM tablespace, containing its own metadata, and a SYSAUX tablespace.
* Th PDBs can create tablespaces within the PDB according to application needs.
* Each data file is associated with a specific container, named CON\_ID.

## Automatic Storage Management (ASM) architecture



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

### Flex ASM

In a typical Grid Infrastructure installation, each node will have its own ASM instance running and act the as the storage container for the databases running on the node. There is a single point-of-failure threat with this setup. For instance, if the ASM instance on the node suffers or fails all the databases and instances running on the node will be impacted. To avoid ASM instance single-point-failure, Oracle 12c provides a Flex ASM feature. The Flex ASM is a different concept and architecture all together. Only a fewer number of ASM Instances need to run on a group of servers in the cluster. When an ASM instance fails on a node, Oracle Clusterware automatically starts surviving (replacement) ASM instance on a different node to maintain availability. In addition, this setup also provides ASM instance load balancing capabilities for the instances running on the node. Another advantage of Flex ASM is that it can be configured on a separate node.

When you choose Flex Cluster option as part of the cluster installation, Flex ASM configuration will be automatically selected as it is required by the Flex Cluster. You can also have traditional cluster over Flex ASM. When you decide to use Flex ASM, you must ensure the required networks are available. You can choose the Flex ASM storage option as part of Cluster installation, or use ASMCA to enable Flex ASM in a standard cluster environment.

The following command shows the current ASM mode:

$ ./asmcmd showclustermode

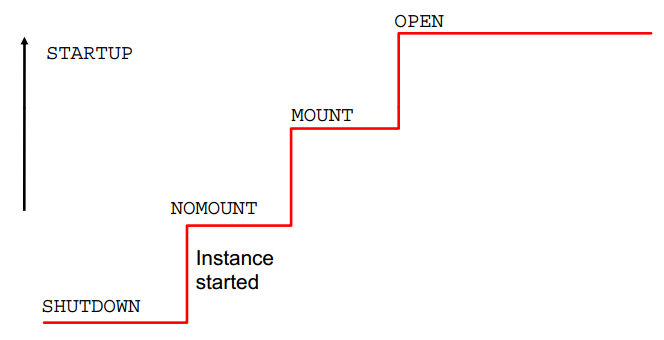
$ ./srvctl config asm

Or connect to the ASM instances and query the INSTANCE\_TYPE parameter. If the output value is ASMPROX, then, the Flex ASM is configured.

# Oracle Database Admin guide

## Startup và Shutdown Database

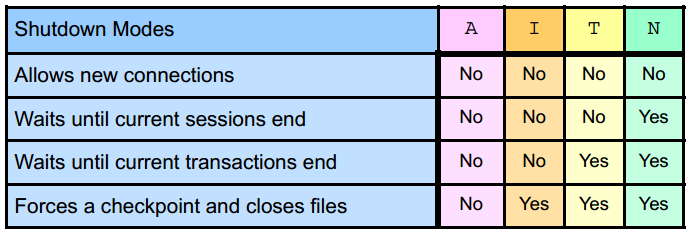
Startup database



Database chỉ có thể được khởi động bởi SYSDBA hoặc SYSOPER, bao gồm start instance và open database. Các bước như sau:

* + Nomount
  + Instance đã được khởi động nhưng chưa được kết nối vs db
    - Oracle sẽ tìm đến spfile (hoặc pfile nếu được chỉ định)
    - Đọc parameter file để cài đặt các tham số
    - Khởi động các background process cần thiết
    - Mở alertlog và trace file để ghi lại log về các tham số đã khởi động
  + Mount
    - Instance đã được kết nối với db bằng cách đọc control file
    - DB chưa được mở, chỉ có DBA mới có quyền truy cập
    - Oracle sau khi đọc spfile (hoặc pfile) xác định được vị trí control file và đọc nó để có tên và vị trí của các datafile và redolog file
  + Open (CDB)
  + DB đã mở, người dùng hợp lệ có thể thao tác vs dữ liệu trên db thông qua instance
  + Oracle open datafile và online redo logfile
  + Open PDB (in Multitant architechture)
  + Mở PDB để ứng dụng có thể truy cập dữ liệu

Shutdown database



Shutdown database sẽ bao gồm close database (close DB or PDB) và shutdown instance. Có 4 mode bao gồm:

* + A = ABORT
  + I = IMMEDIATE
  + T = TRANSACTIONAL
  + N = NORMAL

Shutdown modes are progressively more accommodating of current activity in this order:

* + **ABORT**: Performs the least amount of work before shutting down. Because this mode requires recovery before startup, use it only when necessary. It is typically used when no other form of shutdown works, when there are problems with starting the instance, or when you need to shut down immediately because of an impending situation (such as notice of a power outage within seconds).
  + A shutdown in IMMEDIATE mode proceeds with the following conditions:
    - Current SQL statements being processed by the Oracle database are not completed.
    - The Oracle server does not wait for the users who are currently connected to the database to disconnect.
    - The Oracle server rolls back active transactions and disconnects all connected users.
    - The Oracle server closes and dismounts the database before shutting down the instance.
    - The next startup does not require an instance recovery. TRANSACTIONAL: Allows existing transactions to finish, but does not allow new transactions to start
  + **TRANSACTIONAL**: Allows existing transactions to finish, but does not allow new transaction to start
  + **NORMAL**: Waits for sessions to disconnect

## Loging: Admin log in database

**Alert log file**

Location*: $ORACLE\_BASE/diag/rdbms/<db\_name>/<SID>/trace*

The alert file of a database is a chronological log of messages such as the following:

* + Any nondefault initialization parameters used at startup
  + All internal errors (ORA-600), block corruption errors (ORA-1578), and deadlock errors (ORA-60) that occurred
  + Administrative operations, such as the SQL statements *CREATE, ALTER*, *DROP*

*DATABASE*, and *TABLESPACE*; and the Enterprise Manager or SQL\*Plus statements *STARTUP, SHUTDOWN, ARCHIVE LOG,* and *RECOVER*

* + Several messages and errors relating to the functions of shared server and dispatcherprocesses
  + Errors during the automatic refresh of a materialized view

**DDL Log File**

The DDL log is created only if the ENABLE\_DDL\_LOGGING initialization parameter is set to TRUE

* + DDL log contains one log record for each DDL statement.
  + Two DDL logs containing the same information:
    - XML DDL log: *log.xml* written to

*$ORACLE\_BASE/diag/rdbms/<dbname>/<SID>/log/ddl*

* + - Text DDL: *ddlsid.log* written to

*$ORACLE\_BASE/diag/rdbms/<dbname>/<SID>/log*

**Debug Log File**

Debug log contains warnings about conditions, states, or events that do not inhibit correct operation of an Oracle Database component.

