# TÀI LIỆU THỰC HÀNH KHÓA HỌC OAZ

*(Lưu hành nội bộ)*

*Copyright Trần Văn Bình Master – Founder & Chairman “DBA Việt Nam”*

*(Mong rằng các em đây sẽ là tài liệu giúp chắp cánh vươn xa các em)*

Lịch sử cập nhật:

|  |  |
| --- | --- |
| **Ngày** | **Nội dung cập nhật** |
| 04/11/2023 | Bản v1 |
|  |  |
|  |  |

# PRE-TRAINING:

## VÀO HỌC TRƯỚC CÁC KHÓA HỌC TRÊN UNICA:

\* Workshop 1: <https://bit.ly/2AhpUeY>

\* Linux 21h: <https://bit.ly/31wwolr>

(Tạo account sau đó vào lại các link trên)

## HỌC ORACLE DATABASE CƠ BNAR TỪ A-Z  (ORACLE DB TUTORIAL)

### [Kiến trúc Cơ sở dữ liệu Oracle hoạt động như thế nào Khi thực hiện các thao tác SELECT, INSERT, UPDATE, DELETE ?](https://www.tranvanbinh.vn/2021/02/kien-truc-co-so-du-lieu-oracle-hoat-ong.html)

### [BÀI 1: CÁC ĐIỂM MỚI TRONG ORACLE](https://www.tranvanbinh.vn/2020/02/chuong-1-cac-iem-moi-trong-oracle-11g.html)

### [BÀI 2: CÁC THÀNH PHẦN KIẾN TRÚC](https://www.tranvanbinh.vn/2020/02/chuong-2-cac-thanh-phan-kien-truc.html)

### [BÀI 3: CÁC CÔNG CỤ QUẢN TRỊ ORACLE](https://www.tranvanbinh.vn/2020/02/tutorial-dba-co-ban-chuong-3-cac-cong.html)

### [BÀI 4: TẠO DATABASE](https://www.tranvanbinh.vn/2020/02/tutorial-dba-co-ban-chuong-4-tao.html)

### [BÀI 5: QUẢN TRỊ ORACLE DATABASE](https://www.tranvanbinh.vn/2020/02/tutorial-dba-co-ban-chuong-5-quan-tri.html)

### [BÀI 6: DATA DICTIONARY, VIEWS VÀ PACKAGES](https://www.tranvanbinh.vn/2020/02/tutorial-dba-co-ban-chuong-6-data.html)

### [BÀI 7: QUẢN TRỊ CONTROL FILES](https://www.tranvanbinh.vn/2020/06/hoc-oracle-db-tu-z-bai-7-quan-tri.html)

### [BÀI 8: QUẢN LÝ REDO LOG FILES](https://www.tranvanbinh.vn/2020/02/tutorial-dba-chuong-8-quan-ly-redo-log.html)

### [BÀI 9: QUẢN TRỊ TABLESPACES VÀ DATA FILES](https://www.tranvanbinh.vn/2020/06/hoc-oracle-db-tu-z-bai-9-quan-tri.html)

### [BÀI 10: CẤU TRÚC LƯU TRỮ](https://www.tranvanbinh.vn/2020/06/hoc-oracle-db-tu-z-bai-10-cau-truc-luu.html)

### [BÀI 11: QUẢN LÝ ROLLBACK SEGMENTS](https://www.tranvanbinh.vn/2020/06/hoc-oracle-db-tu-z-bai-11-quan-ly.html)

### [BÀI 12: QUẢN LÝ TEMPORARY SEGMENTS](https://www.tranvanbinh.vn/2020/02/tutorial-dba-co-ban-chuong-12-quan-ly.html)

### [BÀI 13: CLUSTERS VÀ INDEX-ORGANIZED TABLES](https://www.tranvanbinh.vn/2020/06/hoc-oracle-db-tu-z-bai-13-clusters-va.html)

### [BÀI 14: QUẢN LÝ CÁC TABLES](https://www.tranvanbinh.vn/2020/02/tutorial-dba-co-ban-chuong-14-quan-ly.html)

### [BÀI 15: QUẢN LÝ CÁC INDEXES](https://www.tranvanbinh.vn/2020/02/tutorial-dba-co-ban-chuong-15-quan-ly.html)

### [BÀI 16: NẠP VÀ TỔ CHỨC LƯU TRỮ DỮ LIỆU](https://www.tranvanbinh.vn/2020/02/tutorial-dba-co-ban-chuong-16-nap-va-to.html)

### [BÀI 17: QUẢN LÝ USER](https://www.tranvanbinh.vn/2020/02/tutorial-dba-co-ban-chuong-17-quan-ly.html)

### [BÀI 18: QUẢN LÝ THÔNG TIN PROFILES](https://www.tranvanbinh.vn/2020/02/tutorial-dba-co-ban-chuong-18-quan-ly.html)

### [BÀI 19: CÁC QUYỀN HỆ THỐNG](https://www.tranvanbinh.vn/2020/02/tutorial-dba-co-ban-chuong-19-cac-quyen.html)

### [BÀI 20: QUẢN LÝ CHỨC DANH (ROLE)](https://www.tranvanbinh.vn/2020/06/hoc-oracle-db-tu-z-bai-20-quan-ly-chuc.html)

### [BÀI 21. TÍNH NĂNG HỖ TRỢ NGÔN NGỮ QUỐC GIA](https://www.tranvanbinh.vn/2020/06/hoc-oracle-db-tu-z-bai-21-tinh-nang-ho.html)

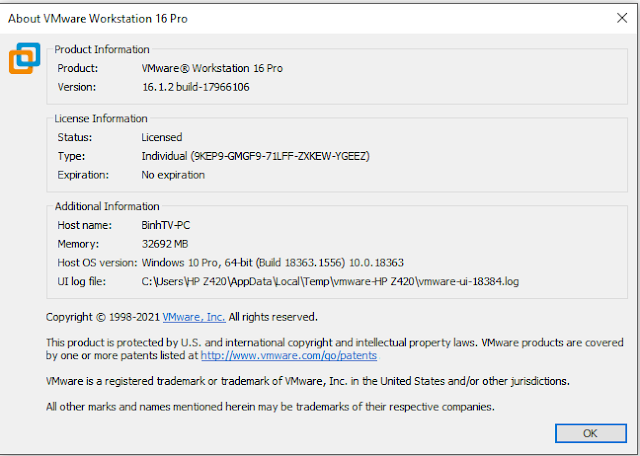
# DỰNG LAB

## Dựng Oracle Database 19c trên máy ảo VMWARE

<https://www.tranvanbinh.vn/2021/10/huong-dan-dung-may-ao-lab-oracle.html>

**1. Cài đặt VMWare 16**<https://drive.google.com/file/d/1Oj-xbLQUhPPRe37UeQfWkjTaHvt4SaMB/view?usp=sharing>

Check phiên bản đảm bảo cài đặt đúng bản 16.1.2

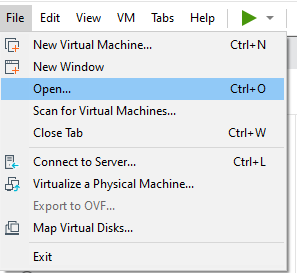
[](https://blogger.googleusercontent.com/img/a/AVvXsEhNImc2swP7c67NM7xvuPXGuEhFpI2pLaw4o60fI9GOTWmiSzayxKkC4l69IxjFpiRZ9FX6SC42Sdvg1sHRx4vSlvxWs9zA7pkm3jJUHHWl_XRNl8X0tHKleDdLu75LP1UI6ASuwdzFLQKdhisbtowcjz3jELG9l2vfqrmjz67sXZ4XxJZPHaeO3uFN)

**2. Down file lab dựng sẵn tại link**<https://drive.google.com/file/d/1HWH6q9fe34ZL2RKGA4EBhIZz1B12mWd6/view?usp=sharing>

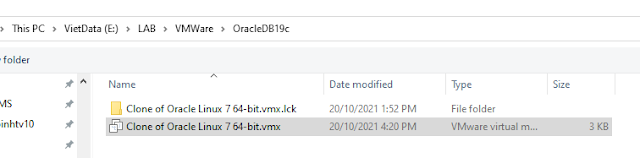
Sau đó giải nén, thư mục sau giải nén là OracleDB19c

**3**. **Mở file vmware 16**

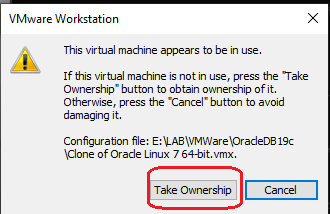
**3.1. Open máy ảo***Vào File --> Open*

[](https://blogger.googleusercontent.com/img/a/AVvXsEgvcQ5OL4IIMQi3lfbQRCV6Cqga8d25_fPBzIBNRQzk8aY9K6O2vL-5VJJiif2bFvLBS9xYv5j1ms2TcaqbXNshnj06PAe-U0Dijumc_gqt9SSb1P5cxlu7r4lHJ1ALuCXJDSPNccY0rY2nTBWqKAe0VXkJIW1Jxy89mjbVMfql-xs7x33_q6xD7VPg)

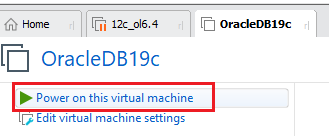
Tìm thư mục chứa file OracleDB19c vừa giải nén và chọn

[](https://blogger.googleusercontent.com/img/a/AVvXsEivcAvDm4X2SZ5LFxXiZU8PRP2vLXcr5CuTBcddnLaz3v5XJT6MJKVtxpKH5A3mVU_jWt21BtTRMWSbH2P621zUTLDjDCE1CSblxhwVf1mWH38LOvzqOQwUy1eXgjt9CbsbaITY3ZfI5BnbHk7QGCPuaJ3oqkn1ZIGquX6gr-KrDX-dGxOe7TSqiKes)

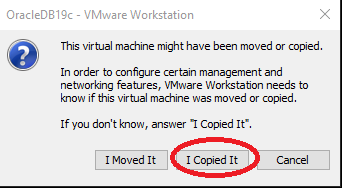
Chọn Take Ownership

[](https://blogger.googleusercontent.com/img/a/AVvXsEimxqlvMrafXCbGbIPqQFLDVn_kBa0qCS9VSu-pwOiqpTqO4lyMXxeOHj1aQ6FlD1dNtkfifyKA4TkNF9MIxNji95wtjYnRjrCOwfoVjOKuSWI5KyoMznwhEafENql-c4q5X3gSEq3NT30qq6fjjrijSWj7DfDKQTgE9hE3vxs64g6NNOUhxDEc02zK)

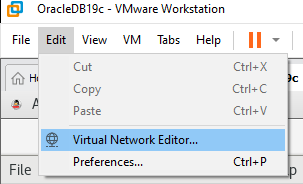
Start máy ảo:

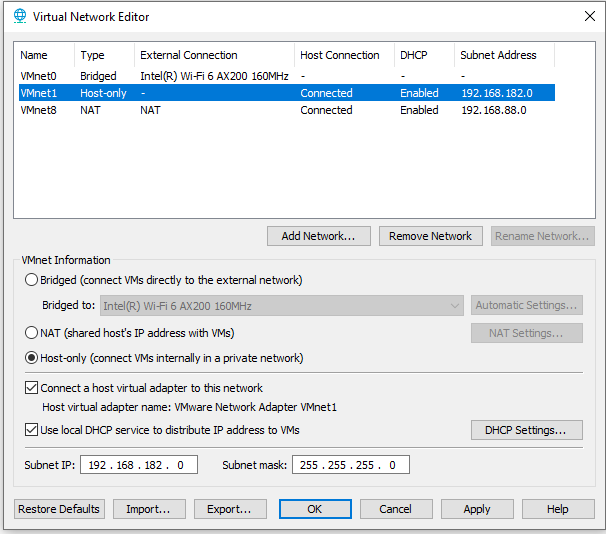
[](https://blogger.googleusercontent.com/img/a/AVvXsEjrKA8y8XsVOyqrVOsQkPwX8s7CBKQtLnOYXOnSyS7uJQzialGvELbeFW7m3EpZWL_8lFNxAQenECu5UU1sq0h_hwY0NV73yO29M-EP7BrzH7HdJhX5Qu_tqIW5LlT2rCE__bOPFG6XIeOBfiNdx-erjbRGr-sUYqTyl8E5XQrolkP-JpdaaMruA00s)

Chọn I Copied It:

[](https://blogger.googleusercontent.com/img/a/AVvXsEi3MaBmag3NToyEasQBTIp0UMSEd0m0mk76hKGgyKelGwvoQJ62SC2rpZazXh0hQmCf08gtOFR0JbHB7osrw7279CdmOE-MdbrjbMKlpXc_RYmjuZbX6UU_SYi1wyYImIyyQK4sCENui63eJeHM_0ftTMkqhelor8pnuAXgLAbRX3_GvUgsYUR5WRRc)

Thông tin đăng nhập: root,oracle/oracle    
**3.2. Vào VMWare Sửa lại địa chỉ HostOnly về dải 192.168.182.0:**

[](https://blogger.googleusercontent.com/img/a/AVvXsEjsbLcbBuHITDKP44f5-evf7HbRLb44c6JQsmbXB35nwgNcDdzIyiu07EKdf_0Z8PP07p9MxbRDEa7KxzP7nE5wioa-P1sOhWsWvMhKGE-sEUoDgPt2-RpkP5KEOigkjqiD8KdlaoE4-V3lywklCcblswGlBkmWz2fBl0zfhr2ApMJNC_P9co1bvxLo)

[](https://blogger.googleusercontent.com/img/a/AVvXsEhP5Cda8UxdQ8sEXEoMy-bEpWvrLSwe-vh5B71reyC2Pvof8AKYDvsdl8GBl-JQzPKIiO_6w2w_Jmf4qUV4iaWzOp7bJiNjHFx8nkOyx7jVtdBIbm6G6jrx9EmApDchP09I8QM0Zig8G8aJEOqSVP1oKQKWLK6mYGDg0hTgxmXdN8n0an3XggErQ_9h)

**3.3. Bật DB**

***oracle$sqlplus / as sysdba  
  
SQL> startup  
  
SQL> show pdbs  
  
SQL> startup  
SQL> show pdbs***

[](https://blogger.googleusercontent.com/img/a/AVvXsEhtYLSoqRPixLDn3N_HqovIxf_hsqh1eUCnaw8y5j-0E6TB9mrv5UlG0ya0wUOzIa6SgWOyyKS9FMvDB0okfVnc_dnVJHGcGTav_YXQPJ5fvzdLfEQi1pXdzHZ2sy2dKQ69H-zyI0SKJof2asBGhGAOlLFdmFKOVUBSHWZ1mGZZharNIIzlgZiIfP3b)

Bật listener:  
  
***$ lsnrctl start***

[](https://blogger.googleusercontent.com/img/a/AVvXsEiv_mJ7AugNKwsD3EVyDE_U8HZe7Cu2ynnJn2I_CDPlU4QoL0Mf-vhF3zxvsKpOv-oO2DqZo_NGjUNZ_H5laDg-oXJzB6UY2xhfU3essyGQ-WCgKShi9-fvH78cOC1_LnOjnzqSyKs5kgDN0FSEx4e12DEorINiIEHqFBhZNqCHjwEw0v0hKlj7rFhD)

   
Mật khẩu sys/123456

**4. Cài thêm các Tool khác:**

TOAD, SQL Navigator, SQL Developer, SecureCRT,  MobaXterm, Putty, XShell

## Lab+ bí kíp:

<https://drive.google.com/drive/folders/1xuc-trS839VjRoCyL4KvNGLIrafw6Ag_>

## Cài đặt thêm các công cụ khác:

Oracle Client 19c (bản 64 bit), TOAD, SecureSCT, Putty, WinSCP, tất cả công cụ ở đây:

<https://drive.google.com/drive/folders/18nIce5fRWOr1C4N8W8pkEzJno1nwqqpj?fbclid=IwAR24h-18-UCObS_xFxYusXSsoDRl6xrpi-zqCl3ccoQienFjIsflQDiyo-A>

## Các câu lệnh hay dùng với 12c, 19c:

SELECT name, open\_mode FROM v$pdbs;

SQL> show pdbs;

---ALTER PLUGGABLE DATABASE pdb1, pdb2 OPEN READ ONLY FORCE;

ALTER PLUGGABLE DATABASE pdb1 OPEN READ ONLY FORCE;

ALTER PLUGGABLE DATABASE pdb1, pdb2 CLOSE IMMEDIATE;

ALTER PLUGGABLE DATABASE pdb1 CLOSE IMMEDIATE;

ALTER PLUGGABLE DATABASE ALL OPEN;

ALTER PLUGGABLE DATABASE ALL CLOSE IMMEDIATE;

ALTER PLUGGABLE DATABASE ALL EXCEPT pdb1 OPEN;

ALTER PLUGGABLE DATABASE ALL EXCEPT pdb1 CLOSE IMMEDIATE;

SHOW CON\_NAME

ALTER SESSION SET container = pdb1;

ALTER SESSION SET container = cdb$root;

**Oracle enterprise express.**

SELECT DBMS\_XDB\_CONFIG.gethttpport FROM dual;

SELECT DBMS\_XDB\_CONFIG.gethttpsport FROM dual;

EXEC DBMS\_XDB\_CONFIG.sethttpsport(5500);

https://<hostname>:<port>/em/

# BÀI 1: GIỚI THIỆU NGHỀ QUẢN TRỊ DATABASE

## Các nhiệm vụ của Oracle DBA

Là quản trị viên cơ sở dữ liệu Oracle (DBA) bạn được giao quản trị "trái tim" của mỗi doanh nghiệp, bạn có thể được tham gia vào các tác vụ sau:

[](https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSSob9NNOoQ4_ul10BU7Nerei7uGo2Vrup3FduMcyfoIgD-aC0K&s)

* Sizing phần cứng để cài đặt hệt thống chạy Oracle Database đảm bảo an toàn, dự phòng cao, tiết kiệm.
* Cài đặt phần mềm Oracle Database, RAC, DataGuard, GoldenGate,...
* Tạo cơ sở dữ liệu Oracle Database
* Thực hiện nâng cấp cơ sở dữ liệu và phần mềm lên các cấp phiên bản mới của Oracle Database
* Bật và tắt instance cơ sở dữ liệu  Oracle Database, RAC, DataGuard, GoldenGate,...
* Quản lý cấu trúc lưu trữ của cơ sở dữ liệu: Quy hoạch lưu trữ, sắp xếp, tổ chức lại lưu trữ, chuyển đổi sang phân vùng SAN mới, chuyển dữ liệu lên Tape
* Quản lý người dùng và bảo mật đảm bảo an toàn tuyệt đối dữ liệu
* Quản lý các đối tượng cơ sở dữ liệu, chẳng hạn như table, index, constraint, view, trigger, sequence,...
* Sao lưu cơ sở dữ liệu (backup) và thực hiện các hoạt động khôi phục khi cần thiết (recovery)
* Theo dõi (monitor) trạng thái của cơ sở dữ liệu đảm bảo cơ sở dữ liệu hoạt động an toàn, hiệu năng cao, hoạt động liên tục 24 , cảnh bảo sớm các nguy cơ, rủi ro với cơ sở dữ liệu (Pro-Active)
* Thực hiện hành động phòng vệ, tránh bị ứng dụng, người dùng, hacker tấn công, truy cập vượt quá tài nguyên.
* Tối ưu hiệu năng (tunning) cơ sở dữ liệu, RAC, DataGuard, GoldenGate
* Chẩn đoán, xử lý các lỗi (troubleshooting) xảy ra với Oracle Database, RAC, DataGuard, GoldenGate,...
* Nghiên cứu các giải pháp bảo mật cho cơ sở dữ liệu đảm bảo an toàn tuyệt đối dữ liệu
* Các công việc khác liên quan đến dữ liệu: viết thủ tục/package bằng PL/SQL, SQL... để phục vụ nghiệp vụ báo cáo, thống kê, phân tích dữ liệu,...

Trong môi trường cơ sở dữ liệu vừa và nhỏ, bạn có thể là người duy nhất thực hiện các tác vụ này. Trong môi trường doanh nghiệp lớn, công việc thường được chia cho một số DBA, mỗi công việc có một chuyên môn, chẳng hạn như 1 người làm bảo mật cơ sở dữ liệu hoặc điều chỉnh cơ sở dữ liệu, 1 người quản trị backup, 1 người quản trị recovery, 1 người quản trị ASM, 2 người super Admin,...

## Tài liệu tham khảo:

<https://www.tranvanbinh.vn/p/tong-hop-cac-bai-viet-ve-nhap-mon.html>

# BÀI 2: KIẾN TRÚC CƠ SỞ DỮ LIỆU ORACLE

## KIẾN TRÚC CƠ SỞ DỮ LIỆU ORACLE

Oracle server là một hệ thống quản trị cơ sở dữ liệu đối tượng-quan hệ cho phép quản lý thông tin một cách toàn diện. Oracle server bao gồm hai thành phần chính là Oracle instance và Oracle database

**NỘI DUNG:**

**1.­­ KIẾN TRÚC ORACLE SERVER**

**1.1.  ORACLE INSTANCE**

                    Cấu trúc bộ nhớ (SGA, PGA)  
                    Cấu trúc tiến trình

**1.2.  Oracle database - Cấu trúc lưu trữ**

Cấu trúc vật lý database

Cấu trúc logic databse

1.3.  Quản trị cơ sở dữ liệu Oracle

1.4.  Thiết lập các tham số khởi tạo ảnh hưởng tới kích cỡ bộ nhớ SGA

**2. KẾT NỐI TỚI ORACLE SERVER**

2.1.  Mô hình kết nối

2.2.  Một số khái niệm cơ bản liên quan đến kết nối

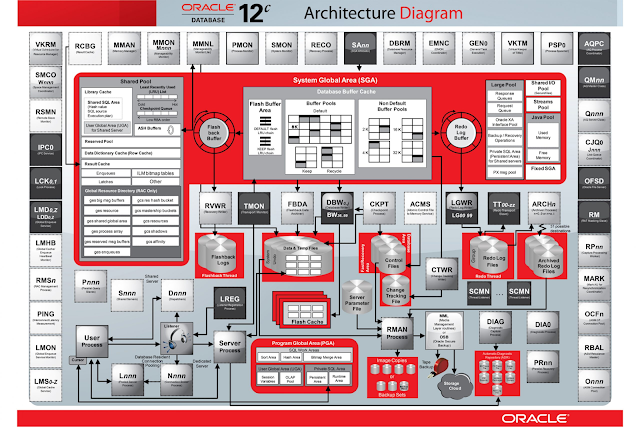
2.3.  Kết nối tới database

**3. Áp dụng kiến trúc Cơ sở dữ liệu Oracle để hiểu hoạt động SELECT, INSERT, UPDATE, DELETE**

Chi tiết bài viết:

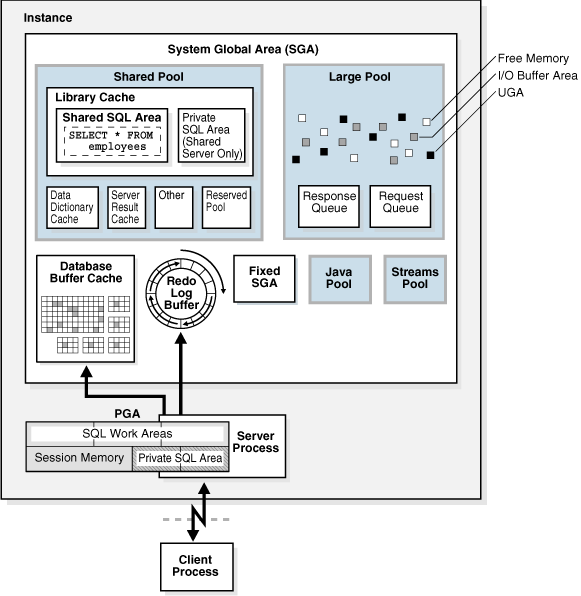
### 1.­­ KIẾN TRÚC ORACLE SERVER

Oracle server là một hệ thống quản trị cơ sở dữ liệu đối tượng-quan hệ cho phép quản lý thông tin một cách toàn diện. *Oracle server bao gồm hai thành phần chính là Oracle instance và Oracle database*.

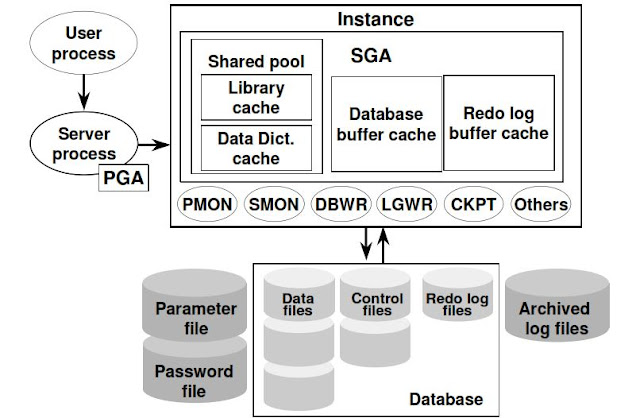
[](https://lh3.googleusercontent.com/-J4ffsmI1mwE/YDoe92zwIZI/AAAAAAAAAtA/dy6b6VSjZhoUWVmx-WCGLKouYGAU4seRwCLcBGAsYHQ/image.png)

### 1.1. ORACLE INSTANCE

Oracle instance bao gồm một **Cấu trúc bộ nhớ** (System Global Area (SGA) và Program Global Area (PGA) được sử dụng để quản trị cơ sở dữ liệu, bộ nhớ lưu mã chương trình đang chạy, dữ liệu chia sẻ cho các user (shared pool, buffer cache, UGA) và dữ liệu riêng cho mỗi user kết nối đến) và **Cấu trúc tiến trình**. Oracle instance được xác định qua tham số môi trường ORACLE\_SID của hệ điều hành.

[](https://blogger.googleusercontent.com/img/a/AVvXsEgKdGRdhmTkic8REYnIMKWER_ZDGInSX6YXHboy3oJNf8ILSVbjQA6MyAEKDH7X8KynGUOqYO28wbAbKpUYx70YpecRRs5p80G5WB4vSq98i0Gmom1NCfL5Lh_N5CdyLncZHkh5bNEmOxYBHp8g-l7Kx8fFPM2DZ0mhRvl-jOlzFyfZSaJoCGVwz-ej)

#### CẤU TRÚC BỘ NHỚ

[](https://www.blogger.com/null)

*Hình vẽ 1.    Kiến trúc Oracle Server*

##### System Global Area - SGA

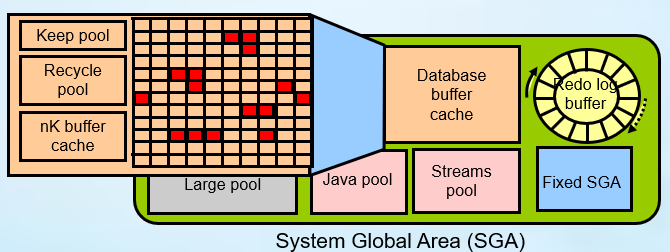
SGA là vùng bộ nhớ chia sẻ được sử dụng để lưu trữ dữ liệu và các thông tin điều khiển của Oracle server. SGA được cấp phát (allocated) trong bộ nhớ của máy tính mà Oracle server đang hoạt động trên đó. Các User kết nối tới Oracle sẽ chia sẻ các dữ liệu có trong SGA, việc mở rộng không gian bộ nhớ cho SGA sẽ làm nâng cao hiệu suất của hệ thống, lưu trữ được nhiều dữ liệu trong hệ thống hơn đồng thời giảm thiểu các thao tác truy xuất đĩa (disk I/O).

SGA bao gồm một vài cấu trúc bộ nhớ chính:

§  Shared pool: Là một phần của SGA lưu các cấu trúc bộ nhớ chia sẻ.

§  Database buffer cache: Lưu trữ các dữ liệu được sử dụng gần nhất.

§  Redo log buffer: Được sử dụng cho việc dò tìm lại các thay đồi trong cơ sở dữ liệu và được thực hiện bởi các background process.

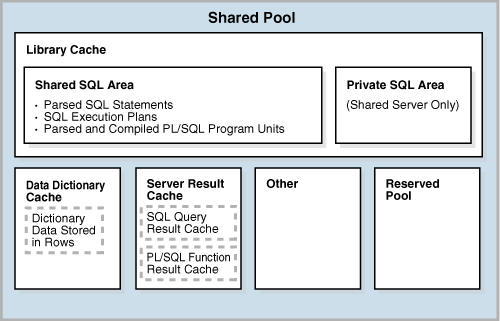
[](https://blogger.googleusercontent.com/img/a/AVvXsEiPTKVRm9abAL_ennqxxBYfxUR1VRi2iaM12e4Azbj4vrn9T5-DEJ8IOWCqRKqDHfE0_ZttcmmbFwl3Z_j_PRC2Lr8-gO1hFsFsJh5uf584vuPj1m9IH7D7_g3HwN_OMz_YtSwIkAd_2WiNrfVqnbqtwpzdQjXpz85h7NIiU5aHf_8WDifgtXcCooaS)

Để chi tiết hơn, ta sẽ xem xét cụ thể từng thành phần.

**Share Pool**

Shared pool là một phần trong SGA và được sử dụng khi thực hiện phân tích câu lệnh (parse phase). Kích thước của Shared pool được xác định bởi tham số SHARED\_POOL\_SIZE có trong parameter file (file tham số).

Các thành phần của Shared pool gồm có: Library cache và Data dictionary cache.



*Hình vẽ 2.    Cấu trúc Share Pool*

***Library Cache***

Library cache lưu trữ thông tin về các câu lệnh SQL được sử dụng gần nhất bao gồm:

§  Nội dung của câu lệnh dạng text (văn bản).

§  Parse tree (cây phân tích) được xây dựng tuỳ thuộc vào câu lệnh.

§  Execution plan (sơ đồ thực hiện lệnh) gồm các bước thực hiện và tối ưu lệnh.

Do các thông tin trên đã được lưu trữ trong Library cache nên khi thực hiện lại một câu lệnh truy vấn, trước khi thực hiện câu lệnh, Server process sẽ lấy lại các thông tin đã được phân tích mà không phải phân tích lại câu lệnh. Do vậy, Library cache có thể giúp nâng cao hiệu suất thực hiện lệnh.

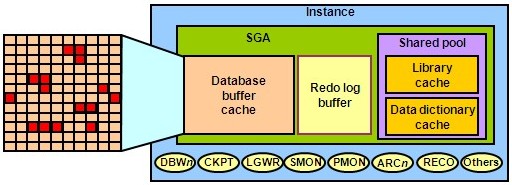
***Data Dictionary Cache***

Data dictionary cache là một thành phần của Shared pool lưu trữ  thông tin của dictionary cache được sử dụng gần nhất như các định nghĩa các bảng, định nghĩa các cột, usernames, passwords, và các privileges (quyền).

Trong giai đoạn phân tích lệnh (parse phase), Server process sẽ tìm các thông tin trong dictionary cache để xác định các đối tượng trong câu lệnh SQL và để xác định các mức quyền tương ứng. Trong trường hợp cần thiết, Server process có thể khởi tạo và nạp các thông tin từ các file dữ liệu.

**Data buffer cache**

Khi thực hiện một truy vấn, Server process sẽ tìm các blocks cần thiết trong database buffer cache. Nếu không tìm thấy block trong database buffer cache, Server process mới đọc các block từ data file và tạo luôn một bản sao của block đó vào trong vùng nhớ đệm (buffer cache). Như vậy, với các lần truy xuất tới block đó sau này sẽ không cần thiết phải truy xuất vào datafile nữa.



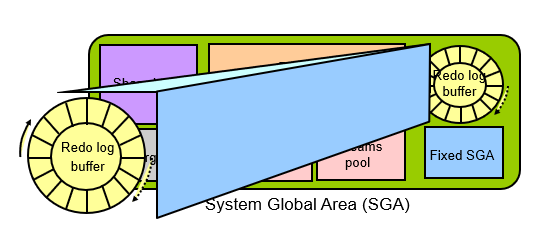
*Hình vẽ 3.    Database buffer cache*

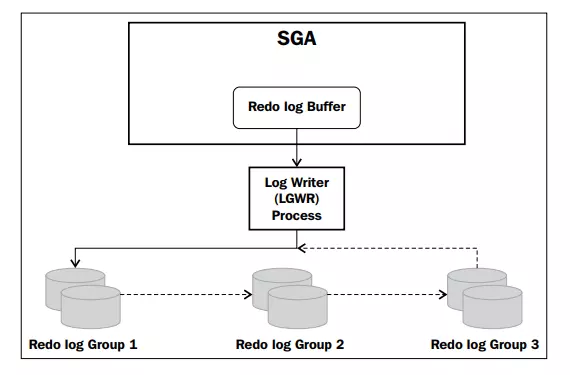
Database buffer cache là vùng nhớ trong SGA sử dụng để lưu trữ các block dữ liệu được sử dụng gần nhất. Tương tự như kích thước của blocks dữ liệu được xác định bởi tham số DB\_BLOCK\_SIZE, kích thước của vùng đệm trong buffer cache cũng được xác định bởi tham số DB\_BLOCK\_BUFFERS.

Oracle server sử dụng giải thuật least recently used (LRU) algorithm để làm tươi lại vùng nhớ. Theo đó, khi nạp mới một block vào bộ đệm, trong trường hợp bộ đệm đã đầy, Oracle server sẽ loại bớt block ít được sử dụng nhất ra khỏi bộ đệm để nạp block mới vào bộ đệm.

**Redo log buffer**

Server process ghi lại các thay đổi của một instance vào redo log buffer, đây cũng là một phần bộ nhớ SGA.

[](https://blogger.googleusercontent.com/img/a/AVvXsEjjun4WaNJePXKTHVoTKtFEBe2LJBGmxTSCbqQOE_dTT5g79KQu4dJhfQJ0HB-oOcEaen9FK1xWfiput5cyy3Q7vQHPiOCTf7PwSRG5OGNYrGUYVlKiJ53qeBjYuaoUu3fd8dxIRuKIFlQIejM-qkG0s5PF2WQvjsomUFIdd02JXUEixAJCHwaSN6_9)

[](https://blogger.googleusercontent.com/img/a/AVvXsEijIicvPOXNX7cmbLrGsKAaQhli_g_Z2ztDZ10eqNuUKKwz_uT8FlNOUQaFBlRiLeaQrlU5qMuF-yfo3c5L5DN2UXacvlvB6M9GDigjnUW0DvqqU7Rm52aSeyWQn4jyMkN0pyPeXaAsmDmcu2JbueGwkjOHRdJ3_94X6TAn46VExsWzFwPp92Ay0pDg)

*Hình vẽ 4.    Redo log buffer*

Có một số đặc điểm cần quan tâm của Redo log buffer:

§  Kích thước được xác định bởi tham số LOG\_BUFFER.

§  Lưu trữ các redo records (bản ghi hồi phục) mỗi khi có thay đổi dữ liệu.

§  Redo log buffer được sử dụng một cách thường xuyên và các thay đổi bởi một transaction có thể nằm đan xen với các thay đổi của các transactions khác.

§  Bộ đệm được tổ chức theo kiểu circular buffer (bộ đệm nối vòng) tức là dữ liệu thay đổi sẽ tiếp tục được nạp lên đầu sau khi vùng đệm đã được sử dụng hết.

