Document

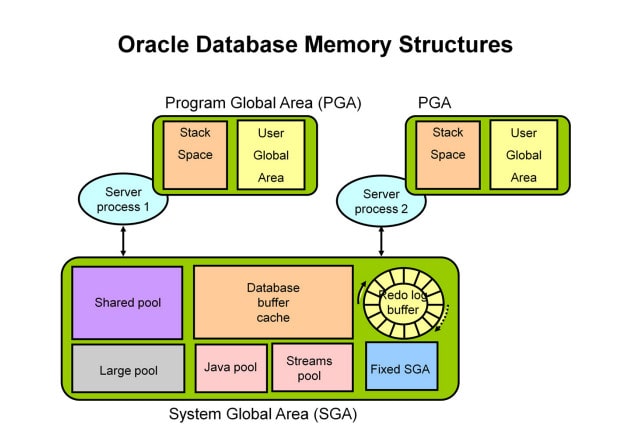
# Kiến trúc

## Kiến trúc bộ nhớ (Memory)

KTBN xử lý các request từ phía người dùng và thực hiện các công việc nội tại của Oracle. KTBN phân thành 2 khu vực:

* SGA – Systen global area: bộ nhớ dùng chung. Tại đây các session sẽ đọc dữ liệu thông qua SGA (từ đĩa gửi lên SGA), đồng thời SGA cũng xử lý các tác vụ như phân tích câu lệnh, thực hiện các tiến trình, …
* PGA – Program global area: bộ nhớ riêng, cấp phát cho từng session. Chứa thông tin của từng session: thông tin cá nhân, các giá trị cụ thể mà session sử dụng, …

SGA – xử lý các hoạt động chung của oracle, được chia thành các phân vùng nhỏ như sau:



### Share Pool (học thuộc lòng)

Sử dụng để phân tích cú pháp câu lệnh SQL dạng text; kiểm tra các object sẽ làm việc và có quyền làm việc trên các đối tượng đó không, đồng thời phân tích phương án thực thi câu lệnh (Exectution plan).

Share pool gồm các thành phần:

* Data dictionary cache: lưu trữ thông tin row image của data dictionary tables. Chứa thông tin về các object trong Cơ sở dữ liệu. Do lưu trữ dạng row nên thường gọi là row cache.
* Library cache: Lưu parse tree và execution plan của các câu lệnh hay thông tin câu lệnh SQl,.PL SQL đã thực thi. Sử dụng giải thuật LRU để lưu trữ thông tin hữu ích, những thông tin kém hiệu quả sẽ được loại bỏ.
* Server result cache: Chứa kết quả của các câu lệnh.
* Reserved pool: chứa các bộ nhớ liền nhau để cấp phát cho share pool.

### Database buffer cache (học thuộc lòng)

Lưu các block dữ liệu đọc từ các datafile vật lý. Khi user process yêu cầu một khối dữ liệu nào đó, trước tiên sẽ tìm trong Database buffer cache, nếu tìm thấy gọi là cache hit và đọc từ memory, nếu không tìm thấy gọi là cache mis thì data block phải copy từ đĩa lên buffer cache, sau đó mới gửi cho người dùng.

Các dữ liệu trong Buffer cache quản lý theo giải thuật LRU.

Các vùng nhớ con:

* Keep pool: Lưu trữ các object lâu hơn so với giải thuật LRU.   
  Create table: <table\_name> <col\_name col\_type> storage (buffer\_pool keep).
* Recycle pool: Xóa các object trong recycle pool nhanh hơn so với giải thuật LRU.  
  Create table <table\_name> <col\_name col\_type> storage (buffer\_pool recycle)
* nK buffer cache: Sử dụng để chứa các data block size khác nhau. Mặc định data block size 8Kb.

Check các bạng đang ở dạng Keep pool hay Recycle pool:

SELECT TABLE\_NAME, BUFFER\_POOL from dba\_tables where BUFFER\_POOL <> ‘default’

ALTER TABLE <table\_name> STORAGE (buffer\_pool default) (chuyển bảng về dạng default)

Khi người thực hiện cập nhật dữ liệu, dữ liệu sẽ không được ghi ngay xuống đĩa mà lưu trong dirty block.

### Redo log buffer (học thuộc lòng)

Lưu trữ những change vector mô tả sự thay đổi của các block (DML, DDL – Lưu trước khi những thay đổi xảy ra). Thông tin sẽ được sử dụng trong quá trình Database Recovery (ví dụ như mất điện đột ngột).

Dung lượng của các change vector này nhỏ hơn rất nhiều so với bản thân các block. Do đó, Log Writer có thể ghi với tần suất lớn hơn rất nhiều Database Writer mà không sợ ảnh hưởng quá nhiều đến băng thông của đĩa.

Định kỳ thông tin sẽ được lưu vào file vật lý (Redo log file).

### Large pool

Sử dụng cho các session với kết nối share server và oracl xa interface. Và sử dụng cho các tiến trình chiếm nhiều tài nguyên I/O như backup và restore database hoặc sử dụng cho các câu lệnh parallel.

Không quản lý theo giải thuật LRU. Phần nhớ nào cache lên thì khi và chỉ khi nó kết thúc câu việc mới được giải phóng.

### Java pool

Lưu java execution code cũng giống như share pool lưu trữ thông tin về plsql.

Show parameter java\_pool size;

### Stream pool

Được sử dụng bởi Oracle Streams – 1 công nghệ hỗ trợ đồng bộ dữ liệu từ db này sang db khác.

Lưu thông tin kiểu Queue message. Cung cấp bùng bộ nhớ cho Oracle Stream lưu thông tin về dữ liệu và các sự kiện đang được chia sẻ trong môi trường phân tán.

Được thiết lập bởi tham số STREAM\_POOL\_SIZE, nếu tham số này = 0 thì các hoạt động vẫn xảy ra và bộ nhớ được lấy từ sharepool (<= 10%).

### In – memory store

## Kiến trúc tiến trình

### User process

Là những ứng dụng, công cụ kết nối vào CSDL. Có thể kết nối trực tiếp trên db hoặc kết nối từ bên ngoài.

### Listener process

Khi user process yêu cầu kết nối đến CSDL cần đi qua 1 process khác để có thể kết nối được gọi là Listerner process.

### Server process

Sau khi đi qua listener thì server process và user process sẽ làm việc trực tiếp với nhau không thông qua listener nữa. Từ đây server process sẽ thực hiện tiếp nhận, xử lý yêu cầu và gửi lại kết quả cho user process.

Khi một server process sinh ra, nó sẽ được cấp phát 1 PGA riêng và được phép sử dụng các tài nguyên trên SGA.

### Backgroud process

Backgroud process bao gồm các process con thực hiện các nhiệm vụ riêng trong quá trình Oracle vận hành database như ghi dữ liệu từ memory xuống đĩa, dọn dẹp xử lý sau khi server process hoạt động xong, ghi những thay đổi do server process sinh ra, …

Backgroud process luôn hoạt động để giữ cho database được vận hành trơn tru.

#### Database writer (DBWRn)

Database writer có nhiệm vụ định kỳ sẽ ghi các dữ liệu từ các Block bị thay đổi trên Data buffer cache xuống đĩa.

DBWRn sẽ ghi dữ liệu theo cơ chế ghi càng ít càng tốt, càng hiếm khi càng tốt:

* Không còn free buffer: xem toàn bộ db buffer cache, nếu k còn vùng trống sẽ ghi dữ liệu, nhưng chỉ ghi một số.
* Quá nhiều dirty buffer: ngưỡng do Oracle ngầm định và cũng chỉ ghi một số.
* 3s/ lần
* Checkpoint:
  + Full checkpoint: Ghi toàn bộ dirty buffer xuống datafile. Xảy ra khi db được tắt đúng quy trình.
  + Partical checkpoint: DBWRn sẽ ghi dirty buffer ít hơn xuống datafile khi offline tablespace; tablespace được đưa về trạng thái backup mode; tablespace chuyển sang read only.