* + The log is intended for use by Oracle Support when diagnosing a problem.
  + It is included in incident packaging service (IPS) incident packages.
  + It is written to

*$ORACLE BASE/diag/rdbms/<db name \_name>/<SID>/debug.*

## User Security

### Database User Accounts

Each database user account has:

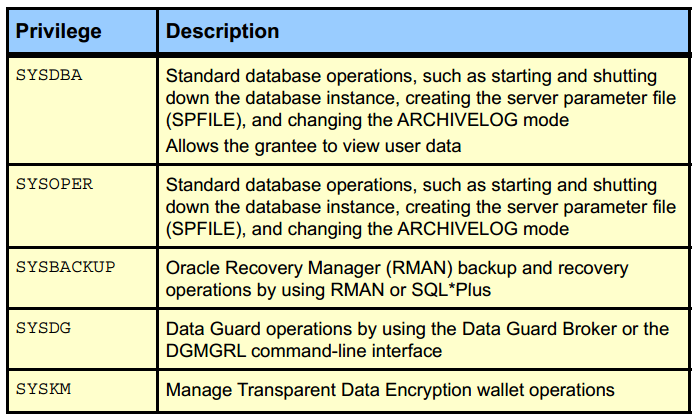
* A unique username
* An authentication method
* A default tablespace
* A temporary tablespace
* A user profile
* An initial consumer group (quota)
* An account status

A schema:

* Is a collection of database objects that are owned by database user
* Has the same name as the user account

### Admin Account

* SYS:
  + Owns the data dictionary and the Automatic Workload Repository (AWR)
  + Used for startup and shutdown of the database instance
* SYSTEM: Owns additional administrative tables and views
* SYSBACKUP: Facilitates Oracle Recovery Manager (RMAN) backup and recovery operations
* SYSDG: Facilitates Oracle Data Guard operations
* SYSKM: Facilitates Transparent Data Encryption walletoperations



### Privileges and Role

Privileges

A privilege is a right to execute a particular type of SQL statement or to access another user’s object. Privileges are divided into two categories:

* **System privileges**: Each system privilege allows a user to perform a particular database operation or class of database operations. For example, the privilege to create tablespaces is a system privilege. System privileges can be granted by the administrator or by someone who has been given explicit permission to administer the privilege. There are more than 170 distinct system privileges. Many system privileges contain the ANY clause.
* **Object privileges**: Object privileges allow a user to perform a particular action on a specific object, such as a table, view, sequence, procedure, function, or package. Without specific permission, users can access only their own objects. Object privileges can be granted by the owner of an object, by the administrator, or by someone who has been explicitly given permission to grant privileges on the object.

**Role**

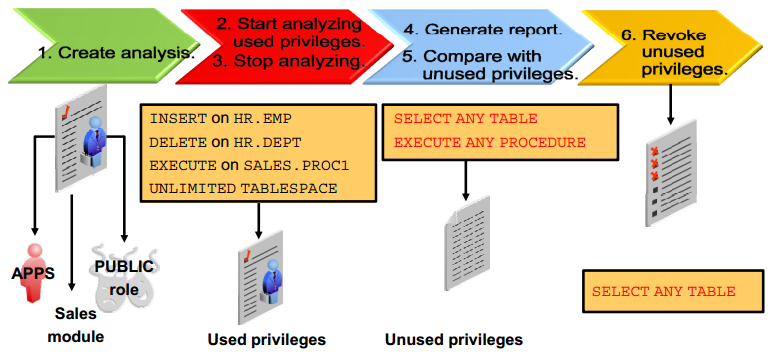
A role is a named group of related privileges that are granted to users or to other roles. You can use roles to administer database privileges. You can add privileges to a role and grant the role to a user. The user can then enable the role and exercise the privileges granted by the role. A role contains all privileges that are granted to that role and all privileges of other roles that are granted to it. Roles provide the following benefits with respect to managing privileges:

* Easier and dynamic privilege management: Use roles to simplify privilege management. Rather than granting the same set of privileges to several users, you can grant the privileges to a role and then grant that role to each user. If the privileges associated with a role are modified, all users who are granted the role acquire the modified privileges automatically and immediately.
* Selective availability of privileges: Roles can be enabled and disabled to turn privileges on and off temporarily. This allows the privileges of the user to be controlled in a given situation.

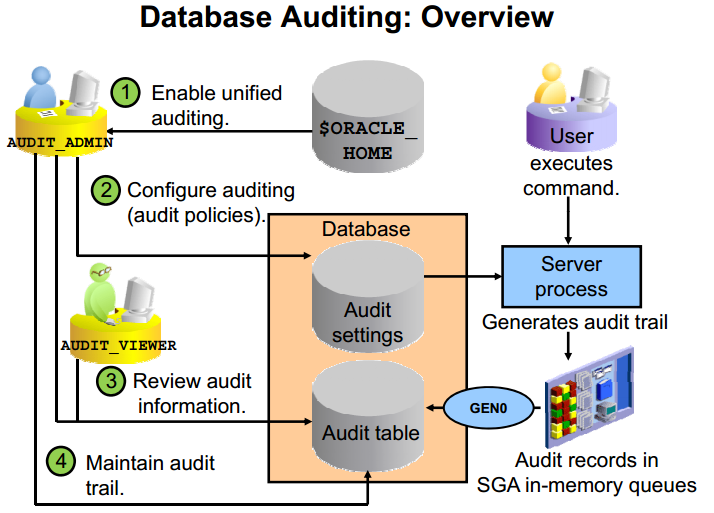
**Privilege Analysis**

* Analyze used privileges to revoke unnecessary privileges.
* Use DBMS\_PRIVILEGE\_CAPTURE package.

A major concern in many databases is that existing database and application users have excessive privileges. Excessive privileges violate the principle of least privilege. To achieve the least privilege principle, unused privileges need to be identified. Oracle Database 12c provides a package named DBMS\_PRIVILEGE\_CAPTURE to analyzeused privileges.



### Auditing



Auditing, which means capturing and storing information about what is happening in the system, increases the amount of work the system must do. Auditing must be focused so that only events that are of interest are captured. Properly focused auditing has minimal impact on system performance. Improperly focused auditing can significantly affect performance.

* **Mandatory auditing**: All Oracle databases audit certain actions regardless of other audit options or parameters. The reason for mandatory audit logs is that the database needs to record some database activities, such as connections by privileged users.
* **Standard database auditing**: Select the objects and privileges that you want to audit and create the appropriate audit policies.
* **Value-based auditing**: Extends standard database auditing, capturing not only the audited event that occurred but also the actual values that were inserted, updated, or deleted. Value-based auditing is implemented through database triggers.
* **Fine-grained auditing (FGA)**: Extends standard database auditing, capturing the actual SQL statement that was issued rather than only the fact that the event occurred.

Through the use of auditing policies, you can configure audit settings for the following activities:

* Logging on to the database and the use of privileges and roles
* Executing SQL statements against specific database objects
* Application context values
  + Utilities and features:
  + Oracle Data Pump
  + Oracle Database Real Application Security
  + Oracle Database Vault
  + Oracle Label Security
  + Oracle Recovery Manager
  + Oracle SQL\*Loader Direct Load

**Unified auditing**

Prior to Oracle Database 12c, audit records from various sources were stored in different locations. Oracle Database 12c supports unified auditing, in which all audit records are stored in a single audit table.

When you create a new Oracle Database 12c database, mixed mode auditing is enabled. This mode enables you to use the auditing features available before Oracle Database 12c and also the unified auditing features. Mixed mode auditing is enabled by default through the ORA\_SECURECONFIG predefined auditing policy for newly created databases.