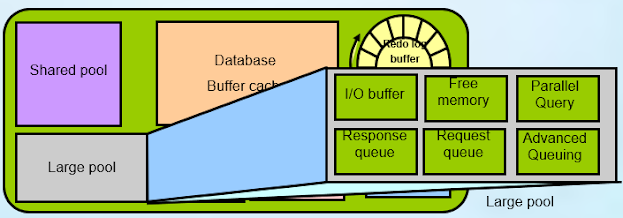
**Large Pool**

Sử dụng cho:

§  Bộ nhớ của session cho shared server và giao diện Oracle XA

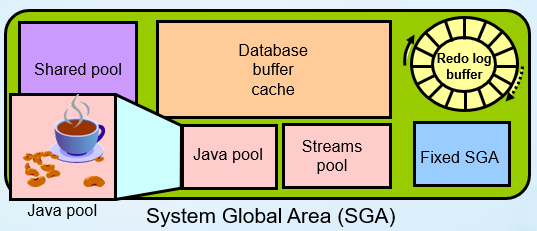
§  Tiến trình server liên quan đến I/O

§  Hoạt động backup and restore CSDL

[](https://blogger.googleusercontent.com/img/a/AVvXsEiecKfQTWiUhVjVy_BqFiVqEyd-5365ShIicxbVh26qgAsKb8BdA_JAQOtrAtM-gwYQc2L5Oge-wxqmSX3N1gCchQ8fYfHmkCyXA91djjrtDNreNvnQVo_tAT4b-nDN5dIj_lmxjFObP2CsvrjjBD1rPvaS5wA_3GgT8ADFAEGN6ZBUiqzhkQHDFdis)

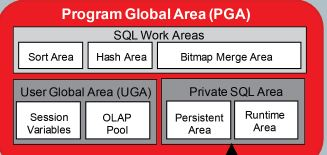
**Java Pool**

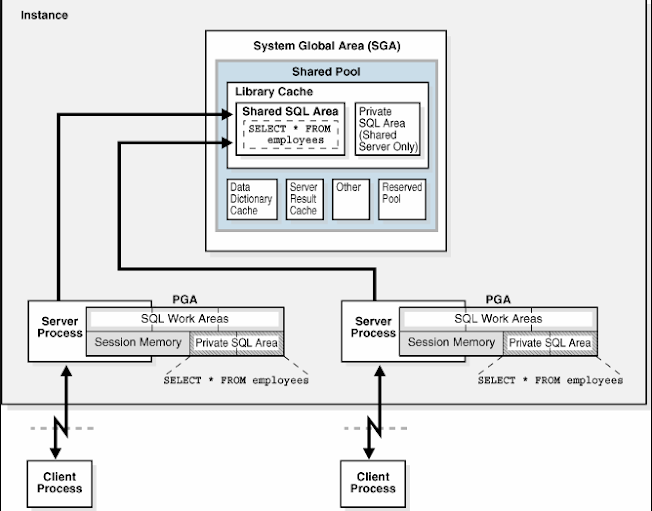
Vùng nhớ Java pool được sử dụng cho các đoạn Java code và dữ liệu trong JVM của các session xác định trước đó

[](https://blogger.googleusercontent.com/img/a/AVvXsEiWRSsK0aJL2nhUbBrHrA37HolkkGE4z6NV8LLMprODr8QAbrRd-xb9FGM9ek8ZQSus3hokgyTc4KB7gs3AF4_nlPhOrbsNug_jeNBD1XwlWCAqs_Osk1LFDb5en9FIfjtPp0KvoO8PDbBY5Nhylza2VkC3c1SEAEzOUciyNkrb3BsRIku39XMvZQPo)

##### PROGRAM GLOBAL AREA - PGA

Trong Oracle Database, PGA (Program Global Area) là một khu vực bộ nhớ đặc biệt được sử dụng bởi mỗi tiến trình (process) của hệ thống database. PGA có tác dụng chứa các cấu trúc dữ liệu và thông tin cần thiết cho việc thực thi các câu lệnh SQL và các hoạt động liên quan.

[](https://blogger.googleusercontent.com/img/a/AVvXsEikExRkxckpYkhFAYrsATBneuA1xa50x6PHPXdp2QGYABBUc1McElgpc5Id-7iBDVbnQKbiR4uxxsu7mECQ_fkM9q2fkj8EUND4SIihyuYPG-03LTM24OY3s7L711AwLjkADvSV_MTzfjfVIW92qwWy5Gfviyvt7Ac_6WIbixbQxa5HPiOsTxoRa8VM)

[](https://blogger.googleusercontent.com/img/a/AVvXsEhYqDUqJj-ahCtjZCDaPsI0bpfN-Na0IdsPccrbOU45eulZ11s_cQ6tVmCvElc0QcZeQpzbxp2Z_p4cqISYdRaNhnx3qhTCCOsH4kxwd03CHVeWf8ka3vmneMPqFRl5H1hCjnPMen_M8A3JjNYwuqlfqGnroidCYwCpuhVgzfD77sJ6LcjrZufDvkt5)

Các tác dụng chính của PGA trong Oracle Database bao gồm:

1. Quản lý bộ nhớ: PGA giúp **quản lý bộ nhớ cho mỗi tiến trình**, bao gồm bộ nhớ cho các vùng cấu trúc dữ liệu như Sorting Area (vùng sắp xếp như ORDER BY, GROUP BY), Hash Area (vùng băm), và Bitmap Merge Area (vùng ghép bitmap), giúp tăng hiệu suất thực thi câu lệnh SQL.
2. Lưu trữ dữ liệu **tạm thời**: PGA cũng được sử dụng để lưu trữ các dữ liệu tạm thời trong quá trình thực thi câu lệnh SQL, chẳng hạn như kết quả của các phép sắp xếp, kết quả của các phép toán băm, và các dữ liệu tạm thời khác.
3. Quản lý các **biến** môi trường: PGA cung cấp không gian để lưu trữ các biến môi trường như biến số, tham số và các giá trị tạm thời khác trong quá trình thực thi các câu lệnh SQL.
4. Quản lý vùng **stack**: PGA quản lý vùng stack cho mỗi quy trình, trong đó lưu trữ các thông tin quan trọng như địa chỉ của các khối mã, các biến cục bộ và thông tin về trạng thái của quy trình.

Tóm lại, PGA trong Oracle Database có vai trò quan trọng trong việc quản lý bộ nhớ, lưu trữ dữ liệu tạm thời và các biến môi trường, giúp cải thiện hiệu suất và độ tin cậy của hệ thống.

#### CẤU TRÚC TIẾN TRÌNH

Tiến trình User

    - Là ứng dụng hoặc công cụ (TOAD, SQL Navigator,..) kết nối tới CSDL Oracle

Tiến trình Database

    - Tiến trình Server: Kết nối tới Oracle instance và được bắt đầu khi một user thiết lập một session

    - Tiến trình Background: Được bật cùng khi Oracle instance được bật.

Tiến trình Daemon / Application

    - Listener mạng

    - Grid Infrastructure daemon

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 tiêu biểu  như:

§  Database Writer (DBW0): Ghi lại các thay đổi trong data buffer cache ra các file dữ liệu.

§  Log Writer (LGWR): Ghi lại các thay đổi được đăng ký trong redo log buffer vào các  redo log files.

§  System Monitor (SMON): Kiểm tra sự nhất quán trong database.

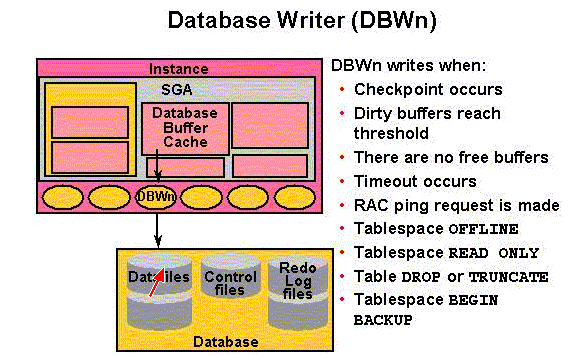
§  Process Monitor (PMON): Dọn dẹp lại tài nguyên khi các tiến trình của Oracle gặp lỗi.

§  Checkpoint Process (CKPT): Cập nhật lại trạng thái của thông tin trong file điều khiển (control file) và file dữ liệu (datafile) mỗi khi có thay đổi trong buffer cache.

**Database Writer (DBW0)**

Server process ghi lại các dữ liệu thay đổi để rollback và dữ liệu của các block trong buffer

cache. Database writer (DBWR) ghi các thông tin được đánh dấu thay đổi từ database buffer cache lên các data files nhằm đảm bảo luôn có khoảng trống bộ đệm cần thiết cho việc sử dụng.



*Hình vẽ 5.    Database Writer (DBWR)*

Với việc sử dụng này, hiệu suất sử dụng database sẽ được cải thiện do Server processes chỉ tạo các thay đổi trên buffer cache, DBWR ghi dữ liệu vào các data file cho tới khi:

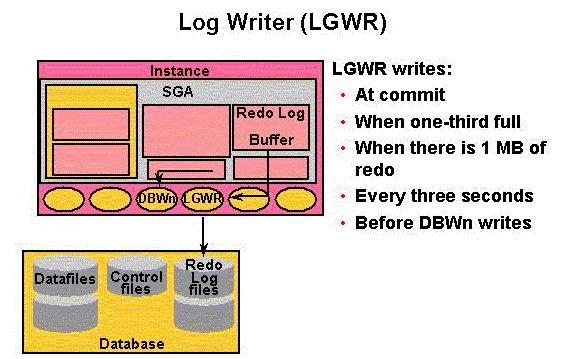
§  Số lượng buffers đánh bị dấu đạt tới giá trị ngưỡng.

§  Tiến trình duyệt tất cả buffer mà vẫn không tìm thấy dữ liệu tương ứng.

§  Quá thời gian quy định.

**Log Writer**

Log Writer (LGWR) là một trong các background process có trách nhiệm quản lý redo log buffer để ghi lại các thông tin trong Redo log buffer vào Redo log file. Redo log buffer là bộ đệm dữ liệu được tổ chức theo kiểu nối vòng.



*Hình vẽ 6.    Log Writer (LGWT)*

LGWR ghi lại dữ liệu một cách tuần tự vào redo log file theo các tình huống sau:

§  Khi redo log buffer đầy

§  Khi xảy ra timeout (thông thường là 3 giây)

§  Trước khi DBWR ghi lại các blocks bị thay đổi trong data buffer cache vào các data files.

§  Khi commit một transaction.

**System Monitor (SMON)**

Tiến trình*system monitor (*SMON*)*thực hiện phục hồi các sự cố (crash recovery) ngay tại thời điểm instance được khởi động (startup), nếu cần thiết. SMON cũng có trách nhiệm dọn dẹp các temporary segments không còn được sử dụng nữa trong dictionary-managed tablespaces.  SMON khôi phục lại các transactions bị chết mỗi khi xảy ra sự cố. SMON đều đặn thực hiện kiểm tra và khắc phục các sự cố khi cần.

Trong môi trường Oracle Parallel Server, SMON process của một instance có thể thực hiện khôi phục instance trong trường hợp instance hay CPU của máy tính đó gặp sự cố.

**Process Monitor (PMON)**

Tiến trình *process monitor (*PMON*)*thực hiện tiến trình phục hồi mỗi khi có một user process gặp lỗi. PMON có trách nhiệm dọn dẹp database buffer cache và giải phóng tài nguyên mà user process đó sử dụng. Ví dụ, nó thiết lập lại (reset) trạng thái của các bảng đang thực hiện trong transaction, giải phóng các locks trên bảng này,  và huỷ bỏ process ID của nó ra khỏi danh sách các active processes.

PMON kiểm tra trạng thái của nơi gửi (dispatcher ) và các server processes, khởi động lại (restarts) mỗi khi xảy ra sự cố. PMON cũng còn thực hiện việc đăng ký các thông tin về instance và dispatcher processes với network listener.

Tương tự như SMON, PMON được gọi đến mỗi khi xảy ra sự cố trong hệ thống.

**Checkpoint Process (CKPT)**

Cập nhật lại trạng thái của thông tin trong file điều khiển và file dữ liệu mỗi khi có thay đổi trong buffer cache. Xảy ra checkpoints khi:

§  Tất cả các dữ liệu trong database buffers đã bị thay đổi tính cho đến thời điểm checkpointed sẽ được background process DBWRn ghi lên data files.

§  Background process CKPT cập nhật phần headers của các data files và các control files.

Checkpoints có thể xảy ra đối với tất cả các data files trong database hoặc cũng có thể xảy ra với một data files cụ thể.

Checkpoint xảy ra theo các tình huống sau:

§  Mỗi khi có log switch

§  Khi một shut down một database với các chế độ trừ chế độ abort

§  Xảy ra theo như thời gian quy định trong các tham số khởi tạo LOG\_CHECKPOINT\_INTERVAL (tính theo block OS, default 0) và LOG\_CHECKPOINT\_TIMEOUT (tinsht heo thời gian, default 1800s). Nếu FAST\_START\_MTTR\_TARGET (tham số này xác định thời gian để phục hồi crash revoery) được thiết lập thì LOG\_CHECKPOINT\_INTERVAL bị ghi đè, để dùng tham số này thì phải disable tham số LOG\_CHECKPOINT\_TIMEOUT về 0.

§  Khi có yêu cầu trực tiếp của quản trị viên (alter system switch logfile/checkpoint)

Thông tin về checkpoint được lưu trữ trong Alert file trong trường hợp các tham số khởi tạo LOG\_CHECKPOINTS\_TO\_ALERT được đặt là TRUE. Và ngược lại với giá trị FALSE.

### 1.2. ORACLE DATABASE

Oracle database là tập hợp các dữ liệu được xem như một đơn vị thành phần (Unit). Database có nhiệm vụ lưu trữ và trả về các thông tin liên quan. Database được xem xét dưới hai góc độ cấu trúc logic và cấu trúc vật lý . Tuy vậy, hai cấu trúc dữ liệu này vẫn tồn tại tách biệt nhau, việc quản lý dữ liệu theo cấu trúc lưu trữ vật lý không gây ảnh hưởng tới cấu trúc logic

Oracle database được xác định bởi tên một tên duy nhất và được quy định trong tham số DB\_NAME của parameter file.

*Hình vẽ 7.    Cấu trúc database*

#### Cấu trúc vật lý database

Cấu trúc vật lý bao gồm tập hợp các control file, online redo log file và các datafile:

**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.

§  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.

**Redo Log Files**

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's *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*.

**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 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.

#### Cấu trúc logic databse

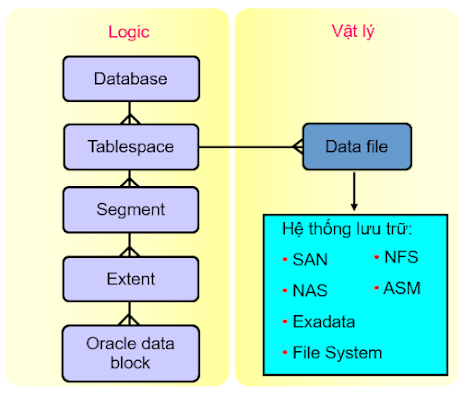
Cấu trúc logic của Oracle database bao gồm các đối tượng tablespaces, schema objects, data blocks, extents, và segments.

**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 có quan hệ logic với nhau.

***Databases, Tablespaces, và Datafiles***

Mối quan hệ giữa các databases, tablespaces, và datafiles có thể được minh hoạ bởi hình vẽ sau:

[](https://blogger.googleusercontent.com/img/a/AVvXsEg1pKB4VJehy_5FEBLwp0yWBkugKzpO5O-KssUOsfeU53jSxZEYI1jAEYzVPqYZRIh7SaCjAGLMadPyUpM9PWUXgpX9qmvi-JDh8NW0gTyu8vq2ywhDYQFAHrhLgFIcc2i-Oex9Li6HSjdu-cYuvw0HM1qbAWQ9F2Gny4wBBGg4IJANYQ9lQFMFlFw4)

*Hình vẽ 8.    Quan hệ giữa database, tablespace và datafile*

Có một số điểm ta cần quan tâm:

§  Mỗi database có thể phân chia về mặt logic thành một hay nhiều tablespace.

§  Mỗi tablespace có thể được tạo nên, về mặt vật lý, bởi một hoặc nhiều datafiles.

§  Kích thước của một tablespace bằng tổng kích thước của các datafiles của nó. Ví dụ: trong hình vẽ ở trên SYSTEM tablespace có kích thước là 2 MB còn USERS tablespace có kích thước là 4 MB.

§  Kích thước của database cũng có thể xác định được bằng tổng kích thước của các tablespaces của nó. Ví dụ: trong hình vẽ trên thì kích thước của database là 6 MB.

**Schema và Schema Objects**

S*chema* là tập hợp các đối tượng (objects) có trong database. *Schema objects* là các cấu trúc logic  cho phép tham chiếu trực tiếp tới dữ liệu trong database. Schema objects bao gồm các cấu trúc như tables, views, sequences, stored procedures, synonyms, indexes, clusters, và database links.

**Data Blocks, Extents, and Segments**

Oracle điểu khiển không gian lưu trữ trên đĩa cứng theo các cấu trúc logic  bao gồm các data blocks, extents, và segments.

***Oracle Data Blocks***

Là mức phân cấp logic  thấp nhất, các dữ liệu của Oracle database được lưu trữ trong các *data blocks*. Một data block tương ứng với một số lượng nhất định các bytes vật lý của database trong không gian đĩa cứng. Kích thước của một data block được chỉ ra cho mỗi Oracle database ngay khi database được tạo lập. Database sử dụng, cấp phát và giải phóng vùng không gian lưu trữ thông qua các Oracle data blocks.

***Extents***

Là mức phân chia cao hơn về mặt logic các vùng không gian trong database. Một *extent* bao gồm một số data blocks liên tiếp nhau, cùng được lưu trữ tại một thiết bị lưu giữ. Extent được sử dụng để lưu trữ các thông tin có cùng kiểu.

***Segments***

Là mức phân chia cao hơn nữa về mặt logic  các vùng không gian trong database. Một *segment* là một tập hợp các extents được cấp phát cho một cấu trúc logic . Segment có thể được phân chia theo nhiều loại khác nhau:

|  |  |
| --- | --- |
| Data segment | Mỗi một non-clustered table có một data segment. Các dữ liệu trong một table được lưu trữ trong các extents thuộc data segment đó. Với một partitioned table thì mỗi each partition lại tương ứng với một data segment.  Mỗi Cluster tương ứng với một data segment. Dữ liệu của tất cả các table trong cluster đó đều được lưu trữ trong data segment thuộc Cluster đó. |
| index segment | Mỗi một index đều có một index segment lưu trữ các dữ liệu của nó. Trong partitioned index thì mỗi partition cũng lại tương ứng với một index segment. |
| rollback segment | Một hoặc nhiều rollback segments của database được tạo lập bởi người quản trị database để lưu trữ các dữ liệu trung gian phục vụ cho việc khôi phục dữ liệu.  Các thông tin trong Rollback segment được sử dụng để:  §  Tạo sự đồng nhất các thông tin đọc được từ database  §  Sử dụng trong quá trình khôi phục dữ liệu  §  Phục hồi lại các giao dịch chưa commit đối với mỗi user |
| temporary segment | Temporary segments được tự động tạo bởi Oracle mỗi khi một câu lệnh SQL statement cần đến một vùng nhớ trung gian để thực hiện các công việc của mình như sắp xếp dữ liệu. Khi kết thúc câu lệnh đó, các extent thuộc temporary segment sẽ lại được hoàn trả cho hệ thống. |

Oracle thực hiện cấp phát vùng không gian lưu trữ một cách linh hoạt mỗi khi các extents cấp phát đã sử dụng hết.

#### Các cấu trúc vật lý khác

Ngoài ra, Oracle Server còn sử dụng các file khác để lưu trữ thông tin. Các file đó bao gồm:

§  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.

§  Archived redo log files: Là bản off line của các redo log files chứa các thông tin cần thiết để phục hồi dữ liệu.

### 1.3. QUẢN TRỊ CƠ SỞ DỮ LIỆU ORACLE

Quản trị cơ sở dữ liệu là công việc bảo trì và vận hành Oracle server để nó có thể tiếp nhận và xử lý được tất cả các yêu cầu (requests) từ phía Client. Để làm được điều này, người quản trị viên cơ sở dữ liệu cần phải hiểu được kiến trúc của Oracle database.

### 1.4. Thiết lập các tham số khởi tạo ảnh hưởng tới kích cỡ bộ nhớ SGA

Tham số khởi tạo ảnh hưởng tới kích thước bộ nhớ cấp phát cho vùng System Global Area. Ngoại trừ tham số SGA\_MAX\_SIZE, còn lại các tham số khác đều là tham số động tức là có thể thay đổi giá trị của chúng ngay trong lúc database đang chạy thông qua câu lệnh ALTER SYSTEM. Kích thước của SGA cũng có thể thay đổi được trong quá trình chạy database.

**Thiết lập tham số cho Buffer Cache**

Tham số khởi tạo buffer cache quy định kích thước của buffer cache là một phần của SGA. .

Ta sử dụng các tham số DB\_CACHE\_SIZE và một trong những tham số DB\_*n*K\_CACHE\_SIZE để cho phép sử dụng chế độ multiple block sizes đối với database. Oracle sẽ tự động gán các giá trị mặc định cho tham số the DB\_CACHE\_SIZE, còn tham số DB\_*n*K\_CACHE\_SIZE sẽ được gán mặc định bằng 0.

Kích thước của buffer cache sẽ có ảnh hưởng nhiều tới hiệu suất thực hiện của hệ thống. Kích thước càng lớn thì càng giảm bớt việc đọc và ghi đĩa. Tuy nhiên, kích thước của cache lớn sẽ tốn nhiều bộ nhớ và sẽ có nhiều tốn kém trong việc thực hiện paging (phân trang) hay swapping (trao đổi) bộ nhớ.

**Tham số DB\_CACHE\_SIZE**

Tham số khởi tạo DB\_CACHE\_SIZE được sử dụng thay thế cho tham số DB\_BLOCK\_BUFFERS của các phiên bản Oracle trước kia. Tham số DB\_CACHE\_SIZE quy định kích thước của block buffers chuẩn. Kích thước của một block chuẩn lại được quy định trong tham số DB\_BLOCK\_SIZE.

Tuy vậy, tham số DB\_BLOCK\_BUFFERS vẫn được sử dụng để tương thích với các phiên bản trước, tuy nhiên giá trị của nó không được sử dụng cho các tham số động.

**Tham số DB\_nK\_CACHE\_SIZE**

Chỉ ra kích cỡ là bội số nguyên lần kích thước của block buffers. Nó được chỉ ra bởi các tham số:

§  DB\_2K\_CACHE\_SIZE

§  DB\_4K\_CACHE\_SIZE

§  DB\_8K\_CACHE\_SIZE

§  DB\_16K\_CACHE\_SIZE

§  DB\_32K\_CACHE\_SIZE.

Mỗi tham số chỉ ra kích cỡ của buffer cache tương ứng với kích cỡ của block.

Ví dụ:

DB\_BLOCK\_SIZE=4096

DB\_CACHE\_SIZE=12M

DB\_2K\_CACHE\_SIZE=8M

DB\_8K\_CACHE\_SIZE=4M

Ở ví dụ này, các tham số chỉ ra kích thước block chuẩn của database là 4K. Kích thước cache tương ứng với kích thước block chuẩn là 12M. Các kích thước mở rộng của cache là 2K và 8K sẽ được đặt lại với giá trị tương ứng là 8M và 4M.

**Điều chỉnh kích cỡ của Shared Pool**

Tham số SHARED\_POOL\_SIZE trong phiên bản Oracle  11g là tham số động, tức là có thể thay đổi được giá trị của nó (điều này không thể thực hiện được trong các phiên bản trước). Nó cho phép ta thay đổi kích thước của shared pool là một trong các thành phần của SGA. Theo mặc định Oracle cũng chọn một giá trị mặc định phù hợp cho tham số này.

**Điều chỉnh kích cỡ của Large Pool**

Tương tự như SHARED\_POOL\_SIZE,  tham số LARGE\_POOL\_SIZE cũng là một tham số động, nó cho phép ta điều chỉnh kích cỡ của large pool, đây cũng là một thành phần trong SGA. .

**Giới hạn kích cỡ của SGA**

Tham số SGA\_MAX\_SIZE quy định kích cỡ lớn nhất của System Global Area . Ta cũng có thể thay đổi kích cỡ của buffer caches, shared pool và large pool, tuy nhiên việc thay đổi này nên là mở rộng giá trị kích thước cho các thành phần của SGA. Giá trị mở rộng thêm này cũng không nên đặt tới ngưỡng của SGA\_MAX\_SIZE.

Trong trường hợp ta không chỉ rõ giá trị của SGA\_MAX\_SIZE thì Oracle sẽ tự động gán giá trị này bằng tổng số kích cỡ của các thành phần của SGA lúc ban đầu.

*Ghi chú: Bạn chỉ cần đặt SGA, PGA hoặc Memory còn lại các bộ nhớ con để tự Oracle phân bổ theo AMM hoặc ASMM.*

### 2.KẾT NỐI TỚI ORACLE SERVER

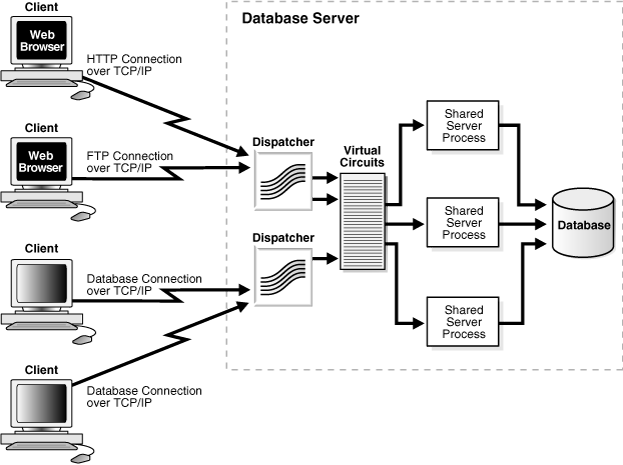
#### 2.1.  MÔ HÌNH KẾT NỐI

Các Client có thể kết nối tới Oracle Server thông qua 03 cách sau:

§  Kết nối trực tiếp: kết nối mà Client nằm trên cùng một máy chủ Oracle server.

§  Kết nối hai lớp (two-tiered) client-server: Client nằm trên một máy tính khác và kết nối trực tiếp tới máy chủ Oracle Server.

§  Kết nối ba lớp (three-tiered): Client nằm trên máy tính khác với máy chủ Oracle Server, nó giao tiếp với một ứng dụng hay một máy chủ mạng (network server) và điều khiển ứng dụng hay máy chủ này kết nối tới Oracle server.



*Hình vẽ 9.    Kết nối tới Oracle server*

#### 2.2.  Một số khái niệm cơ bản liên quan đến kết nối

Connection (liên kết)

Liên kết là đường liên lạc giữa một user process và một Oracle server. Trong trường hợp user sử dụng các tool hoặc các ứng dụng ngay trên cùng một máy với Oracle server, đường liên lạc sẽ được tạo lập ngay trên máy đó. Trong trường hợp user nằm trên một máy khác thì liên kết sẽ sử dụng đường mạng để kết nối tới Oracle server.

Session (phiên)

Một phiên tương ứng với một liên kết cụ thể của một user tới một Oracle server. Phiên bắt đầu khi user kết nối tới Oracle Server đã được kiểm tra hợp lệ và kết thúc khi user thực hiện log out khỏi Oracle Server hoặc user kết thúc một cách bất thường. Một user cùng một lúc có thể có nhiều phiên làm việc để kết nối tới Oracle Server thông qua các ứng dụng hay các tool khác nhau. Ví du: User có thể đồng thời có các phiên làm việc giữa SQL\*Plus, TOAD, SQL Navigator,... tới Oracle Server.

Lưu ý: Phiên chỉ tạo lập được khi Oracle Server đã sẵn sàng cho việc kết nối của các client.

#### 2.3.  KẾT NỐI TỚI DATABASE

Các bước thực hiện kết nối

Để kết nối tới database trước tiên, cần phải tạo liên kết tới Oracle Server. Liên kết tới Oracle Server được tạo theo các bước sau:

§  User sử dụng công cụ SQL\*Plus hay sử dụng các công cụ khác của Oracle như TOAD, SQL Navigator, SQL Developer để khởi tạo tiến trình. Trong mô hình Client-Server, các công cụ hay ứng dụng này được chạy trên máy Client.

§  User thực hiện log in vào Oracle server với việc khai báo username, password và tên liên kết tới database. Các ứng dụng tools sẽ tạo một tiến trình để kết nối tới Oracle server qua các tham số này. Tiến trình này được gọi là tiến trình phục vụ. Tiến trình phục vụ sẽ giao tiếp với Oracle server thay cho tiến trình của user chạy trên máy Client.

Ví dụ thực hiện kết nối tới database

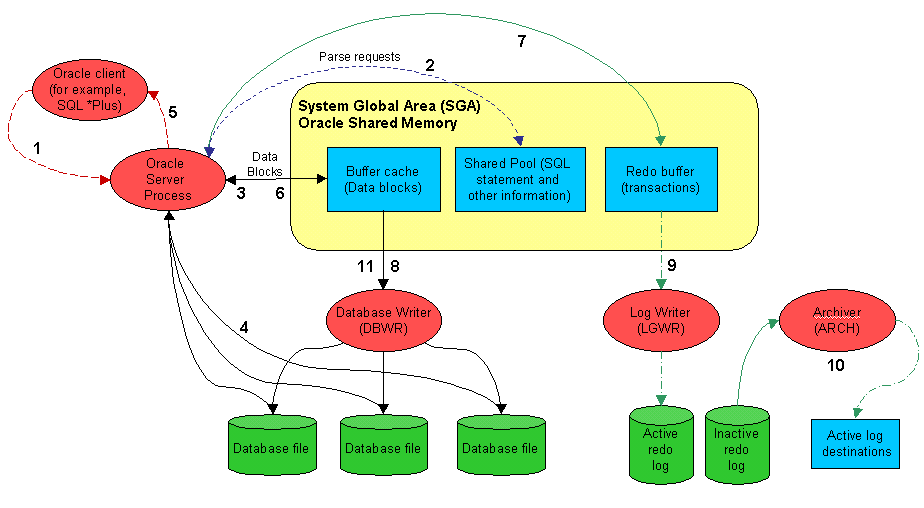
Để hiểu rõ hơn về các bước thực hiện kết nối, ta hãy xem xét một ví dụ mô tả việc kết nối tới Oracle database thực hiện bởi một user tại một máy tính khác có kết nối tới máy tính mà Oracle server đang chạy trên đó. Việc kết nối được thực hiện thông qua đường mạng bằng cách sử dụng dịch vụ Oracle Net8.

1. Tại máy chủ, cần đảm bảo Oracle server đang chạy và sẵn sàng đón nhận các tín hiệu từ phía Client. Máy chủ này được gọi là *host* hay *database server*.
2. Tại một máy trạm có chạy các ứng dụng (gọi là *local machine* hay *client workstation*) sẽ thực hiện các user process để kết nối tới database. Client application thực hiện thiết lập một kết nối tới server thông qua Net driver.
3. Máy chủ server trên đó có các Net driver. Server sẽ thực hiện việc nghe và dò tìm tất cả các yêu cầu gửi đến từ phía client và sau đó sẽ tạo một server process tương ứng với user process.
4. Khi user thực hiện một câu lệnh SQL hay commit một transaction. Ví dụ như user update dữ liệu trên một dòng trong một table.
5. Server process sẽ nhận về câu lệnh gửi tới từ Client, kiểm tra và phân tích câu lệnh, việc này được thực hiện trong shared pool. Tiếp theo đó, Server process sẽ kiểm tra quyền truy nhập dữ liệu của user.
6. Server process trả về các giá trị dữ liệu yêu cầu từ các dữ liệu có trong datafile hay trong system global area.
7. Server process thay đổi các dữ liệu có trong data buffer cache. DBW*n* process ghi lại các blocks đã thay đổi ra ổ đĩa. LGWR process  sẽ ghi lại ngay lập tức các bản ghi thay đổi vào online redo log file ngay khi transaction được commit.
8. Trong trường hợp transaction thực hiện thành công, server process sẽ gửi thông báo hoàn tất qua đường mạng tới Client. Ngược lại, sẽ có một error message gửi tới Client.

### 3. Áp dụng kiến trúc Cơ sở dữ liệu Oracle để hiểu hoạt động SELECT, INSERT, UPDATE, DELETE

Sơ đồ sau minh họa một số thành phần cơ bản của máy chủ Oracle ở các lớp bộ nhớ, cơ sở dữ liệu và đĩa. Kiến trúc này đã được sử dụng để thiết kế Spotlight trên trang chủ Oracle.

Các con số cho biết thứ tự của luồng thông tin.



Các nhãn được đánh số trong sơ đồ kiến ​​trúc Oracle tương ứng với các hoạt động sau:

1. Chương trình máy khách (ví dụ: SQL \* PLUS, Oracle Power Objects hoặc một số công cụ khác) gửi một câu lệnh **SELECT \* from CUSTOMER where CUST\_ID<1000** tới server process.
2. Server process tìm kiếm trong shared pool để tìm một câu lệnh SQL đã phân tích phù hợp. Nếu không tìm thấy, Server process phân tích cú pháp SQL và chèn câu lệnh SQL vào shared pool.
3. Server process tìm kiếm trong bộ đệm cache các block dữ liệu được yêu cầu. Nếu được tìm thấy, block dữ liệu phải được chuyển đến phần cuối được sử dụng gần đây nhất của danh sách Ít được sử dụng Gần đây nhất (Least Recently Used - LRU).
4. Nếu block không thể được tìm thấy trong bộ đệm cache, server process phải tìm nạp nó từ datafile trên đĩa. Điều này yêu cầu một I/O đĩa.
5. Server process trả về các row được truy xuất cho tiến trình máy khách. Điều này có thể liên quan đến  mạng hoặc sự chậm trễ liên lạc.
6. Khi máy khách đưa ra câu lệnh **UPDATE CUSTOMER set name='BINH\_NEW' WHERE CUST\_ID<1234**, quá trình phân tích cú pháp SQL và truy xuất các row được cập nhật phải xảy ra. Sau đó, câu lệnh cập nhật sẽ thay đổi các block có liên quan trong bộ nhớ dùng chung (shared memory) và cập nhật các mục nhập trong bộ đệm rollback segment.
7. Câu lệnh cập nhật cũng tạo một mục nhập trong redo log ghi lại chi tiết giao dịch.
8. Background Process của database-writer sao chép các block đã sửa đổi từ data buffer cache vào các tệp cơ sở dữ liệu. Phiên Oracle thực hiện cập nhật không phải đợi điều này xảy ra.
9. Khi câu lệnh COMMIT được đưa ra, tiến trình log writer (LGWR) phải sao chép nội dung của redo log buffer vào redo log file. Câu lệnh COMMIT không trả lại quyền điều khiển các block đó cho phiên Oracle cho đến khi quá trình ghi này hoàn tất.
10. Nếu chạy ở chế độ ARCHIVELOG, trình lưu trữ sẽ sao chép toàn bộ các redo log đến archive. Redo log không đủ điều kiện để sử dụng lại cho đến khi nó đã được lưu trữ (archived).
11. Trong các khoảng thời gian đều đặn, hoặc khi xảy ra quá trình redo log switch, Oracle thực hiện checkpoint. Checkpoint yêu cầu tất cả các block đã sửa đổi (dirty block) trong bộ đệm đệm phải được ghi vào đĩa. Không thể sử dụng lại redo log file cho đến khi checkpoint hoàn thành.