#### Log writer (LGWR)

LGWR ghi dữ liệu từ redo log buffer xuống Online redo log (Redo log file).

Cấu trúc của Redo log file gồm nhiều group: thường 3 group. LGWR sẽ ghi vào các group dạng xoay vòng (nếu đầy) group 1 🡪 group 2 🡪 group 3 🡪 group 1 (switch log).

LGWR xảy ra khi:

* Khi người dùng thực hiện commit.
* Khi 1/3 Redo log buffer đầy.
* Khi DBWRn muốn thực thi, LGWR sẽ được thực thi trước.
* 3s/ lần.

#### Checkpoint (CKPT)

CKPT là một tiến trình nhằm đảm bảo tính nhất quán và toàn vẹn dữ liệu.

Khi CKPT thực thi, dữ liệu trên Data buffer cache và Redo log buffer sẽ được ghi xuống đĩa.

CKPT sẽ ghi 1 con số đánh dấu – gọi là System Change Number (SCN) vào control file và data file header 🡪 Tại thời điểm checkpoint, số SCN trong controlfile và data file header giống hệt nhau.

CKPT xảy ra khi:

* Redo log file thực hiện switch log
* Khi thực hiện ALTER SYSTEM CHECKPOINT (full checkpoint)
* Khi thực hiện ALTER SYSTEM SWITCH LOGFILE
* Thời gian của Log\_CheckPoint\_Time đạt ngưỡng (1800s)

Các loại Checkpoint:

* Thread checkpoint
* Tablespace and datafile checkpoint
* Incremental checkpoint

#### System Monitor (SMON)

System Monitor (SMON) là background process của Oracle được tạo ra ngay khi khởi động Oracle Database instance (Mount/ Open/ Recovery lúc db instance startup).

SMON có nhiệm vụ khôi phục dữ liệu khi khởi động (instance recovery) như đã nói bên trên.

Ngoài ra, nó còn dọn dẹp các dữ liệu tạm, gom các extend trống gần kề nhau thành 1 extend lớn hơn, nhằm tránh phân mảnh dữ liệu. Các công việc này, SMON sẽ thực hiện 5 phút 1 lần.

SMON phải luôn thường trực trong suốt quá trình instance hoạt động, nếu bạn thử kill tiến trình này, instance cũng chết luôn.

“ps -ef |grep smon” – tìm tiến trình smon: ora\_smon\_<db instance>

#### Program Monitor (PMON)

Program Monitor (PMON) là một **background process** của Oracle cũng được tạo khi khởi động Oracle Database instance.

Nhiệm vụ của PMON:

* Dọn dẹp các Database buffer cache của user process.
* Giải phóng tài nguyên được cấp phát nếu user process bị lỗi.
* Giám sát các session kết nối vào database cho việc quản lý session time out.

#### Recoverer (RECO)

Làm việc trong giao dịch dạng phân tán (Distributed transaction). Đảm bảo các giao dịch phân tán hoạt động như một thể thống nhất (cùng commit hoặc cùng rollback).

Ví dụ: 1 transaction làm việc trên 2 database A, B

* DB A: Update employee Set Salary = Salary \* 1.1 Where Id = 1
* DB B: Update employee@DatabaseB Set Salary = Salary \* 1.2 Where Id = 10
  + - * Khi commit thì cả 2 câu lệnh trên đều cùng commit.

#### Listener Registrator (LREG)

Nhiệm vụ của LREG là đăng ký thông tin của Database instance với Listener. Thông thường mất 60s để việc tự động đăng ký được hoàn thành.

Có thể sử dụng: ALTER SYSTEM REGISTER để diễn ra ngay lập tức.

#### Archiver (ARCn)

ARCn sao lưu thông tin của redo log file xuống archiver log file.

Nếu bật chế độ No archiver log mode, sau khi LGWR ghi đầy group 3, nó sẽ ghi đè vào group 1 🡪 Ảnh hưởng việc restore, recovery database.

Nếu bật chế độ Archiver log mode, sau khi LGWR ghi đầy group 3, dữ liệu trong group 1 sẽ được sao lưu vào Archiver log file, sau đó LGWR sẽ ghi tiếp vào group 1 🡪 Có thể dựa vào Archiver log file để khôi phục dữ liệu.

Check chế độ: Archiver log list.

## Kiến trúc lưu trữ

### Control file

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 hay redo log files.
* Time stamp (mốc thời gian) tạo lập database, ...

Mỗi khi nào một instance của Oracle database được mở, control file của nó sẽ được sử dụng để xác định data files và các redo log files đi kèm. Khi các thành phần vật lý cả database bị thay đổi (ví dụ như, tạo mới datafile hay redo log file), Control file sẽ được tự động thay đổi tương ứng bởi Oracle.

Control file cũng được sử dụng đến khi thực hiện khôi phục lại dữ liệu

### Online redo log file

Mỗi Oracle database đều có một tập hợp từ 02 *redo log files* trở lên. Các redo log files trong database thường được gọi là database *redo log*. Một redo log được tạo thành từ nhiều redo entries (gọi là các *redo records*).

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 hỏng hóc do 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.

Các thông tin trong redo log file chỉ được sử dụng để khôi phục lại database trong trường hợp hệ thống gặp sự cố và không cho phép viết trực tiếp dữ liệu trong database lên các datafiles trong database. Ví dụ: khi có sự cố xảy ra như mất điện bất chợt chẳng hạn, các dữ liệu trong bộ nhớ không thể ghi trực tiếp lên các datafiles và gây ra hiện tượng mất dữ liệu. Tuy nhiên, tất cả các dữ liệu bị mất này đều có thể khôi phục lại ngay khi database được mở trở lại. Việc này có thể thực hiện được thông qua việc sử dụng ngay chính các thông tin mới nhất có trong các redo log files thuộc datafiles. Oracle sẽ khôi phục lại các database cho đến thời điểm trước khi xảy ra sự cố.

Công việc khôi phục dữ liệu từ các redo log được gọi là *rolling forward*.

### Data file

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.
* Bên cạnh đó, 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.

Dữ liệu trong một datafile có thể đọc ra và lưu vào vùng nhớ bộ đệm của Oracle. Ví dụ: khi một user muốn truy cập dữ liệu trong một table thuộc database. Trong trường hợp thông tin yêu cầu không có trong cache memory hiện thời, nó sẽ được đọc trực tiếp từ các datafiles ra và lưu trữ vào trong bộ nhớ.

Tuy nhiên, việc bổ sung hay thêm mới dữ liệu vào database không nhất thiết phải ghi ngay vào các datafile. Các dữ liệu có thể tạm thời ghi vào bộ nhớ để giảm thiểu việc truy xuất tới bộ nhớ ngoài (ổ đĩa) làm tăng hiệu năng sử dụng hệ thống. Công việc ghi dữ liệu này được thực hiện bởi DBW*n* background process.

### Parameter file

Parameter file chỉ ra các tham số được sử dụng trong database. Người quản trị database có thể sửa đổi một vài thông tin có trong file này. Các tham số trong parameter file được viết ở dạng văn bản.

### Password file

Xác định quyền của từng user trong database. Cho phép người sử dụng khởi động và tắt một Oracle instance.

### Archiver log file

Là bản offline của các redo log files chứa các thông tin cần thiết để phục hồi dữ liệu.

### Alert log & trace file

Alert log: Chứa thông tin log trong quá trình Database hoạt động (khởi động, kết thúc db instance hoặc thay đổi tham số, cấu trúc db).

Trace file: Chứa thông tin khi các process gặp sự cố hoặc có event xảy ra.

### Backup file

Toàn bộ các file mà đã thực hiện backup.

# quản trị database instance

## Tham số khởi tạo

### Tổng quan

Tham số khởi tạo hay parameter file chứa các tham số giúp Oracle bật Database instance.