A user must be granted one of the following roles to perform auditing:

* AUDIT\_ADMIN enables the user to:
  + Create unified and fine-grained audit policies
  + Execute the AUDIT and NOAUDIT SQL statements
  + View audit data
  + Manage the audit trail (table in the AUDSYS schema)
* AUDIT\_VIEWER enables the user to:
  + View and analyze audit data

## Query SQL

### Oracle Statistic

Data statistics trong Oracle Database liên quan đến việc thu thập và lưu trữ thông tin về các đối tượng cơ sở dữ liệu như bảng (tables), chỉ mục (indexes), và phân vùng (partitions). Các thống kê này rất quan trọng cho bộ tối ưu hóa truy vấn (query optimizer) của Oracle để tạo ra các kế hoạch thực thi (execution plans) hiệu quả cho các truy vấn SQL. Bao gồm một số khía cạnh sau:

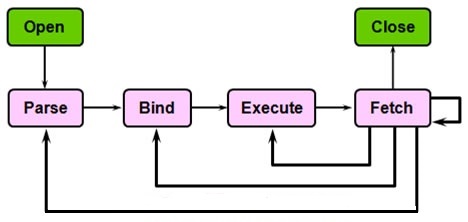
* **Table Statistics**: Bao gồm số lượng hàng (rows), số khối (blocks), số lượng giá trị duy nhất (distinct values) cho các cột, và các phân phối tần suất (frequency distribution) của dữ liệu. Thông tin này giúp bộ tối ưu hóa dự đoán số lượng kết quả truy vấn và chọn phương pháp truy vấn phù hợp.
* **Column Statistics**: Bao gồm thông tin về độ dài trung bình của các giá trị trong cột, số lượng giá trị null, giá trị nhỏ nhất và lớn nhất, và phân phối tần suất. Thông tin này giúp bộ tối ưu hóa đánh giá chi phí và lựa chọn các chỉ mục (indexes) phù hợp cho các truy vấn.
* **Index Statistics**: Bao gồm thông tin về độ sâu của cây chỉ mục (B-tree), số lượng nhánh và lá, và phân phối tần suất của các giá trị chỉ mục. Thông tin này giúp bộ tối ưu hóa quyết định khi nào nên sử dụng chỉ mục để truy vấn dữ liệu.
* **Histograms**: Là các phân phối tần suất chi tiết cho các giá trị trong cột. Histograms cho phép bộ tối ưu hóa hiểu rõ hơn về phân phối dữ liệu, đặc biệt là khi dữ liệu không đồng đều hoặc có các giá trị phổ biến chiếm tỷ lệ cao.
* **Partition Statistics**: Nếu bảng hoặc chỉ mục được phân vùng, các thống kê sẽ được thu thập cho từng phân vùng để bộ tối ưu hóa có thể tạo ra các kế hoạch thực thi hiệu quả hơn khi truy vấn các phân vùng cụ thể.

Oracle cung cấp nhiều cách để thu thập và duy trì các thống kê (Gathering and Maintaining Statistics)

* **Automatic Statistics Collection**: Oracle có thể tự động thu thập thống kê trong các khoảng thời gian nhất định thông qua các job được lập trình sẵn.
* **Manual Statistics Collection**: DBA có thể sử dụng các lệnh như DBMS\_STATS.GATHER\_TABLE\_STATS, DBMS\_STATS.GATHER\_INDEX\_STATS để thu thập thống kê một cách thủ công khi cần thiết.
* **Incremental Statistics**: Cho phép thu thập thống kê chỉ cho các phần dữ liệu thay đổi, giúp tiết kiệm thời gian và tài nguyên.

### **Các bước thực thi một câu lệnh SQL trong Oracle**

Oracle Server xử lý câu lệnh SQL theo trình tự chính sách thực thi các bước sau:



**Open:** Khai báo và khởi tạo Cursor cho câu lệnh SQL

**Parse**:

* Syntatic: Kiểm tra cú pháp
* Semantic: Kiểm tra đối tượng (object)
* View merging: Rewrite lại câu lệnh dựa vào các based table thay vì sử dụng view
* Statement Transformation: Rewirte lại sự biến đổi của câu lệnh để phân tích thành những câu đơn giản hơn.
* Optmization: Tối ưu hóa câu lệnh
* QEP Generation: Query Evulation Plan: đánh giá kế hoạch cho câu lệnh

**Bind:** Tìm và gán giá trị cho các [**bind-variable**](http://www.vietpace.com/kienthuc/VietPace_toiuu_caulenh_Oracle_SQL_Phan1.html#_Giải_nghĩa_từ) nếu có

**Execute**: Thực thi các bước mô tả trong “sơ đồ thực thi câu lệnh SQL”

**Fetch**: Trả kết quả về client gọi thực thi lệnh.

* Fetch ở đây có thể lặp lại nhiều lần do tham số limit của nó ( giới hạn xử lý mỗi lần)

**Close**: Ngầm định đóng Cursor cho câu lệnh

### Excution plan

EXPLAIN PLAN

SET statement\_id = 'ex\_plan2' FOR

SELECT last\_name FROM employees

WHERE last\_name LIKE 'Pe%';

SELECT PLAN\_TABLE\_OUTPUT

FROM TABLE(DBMS\_XPLAN.DISPLAY(NULL, 'ex\_plan2','BASIC'));

----------------------------------------

| Id | Operation | Name |

----------------------------------------

| 0 | SELECT STATEMENT | |

| 1 | INDEX RANGE SCAN| EMP\_NAME\_IX |

----------------------------------------

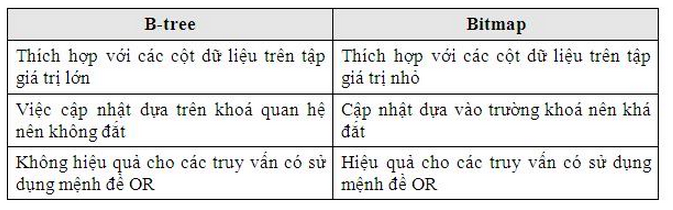
### Cơ chế index của Oracle.

Index là những cấu trúc tùy chọn liên quan đến bảng và cột.

* + Có thể tạo index trên một hoặc nhiều cột của một bảng để tăng tốc độ thực hiện câu lệnh SQL trên bảng đó.
  + Các index này sẽ giúp bạn xác định vị trí thông tin nhanh hơn. Index là cách thức chính của việc giảm đĩa I / O khi được sử dụng đúng cách.
  + Quyết định index đối với 1 bảng dựa vào yêu tố
    - Tạo Index nếu kết quả nhỏ hơn 5->15% tổng số row của 1 bảng lớn
    - Tăng hiệu năng cho phép join bảng (bằng cách index cột dùng để join)
    - Không index cho bảng nhỏ
  + Không index tại cột kiểu LONG và LONGRAW

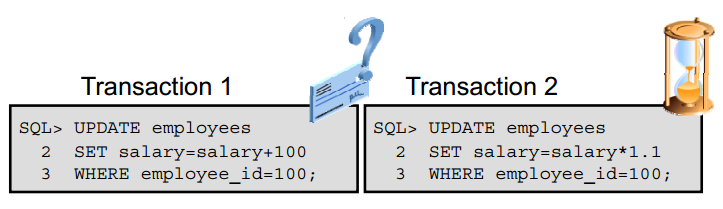
Các loại index chính:

* + B-Tree indexes
    - Btree Index được dùng để giúp truy vấn các câu truy vấn dạng Insert, Update, delete.
    - B-tree là loại index mặc định - nếu tạo ra một index mà không xác định bất cứ điều gì, thì đó là một index B-tree.
  + Bitmap indexes
    - Là 1 index được dùng để làm việc với những trường có dữ liệu rời rạc, với số lượng ít các giá trị khác nhau (tức mức độ lặp lại ở trong trường này thường là lớn)
    - Được sử dụng khI Mức độ dữ liệu trùng lặp lớn: trong oracle thì quyết định là: distinct val/ total val < 1% thì dùng bitmap index: Tức nếu số giá trị rời rạc của 1 cột trong bảng trên tổng số dòng của bảng mà nhỏ hơn 1% thì ta dùngBit map index.
    - Ví dụ trường giới tính: distinct val = 2 <số giá trị riêng biệt> và Total val = 1000 dòng. 2/1000 < 1% dùng bitmap.
    - Không hoặc là ít thao tác update hoặc insert lên bảng dữ liệu.
    - Bảng có rất nhiều cột