**ĐỌC THÊM:**

[*Oracle Database Architecture (Kiến trúc Oracle Database)*](https://www.tranvanbinh.vn/2017/04/kien-truc-oracle-database-server.html)

[*Kiến trúc Cơ sở dữ liệu Oracle hoạt động như thế nào Khi thực hiện các thao tác SELECT, INSERT, UPDATE, DELETE ?*](https://www.tranvanbinh.vn/2021/02/kien-truc-co-so-du-lieu-oracle-hoat-ong.html)

[*Redo record (redo entry) trong REDO LOG có những gì?*](https://www.tranvanbinh.vn/2020/08/bi-mat-su-dung-dbmslogmnr-e-hieu-sau-ve.html)

## TÀI LIỆU THAM KHẢO:

<https://www.tranvanbinh.vn/2020/02/chuong-2-cac-thanh-phan-kien-truc.html>

<https://www.tranvanbinh.vn/2021/02/kien-truc-co-so-du-lieu-oracle-hoat-ong.html>

<https://www.tranvanbinh.vn/2020/08/bi-mat-su-dung-dbmslogmnr-e-hieu-sau-ve.html>

<https://www.tranvanbinh.vn/2017/04/kien-truc-oracle-database-server.html>

<https://www.tranvanbinh.vn/2019/12/hoc-quan-tri-co-so-du-lieu-12c-trong-21.html>

<https://www.tranvanbinh.vn/2022/01/redo-log-cau-hinh-size-nhu-nao-la-toi-uu.html>

<https://www.tranvanbinh.vn/2020/02/tutorial-dba-chuong-8-quan-ly-redo-log.html>

# BÀI 3: CÔNG CỤ QUẢN TRỊ ORACLE DATABASE

Hướng dẫn sử dụng thủ tục MonitorDB\_Daily để giám sát tải CSDL Oracle (khóa trước) <https://youtu.be/V6weg7WV0xY>

## Tài liệu tham khảo:

TOAD: <https://www.tranvanbinh.vn/2020/01/toad-cong-cu-cho-oracle-dba-tot-nhat.html>

SQL\*Plus: <https://www.tranvanbinh.vn/2020/01/sqlplus-mot-so-cau-lenh-hay-su-dung.html>

<https://www.tranvanbinh.vn/2020/01/toad-cong-cu-cho-oracle-dba-tot-nhat.html>

SQL Navigator: <https://www.tranvanbinh.vn/2020/01/1_9.html>

Oracle SQl Developer: <https://www.tranvanbinh.vn/2020/01/huong-dan-su-dung-cong-cu-oracle-sql.html>

PL/SQL Developer: <https://www.tranvanbinh.vn/2023/06/cai-at-plsql-developer-tren-windows.html>

Oracle Enterprise Manager Express 19c: <https://www.tranvanbinh.vn/2020/06/enterprise-manager-database-express.html>

Oracle Cloud Control: <https://www.tranvanbinh.vn/2020/03/cai-at-oracle-enterprise-manager-cloud.html>

Cài đặt Oracle Database:

<https://www.tranvanbinh.vn/2021/11/cai-at-oracle-database-19c-va-tool-sql.html>

<https://www.tranvanbinh.vn/2021/01/cai-at-oracle-database-19c-tren-oracle.html>

Mọi người dựng lab máy ảo theo link này nhé: Nên dùng Lab 19c (nếu RAM PC >=8GB) <https://www.tranvanbinh.vn/2021/10/huong-dan-dung-may-ao-lab-oracle.html>

Nếu không chạy được Lab 19c (hay giật, lag) thì dùng Lab 12c: <https://www.tranvanbinh.vn/2020/06/huong-dung-may-ao-virtualbox-cho-lab.html>

Ngoài ra cài đặt thêm các công cụ khác là Oracle Client 12c, TOAD, SecureSCT, Putty, WinSCP, tất cả công cụ ở đây: <https://drive.google.com/drive/folders/1lAJSiGhsKA6F20LA9JMv8Py2o_kvS-hy?usp=sharing>

Các bộ cài khác (bộ cài, patch OS, Database, Weblogic):https://drive.google.com/drive/folders/1kvlkTuIfh61zverXjiHaXHXDS2t6yBEK?usp=sharing

<https://edelivery.oracle.com/EPD/Search/handle_go>

# BÀI 4: QUẢN LÝ INSTANCE

## Bật/tắt Instance

--1. Check

SQL> SELECT database\_status FROM v$instance;

-- 2. Startup

STARTUP [FORCE] [RESTRICT] [PFILE=filename]

[EXCLUSIVE | PARALLEL | SHARED]

[OPEN [RECOVER][database]|MOUNT |NOMOUNT]

ALTER database { MOUNT | OPEN |OPEN READ ONLY | OPEN READ WRITTE}

-- 3. Shutdown

SHUTDOWN [NORMAL | TRANSACTIONAL | IMMEDIATE | ABORT ]

-- 4.Suspend, resume

ALTER system { SUSPEND | RESUME }

-- 5. Automatically start Instance

/var/opt/oracle/oratab (OR /etc/oratab)

-- Automatically start Oracle 11g on solaris after server reboot (N), disable (Y)

+ASM2:/oracle/grid:N # line added by Agent

fpdb:/oracle/db:N # line added by Agent

select \* from dictionary;

select \* from dba\_users where account\_status='OPEN';

select \* from dba\_tables;

select \* from dba\_segments;

select \* from v$lock;

select \* from v$fixed\_table where name like 'GV%';

## Thay đổi bộ nhớ:

Check select \* from v$sga\_dynamic\_components;

PGA usage statistics:

select \* from v$pgastat;

- Determine a good setting for pga\_aggregate\_target:

select \* from v$pga\_target\_advice order by pga\_target\_for\_estimate;

- Show the maximum PGA usage per process:

select max(pga\_used\_mem), max(pga\_alloc\_mem), max(pga\_max\_mem) from v$process;

SQL> select \* from v$sga\_target\_advice order by sga\_size;

select \* from V$SGASTAT;

Thay đổi tham số sga\_max\_size, SGA\_TARGET

SQL> show parameter sga;

SQL> alter system set sga\_max\_size=8G scope=spfile sid='\*'

$ CRM\_HOME/bin/srvctl stop database -d card

$ CRM\_HOME/bin/srvctl start database -d card

SQL> alter system set sga\_target=7680M scope=spfile sid='\*'

1. THAY ĐỔI THAM SỐ

· Kết nối đến instance ASM:

$ . grid

$ sqlplus / as sysdba

· Thay đổi tham số memory\_max\_target trên các instance ASM:

Trên instance ASM của node 1:

SQL> alter system set memory\_max\_target =1040M scope=’spfile’ SID=’+ASM1’;

Trên instance ASM của node 2:

SQL> alter system set memory\_max\_target =1040M scope=’spfile’ SID=’+ASM2’;

· Thay đổi tham số memory\_target trên các instance ASM:

Trên instance ASM của node 1:

SQL> alter system set memory\_target=1040M scope=’spfile’ SID=’+ASM1’;

Trên instance ASM của node 2:

SQL> alter system set memory\_target=1040M scope=’spfile’ SID=’+ASM2’;

· Thay đổi tham số sga\_max\_size trên các instance ASM:

Trên instance ASM của node 1:

SQL> alter system set sga\_max\_size =1040M scope=’spfile’ SID=’+ASM1’;

Trên instance ASM của node 2:

SQL> alter system set sga\_max\_size =1040M scope=’spfile’ SID=’+ASM2’;

2.KHỞI ĐỘNG LẠI RAC

· Shutdown instance database trên node 1:

SQL> shutdown immediate

+ Shutdown instance ASM trên node 1:

SQL> connect / as sysasm

SQL> shutdown abort

· Startup instance ASM trên node 1:

SQL> startup

+ Startup instance database trên node 1:

SQL> startup

· Kiểm tra tham số memory\_target, memory\_max\_target sau khi instance database, ASM trên node 1 được bật lên

SQL> show parameter memory;

$ crsctl status resource –t

· Sau khi các tham số memory\_target, memory\_max\_target được thay đổi đúng thì thực hiện restart lại node 2 theo các bước tương tự trên.

3. KIỂM TRA LẠI

Kiểm tra CSDL theo khi thay đổi tham số

Lưu ý: Do OCS, Voting lưu trong diskgroup do đó phải bật instance ASM thì CRS mới online được (CSS thì luôn online)

Một số lệnh bổ sung  
// start, stop all service clusterware

root# /u01/grid/bin/crs\_start -all

root# /u01/grid/bin/crs\_stop -all

// Kiem tra CRS

$ crs\_stat -a

alter system pga\_aggregate\_target=200M;

## Tài liệu tham khảo:

<https://www.tranvanbinh.vn/p/admin-van-hanh.html>

<https://www.tranvanbinh.vn/2020/11/bat-tat-instance.html>

<https://www.tranvanbinh.vn/2020/11/bat-tat-db.html>

<https://www.tranvanbinh.vn/2022/04/bi-kip-tat-instance-oracle-rac-db-khung.html>

<https://www.tranvanbinh.vn/2020/02/tutorial-dba-co-ban-chuong-5-quan-tri.html>

<https://www.tranvanbinh.vn/2020/11/quan-tri-spfile.html>

<https://www.tranvanbinh.vn/p/rac-quan-ly-database-instance-rac.html>

<https://www.tranvanbinh.vn/p/12c-van-hanh.html>

# BÀI 5: QUẢN LÝ MẠNG

## TNS Name:

db19c\_cdb\_service=

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.182.128)(PORT = 1521))

(CONNECT\_DATA =

(SERVICE\_NAME = orcl)

)

)

db19c\_cdb\_SID=

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.182.128)(PORT = 1521))

(CONNECT\_DATA =

(SID = orcl)

)

)

db19c\_pdb1=

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.182.128)(PORT = 1521))

(CONNECT\_DATA =

(SERVICE\_NAME = pdb1)

)

)

--RAC

DBAVIET=

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.186.218)(PORT = 1521))

(ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.186.222)(PORT = 1521))

(LOAD\_BALANCE = yes)

(FAILOVER = yes)

(CONNECT\_DATA =

(SERVER = DEDICATED)

(SERVICE\_NAME = dbaviet)

)

)

DBAVIET1\_SID=

(DESCRIPTION=

(ADDRESS=

(PROTOCOL=TCP)

(HOST=192.168.186.218)

(PORT=1521)

)

(CONNECT\_DATA=

(SERVER=dedicated)

(SID=dbaviet1)

)

)

DBAVIET2\_SID=

(DESCRIPTION=

(ADDRESS=

(PROTOCOL=TCP)

(HOST=192.168.186.222)

(PORT=1521)

)

(CONNECT\_DATA=

(SERVER=dedicated)

(SID=dbaviet2)

)

## Database link:

select \* from c##oaz20.t1;

insert into c##oaz20.t1 values(1);

commit;

create table c##oaz20.t1 (id number);

--1 vs 11

select \* from c##oaz20.t1;

select \* from oaz20.oaz20\_t1;

select \* from c##oaz20.c##oaz20\_t1;

CREATE PUBLIC DATABASE LINK dbl\_pdb1

CONNECT TO c##oaz20

IDENTIFIED BY oracle

USING ' (DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.182.128)(PORT = 1521))

(CONNECT\_DATA =

(SERVICE\_NAME = pdb1)

)

)

';

select \* from dual@dbl\_pdb1;

select \* from c##oaz20.c##oaz20\_t1@dbl\_pdb1;

select \* from oaz20.oaz20\_t1@dbl\_pdb1;

create table xxx\_t1 as select \* from oaz20.oaz20\_t1@dbl\_pdb1;

select \* from xxx\_t1;

## Tài liệu tham khảo:

<https://www.tranvanbinh.vn/2021/05/cau-hinh-mang-trong-oracle-database.html>

<https://www.tranvanbinh.vn/2021/08/ket-noi-toi-container-databases-cdb-va.html>

<https://www.tranvanbinh.vn/2020/12/quan-tri-database-link-trong-oracle.html>

<https://www.tranvanbinh.vn/2020/12/thu-tuc-tao-dblink-bang-user-khac.html>

# Bài 6: QUẢN LÝ NGƯỜI DÙNG

## Người dùng, quyền:

-- 1.TAO USER COMMON c##oaz20

create user c##oaz20 identified by oracle container=all;

-- Tạo xong ấn F4 kiểm tra lại

grant connect, RESOURCE , dba to c##oaz20 container=all;

-- GRANT

-- System Privilege

select \* from dba\_sys\_privs where grantee ='C##OAZ20';

-- Object privilege

select \* from dba\_tab\_privs where grantee='C##OAZ20';

-- Role:DBA, CONNECT, RESOURCE

select \* from dba\_role\_privs where grantee='C##OAZ20';

-- c##oaz20: Session

select \* from session\_privs;

select \* from session\_roles;

-- Gan quyen

grant select on SYS.FGA\_LOG$ to c##oaz20 WITH ADMIN OPTION;

grant alter any procedure to c##oaz20 container=all;

-- Quyen view moi package:

grant SELECT ANY DICTIONARY to c##oaz20;

grant debug any procedure to c##oaz20;

--REVOKE

revoke debug any procedure from c##oaz20;

-- Vao PDB1 bang c##oaz20

-- 2.TAO USER LOCAL TRONG PDB1

-- Vao c##az13 TNS db19c\_pdb1 de tao:

create user oaz20 identified by oracle;

grant connect, resource, dba to oaz20;

-- Tuong tu

## -- PROFILE

create PROFILE "PROFILE\_USER" LIMIT

SESSIONS\_PER\_USER 3

CPU\_PER\_SESSION UNLIMITED

CPU\_PER\_CALL UNLIMITED

CONNECT\_TIME UNLIMITED

IDLE\_TIME UNLIMITED

LOGICAL\_READS\_PER\_SESSION UNLIMITED

LOGICAL\_READS\_PER\_CALL UNLIMITED

COMPOSITE\_LIMIT UNLIMITED

PRIVATE\_SGA UNLIMITED

FAILED\_LOGIN\_ATTEMPTS UNLIMITED

INACTIVE\_ACCOUNT\_TIME UNLIMITED

PASSWORD\_LIFE\_TIME 45

PASSWORD\_REUSE\_TIME UNLIMITED

PASSWORD\_REUSE\_MAX UNLIMITED

PASSWORD\_LOCK\_TIME UNLIMITED

PASSWORD\_GRACE\_TIME UNLIMITED

PASSWORD\_VERIFY\_FUNCTION Default;

create PROFILE "PROFILE\_APP" LIMIT

SESSIONS\_PER\_USER UNLIMITED

CPU\_PER\_SESSION UNLIMITED

CPU\_PER\_CALL UNLIMITED

CONNECT\_TIME UNLIMITED

IDLE\_TIME UNLIMITED

LOGICAL\_READS\_PER\_SESSION UNLIMITED

LOGICAL\_READS\_PER\_CALL UNLIMITED

COMPOSITE\_LIMIT UNLIMITED

PRIVATE\_SGA UNLIMITED

FAILED\_LOGIN\_ATTEMPTS UNLIMITED

INACTIVE\_ACCOUNT\_TIME UNLIMITED

PASSWORD\_LIFE\_TIME UNLIMITED

PASSWORD\_REUSE\_TIME UNLIMITED

PASSWORD\_REUSE\_MAX UNLIMITED

PASSWORD\_LOCK\_TIME UNLIMITED

PASSWORD\_GRACE\_TIME UNLIMITED

PASSWORD\_VERIFY\_FUNCTION Default;

ALTER USER oaz20

PROFILE PROFILE\_USER;

-- Gan quyen tu oaz14 --> tab18

--grant UNLIMITED TABLESPACE to oaz20;

select 'grant ' || privilege || ' to oaz20;' from dba\_sys\_privs

where grantee='OAZ18';

--grant ALTER on HR.EMPLOYEES to tab18;

--grant DELETE on HR.EMPLOYEES to tab18;

--grant INDEX on HR.EMPLOYEES to tab18;

--grant INSERT on HR.EMPLOYEES to tab18;

--grant SELECT on HR.EMPLOYEES to tab18;

--grant UPDATE on HR.EMPLOYEES to tab18;

--grant REFERENCES on HR.EMPLOYEES to tab18;

--grant READ on HR.EMPLOYEES to tab18;

--grant ON COMMIT REFRESH on HR.EMPLOYEES to tab18;

--grant QUERY REWRITE on HR.EMPLOYEES to tab18;

--grant DEBUG on HR.EMPLOYEES to tab18;

--grant FLASHBACK on HR.EMPLOYEES to tab18;

select 'grant ' || privilege || ' on ' || owner||'.' || table\_name ||' to oaz20;' from dba\_tab\_privs

where grantee='oaz20';

grant select on sys.fga\_log$ to oaz20;

select \* from dba\_tab\_privs

where grantee='oaz20';

## ---ROLE PDB----

CREATE ROLE ROLE\_oaz20 NOT IDENTIFIED;

alter role ROLE\_oaz20 identified by oracle;

grant select any table, drop any table, insert any table, delete any table to role\_oaz20;

grant all on sys.fga\_log$ to role\_oaz20;

grant role\_oaz20 to oaz20;

grant role\_oaz20 to c##oaz20;

## ---ROLE CDB----

CREATE ROLE c##ROLE\_oaz20 NOT IDENTIFIED;

alter role c##ROLE\_oaz20 identified by oracle;

grant select any table, drop any table, insert any table, delete any table to c##role\_oaz20;

grant all on sys.fga\_log$ to c##role\_oaz20;

grant c##role\_oaz20 to c##oaz20;

-- Vao PDB

grant c##role\_oaz20 to oaz20;

## Role DBA

CREATE ROLE role20 NOT IDENTIFIED;

--dba

-- Object privileges granted to role20

GRANT DELETE, INSERT, SELECT, UPDATE ON XDB.X$PT46MP5MEPA004NE0KWN0SK0K1LN TO role20;

GRANT EXECUTE ON ORDSYS.CARTRIDGE TO role20 ;

GRANT EXECUTE ON SYS.ADR\_HOME\_T TO role20 ;

GRANT EXECUTE ON SYS.ADR\_INCIDENT\_CORR\_KEYS\_T TO role20 ;

GRANT EXECUTE ON SYS.ADR\_INCIDENT\_FILES\_T TO role20 ;

GRANT EXECUTE ON SYS.ADR\_INCIDENT\_INFO\_T TO role20 ;

GRANT EXECUTE ON SYS.ADR\_INCIDENT\_T TO role20 ;

GRANT EXECUTE ON SYS.ADR\_LOG\_MSG\_ARGS\_T TO role20 ;

GRANT EXECUTE ON SYS.ADR\_LOG\_MSG\_ARG\_T TO role20 ;

GRANT EXECUTE ON SYS.ADR\_LOG\_MSG\_ECID\_T TO role20 ;

GRANT EXECUTE ON SYS.ADR\_LOG\_MSG\_ERRID\_T TO role20 ;

GRANT EXECUTE ON SYS.ADR\_LOG\_MSG\_SUPPL\_ATTRS\_T TO role20 ;

GRANT EXECUTE ON SYS.ADR\_LOG\_MSG\_SUPPL\_ATTR\_T TO role20 ;

GRANT EXECUTE ON SYS.ADR\_LOG\_MSG\_T TO role20 ;

GRANT EXECUTE ON SYS.AS\_REPLAY TO role20 ;

GRANT SELECT ON SYS.CDB\_XS\_ACTIVE\_SESSIONS TO role20 ;

GRANT SELECT ON SYS.CDB\_XS\_SESSIONS TO role20 ;

GRANT SELECT ON SYS.CDB\_XS\_SESSION\_NS\_ATTRIBUTES TO role20 ;

GRANT SELECT ON SYS.CDB\_XS\_SESSION\_ROLES TO role20 ;

GRANT DELETE, INSERT, SELECT, UPDATE ON SYS.DBA\_REGISTRY\_SQLPATCH TO role20 ;

GRANT DELETE, INSERT, SELECT, UPDATE ON SYS.DBA\_REGISTRY\_SQLPATCH\_RU\_INFO TO role20 ;

GRANT SELECT ON SYS.DBA\_XS\_ACTIVE\_SESSIONS TO role20 ;

GRANT SELECT ON SYS.DBA\_XS\_SESSIONS TO role20 ;

GRANT SELECT ON SYS.DBA\_XS\_SESSION\_NS\_ATTRIBUTES TO role20 ;

GRANT SELECT ON SYS.DBA\_XS\_SESSION\_ROLES TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_ADR TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_ADR\_APP TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_APP\_CONT TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_APP\_CONT\_ADMIN TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_AUTO\_INDEX TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_AUTO\_SQLTUNE TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_AUTO\_TASK\_ADMIN TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_AUTO\_TASK\_IMMEDIATE TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_AWR\_WAREHOUSE\_SERVER TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_AWR\_WAREHOUSE\_SOURCE TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_CACHEUTIL TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_DNFS TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_FEATURE\_USAGE\_REPORT TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_FLASHBACK TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_FLASHBACK\_ARCHIVE TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_FS TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_GSM\_FIX TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_HANG\_MANAGER TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_HM TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_ILM\_ADMIN TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_INMEMORY\_ADMIN TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_IR TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_LOGSTDBY TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_MANAGEMENT\_BOOTSTRAP TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_MEMOPTIMIZE\_ADMIN TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_MONITOR TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_NETWORK\_ACL\_ADMIN TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_PERF TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_RAT\_MASK TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_RESULT\_CACHE TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_RESUMABLE TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_ROLLING TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_SERVER\_ALERT TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_SERVER\_TRACE TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_SERVICE TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_SERVICE\_CONST TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_SERVICE\_ERR TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_SERVICE\_PRVT TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_STORAGE\_MAP TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_TDB TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_TNS TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_UADV\_ARR TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_UMF TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_UNDO\_ADV TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_WORKLOAD\_CAPTURE TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_WORKLOAD\_REPLAY TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_WORKLOAD\_REPOSITORY TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_WRR\_STATE TO role20 ;

GRANT EXECUTE ON SYS.DBMS\_XS\_PRINCIPALS TO role20 ;

GRANT EXECUTE ON SYS.GRANT\_RDF\_OWNER\_DR\_PRIVS TO role20 ;

GRANT EXECUTE ON SYS.HM\_SQLTK\_INTERNAL TO role20 ;

GRANT EXECUTE ON SYS.LOAD\_UNDO\_STAT TO role20 ;

GRANT DELETE, INSERT, SELECT, UPDATE ON SYS.MAP\_OBJECT TO role20 ;

GRANT EXECUTE ON SYS.OJDS\_CONTEXT TO role20 ;

GRANT EXECUTE ON SYS.ORD\_ADMIN TO role20 ;

GRANT EXECUTE ON SYS.OUTLN\_PKG TO role20 ;

GRANT EXECUTE ON SYS.RDF\_NETWORK\_CREATOR\_PRIVS TO role20 ;

GRANT EXECUTE ON SYS.RESET\_UNDO\_STAT TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_ADR\_CONTROL TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_ADR\_CONTROL\_AUX TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_ADR\_INVALIDATION TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_ALERT\_EXT TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_AMS\_XACTION TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_DDE\_USER\_ACTION TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_DDE\_USER\_ACTION\_DEF TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_DDE\_USR\_ACT\_PARAM TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_DDE\_USR\_ACT\_PARAM\_DEF TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_DDE\_USR\_INC\_ACT\_MAP TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_DDE\_USR\_INC\_TYPE TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_DFW\_CONFIG\_CAPTURE TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_DFW\_CONFIG\_ITEM TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_DFW\_PATCH\_CAPTURE TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_DFW\_PATCH\_ITEM TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_DFW\_PURGE TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_DFW\_PURGE\_ITEM TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_DIAGV\_INCIDENT TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_DIR\_EXT TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_EM\_DIAG\_JOB TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_EM\_TARGET\_INFO TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_EM\_USER\_ACTIVITY TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_HM\_FDG\_SET TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_HM\_FINDING TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_HM\_INFO TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_HM\_MESSAGE TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_HM\_RECOMMENDATION TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_HM\_RUN TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_INCCKEY TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_INCIDENT TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_INCIDENT\_FILE TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_INC\_METER\_CONFIG TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_INC\_METER\_IMPT\_DEF TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_INC\_METER\_INFO TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_INC\_METER\_PK\_IMPTS TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_INC\_METER\_SUMMARY TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_IPS\_CONFIGURATION TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_IPS\_FILE\_COPY\_LOG TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_IPS\_FILE\_METADATA TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_IPS\_PACKAGE TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_IPS\_PACKAGE\_FILE TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_IPS\_PACKAGE\_HISTORY TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_IPS\_PACKAGE\_INCIDENT TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_IPS\_PKG\_UNPACK\_HIST TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_IPS\_PROGRESS\_LOG TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_IPS\_REMOTE\_PACKAGE TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_LOG\_EXT TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_PDB\_PROBLEM TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_PDB\_SPACE\_MGMT TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_PICKLEERR TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_PROBLEM TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_RELMD\_EXT TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_SWEEPERR TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VADR\_CONTROL TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VEM\_USER\_ACTLOG TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VEM\_USER\_ACTLOG1 TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VHM\_RUN TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VIEW TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VIEWCOL TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VINCIDENT TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VINCIDENT\_FILE TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VINC\_METER\_INFO TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VIPS\_FILE\_COPY\_LOG TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VIPS\_FILE\_METADATA TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VIPS\_PACKAGE\_FILE TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VIPS\_PACKAGE\_HISTORY TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VIPS\_PACKAGE\_MAIN\_INT TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VIPS\_PACKAGE\_SIZE TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VIPS\_PKG\_FILE TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VIPS\_PKG\_INC\_CAND TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VIPS\_PKG\_INC\_DTL TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VIPS\_PKG\_INC\_DTL1 TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VIPS\_PKG\_MAIN\_PROBLEM TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VNOT\_EXIST\_INCIDENT TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VPDB\_PROBLEM TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VPROBLEM TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VPROBLEM1 TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VPROBLEM2 TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VPROBLEM\_BUCKET TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VPROBLEM\_BUCKET1 TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VPROBLEM\_BUCKET\_COUNT TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VPROBLEM\_INT TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VPROBLEM\_LASTINC TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VSHOWCATVIEW TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VSHOWINCB TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VSHOWINCB\_I TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_VTEST\_EXISTS TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_V\_ACTINC TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_V\_ACTPROB TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_V\_INCCOUNT TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_V\_INCFCOUNT TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_V\_INC\_METER\_INFO\_PROB TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_V\_IPSPRBCNT TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_V\_IPSPRBCNT1 TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_V\_NFCINC TO role20 ;

GRANT SELECT ON SYS.V\_$DIAG\_V\_SWPERRCOUNT TO role20 ;

GRANT INSERT, SELECT, UPDATE ON SYS.XSDB$SCHEMA\_ACL TO role20 ;

GRANT EXECUTE ON SYS.XS\_MTCACHE\_INT TO role20 ;

GRANT EXECUTE ON XDB.DBMS\_CSX\_ADMIN TO role20 ;

GRANT EXECUTE ON XDB.DBMS\_XDBT TO role20 ;

GRANT EXECUTE ON XDB.DBMS\_XDB\_ADMIN TO role20 ;

GRANT DELETE, INSERT, SELECT, UPDATE ON XDB.XDB$ACL TO role20 ;

GRANT DELETE, INSERT, SELECT, UPDATE ON XDB.XDB$CHECKOUTS TO role20 ;

GRANT DELETE, INSERT, SELECT, UPDATE ON XDB.XDB$CONFIG TO role20 ;

GRANT DELETE, INSERT, SELECT, UPDATE ON XDB.XDB$D\_LINK TO role20 ;

GRANT DELETE, INSERT, SELECT, UPDATE ON XDB.XDB$H\_INDEX TO role20 ;

GRANT DELETE, INSERT, SELECT, UPDATE ON XDB.XDB$H\_LINK TO role20 ;

GRANT DELETE, INSERT, SELECT, UPDATE ON XDB.XDB$NLOCKS TO role20 ;

GRANT DELETE, INSERT, SELECT, UPDATE ON XDB.XDB$RESCONFIG TO role20 ;

GRANT DELETE, INSERT, SELECT, UPDATE ON XDB.XDB$RESOURCE TO role20 ;

-- System privileges granted to role20

GRANT ADMINISTER ANY SQL TUNING SET TO role20 ;

GRANT ADMINISTER DATABASE TRIGGER TO role20 ;

BEGIN

SYS.DBMS\_RESOURCE\_MANAGER\_PRIVS.GRANT\_SYSTEM\_PRIVILEGE

(GRANTEE\_NAME => 'role20',

PRIVILEGE\_NAME => 'ADMINISTER\_RESOURCE\_MANAGER',

ADMIN\_OPTION => FALSE);

END;

/

GRANT ADMINISTER SQL MANAGEMENT OBJECT TO role20 ;

GRANT ADMINISTER SQL TUNING SET TO role20 ;

GRANT ADVISOR TO role20 ;

GRANT ALTER ANY ANALYTIC VIEW TO role20 ;

GRANT ALTER ANY ASSEMBLY TO role20 ;

GRANT ALTER ANY ATTRIBUTE DIMENSION TO role20 ;

GRANT ALTER ANY CLUSTER TO role20 ;

GRANT ALTER ANY CUBE TO role20 ;

GRANT ALTER ANY CUBE BUILD PROCESS TO role20 ;

GRANT ALTER ANY CUBE DIMENSION TO role20 ;

GRANT ALTER ANY DIMENSION TO role20 ;

GRANT ALTER ANY EDITION TO role20 ;

BEGIN

SYS.DBMS\_RULE\_ADM.GRANT\_SYSTEM\_PRIVILEGE(

PRIVILEGE => SYS.DBMS\_RULE\_ADM.ALTER\_ANY\_EVALUATION\_CONTEXT,

GRANTEE => 'role20',

GRANT\_OPTION => FALSE);

END;

/

GRANT ALTER ANY HIERARCHY TO role20 ;

GRANT ALTER ANY INDEX TO role20 ;

GRANT ALTER ANY INDEXTYPE TO role20 ;

GRANT ALTER ANY LIBRARY TO role20 ;

GRANT ALTER ANY MATERIALIZED VIEW TO role20 ;

GRANT ALTER ANY MEASURE FOLDER TO role20 ;

GRANT ALTER ANY MINING MODEL TO role20 ;

GRANT ALTER ANY OPERATOR TO role20 ;

GRANT ALTER ANY OUTLINE TO role20 ;

GRANT ALTER ANY PROCEDURE TO role20 ;

GRANT ALTER ANY ROLE TO role20 ;

BEGIN

SYS.DBMS\_RULE\_ADM.GRANT\_SYSTEM\_PRIVILEGE(

PRIVILEGE => SYS.DBMS\_RULE\_ADM.ALTER\_ANY\_RULE,

GRANTEE => 'role20',

GRANT\_OPTION => FALSE);

END;

/

BEGIN

SYS.DBMS\_RULE\_ADM.GRANT\_SYSTEM\_PRIVILEGE(

PRIVILEGE => SYS.DBMS\_RULE\_ADM.ALTER\_ANY\_RULE\_SET,

GRANTEE => 'role20',

GRANT\_OPTION => FALSE);

END;

/

GRANT ALTER ANY SEQUENCE TO role20 ;

GRANT ALTER ANY SQL PROFILE TO role20 ;

GRANT ALTER ANY SQL TRANSLATION PROFILE TO role20 ;

GRANT ALTER ANY TABLE TO role20 ;

GRANT ALTER ANY TRIGGER TO role20 ;

GRANT ALTER ANY TYPE TO role20 ;

GRANT ALTER DATABASE TO role20 ;

GRANT ALTER LOCKDOWN PROFILE TO role20 ;

GRANT ALTER PROFILE TO role20 ;

GRANT ALTER RESOURCE COST TO role20 ;

GRANT ALTER ROLLBACK SEGMENT TO role20 ;

GRANT ALTER SESSION TO role20 ;

GRANT ALTER SYSTEM TO role20 ;

GRANT ALTER TABLESPACE TO role20 ;

GRANT ALTER USER TO role20 ;

GRANT ANALYZE ANY TO role20 ;

GRANT ANALYZE ANY DICTIONARY TO role20 ;

GRANT AUDIT ANY TO role20 ;

GRANT AUDIT SYSTEM TO role20 ;

GRANT BACKUP ANY TABLE TO role20 ;

GRANT BECOME USER TO role20 ;

GRANT CHANGE NOTIFICATION TO role20 ;

GRANT COMMENT ANY MINING MODEL TO role20 ;

GRANT COMMENT ANY TABLE TO role20 ;

GRANT CREATE ANALYTIC VIEW TO role20 ;

GRANT CREATE ANY ANALYTIC VIEW TO role20 ;

GRANT CREATE ANY ASSEMBLY TO role20 ;

GRANT CREATE ANY ATTRIBUTE DIMENSION TO role20 ;

GRANT CREATE ANY CLUSTER TO role20 ;

GRANT CREATE ANY CONTEXT TO role20 ;

GRANT CREATE ANY CREDENTIAL TO role20 ;

GRANT CREATE ANY CUBE TO role20 ;

GRANT CREATE ANY CUBE BUILD PROCESS TO role20 ;

GRANT CREATE ANY CUBE DIMENSION TO role20 ;

GRANT CREATE ANY DIMENSION TO role20 ;

GRANT CREATE ANY DIRECTORY TO role20 ;

GRANT CREATE ANY EDITION TO role20 ;

BEGIN

SYS.DBMS\_RULE\_ADM.GRANT\_SYSTEM\_PRIVILEGE(

PRIVILEGE => SYS.DBMS\_RULE\_ADM.CREATE\_ANY\_EVALUATION\_CONTEXT,

GRANTEE => 'role20',

GRANT\_OPTION => FALSE);

END;

/

GRANT CREATE ANY HIERARCHY TO role20 ;

GRANT CREATE ANY INDEX TO role20 ;

GRANT CREATE ANY INDEXTYPE TO role20 ;

GRANT CREATE ANY JOB TO role20 ;

GRANT CREATE ANY LIBRARY TO role20 ;

GRANT CREATE ANY MATERIALIZED VIEW TO role20 ;

GRANT CREATE ANY MEASURE FOLDER TO role20 ;

GRANT CREATE ANY MINING MODEL TO role20 ;

GRANT CREATE ANY OPERATOR TO role20 ;

GRANT CREATE ANY OUTLINE TO role20 ;

GRANT CREATE ANY PROCEDURE TO role20 ;

BEGIN

SYS.DBMS\_RULE\_ADM.GRANT\_SYSTEM\_PRIVILEGE(

PRIVILEGE => SYS.DBMS\_RULE\_ADM.CREATE\_ANY\_RULE,

GRANTEE => 'role20',

GRANT\_OPTION => FALSE);

END;