Oracle gồm 2 loại Parameter file:

* Pfile: Dạng text, read – only.
* Spfile: Dạng binary, có thể đọc và ghi (khuyến cáo sử dụng spfile)

Check parameter file: SQL> show parameter file 🡪 nếu value is not null 🡪 đang dùng spfile.

Spfile định dạng spfile<SID>.ora. Thường nằm trong đường dẫn: $ORACLE\_HOME/dbs;

Mặc định DB instance bật bằng spfile, để bật db instance bằng pfile:

SQL> startup pfile = ‘/tmp/pfile\_<SID>.ora’;

Tạo pfile từ spfile: SQL> create pfile = ‘/tmp/pfile\_<SID>.ora’ from spfile. Việc chuyển đổi này giúp dba có thể đọc được toàn bộ các tham số mà Database đang sử dụng.

Tạo spfile từ pfile: SQL> create spfile from = ‘/tmp/pfile\_<SID>.ora’;

### Loại

Gồm 2 loại:

* Tham số tĩnh: Phải restart db instance thì mới nhận tham số mới.
* Tham số động: Có thể thay đổi ngay trong khi db đang hoạt động

Xem mô tả về parameter: SELECT Name, **Value**, **Description** FROM v$parameter.

Xem tham số là động hãy tĩnh: SELECT Name, **issys\_modifiable** FROM v$parameter.

* False: tham số tĩnh.
* Immerdiate: tham số được cập nhật ngay lập tức.
* Deffer: tham số được cập nhật vào session tiếp theo.

Thay đổi giá trị tham số:

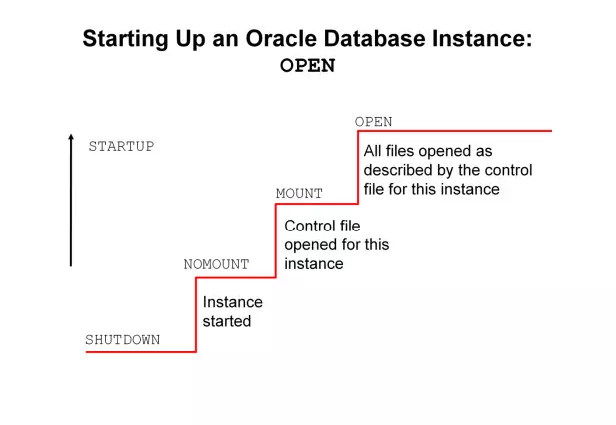
SQL> alter system set <Parameter\_name> = “value” scope = memory – cập nhật ngay lập tức và chỉ hiệu lực ở session hiện tại/ spfile – chỉ hiệu lực khi restart db / both – cập nhật ngay lập tức và đồng thời cập nhật ở spfile.

Lưu ý: Để thay đổi tham số tĩnh cần bật db instance ở spfile.

Câu hỏi: Làm sao để thay đổi giá trị ở trong pfile??? (4.3) Dùng lệnh vi của linux

## Bật, tắt Database instance

### Bật DB instance



Quy trình:

* Từ shutdown 🡪 Nomount: Oracle sẽ tìm file tham số cấu hình trong $ORACLE\_HOME/dbs (spfile<SID>.ora 🡪 spfile.ora 🡪 init<SID>.ora). Khi tìm thấy file tham số, oracle cấp phát tài nguyên theo file tham số, trường hợp file tham số lỗi hoặc tài nguyên không đủ 🡪 Database không thể chuyển lên Nomount.
* Từ Nomount 🡪 Mount: Oracle tìm và bật control file để chuyển lên trạng thái Mount.
* Từ Mount 🡪 Open: Tìm tất cả các datafile và online redo log file, đồng thời kiểm tra tính nhất quán giữa các file.

Bật database instance ở mode cụ thể và chuyển lên mode cao hơn

SQL> startup nomount;

SQL> alter database mount;

SQL> alter database open;

Lưu ý: Nếu bật ở mode thấp thì các mode sau cần bật lần lượt, không thể nhảy cóc đồng thời cũng không thể chuyển về mode thấp hơn.

### Tắt db instance

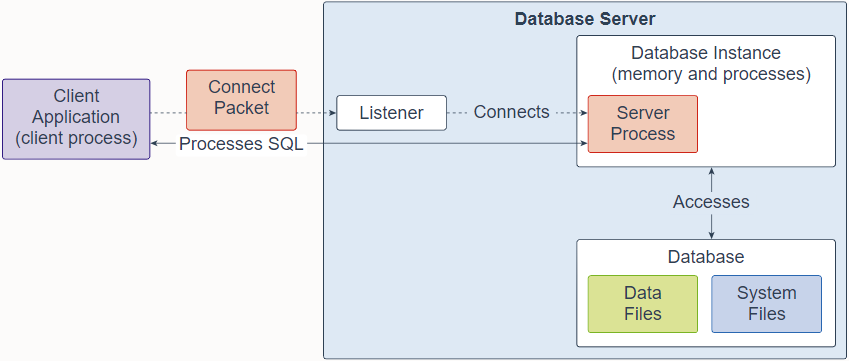


|  |  |  |
| --- | --- | --- |
| Cách thức | Quá trình tắt | Quá trình bật |
| Shutdown normal, transactional, immediate;  (Consisten database) | Uncommited changes rolled back, for immediate.  Database buffer cache written to data files.  Resoures released. | No instance recovery |
| Shutdown abort or Instance failure or startup force;  (Inconsisten database) | Modified buffer not written to datafiles.  Uncommited changes not rolled back. | Online redo log file used to reapply changes.  Undo segments used to roll back uncommited changes.  Resoures released. |

# cấu hình kết nối trong cơ sở dữ liệu

## Khái niệm

Với Listener, đây là một process của máy chủ cơ sở dữ liệu. Nó nhận các yêu cầu từ client, thiết lập kết nối với database instance và sau đó chuyển kết nối của client cho tiến trình máy chủ. Listener có thể chạy cục bộ trên máy chủ cơ sở dữ liệu hoặc chạy từ xa.



## Cấu hình

Đường dẫn mặc định: $ORACLE\_HOME/netword/admin/. Port mặc định: 1521.

Thêm, sửa, xóa Listener bằng giao diện dùng command: netca.

Để Listener có thể đăng ký db instance cần chỉnh sửa tham số local\_listener.

Có thể set nhiều Listener:

SQL> alter system set local\_listener = ‘Listener\_Name1’, ‘Listener\_Name2’ scope = spfile;

## Kết nối client đến server

Để máy client có thể kết nối đến Database trên server thì client cần gửi các thông tin sau đến Listener:

* Host hoặc địa chỉ ip nơi Listener đang hoạt động
* Port của Listener
* Giao thức Listener đang hoạt động
* Tên của service mà Database đăng ký với Listener

Cách thức:

* Easy connect: dùng chuỗi kết nối chứa toàn bộ thông tin

CMD: sqlplus username/password@host:port/SID

sqlplus [hr/hr@192.168.1.9:1521/lab](mailto:hr/hr@192.168.1.9:1521/lab)

* Local naming: Để sử dụng cách thức này máy client cần cài Oracle client. Truy cập: C:\app\client\admin\product\19.0.0\client\_1\network\admin\tnsnames.ora.

Tại đây thiết lập chuỗi kết nối đến Listener:

“LAB =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = **TCP**)(HOST = **192.168.1.9**)(PORT = **1521**))

(CONNECT\_DATA =

(SERVICE\_NAME = **lab**)

)

)”

Lưu ý: Có thể thiết lập nhiều chuỗi kết nối trong file tnsname.ora

## Kết nối giữa nhiều database

Để làm việc trên nhiều database trên 1 session cần:

* Cấu hình netword giữa 2 hệ thống (tnsname.ora)
* Tạo Database Link

CREATE DATABASE LINK<dblink \_name> CONNECT TO <user> IDENTIFIED BY <password> USING <connect\_string\_for\_remote\_db>;

Truy vấn đến object của db khác:

SQL> SELECT \* FROM employees@<dblink\_name>

# quản trị người dùng

## Đặc điểm

Mỗi user trong Oracle đều có:

* Một user\_name duy nhất
* Một phương thức xác thực
* Một default tablespace
* Một tempory tablespace
* Một user profile
* Một initial consumer group
* Một trạng thái sử dụng

Schema là tập hợp các object được sở hữu bởi Database user. Trong Oracle sau khi tạo user, schema cũng được tạo và có cùng tên với user.