* + Bitmap join indexes
    - Bitmap join index cho tham gia giữa các bảng (2+). Tạo Bitmap join index được xác định trên một bảng duy nhất. Nó lưu trữ các kết quả của sự join.
  + Function-based indexes
    - Có thể tạo fuction-based indexes cho những truy vấn sử dụng các function đối với các trường:
      * Upper(abc)
      * 12 \*salary

### Locks



Before the database server allows a session to modify data, the session must first lock the data that is being modified. A lock gives the session exclusive control over the data so that no other transaction can modify the locked data until the lock is released

* + Prevent multiple sessions from changing the same data at the same time
  + Are automatically obtained at the lowest possible level for a given statement
  + Do not escalate

**Locking Mechanism**

* + High level of data concurrency:
    - Row-level locks for inserts, updates, and deletes
    - No locks required for queries
  + Automatic queue management
  + Locks held until the transaction ends (with a commit or rollback operation)

**Lock modes**

* + ROW SHARE: Permits concurrent access to the locked table but prohibits sessions from locking the entire table for exclusive access
  + ROW EXCLUSIVE: Is the same as ROW SHARE, but also prohibits locking in SHARE mode. The ROW EXCLUSIVE locks are automatically obtained when updating, inserting, or deleting data. ROW EXCLUSIVE locks allow multiple readers and one writer.
  + SHARE: Permits concurrent queries but prohibits updates to the locked table. Share locks allow multiple readers and no writers.
    - SHARE ROW EXCLUSIVE: Is used to query y a whole table and to allow others to query rows in the table, but prohibits others from locking the table in SHARE mode or updating rows
    - EXCLUSIVE: Permits queries on the locked table but prohibits any other activity on it. An EXCLUSIVE lock is required to drop a table

**DML Locks**

Each DML transaction obtains two locks:

* + An EXCLUSIVE row lock on the row or rows being updated
  + A table lock (TM) in ROW EXCLUSIVE (RX) mode on the table being updated This prevents another session from locking the whole table (possibly to drop or truncate it) while the change is being made. This mode is also called a subexclusive table lock (SX)

**Lock Conflicts**

Cause

* + Uncommitted changes
  + Long-running transactions
  + Unnecessarily high locking levels

To resolve a lock conflict:

* + Have the session holding the lock commit or roll back
  + Terminate the session holding the lock (in an emergency)

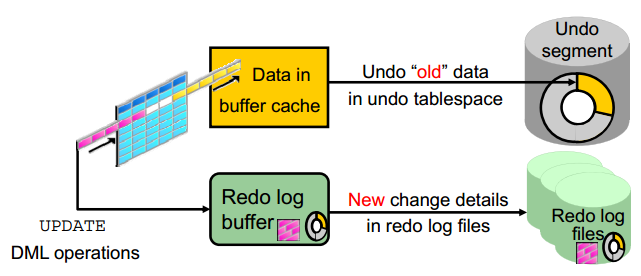
**Deadlock**

A deadlock is a special example of a lock conflict. Deadlocks arise when two or more sessions wait for data that has been locked by the other. Because each is waiting for the other, neither can complete their transaction to resolve the conflict.

Oracle Database automatically detects deadlocks and terminates the statement with an error. The proper response to that error is either commit or roll back, which releases any other locks in that session so that the other session can continue its transaction

## Backup & Recover Database

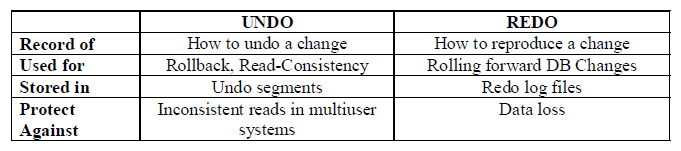
### Undo and Redo



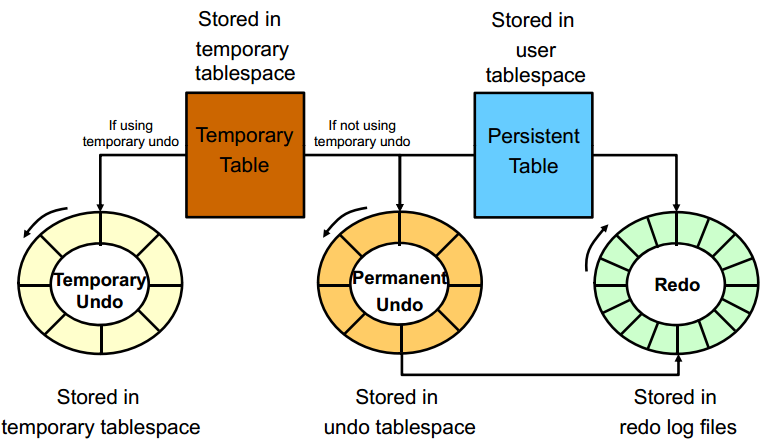
Khi các thao tác DML như: insert/ update / delete thực sự tạo thay đổi cho data block, mặc dù chưa commit thao tác. Để chắc chắn tính nguyên vẹn của database Oracle ghi dữ liệu cũ vào UNDO để có thể rollback lại. Còn các thao tác với dữ liệu ghi vào REDO log để re-play lại thao tác đó nếu như nó thất bại

Undo data is:

* + A record of the action of a transaction
  + Captured for every transaction that changes data
  + Retained at least until the transaction is ended
  + Used to support:
    - Rollback operations
    - Read-consistent queries
    - Oracle Flashback Query, Oracle Flashback Transaction, and Oracle Flashback Table
    - Recovery from failed transactions



**Temporary Undo**



Starting with Oracle Database 12c it is possible for undo generated by temporary tables’ transactions to be stored in a separate undo stream directly in the temporary tablespace to avoid for that undo to be logged in the redo stream. This mode is called temporary undo

Benefits

* + Temporary undo reduces the amount of undo stored in the undo tablespaces.
  + Temporary undo reduces the size of the redo log.
  + Temporary undo enables DML operations on temporary tables in a physical standby database with the Oracle Active Data Guard option.

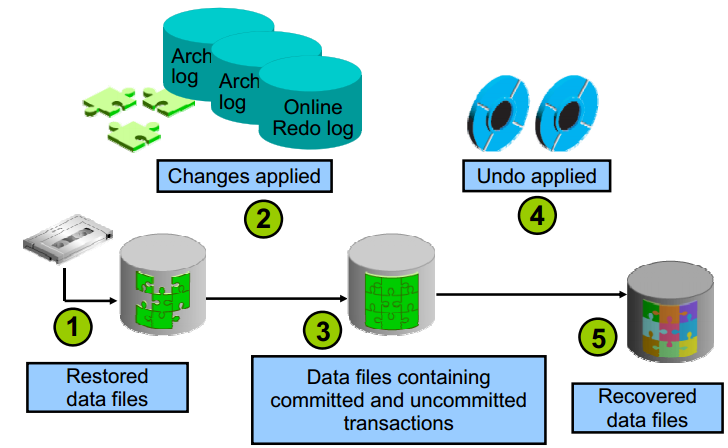
### Recovery

**Automatic Recovery**

Nếu database bị crash hoặc shutdown đột ngột thì khi khởi động lại thì Oracle thực hiện công việc recovery. Công việc này diễn ra như sau