/

BEGIN

SYS.DBMS\_RULE\_ADM.GRANT\_SYSTEM\_PRIVILEGE(

PRIVILEGE => SYS.DBMS\_RULE\_ADM.CREATE\_ANY\_RULE\_SET,

GRANTEE => 'role20',

GRANT\_OPTION => FALSE);

END;

/

GRANT CREATE ANY SEQUENCE TO role20 ;

GRANT CREATE ANY SQL PROFILE TO role20 ;

GRANT CREATE ANY SQL TRANSLATION PROFILE TO role20 ;

GRANT CREATE ANY SYNONYM TO role20 ;

GRANT CREATE ANY TABLE TO role20 ;

GRANT CREATE ANY TRIGGER TO role20 ;

GRANT CREATE ANY TYPE TO role20 ;

GRANT CREATE ANY VIEW TO role20 ;

GRANT CREATE ASSEMBLY TO role20 ;

GRANT CREATE ATTRIBUTE DIMENSION TO role20 ;

GRANT CREATE CLUSTER TO role20 ;

GRANT CREATE CREDENTIAL TO role20 ;

GRANT CREATE CUBE TO role20 ;

GRANT CREATE CUBE BUILD PROCESS TO role20 ;

GRANT CREATE CUBE DIMENSION TO role20 ;

GRANT CREATE DATABASE LINK TO role20 ;

GRANT CREATE DIMENSION TO role20 ;

BEGIN

SYS.DBMS\_RULE\_ADM.GRANT\_SYSTEM\_PRIVILEGE(

PRIVILEGE => SYS.DBMS\_RULE\_ADM.CREATE\_EVALUATION\_CONTEXT\_OBJ,

GRANTEE => 'role20',

GRANT\_OPTION => FALSE);

END;

/

GRANT CREATE EXTERNAL JOB TO role20 ;

GRANT CREATE HIERARCHY TO role20 ;

GRANT CREATE INDEXTYPE TO role20 ;

GRANT CREATE JOB TO role20 ;

GRANT CREATE LIBRARY TO role20 ;

GRANT CREATE LOCKDOWN PROFILE TO role20 ;

GRANT CREATE MATERIALIZED VIEW TO role20 ;

GRANT CREATE MEASURE FOLDER TO role20 ;

GRANT CREATE MINING MODEL TO role20 ;

GRANT CREATE OPERATOR TO role20 ;

GRANT CREATE PLUGGABLE DATABASE TO role20 ;

GRANT CREATE PROCEDURE TO role20 ;

GRANT CREATE PROFILE TO role20 ;

GRANT CREATE PUBLIC DATABASE LINK TO role20 ;

GRANT CREATE PUBLIC SYNONYM TO role20 ;

GRANT CREATE ROLE TO role20 ;

GRANT CREATE ROLLBACK SEGMENT TO role20 ;

BEGIN

SYS.DBMS\_RULE\_ADM.GRANT\_SYSTEM\_PRIVILEGE(

PRIVILEGE => SYS.DBMS\_RULE\_ADM.CREATE\_RULE\_OBJ,

GRANTEE => 'role20',

GRANT\_OPTION => FALSE);

END;

/

BEGIN

SYS.DBMS\_RULE\_ADM.GRANT\_SYSTEM\_PRIVILEGE(

PRIVILEGE => SYS.DBMS\_RULE\_ADM.CREATE\_RULE\_SET\_OBJ,

GRANTEE => 'role20',

GRANT\_OPTION => FALSE);

END;

/

GRANT CREATE SEQUENCE TO role20 ;

GRANT CREATE SESSION TO role20 ;

GRANT CREATE SQL TRANSLATION PROFILE TO role20 ;

GRANT CREATE SYNONYM TO role20 ;

GRANT CREATE TABLE TO role20 ;

GRANT CREATE TABLESPACE TO role20 ;

GRANT CREATE TRIGGER TO role20 ;

GRANT CREATE TYPE TO role20 ;

GRANT CREATE USER TO role20 ;

GRANT CREATE VIEW TO role20 ;

GRANT DEBUG ANY PROCEDURE TO role20 ;

GRANT DEBUG CONNECT SESSION TO role20 ;

GRANT DELETE ANY CUBE DIMENSION TO role20 ;

GRANT DELETE ANY MEASURE FOLDER TO role20 ;

GRANT DELETE ANY TABLE TO role20 ;

BEGIN

SYS.DBMS\_AQADM.GRANT\_SYSTEM\_PRIVILEGE (

PRIVILEGE => 'DEQUEUE\_ANY',

GRANTEE => 'role20',

ADMIN\_OPTION => FALSE);

END;

/

GRANT DROP ANY ANALYTIC VIEW TO role20 ;

GRANT DROP ANY ASSEMBLY TO role20 ;

GRANT DROP ANY ATTRIBUTE DIMENSION TO role20 ;

GRANT DROP ANY CLUSTER TO role20 ;

GRANT DROP ANY CONTEXT TO role20 ;

GRANT DROP ANY CUBE TO role20 ;

GRANT DROP ANY CUBE BUILD PROCESS TO role20 ;

GRANT DROP ANY CUBE DIMENSION TO role20 ;

GRANT DROP ANY DIMENSION TO role20 ;

GRANT DROP ANY DIRECTORY TO role20 ;

GRANT DROP ANY EDITION TO role20 ;

BEGIN

SYS.DBMS\_RULE\_ADM.GRANT\_SYSTEM\_PRIVILEGE(

PRIVILEGE => SYS.DBMS\_RULE\_ADM.DROP\_ANY\_EVALUATION\_CONTEXT,

GRANTEE => 'role20',

GRANT\_OPTION => FALSE);

END;

/

GRANT DROP ANY HIERARCHY TO role20 ;

GRANT DROP ANY INDEX TO role20 ;

GRANT DROP ANY INDEXTYPE TO role20 ;

GRANT DROP ANY LIBRARY TO role20 ;

GRANT DROP ANY MATERIALIZED VIEW TO role20 ;

GRANT DROP ANY MEASURE FOLDER TO role20 ;

GRANT DROP ANY MINING MODEL TO role20 ;

GRANT DROP ANY OPERATOR TO role20 ;

GRANT DROP ANY OUTLINE TO role20 ;

GRANT DROP ANY PROCEDURE TO role20 ;

GRANT DROP ANY ROLE TO role20 ;

BEGIN

SYS.DBMS\_RULE\_ADM.GRANT\_SYSTEM\_PRIVILEGE(

PRIVILEGE => SYS.DBMS\_RULE\_ADM.DROP\_ANY\_RULE,

GRANTEE => 'role20',

GRANT\_OPTION => FALSE);

END;

/

BEGIN

SYS.DBMS\_RULE\_ADM.GRANT\_SYSTEM\_PRIVILEGE(

PRIVILEGE => SYS.DBMS\_RULE\_ADM.DROP\_ANY\_RULE\_SET,

GRANTEE => 'role20',

GRANT\_OPTION => FALSE);

END;

/

GRANT DROP ANY SEQUENCE TO role20 ;

GRANT DROP ANY SQL PROFILE TO role20 ;

GRANT DROP ANY SQL TRANSLATION PROFILE TO role20 ;

GRANT DROP ANY SYNONYM TO role20 ;

GRANT DROP ANY TABLE TO role20 ;

GRANT DROP ANY TRIGGER TO role20 ;

GRANT DROP ANY TYPE TO role20 ;

GRANT DROP ANY VIEW TO role20 ;

GRANT DROP LOCKDOWN PROFILE TO role20 ;

GRANT DROP PROFILE TO role20 ;

GRANT DROP PUBLIC DATABASE LINK TO role20 ;

GRANT DROP PUBLIC SYNONYM TO role20 ;

GRANT DROP ROLLBACK SEGMENT TO role20 ;

GRANT DROP TABLESPACE TO role20 ;

GRANT DROP USER TO role20 ;

GRANT EM EXPRESS CONNECT TO role20 ;

BEGIN

SYS.DBMS\_AQADM.GRANT\_SYSTEM\_PRIVILEGE (

PRIVILEGE => 'ENQUEUE\_ANY',

GRANTEE => 'role20',

ADMIN\_OPTION => FALSE);

END;

/

GRANT EXECUTE ANY ASSEMBLY TO role20 ;

GRANT EXECUTE ANY CLASS TO role20 ;

BEGIN

SYS.DBMS\_RULE\_ADM.GRANT\_SYSTEM\_PRIVILEGE(

PRIVILEGE => SYS.DBMS\_RULE\_ADM.EXECUTE\_ANY\_EVALUATION\_CONTEXT,

GRANTEE => 'role20',

GRANT\_OPTION => FALSE);

END;

/

GRANT EXECUTE ANY INDEXTYPE TO role20 ;

GRANT EXECUTE ANY LIBRARY TO role20 ;

GRANT EXECUTE ANY OPERATOR TO role20 ;

GRANT EXECUTE ANY PROCEDURE TO role20 ;

GRANT EXECUTE ANY PROGRAM TO role20 ;

BEGIN

SYS.DBMS\_RULE\_ADM.GRANT\_SYSTEM\_PRIVILEGE(

PRIVILEGE => SYS.DBMS\_RULE\_ADM.EXECUTE\_ANY\_RULE,

GRANTEE => 'role20',

GRANT\_OPTION => FALSE);

END;

/

BEGIN

SYS.DBMS\_RULE\_ADM.GRANT\_SYSTEM\_PRIVILEGE(

PRIVILEGE => SYS.DBMS\_RULE\_ADM.EXECUTE\_ANY\_RULE\_SET,

GRANTEE => 'role20',

GRANT\_OPTION => FALSE);

END;

/

GRANT EXECUTE ANY TYPE TO role20 ;

GRANT EXECUTE ASSEMBLY TO role20 ;

GRANT EXPORT FULL DATABASE TO role20 ;

GRANT FLASHBACK ANY TABLE TO role20 ;

GRANT FLASHBACK ARCHIVE ADMINISTER TO role20 ;

GRANT FORCE ANY TRANSACTION TO role20 ;

GRANT FORCE TRANSACTION TO role20 ;

GRANT GLOBAL QUERY REWRITE TO role20 ;

GRANT GRANT ANY OBJECT PRIVILEGE TO role20 ;

GRANT GRANT ANY PRIVILEGE TO role20 ;

GRANT GRANT ANY ROLE TO role20 ;

GRANT IMPORT FULL DATABASE TO role20 ;

GRANT INSERT ANY CUBE DIMENSION TO role20 ;

GRANT INSERT ANY MEASURE FOLDER TO role20 ;

GRANT INSERT ANY TABLE TO role20 ;

GRANT LOCK ANY TABLE TO role20 ;

GRANT LOGMINING TO role20 ;

GRANT MANAGE ANY FILE GROUP TO role20 ;

BEGIN

SYS.DBMS\_AQADM.GRANT\_SYSTEM\_PRIVILEGE (

PRIVILEGE => 'MANAGE\_ANY',

GRANTEE => 'role20',

ADMIN\_OPTION => FALSE);

END;

/

GRANT MANAGE FILE GROUP TO role20 ;

GRANT MANAGE SCHEDULER TO role20 ;

GRANT MANAGE TABLESPACE TO role20 ;

GRANT MERGE ANY VIEW TO role20 ;

GRANT ON COMMIT REFRESH TO role20 ;

GRANT QUERY REWRITE TO role20 ;

GRANT READ ANY FILE GROUP TO role20 ;

GRANT READ ANY TABLE TO role20 ;

GRANT REDEFINE ANY TABLE TO role20 ;

GRANT RESTRICTED SESSION TO role20 ;

GRANT RESUMABLE TO role20 ;

GRANT SELECT ANY CUBE TO role20 ;

GRANT SELECT ANY CUBE BUILD PROCESS TO role20 ;

GRANT SELECT ANY CUBE DIMENSION TO role20 ;

GRANT SELECT ANY DICTIONARY TO role20 ;

GRANT SELECT ANY MEASURE FOLDER TO role20 ;

GRANT SELECT ANY MINING MODEL TO role20 ;

GRANT SELECT ANY SEQUENCE TO role20 ;

GRANT SELECT ANY TABLE TO role20 ;

GRANT SELECT ANY TRANSACTION TO role20 ;

GRANT SET CONTAINER TO role20 ;

GRANT UNDER ANY TABLE TO role20 ;

GRANT UNDER ANY TYPE TO role20 ;

GRANT UNDER ANY VIEW TO role20 ;

GRANT UPDATE ANY CUBE TO role20 ;

GRANT UPDATE ANY CUBE BUILD PROCESS TO role20 ;

GRANT UPDATE ANY CUBE DIMENSION TO role20 ;

GRANT UPDATE ANY TABLE TO role20 ;

GRANT USE ANY JOB RESOURCE TO role20 ;

GRANT USE ANY SQL TRANSLATION PROFILE TO role20 ;

-- Roles granted to role20

GRANT CAPTURE\_ADMIN TO role20 ;

GRANT DATAPUMP\_EXP\_FULL\_DATABASE TO role20 ;

GRANT DATAPUMP\_IMP\_FULL\_DATABASE TO role20 ;

GRANT EM\_EXPRESS\_ALL TO role20 ;

GRANT EXECUTE\_CATALOG\_ROLE TO role20 ;

GRANT EXP\_FULL\_DATABASE TO role20 ;

GRANT GATHER\_SYSTEM\_STATISTICS TO role20 ;

GRANT IMP\_FULL\_DATABASE TO role20 ;

GRANT JAVA\_ADMIN TO role20 ;

GRANT OLAP\_DBA TO role20 ;

GRANT OLAP\_XS\_ADMIN TO role20 ;

GRANT OPTIMIZER\_PROCESSING\_RATE TO role20 ;

GRANT SCHEDULER\_ADMIN TO role20 ;

GRANT SELECT\_CATALOG\_ROLE TO role20 ;

GRANT WM\_ADMIN\_ROLE TO role20 ;

GRANT XDBADMIN TO role20 ;

GRANT XDB\_SET\_INVOKER TO role20 ;

-- Grantees of role20

GRANT role20 TO C##OAZ14 ;

GRANT role20 TO C##role20 ;

GRANT role20 TO HR;

GRANT role20 TO OAZ13;

GRANT role20 TO OAZ14;

GRANT role20 TO SYS WITH ADMIN OPTION ;

GRANT role20 TO SYSTEM ;

GRANT role20 TO USER1;

## Tài liệu tham khảo:

<https://www.tranvanbinh.vn/2021/01/tao-user-database-12c.html>

<https://www.tranvanbinh.vn/2020/02/tutorial-dba-co-ban-chuong-17-quan-ly.html>

<https://www.tranvanbinh.vn/2020/11/quan-ly-user-trong-oracle-database.html>

<https://www.tranvanbinh.vn/2020/11/quan-ly-quyen-trong-oracle-database.html>

<https://www.tranvanbinh.vn/2020/11/quan-ly-profile.html>

<https://www.tranvanbinh.vn/2020/11/quan-ly-role.html>

<https://www.tranvanbinh.vn/2020/06/hoc-oracle-db-tu-z-bai-20-quan-ly-chuc.html>

# BÀI 7: QUẢN LÝ DUNG LƯỢNG LƯU TRỮ (STORAGE)

## Thực hành tạo tablespace

select \* from dba\_tablespaces;

SELECT a.tablespace\_name,100 - ROUND ( (NVL (b.bytes\_free, 0) / a.bytes\_alloc) \* 100) "%Usage", ROUND

(a.bytes\_alloc / 1024 / 1024) "Size MB", ROUND (NVL (b.bytes\_free, 0) / 1024 / 1024) "Free MB",

(ROUND (a.bytes\_alloc / 1024 / 1024)- ROUND (NVL (b.bytes\_free, 0) / 1024 / 1024)) "Used MB", ROUND ( (NVL (b.bytes\_free, 0) / a.bytes\_alloc) \* 100) "%Free", ROUND (maxbytes / 1048576) "Max MB",

ROUND (ROUND ( (a.bytes\_alloc - NVL (b.bytes\_free, 0)) / 1024 / 1024)/ ROUND (maxbytes / 1048576) \* 100) "%Used of Max"

FROM ( SELECT f.tablespace\_name, SUM (f.bytes) bytes\_alloc, SUM (DECODE (f.autoextensible, 'YES', f.maxbytes, 'NO', f.bytes)) maxbytes

FROM dba\_data\_files f

GROUP BY tablespace\_name) a,

( SELECT f.tablespace\_name, SUM (f.bytes) bytes\_free FROM dba\_free\_space f GROUP BY tablespace\_name) b

WHERE a.tablespace\_name = b.tablespace\_name(+) --and a.tablespace\_name in ('DATA201505','DATA201506','IMPORT\_TBS','INDX201505','INDX201506')

order by "%Used of Max" desc;

select \* from DBA\_DATA\_FILES;

-- Cach 1: OMF

SQL> alter session set container=pdb1;

Session altered.

SQL> show parameter db\_create;

NAME TYPE VALUE

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

db\_create\_file\_dest string /u01/app/oracle/oradata

db\_create\_online\_log\_dest\_1 string /u01/app/oracle/redolog

-- Boi den doan duoi va an F5

create tablespace DATA datafile size 1m autoextend on next 10m;

create tablespace DATA2;

create tablespace indx datafile size 1m autoextend on next 10m;

create tablespace data2022 datafile size 1m autoextend on next 10m;

create tablespace indx2022 datafile size 1m autoextend on next 10m;

create tablespace data2023 datafile size 1m autoextend on next 10m;

create tablespace indx2023 datafile size 1m autoextend on next 10m;

create tablespace dump datafile size 1m autoextend on next 10m;

create tablespace lob datafile size 1m autoextend on next 10m;

create tablespace data\_nghiepvu1 datafile size 1m autoextend on next 10m;

create tablespace indx\_nghiepvu1 datafile size 1m autoextend on next 10m;

--ASM

--create tablespace data\_omf datafile '+DATA' size 10m autoextend on next 100m;

-- Cach 2: Duong dan tuong minh (khong dung)

create tablespace data3 datafile '/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/data3\_02.dbf' size 10m autoextend on next 100m;

create tablespace data datafile '/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/data01.dbf' size 10m autoextend on next 100m;

create tablespace indx datafile '/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/indx01.dbf' size 10m autoextend on next 100m;

create tablespace data2021 datafile '/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/data2021\_01.dbf' size 10m autoextend on next 100m;

create tablespace indx2021 datafile '/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/indx2021\_01.dbf' size 10m autoextend on next 100m;

create tablespace data2022 datafile '/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/data2022\_01.dbf' size 10m autoextend on next 100m;

create tablespace indx2022 datafile '/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/indx2022\_01.dbf' size 10m autoextend on next 100m;

create tablespace dump datafile '/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/dump01.dbf' size 10m autoextend on next 100m;

create tablespace data\_nghiepvu1 datafile '/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/data\_nghiepvu1\_01.dbf' size 10m autoextend on next 100m;

create tablespace indx\_nghiepvu1 datafile '/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/indx\_nghiepvu1\_01.dbf' size 10m autoextend on next 100m;

ALTER TABLESPACE data ADD DATAFILE;

drop tablespace data3;

select \* from dba\_tablespaces;

select \* from dba\_data\_files where file\_name like '%data3%';

## Quản trị tablespace Oracle Database

--Các câu lệnh thường sử dụng để quản trị tablespace:

--1. CHECK

--Hiển thị dung lượng trống của tablespace

SELECT a.tablespace\_name,100 - ROUND ( (NVL (b.bytes\_free, 0) / a.bytes\_alloc) \* 100) "%Usage",

ROUND (a.bytes\_alloc / 1024 / 1024) "Size MB",

ROUND (a.bytes\_alloc / 1024 / 1024)- ROUND (NVL (b.bytes\_free, 0) / 1024 / 1024) "Used MB",

ROUND (NVL (b.bytes\_free, 0) / 1024 / 1024) "Free MB",

--ROUND ( (NVL (b.bytes\_free, 0) / a.bytes\_alloc) \* 100) "%Free",

ROUND (maxbytes / 1048576) "Max MB",

round(maxbytes/1048576-(ROUND (a.bytes\_alloc / 1024 / 1024)- ROUND (NVL (b.bytes\_free, 0) / 1024 / 1024)),0) "Free\_MB\_Max",

ROUND (ROUND ( (a.bytes\_alloc - NVL (b.bytes\_free, 0)) / 1024 / 1024)/ ROUND (maxbytes / 1048576) \* 100) "%Used of Max"

FROM (SELECT f.tablespace\_name, SUM (f.bytes) bytes\_alloc, SUM (DECODE (f.autoextensible, 'YES', f.maxbytes, 'NO', f.bytes)) maxbytes

FROM dba\_data\_files f

GROUP BY tablespace\_name) a,

(SELECT f.tablespace\_name, SUM (f.bytes) bytes\_free FROM dba\_free\_space f GROUP BY tablespace\_name) b

WHERE a.tablespace\_name = b.tablespace\_name(+)

--and (a.tablespace\_name in ('TEMP1','DATA201511','DATA2016','INDX','INDX2016'))

--order by "%Used of Max" desc;

order by tablespace\_name;

--Dung luong của từng data files

SELECT FILE\_NAME, BLOCKS, round(bytes/1024/1024,2) "MB", TABLESPACE\_NAME

FROM DBA\_DATA\_FILES

WHERE file\_name like '%undo%'

order by tablespace\_name;

--Dung lượng DB

SELECT ROUND(SUM(BYTES)/1024/1024,2) "MB" FROM DBA\_DATA\_FILES;

--Dung lượng schema

select owner,round(sum(bytes/1024/1024),2) "MB" from dba\_segments

group by owner order by owner;

--Dung lượng schema.segment

select owner, segment\_name,round(sum(bytes/1024/1024),2) "MB" from dba\_segments

group by owner, segment\_name

order by owner, segment\_name;

-- Dung luong tablespace READ ONLY: 7,15GB

select sum(bytes)/1024/1024/1024 "GB" from dba\_data\_files;

--2,92TB

select sum(bytes)/1024/1024/1024 "GB" from dba\_data\_files where tablespace\_name in

(select tablespace\_name from dba\_tablespaces where

status='READ ONLY');

-- Check duong dan cua datafile trong tablesapce

select \* from dba\_data\_files where tablespace\_name='USERS';

--Check trạng thái của datafile, tablespace

--Offline

select file\_name,tablespace\_name,online\_status from dba\_data\_files where tablespace\_name='DATA201008'

-- Read Only (co the online hoac offline), Online

select tablespace\_name, status from dba\_tablespaces where

--tablespace\_name like '%201208%' and

status='OFFLINE'

order by tablespace\_name;

-- Tim kiem volume cho tablespace

SELECT

A.TABLESPACE\_NAME,

TOTAL\_SPACE,

TOTAL\_SPACE-FREE\_SPACE USED,

FREE\_SPACE

FROM

(SELECT SUM(BYTES/1024/1024) TOTAL\_SPACE, TABLESPACE\_NAME FROM DBA\_DATA\_FILES GROUP BY TABLESPACE\_NAME) A,

(SELECT SUM(BYTES/1024/1024) FREE\_SPACE, TABLESPACE\_NAME FROM DBA\_FREE\_SPACE GROUP BY TABLESPACE\_NAME) B

WHERE

A.TABLESPACE\_NAME=B.TABLESPACE\_NAME(+)

AND A.TABLESPACE\_NAME like 'USERS'

ORDER BY FREE\_SPACE ASC;

-- Thống kê dung lượng tăng theo tháng

select to\_char(creation\_time,'yyyy/mm'), round(sum(bytes)/1024/1024/1024) "GB" from v$datafile

group by to\_char(creation\_time,'yyyy/mm')

order by 1;

(tháng 2021/07 tạo 16.3TB, tháng 08/2021 tạo 10.2TB, tháng 09/2021 dến hôm nay là 09/09/2021 tạo 3.6TB, cần tạo khoảng > 7TB nữa để duy trì hết tháng)

--2. CREATE TABLESPACE

CREATE TABLESPACE data201102 DATAFILE '/u02/oradata/ORCL/datafile/DATA201102\_01.DBF' SIZE 6G;

CREATE TABLESPACE data201102 DATAFILE '/u02/oradata/ORCL/datafile/DATA201102\_01.DBF' SIZE 100M;

'/u02/oradata/ORCL/datafile/DATA201103\_01.DBF' SIZE 50M AUTOEXTEND ON NEXT 100M MAXSIZE 2048M,

'/u02/oradata/ORCL/datafile/DATA201103\_02.DBF' SIZE 50M AUTOEXTEND ON NEXT 100M MAXSIZE 2048M,

'/u02/oradata/ORCL/datafile/DATA201103\_03.DBF' SIZE 50M AUTOEXTEND ON NEXT 100M MAXSIZE 2048M;

CREATE TABLESPACE data201103 DATAFILE

CREATE TABLESPACE /\*FileSystem\*/ DATA202304 datafile size 10M autoextend on next 10M ;

CREATE TABLESPACE app\_ts;

CREATE TABLESPACE app\_ts2 DATAFILE SIZE 150M;

CREATE TABLESPACE /\*ASM\*/ DATA202304 datafile '+DATA' size 1G autoextend on next 100M ;

--3.ALTER TABLESPACE

--ADD DATAIFILE ASM

--+ Check dung luong diskgroup trong

select \* from gv$asm\_diskgroup;

--+ Check duong dan chua datafile

select \* from dba\_data\_files where tablespace\_name='DUMP\_DATA';

--+ Add datafile trong ASM:

alter tablespace DATA\_ add datafile '+DATA' size 1G autoextend on next 200M maxsize 10G;

ALTER TABLESPACE DATA201206 ADD datafile size 512M autoextend on next 200M maxsize 8g

--ADD DATAIFILE FILESYSTEM

select \* from dba\_data\_files where tablespace\_name='DATA2022';

alter tablespace app\_ts add datafile size 1G autoextend on next 100m;

alter tablespace app\_ts add datafile size 10m autoextend on next 10m;

alter tablespace app\_ts add datafile size 8G autoextend off;

--+ Add datafile với file system

ALTER TABLESPACE DATA2022 ADD DATAFILE SIZE 1M AUTOEXTEND ON NEXT 10M;

select \* from dba\_data\_files where tablespace\_name='DATA2022';

select \* from v$datafile where ts# in (select ts# from v$tablespace where name='DATA2022');

select \* from v$tablespace;

select \* from dba\_tablespaces;

ALTER TABLESPACE DUMP\_DATA ADD DATAFILE '/u02/oracle/oradata/datafile/DUMP\_DATA\_04.dbf' SIZE 2G AUTOEXTEND ON NEXT 200M MAXSIZE 10G;

ALTER TABLESPACE DUMP\_DATA ADD DATAFILE '/u02/oracle/oradata/datafile/DUMP\_DATA\_04.dbf' SIZE 2G AUTOEXTEND ON NEXT 200M; --unlimited

ALTER TABLESPACE lmtemp ADD TEMPFILE '/u02/oracle/data/lmtemp02.dbf' SIZE 18M REUSE;

-- Drop datafile

ALTER TABLESPACE DATA201302 DROP DATAFILE '/u03/oradata/ORCL/DATA201302\_0004.dbf';

-- Rename datafile

ALTER TABLESPACE users

RENAME DATAFILE '/u02/oracle/ORCL/user1.dbf',

'/u02/oracle/ORCL/user2.dbf'

TO '/u02/oracle/ORCL/users01.dbf',

'/u02/oracle/ORCL/users02.dbf';

-- ONLINE, OFFLINE

ALTER TABLESPACE APP\_TS ONLINE;

ALTER TABLESPACE APP\_TS OFFLINE [force | normal];

-- Read write, read only:

ALTER TABLESPACE DATA201101 READ WRITE;

ALTER TABLESPACE DATA201101 READ ONLY;

-- Renaming Tablespaces

ALTER TABLESPACE my\_space RENAME TO your\_space;

--4.DROP TABLESPACE

DROP TABLESPACE APP\_TS INCLUDING CONTENTS AND DATAFILES;

## Quản trị temporary tablespace

--Quản trị Temp files trong Oracle Database

--Keyword: Recreate temp tablespace Oracle Database 10g, 11g, 12c, 19c

-- Temporary Tablespace Usage

select \* from v$tempfile;

select \* from dba\_temp\_files;

-- Xac dinh session dang dung TEMP

SELECT A.inst\_id,b.tablespace, 'ALTER SYSTEM KILL SESSION '''||a.sid||','||a.serial#||',@'||a.inst\_id||''' IMMEDIATE;',ROUND(((b.blocks\*p.value)/1024/1024),2)||'M' "SIZE",

a.sid||','||a.serial# SID\_SERIAL,a.username,a.program

FROM sys.Gv\_$session a,sys.Gv\_$sort\_usage b,sys.v\_$parameter p

WHERE p.name = 'db\_block\_size' and a.inst\_id=b.inst\_id AND a.saddr = b.session\_addr and b.tablespace like 'TEMP'

--AND A.USERNAME IS NOT NULL AND A.USERNAME not like 'SYS%'

ORDER BY b.tablespace, b.blocks;

select b.Total\_MB,

b.Total\_MB - round(a.used\_blocks\*8/1024) Current\_Free\_MB,

round(used\_blocks\*8/1024) Current\_Used\_MB,

round(max\_used\_blocks\*8/1024) Max\_used\_MB

from v$sort\_segment a,

(select round(sum(bytes)/1024/1024) Total\_MB from dba\_temp\_files ) b;

col hash\_value for a40

col tablespace for a10

col username for a15

set linesize 132 pagesize 1000

SELECT s.sid, s.username, u.tablespace, s.sql\_hash\_value||'/'||u.sqlhash hash\_value, u.segtype, u.contents, u.blocks

FROM v$session s, v$tempseg\_usage u

WHERE s.saddr=u.session\_addr

order by u.blocks;

select s.inst\_id,

s.sid , s.serial# serial,

s.username,

s.osuser,

p.spid,

s.module,

p.program,

sum (t.blocks) \* tbs.block\_size / 1024 / 1024 mb\_used,

t.tablespace,

count(\*) nbr\_statements

from

gv$sort\_usage t,

gv$session s,

dba\_tablespaces tbs,

gv$process p

where

t.session\_addr = s.saddr

and

s.paddr = p.addr

and

t.tablespace = tbs.tablespace\_name

group by

s.inst\_id,

s.sid,

s.serial#,

s.username,

s.osuser,

p.spid,

s.module,

p.program,

tbs.block\_size,

t.tablespace

order by MB\_used desc;

--Tìm các object đang sử dụng temp

select tu.tablespace,tu.username,s.sid,s.serial#,s.inst\_id from gv$tempseg\_usage tu, gv$session s

where tu.session\_addr=s.saddr;

--Tạo script kill các session đó:

select 'ALTER SYSTEM KILL SESSION '''||s.sid||','||s.serial#||',@'||s.inst\_id||''' immediate;' from gv$tempseg\_usage tu, gv$session s

where tu.session\_addr=s.saddr and tu.tablespace='TEMP';

-- Create tempprary tablespace

CREATE SMALLFILE TEMPORARY TABLESPACE TEMP2

TEMPFILE '/u04/oracle/DBAViet/temp00.dbf' SIZE 512M AUTOEXTEND ON NEXT 150M MAXSIZE 1024M,

'/u04/oracle/DBAViet/temp01.dbf' SIZE 512M AUTOEXTEND ON NEXT 150M MAXSIZE 1024M,

'/u04/oracle/DBAViet/temp02.dbf' SIZE 512M AUTOEXTEND ON NEXT 150M MAXSIZE 1024M,

'/u04/oracle/DBAViet/temp03.dbf' SIZE 512M AUTOEXTEND ON NEXT 150M MAXSIZE 1024M,

'/u04/oracle/DBAViet/temp04.dbf' SIZE 512M AUTOEXTEND ON NEXT 150M MAXSIZE 1024M,

'/u04/oracle/DBAViet/temp05.dbf' SIZE 512M AUTOEXTEND ON NEXT 150M MAXSIZE 1024M,

'/u04/oracle/DBAViet/temp06.dbf' SIZE 512M AUTOEXTEND ON NEXT 150M MAXSIZE 1024M,

'/u04/oracle/DBAViet/temp07.dbf' SIZE 512M AUTOEXTEND ON NEXT 150M MAXSIZE 1024M,

'/u04/oracle/DBAViet/temp08.dbf' SIZE 512M AUTOEXTEND ON NEXT 150M MAXSIZE 1024M,

'/u04/oracle/DBAViet/temp09.dbf' SIZE 512M AUTOEXTEND ON NEXT 150M MAXSIZE 1024M,

'/u04/oracle/DBAViet/temp10.dbf' SIZE 512M AUTOEXTEND ON NEXT 150M MAXSIZE 1024M,

'/u04/oracle/DBAViet/temp11.dbf' SIZE 512M AUTOEXTEND ON NEXT 150M MAXSIZE 1024M,

'/u04/oracle/DBAViet/temp12.dbf' SIZE 512M AUTOEXTEND ON NEXT 150M MAXSIZE 1024M,

'/u04/oracle/DBAViet/temp13.dbf' SIZE 512M AUTOEXTEND ON NEXT 150M MAXSIZE 1024M,

'/u04/oracle/DBAViet/temp14.dbf' SIZE 512M AUTOEXTEND ON NEXT 150M MAXSIZE 1024M,

'/u04/oracle/DBAViet/temp15 .dbf' SIZE 512M AUTOEXTEND ON NEXT 150M MAXSIZE 1024M;

CREATE SMALLFILE TEMPORARY TABLESPACE TEMP2

TEMPFILE '+DATA' SIZE 1G AUTOEXTEND ON NEXT 10M MAXSIZE UNLIMITED;

alter database default temporary tablespace temp2;

drop tablespace temp including contents and datafiles;

select \* from dba\_temp\_files order by file\_name desc

-- Chuyển tablespaces sang temporary tablespace

ALTER TABLESPACE temp2 TEMPORARY;

-- Add tempfile file system

ALTER TABLESPACE temp\_demo ADD TEMPFILE 'temp05.dbf' size 20480m autoextend on next 100M max size 10G;

ALTER TABLESPACE TEMP ADD TEMPFILE '+DATA' SIZE 1G AUTOEXTEND ON NEXT 100M

MAXSIZE UNLIMITED;

-- Add tempfile trong ASM

-- Check đường dẫn lưu temp file

select \* from dba\_temp\_files where tablespace\_name='TEMP';

--Check dung luong diskgroup:Giả từ DATA còn dư

select \* from v$asm\_diskgroup;

-- Add tempfile: Tùy 1 hoặc 5 hoặc 10 file

alter tablespace TEMP1 add tempfile '+DATA' size 1G autoextend on next 100m;

alter tablespace TEMP1 add tempfile '+DATA' size 1G autoextend on next 100m;

alter tablespace TEMP1 add tempfile '+DATA' size 1G autoextend on next 100m;

alter tablespace TEMP1 add tempfile '+DATA' size 1G autoextend on next 100m;

alter tablespace TEMP1 add tempfile '+DATA' size 1G autoextend on next 100m;

alter tablespace TEMP add tempfile '/SID/oradata/data02/temp05.dbf' size 1800m reuse;