## User quản trị

|  |  |  |
| --- | --- | --- |
| User | Nội dung | Quyền |
| SYS | Sử dụng để bật, tắt database intance  Sở hữu data dictionary và Automatic wordload repository (AWR) | Có quyền lớn nhất trong DB, thực hiện các hành động bật tắt db instacne. Tạo spfile hoặc chuyển chế độ Archive log  Được quyền xem và tương tác dữ liệu của các user khác |
| SYSTEM | Sở hữu một số table và view quản trị | Thực hiện các hành động bật tắt db instancce, tạo spfile hoặc chuyển chế độ Archive log |
| SYSBACKUP | Thực hiện các nhiệm vụ backup và recover sử dụng Oracle RMAN | Thực hiện các hoạt động Oracle recover manager (Rman) backup và recovery sử dụng RMAN hoặc sqlplus |
| SYSDG | Thực hiện các tác vụ của Oracle Data Guard | THực hiện các hoạt động quản trị Data Guard sử dụng Data Guard Broker hoặc DGMGRL |
| SYSKM | Thực hiện quản lý Transparent Data Encryption wallet | Thực hiện quản lý Transparent Data Encryption wallet operation |

## Quyền

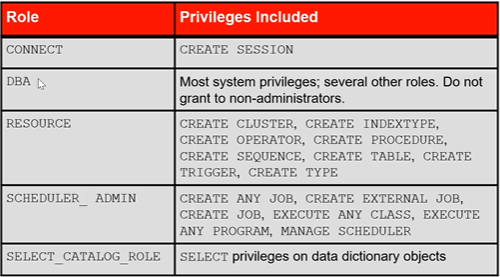
Có 2 loại quyền:

* System: Cho phép user thực hiện các hành động thay đổi mức Database.
  + CREATE SESSION: Để người dùng có thể kết nối vào Database
  + ALTER SYSTEM: User có thể chỉnh sửa các tham số cấu hình của db instance
  + SELET ANY TABLE: Quyền SELECT tất các các bảng trong DB
  + INSERT ANY TABLE, UPDATE ANY TABLE, DELETE ANY TABLE: Thực hiện các lệnh DML trên bất kỳ table nào
  + CREATE ANY TABLE: User có thể tạo bảng trên user khác
* Object: Cho phép user thực hiện các hành động trên các object xác định.
  + SELECT: tables, views, sequences, synonyms
  + INSERT: tables, views, synonyms
  + UPDATE: tables, views, synonyms
  + DELETE: tables, views, synonyms
  + ALTER: tables, synonyms
  + EXECUTE: procedures, functions, packages, synonyms

Quản trị viên thực hiện gán quyền và thu hồi quyền cho các User:

* Gán: Grant <danh sách các quyền> to user;
* Thu hồi: Revoke <danh sách các quyền> from user;

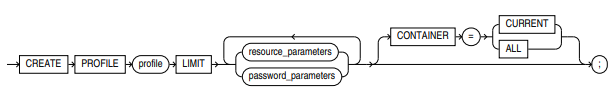
## Role



## Profiles

Chính sách giới hạn tài nguyên database, và quy định cho mật khẩu của user.

Câu lệnh:



Gán profiles:

resource\_parameters: Chỉ áp dụng khi tham số RESOURCE\_LIMIT = TRUE

* SESSION\_PER\_USER
* CPU\_PER\_SESSION
* CPU\_PER\_CALL
* LOGICAL\_READS\_PER\_SESSION: số lượng block có thể được đọc bởi một session
* LOGICAL\_READS\_PER\_CALL: số lượng block có thể đọc được bởi 1 câu lệnh
* PRIATE\_SGA: Nếu session kết nối sử dụng kiến trúc shared serer, tham số này giiowis hạn số lượng Kb mà session
* CONNECT\_TIME: số thời gain kết nối tối đa của 1 session
* IDLE\_TIME: Thời gian tối đa k hoạt động của một session
* COMPOSITE\_LIMIT: tổng tài nguyên tiêu tốn của session

password\_parameters:

* FAILER\_LOGIN\_ATTEMPTS: giới hạn số lần nhập sai mk trước khi acc bị lock
* PASSSWORD\_LOCK\_TIME: số ngày sẽ lock acc nếu vi phạm chính sách Fail\_login\_attempts
* **PASSWORD\_LIFE\_TIME**: số ngày mà mk của user sẽ bị hết hạn, user cần đổi mk
* PASSWORD\_GRACE\_TIME: số ngày gia hạn sau khi tham số Password\_life\_time bị bi phạm
* PASSWORD\_REUSE\_TIME: số ngày trước khi mk có thể được tái sử dụng
* PASSWORD\_REUSE\_MAX: số lần mk có thể tái sử dụng
* PASSWORD\_VERIFY\_FUNTION: hàm kiểm tra mk của user

# Quản lý lưu trữ trong cơ sở dữ liệu oracle

## Tablespace và Quota

### Khái niệm

Tablespace là vùng tập hợp nhiều datafile chứa dữ liệu trong database. Tablespace có những đặc điểm:

* Một tablespace chỉ thuộc một database.
* Mỗi tablespace có thể chứa một hay nhiều data file thuộc hệ điều hành.
* Tablespaces có thể đặt ở trạng thái online hay offline trong lúc database đang chạy.
* Ngoại trừ tablespace SYSTEM hay tablespace chứa rollback segments đang có trạng thái ACTIVE, các tablespaces đều có thể chuyển về trạng thái offline trong lúc database đang chạy.
* Các tablespaces cũng có thể chuyển đổi trạng thái read-write hay read-only.

Mặc định các bảng được tạo ra nếu không chỉ định sẽ được đưa vào tablespace Users

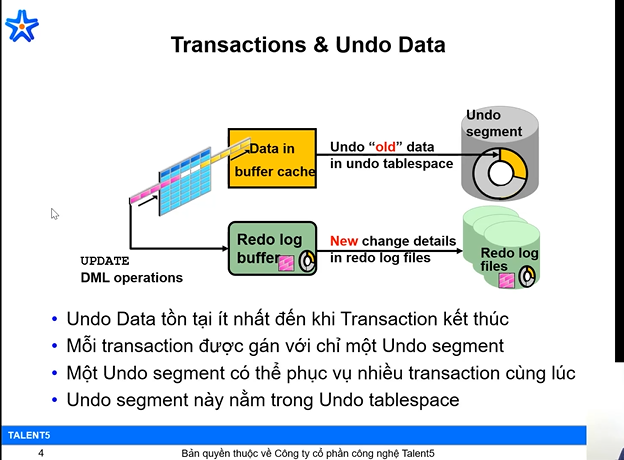
Quota là dung lượng cấp phát cho User để làm việc trên tablespace. Gán Quota cho user:

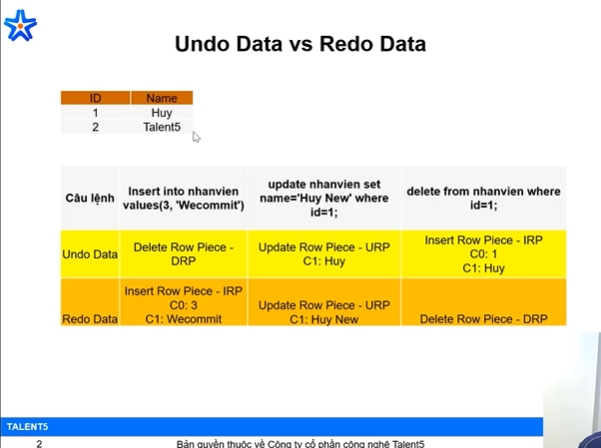
SQL> alter user <user\_nanme> quota <dung lượng M> on <tablespace>;