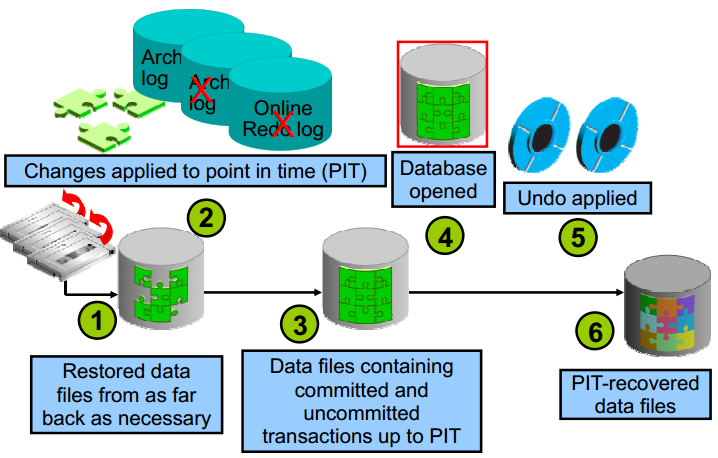
* + Instance startup (data file are out of sync)
  + Roll forward (redo)
  + Committed and uncommitted data in files
  + Database opened
  + Roll back (undo)
  + Committed data in files

**Complete Recovery**



* + Damaged or missing files are restored from a backup.
  + Changes from incremental backups archived redo log files and online redo log files are applied as necessary. The redo log changes are applied to the data files until the current online log is reached and the most recent transactions have been re-entered. Undo blocks are generated during this entire process. This is referred to as rolling forward or cache recovery.
  + The restored data files may now contain committed and uncommitted changes.
  + The undo blocks are used to roll back any uncommitted changes. This is sometimes referred to as transaction recovery.
  + The data files are now in a recovered state and are consistent with the other data files in the database.

**Point-in-Time Recovery**



Incomplete recovery, or database point-in-time recovery (DBPITR), uses a backup to produce a noncurrent version of the database. That is, you do not apply all of the redo records generated after the most recent backup. You need:

* + A valid offline or online backup of all the data files made before the recovery point
  + All archived logs from the time of the backup until the specified time of recovery

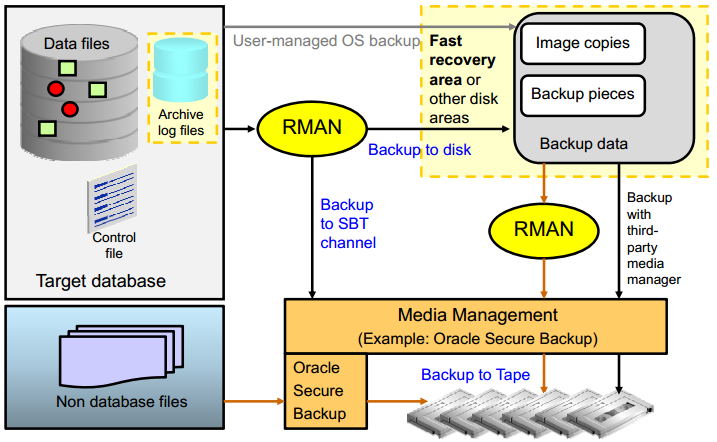
The steps to perform a point-in-time recovery are as follows:

* + Restore the data files from backup: The backup that is used must be from before your target recovery point, using the RMAN RESTORE command.
  + Use the RECOVER command: Apply redo from the archived redo log files, including as many as necessary to reach the restore point destination.
  + State of over-recovery: Now the data files contain some committed and some

uncommitted transactions because the redo can contain uncommitted data.

* + Use the ALTER DATABASE OPEN command: The database is opened before undo is applied This is to provide higher availability.
  + Apply undo data: While the redo was being applied, redo supporting the undo data files was also applied.
  + Process complete: The data files are now recovered to the point in time that you chose.

### Backup



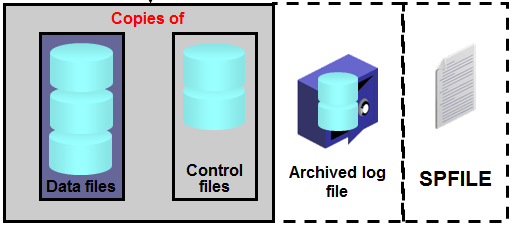
Data backup là việc tạo ra các bản sao của dữ liệu gốc, cất giữ ở một nơi an toàn. Và lấy ra sử dụng (restore) khi hệ thống gặp sự cố. Sao lưu (backup) dữ liệu là cách tốt nhất hiện nay để bảo vệ dữ liệu

Mục địch của việc backup-restore dữ liệu này là để đưa hệ thống trở lại trạng thái trước khi gặp sự cố. Nguyên nhân của sự cố gây ảnh hưởng đến dữ liệu có thể thuộc một trong 2 loại sau:

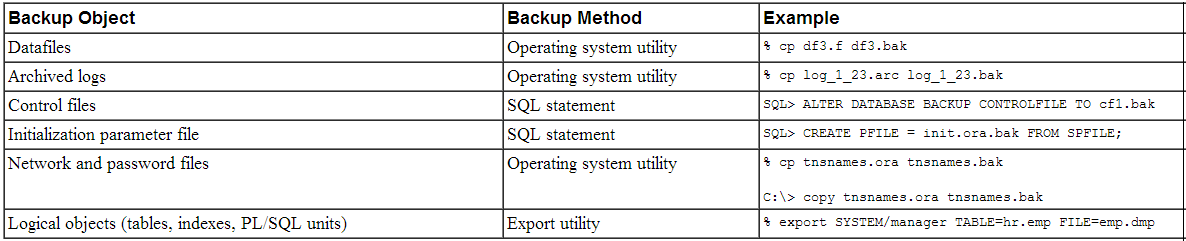
* + Nguyên nhân khách quan: Sự cố xảy ra ngoài ý muốn, con người không thể biết trước được, thường là các thảm họa (VD: thiên tai, cháy nổ).
  + Nguyên nhân chủ quan: Sự cố xảy ra do những thao tác không chính xác của con người (ví dụ: lỗi phần cứng, lỗi phần mềm, thao tác nhầm…).

**Oracle Backup method:**

* + Export và import: data pump
    - Oracle export dữ liệu ra file .dmp
    - File .dmp có thể được lưu trữ như 1 file backup cho db
  + Backup bằng cách sử dụng RMAN
    - Oracle copy toàn bộ các data file và control file, có thể bao gồm archivelog file và spfile thành 1 hoặc nhiều file, lưu trữ ở vùng FRA. Khi cần recover thì mang ra sử dụng



* + User-manged backup
    - Phương thức backup thủ công do DBA thực hiện
    - Thực hiện: Shutdown or freeze DB ở suspended mode, sau đó, copy database sang một phân vùng khác và giải băng cho DB.



**RMAN**

Recovery Manager (RMAN) is the recommended method of backing up your Oracle database. You can use it to back up to disk or to a system backup to tape (SBT) channel. Oracle recommends that disk backups be stored in the fast recovery area (FRA).