--RESIZE TEMPFILE

select \* from dba\_temp\_files;

alter database tempfile '/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/o1\_mf\_temp\_jngplfvb\_.dbf' resize 10M

alter database tempfile '/SID/oradata/data02/temp12.dbf' autoextend on maxsize 1800M;

alter database tempfile '/u02/oracle/oradata/DBAViet/temp\_02.dbf' resize 10240m

--+ Script resize mọi tempfile cả tablespace TEMP về 10MB

select 'alter database tempfile ''' || file\_name || ''' resize 10M;' from dba\_temp\_files where tablespace\_name='TEMP';

-- Drop tempfile

ALTER TABLESPACE temp\_demo DROP TEMPFILE 'temp05.dbf';

-- Drop tablespace TEMP

drop tablespace temp including contents and datafiles;

## Tài liệu tham khảo:

<https://www.tranvanbinh.vn/2020/07/quan-tri-tablespace-oracle.html>

<https://www.tranvanbinh.vn/2021/02/xu-ly-khi-u01-u02-chua-datafile-ay-99.html>

<https://www.tranvanbinh.vn/2021/02/cach-xu-ly-loi-ora-19815-warning.html>

<https://www.tranvanbinh.vn/2021/04/job-tu-ong-xoa-log-file-trong-oracle.html>

<https://www.tranvanbinh.vn/2021/02/xu-ly-khi-u01-u02-chua-datafile-ay-99.html>

<https://www.tranvanbinh.vn/2021/01/quy-trinh-kiem-tra-nhanh-oracle.html>

<https://www.tranvanbinh.vn/2021/10/thu-tuc-add-ia-vao-asm-diskgroup-trong.html>

<https://www.tranvanbinh.vn/2020/07/quy-trinh-add-o-ia-moi-vao-diskgroup.html>

<https://www.tranvanbinh.vn/2018/08/thu-tuc-add-ia-vao-diskgroup-data-trong.html>

<https://www.tranvanbinh.vn/2020/11/script-resize-datafile.html>

<https://www.tranvanbinh.vn/2018/09/chuyen-du-lieu-tu-tablespace-indx-sang.html>

<https://www.tranvanbinh.vn/2020/11/asmchuyen-du-lieu-tu-phan-vung-data.html>

<https://www.tranvanbinh.vn/2018/09/chuyen-du-lieu-tu-tablespace-khac-sang.html>

<https://www.tranvanbinh.vn/2020/11/thu-tuc-chuyen-du-lieu-sang-phan-vung.html>

<https://www.tranvanbinh.vn/p/admin.html>

# BÀI 8: QUẢN LÝ KHÔNG GIAN LƯU TRỮ (SPACE)

## Resize datafile:

declare

cursor c1 is select tablespace\_name,file\_id,max(block\_id) mbid from dba\_extents

-- where tablespace\_name in ('DATA\_GPRS')

group by tablespace\_name,file\_id;

fs number;

fn varchar2(100);

ts number;

x number;

gtot number:=0;

tffs number:=0;

ffrags number:= 0;

begin

for v1 in c1 loop

select sum(bytes)/1024/1024 into fs from dba\_free\_space where tablespace\_name = v1.tablespace\_name and

file\_id = v1.file\_id and block\_id>v1.mbid;

select file\_name,bytes/1024/1024 into fn,ts from dba\_data\_files where file\_id=v1.file\_id and tablespace\_name=v1.tablespace\_name;

select sum(bytes/1024/1024) into tffs from

(select a.bytes/1024/1024 as bytes from dba\_free\_space a where file\_id=v1.file\_id and tablespace\_name=v1.tablespace\_name

union all

select 0.000001 as bytes from dual);

x:=(ts-fs)+64;

if (ts>x) and (x>10) then

dbms\_output.put\_line('alter database datafile ' || '''' || fn || '''' || ' resize ' || round(x+1,0) || 'm;' || '--total size' || round(ts) || 'm;' );

--EXECUTE IMMEDIATE 'alter database datafile ' || '''' || fn || '''' || ' resize ' || round(x) || 'm;' || '--total size' || round(ts) || 'm;';

gtot := gtot + (ts-x);

end if;

ffrags := nvl(tffs - fs,0);

dbms\_output.put\_line('alter database datafile ' || '''' || fn || '''' || ' resize ' || round(x+1,0) || 'm;' || '--frag size' || round(ffrags) || 'm;');

--EXECUTE IMMEDIATE 'alter database datafile ' || '''' || fn || '''' || ' resize ' || round(x) || 'm;' || '--frag size' || round(ffrags) || 'm;';

end loop;

dbms\_output.put\_line('Total space reclaimation:' || gtot || 'MB');

end;

## Tài liệu tham khảo:

<https://www.tranvanbinh.vn/2021/09/cac-giai-phap-thu-hoi-dung-luong-cua.html>

<https://www.tranvanbinh.vn/2021/01/cau-truc-lai-bang-voi-dbmsredefinition.html>

<https://www.tranvanbinh.vn/2022/04/cach-e-tim-va-bo-nhung-phan-manh-tren.html>

<https://www.tranvanbinh.vn/2020/11/script-resize-datafile.html>

<https://www.tranvanbinh.vn/2021/03/truncate-table-khong-giai-phong-khong.html>

<https://www.tranvanbinh.vn/2021/03/cach-lay-lai-toan-bo-khong-gian-cua.html>

<https://www.tranvanbinh.vn/2021/09/cac-giai-phap-thu-hoi-dung-luong-cua.html>

<https://www.tranvanbinh.vn/2022/04/cach-e-tim-va-bo-nhung-phan-manh-tren.html>

<https://www.tranvanbinh.vn/2022/05/cach-tim-phan-manh-fragmentation-trong.html>

<https://www.tranvanbinh.vn/2020/12/giai-phap-nen-du-lieu-lam-giam-dung.html>

<https://www.tranvanbinh.vn/2022/04/cach-e-tim-va-bo-nhung-phan-manh-tren.html>

<https://www.tranvanbinh.vn/2022/05/cach-tim-phan-manh-fragmentation-trong.html>

<https://www.tranvanbinh.vn/2020/12/giai-phap-nen-du-lieu-lam-giam-dung.html>

# BÀI 9: QUẢN LÝ UNDO

--. MONITOR

- Các view sử dụng:

• V$UNDOSTAT

• V$ROLLSTAT

• V$TRANSACTION

• DBA\_UNDO\_EXTENTS

-- Hiển thị các transaction cần rollback và thời gian tương ứng

select usn, state, undoblockstotal "Total", undoblocksdone "Done", undoblockstotal-undoblocksdone "ToDo",decode(cputime,0,'unknown',sysdate+(((undoblockstotal-undoblocksdone) / (undoblocksdone / cputime)) / 86400)) "Estimated time to complete"from v$fast\_start\_transactions;

--Hiển thị dung lượng trống của tablespace UNDOTBS1 UNDOTBS2

SELECT a.tablespace\_name,100 - ROUND ( (NVL (b.bytes\_free, 0) / a.bytes\_alloc) \* 100) "%Usage",

ROUND (a.bytes\_alloc / 1024 / 1024) "Size MB",

ROUND (a.bytes\_alloc / 1024 / 1024)- ROUND (NVL (b.bytes\_free, 0) / 1024 / 1024) "Used MB",

ROUND (NVL (b.bytes\_free, 0) / 1024 / 1024) "Free MB",

--ROUND ( (NVL (b.bytes\_free, 0) / a.bytes\_alloc) \* 100) "%Free",

ROUND (maxbytes / 1048576) "Max MB",

round(maxbytes/1048576-(ROUND (a.bytes\_alloc / 1024 / 1024)- ROUND (NVL (b.bytes\_free, 0) / 1024 / 1024)),0) "Free\_MB\_Max",

ROUND (ROUND ( (a.bytes\_alloc - NVL (b.bytes\_free, 0)) / 1024 / 1024)/ ROUND (maxbytes / 1048576) \* 100) "%Used of Max"

FROM (SELECT f.tablespace\_name, SUM (f.bytes) bytes\_alloc, SUM (DECODE (f.autoextensible, 'YES', f.maxbytes, 'NO', f.bytes)) maxbytes

FROM dba\_data\_files f

GROUP BY tablespace\_name) a,

(SELECT f.tablespace\_name, SUM (f.bytes) bytes\_free FROM dba\_free\_space f GROUP BY tablespace\_name) b

WHERE a.tablespace\_name = b.tablespace\_name(+) and (a.tablespace\_name in ('UNDOTBS1','UNDOTBS1'))

order by "%Used of Max" desc;

--Kiểm tra session nào đang sử dụng:

SELECT s.username,

s.sid,

s.serial#,

t.used\_ublk,

t.used\_urec,

rs.segment\_name,

r.rssize,

r.status

FROM v$transaction t,

v$session s,

v$rollstat r,

dba\_rollback\_segs rs

WHERE s.saddr = t.ses\_addr

AND t.xidusn = r.usn

AND rs.segment\_id = t.xidusn

ORDER BY t.used\_ublk DESC;

select \* from dba\_data\_Files where tablespace\_name in ('UNDOTBS1','UNDOTBS21');

--1/Create undo tablespace

CREATE SMALLFILE UNDO TABLESPACE "TUNDOTBS1" DATAFILE

'/u03/oracle/oradata/DBAViet/tundo1\_01.dbf' SIZE 10G REUSE ,

'/u03/oracle/oradata/DBAViet/tundo1\_02.dbf' SIZE 10G REUSE ,

'/u03/oracle/oradata/DBAViet/tundo1\_03.dbf' SIZE 10G REUSE ,

'/u03/oracle/oradata/DBAViet/tundo1\_04.dbf' SIZE 10G REUSE ,

'/u03/oracle/oradata/DBAViet/tundo1\_05.dbf' SIZE 10G REUSE ,

'/u03/oracle/oradata/DBAViet/tundo1\_06.dbf' SIZE 10G REUSE ,

'/u03/oracle/oradata/DBAViet/tundo1\_07.dbf' SIZE 10G REUSE ,

'/u03/oracle/oradata/DBAViet/tundo1\_08.dbf' SIZE 10G REUSE

CREATE SMALLFILE UNDO TABLESPACE "TUNDOTBS2" DATAFILE

'/u03/oracle/oradata/DBAViet/tundo2\_01.dbf' SIZE 10G REUSE ,

'/u03/oracle/oradata/DBAViet/tundo2\_02.dbf' SIZE 10G REUSE ,

'/u03/oracle/oradata/DBAViet/tundo2\_03.dbf' SIZE 10G REUSE ,

'/u03/oracle/oradata/DBAViet/tundo2\_04.dbf' SIZE 10G REUSE ,

'/u03/oracle/oradata/DBAViet/tundo2\_05.dbf' SIZE 10G REUSE ,

'/u03/oracle/oradata/DBAViet/tundo2\_06.dbf' SIZE 10G REUSE ,

'/u03/oracle/oradata/DBAViet/tundo2\_07.dbf' SIZE 10G REUSE ,

'/u03/oracle/oradata/DBAViet/tundo2\_08.dbf' SIZE 10G REUSE

-- 2/ Set new undo tablespace for each intance

ALTER SYSTEM SET undo\_tablespace = TUNDOTBS1 SID='DBAViet1'

ALTER SYSTEM SET undo\_tablespace = TUNDOTBS2 SID='DBAViet2'

-- 3/ DROP TABLESPACE UNDOTBS1\_old including contents;

DROP TABLESPACE UNDOTBS1 including contents and datafiles;

DROP TABLESPACE UNDOTBS2 including contents and datafiles;

## Tài liệu tham khảo:

<https://www.tranvanbinh.vn/2020/06/hoc-oracle-db-tu-z-bai-11-quan-ly.html>

<https://www.tranvanbinh.vn/2020/02/tutorial-dba-chuong-8-quan-ly-redo-log.html>

<https://www.tranvanbinh.vn/2021/09/quan-ly-undo-data-trong-oracle-database.html>

<https://www.tranvanbinh.vn/2020/11/quan-ly-undo-tablespace.html>

<https://www.tranvanbinh.vn/2021/03/thu-tuc-tao-lai-online-redo-log-trong.html>

<https://www.tranvanbinh.vn/2020/11/quan-tri-redo-log-files.html>

<https://www.tranvanbinh.vn/2022/01/redo-log-cau-hinh-size-nhu-nao-la-toi-uu.html>

# BÀI 10: QUÁN LÝ GIAO DỊCH ĐỒNG THỜI

-- User1

--

create user user1 identified by oracle;

grant connect, resource, dba to user1;

create user user2 identified by oracle;

grant connect, resource, dba to user2;

create user user3 identified by oracle;

grant connect, resource, dba to user3;

select \* from hr.employees;

-- Cau 1: Bảng 100 triệu row

alter table hr.employees add salary number default 1000; --lock toan bo bang, co the treo DB neu HT online

-->

alter table hr.employees add salary number;

alter table hr.employees modify salary default 1000; --New

update hr.EMPlOYEES set salary=3000;

rollback;

commit;

-- Cau 2:

create table user1.test(id number); --> DDL la implicit commit;

alter table hr.employees add salary\_max1 number;

--alter table hr.employees modify salary default 1000;

--

UPDATE hr.employees

SET salary=salary+101

WHERE name='Dean Bollich';

rollback;

UPDATE hr.employees

SET salary=salary+102

WHERE name='Dean Bollich';

select \* from hr.employees order by 1;

UPDATE hr.employees

SET salary=salary+102

WHERE name='Milo Manoni';

-- User2

UPDATE employees

SET salary=salary\*1.1

WHERE name='Dean Bollich';

commit;

-- rollback;

-- Giai quyet lock

SELECT sid, serial#, username

FROM v$session WHERE sid IN

(SELECT blocking\_session FROM v$session);

ALTER SYSTEM KILL SESSION '257,48772' immediate;

SELECT /\*SID\*/ 'kill -9 ' || spid a, a.INST\_ID,A.SQL\_ID,A.SID, A.SERIAL#, a.USERNAME, a.STATUS,A.SCHEMANAME,a.OSUSER,A.MACHINE,A.PROGRAM,A.TYPE,A.LOGON\_TIME,a.prev\_exec\_start,BACKGROUND

FROM gv$session a, gv$process b

WHERE b.addr = a.paddr

AND a.inst\_id=b.inst\_id

--and b.inst\_id=2

AND a.sid in (

257

)

and type='USER'

order by inst\_id;

--deadlock

SELECT \* FROM hr.employees;

--t1 User1

update hr.EMPlOYEES set salary=3000 where employee\_id= 200;

--t1 User2

update hr.EMPlOYEES set salary=2500 where employee\_id= 201;

--t2 User1

update hr.EMPlOYEES set salary=3000 where employee\_id= 201;

--t2 User2

update hr.EMPlOYEES set salary=3000 where employee\_id= 200;

--FLASHBACK

select \* from v$parameter where name like '%undo%';

--FLASHBACK QUERY1

create table hr.xxx\_employees\_20230915\_21h45 as select \* FROM hr.employees AS OF TIMESTAMP TO\_TIMESTAMP('2023-09-15 21:45:00', 'YYYY-MM-DD HH24:MI:SS');

create table hr.xxx\_employees\_20230915\_21h40 as select \* FROM hr.employees AS OF TIMESTAMP TO\_TIMESTAMP('2023-09-15 21:40:00', 'YYYY-MM-DD HH24:MI:SS');

select \* from hr.xxx\_employees\_20230915\_21h40;

select \* from hr.employees;

alter table hr.employees rename to xxx\_employees;

alter table hr.xxx\_employees\_20230915\_21h40 rename to employees;

select \* from hr.employees;

--Flashback table

alter table hr.employees enable row movement;

FLASHBACK TABLE hr.employees TO TIMESTAMP TO\_TIMESTAMP('2023-09-15 21:59:00', 'YYYY-MM-DD HH24:MI:SS');

select \* from hr.employees;

## Tài liệu tham khảo:

<https://www.tranvanbinh.vn/2020/12/bi-quyet-cap-nhat-cac-table-package.html>

<https://www.tranvanbinh.vn/2020/12/bi-quyet-cap-nhat-table-package-core.html>

<https://www.tranvanbinh.vn/2022/03/lock-du-lieu-trong-oracle.html>

<https://www.tranvanbinh.vn/2020/02/giam-sat-tai-active-sesion-lock.html>

<https://www.tranvanbinh.vn/2021/11/tim-va-kill-cac-session-tac-ong-vao-1.html>

<https://www.tranvanbinh.vn/2020/12/bi-quyet-cap-nhat-table-package-core.html>

<https://www.tranvanbinh.vn/2020/12/bi-quyet-cap-nhat-cac-table-package.html>

<https://www.tranvanbinh.vn/2022/02/quan-ly-tien-trinh-process-trong-he-ieu.html>

<https://www.tranvanbinh.vn/2022/02/tim-tien-trinh-chiem-nhieu-cpu-nhat.html>

<https://www.tranvanbinh.vn/2021/11/cap-cuu-sos-kill-9-e-tranh-treo-oracle.html>

<https://www.tranvanbinh.vn/2021/09/ung-dung-tu-nhien-chay-cham-check-nhu.html>

# BÀI 11: AUDIT DATABASE

## Trigger Firewall DB

--0.View

--Nhuoc diem

--1.Role DBA thi khong bat duoc

--2.User khong khai bao IP thi khong duoc

--Sys.logon\_trig

select \* from DBA\_TRIGGERS where trigger\_name like '%LOGON\_TRIG%';

drop trigger SYS.LOGON\_TRIG;

drop table sys.table\_ip purge;

select \* from SYS.TABLE\_IP\_LOG ;

drop table SYS.TABLE\_IP\_LOG purge;

select rowid,a.\* from sys.table\_ip a;

commit;

revoke dba from user2;

revoke dba from user1;

--1.Thu thập IP kết nối vào DB

CREATE TABLE SYS.TABLE\_IP\_LOG

(

USERNAME VARCHAR2(50 BYTE),

IP\_ADDRESS VARCHAR2(500 BYTE),

DESCRIPTION VARCHAR2(200 BYTE)

);

-- Tao trigger ghi log IP, tuong ung voi user

CREATE OR REPLACE TRIGGER SYS.LOGON\_TRIG

AFTER

LOGON

ON DATABASE

DECLARE

v\_ip VARCHAR2 (1000);

BEGIN

/\* comment doan chan ket noi

SELECT ip\_address

INTO v\_ip

FROM sys.table\_ip

WHERE username = SYS\_CONTEXT ('USERENV', 'SESSION\_USER');

IF sys.pck\_admin.checkip (SYS\_CONTEXT ('USERENV', 'IP\_ADDRESS'),

v\_ip) or SYS\_CONTEXT ('USERENV', 'HOST') in ('ORAAZ-HOST')

THEN

NULL;

ELSE

raise\_application\_error (

-20001,

'You are not allowed to connect to the database!');

END IF;

\*/

insert into sys.table\_ip\_log (username,ip\_address) values (SYS\_CONTEXT ('USERENV', 'SESSION\_USER'),SYS\_CONTEXT ('USERENV', 'IP\_ADDRESS'));

commit;

EXCEPTION

-- Nhung account ko duoc khai bao trong bang table\_ip

WHEN NO\_DATA\_FOUND THEN NULL;

END;

/

select \* from SYS.TABLE\_IP\_LOG;

-- User1, user2 login test

-- Sys kiem tra IP da thu thap

select \* from SYS.TABLE\_IP\_LOG;

create table SYS.TABLE\_IP as select \* from SYS.TABLE\_IP\_LOG;

select \* from SYS.TABLE\_IP;

--insert into SYS.TABLE\_IP select \* from SYS.TABLE\_IP\_LOG;

--commit;

--select \* from SYS.TABLE\_IP;

--2.Cài đặt

--Thay đổi lại trigger LOGON\_TRIG:

--CREATE TABLE SYS.TABLE\_IP

--(

-- USERNAME VARCHAR2(50 BYTE),

-- IP\_ADDRESS VARCHAR2(500 BYTE),

-- DESCRIPTION VARCHAR2(200 BYTE)

--)

select rowid,a.\* from SYS.TABLE\_IP a;

revoke dba from oaz18;

-- Tao trigger bat user truy cap dung IP da khi bao trong sys.table\_Ip

CREATE OR REPLACE TRIGGER SYS.LOGON\_TRIG

AFTER

LOGON

ON DATABASE

DECLARE

v\_ip VARCHAR2 (1000);

BEGIN

SELECT ip\_address

INTO v\_ip

FROM sys.table\_ip

WHERE username = SYS\_CONTEXT ('USERENV', 'SESSION\_USER');

IF sys.pck\_admin.checkip (SYS\_CONTEXT ('USERENV', 'IP\_ADDRESS'),

v\_ip) or SYS\_CONTEXT ('USERENV', 'HOST') in ('ORAAZ-HOST')

THEN

NULL;

ELSE

raise\_application\_error (

-20001,

'You are not allowed to connect to the database!');

END IF;

--insert into sys.table\_ip\_log (username,ip\_address) values (SYS\_CONTEXT ('USERENV', 'SESSION\_USER'),SYS\_CONTEXT ('USERENV', 'IP\_ADDRESS'));

commit;

EXCEPTION

-- Nhung account ko duoc khai bao trong bang table\_ip

WHEN NO\_DATA\_FOUND THEN NULL;

END;

/

-- Doi ip 192.168.182.1 thanh 192.168.182.10

select rowid,a.\* from SYS.TABLE\_IP a;

--oaz18

revoke dba from oaz18;

revoke dba from oaz19;

-- Gan quyen DBA cho user1

grant dba to user1; --Nhung USER1 192.168.182.77, ko duoc truy cap

-- Thuc te duoc truy cap do user1 co quyen DBA nen bypasss

revoke dba from user1;

-- Thu hoi quyen DBA tu c##oaz18

revoke dba from c##oaz18;

--CASE STUDY: USER2 KHONG KHAI BAO IP

select rowid,a.\* from SYS.TABLE\_IP a;

-- Xu ly

select username from dba\_users where account\_status='OPEN'

minus

select username from sys.table\_ip

order by 1;

-- Tao trigger chi cho phep dang nhap tu TOAD

CREATE OR REPLACE TRIGGER SYS.LOGON\_TRIG

AFTER

LOGON

ON DATABASE

DECLARE

v\_ip VARCHAR2 (1000);

v\_action varchar2(50);

BEGIN

-- SELECT ip\_address

-- INTO v\_ip

-- FROM sys.table\_ip

-- WHERE username = SYS\_CONTEXT ('USERENV', 'SESSION\_USER');

--select sys\_context('USERENV','ACTION') into v\_action from dual;

insert into sys.xxx\_client\_info values(sys\_context('USERENV','MODULE'));

commit;

if sys\_context('USERENV','MODULE') in ('Toad.exe') then

NULL;

else raise\_application\_error (

-20001,

'Bạn phải dùng chương trình TOAD để kết nối vào Database');

END IF;

-- IF sys.pck\_admin.checkip (SYS\_CONTEXT ('USERENV', 'IP\_ADDRESS'),

-- v\_ip) or SYS\_CONTEXT ('USERENV', 'HOST') in ('HOST-ORAAZ')

--

-- THEN

-- NULL;

-- ELSE

-- raise\_application\_error (

-- -20001,

-- 'You are not allowed to connect to the database!');

-- END IF;

--insert into table\_ip\_log (username,ip\_address) values (SYS\_CONTEXT ('USERENV', 'SESSION\_USER'),SYS\_CONTEXT ('USERENV', 'IP\_ADDRESS'));

--commit;

EXCEPTION

-- Nhung account ko duoc khai bao trong bang table\_ip

WHEN NO\_DATA\_FOUND THEN NULL;

END;

/

select \* from sys.xxx\_cli ent\_info;

-- Sua IP USER1 thanh 192.168.182.2

select rowid, a.\* from sys.table\_ip a;

-- User1 login lai bao loi:

revoke dba from user1;

revoke dba from user2;

--User1, user2 la DBA thi bypass

revoke dba from user1;

revoke dba from user2;

--Login lai bao loi

-- Sua IP user1 thanh 192.168.182.1

select rowid, a.\* from sys.table\_ip a;

--3.Khắc phục sự cố

-- Trong trường hợp cần thiết có thể xoá trigger đi:

DROP TRIGGER SYS.LOGON\_TRIG;

--Package SYS.pck\_admin

CREATE OR REPLACE PACKAGE SYS.pck\_admin

IS

Function ConvertIP(p\_IP varchar2)

Return Varchar2;

FUNCTION checkIP (p\_ip VARCHAR2, p\_Grant\_IP Varchar2)

RETURN BOOLEAN;

END;

/

CREATE OR REPLACE PACKAGE BODY SYS.pck\_admin

IS

FUNCTION convertip (p\_ip VARCHAR2)

RETURN VARCHAR2

AS

v\_resuilt VARCHAR2 (12);

v\_temp VARCHAR2 (4);

v\_i NUMBER;

v\_ip VARCHAR2 (15);

BEGIN

v\_ip := p\_ip;

v\_resuilt := NULL;

IF LENGTH (v\_ip) = 15

THEN

v\_resuilt := REPLACE (v\_ip, '.', NULL);

ELSE

v\_i := 1;

WHILE v\_ip IS NOT NULL

LOOP

v\_i := INSTR (v\_ip, '.', 1);

IF v\_i > 0

THEN

v\_temp := SUBSTR (v\_ip, 1, v\_i - 1);

v\_ip := TRIM (SUBSTR (v\_ip, v\_i + 1));

ELSE

v\_temp := SUBSTR (v\_ip, 1);

v\_ip := NULL;

END IF;

IF LENGTH (v\_temp) < 3

THEN

SELECT LPAD (v\_temp, 3, '0')

INTO v\_temp

FROM DUAL;

END IF;

v\_resuilt := v\_resuilt || TRIM (v\_temp);

END LOOP;

END IF;

RETURN v\_resuilt;

EXCEPTION

WHEN OTHERS

THEN

RETURN NULL;

END;

FUNCTION checkip (p\_ip VARCHAR2, p\_grant\_ip VARCHAR2)

RETURN BOOLEAN

AS

v\_ip VARCHAR2 (20);

v\_grant\_ip VARCHAR2 (1500);

v\_grantip VARCHAR2 (50);

v\_startip VARCHAR2 (50);

v\_endip VARCHAR2 (50);

v\_start NUMBER;

v\_fix NUMBER;

BEGIN

v\_ip := convertip (p\_ip);

v\_start := 1;

v\_grant\_ip := p\_grant\_ip || ';';

FOR i IN 1 .. LENGTH (v\_grant\_ip)

LOOP

IF SUBSTR (v\_grant\_ip, i, 1) = ',' OR SUBSTR (v\_grant\_ip, i, 1) =

';'

THEN

v\_grantip := SUBSTR (v\_grant\_ip, v\_start, i - v\_start);

v\_fix := INSTR (v\_grantip, '-');

IF v\_fix > 1

THEN --dai ip

v\_startip := SUBSTR (v\_grantip, 1, v\_fix - 1);

v\_endip := SUBSTR (v\_grantip, v\_fix + 1);

IF v\_ip >= convertip (v\_startip)

AND v\_ip <= convertip (v\_endip)

THEN

RETURN TRUE;

END IF;

ELSE --1 ip xac dinh

IF v\_ip = convertip (v\_grantip)

THEN

RETURN TRUE;

END IF;

END IF;

v\_start := i + 1;

END IF;

END LOOP;

RETURN FALSE;

END;

END;

/

## FGA

-- 1.Check Policy da duoc cau hinh

select \* from DBA\_AUDIT\_POLICIES;

-- Drop

BEGIN

DBMS\_FGA.drop\_policy(

object\_schema => 'HR',

object\_name => 'EMPLOYEES',

policy\_name => 'FGA\_EMPLOYEES');

END;

--SQL> show parameter audit;

--audit\_trail DB

select \* from hr.employees;

/\*\*\*\*\* 2.ADD \*\*\*\*\*/

BEGIN

DBMS\_FGA.add\_policy(

object\_schema => 'HR',

object\_name => 'EMPLOYEES',

policy\_name => 'FGA\_EMPLOYEES',

--AUDIT\_CONDITION => 'SYS.check\_ip\_machine = 1',

--audit\_condition => SYS\_CONTEXT('USERENV','SESSION\_USER') <> 'BINHTV',

statement\_types => 'SELECT, INSERT,UPDATE,DELETE'

);

END;

/\*\*\*\*\* 3.ROLLBACK: DISABLE, DROP\*\*\*\*\*/

-- Disable

BEGIN

DBMS\_FGA.disable\_policy(

object\_schema => 'HR',

object\_name => 'EMPLOYEES',

policy\_name => 'FGA\_EMPLOYEES');

END;

/

BEGIN

DBMS\_FGA.enable\_policy(

object\_schema => 'HR',

object\_name => 'EMPLOYEES',

policy\_name => 'FGA\_EMPLOYEES');

END;

/

-- Drop

BEGIN

DBMS\_FGA.drop\_policy(

object\_schema => 'HR',

object\_name => 'EMPLOYEES',

policy\_name => 'FGA\_EMPLOYEES');

END;

--4.TEST

grant all on hr.employees to oaz19;

create user user1 identified by oracle;

grant connect, resource, select any table to user1;

grant all on hr.employees to user1;

create user user2 identified by oracle;

grant connect, resource, select any table to user2;

grant all on hr.employees to user2;

-- oaz19

SELECT \*

FROM hr.employees

WHERE

department\_id = 1;

--User1,2

SELECT \*

FROM hr.employees

WHERE

department\_id = 1;

update hr.employees set email='JWHALEN\_NEW' where department\_id = 10;

commit;

--USER SYS CO GIAM DUOC KHONG: KHONG

SELECT \*

FROM hr.employees

WHERE

department\_id = 1;

update hr.employees set email='JWHALEN\_NEW' where department\_id = 10;

commit;

--USER DBA CO GIAM DUOC KHONG: CO

create user test\_dba identified by oracle;

grant connect, resource, dba to test\_dba;

SELECT \*

FROM hr.employees

WHERE

department\_id = 1;

update hr.employees set email='JWHALEN\_NEW' where department\_id = 10;

commit;

--DBA KHONG TRUNCTE DUOC: truncate table sys.fga\_log$;, PHAI QUYEN SYSDBA

-- 5.Check log da audit (bang sys.fga\_log$)

select \* from dba\_fga\_audit\_trail

where timestamp > sysdate-1 order by timestamp desc

select \* from sys.fga\_log$ where ntimestamp# > sysdate-1 order by ntimestamp# desc;

select \* from DBA\_COMMON\_AUDIT\_TRAIL;

--truncate table fga\_log$;

--truncate table hr.employees; -- Khong bat duoc

--Phụ lục: Hàm check\_ip\_machine

-- Bỏ các user máy chủ ứng dụng do ứng dụng đã ghi log rồi

CREATE OR REPLACE FUNCTION SYS.check\_ip\_machine

return number

is

begin

-- if (SYS\_CONTEXT('USERENV', 'SESSION\_USER') <> 'BINHTV'

-- and (SYS\_CONTEXT ('USERENV', 'IP\_ADDRESS') like '192.168.1%' or SYS\_CONTEXT ('USERENV', 'IP\_ADDRESS') like '1192.168.2%'

-- )) then

if (SYS\_CONTEXT ('USERENV', 'IP\_ADDRESS') not like '192.168.1.%' and

SYS\_CONTEXT ('USERENV', 'IP\_ADDRESS') not like '192.168.2.%' and

SYS\_CONTEXT ('USERENV', 'IP\_ADDRESS') not like '192.168.3.%') then

return 1;

else

return 0;

end if;

end;

## FGA với bảng partition

--C?u hình FGA v?i b?ng Partition trong Oracle Database

--M?c dích: T?o FGA v?i b?ng partition d? d? d?n d?p

--0.Check

-- • Kiem tra policy

select \* from DBA\_AUDIT\_POLICIES;

-- • Kiem tra log dã audit

select \* from DBA\_FGA\_AUDIT\_TRAIL where timestamp > sysdate-1 order by timestamp desc;

truncate table sys.fga\_log$;

-- 1. Create table to store log:

create tablespace DATA\_AUDIT datafile size 10m autoextend on next 10m;

create tablespace INDX\_AUDIT datafile size 10m autoextend on next 10m;

CREATE TABLE SYS.FGA\_AUDIT\_LOG

(

FGASID NUMBER(20),

ENTRYID NUMBER(20),

AUDIT\_DATE DATE,

FGA\_POLICY VARCHAR2(100 BYTE),

DB\_USER VARCHAR2(100 BYTE),

OS\_USER VARCHAR2(200 BYTE),

AUTHENT\_TYPE VARCHAR2(100 BYTE),

CLIENT\_ID VARCHAR2(100 BYTE),

CLIENT\_INFO VARCHAR2(500 BYTE),

HOST\_NAME VARCHAR2(200 BYTE),

INSTANCE\_ID NUMBER(2),

IP VARCHAR2(50 BYTE),

TERM VARCHAR2(100 BYTE),

SCHEMA\_OWNER VARCHAR2(100 BYTE),

TABLE\_NAME VARCHAR2(100 BYTE),

SQL\_BIND VARCHAR2(4000 BYTE),

SQL\_TEXT VARCHAR2(4000 BYTE),

SCN NUMBER(20)

)

NOCOMPRESS

TABLESPACE data\_audit

PCTUSED 0

PCTFREE 10

INITRANS 1

MAXTRANS 255

STORAGE (

BUFFER\_POOL DEFAULT

)

PARTITION BY RANGE (AUDIT\_DATE)

(

PARTITION DATA20230626 VALUES LESS THAN (TO\_DATE(' 2023-06-27 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))

LOGGING

NOCOMPRESS

TABLESPACE data\_audit

PCTFREE 10

INITRANS 1

MAXTRANS 255

STORAGE (

INITIAL 8M

NEXT 1M

MAXSIZE UNLIMITED

MINEXTENTS 1

MAXEXTENTS UNLIMITED

BUFFER\_POOL DEFAULT

)

)

NOCACHE

MONITORING;

-- Add partition het nam 2023

DECLARE

v\_nam NUMBER (4) := 2023;

v\_owner varchar2 (50) := 'sys';

v\_tablename VARCHAR2 (50) := 'FGA\_AUDIT\_LOG';

v\_date\_from date := to\_date('27/06/2023','dd/mm/yyyy');

v\_date\_to date := to\_date('31/12/2023','dd/mm/yyyy');

v\_numday number(5);

v\_tablespace varchar2(50):='DATA\_AUDIT';

BEGIN

v\_numday:=v\_date\_to-v\_date\_from;

FOR i IN 0 .. v\_numday

LOOP

DBMS\_OUTPUT.put\_line ('alter table '||v\_owner||'.'|| v\_tablename || ' add PARTITION DATA' ||to\_char(v\_date\_from+i,'YYYYMMDD')||' VALUES LESS THAN (TO\_DATE('''|| to\_char(v\_date\_from+i+1,'YYYY-MM-DD')||' 00:00:00'',''SYYYY-MM-DD HH24:MI:SS'', ''NLS\_CALENDAR=GREGORIAN'')) LOGGING TABLESPACE DATA\_AUDIT;');

END LOOP;

END;

create index sys.FGA\_AUDIT\_LOG on FGA\_AUDIT\_LOG(DB\_USER) tablespace indx\_audit local;

create index sys.FGA\_AUDIT\_LOG\_I2 on FGA\_AUDIT\_LOG(ip) tablespace indx\_audit local;

--2. Create audit procedure

CREATE OR REPLACE PROCEDURE SYS.pr\_fga\_auditing (p\_table\_owner IN VARCHAR2,

p\_table\_name IN VARCHAR2,

p\_fga\_policy IN VARCHAR2)