Tablespace mặc định:

* SYSTEM: Tablespace của hệ thống. Chứa thông tin về data dictionary của hệ thống
* SYSAUX: Giảm tải cho SYSTEM tablespace. Lưu thông tin AWR
* TEMP
* UNDO: Lưu rollback segment
* USER: Chứa dữ liệu và các object. Nếu người dùng không chị định Default Tablespace thì dữ liệu sẽ được ghi vào Tablespace users

# quản lý undo





Các loại undo:

* Active: lưu trữ các thông tin của những transaction chưa commit. Không cho phép ghi đè.
* Unexpire: lưu trữ thông tin các transaction đã commit. Sẽ lưu trữ thêm một khoảng thời gian được xác định bằng tham số retention.
* Expire: Dữ liệu không còn cần thiết nữa. Có thể ghi đè bởi các active transaction.
* Lệnh check: select status, round((SUM(bytes) / 1024 / 1024), 2) MB from dba\_undo\_extents group by status.

Các lỗi thường gặp khi làm việc với undo

* ORA – 01555: snapshot too old
  + Giả sử 8h có session thực hiện update dữ liệu bảng nhân viên, và thực hiện từ 8h – 9h mới xong 🡪 undo active.
  + 8h10 có 1 session khác thực hiện một báo cáo cần sử dụng dữ liệu từ bảng nhanvien
* ORA – 30036: unable to extend segment by 8 in undo tablespace “UNDOTBS”.
  + Lỗi đầy tablespace do active undo cứ nở dần.

Tempory Undo (Có từ 12c):

* Nếu không bật (mặc định): Các câu lệnh DML đối với tempory table được coi như làm việc trên bảng thật 🡪 Permanent undo (lưu trong undo tablespace) 🡪 redo (luu trong redo log files)
* Nếu bật: Các câu lệnh DML sẽ được lưu trong tempory undo (tempory tablesplace) 🡪 giảm lượng undo cần lưu trữ trong undo tablespace 🡪 giảm dung lượng redolog sinh ra 🡪 có thể sử dụng được các câu lệnh DML trên tempory table tại hệ thống Physical standby (sử dụng Oracle Active Data Guard).

# quản lý tranh chấp khi xử lý dữ liệu (data concurrency)

## Cơ chế lock

### Tổng quan

Cơ chế tự động của Oracle nhằm ngăn chặn nhiều session thay đổi cùng dữ liệu tại cùng thời điểm. Cụ thể khi thực hiện lệnh DML, Oracle sẽ lock bản ghi (row – level) 🡪 các session khác không thể tương tác đến các bản ghi đấy.

Việc quản lý queue lock được diễn ra tự động. Lock sẽ tồn tại cho đến khi transaction commit hoặc rollback.

Khi 1 transaction thực hiện lệnh DML mà không kết thúc, 1 transaction khác trong cùng thời điểm đấy cũng thực hiện lệnh DML và tương tác đến những bản ghi mà transaction đầu đang lock thì gọi là lock confict.

### Kill session

Lệnh: SQL> alter syster kill session <‘SID,Serial#’> immediate;

Check session locked and locking:

select --transaction is locked

t1.SID SID1, t1.SERIAL# SERIAL#1, t1.Username Username1, t1.SQL\_ID SQL\_ID1,

s1.SQL\_TEXT SQL\_TEXT1,

--transaction is locking

t2.SID SID2, t2.SERIAL# SERIAL#2, t2.Username Username2, t2.PREV\_SQL\_ID SQL\_ID2,

s2.SQL\_TEXT SQL\_TEXT2

from v$session t1

left join v$sql s1 on s1.SQL\_ID = t1.SQL\_ID

left join v$session t2 on t2.SID = t1.Blocking\_session

left join v$sql s2 on s2.SQL\_ID = t2.PREV\_SQL\_ID

where t1.blocking\_session is not null;

# triển khai oracle database auditing

## Khái niệm

Audit là việc giám sát và ghi lại những hành động trong CSDL.

* Có thể Audit trên những câu lệnh SQL
* Có thể Audit trên tập hợp nhiều điều kiện (user đăng nhập, ứng dụng kết nối, câu lệnh nào, …)
* Các hành động có thể Audit
  + Những hành động thực hiện thành công
  + Những hành động không thực hiện thành công (user k có quyền nhưng cố tình truy cập)

Unified Auditing:

* Các thông tin audit được lưu trữ tập trung trong unified audit trail
* Unified audit trail nằm trong một read – only table thuộc AUDSYS schema và chứa trong SYSAUX tablespace
* Tính năng
  + Không cần thiết lập nhiều tham số Audit như các phiên bản trước
  + Tất cả các thông tin audit được lưu trữ trong cùng một định dạng, cùng một nơi, giúp việc quản trị dễ dàng hơn
  + Hiệu năng chung của audit tốt hơn các phiên bản trước
  + Check: SELECT VALUE FROM V$OPTION WHERE PARAMETER = ‘UNIFIED AUDITING’ (TRUE)

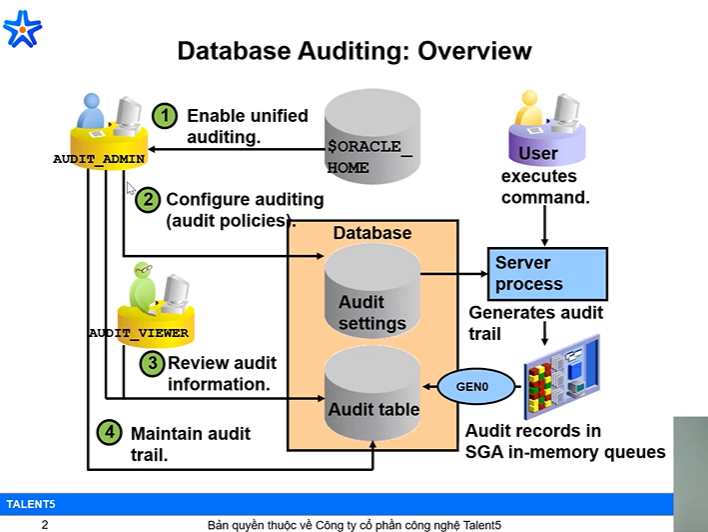
Cách triển khai

* Mixed mode auditing: đối với các phiên bản cũ khi nâng lên 12c sẽ mặc định ở dạng này để giữ các tham số như cũ.
* Unified auditing: Để thực hiện cần thao tác link oracle binary với uniaud\_on và yêu cầu restart database.