RMAN Backup Types

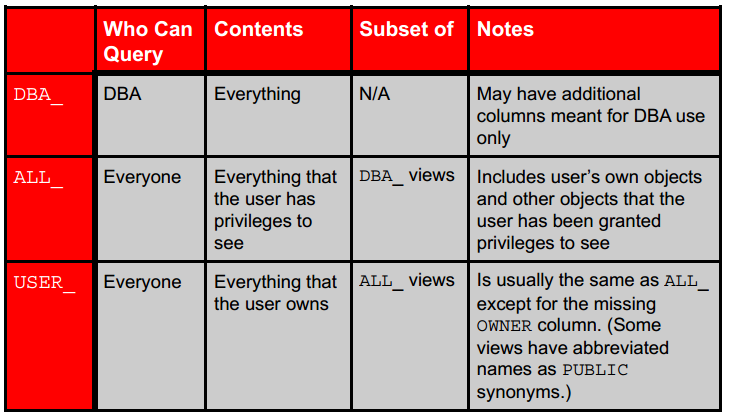
* + Full Backups
    - A full data file backup is a backup that includes every used data block in the file. RMAN copies all blocks into the backup set or image copy, skipping only those data file blocks that are not part of an existing segment. For a full image copy, the entire file contents are reproduced exactly.
    - A full backup cannot be part of an incremental backup strategy; it cannot be the parent for a subsequent incremental backup.
  + Incremental Backups
    - An incremental backup is either a level 0 backup, which includes every block in the data files except blocks that have never been used, or a level 1 backup, which includes only those blocks that have been changed since a previous backup was taken. A level 0 incremental backup is physically identical to a full backup.
    - Mục đích chính của incremental backup:
      * Tiết kiệm thời gian thực hiện và dung lượng của các bản backup.
      * Restore nhanh hơn so với Full backup. Tuy nhiên khi Restore cần đủ file:1 File Full backup lần gần nhất và tất cả các File Incremental backup từ thời điểm Full backup đến thời điểm cần restore.
    - The only difference is that the level 0 backup (as well as an image copy) can be used as the base for a level 1 backup, but a full backup can never be used as the base for a level 1 backup. Incremental backups are specified using the INCREMENTAL keyword of the BACKUP command. You specify INCREMENTAL LEVEL [0 | 1].
      * **Differential**: (default) backs up all blocks changed after the most recent incremental backup at either level 1 or level 0
      * **Cumulative**: Backs up all blocks changed after the most recent backup at level 0

## Database others concept

### Database view

**Data dictionary views**

Oracle cung cấp các cho tất cả các vấn đề trong database với các tiền tố như sau:



**Instance/Database**

* V$DATABASE:  hiển thị thông tin về cơ sở dữ liệu từ các tập tin control file.
* V$INSTANCE:  hiển thị trạng thái của instance hiện tại.
* V$PARAMETER:  hiển thị thông tin về các tham số khởi tạo hiện đang có hiệu lực trong phiên làm việc. Một phiên làm việc mới được thừa hưởng giá trị tham số từ các giá trị sử dụng hoàn toàn hiển thị bởi V $ SYSTEM\_PARAMETER.
* V$SPPARAMETER: hiển thị thông tin nội dung của các tập tin tham số máy chủ.
* V$SYSTEM\_PARAMETER:  hiển thị thông tin về các tham số parameter hiện đang có hiệu lực. Một session mới được thừa hưởng giá trị tham số từ các giá trị instance mở rộng.
* V$PROCESS: chứa thông tin về các tiến trình đang hoạt động
* V$BGPROCESS: hiển thị thông tin về các tiến trình nền(background processes)
* V$PX\_PROCESS\_SYSSTAT: chứa thông tin về các session chạy thực hiện song song
* V$SYSTEM\_EVENT: chứa thông tin về tổng số chờ đợi cho một sự kiện.

**Disk**

* + V$DATAFILE:  chứa thông tin datafile từ các tập tin control file.
  + V$FILESTAT: hiển thị số lượng vật lý đọc và ghi được thực hiện và tổng số single-block và multi-block I / O được thực hiện.
  + V$LOG\_FILE: chứa thông tin về redo log file.
  + V$LOG\_HISTORY: chứa log thông tin đăng nhập lịch sử từ control file.
  + V$DBFILE: chứa thông tin datafile từ các tập tin control file.
  + V$TEMPFILE: hiển thị thông tin tempfile.
  + V$TEMPSEG\_USAGE: mô tả chi tiết sử dụng temporary segment (Bảng chứa dữ liệu tạm thời)
  + V$SEGMENT\_STATISTICS:  hiển thị thông tin về số liệu thống kê segment-level statistics. Tham số tĩnh.

**Memory**

* + V$BUFFER\_POOL\_STATISTICS:  hiển thị số liệu thống kê về tất cả tham số về vùng nhớ đệm buffer pools.
  + V$LIBRARYCACHE: chứa số liệu thống kê tham số về hiệu suất bộ nhớ cache thư viện và hoạt động.
  + V$SGAINFO: hiển thị thông tin về kích thước SGA, bao gồm các kích thước của các thành phần khác nhau SGA, kích thước hạt, và bộ nhớ miễn phí.
  + V$PGASTAT:  cung cấp thông tin về PGA thống kê sử dụng bộ nhớ cũng như thống kê về quản lý bộ nhớ PGA tự động khi nó được kích hoạt.

**Contention**

* + V$LOCK: liệt kê các lock hiện đang nắm giữ cơ sở dữ liệu Oracle và yêu cầu requests lock
  + V$UNDOSTAT: hiển thị một biểu đồ số liệu thống kê cho thấy hệ thống làm việc như thế nào. Số liệu thống kê bao gồm tiêu thụ undo space, các transaction đồng thời , và chiều dài của các truy vấn được thực hiện trong mọi trường hợp.
  + V$WAITSTAT: thông số block contention. Bảng này chỉ được cập nhật khi time statistic được kích hoạt.

### Oracle SCN

SCN (System Change Number): đây là một số gia tăng duy nhất trong cơ sở dữ liệu (như đồng hồ thời gian của bạn). Số SCN được tăng lên mỗi 3 giây. Con số này là rất hữu ích trong khi phục hồi cơ sở dữ liệu.

* + SCN xác định một phiên bản đã commit của cơ sở dữ liệu tại một thời điểm. Oracle gán mọi transaction đã commit một SCN.
  + SCN là một dấu thời gian nội bộ cho một commited version của cơ sở dữ liệu . Các máy chủ cơ sở dữ liệu Oracle sử dụng SCN clock để đảm bảo tính nhất quán SCN transaction. Ví dụ, khi một người sử dụng commit một transaction, cơ sở dữ liệu ghi lại một SCN này và commit ghi trên redo log.
  + SCNs là quan trọng đối với transaction bởi vì nó có chức năng như một dấu thời gian đồng bộ commit transaction , ngay cả khi transaction không thành công. Nếu một transaction thay đổi dữ liệu sai hoặc không phù hợp, một quản trị viên có thể sử dụng SCN này để phối hợp thay đổi trên cơ sở dữ liệu. SCN cho transaction commit cũng có thể được sử dụng để xác định các transaction sau đó.
  + Nếu SCN không đồng nhất giữa các thành phần CSDL sẽ nảy sinh vấn đề. Database sẽ lấy SCN của control file và apply cho tất cả các thành phần khác
  + Control file lưu 1 thông tin rất quan trọng là SCN. Con số này được phát sinh và tăng liên tục theo thời gian. Oracle dựa vào con số này đề đồng bộ tất cả các file trong database như: datafile, control file, redo log file v.v...

### Oracle Service

An automatic workload management facility, called database services. Database services (services) are logical abstractions for managing workloads in Oracle Database. Services divide workloads into mutually disjoint groupings. Each service represents a workload with common attributes, service-level thresholds, and priorities.

A single service can represent an application, multiple applications or a subset of a single application. For example, the Oracle E-Business suite defines a service for each responsibility, such as general ledger, accounts receivable, order entry, and so on. A single service can be associated with one or more instances of an Oracle RAC database, and a single instance can support multiple services.

### Oracle Resource Management

* + Purpose: cấp quota cho các resource consumer group
  + Đọc thêm trong tài liệu

### Database Link

Sending data or messages between sites requires network configuration on both sites.