IS

l\_fgasid NUMBER (20);

l\_entryid NUMBER (20);

l\_term VARCHAR2 (2000);

l\_db\_user VARCHAR2 (30);

l\_os\_user VARCHAR2 (30);

l\_authent\_type VARCHAR2 (2000);

l\_client\_id VARCHAR2 (100);

l\_client\_info VARCHAR2 (100);

l\_host\_name VARCHAR2 (100);

l\_instance\_id NUMBER (2);

l\_ip VARCHAR2 (30);

l\_sql\_text VARCHAR2 (4000);

l\_sql\_bind VARCHAR2 (4000);

l\_scn NUMBER;

BEGIN

l\_fgasid := SYS\_CONTEXT ('USERENV', 'SESSIONID');

l\_entryid := SYS\_CONTEXT ('USERENV', 'ENTRYID');

l\_term := SYS\_CONTEXT ('USERENV', 'TERMINAL');

l\_db\_user := SYS\_CONTEXT ('USERENV', 'SESSION\_USER');

l\_os\_user := SYS\_CONTEXT ('USERENV', 'OS\_USER');

--l\_authent\_type := SYS\_CONTEXT ('USERENV', 'AUTHENTICATION\_TYPE');

l\_authent\_type := SYS\_CONTEXT ('USERENV', 'CLIENT\_IDENTIFIER');

l\_client\_id := SYS\_CONTEXT ('USERENV', 'CLIENT\_IDENTIFIER');

l\_client\_info := SYS\_CONTEXT ('USERENV', 'CLIENT\_INFO');

l\_host\_name := SYS\_CONTEXT ('USERENV', 'HOST');

l\_instance\_id := SYS\_CONTEXT ('USERENV', 'INSTANCE');

l\_ip := SYS\_CONTEXT ('USERENV', 'IP\_ADDRESS');

l\_sql\_text := SYS\_CONTEXT ('USERENV', 'CURRENT\_SQL');

l\_sql\_bind := SYS\_CONTEXT ('USERENV', 'CURRENT\_BIND');

l\_scn := SYS.DBMS\_FLASHBACK.get\_system\_change\_number;

INSERT INTO sys.fga\_audit\_log (fgasid,

entryid,

audit\_date,

fga\_policy,

db\_user,

os\_user,

authent\_type,

client\_id,

client\_info,

host\_name,

instance\_id,

ip,

term,

schema\_owner,

table\_name,

sql\_text,

sql\_bind,

scn)

VALUES (l\_fgasid,

l\_entryid,

SYSDATE,

p\_fga\_policy,

l\_db\_user,

l\_os\_user,

l\_authent\_type,

l\_client\_id,

l\_client\_info,

l\_host\_name,

l\_instance\_id,

l\_ip,

l\_term,

p\_table\_owner,

p\_table\_name,

l\_sql\_text,

l\_sql\_bind,

l\_scn);

END;

/

--3. Create check ip procedure

CREATE OR REPLACE FUNCTION SYS.check\_ip\_app\_srv

return number

is

begin

if (SYS\_CONTEXT ('USERENV', 'IP\_ADDRESS') not in ('100.10.18.10','100.10.18.11')

and SYS\_CONTEXT ('USERENV', 'IP\_ADDRESS') not like '100.10.8%'

and SYS\_CONTEXT ('USERENV', 'IP\_ADDRESS') not like '100.10.9%'

and SYS\_CONTEXT ('USERENV', 'IP\_ADDRESS') not like '100.10.10%'

and SYS\_CONTEXT ('USERENV', 'IP\_ADDRESS') not like '100.10.11%'

) then

return 1;

else

return 0;

end if;

end;

/

--4. Add policies:

select \* from DBA\_AUDIT\_POLICIES;

BEGIN

DBMS\_FGA.add\_policy (object\_schema => 'HR',

object\_name => 'EMPLOYEES',

policy\_name => 'FGA\_EMPLOYEES',

audit\_condition => 'SYS.CHECK\_IP\_APP\_SRV = 1',

statement\_types => 'SELECT,INSERT,UPDATE,DELETE',

handler\_schema => 'SYS',

handler\_module => 'PR\_FGA\_AUDITING',

enable => TRUE);

END;

-- Cao t?i thì disable

BEGIN

DBMS\_FGA.disable\_policy(

object\_schema => 'HR',

object\_name => 'EMPLOYEES',

policy\_name => 'FGA\_EMPLOYEES');

END;

/

-- Enable lai

BEGIN

DBMS\_FGA.enable\_policy(

object\_schema => 'HR',

object\_name => 'EMPLOYEES',

policy\_name => 'FGA\_EMPLOYEES');

END;

/

-- Drop

BEGIN

DBMS\_FGA.drop\_policy(

object\_schema => 'HR',

object\_name => 'EMPLOYEES',

policy\_name => 'FGA\_EMPLOYEES');

END;

/

--4.TEST

grant all on hr.employees to oaz18;

create user user1 identified by oracle;

grant connect, resource, select any table to user1;

grant all on hr.employees to user1;

create user user2 identified by oracle;

grant connect, resource, select any table to user2;

grant all on hr.employees to user2;

-- oaz18

SELECT \*

FROM hr.employees

WHERE

department\_id = 1;

--User1,2

SELECT \*

FROM hr.employees

WHERE

department\_id = 1;

update hr.employees set email='JWHALEN\_NEW' where department\_id = 10;

commit;

--USER SYS CO GIAM DUOC KHONG: KHONG

SELECT \*

FROM hr.employees

WHERE

department\_id = 1;

update hr.employees set email='JWHALEN\_NEW' where department\_id = 10;

commit;

--USER DBA CO GIAM DUOC KHONG: CO

create user test\_dba identified by oracle;

grant connect, resource, dba to test\_dba;

SELECT \*

FROM hr.employees

WHERE

department\_id = 1;

update hr.employees set email='JWHALEN\_NEW' where department\_id = 10;

commit;

--DBA KHONG TRUNCTE DUOC: truncate table sys.fga\_log$;, PHAI QUYEN SYSDBA

--5.Check

-- • Ki?m tra l?i policy

select \* from DBA\_AUDIT\_POLICIES;

-- • Ki?m tra log dã audit

select \* from DBA\_FGA\_AUDIT\_TRAIL where timestamp > sysdate-1 order by timestamp desc;

--truncate table sys.fga\_log$;

select \* from sys.fga\_log$ order by ntimestamp# desc;

select \* from SYS.FGA\_AUDIT\_LOG where audit\_date>sysdate-7 and DB\_USER='OAZ18';

truncate table SYS.FGA\_AUDIT\_LOG;

--N?u c?n dùng scrip thì tham kh?o:

--Script drop all policy

select 'BEGIN

DBMS\_FGA.drop\_policy(object\_schema => '''||object\_schema||''',

object\_name => '''|| object\_name|| ''',

policy\_name => ''' ||policy\_name||''');

END;

'

from DBA\_AUDIT\_POLICIES;

-- Script add policy

select 'BEGIN

DBMS\_FGA.add\_policy(object\_schema => '''||object\_schema||''',

object\_name => '''|| object\_name|| ''',

policy\_name => ''' ||policy\_name||''',

audit\_condition => ''sys.check\_ip\_app\_srv = 1'',

statement\_types => ''SELECT,INSERT,UPDATE,DELETE'',

handler\_schema => ''sys'',

handler\_module => ''pr\_fga\_auditing'',

enable => ''TRUE''

);

END;

'

from DBA\_AUDIT\_POLICIES;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230627 VALUES LESS THAN (TO\_DATE('2023-06-28 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230628 VALUES LESS THAN (TO\_DATE('2023-06-29 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230629 VALUES LESS THAN (TO\_DATE('2023-06-30 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230630 VALUES LESS THAN (TO\_DATE('2023-07-01 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230701 VALUES LESS THAN (TO\_DATE('2023-07-02 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230702 VALUES LESS THAN (TO\_DATE('2023-07-03 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230703 VALUES LESS THAN (TO\_DATE('2023-07-04 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230704 VALUES LESS THAN (TO\_DATE('2023-07-05 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230705 VALUES LESS THAN (TO\_DATE('2023-07-06 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230706 VALUES LESS THAN (TO\_DATE('2023-07-07 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230707 VALUES LESS THAN (TO\_DATE('2023-07-08 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230708 VALUES LESS THAN (TO\_DATE('2023-07-09 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230709 VALUES LESS THAN (TO\_DATE('2023-07-10 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230710 VALUES LESS THAN (TO\_DATE('2023-07-11 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230711 VALUES LESS THAN (TO\_DATE('2023-07-12 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230712 VALUES LESS THAN (TO\_DATE('2023-07-13 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230713 VALUES LESS THAN (TO\_DATE('2023-07-14 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230714 VALUES LESS THAN (TO\_DATE('2023-07-15 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230715 VALUES LESS THAN (TO\_DATE('2023-07-16 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230716 VALUES LESS THAN (TO\_DATE('2023-07-17 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230717 VALUES LESS THAN (TO\_DATE('2023-07-18 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230718 VALUES LESS THAN (TO\_DATE('2023-07-19 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230719 VALUES LESS THAN (TO\_DATE('2023-07-20 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230720 VALUES LESS THAN (TO\_DATE('2023-07-21 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230721 VALUES LESS THAN (TO\_DATE('2023-07-22 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230722 VALUES LESS THAN (TO\_DATE('2023-07-23 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230723 VALUES LESS THAN (TO\_DATE('2023-07-24 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230724 VALUES LESS THAN (TO\_DATE('2023-07-25 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230725 VALUES LESS THAN (TO\_DATE('2023-07-26 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230726 VALUES LESS THAN (TO\_DATE('2023-07-27 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230727 VALUES LESS THAN (TO\_DATE('2023-07-28 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230728 VALUES LESS THAN (TO\_DATE('2023-07-29 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230729 VALUES LESS THAN (TO\_DATE('2023-07-30 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230730 VALUES LESS THAN (TO\_DATE('2023-07-31 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230731 VALUES LESS THAN (TO\_DATE('2023-08-01 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230801 VALUES LESS THAN (TO\_DATE('2023-08-02 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230802 VALUES LESS THAN (TO\_DATE('2023-08-03 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230803 VALUES LESS THAN (TO\_DATE('2023-08-04 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230804 VALUES LESS THAN (TO\_DATE('2023-08-05 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230805 VALUES LESS THAN (TO\_DATE('2023-08-06 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230806 VALUES LESS THAN (TO\_DATE('2023-08-07 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230807 VALUES LESS THAN (TO\_DATE('2023-08-08 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230808 VALUES LESS THAN (TO\_DATE('2023-08-09 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230809 VALUES LESS THAN (TO\_DATE('2023-08-10 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230810 VALUES LESS THAN (TO\_DATE('2023-08-11 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230811 VALUES LESS THAN (TO\_DATE('2023-08-12 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230812 VALUES LESS THAN (TO\_DATE('2023-08-13 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230813 VALUES LESS THAN (TO\_DATE('2023-08-14 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230814 VALUES LESS THAN (TO\_DATE('2023-08-15 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230815 VALUES LESS THAN (TO\_DATE('2023-08-16 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230816 VALUES LESS THAN (TO\_DATE('2023-08-17 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230817 VALUES LESS THAN (TO\_DATE('2023-08-18 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230818 VALUES LESS THAN (TO\_DATE('2023-08-19 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230819 VALUES LESS THAN (TO\_DATE('2023-08-20 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230820 VALUES LESS THAN (TO\_DATE('2023-08-21 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230821 VALUES LESS THAN (TO\_DATE('2023-08-22 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230822 VALUES LESS THAN (TO\_DATE('2023-08-23 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230823 VALUES LESS THAN (TO\_DATE('2023-08-24 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230824 VALUES LESS THAN (TO\_DATE('2023-08-25 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230825 VALUES LESS THAN (TO\_DATE('2023-08-26 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230826 VALUES LESS THAN (TO\_DATE('2023-08-27 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230827 VALUES LESS THAN (TO\_DATE('2023-08-28 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230828 VALUES LESS THAN (TO\_DATE('2023-08-29 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230829 VALUES LESS THAN (TO\_DATE('2023-08-30 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230830 VALUES LESS THAN (TO\_DATE('2023-08-31 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230831 VALUES LESS THAN (TO\_DATE('2023-09-01 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230901 VALUES LESS THAN (TO\_DATE('2023-09-02 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230902 VALUES LESS THAN (TO\_DATE('2023-09-03 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230903 VALUES LESS THAN (TO\_DATE('2023-09-04 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230904 VALUES LESS THAN (TO\_DATE('2023-09-05 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230905 VALUES LESS THAN (TO\_DATE('2023-09-06 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230906 VALUES LESS THAN (TO\_DATE('2023-09-07 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230907 VALUES LESS THAN (TO\_DATE('2023-09-08 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230908 VALUES LESS THAN (TO\_DATE('2023-09-09 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230909 VALUES LESS THAN (TO\_DATE('2023-09-10 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230910 VALUES LESS THAN (TO\_DATE('2023-09-11 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230911 VALUES LESS THAN (TO\_DATE('2023-09-12 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230912 VALUES LESS THAN (TO\_DATE('2023-09-13 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230913 VALUES LESS THAN (TO\_DATE('2023-09-14 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230914 VALUES LESS THAN (TO\_DATE('2023-09-15 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230915 VALUES LESS THAN (TO\_DATE('2023-09-16 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230916 VALUES LESS THAN (TO\_DATE('2023-09-17 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230917 VALUES LESS THAN (TO\_DATE('2023-09-18 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230918 VALUES LESS THAN (TO\_DATE('2023-09-19 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230919 VALUES LESS THAN (TO\_DATE('2023-09-20 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230920 VALUES LESS THAN (TO\_DATE('2023-09-21 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230921 VALUES LESS THAN (TO\_DATE('2023-09-22 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230922 VALUES LESS THAN (TO\_DATE('2023-09-23 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230923 VALUES LESS THAN (TO\_DATE('2023-09-24 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230924 VALUES LESS THAN (TO\_DATE('2023-09-25 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230925 VALUES LESS THAN (TO\_DATE('2023-09-26 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230926 VALUES LESS THAN (TO\_DATE('2023-09-27 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230927 VALUES LESS THAN (TO\_DATE('2023-09-28 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230928 VALUES LESS THAN (TO\_DATE('2023-09-29 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230929 VALUES LESS THAN (TO\_DATE('2023-09-30 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20230930 VALUES LESS THAN (TO\_DATE('2023-10-01 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231001 VALUES LESS THAN (TO\_DATE('2023-10-02 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231002 VALUES LESS THAN (TO\_DATE('2023-10-03 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231003 VALUES LESS THAN (TO\_DATE('2023-10-04 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231004 VALUES LESS THAN (TO\_DATE('2023-10-05 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231005 VALUES LESS THAN (TO\_DATE('2023-10-06 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231006 VALUES LESS THAN (TO\_DATE('2023-10-07 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231007 VALUES LESS THAN (TO\_DATE('2023-10-08 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231008 VALUES LESS THAN (TO\_DATE('2023-10-09 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231009 VALUES LESS THAN (TO\_DATE('2023-10-10 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231010 VALUES LESS THAN (TO\_DATE('2023-10-11 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231011 VALUES LESS THAN (TO\_DATE('2023-10-12 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231012 VALUES LESS THAN (TO\_DATE('2023-10-13 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231013 VALUES LESS THAN (TO\_DATE('2023-10-14 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231014 VALUES LESS THAN (TO\_DATE('2023-10-15 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231015 VALUES LESS THAN (TO\_DATE('2023-10-16 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231016 VALUES LESS THAN (TO\_DATE('2023-10-17 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231017 VALUES LESS THAN (TO\_DATE('2023-10-18 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231018 VALUES LESS THAN (TO\_DATE('2023-10-19 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231019 VALUES LESS THAN (TO\_DATE('2023-10-20 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231020 VALUES LESS THAN (TO\_DATE('2023-10-21 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231021 VALUES LESS THAN (TO\_DATE('2023-10-22 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231022 VALUES LESS THAN (TO\_DATE('2023-10-23 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231023 VALUES LESS THAN (TO\_DATE('2023-10-24 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231024 VALUES LESS THAN (TO\_DATE('2023-10-25 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231025 VALUES LESS THAN (TO\_DATE('2023-10-26 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231026 VALUES LESS THAN (TO\_DATE('2023-10-27 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231027 VALUES LESS THAN (TO\_DATE('2023-10-28 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231028 VALUES LESS THAN (TO\_DATE('2023-10-29 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231029 VALUES LESS THAN (TO\_DATE('2023-10-30 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231030 VALUES LESS THAN (TO\_DATE('2023-10-31 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231031 VALUES LESS THAN (TO\_DATE('2023-11-01 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231101 VALUES LESS THAN (TO\_DATE('2023-11-02 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231102 VALUES LESS THAN (TO\_DATE('2023-11-03 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231103 VALUES LESS THAN (TO\_DATE('2023-11-04 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231104 VALUES LESS THAN (TO\_DATE('2023-11-05 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231105 VALUES LESS THAN (TO\_DATE('2023-11-06 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231106 VALUES LESS THAN (TO\_DATE('2023-11-07 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231107 VALUES LESS THAN (TO\_DATE('2023-11-08 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231108 VALUES LESS THAN (TO\_DATE('2023-11-09 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231109 VALUES LESS THAN (TO\_DATE('2023-11-10 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231110 VALUES LESS THAN (TO\_DATE('2023-11-11 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231111 VALUES LESS THAN (TO\_DATE('2023-11-12 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231112 VALUES LESS THAN (TO\_DATE('2023-11-13 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231113 VALUES LESS THAN (TO\_DATE('2023-11-14 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231114 VALUES LESS THAN (TO\_DATE('2023-11-15 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231115 VALUES LESS THAN (TO\_DATE('2023-11-16 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231116 VALUES LESS THAN (TO\_DATE('2023-11-17 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231117 VALUES LESS THAN (TO\_DATE('2023-11-18 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231118 VALUES LESS THAN (TO\_DATE('2023-11-19 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231119 VALUES LESS THAN (TO\_DATE('2023-11-20 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231120 VALUES LESS THAN (TO\_DATE('2023-11-21 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231121 VALUES LESS THAN (TO\_DATE('2023-11-22 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231122 VALUES LESS THAN (TO\_DATE('2023-11-23 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231123 VALUES LESS THAN (TO\_DATE('2023-11-24 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231124 VALUES LESS THAN (TO\_DATE('2023-11-25 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231125 VALUES LESS THAN (TO\_DATE('2023-11-26 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231126 VALUES LESS THAN (TO\_DATE('2023-11-27 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231127 VALUES LESS THAN (TO\_DATE('2023-11-28 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231128 VALUES LESS THAN (TO\_DATE('2023-11-29 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231129 VALUES LESS THAN (TO\_DATE('2023-11-30 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231130 VALUES LESS THAN (TO\_DATE('2023-12-01 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231201 VALUES LESS THAN (TO\_DATE('2023-12-02 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231202 VALUES LESS THAN (TO\_DATE('2023-12-03 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231203 VALUES LESS THAN (TO\_DATE('2023-12-04 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231204 VALUES LESS THAN (TO\_DATE('2023-12-05 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231205 VALUES LESS THAN (TO\_DATE('2023-12-06 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231206 VALUES LESS THAN (TO\_DATE('2023-12-07 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231207 VALUES LESS THAN (TO\_DATE('2023-12-08 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231208 VALUES LESS THAN (TO\_DATE('2023-12-09 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231209 VALUES LESS THAN (TO\_DATE('2023-12-10 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231210 VALUES LESS THAN (TO\_DATE('2023-12-11 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231211 VALUES LESS THAN (TO\_DATE('2023-12-12 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231212 VALUES LESS THAN (TO\_DATE('2023-12-13 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231213 VALUES LESS THAN (TO\_DATE('2023-12-14 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231214 VALUES LESS THAN (TO\_DATE('2023-12-15 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231215 VALUES LESS THAN (TO\_DATE('2023-12-16 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231216 VALUES LESS THAN (TO\_DATE('2023-12-17 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231217 VALUES LESS THAN (TO\_DATE('2023-12-18 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231218 VALUES LESS THAN (TO\_DATE('2023-12-19 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231219 VALUES LESS THAN (TO\_DATE('2023-12-20 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231220 VALUES LESS THAN (TO\_DATE('2023-12-21 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231221 VALUES LESS THAN (TO\_DATE('2023-12-22 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231222 VALUES LESS THAN (TO\_DATE('2023-12-23 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231223 VALUES LESS THAN (TO\_DATE('2023-12-24 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231224 VALUES LESS THAN (TO\_DATE('2023-12-25 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231225 VALUES LESS THAN (TO\_DATE('2023-12-26 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231226 VALUES LESS THAN (TO\_DATE('2023-12-27 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231227 VALUES LESS THAN (TO\_DATE('2023-12-28 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231228 VALUES LESS THAN (TO\_DATE('2023-12-29 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231229 VALUES LESS THAN (TO\_DATE('2023-12-30 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231230 VALUES LESS THAN (TO\_DATE('2023-12-31 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

alter table sys.FGA\_AUDIT\_LOG add PARTITION DATA20231231 VALUES LESS THAN (TO\_DATE('2024-01-01 00:00:00','SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) LOGGING TABLESPACE DATA\_AUDIT;

## Audit DDL:

--Cấu hình Audit DDL Log Oracle Database

--Mục đích: Lưu log các thao tác DDL (drop table, create table, drop procedure, drop package,...) để truy vết các thao tác của DBA hay quản trị ứng dụng (có dùng user DB) làm sai, nhầm.

create tablespace data\_log datafile size 1M autoextend on next 10m;

create tablespace indx\_log datafile size 1M autoextend on next 10m;

--1. Tạo bản lưu log

--drop table SYS.DDL\_LOG purge;

--alter table SYS.DDL\_LOG modify SQL\_TEXT VARCHAR2(4000 BYTE);

CREATE TABLE SYS.DDL\_LOG

(

USER\_NAME VARCHAR2(30 BYTE),

IP\_ADDRESS VARCHAR2(30 BYTE),

HOSTNAME VARCHAR2(30 BYTE),

DDL\_DATE DATE,

DDL\_TYPE VARCHAR2(30 BYTE),

OBJECT\_TYPE VARCHAR2(18 BYTE),

OWNER VARCHAR2(30 BYTE),

OBJECT\_NAME VARCHAR2(128 BYTE),

SQL\_TEXT VARCHAR2(2000 BYTE)

)

TABLESPACE DATA\_LOG

PCTUSED 0

PCTFREE 10

INITRANS 1

MAXTRANS 255

STORAGE (

INITIAL 64K

NEXT 1M

MINEXTENTS 1

MAXEXTENTS UNLIMITED

PCTINCREASE 0

BUFFER\_POOL DEFAULT

)

LOGGING

NOCOMPRESS

NOCACHE

MONITORING;

CREATE INDEX SYS.DDL\_LOG\_I1 ON SYS.DDL\_LOG

(DDL\_DATE)

LOGGING

TABLESPACE INDX\_LOG

PCTFREE 10

INITRANS 2

MAXTRANS 255

STORAGE (

INITIAL 64K

NEXT 1M

MINEXTENTS 1

MAXEXTENTS UNLIMITED

PCTINCREASE 0

BUFFER\_POOL DEFAULT

);

--2. Tạo trigger bắt log DDL

CREATE OR REPLACE TRIGGER SYS.ddl\_trig

AFTER DDL

ON DATABASE

--ON binhtv.tab1 -- Chỉ giám sát 1 bảng

DECLARE

ip\_addr VARCHAR2 (30);

hostname VARCHAR2 (30);

l\_count NUMBER;

l\_sql\_text ora\_name\_list\_t;

l\_sql VARCHAR2 (4000);

BEGIN

l\_count := ora\_sql\_txt (l\_sql\_text);

IF l\_count > 1

THEN

l\_sql := l\_sql\_text (1) || l\_sql\_text (2);

ELSE

l\_sql := l\_sql\_text (1);

END IF;

SELECT SYS\_CONTEXT ('USERENV', 'IP\_ADDRESS') INTO ip\_addr FROM DUAL;

SELECT SYS\_CONTEXT ('USERENV', 'HOST') INTO hostname FROM DUAL;

--IF (ip\_addr NOT LIKE '192.168.1.%')

--THEN

INSERT INTO ddl\_log (user\_name,ip\_address,hostname,ddl\_date,ddl\_type,object\_type,

owner,object\_name,sql\_text)

VALUES (ora\_login\_user,ip\_addr,hostname,SYSDATE,ora\_sysevent,ora\_dict\_obj\_type,

ora\_dict\_obj\_owner,ora\_dict\_obj\_name,l\_sql);

--END IF;

END ddl\_trig;

/

--3. Test

--SYS

CREATE TABLE oaz19.tab11 (col1 DATE);

DROP TABLE oaz19.tab11;

--oaz19

CREATE TABLE oaz19.tab11 (col1 DATE);

truncate table oaz19.tab11;

DROP TABLE oaz19.tab11;

CREATE OR REPLACE FUNCTION oaz19.check\_ip\_machine

return number

is

begin

-- if (SYS\_CONTEXT('USERENV', 'SESSION\_USER') <> 'BINHTV'

-- and (SYS\_CONTEXT ('USERENV', 'IP\_ADDRESS') like '192.168.1%' or SYS\_CONTEXT ('USERENV', 'IP\_ADDRESS') like '1192.168.2%'

-- )) then

if (SYS\_CONTEXT ('USERENV', 'IP\_ADDRESS') not like '192.168.1.%' and

SYS\_CONTEXT ('USERENV', 'IP\_ADDRESS') not like '192.168.2.%' and

SYS\_CONTEXT ('USERENV', 'IP\_ADDRESS') not like '192.168.3.%') then

return 1;

else

return 0;

end if;

end;

--4. Kiểm tra log

SELECT \* FROM sys.DDL\_LOG order by ddl\_date desc;

SELECT \* FROM oaz11."BIN$2aKBPDtiQKfgU4C2qMAStA==$0";

select \* from oaz19.tab11 ;

select \* from dba\_recyclebin;

purge dba\_recyclebin;

select \* from dba\_segments where tablespace\_name='USERS';

-- Flashback before drop

flashback table oaz19.tab11 to before drop;

select \* from oaz19.tab11 ;

## Audit Trail:

-- Cấu hình extended auditing.

CONN sys/password AS SYSDBA

--CDB

ALTER SYSTEM SET audit\_trail=DB,EXTENDED SCOPE=SPFILE;

SHUTDOWN IMMEDIATE

STARTUP

TRUNCATE TABLE aud$;

TRUNCATE TABLE fga\_log$;

select \* from aud$;

select \* from fga\_log$;

create table oaz18.BANG\_NHO as select \* from oaz18.bang\_to where rownum < 10;

-- All la DDL

AUDIT ALL BY user1 BY ACCESS;

--All la DDL

grant all on OAZ18.BANG\_TO to user1;

grant all on OAZ18.BANG\_NHO to user1;

-- Audit DML

AUDIT insert,update,delete on oaz18.BANG\_TO BY ACCESS;

AUDIT insert,update,delete on oaz18.BANG\_NHO BY ACCESS;

CREATE AUDIT POLICY sl\_BANG\_NHO\_pol ACTIONS select on OAZ18.BANG\_NHO;

CREATE AUDIT POLICY audit\_objpriv\_pol6 ACTIONS ALL;

ALTER AUDIT POLICY audit\_objpriv\_pol6

ADD ACTIONS CREATE TABLE, DROP TABLE, TRUNCATE TABLE;

ALTER AUDIT POLICY audit\_objpriv\_pol6

ADD ACTIONS SELECT, UPDATE, DELETE, INSERT ON oaz18.bangnho;

ALTER AUDIT POLICY audit\_objpriv\_pol6

ADD ACTIONS SELECT, UPDATE, DELETE, INSERT ON user1.xxx1;

audit policy sl\_BANG\_NHO\_pol;

audit policy audit\_objpriv\_pol6;

noaudit policy audit\_objpriv\_pol6;

SELECT policy\_name, audit\_option, condition\_eval\_opt

FROM audit\_unified\_policies where policy\_name in ('AUDIT\_OBJPRIV\_POL6','SL\_BANG\_NHO\_POL');

SELECT \*

FROM audit\_unified\_enabled\_policies;

SELECT \*

FROM dba\_stmt\_audit\_opts;

grant dba to user1;

-- Thực hiện audit

CONN user1/oracle@pdb1;

select \* from oaz18.BANG\_NHO;

UPDATE oaz18.BANG\_NHO SET address = address||'\_NEW';

commit;

a

-- Kiểm tra dữ liệu audit trail

SELECT \* FROM dba\_common\_audit\_trail order by extended\_timestamp desc;

select \* from aud$ order by ntimestamp# desc;

SQL\_TEXT

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

UPDATE oaz18.BANG\_TO SET col2 = cold2||'\_NEW'

UPDATE oaz18.BANG\_TO SET cold2 = cold2||'\_NEW'

## Trigger AU

create table sys.audit\_employees(os\_user varchar2(20), event\_date date, ip\_address varchar2(30), emp\_salold\_salnew varchar2(50));

CREATE OR REPLACE TRIGGER sys.hrsalary\_audit

AFTER UPDATE OF salary

ON hr.employees

REFERENCING NEW AS NEW OLD AS OLD FOR EACH ROW

BEGIN

IF :old.salary != :new.salary

THEN

INSERT INTO sys.audit\_employees

VALUES (sys\_context('userenv','os\_user'), sysdate, sys\_context('userenv','ip\_address'), :new.employee\_id ||

' salary changed from '||:old.salary|| ' to '||:new.salary);

END IF;

END;

/

select \* from sys.audit\_employees ;

## VPD:

SELECT \* FROM hr.employees;

/\*\*\*\*\* Chính sách VPD mức cột \*\*\*\*\*/

--Chính sách VPD được áp dụng cho toàn bộ row. Theo mặc định, Chính sách VPD ở mức cột cho phép bạn hạn chế các row được hiển thị chỉ khi các cột được chỉ định được truy cập.

--CONN sys/password@db10g AS SYSDBA

GRANT EXECUTE ON dbms\_rls TO hr;

CONN hr/oracle

-- Tạo hàm với các polcy để hạn chế quyền truy cập vào các cột SAL và COMM

-- nếu nhân viên không thuộc deptno=20.

CREATE OR REPLACE FUNCTION pf\_job (oowner IN VARCHAR2, ojname IN VARCHAR2)

RETURN VARCHAR2 AS

con VARCHAR2 (200);

BEGIN

con := 'department\_id = 20';

RETURN (con);

END pf\_job;

/

-- Áp các policy cho bảng.

BEGIN

DBMS\_RLS.ADD\_POLICY (object\_schema => 'hr',

object\_name => 'employees',

policy\_name => 'sp\_job',

function\_schema => 'hr',

policy\_function => 'pf\_job',

sec\_relevant\_cols => 'salary,commission\_pct');

END;

/

-- Chúng ta sẽ xem được tất cả các bản ghi nếu deptno khác 20, nhưng SAL, COMM không hiển thị

SELECT employee\_id, first\_name, job\_id,department\_id FROM employees;

100 Steven AD\_PRES 90

101 Neena AD\_VP 90

102 Lex AD\_VP 90

103 Alexander IT\_PROG 60

...

107 rows selected.

-- Các row hiển thị thêm thông tin SAL, COMM nếu deptno = 20

SELECT employee\_id, first\_name, job\_id, salary, commission\_pct FROM employees;

201 Michael MK\_MAN 13000

202 Pat MK\_REP 6000

2 row selected

5 rows selected.

-- Bỏ policy ra khỏi bảng.

BEGIN

DBMS\_RLS.DROP\_POLICY (object\_schema => 'hr',

object\_name => 'employees',

policy\_name => 'sp\_job');

END;

/

/\*\*\*\*\* Column Masking \*\*\*\*\*/

-- Tạo policy

BEGIN

DBMS\_RLS.ADD\_POLICY (object\_schema => 'hr',

object\_name => 'employees',

policy\_name => 'sp\_job',

function\_schema => 'hr',

policy\_function => 'pf\_job',

sec\_relevant\_cols => 'salary,commission\_pct',

sec\_relevant\_cols\_opt => DBMS\_RLS.ALL\_ROWS);

END;

/

-- Tất cả các row được trả về với SAL và COMM là NULL nhưng nếu deptno=20 sẽ hiển thị giá trị SAL và COMM

SELECT employee\_id, first\_name, job\_id, salary, commission\_pct FROM employees;

--SELECT empno, ename, job, sal, comm FROM emp;

EMPNO ENAME JOB SAL COMM

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

7369 SMITH CLERK 10000

7499 ALLEN SALESMAN

7521 WARD SALESMAN

7566 JONES MANAGER 2975

7654 MARTIN SALESMAN

7698 BLAKE MANAGER

7782 CLARK MANAGER

7788 SCOTT ANALYST 3000

7839 KING PRESIDENT

7844 TURNER SALESMAN

7876 ADAMS CLERK 1100

EMPNO ENAME JOB SAL COMM

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

7900 JAMES CLERK

7902 FORD ANALYST 3000

7934 MILLER CLERK

14 rows selected.