Ai có thể triển khai auditing:

* AUDIT\_ADMIN\_Role: Sử dụng để tạo các chính sách audit. Quản lý audit trail
* AUDIT\_VIEWER\_Role: Sử dụng để xem thông tin audit

Kiến trúc:



## Bật Unified auditing

Cách thức:

* Kiểm tra: SELECT VALUE FROM V$OPTION WHERE PARAMETER = ‘Unified Auditing’ -> TRUE
* Nếu kết quả trả về False 🡪 Tắt db instance and listener;
* Cd /$ORACLE\_HOME/rdbms/lib
* Make -f ín\_rdbms.mk uniaud\_on ioracle ORACLE\_HOME=$ORACLE\_HOME
* Bật db instance and listener

## Triển khai unified audit policy

Cách thức:

* Tạo policy
  + Tạo policy trên object: CREATE AUDIT POLICY <audit\_name> ACTIONS <SELECT, UPDATE, LOCK, EXECUTE, …, ALL> ON <table\_name>
  + Tạo policy mức system:

CREATE AUDIT POLICY <audit\_name> PRIVILEGES SELECT ANY TABLE, CREATE LIBRARY

CREATE AUDIT POLICY <audit\_name> ACTIONS AUDIT, ALTER TRIGGER

CREATE AUDIT POLICY <audit\_name> ROLES <role\_name>

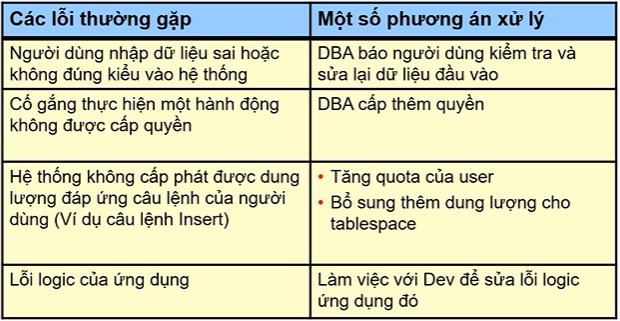
OR mix many audit

* + Tạo policy trên các đk cụ thể:
    - PER SESSION
    - PER STATEMENT
    - PER INSTANCE
* Bật policy đã được thiết lập: AUDIT POLICY <audit\_name> (BY user/ EXCEPT user/ WHENEVER (NOT) SUCCESSFUL).
* Thay đổi policy: ALTER AUDIT POLICY <audit\_name> ADD …
* Disable policy: NOAUDIT
* Các audit đã khởi tạo: SELECT \* FROM audit\_unified\_policies
* Query những lần audit chạy: SELECT \* FROM *unified\_audit\_trail*

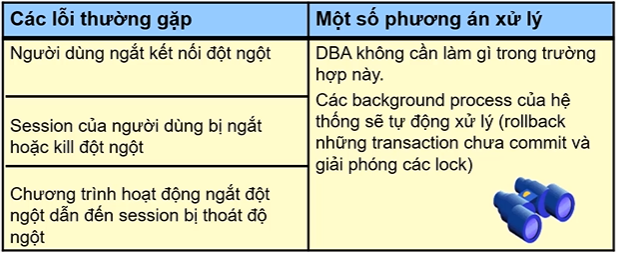
# sao lưu và khôi phục dữ liệu

## Các lỗi có thể gặp

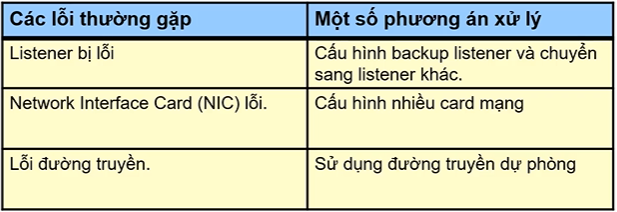
### Lỗi câu lệnh



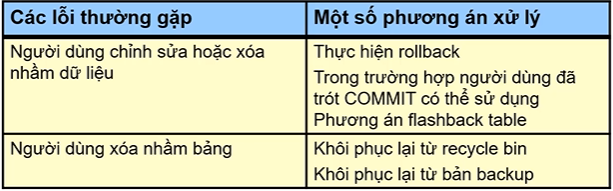
### Lỗi các tiến trình người dùng



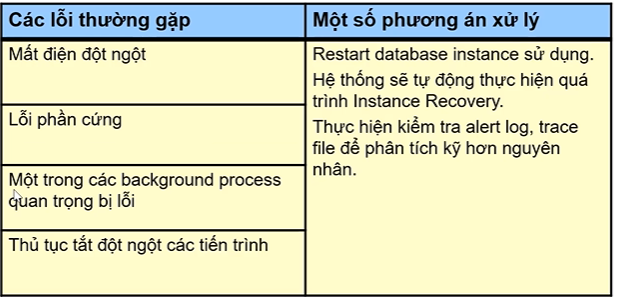
### Lỗi kết nối



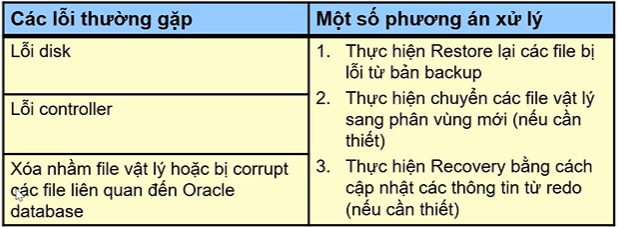
### Lỗi người dùng



### Lỗi database instance

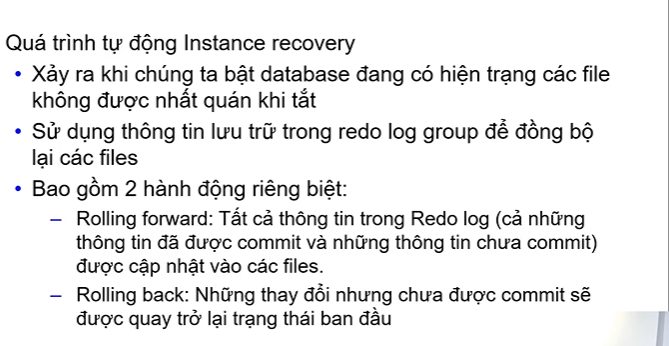


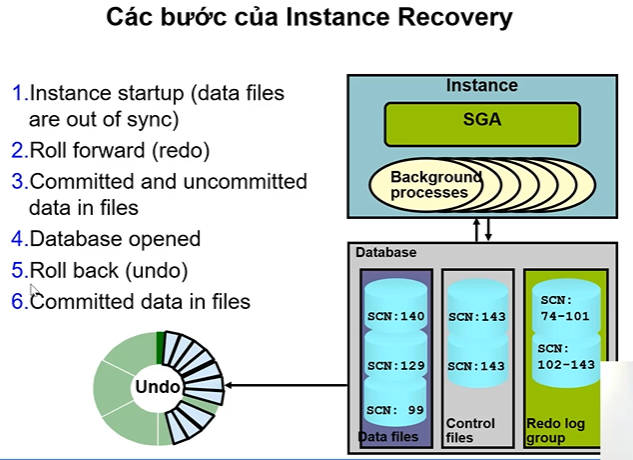
### Lỗi phần cứng



## Instance recovery

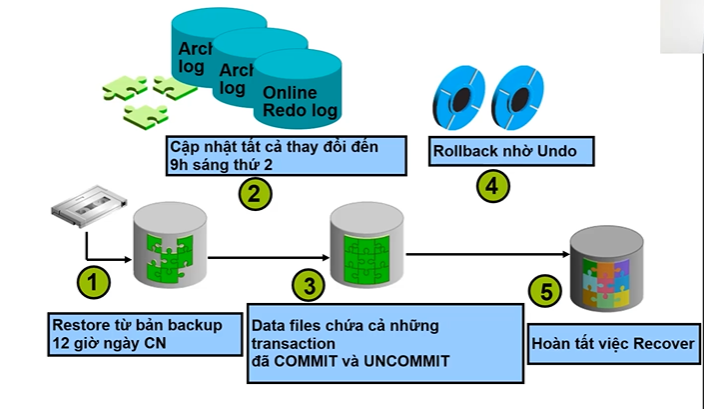
### Tổng quan



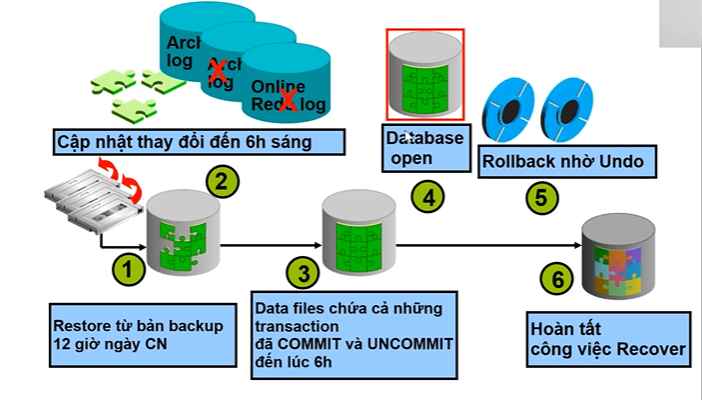


### So sánh complete & Incomplete recovery

Complete:



Incomplete:



## Flashback table

### Khái niệm

Flashback Table là kỹ thuật đưa dữ liệu của 1 bảng quay lại 1 thời điểm cụ thể trong quá khứ. Kỹ thuật này thường được sử dụng khi người dùng hay ứng dụng lỡ thay đổi hay xoá nhầm dữ liệu.