You must configure the following:

* + Network connectivity (for example, tnsnames.ora)
  + Database links

SQL> CREATE [PUBLIC] DATABASE LINK <link\_name>

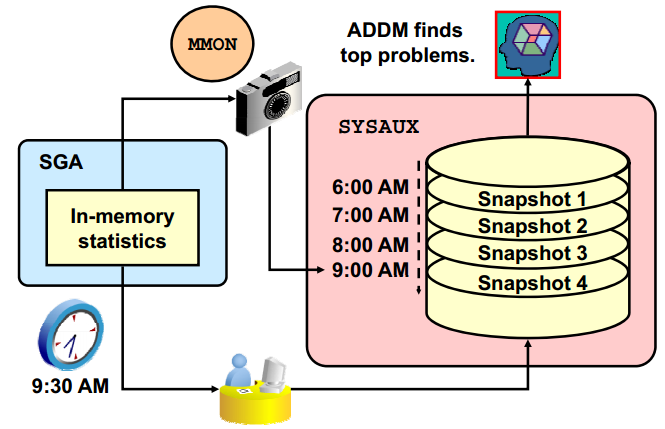
CONNECT TO <user\_name>

IDENTIFIED BY <password>

USING '<service\_name>';

# Oracle Database Advanced feature

## AWR



The Automatic Workload Repository (AWR) is a collection of persistent system performance statistics owned by SYS. The AWR resides in the SYSAUX tablespace.

A snapshot is a set of performance statistics captured at a certain time and stored in the AWR. Each snapshot is identified by a snapshot sequence number (SNAP\_ID) that is unique in the AWR. By default, snapshots are generated every 60 minutes. You can adjust this frequency by changing the snapshot INTERVAL parameter. Because the database advisors rely on these snapshots, be aware that adjustment of the interval setting can affect diagnostic precision. For example, if the INTERVAL is set to four hours, you may miss transient events that would be noticeable in 60-minute intervals.

## Oracle Scheduler job

**Purpose:**

* + Tự động hóa các công việc theo chu kỳ của ứng dụng hay các công việc maintain database hay làm báo cáo

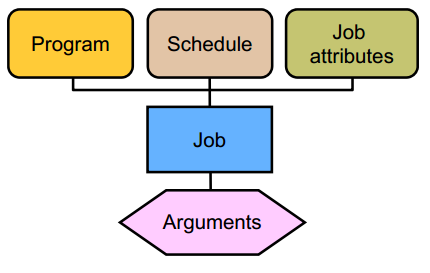
**Concept**

Oracle Database provides advanced scheduling capabilities through the Oracle Scheduler, which is a collection of functions and procedures in the DBMS\_SCHEDULER package.

A job has two key components:

* + Action: “What” needs to be done
  + Schedule: “When” the action occurs

**Components**



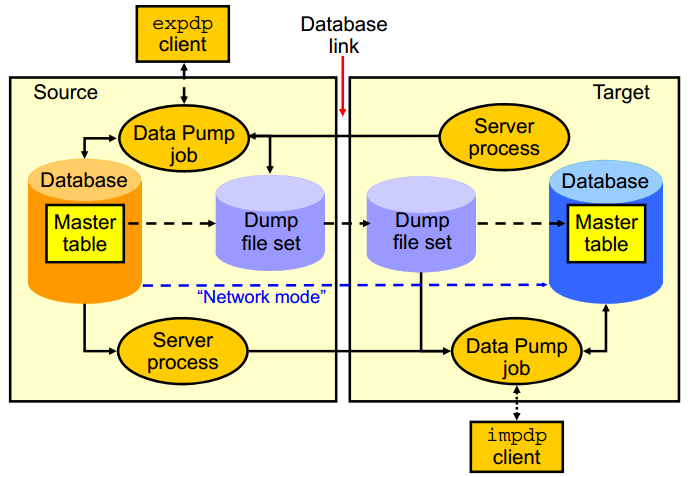
The Scheduler uses the following basic components:

* + **A job** specifies what needs to be executed. It could be as an example a PL/SQL procedure, a native binary executable, a Java application, or a shell script.
  + **A schedule** specifies when and how many times a job is executed A schedule can be . A schedule can be based on time or an event. You can define a schedule for a job by using a series of dates, an event, or a combination of the two, along with additional specifications to denote repeating intervals. You can store the schedule for a job separately and then use the same schedule for multiple jobs.
  + **A program** is a collection of metadata about a particular executable, script, or procedure. An automated job executes some task. Using a program enables you to modify the job task, or the “what,” without modifying the job itself. You can define arguments for a program, enabling users to modify the runtime behavior of the task.

Using Oracle Scheduler

* + Create a program (enabled or disabled)- optional
    - To reuse this action within multiple jobs
    - To change the schedule for a job without having to re-create the PL/SQL block
  + Create and use a schedule (event base, time base, fix base)
  + Create and submit a job.

## Oracle Data Pump



**Purpose:**

* + Expdp: backup data, migration data
  + Impdp: load data to database

**Concept:**

Oracle Data Pump enables very high-speed data and metadata loading and unloading of Oracle databases. The Data Pump infrastructure is callable via the DBMS\_DATAPUMP PL/SQL package. Thus, custom data movement utilities can be built by using Data Pump.

Oracle Database provides the following tools:

* + Command-line export and import clients called expdp and impdp, respectively
  + Export and import interface in Enterprise Manager Cloud Control

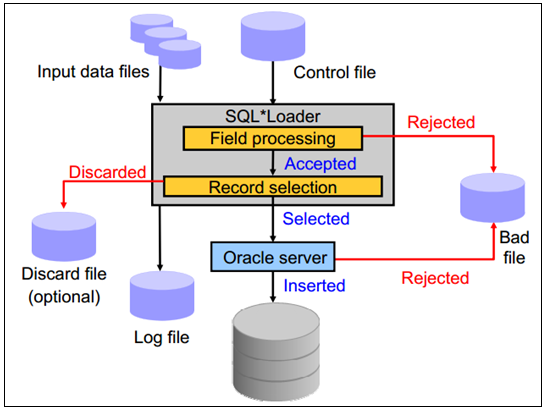
Data Pump automatically decides the data access methods to use; these can be either direct path or external tables. Data Pump uses direct path load and unload when a table’s structure allows it and when maximum single-stream, performance is desired. However, if there are clustered tables, referential integrity constraints, encrypted columns, or several other items, Data Pump uses external tables rather than direct path to move the data.

The ability to detach from and re-attach to long-running jobs without affecting the job itself enables you to monitor jobs from multiple locations while they are running. All stopped Data Pump jobs can be restarted without loss of data as long as the metainformation remains undisturbed. It does not matter whether the job is stopped voluntarily or involuntarily due to a crash.