-- Xóa policy

BEGIN

DBMS\_RLS.DROP\_POLICY (object\_schema => 'hr',

object\_name => 'employees',

policy\_name => 'sp\_job');

END;

/

/\*\*\*\* DUNG VIEW \*\*\*/

select \* from hr.employees;

create view hr.v\_employees as select \* from hr.employees where department\_id=20;

select \* from hr.v\_employees;

## Tài liệu tham khảo:

<https://www.tranvanbinh.vn/2018/08/create-firewall-block-ip-on-oracle.html>

<https://www.tranvanbinh.vn/2022/11/chan-truy-cap-theo-ip-qua-listener.html>

<https://www.tranvanbinh.vn/2021/03/labcau-hinh-fga-trong-oracle-database.html>

<https://www.tranvanbinh.vn/2022/10/cau-hinh-fga-voi-bang-partition-trong.html>

<https://www.tranvanbinh.vn/2018/08/setup-fga-on-oracle-db.html>

<https://www.tranvanbinh.vn/2018/08/setup-audit-ddl-log-oracle-database.html>

<https://www.tranvanbinh.vn/2021/04/audit-thoi-iem-logon-logofff-cua-user.html>

<https://www.tranvanbinh.vn/2020/11/thu-tuc-thay-oi-tham-so-audittrail-csdl.html>

<https://www.tranvanbinh.vn/2021/08/bao-mat-trong-oracle-database-10g-11g.html>

<https://www.tranvanbinh.vn/2021/05/che-dau-du-lieu-voi-giai-phap-dung-view.html>

<https://www.tranvanbinh.vn/2021/08/ham-syscontext.html>

<https://www.tranvanbinh.vn/2022/01/chuyen-audit-aud-fgalog-ra-khoi.html>

<https://www.tranvanbinh.vn/2021/09/tim-hieu-trigger-trong-oracle.html>

# BÀI 12: NGUYÊN LÝ BACKUP, RECOVERY DATABASE

## Tài liệu tham khảo:

# BÀI 13: CẤU HÌNH BACKUP DATABASE

## Show all

[oracle@linux7 ~]$ bash

[oracle@linux7 ~]$ rman target /

RMAN> show all

using target database control file instead of recovery catalog

RMAN configuration parameters for database with db\_unique\_name binh are:

CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 4 DAYS;

#CONFIGURE RETENTION POLICY TO REDUNDANCY 1;

CONFIGURE BACKUP OPTIMIZATION ON;

CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default

CONFIGURE CONTROLFILE AUTOBACKUP ON;

CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '+reco/binh/backup\_controlfile/ctl\_%F';

CONFIGURE DEVICE TYPE DISK PARALLELISM 16 BACKUP TYPE TO COMPRESSED BACKUPSET;

CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 1;

CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 1;

CONFIGURE MAXSETSIZE TO UNLIMITED; # default

CONFIGURE ENCRYPTION FOR DATABASE OFF; # default

CONFIGURE ENCRYPTION ALGORITHM 'AES128'; # default

CONFIGURE COMPRESSION ALGORITHM 'BASIC' AS OF RELEASE 'DEFAULT' OPTIMIZE FOR LOAD TRUE;

CONFIGURE EXCLUDE FOR TABLESPACE 'DUMP';

CONFIGURE EXCLUDE FOR TABLESPACE 'INDX2017';

CONFIGURE ARCHIVELOG DELETION POLICY TO APPLIED ON STANDBY; #BINHTV

#CONFIGURE ARCHIVELOG DELETION POLICY CLEAR;

CONFIGURE ARCHIVELOG DELETION POLICY TO BACKED UP 1 TIMES TO DISK;

CONFIGURE SNAPSHOT CONTROLFILE NAME TO '+RECO/binh/backup\_controlfile/snapcf\_bill1.f';

CONFIGURE SNAPSHOT CONTROLFILE NAME TO '+reco/binh/backup\_controlfile/snapcf\_bill1.f';

## Cấu hình:

--1.Kiem tra control file: Dam bao co 2 control file

select \* from v$controlfile

--/u01/app/oracle/oradata/ORCL/controlfile/o1\_mf\_jngnomtw\_.ctl

--/u01/app/oracle/fast\_recovery\_area/ORCL/controlfile/o1\_mf\_jngnomvm\_.ctl

SQL> show parameter control;

--+ Them control file

SQL> alter system set control\_files='/u01/app/oracle/oradata/ORCL/controlfile/o1\_mf\_jngnomtw\_.ctl','/u01/app/oracle/fast\_recovery\_area/ORCL/controlfile/o1\_mf\_jngnomvm\_.ctl','/u01/app/oracle/oradata/ORCL/controlfile/control03.ctl','/u01/app/oracle/oradata/ORCL/controlfile/control04.ctl' scope=spfile;

SQL> shutdown immediate;

SQL> exit

$ cp /u01/app/oracle/oradata/ORCL/controlfile/o1\_mf\_jngnomtw\_.ctl /u01/app/oracle/oradata/ORCL/controlfile/control03.ctl

[oracle@linux7 ~]$ sqlplus / as sysdba

SQL> startup

SQL> show parameter control;

--2.Cau hinh phan vung fast recovery area mac danh

select \* from v$parameter where name like '%recovery\_file%'

SQL> show parameter db\_recovery\_file\_dest;

[root@linux7 ~]# mkdir /fra

[root@linux7 fra]# chown -Rf oracle:oinstall /fra

SQL> alter system set db\_recovery\_file\_dest='/fra';

SQL> show parameter db\_recovery\_file\_dest\_size;

SQL> alter system set DB\_RECOVERY\_FILE\_DEST\_SIZE=14G;

SQL> show parameter db\_recovery\_file\_dest;

--3.Cau hinh nhieu redo log group

select \* from v$log;

--1 ONLINE /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_1\_l6q61bd3\_.log NO 0

--2 ONLINE /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_2\_l6q61bdq\_.log NO 0

--4 ONLINE /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_4\_l6q61c8h\_.log NO 0

--4 ONLINE /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo04bb.log NO 0

--5 ONLINE /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_5\_l6q61c73\_.log NO 0

--5 ONLINE /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo05bb.log NO 0

select \* from v$logfile order by 1;

alter system switch logfile;

-- Kiem tra xem co OMF khong

SQL> show parameter online;

NAME TYPE VALUE

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

db\_create\_online\_log\_dest\_1 string /u01/app/oracle/redolog

-- Add Log member

ALTER DATABASE ADD LOGFILE MEMBER '/u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo01b.log' TO GROUP 1;

ALTER DATABASE ADD LOGFILE MEMBER '/u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo02b.log' TO GROUP 2;

--OMF

--ALTER DATABASE ADD LOGFILE GROUP 1 size 200m;

ALTER DATABASE ADD LOGFILE MEMBER '/u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo01b.log' to group 1;

alter system switch logfile;

-- Xoa

Alter database drop logfile member '/u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/o1\_mf\_1\_jngnotg1\_.log';

Alter database drop logfile member '/u01/app/oracle/oradata/ORCL/onlinelog/redo01c.log';

Alter database drop logfile member '/u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/o1\_mf\_2\_jngnotfp\_.log';

Alter database drop logfile member '/u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo01b.log';

-- Them member redo log vao redo log group

ALTER DATABASE ADD LOGFILE MEMBER '/u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo01a.log' TO GROUP 1;

ALTER DATABASE ADD LOGFILE MEMBER '/u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo01bb.log' TO GROUP 2;

ALTER DATABASE ADD LOGFILE MEMBER '/u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo01c.log' TO GROUP 3;

Alter database drop logfile group 5;

Alter database drop logfile group 3;

Alter database drop logfile group 4;

--4.Dat CSDL o che do ARCHIVELOG

--select name, OPEN\_MODE,Log\_mode from v$database;

select \* from v$database;

SQL> archive log list;

Database log mode Archive Mode

Automatic archival Enabled

Archive destination USE\_DB\_RECOVERY\_FILE\_DEST

Oldest online log sequence 18

Next log sequence to archive 20

Current log sequence 20

SQL> show parameter recovery;

NAME TYPE VALUE

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

db\_recovery\_file\_dest string /backup

db\_recovery\_file\_dest\_size big integer 15G

recovery\_parallelism integer 0

remote\_recovery\_file\_dest string

## Fashback data archive

--Flashback Data Archive (FDA) trong Oracle Database 11g

--Flashback Data Archive (FDA) , còn được gọi là Flashback Archive (FBA), được giới thiệu từ Oracle 11g để cung cấp khả năng lưu trữ lâu dài dữ liệu undo, cho phép thực hiện các hoạt động flashback dựa trên undo trong một khoảng thời gian dài.

--Tạo tablespace để FDA lưu 1 năm, tablespace khác lưu 2 năm, user test cần quyền flashback archive

CREATE TABLESPACE fda\_ts DATAFILE

SIZE 1M AUTOEXTEND ON NEXT 1M;

CREATE TABLESPACE fda\_ts2 DATAFILE

SIZE 1M AUTOEXTEND ON NEXT 1M;

CREATE FLASHBACK ARCHIVE DEFAULT fda\_1year TABLESPACE fda\_ts

QUOTA 5G RETENTION 1 YEAR;

CREATE FLASHBACK ARCHIVE fda\_2year TABLESPACE fda\_ts

RETENTION 2 YEAR;

--Có 3 kiểu cần quản lý:

--Quản lý Tablespace .

ALTER FLASHBACK ARCHIVE fda\_1year SET DEFAULT;

ALTER FLASHBACK ARCHIVE fda\_1year ADD TABLESPACE fda\_ts2 QUOTA 5G;

ALTER FLASHBACK ARCHIVE fda\_1year ADD TABLESPACE fda\_ts2;

ALTER FLASHBACK ARCHIVE fda\_1year REMOVE TABLESPACE fda\_ts2;

ALTER FLASHBACK ARCHIVE fda\_1year MODIFY TABLESPACE fda\_ts2 QUOTA 6G;

ALTER FLASHBACK ARCHIVE fda\_2year MODIFY TABLESPACE fda\_ts2;

ALTER FLASHBACK ARCHIVE fda\_1year REMOVE TABLESPACE fda\_ts2;

--Chỉnh sửa thành 3 năm

ALTER FLASHBACK ARCHIVE fda\_1year MODIFY RETENTION 3 YEAR;

--Xóa data.

-- Xóa toàn bộ dữ liệu lịch sử

ALTER FLASHBACK ARCHIVE fda\_1year PURGE ALL;

-- Xóa mọi dữ liệu theo thời gian nhất định

ALTER FLASHBACK ARCHIVE fda\_1year PURGE BEFORE TIMESTAMP (SYSTIMESTAMP - INTERVAL '1' DAY);

-- Xóa mọi dữ liệu trước SCN.

ALTER FLASHBACK ARCHIVE fba\_name PURGE BEFORE SCN 728969;

--Drop tablespace lưu flashback:

DROP FLASHBACK ARCHIVE fba\_name;

--Tạo user:

CONN sys/password AS SYSDBA

CREATE USER fda\_test\_user IDENTIFIED BY oracle

QUOTA UNLIMITED ON users

QUOTA UNLIMITED ON fda\_ts;

GRANT CONNECT, CREATE TABLE TO fda\_test\_user;

GRANT FLASHBACK ARCHIVE ON fda\_1year TO fda\_test\_user;

GRANT FLASHBACK ARCHIVE ON fda\_2year TO fda\_test\_user;

--Tạo bảng:

alter user fda\_test\_user identified by oracle;

CONN fda\_test\_user/oracle

CREATE TABLE fda\_test\_user.test\_tab\_1 (

id NUMBER,

desription VARCHAR2(50),

CONSTRAINT test\_tab\_1\_pk PRIMARY KEY (id)

)

FLASHBACK ARCHIVE;

--Nếu tạo bảng thiếu quyền FLASHBACK ARCHIVE sẽ hiển thị lỗi:

CONN fda\_test\_user/oracle

CREATE TABLE test\_tab\_2 (

id NUMBER,

desription VARCHAR2(50),

CONSTRAINT test\_tab\_2\_pk PRIMARY KEY (id)

)

FLASHBACK ARCHIVE fda\_2year;

CREATE TABLE test\_tab\_2 (

\*

ERROR at line 1:

ORA-55620: No privilege to use Flashback Archive

SQL>

--Thiết lập flashback cho bảng:

-- Kích hoạt sử dụng FBDA mặc định.

ALTER TABLE fda\_test\_user.test\_tab\_1 FLASHBACK ARCHIVE;

-- Cho phép sử dụng FBDA cụ thể.

ALTER TABLE fda\_test\_user.test\_tab\_1 FLASHBACK ARCHIVE fda\_2year;

-- Tắt FBDA

ALTER TABLE fda\_test\_user.test\_tab\_1 NO FLASHBACK ARCHIVE;

--Nếu alter thiếu quyền FLASHBACK ARCHIVE ADMINISTER thì báo lỗi:

SQL> ALTER TABLE test\_tab\_1 NO FLASHBACK ARCHIVE;

ALTER TABLE test\_tab\_1 NO FLASHBACK ARCHIVE

\*

ERROR at line 1:

ORA-55620: No privilege to use Flashback Archive

SQL>

--Kiểm tra lại từ view %\_FLASHBACK\_ARCHIVE:

CONN sys/password AS SYSDBA

COLUMN flashback\_archive\_name FORMAT A20

SELECT flashback\_archive\_name, retention\_in\_days, status

FROM dba\_flashback\_archive;

FLASHBACK\_ARCHIVE\_NA RETENTION\_IN\_DAYS STATUS

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

FDA\_2YEAR 730

FDA\_1YEAR 365 DEFAULT

2 rows selected.

SQL>

--View %\_FLASHBACK\_ARCHIVE\_TS hiển thị thông tin tablespaces và quotas của flashback archive

COLUMN flashback\_archive\_name FORMAT A20

COLUMN quota\_in\_mb FORMAT A10

SELECT flashback\_archive\_name, tablespace\_name, quota\_in\_mb

FROM dba\_flashback\_archive\_ts;

FLASHBACK\_ARCHIVE\_NA TABLESPACE\_NAME QUOTA\_IN\_M

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

FDA\_2YEAR FDA\_TS

FDA\_1YEAR FDA\_TS 10240

2 rows selected.

SQL>

--View %\_FLASHBACK\_ARCHIVE\_TABLES hiển thị flashback archive, tên bảng table lưu dữ liệu lịch sử:

COLUMN table\_name FORMAT A15

COLUMN owner\_name FORMAT A15

COLUMN flashback\_archive\_name FORMAT A20

COLUMN archive\_table\_name FORMAT A20

SELECT table\_name, owner\_name, flashback\_archive\_name, archive\_table\_name

FROM dba\_flashback\_archive\_tables;

TABLE\_NAME OWNER\_NAME FLASHBACK\_ARCHIVE\_NA ARCHIVE\_TABLE\_NAME

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

TEST\_TAB\_1 FDA\_TEST\_USER FDA\_1YEAR SYS\_FBA\_HIST\_72023

1 row selected.

SQL>

select rowid, a.\* from fda\_test\_user.test\_tab\_1 a;

## Tài liệu tham khảo:

<https://www.tranvanbinh.vn/2018/08/configure-rman-parameters.html>

<https://www.tranvanbinh.vn/2022/03/oracle-fast-recovery-area-fra-trong.html>

<https://www.tranvanbinh.vn/2020/11/quan-tri-control-files.html>

<https://www.tranvanbinh.vn/2020/11/quan-tri-redo-log-files.html>

<https://www.tranvanbinh.vn/2021/02/cach-xu-ly-loi-ora-19815-warning.html>

<https://www.tranvanbinh.vn/2020/10/tong-hop-cau-lenh-rman-trong-oracle.html>

<https://www.tranvanbinh.vn/2018/09/rman-backup-command.html>

<https://www.tranvanbinh.vn/2020/10/tong-hop-cau-lenh-rman-trong-oracle.html>

<https://www.tranvanbinh.vn/2021/09/flashback-data-archive-fda-trong-oracle.html>

# BÀI 14: BACKUP DATABASE

## Backup tự động

Tự động Backup Oracle Database hàng ngày qua Crontab\_New

Mục đích: Tự động Backup Oracle Database hàng ngày qua Crontab

1. Tạo script backup:

1.1. Tạo scritp backup level 0

$ mkdir ~/backup (hoặc cd /home/oracle/backup)

$ cd ~/backup

$ mkdir logs

$ mkdir level0

$ mkdir level1

oracle@dbaviet-db02 # vi /home/oracle/backup/level0.rman

#################################

# Script for backup full database dbaviet

# Created Date 13/10/2020

CONFIGURE CONTROLFILE AUTOBACKUP ON;

CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '/home/oracle/backup/level0/auto\_dbaviet\_ctl%F';

run {

CROSSCHECK BACKUP;

DELETE NOPROMPT EXPIRED BACKUP;

DELETE NOPROMPT OBSOLETE;

CROSSCHECK ARCHIVELOG ALL;

delete noprompt expired archivelog all;

BACKUP AS COMPRESSED BACKUPSET DEVICE TYPE DISK INCREMENTAL LEVEL 0 DATABASE FORMAT '/home/oracle/backup/level0/dbaviet\_%T\_%d\_%u\_%s' FILESPERSET 10 MAXSETSIZE 40G TAG LEVEL0;

sql 'ALTER SYSTEM ARCHIVE LOG CURRENT';

BACKUP AS COMPRESSED BACKUPSET DEVICE TYPE DISK ARCHIVELOG ALL FORMAT '/home/oracle/backup/level0/arc\_%T\_%d\_%u\_%s\_%U' FILESPERSET 10 TAG ARCH;

DELETE ARCHIVELOG UNTIL TIME 'SYSDATE-2';

#sql ' BEGIN TEST\_SEND\_MAIL\_BACKUP; END; ';

}

EXIT;

oracle@dbaviet-db02 # vi /home/oracle/backup/level0.sh

#################################

# Script for backup full database dbaviet

# Created Date 13/10/2020

#su - oracle

logfile=`date +%Y%m%d`\_level0.log

export ORACLE\_SID=orcl

export NLS\_DATE\_FORMAT="yyyy-mm-dd hh24:mi:ss"

export ORACLE\_BASE=/u01/app/oracle

export ORACLE\_HOME=/u01/app/oracle/product/19.0.0/dbhome\_1

export PATH=$ORACLE\_HOME/bin:$PATH

rman target / nocatalog cmdfile /home/oracle/backup/level0.rman log /home/oracle/backup/logs/$logfile

exit

1.2. Tạo script backup level 1

oracle@dbaviet-db02 # vi /home/oracle/backup/level1.rman

#################################

# Script for backup incremental database dbaviet

# Created Date 13/10/2020

CONFIGURE CONTROLFILE AUTOBACKUP ON;

CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '/home/oracle/backup/level1/auto\_dbaviet\_ctl%F';

run {

CROSSCHECK BACKUP;

DELETE NOPROMPT EXPIRED BACKUP;

DELETE NOPROMPT OBSOLETE;

CROSSCHECK ARCHIVELOG ALL;

delete noprompt expired archivelog all;

BACKUP AS COMPRESSED BACKUPSET DEVICE TYPE DISK INCREMENTAL LEVEL 1 DATABASE FORMAT '/home/oracle/backup/level1/dbaviet\_%T\_%d\_%u\_%s' FILESPERSET 10 MAXSETSIZE 80G TAG LEVEL1;

sql 'ALTER SYSTEM ARCHIVE LOG CURRENT';

BACKUP AS COMPRESSED BACKUPSET DEVICE TYPE DISK ARCHIVELOG ALL FORMAT '/home/oracle/backup/level1/arc\_%T\_%d\_%u\_%s\_%U' FILESPERSET 10 TAG ARCH;

DELETE ARCHIVELOG UNTIL TIME 'SYSDATE-1';

#sql ' BEGIN TEST\_SEND\_MAIL\_BACKUP; END; ';

}

EXIT;

oracle@dbaviet-db02 # vi /home/oracle/backup/level1.sh

#################################

# Script for backup incremental database dbaviet

# Created Date 13/10/2020

logfile=`date +%Y%m%d`\_level1.log

export ORACLE\_SID=db12c

export NLS\_DATE\_FORMAT="yyyy-mm-dd hh24:mi:ss"

export ORACLE\_BASE=/u01/app/oracle

export ORACLE\_HOME=/u01/app/oracle/product/19.0.0/dbhome\_1

export PATH=$ORACLE\_HOME/bin:$PATH

rman target / nocatalog cmdfile /home/oracle/backup/level1.rman log /home/oracle/backup/logs/$logfile

exit

2. Phân quyền chạy

$ cd /home/oracle/backup/

$ chmod +x \*.sh

Yêu cầu phải có đầy đru các file sau:

(file swp bỏ qua)

3. Tạo crontab backup tự động trên OS, user oracle

#########backupdb\_dbaviet########

Vào bằng user oracle

(oracle)$ crontab -e

00 00 \* \* 0,3 /home/oracle/backup/level0.sh #00h ngày thứ 4 và chủ nhật backup level 0

00 00 \* \* 1,2,4,5,6 /home/oracle/backup/level1.sh #Các ngày còn lại backup level 1

Sau đó lưu lại bằng phím Esc --> :wq (write and quite)

Kiểm tra lại nếu có 2 dòng như trên là thành công:

(oracle)$ crontab -l

Kiểm tra log lịch chạy bằng crontab sinh ra:

oot@ol7 log]# tail -1000f /var/log/cron

4. Kiểm tra lại bản backup

RMAN > list backup;

RMAN> list backup summary;

RMAN> list backup of datafile 1;

SQL> select command\_id, start\_time, end\_time, status,INPUT\_TYPE, input\_bytes\_display, output\_bytes\_display, time\_taken\_display, round(compression\_ratio,2) RATIO , input\_bytes\_per\_sec\_display, output\_bytes\_per\_sec\_display

from v$rman\_backup\_job\_details

where trunc(end\_time)>=trunc(sysdate-120)

order by end\_time desc;

## Backup nhiều mount point

--binh1: Ten server

--binh: Ten DB

root@binh1 # more /u01/oracle/backup\_db/scripts/backup\_db\_level\_0.rcv

RUN

{

Crosscheck archivelog all;

Delete noprompt expired archivelog all;

--DELETE BACKUP COMPLETED BEFORE 'SYSDATE-4' DEVICE TYPE DISK;

ALLOCATE CHANNEL c1 DEVICE TYPE DISK format '/s01/backup\_db/binh/c1\_binh\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c2 DEVICE TYPE DISK format '/s02/backup\_db/binh/c2\_binh\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c3 DEVICE TYPE DISK format '/s03/backup\_db/binh/c3\_binh\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c4 DEVICE TYPE DISK format '/s04/backup\_db/binh/c4\_binh\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c5 DEVICE TYPE DISK format '/s04/backup\_db/binh/c5\_binh\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c6 DEVICE TYPE DISK format '/s01/backup\_db/binh/c6\_binh\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c7 DEVICE TYPE DISK format '/s02/backup\_db/binh/c7\_binh\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c8 DEVICE TYPE DISK format '/s03/backup\_db/binh/c8\_binh\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c9 DEVICE TYPE DISK format '/s04/backup\_db/binh/c9\_binh\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c10 DEVICE TYPE DISK format '/s04/backup\_db/binh/c10\_binh\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c11 DEVICE TYPE DISK format '/s01/backup\_db/binh/c11\_binh\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c12 DEVICE TYPE DISK format '/s02/backup\_db/binh/c12\_binh\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c13 DEVICE TYPE DISK format '/s03/backup\_db/binh/c13\_binh\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c14 DEVICE TYPE DISK format '/s04/backup\_db/binh/c14\_binh\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c15 DEVICE TYPE DISK format '/s04/backup\_db/binh/c15\_binh\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c16 DEVICE TYPE DISK format '/s01/backup\_db/binh/c16\_binh\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c17 DEVICE TYPE DISK format '/s02/backup\_db/binh/c17\_binh\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c18 DEVICE TYPE DISK format '/s03/backup\_db/binh/c18\_binh\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c19 DEVICE TYPE DISK format '/s01/backup\_db/binh/c19\_binh\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c20 DEVICE TYPE DISK format '/s02/backup\_db/binh/c20\_binh\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c21 DEVICE TYPE DISK format '/s03/backup\_db/binh/c21\_binh\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c22 DEVICE TYPE DISK format '/s04/backup\_db/binh/c22\_binh\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c23 DEVICE TYPE DISK format '/s01/backup\_db/binh/c23\_binh\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c24 DEVICE TYPE DISK format '/s02/backup\_db/binh/c24\_binh\_level0\_ora\_df%T\_s%s\_s%p';

BACKUP AS COMPRESSED BACKUPSET INCREMENTAL LEVEL 0 DATABASE FILESPERSET 8 MAXSETSIZE 40 G TAG LEVEL0;

BACKUP AS COMPRESSED BACKUPSET ARCHIVELOG ALL FILESPERSET 4 TAG ARCH delete input;

#Delete noprompt archivelog until time 'sysdate-1';

CROSSCHECK BACKUP;

DELETE NOPROMPT EXPIRED BACKUP;

DELETE OBSOLETE;

}

# more /u01/oracle/backup\_db/scripts/backup\_db\_level\_1.rcv

RUN

{

SQL "BEGIN binhtv.send\_sms\_binhtv(''3.binh\_9.215 Backup Level 1 Start''); END; ";

Crosscheck archivelog all;

Delete noprompt expired archivelog all;

--DELETE BACKUP COMPLETED BEFORE 'SYSDATE-4' DEVICE TYPE DISK;

ALLOCATE CHANNEL c1 DEVICE TYPE DISK format '/s01/backup\_db/binh/c1\_binh\_level1\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c2 DEVICE TYPE DISK format '/s02/backup\_db/binh/c2\_binh\_level1\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c3 DEVICE TYPE DISK format '/s03/backup\_db/binh/c3\_binh\_level1\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c4 DEVICE TYPE DISK format '/s04/backup\_db/binh/c4\_binh\_level1\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c5 DEVICE TYPE DISK format '/s04/backup\_db/binh/c5\_binh\_level1\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c6 DEVICE TYPE DISK format '/s01/backup\_db/binh/c6\_binh\_level1\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c7 DEVICE TYPE DISK format '/s02/backup\_db/binh/c7\_binh\_level1\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c8 DEVICE TYPE DISK format '/s03/backup\_db/binh/c8\_binh\_level1\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c9 DEVICE TYPE DISK format '/s04/backup\_db/binh/c9\_binh\_level1\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c10 DEVICE TYPE DISK format '/s04/backup\_db/binh/c10\_binh\_level1\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c11 DEVICE TYPE DISK format '/s01/backup\_db/binh/c11\_binh\_level1\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c12 DEVICE TYPE DISK format '/s02/backup\_db/binh/c12\_binh\_level1\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c13 DEVICE TYPE DISK format '/s03/backup\_db/binh/c13\_binh\_level1\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c14 DEVICE TYPE DISK format '/s04/backup\_db/binh/c14\_binh\_level1\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c15 DEVICE TYPE DISK format '/s02/backup\_db/binh/c15\_binh\_level1\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c16 DEVICE TYPE DISK format '/s01/backup\_db/binh/c16\_binh\_level1\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c17 DEVICE TYPE DISK format '/s02/backup\_db/binh/c17\_binh\_level1\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c18 DEVICE TYPE DISK format '/s03/backup\_db/binh/c18\_binh\_level1\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c19 DEVICE TYPE DISK format '/s04/backup\_db/binh/c19\_binh\_level1\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c20 DEVICE TYPE DISK format '/s04/backup\_db/binh/c20\_binh\_level1\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c21 DEVICE TYPE DISK format '/s01/backup\_db/binh/c21\_binh\_level1\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c22 DEVICE TYPE DISK format '/s02/backup\_db/binh/c22\_binh\_level1\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c23 DEVICE TYPE DISK format '/s03/backup\_db/binh/c23\_binh\_level1\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c24 DEVICE TYPE DISK format '/s04/backup\_db/binh/c24\_binh\_level1\_ora\_df%T\_s%s\_s%p';

BACKUP AS COMPRESSED BACKUPSET INCREMENTAL LEVEL 1 DATABASE FILESPERSET 10 MAXSETSIZE 40 G TAG LEVEL1 ;

BACKUP AS COMPRESSED BACKUPSET ARCHIVELOG ALL FILESPERSET 10 TAG ARCH DELETE INPUT;

#Delete noprompt archivelog until time 'sysdate-1';

CROSSCHECK BACKUP;

DELETE NOPROMPT EXPIRED BACKUP;

DELETE OBSOLETE;

--SQL "BEGIN binhtv.send\_sms\_binhtv(''3.binh\_9.215 Backup Level 1 Success''); END; ";

}

## Backup vào phân vùng ASM

# more backup\_db\_level\_0.rcv

RUN

{

#SQL 'ALTER SYSTEM FLUSH SHARED\_POOL';

CROSSCHECK BACKUP;

#DELETE NOPROMPT EXPIRED BACKUP;

#CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT '+reco/coredb/backupset/level0/ora\_df%T\_s%s\_s%p';

# MAXPIECESIZE 8192 M;

DELETE BACKUP COMPLETED BEFORE 'SYSDATE-4' DEVICE TYPE DISK;

BACKUP AS COMPRESSED BACKUPSET INCREMENTAL LEVEL 0 DATABASE FORMAT '+reco/core/backupset/level0/l0\_df\_%T\_s%s\_s%p' FILESPERSET 8 SKIP READONLY SKIP OFFLINE

TAG LEVEL0;

crosscheck archivelog all;

delete NOPROMPT expired archivelog all;

BACKUP AS COMPRESSED BACKUPSET ARCHIVELOG ALL NOT BACKED UP FORMAT '+reco/

coredb/backupset/level0/l0\_arc\_%T\_%s%p' FILESPERSET 10 TAG ARCH;

delete noprompt force archivelog until time 'sysdate-1';

#DELETE BACKUP COMPLETED BEFORE 'SYSDATE-7' DEVICE TYPE DISK;

DELETE OBSOLETE;

}

# more backup\_db\_level\_1.rcv

RUN

{

#SQL 'ALTER SYSTEM FLUSH SHARED\_POOL';

CROSSCHECK BACKUP;

#DELETE NOPROMPT EXPIRED BACKUP;

#CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT '+reco/coredb/backupset/level0/ora\_df%T\_s%s\_s%p';

# MAXPIECESIZE 8192 M;

DELETE BACKUP COMPLETED BEFORE 'SYSDATE-4' DEVICE TYPE DISK;

BACKUP AS COMPRESSED BACKUPSET INCREMENTAL LEVEL 1 DATABASE FORMAT '+reco/coredb/backupset/level1/l1\_df\_%T\_s%s\_s%p' FILESPERSET 8 SKIP READONLY SKIP OFFLINE

TAG LEVEL1;

crosscheck archivelog all;

delete NOPROMPT expired archivelog all;

BACKUP AS COMPRESSED BACKUPSET ARCHIVELOG ALL NOT BACKED UP FORMAT '+reco/

coredb/backupset/level1/l1\_arc\_%T\_%s%p' FILESPERSET 10 TAG ARCH;

delete noprompt force archivelog until time 'sysdate-1';

#DELETE BACKUP COMPLETED BEFORE 'SYSDATE-7' DEVICE TYPE DISK;

DELETE OBSOLETE;

}

## Backup read only tablespace

Read Only, Offline: Da backup DATA2001-2018

-- Dung luong 3.5TB ban backup 96GB

--1.Chạy script backup trên node 2 hoặc node 4

$cd /u01/oracle/backup\_db/scripts

[oracle@ scripts]$ more backup\_db\_ro.rcv

RUN

{

#SQL 'ALTER SYSTEM FLUSH SHARED\_POOL';

#CROSSCHECK BACKUP;

#DELETE NOPROMPT EXPIRED BACKUP;

#CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT '+reco\_coredb/coredb/backupset/level0/ora\_df%T\_s%s\_s%p';

# MAXPIECESIZE 8192 M;

BACKUP AS COMPRESSED BACKUPSET TABLESPACE DATA2001,DATA2003,DATA2004,DATA2005,DATA2006,DATA2007,DATA2008,DATA2009,DATA2010,DATA2011,DATA2012,DATA2013,DATA2014,DATA2015,DATA2016,DATA2017,DATA2018,INDX2003,INDX2004,INDX2005,INDX2006,INDX2007,INDX2008,INDX2009,INDX2010,INDX2011,INDX2012,INDX2013,INDX2014,INDX2015,INDX2016,INDX2017,INDX2018 FORMAT '+reco/coredb/backupset/backup\_readonly/tbs\_readonly\_%T\_%d\_%U' FILESPERSET 10 MAXSETSIZE 40 G TAG TBS\_READONLY;

}

[oracle@ scripts]$ more run\_backup\_db\_ro.sh

target='target /'

rcvcat='nocatalog'

freq=24

time=`date '+%H%M%S'`

cmdfile=/u01/app/backup\_db/scripts/backup\_db\_ro.rcv

#rman $target $rcvcat cmdfile $cmdfile msglog $msglog

#rman target / cmdfile=/u01/app/backup\_db/scripts/end\_backup.rcv

rman $target top$rcvcat cmdfile $cmdfile

exit

--Tu 02h15 20/11/2020 -

$ nohup ./run\_backup\_tbs\_readonly\_offline.sh &

--2. Sau khi backup xong doi ten tu '/+reco\_coredb/coredb/backupset/backup\_readonly thanh /+reco\_coredb/coredb/backupset/backup\_readonly\_20190124:

mv /+reco/coredb/backupset/backup\_readonly thanh /+reco\_coredb/coredb/backupset/backup\_readonly\_20190124

Doi quyen root de tranh bi xoa

# chown -Rf root /+reco/coredb/backupset/backup\_readonly\_20190124

24/01/2019: Da chuyen sang phan vung /+reco\_coredb/coredb/backupset/backup\_readonly

drwxr-xr-x 2 1000 1001 1536 Nov 27 2017 coredb.20171127

--3.Khi can su dung

# mv /+reco/coredb/backupset/backup\_readonly\_20190124 /+reco\_coredb/coredb/backupset/backup\_readonly

# chown -Rf oracle:oinstall /+reco\_coredb/coredb/backupset/backup\_readonly

## Tạo bảng backup full gấp:

Thủ tục backup full Oracle Database

1. Tạo file backup rcv

$ vi /backup01/scripts/backup\_db\_level\_0.rcv

RUN

{

Crosscheck archivelog all;

Delete noprompt expired archivelog all;

#DELETE BACKUP COMPLETED BEFORE 'SYSDATE-4' DEVICE TYPE DISK;

ALLOCATE CHANNEL c1 DEVICE TYPE DISK format '/backup01/full0206/c1\_cl\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c2 DEVICE TYPE DISK format '/backup01/full0206/c2\_cl\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c3 DEVICE TYPE DISK format '/backup01/full0206/c3\_cl\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c4 DEVICE TYPE DISK format '/backup01/full0206/c4\_cl\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c5 DEVICE TYPE DISK format '/backup01/full0206/c5\_cl\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c6 DEVICE TYPE DISK format '/backup01/full0206/c6\_cl\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c7 DEVICE TYPE DISK format '/backup01/full0206/c7\_cl\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c8 DEVICE TYPE DISK format '/backup01/full0206/c8\_cl\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c9 DEVICE TYPE DISK format '/backup01/full0206/c9\_cl\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c10 DEVICE TYPE DISK format '/backup01/full0206/c10\_cl\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c11 DEVICE TYPE DISK format '/backup01/full0206/c11\_cl\_level0\_ora\_df%T\_s%s\_s%p';

ALLOCATE CHANNEL c12 DEVICE TYPE DISK format '/backup01/full0206/c12\_cl\_level0\_ora\_df%T\_s%s\_s%p';

BACKUP AS COMPRESSED BACKUPSET INCREMENTAL LEVEL 0 DATABASE FILESPERSET 4 MAXSETSIZE 40 G TAG LEVEL0;

BACKUP AS COMPRESSED BACKUPSET ARCHIVELOG ALL FILESPERSET 4 TAG ARCH;

delete noprompt archivelog until time 'sysdate-1';

CROSSCHECK BACKUP;

DELETE NOPROMPT EXPIRED BACKUP;

DELETE OBSOLETE;

}

Có thể chạy luôn:

$ rman target / nolog @/backup01/scripts/backup\_db\_level\_0.rcv

Hoặc muốn chạy từ script thì thực hiện các bước bên dưới:

2. Tạo file sh

$ vi /backup01/scripts/run\_backup\_db\_level\_0.sh

target='target /'

rcvcat='nocatalog'

freq=24

#time=`date '+%H%M%S'`

cmdfile=/backup01/scripts/backup\_db\_level\_0.rcv

#msglog=/backup01/scripts/backup\_db\_level\_0\_$time.log

#msglog=/backup01/scripts/backup\_db\_level\_0\_$time.log

#rman $target $rcvcat cmdfile $cmdfile msglog $msglog

rman target / cmdfile=/backup01/scripts/backup\_db\_level\_0.rcv

#rman $target $rcvcat cmdfile $cmdfile

Exit

3. Chạy script sh

$ cd /backup01/scripts/

$ nohup ./run\_backup\_db\_level\_0.sh

4. Theo dõi

$ tail -1000f nohup.out

## Tài liệu tham khảo:

<https://www.tranvanbinh.vn/2021/06/thu-tuc-backup-full-oracle-database.html>

<https://www.tranvanbinh.vn/2018/09/rman-backup-command.html>

<https://www.tranvanbinh.vn/2020/10/tong-hop-cau-lenh-rman-trong-oracle.html>

<https://www.tranvanbinh.vn/2023/06/cac-cau-lenh-kiem-tra-thong-tin-ban.html>

<https://www.tranvanbinh.vn/2021/12/rman-list-commands.html>

<https://www.tranvanbinh.vn/2020/12/tu-ong-backup-oracle-database-hang-ngay.html>