### Một số chú ý quan trọng khi sử dụng Flashback

* Không flashback được những đối tượng thuộc schema SYS.
* Nếu bảng cần flashback có bảng con tham chiếu đến, cần flashback bảng con trước.
* Nếu những hành động sau diễn ra sau thời điểm bạn cần flashback, thao tác flashback sẽ không thực hiện được:
  + Upgrading, moving, truncating table,
  + Thêm constraint vào table, thêm table vào cluster
  + Thay đổi hay xoá cột trong bảng
  + Thay đổi column encryption key
  + Add, drop, merg, split, coales, hay truncate partition hay subpartition (trừ việc thêm partition kiểu range).
* Các đối tượng dependant với table (indexes, trigger) sẽ quay lại nhưng tên sẽ giữ nguyên như lúc ở recyclebin. Cần thay đổi lại tên sau khi flashback.

### Các bước thực hiện

* Kiểm tra đã enable flashback log chưa: SQL> select flashback\_on from v$database;
* Nếu *false* thì enable: SQL> alter database flashback on;
* Đặt thời gian lưu flashback: SQL> alter system set db\_flashback\_retention\_target =2880; (mặc định 1440p)
* Kiểm tra SCN or thời gian hiện tại:

SQL> select to\_char(current\_scn) from v$database;

SQL> select to\_char(sys\_date, ‘dd-MON-YYYY hh24.MI.SS’) from dual;

* Thực hiện xóa bảng và commit: SQL> delete <table\_name>; commit;
* Enable row movement: SQL> alter table <table\_name> enable row movement;
* Flashback to scn: SQL> flashback table <table\_name> to scn <scn\_number>;
* Flashback to timestamp:

SQL> flashback table <table\_name> to timestamp to\_timestamp(‘<time>’, ‘dd-MON-YYYY hh24.MI.SS’);

* Disable row movement: SQL> alter table <table\_name> disable row movement;

## Fash recovery area (FRA)

Mục đích: thiết lập 1 folder mặc định để khi backup file thì file sẽ được đưa vào thư mục chỉ định.

Check: SQL> show parameter recovery

* Db\_recovery\_file\_dest\_size: dung lượng giới hạn (thiết lập trước)
* Db\_recovery\_file\_dest: đường dẫn thiết lập thư mục

## Multiplex control file

Mục đích: thiết lập nhiều control file để nếu 1 control file bị lỗi thì hệ thống vẫn hoạt động được. Khuyến cáo: hệ thống cần tối thiểu 2 control file ở các nơi khác nhau.

Check: SQL> show parameter control\_files;

Thiết lập:

* Khai báo path cho nhiều control file: alter system set control\_files = ‘path1’,’path2’, … scope = spfile;
* Shutdown immediate;
* Tạo control file mới theo control file ban đầu đúng như path đã khai báo: cp controlfile.ctl (ctl hiện có) controlfilen.ctl (ctl mới tạo)

## Multiplex redo log file

Mục đích: Thiết lập nhiều redo log file để phòng tránh file gặp sự cố. Khuyến cáo mỗi group cần 2 member ở các nơi khác nhau.

Check: SQL> select \* from v$log (logfile)

Thiết lập:

* Đã thiết lập FRA
* Sử dụng toad: Database 🡪 admistrater 🡪 redo log manager
* Add group or add member
* Restart dbf

## Cấu hình archivelog mode

Thực hiện:

* Check: SQL>archive log lis, nếu db log mode hiển thị No archive mode 🡪 chưa thiết lập
* Shutdown và bật ở chế độ mount
* SQL> alter database archivelog;
* SQL> alter database open;
* Tạo archive log ngay lập tức: SQL> alter system archive log current;
* Để quản trị tập trung -> thiết lập archive log file vào thư mục FRA:

SQL> alter system set log\_archive\_dest\_1=’location=use\_db\_recovery\_file\_dest’;

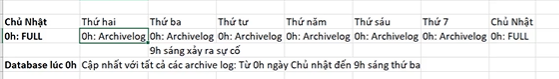
## Sao lưu CSDL

### Chiến lược

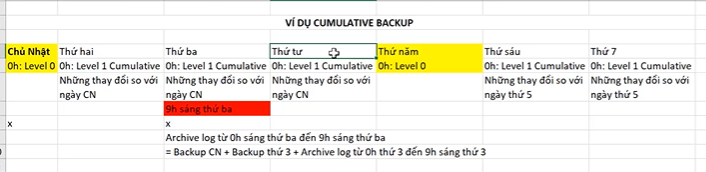
* Sao lưu toàn bộ CSDL: Tất cả data file và control file
* Sao lưu 1 phần: một hoặc nhiều tablespace, datafile, có hoặc k có controlfile

### Kiểu backup

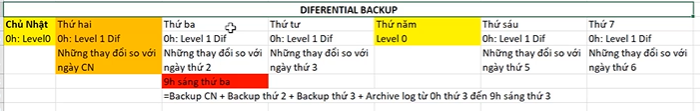
* Backup full: đi kèm backup archivelog



* Backup incremental
  + Incremental 0
  + Incremental 1
    - Cumulative



* + - Differential



### Trạng thái backup

* Hot backup: thực hiện khi db đang hoạt động
* Cold backup: thực hiện khi db không hoạt động

### Định dạng

* Backup set
  + Là tập các file binary chưa: datafie, archive log, control file, parameter file
  + Dung lượng nhỏ hơn do không backup những block trống
  + Thường sẽ sử dụng giải thuật nén giúp tiết kiệm không gian lưu trữ
  + Nơi lưu trữ: có thể backup ra disk hoặc ra tape
* Image copies:
  + Sử dụng cho datafile, archive logs
  + Dung lượng giống hệt với các file cần backup
  + Nhanh chóng trong việc khôi phục
  + Nơi lưu trữ: chỉ có thể đưa trực tiếp ra disk.

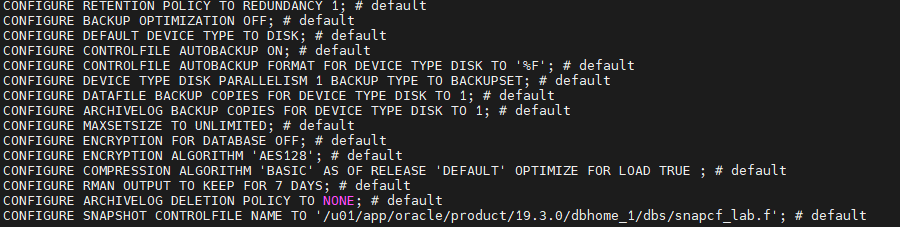
### Thực hiện

* Rman target / : đăng nhập vào db hiện tại
* Show all: tìm hiểu các tham số???
  + **CONFIGURE RETENTION POLICY TO REDUNDANCY 1**: số bản full backup or backup level0 của mỗi datafile và controlfile được lưu trữ. Nếu vượt quá thì các bản cũ được coi là bản cũ, chúng sẽ bị xóa sau khi chạy: RMAN> DELETE OBSOLETE
  + **CONFIGURE BACKUP OPTIMIZATION OFF**: Nếu để ON, rman bỏ qua việc sao lưu file giống hệt với file đã được sao lưu. RMAN xác định file có giống hệt file đã được sao lưu hay không qua tiêu chí:

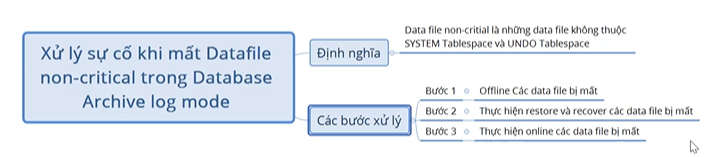
|  |  |
| --- | --- |
| Type | Criteria |
| Data file | The data file must have the same dbid, SCN, creation SCN, and RESETLOGS SCN and time as a data file in a backup |
| Archive log file | Same DBID, thread, sequence number and RESETLOGS SCN and time |
| Backup set | Same backup set recid and stamp |

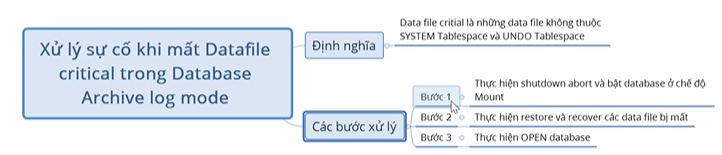
* + **CONFIGURE DEFAULT DEVICE TYPE TO DISK**: chỉ định vùng lưu trữ các bản sao lưu, mặc định được lưu trong FRA
  + **CONFIGURE CONTROLFILE AUTOBACKUP ON**: tự động sao lưu control file và parameter file sau khi lệnh backup thực hiện thành công. Nếu db bật archivelog mode thì rman sẽ tự động sao lưu controlfile khi nó bị thay đổi
  + **CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO ‘%F’**: định dạng thư mục chứa backup file. %F 🡪 …/DBID/Date/id file backup.  
    
  + **CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO BACKUPSET**: xác định số lượng bản backup đọc ghi song song
  + **CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO** **n** (n>=1): chỉ định số lượng bản backup. Có thể chỉ định n nơi lưu trữ khác nhau:  
    BACKUP DEVICE TYPE DISK COPIES 3 DATAFILE 7 FORMAT  
    ‘/tmp/%U’,’?/oradata/%U’,’?/%U’
  + **CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 1**: tương tự phần trên
  + **CONFIGURE MAXSETSIZE TO UNLIMITED**: giới hạn dung lượng bản backup. Nếu vượt quá sẽ báo lỗi.
  + **CONFIGURE ENCRYPTION FOR DATABASE OFF**: có mã hóa bản backup không
  + **CONFIGURE ENCRYPTION ALGORITHM ‘AES128’**: mật khẩu bản backup nếu có mã hóa.
  + **CONFIGURE COMPRESSION ALGORITHM ‘BASIC’ AS OF RELEASE ‘DEFAULT’ OPTIMIZE FOR LOAD TRUE**: xác định kiểu nén, có 2 kiểu là ZLIB và BZIP2. ZLIB dùng cpu thấp nhưng tốc độ chậm, BZIP2 ngược lại:

**CONFIGURE COMPRESSION ALGORITHM ‘ZLIB’**;

* + **CONFIGURE RMAN OUTPUT TO KEEP FOR 7 DAYS**: thời hạn lưu trữ bản backup??
  + **CONFIGURE ARCHIVELOG DELETION POLICY TO NONE**: chỉ định chính sách xóa archive log.
  + **CONFIGURE SNAPSHOT CONTROLFILE NAME TO ‘…PATH…’**: RMAN cần 1 bản snapshot khi đồng bộ hóa với recovery catalog hoặc khi backup control file hiện tại. Đường dẫn mặc định ở $ORACLE\_HOME/dbs
  + 
  + https://taliphakanozturken.wordpress.com/tag/controlfile-autobackup/
* Backup full: backup database;
* Backup 1 phần: backup datafile 8,9, …

## Khôi phục CSDL







# Các phương án dịch chuyển dữ liệu

## Datapump

### Thực hiện

* Export dữ liệu: expdp
  + SQL> create directory <dir\_name> as <dir\_path> (mục đích: dir\_name trỏ đến đường dẫn dir\_path)
  + OS$: expdb system/oracle directory = <dir\_name> tables=<schema.table\_name> dumpfile = <dump\_name.dmp> logfile = <logfile\_name.log>
* Import dữ liệu: impdp
  + OS$: impdb system/oracle directory=<dir\_name> dumpfile=<dump\_name.dmp> logfile=<logfile.log> full=y

### Một số tính năng và lợi ích

* Các tham số parallel giúp tăng tốc độ thực hiện
* Sử dụng để chuyển đổi chéo giữa các môi trường: chéo OS, chéo db version
* Nén dữ liệu giúp giảm dung lượng file xuất ra
* Có thể sử dụng qua db link
* Có thể đổi tên (REMAP) trong quá trình thực hiện: table, tablespace, user

# bảo trì cơ sở dữ liệu

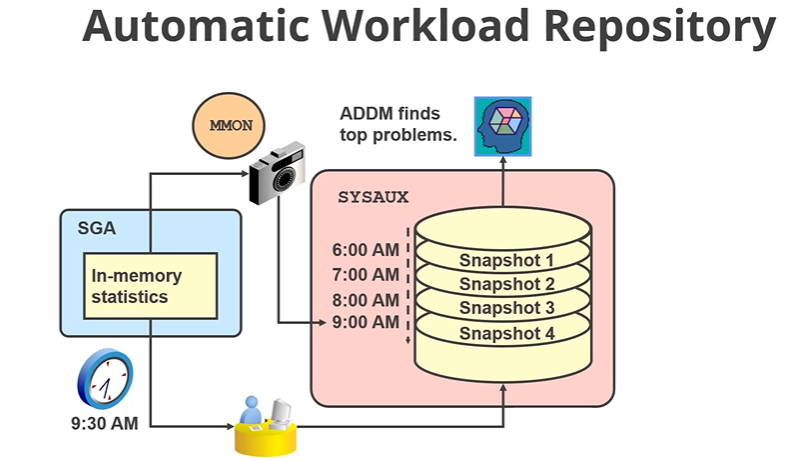
## Tổng quan

Trong quá trình DB hoạt động, toàn bộ thông tin thông số memory sẽ lưu trữ trong SQA, định kỳ MMON process sẽ chụp lại toàn bộ thông số hoạt động của SGA 1 lần / 1h. Toàn bộ thông tin này gọi là AWR.

Oracle xây dựng tính năng gọi là ADDM – mỗi khi AWR sinh ra, ADDM sẽ tự động đọc các chỉ số và đưa ra các phương án giúp tăng hiệu năng của hệ thống.

Mặc định AWR:

* Mặc định chỉ lưu trữ thông tin AWR trong 8 ngày. Đối với các hệ thống quan trọng nên chỉnh lại phần lưu trữ này.
* Mặc định khoảng thời gian chụp là 60 phút. Lưu ý không thể lấy thông tin AWR trong khoảng thời gian DB instance bị restart.
* Các thông tin thu thập dựa vào việc cấu hình tham số: STATISTICS\_LEVEL
  + BASIC: disable các chức năng ADDM
  + TYPICAL: giá trị default – nên chọn tham số này
  + ALL: bổ sung thêm những thông tin chi tiết hơn về SQL tuning



## Tạo AWR

* Đăng nhập sysdba
* SQL> @?/rdbms/admin/awrrpt.sql
* Chọn định dạng 🡪 chọn thông tin muốn lấy từ mấy ngày trước 🡪 chọn thời gian theo Id 🡪 đặt tên (enter – mặc định)
* Copy ra máy local để xem

## ADDM

* Thực hiện sau mỗi lần AWR snapshot
* Giám sát instance, phát hiện các điểm nghẽn
* Kết quả lưu trong AWR
* Cách tạo: SQL>@?/rdbms/admin/addmrpt.sql

# quản lý hiệu năng cơ sở dữ liệu