New feature from 12c:

* + Turn off redo log generation
  + Transport view as table

## SQL\*Loader



**Purpose:**

* + Load một lượng dữ liệu lớn vào database

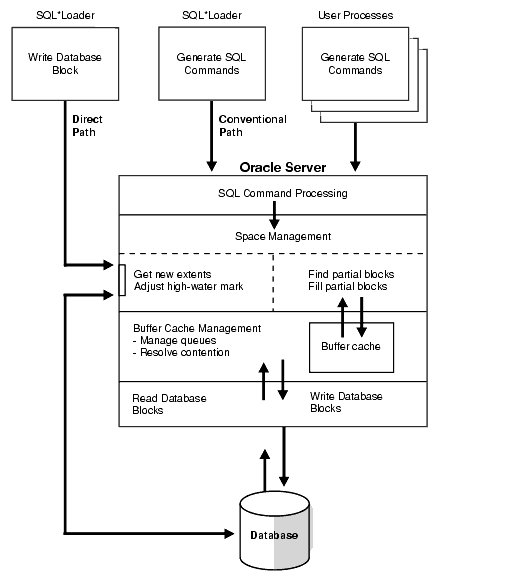
**Concept:**

SQL\*Loader loads data from external files into tables of an Oracle database. It has a powerful data parsing engine that puts little limitation on the format of the data in the data file. SQL\*Loader uses the following files:

* + **Input data files**: SQL\*Loader reads data from one or more files (or operating system equivalents of files) that are specified in the control file.
  + **Control file**: The control file is a text file that is written in a language that SQL\*Loader understands. The control file indicates to SQL\*Loader where to find the data, how to parse and interpret the data, where to insert the data, and so on. • The first section contains such session-wide information as the following:
  + **Log file**: When SQL\*Loader begins execution, it creates a log file. If it cannot create a log file, execution terminates. The log file contains a detailed summary of the load, including a description of any errors that occurred during the load.
  + **Bad file**: The bad file contains records that are rejected, either by SQL\*Loader or by the Oracle database. Data file records are rejected by SQL\*Loader when the input format is invalid.
  + **Discard file**: This file is created only when it is needed and only if you have specified that a discard file should be enabled. The discard file contains records that are filtered out of the load because they do not match any record-selection criteria specified in the control file.

**Direct path và conventional path**

SQL Loader cung cấp 2 phương thức để insert dữ liệu vào database

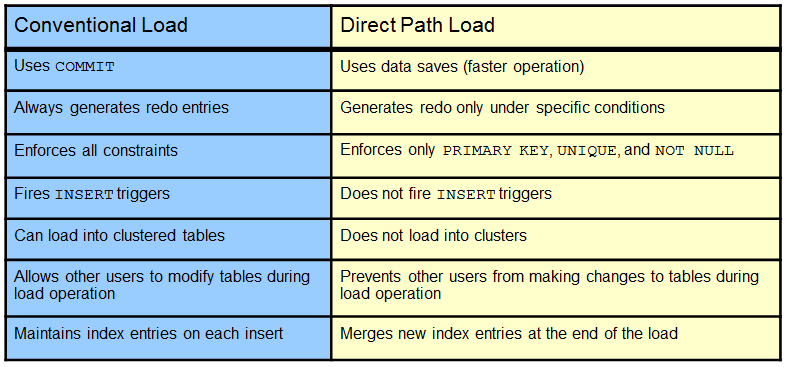


* + Conventional path thực thi câu lệnh insert tới các bảng trong 1 Oracle DB:

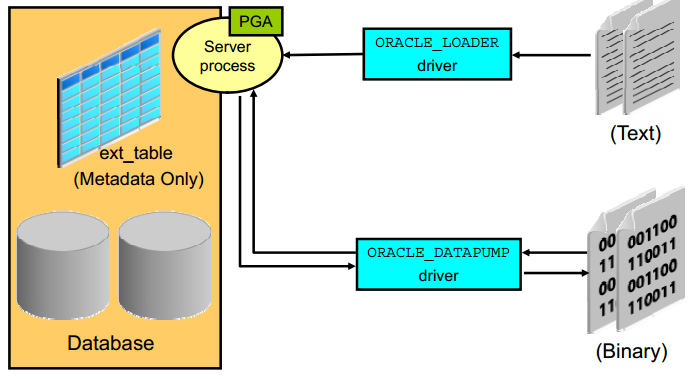
INSERT INTO TABLE T PARTITION (P) VALUES ...

* + Direct path ghi trực tiếp lên các datablock trên datafile

LOAD INTO TABLE T PARTITION (P) VALUES ...



## External Tables



External tables access data in external sources as if it were in a table in the database. You can connect to the database and create metadata for the external table using DDL. The DDL for an external table consist of two parts: one part that describes the Oracle Database column types, and another part that describes the mapping of the external data to the Oracle Database data columns.

An external table does not describe any data that is stored in the database. Nor does it describe how data is stored in the external source. Instead, it describes how the external table layer must present the data to the server. It is the responsibility of the access driver and the external table layer to do the necessary transformations required on the data in the external file so that it matches the external table definition External tables are read only; therefore no. External tables are read only; therefore, no DML operations are possible, and no index can be created on them.

There are two access drivers used with external tables. The ORACLE\_LOADER access driver can be used only to read table data from an external table and load it into the database. It uses text files as the data source. The ORACLE\_DATAPUMP access driver can both load table data from an external file into the database and also unload data from the database into an external file. It uses binary files as the external files. The binary files have the same format as the file used by the Data pump Import and Export utilities and can be interchanged with

## In-Memory Column Store

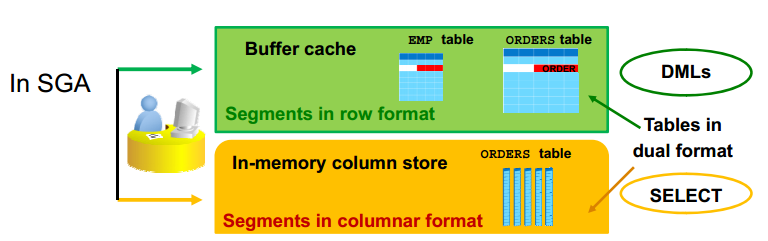
The In-Memory Column Store feature enables objects (tables, partitions, and other types) to be stored in memory in a new format known as the columnar format. This format enables scans, joins, and aggregates to perform much faster than the traditional on-disk format, thus providing fast reporting and DML performance for both OLTP and DW environments.

This is particularly useful for analytic applications that operate on **few columns returning many rows** rather than for OLTP that operates on few rows returning many columns. **The DBA must define** the segments that are to be populated into the in-memory column store (IM column store), such as hot tables, partitions, and more precisely the more frequently accessed columns.

The in-memory columnar format does not replace the on-disk or buffer cache format It is a consistent copy of a table or of some columns of a table converted to the new columnar format that is independent of the disk format and only available in memory. Because of this independence, applications are able to transparently use this option without any changes. For the data to be converted into the new columnar format, a new pool is requested in the SGA. The pool is the **IM column store**.

There are three main advantages:

* + Queries run a lot faster: All data can be populated in memory in a compressed columnar format. No index is required and used. Queries run at least 100 times faster than when fetching, p data from the buffer cache, thanks to the columnar compressed format.
  + DMLs are faster: Analytics indexes can be eliminated by being replaced by scans of the IM column store representation of the table.
  + Arbitrary ad hoc queries run with good performance, because the table behaves as if all columns are indexed.



## Full Database In-Memory Caching

The Full Database In-memory Caching feature enables an entire database to be cached in memory when the database size (sum of all data files, SYSTEM tablespace LOB CACHE files, LOB CACHE files minus SYSAUX, TEMP) is smaller than the buffer cache size. Caching and running a database from memory leads to huge performance benefits. Two modes can be used:

Full Database Caching: Implicit default and automatic mode in which an internal calculation determines if the database can be fully cached for an instance. NOCACHE LOBs are not cached in Full Database Caching but in Force Full Database Caching mode even NOCACHE LOBs are cached.

Force Full Database Caching: Neither Full Database Caching nor Force Full Database Caching forces or prefetches data into memory. Workload must access the data first for them to be cached. It considers the entire database as eligible to be completely cached in the buffer cache. This mode requires the DBA to execute the *ALTER DATABASE FORCE FULL DATABASE CACHING* command. This mode takes precedence over Full Database Caching mode. To revert to traditional caching, use the *ALTER DATABASE NO FORCE FULL DATABASE CACHING* command