<https://www.tranvanbinh.vn/2022/03/tu-ong-backup-oracle-database-19c-hang.html>

<https://www.tranvanbinh.vn/2020/12/backup-va-recovery-cua-container.html>

<https://www.tranvanbinh.vn/2021/04/huong-dan-su-dung-crontab-linuxunix.html>

# BÀI 15: RECOVERY DATABASE

## Khôi phục lịa dữ liệu bảng purge, truncate

select command\_id, start\_time, end\_time, status,INPUT\_TYPE, input\_bytes\_display, output\_bytes\_display, time\_taken\_display, round(compression\_ratio,2) RATIO , input\_bytes\_per\_sec\_display, output\_bytes\_per\_sec\_display

from v$rman\_backup\_job\_details

where trunc(end\_time)>=trunc(sysdate-120)

order by end\_time desc;

select \* from OAZ14.TABLE1;

drop table OAZ14.TABLE1;

FLASHBACK TABLE OAZ14.TABLE1 TO BEFORE DROP;

select \* from dba\_segments where owner='OAZ14';

select \* from OAZ14."BIN$62UUExFxAcrgU4C2qMB6hA==$0";

select \* from dba\_recyclebin

where owner='OAZ14'

order by droptime desc;

purge table OAZ14.TABLE1;

RECOVER TABLE 'OAZ14'.'TABLE1' OF PLUGGABLE DATABASE pdb1

UNTIL TIME "TO\_DATE('19-10-2022 21:30', 'DD-MM-YYYY HH24:MI')"

AUXILIARY DESTINATION '/u01/aux' ;

## Các trường hợp khôi phục lại dữ liệu

--15.1.KHOI PHUC 1 DATAFILE CUA TABLESPACE USERS BI MAT

--PDB

--/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/o1\_mf\_users\_lb5qd279\_.dbf USERS

select \* from dba\_data\_files order by tablespace\_name;

--USERS

--/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/o1\_mf\_users\_lb5qd279\_.dbf USERS

ls -lt /u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/o1\_mf\_users\_lb5qd279\_.dbf;

-- Doi ten file thanh xxx

rm /u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/o1\_mf\_users\_lb5qd279\_.dbf ;

ls -lt /u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/o1\_mf\_users\_lb5qd279\_.dbf;

create table oaz19.tab19(id number, name varchar2(100)) tablespace users;

grant unlimited tablespace to oaz19;

declare

n number :=0;

begin

for n in 1 .. 100000

loop

insert into oaz19.tab19 values(n, 'Binh ' || n);

commit;

--n := n+1;

end loop;

end;

ORA-01116: error in opening database file 12

ORA-01110: data file 12: '/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/o1\_mf\_users\_lb5qd279\_.dbf'

ORA-27041: unable to open file

Linux-x86\_64 Error: 2: No such file or directory

Additional information: 3

ORA-06512: at line 6

select \* from oaz19.tab19 ;

--Recovery

rman target=sys@pdb1;

RUN {

SQL 'ALTER DATABASE DATAFILE 12 OFFLINE';

RESTORE DATAFILE 12;

RECOVER DATAFILE 12;

SQL 'ALTER DATABASE DATAFILE 12 ONLINE';

}

-- XOA 1 DATAFILE ORCL (CDB)

select \* from dba\_data\_files order by tablespace\_name;

--USERS

--/u01/app/oracle/oradata/ORCL/datafile/o1\_mf\_users\_lb5rrt69\_.dbf 7 USERS

ls -lt /u01/app/oracle/oradata/ORCL/datafile/o1\_mf\_users\_lb5rrt69\_.dbf;

-- Doi ten file thanh xxx

rm -rf /u01/app/oracle/oradata/ORCL/datafile/o1\_mf\_users\_lb5rrt69\_.dbf ;

ls -lt /u01/app/oracle/oradata/ORCL/datafile/o1\_mf\_users\_lb5rrt69\_.dbf;

create user c##oaz19 identified by oracle;

grant connect, resource to c##oaz19 container=all;

grant unlimited tablespace to c##oaz19 container=all;

create table c##oaz19.tab19(id number, name varchar2(100)) tablespace users;

declare

n number :=0;

begin

for n in 1 .. 100000

loop

insert into c##oaz19.tab19 values(n, 'CDB\_Binh ' || n);

commit;

--n := n+1;

end loop;

end;

select \* from c##oaz19.tab19;

ORA-01116: error in opening database file 7

ORA-01110: data file 7: '/u01/app/oracle/oradata/ORCL/datafile/o1\_mf\_users\_lb5rrt69\_.dbf'

ORA-27041: unable to open file

Linux-x86\_64 Error: 2: No such file or directory

Additional information: 3

ORA-06512: at line 6

--Recovery

rman target /;

RMAN > list backup of datafile 7;

RUN {

SQL 'ALTER DATABASE DATAFILE 7 OFFLINE';

RESTORE DATAFILE 7;

RECOVER DATAFILE 7;

SQL 'ALTER DATABASE DATAFILE 7 ONLINE';

}

declare

n number :=0;

begin

for n in 1 .. 1000

loop

insert into c##oaz19.tab19 values(n, 'CDB\_Binh ' || n);

commit;

--n := n+1;

end loop;

end;

select \* from c##oaz19.tab19;

--

--create table oaz19.tab17(id number, name varchar2(100)) tablespace system;

--

--grant unlimited tablespace to oaz19;

--

--

--declare

-- n number :=0;

--begin

-- for n in 1 .. 10000

-- loop

-- insert into test.tab7 values(n, 'Binh ' || n);

-- commit;

-- --n := n+1;

-- end loop;

--end;

--

-- Xoa file va restart DB:

/u01/app/oracle/oradata/ORCL/datafile/o1\_mf\_users\_lb5r05l2\_.dbf 7 USERS

rm -rf /u01/app/oracle/oradata/ORCL/datafile/o1\_mf\_users\_lb5r05l2\_.dbf;

ls -la /u01/app/oracle/oradata/ORCL/datafile/o1\_mf\_users\_lb5r05l2\_.dbf;

declare

n number :=0;

begin

for n in 1 .. 100000

loop

insert into c##oaz19.tab19 values(n, 'CDB\_Binh ' || n);

commit;

--n := n+1;

end loop;

end;

ORA-01565: error in identifying file '/u01/app/oracle/oradata/ORCL/datafile/o1\_mf\_users\_lb5r05l2\_.dbf'

ORA-27037: unable to obtain file status

Linux-x86\_64 Error: 2: No such file or directory

Additional information: 7

ORA-06512: at line 6

SQL> shutdown immediate;

--PDB1(3):ALTER PLUGGABLE DATABASE OPEN

--PDB1(3):Autotune of undo retention is turned on.

--2022-12-16T21:13:23.023153+07:00

--Errors in file /u01/app/oracle/diag/rdbms/orcl/orcl/trace/orcl\_dbw0\_13008.trc:

--ORA-01157: cannot identify/lock data file 48 - see DBWR trace file

--ORA-01110: data file 48: '/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/o1\_mf\_users\_koqrktbs\_.dbf'

--ORA-27037: unable to obtain file status

--Linux-x86\_64 Error: 2: No such file or directory

--Additional information: 7

--2022-12-16T21:13:23.032919+07:00

--Pdb PDB1 hit error 1113 during open read write (1) and will be closed.

--SQL> Shutdown abort;

--Database mounted.

--ORA-01157: cannot identify/lock data file 7 - see DBWR trace file

--ORA-01110: data file 7:

--'/u01/app/oracle/oradata/ORCL/datafile/o1\_mf\_users\_lb5r05l2\_.dbf'

SQL> startup;

SQL> show pdbs;

CON\_ID CON\_NAME OPEN MODE RESTRICTED

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

2 PDB$SEED READ ONLY NO

3 PDB1 MOUNTED --> Binh thuong la READ WRITE

RUN {

SQL 'ALTER DATABASE DATAFILE 7 OFFLINE';

RESTORE DATAFILE 7;

RECOVER DATAFILE 7;

SQL 'ALTER DATABASE DATAFILE 7 ONLINE';

}

====

--SQL> alter session set container=pdb1;

--

--SQL> startup

--ORA-65368: unable to open the pluggable database due to errors during recovery

--ORA-01110: data file 48:

--'/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/o1\_mf\_us

--ers\_koqrktbs\_.dbf'

--ORA-01157: cannot identify/lock data file 48 - see DBWR trace file

--ORA-01110: data file 48:

--'/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/o1\_mf\_us

--ers\_koqrktbs\_.dbf'

--ORA-01113: file 83 needs media recovery

--ORA-01110: data file 83:

--'/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/o1\_mf\_da

--ta\_kr9ng663\_.dbf'

--

--[oracle@linux7 ~]$ rman target /

--

--RMAN> list backup of datafile 48;

-- 48 0 Incr 3925611 16-DEC-22 NO /u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/o1\_mf\_users\_koqrktbs\_.dbf

--

--

--[oracle@linux7 ~]$ ls -la /u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/o1\_mf\_users\_k2ysofg0\_.dbf

--ls: cannot access /u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/o1\_mf\_users\_k2ysofg0\_.dbf: No such file or directory

--

--rman target=sys@pdb1;

--

---- mat khau oracle

--

--RUN {

-- ALTER DATABASE DATAFILE 48,83 OFFLINE;

-- RESTORE DATAFILE 48,83;

-- RECOVER DATAFILE 48,83;

-- ALTER DATABASE DATAFILE 48,83 ONLINE;

--}

--

--RMAN> backup database plus archivelog ;

--

--SQL> show pdbs;

--

-- CON\_ID CON\_NAME OPEN MODE RESTRICTED

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

-- 2 PDB$SEED READ ONLY NO

-- 3 PDB1 MOUNTED

--SQL> alter pluggable database pdb1 open;

--

--Pluggable database altered.

--

--SQL> show pdbs;

--

-- CON\_ID CON\_NAME OPEN MODE RESTRICTED

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

-- 2 PDB$SEED READ ONLY NO

-- 3 PDB1 READ WRITE NO

--

--SQL> alter session set container=pdb1;

--

--Session altered.

--

--SQL> startup;

--ORA-65019: pluggable database PDB1 already open

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 15.2.KHOI PHUC DATAFILE SYSTEM CUA PDB1 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

select \* from dba\_data\_files where tablespace\_name='SYSTEM';

/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/o1\_mf\_system\_l6q0vnm9\_.dbf 9

mv /u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/o1\_mf\_system\_l6q0vnm9\_.dbf /u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/o1\_mf\_system\_l6q0vnm9\_.dbf.xxx;

(hoac echo 1 >> /u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/o1\_mf\_system\_l6q0vnm9\_.dbf)

SQL> show pdbs;

CON\_ID CON\_NAME OPEN MODE RESTRICTED

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

2 PDB$SEED READ ONLY NO

3 PDB1 READ WRITE NO

SQL> alter pluggable database pdb1 close;

Pluggable database altered.

SQL> alter pluggable database pdb1 open;

alter pluggable database pdb1 open

ORA-01110: data file 9:

'/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/o1\_mf\_sy

stem\_l6q0vnm9\_.dbf'

ORA-01157: cannot identify/lock data file 9 - see DBWR trace file

ORA-01110: data file 9:

'/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/o1\_mf\_system\_l6q0vnm9\_.dbf'

ORA-01113: file 110 needs media recovery

ORA-01110: data file 110:

'/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/o1\_mf\_in

dx\_log\_lhpl39f6\_.dbf'

--/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/o1\_mf\_data\_kr9ng663\_.dbf 83 DATA

select \* from dba\_data\_files where file\_id=9;

select \* from v$datafile where file#=9;

$ rman target=sys@pdb1;

-- Go mat khau oracle

RMAN>

RUN {

ALTER DATABASE DATAFILE 9 OFFLINE;

RESTORE DATAFILE 9;

RECOVER DATAFILE 9;

ALTER DATABASE DATAFILE 9 ONLINE;

}

SQL> alter pluggable database pdb1 open;

SQL> show pdbs;

SQL> show pdbs;

CON\_ID CON\_NAME OPEN MODE RESTRICTED

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

2 PDB$SEED READ ONLY NO

3 PDB1 READ WRITE NO

RMAN > backup database plus archivelog ;

--15.3.KHOI PHUC CONTROL FILE TU BACKUP

-- MAT 2/3 CONTROL FILE

--rman> backup database plus archivelog;

--RMAN> backup current controlfile;

/home/oracle/backup/level0.sh

Alert log: Control autobackup written to DISK device handle '/home/oracle/backup/level0/auto\_dbaviet\_ctlc-1611489315-20230925-02'

SQL> show parameter control;

control\_files string /u01/app/oracle/oradata/ORCL/controlfile/o1\_mf\_jngnomtw\_.ctl, /u01/app/oracle/fast\_recovery\_area/ORCL/controlfile/o1\_mf\_jngnomvm\_.ctl, /u01/app/oracle/oradata/ORCL/controlfile/conrol03.ctl

--Control autobackup written to DISK device

--handle '/home/oracle/backup/level1/auto\_dbaviet\_ctlc-1611489315-20230522-01'

rm -rf /u01/app/oracle/oradata/ORCL/controlfile/o1\_mf\_jngnomtw\_.ctl;

rm -rf /u01/app/oracle/fast\_recovery\_area/ORCL/controlfile/o1\_mf\_jngnomvm\_.ctl;

[oracle@linux7 ~]$ ls /u01/app/oracle/oradata/ORCL/controlfile/o1\_mf\_jngnomtw\_.ctl;

[oracle@linux7 ~]$ ls /u01/app/oracle/fast\_recovery\_area/ORCL/controlfile/o1\_mf\_jngnomvm\_.ctl;

SQL> alter system checkpoint;

SQL> alter system switch logfile;

ORA-00202: control file: '/u01/app/oracle/oradata/ORCL/controlfile/o1\_mf\_jngnomtw\_.ctl'

ORA-27041: unable to open file

Linux-x86\_64 Error: 2: No such file or directory

-- Con lai:

/u01/app/oracle/oradata/ORCL/controlfile/conrol03.ctl

--Bo qua: mv /u01/app/oracle/oradata/ORCL/controlfile/control03.ctl /u01/app/oracle/oradata/ORCL/controlfile/control03.ctl.xxx

SQL> shutdown immediate;

ORA-00210: cannot open the specified control file

ORA-00202: control file: '/u01/app/oracle/oradata/ORCL/controlfile/o1\_mf\_jngnomtw\_.ctl'

ORA-27041: unable to open file

Linux-x86\_64 Error: 2: No such file or directory

Additional information: 3

SQL> shutdown abort;

ORACLE instance shut down.

SQL> startup

ORACLE instance started.

Total System Global Area 1.0737E+10 bytes

Fixed Size 12688456 bytes

Variable Size 1744830464 bytes

Database Buffers 8959033344 bytes

Redo Buffers 20865024 bytes

ORA-00205: error in identifying control file, check alert log for more info

ORA-00202: control file: '/u01/app/oracle/fast\_recovery\_area/ORCL/controlfile/o1\_mf\_jngnomvm\_.ctl'

ORA-27037: unable to obtain file status

Linux-x86\_64 Error: 2: No such file or directory

Additional information: 7

ORA-00210: cannot open the specified control file

ORA-00202: control file: '/u01/app/oracle/oradata/ORCL/controlfile/o1\_mf\_jngnomtw\_.ctl'

ORA-27037: unable to obtain file status

Linux-x86\_64 Error: 2: No such file or directory

cp /u01/app/oracle/oradata/ORCL/controlfile/conrol03.ctl /u01/app/oracle/oradata/ORCL/controlfile/o1\_mf\_jngnomtw\_.ctl;

cp /u01/app/oracle/oradata/ORCL/controlfile/conrol03.ctl /u01/app/oracle/fast\_recovery\_area/ORCL/controlfile/o1\_mf\_jngnomvm\_.ctl;

[oracle@linux7 ~]$ ls -la /u01/app/oracle/oradata/ORCL/controlfile/o1\_mf\_jngnomtw\_.ctl;

-rw-r----- 1 oracle oinstall 18759680 Sep 25 22:03 /u01/app/oracle/oradata/ORCL/controlfile/o1\_mf\_jngnomtw\_.ctl

[oracle@linux7 ~]$ ls -la /u01/app/oracle/fast\_recovery\_area/ORCL/controlfile/o1\_mf\_jngnomvm\_.ctl;

-rw-r----- 1 oracle oinstall 18759680 Sep 25 22:03 /u01/app/oracle/fast\_recovery\_area/ORCL/controlfile/o1\_mf\_jngnomvm\_.ctl

SQL> alter database mount;

SQL> alter database open;

-- MAT 3/3 CONTROL FILE

SQL> show parameter control;

control\_files string /u01/app/oracle/oradata/ORCL/controlfile/o1\_mf\_jngnomtw\_.ctl, /u01/app/oracle/fast\_recovery\_area/ORCL/controlfile/o1\_mf\_jngnomvm\_.ctl, /u01/app/oracle/oradata/ORCL/controlfile/conrol03.ctl

piece handle=/home/oracle/backup/level0/auto\_dbaviet\_ctlc-1611489315-20230925-01 comment=N

ONE

piece handle=/home/oracle/backup/level0/auto\_dbaviet\_ctlc-1611489315-20230925-02 comment=NONE

Finished Control File and SPFILE Autobackup at 2023-09-25 21:57:12

rm -rf /u01/app/oracle/oradata/ORCL/controlfile/o1\_mf\_jngnomtw\_.ctl;

rm -rf /u01/app/oracle/fast\_recovery\_area/ORCL/controlfile/o1\_mf\_jngnomvm\_.ctl;

rm -rf /u01/app/oracle/oradata/ORCL/controlfile/conrol03.ctl;

ls -la /u01/app/oracle/oradata/ORCL/controlfile/o1\_mf\_jngnomtw\_.ctl;

ls -la /u01/app/oracle/fast\_recovery\_area/ORCL/controlfile/o1\_mf\_jngnomvm\_.ctl;

ls -la /u01/app/oracle/oradata/ORCL/controlfile/conrol03.ctl;

SQL> alter system checkpoint;

SQL> alter system switch logfile;

SQL> alter system switch logfile;

Errors in file /u01/app/oracle/diag/rdbms/orcl/orcl/trace/orcl\_arc3\_25471.trc:

ORA-00210: cannot open the specified control file

ORA-00202: control file: '/u01/app/oracle/oradata/ORCL/controlfile/o1\_mf\_jngnomtw\_.ctl'

ORA-27041: unable to open file

Linux-x86\_64 Error: 2: No such file or directory

Additional information: 3

2023-05-22T22:22:25.853159+07:00

Errors in file /u01/app/oracle/diag/rdbms/orcl/orcl/trace/orcl\_mz00\_25881.trc:

ORA-00202: control file: '/u01/app/oracle/oradata/ORCL/controlfile/o1\_mf\_jngnomtw\_.ctl'

ORA-27041: unable to open file

Linux-x86\_64 Error: 2: No such file or directory

Additional information: 3

2023-05-22T22:22:25.977593+07:00

Errors in file /u01/app/oracle/diag/rdbms/orcl/orcl/trace/orcl\_mz00\_25881.trc:

ORA-00210: cannot open the specified control file

ORA-00202: control file: '/u01/app/oracle/oradata/ORCL/controlfile/o1\_mf\_jngnomtw\_.ctl'

ORA-27041: unable to open file

Linux-x86\_64 Error: 2: No such file or directory

Additional information: 3

2023-05-22T22:22:29.902945+07:00

SQL> shutdown immediate;

Database closed.

ORA-00210: cannot open the specified control file

ORA-00202: control file:

'/u01/app/oracle/oradata/ORCL/controlfile/o1\_mf\_jngnomtw\_.ctl'

ORA-27041: unable to open file

Linux-x86\_64 Error: 2: No such file or directory

Additional information: 3

SQL> shutdown abort;

ORACLE instance shut down.

SQL> startup

Total System Global Area 1.0737E+10 bytes

Fixed Size 12688456 bytes

Variable Size 1744830464 bytes

Database Buffers 8959033344 bytes

Redo Buffers 20865024 bytes

ORA-00205: error in identifying control file, check alert log for more info

Errors in file /u01/app/oracle/diag/rdbms/orcl/orcl/trace/orcl\_mz00\_26072.trc:

ORA-00202: control file: '/u01/app/oracle/fast\_recovery\_area/ORCL/controlfile/o1\_mf\_jngnomvm\_.ctl'

ORA-27037: unable to obtain file status

Linux-x86\_64 Error: 2: No such file or directory

Additional information: 7

ORA-00210: cannot open the specified control file

ORA-00202: control file: '/u01/app/oracle/oradata/ORCL/controlfile/o1\_mf\_jngnomtw\_.ctl'

ORA-27037: unable to obtain file status

Linux-x86\_64 Error: 2: No such file or directory

Additional information: 7

2023-05-22T22:24:18.984259+07:00

Errors in file /u01/app/oracle/diag/rdbms/orcl/orcl/trace/orcl\_mz00\_26072.trc:

ORA-00202: control file: '/u01/app/oracle/oradata/ORCL/controlfile/conrol03.ctl'

[oracle@linux7 datafile]$ rman target /

--Tim trong ALERT LOG co autobackup gan nhat2023-07-03T22:16:49.465947+07:00

--Control autobackup written to DISK device

--handle '/home/oracle/backup/level1/auto\_dbaviet\_ctlc-1611489315-20230703-02'

--2023-07-03T22:16:50.881356+07:00

-- Tim trong log backup dinh ky:

--piece handle=/home/oracle/backup/level1/auto\_dbaviet\_ctlc-1611489315-20230703-02 comment=NONE

--Finished Control File and SPFILE Autobackup at 2023-07-03 22:16:50

RMAN> restore controlfile from '/home/oracle/backup/level0/auto\_dbaviet\_ctlc-1611489315-20230925-02';

(hoac restore controlfile from autobackup;)

Starting restore at 25-SEP-23

using channel ORA\_DISK\_1

channel ORA\_DISK\_1: restoring control file

channel ORA\_DISK\_1: restore complete, elapsed time: 00:00:01

output file name=/u01/app/oracle/oradata/ORCL/controlfile/o1\_mf\_jngnomtw\_.ctl

output file name=/u01/app/oracle/fast\_recovery\_area/ORCL/controlfile/o1\_mf\_jngnomvm\_.ctl

output file name=/u01/app/oracle/oradata/ORCL/controlfile/conrol03.ctl

Finished restore at 25-SEP-23

RMAN> alter database mount;

--RMAN> restore database;

RMAN> recover database;

SQL> alter database open resetlogs;

SQL> show pdbs;

CON\_ID CON\_NAME OPEN MODE RESTRICTED

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

2 PDB$SEED READ ONLY NO

3 PDB1 READ WRITE NO

select \* from v$archived\_log

--where name like '%o1\_mf\_1\_1\_lk39w1hn\_%'

order by resetlogs\_time;

select \* from v$log;

select \* from v$logfile;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 15.4.MAT REDOLOG \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

-- 15.4.1.MAT 1 MEMBER cua group INACTIVE

--1 1 1 209715200 512 2 NO CURRENT

--2 1 0 209715200 512 2 YES UNUSED

--4 1 0 209715200 512 2 YES UNUSED

--5 1 0 209715200 512 2 YES UNUSED

select \* from v$log;

--1 ONLINE /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_1\_l6q61bd3\_.log

--1 ONLINE /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo01b.log

--2 ONLINE /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_2\_l6q61bdq\_.log

--2 ONLINE /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo02b.log

--4 ONLINE /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo04bb.log

--4 ONLINE /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_4\_l6q61c8h\_.log

--5 ONLINE /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo05bb.log

--5 ONLINE /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_5\_l6q61c73\_.log

select \* from v$logfile order by 1;

alter system switch logfile;

-- Doi ten, 1 member group 2

mv /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_2\_l6q61bdq\_.log /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_2\_l6q61bdq\_.log.xxx;

SQL> alter system switch logfile;

Errors in file /u01/app/oracle/diag/rdbms/orcl/orcl/trace/orcl\_mz00\_15855.trc:

ORA-00313: open failed for members of log group 2 of thread 1

ORA-00312: online log 2 thread 1: '/u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_2\_l6q61bdq\_.log'

ORA-27037: unable to obtain file status

Linux-x86\_64 Error: 2: No such file or directory

select \* from v$logfile;

alter database drop logfile member '/u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_2\_l6q61bdq\_.log';

ALTER DATABASE ADD LOGFILE MEMBER '/u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_2\_l6q61bdq\_new.log' TO GROUP 2;

--Backup controlfile written to trace file /u01/app/oracle/diag/rdbms/orcl/orcl/trace/orcl\_ora\_15709.trc

--15.4.2.MAT TOAN BO GROUP CURRENT

--1 1 13 209715200 512 2 YES ACTIVE

--2 1 14 209715200 512 2 YES ACTIVE

--4 1 15 209715200 512 2 NO CURRENT

--5 1 12 209715200 512 2 YES ACTIVE

select \* from v$log;

--4 ONLINE /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo04bb.log

--4 ONLINE /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_4\_l6q61c8h\_.log

select \* from v$logfile order by 1;

select 'rm -rf ' || member ||';' from v$logfile order by group#;

--Group 4

rm -rf /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo04bb.log;

rm -rf /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_4\_l6q61c8h\_.log;

SQL> alter system switch logfile;

Errors in file /u01/app/oracle/diag/rdbms/orcl/orcl/trace/orcl\_arc2\_14280.trc:

ORA-00313: open failed for members of log group 4 of thread 1

ORA-00312: online log 4 thread 1: '/u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_4\_l6q61c8h\_.log'

ORA-27041: unable to open file

Linux-x86\_64 Error: 2: No such file or directory

Additional information: 3

ORA-00312: online log 4 thread 1: '/u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo04bb.log'

ORA-27041: unable to open file

Linux-x86\_64 Error: 2: No such file or directory

--Fix

cd /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/

cp redo05b.log /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo04bb.log

cd /u01/app/oracle/redolog/ORCL/onlinelog/

cp o1\_mf\_2\_l6q61bdq\_new.log /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_4\_l6q61c8h\_.log

ALTER DATABASE CLEAR UNARCHIVED LOGFILE GROUP 4;

alter system switch logfile;

select \* from v$log;

SQL> alter database open;

Database altered.

SQL> show pdbs;

CON\_ID CON\_NAME OPEN MODE RESTRICTED

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

2 PDB$SEED READ ONLY NO

3 PDB1 READ WRITE NO

--15.4.3.MAT TOAN BO ONLINE REDO LOG GROUP

--1 1 22 209715200 512 2 YES INACTIVE

--2 1 23 209715200 512 2 YES INACTIVE

--4 1 25 209715200 512 2 NO CURRENT

--5 1 24 209715200 512 2 YES INACTIVE

select \* from v$log;

--1 ONLINE /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_1\_l6q61bd3\_.log

--1 ONLINE /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo01b.log

--2 ONLINE /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_2\_l6q61bdq\_new.log

--2 ONLINE /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo02b.log

--4 ONLINE /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo04bb.log

--4 ONLINE /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_4\_lk3c5ph8\_.log

--5 ONLINE /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo05bb.log

--5 ONLINE /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_5\_l6q61c73\_.log

select \* from v$logfile order by 1;

select 'rm -rf ' || member ||';' from v$logfile order by 1;

rm -rf /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo01b.log;

rm -rf /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo02b.log;

rm -rf /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo04bb.log;

rm -rf /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo05bb.log;

rm -rf /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_1\_l6q61bd3\_.log;

rm -rf /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_2\_l6q61bdq\_new.log;

rm -rf /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_4\_lk3c5ph8\_.log;

rm -rf /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_5\_l6q61c73\_.log;

alter system switch logfile;

Instance terminated by PMON, pid = 25999

--shutdown immediate;

--ORA-00313: open failed for members of log group 2 of thread 1

--ORA-00312: online log 2 thread 1:

--'/u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/o1\_mf\_2\_jngnotfp\_1.log'

--ORA-27037: unable to obtain file status

--Linux-x86\_64 Error: 2: No such file or directory

--Additional information: 7

--ORA-00312: online log 2 thread 1:

--'/u01/app/oracle/oradata/ORCL/onlinelog/o1\_mf\_2\_jngnoq3d\_.log'

--ORA-27037: unable to obtain file status

--Linux-x86\_64 Error: 2: No such file or directory

--Additional information: 7

-- Tao lai cac group

cp /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo05b.log /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo01b.log;

cp /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo05b.log /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo02b.log;

cp /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo05b.log /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo04bb.log;

cp /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo05b.log /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo05bb.log;

cp /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo05b.log /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_1\_l6q61bd3\_.log;

cp /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo05b.log /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_2\_l6q61bdq\_new.log;

cp /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo05b.log /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_4\_lk3c5ph8\_.log;

cp /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo05b.log /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_5\_l6q61c73\_.log;

SQL>

SQL> ALTER DATABASE CLEAR UNARCHIVED LOGFILE GROUP 1;

Database altered.

SQL> ALTER DATABASE CLEAR UNARCHIVED LOGFILE GROUP 2;

ALTER DATABASE CLEAR UNARCHIVED LOGFILE GROUP 2

\*

ERROR at line 1:

ORA-01624: log 2 needed for crash recovery of instance orcl (thread 1)

ORA-00312: online log 2 thread 1: '/u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo02b.log'

ORA-00312: online log 2 thread 1: '/u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_2\_l6q61bdq\_new.log'

--Xoa 2 log file

rm /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo02b.log;

rm /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_2\_l6q61bdq\_new.log;

ls /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo02b.log;

ls /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_2\_l6q61bdq\_new.log;

--Tao lai

cp /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo05b.log /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo02b.log;

cp /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo05b.log /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_2\_l6q61bdq\_new.log;

SQL> ALTER DATABASE CLEAR UNARCHIVED LOGFILE GROUP 2;

Database altered.

--SQL> ALTER DATABASE CLEAR UNARCHIVED LOGFILE GROUP 5;

--

--Database altered.

SQL> shutdown immediate;

ORA-01109: database not open

Database dismounted.

ORACLE instance shut down.

SQL> startup

ORACLE instance started.

Total System Global Area 1.0737E+10 bytes

Fixed Size 12688456 bytes

Variable Size 1744830464 bytes

Database Buffers 8959033344 bytes

Redo Buffers 20865024 bytes

Database mounted.

ORA-00341: log 2 of thread 1, wrong log # 5 in header

ORA-00312: online log 2 thread 1:

'/u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/o1\_mf\_2\_jngnotfp\_1.log'

ORA-00341: log 2 of thread 1, wrong log # 5 in header

ORA-00312: online log 2 thread 1: '/u01/app/oracle/oradata/ORCL/onlinelog/o1\_mf\_2\_jngnoq3d\_.log'

SQL> select group# from v$logfile where member in ('/u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_2\_l6q61bdq\_new.log','/u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo02b.log');

GROUP#

----------

2

2

--Tao lai redo log group

--alter database drop logfile group 1;

--alter database drop logfile group 5;

SQL> alter database drop logfile group 2;

alter database drop logfile group 2

\*

ERROR at line 1:

ORA-01624: log 2 needed for crash recovery of instance orcl (thread 1)

ORA-00312: online log 2 thread 1: '/u01/app/oracle/oradata/ORCL/onlinelog/o1\_mf\_2\_jngnoq3d\_.log'

ORA-00312: online log 2 thread 1:

'/u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/o1\_mf\_2\_jngnotfp\_1.log'

SQL> alter database drop logfile group 4;

alter database drop logfile group 4

\*

ERROR at line 1:

ORA-01623: log 4 is current log for instance orcl (thread 1) - cannot drop

ORA-00312: online log 4 thread 1: '/u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo04bb.log'

ORA-00312: online log 4 thread 1: '/u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_4\_l6q3gdso\_.log'

alter database add logfile thread 1 group 1 size 200m;

alter database add logfile thread 1 group 3 size 200m;

ORA-00341: log 4 of thread 1, wrong log # 5 in header

ORA-00312: online log 4 thread 1: '/u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo04bb.log'

ALTER DATABASE CLEAR LOGFILE GROUP 2;

ALTER DATABASE CLEAR LOGFILE GROUP 4;

--2

cp /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_1\_l6q4s736\_.log /u01/app/oracle/oradata/ORCL/onlinelog/o1\_mf\_2\_jngnoq3d\_.log;

cp /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_1\_l6q4s736\_.log /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/o1\_mf\_2\_jngnotfp\_1.log;

--4

cp /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_1\_l6q4s736\_.log /u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/redo05bb.log;

cp /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_1\_l6q4s736\_.log /u01/app/oracle/redolog/ORCL/onlinelog/o1\_mf\_5\_l6q3x7r5\_.log;

ALTER DATABASE CLEAR LOGFILE GROUP 2;

ALTER DATABASE CLEAR LOGFILE GROUP 4;

ALTER DATABASE CLEAR UNARCHIVED LOGFILE GROUP 2;

-- Recovery database;

--+Case 1:

SQL> startup nomount;

RMAN> restore controlfile from '/home/oracle/backup/level0/auto\_dbaviet\_ctlc-1611489315-20230925-04';

RMAN> alter database mount;

RMAN> restore database;

RMAN> recover database noredo;

RMAN> alter database open resetlogs;

--+Case 2: Toi da recovery

SQL> startup nomount;

RMAN> restore controlfile from '/home/oracle/backup/level0/auto\_dbaviet\_ctlc-1611489315-20230925-04';

RMAN> alter database mount;

RMAN> restore database;

RMAN> recover database;

archived log thread=1 sequence=15

RMAN-00571: ===========================================================

RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS ===============

RMAN-00571: ===========================================================

RMAN-03002: failure of recover command at 09/25/2023 23:29:03

RMAN-06054: media recovery requesting unknown archived log for thread 1 with sequence 15 and starting SCN of 7182816

--Check sequence co the recovery

SQL> select sequence#,first\_change#, to\_char(first\_time,'HH24:MI:SS') from v$log;

SEQUENCE# FIRST\_CHANGE# TO\_CHAR(

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

13 7182804 22:47:01

16 7182881 22:48:56 --> lay sequence 16

15 7182816 22:47:05

14 7182813 22:47:03

run {

set until sequence=16;

restore database;

recover database;

}

archived log thread=1 sequence=15

RMAN-00571: ===========================================================

RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS ===============

RMAN-00571: ===========================================================

RMAN-03002: failure of recover command at 09/25/2023 23:29:03

RMAN-06054: media recovery requesting unknown archived log for thread 1 with sequence 15 and starting SCN of 7182816

run {

set until sequence=15;

restore database;

recover database;

}

RMAN> alter database open resetlogs;

## Tạo lại controlfile từ trace file

Backup controlfile written to trace file /u01/app/oracle/diag/rdbms/orcl/orcl/trace/orcl\_ora\_4799.trc

/u01/app/oracle/oradata/ORCL/controlfile/o1\_mf\_jngnomtw\_.ctl, /u01/app/oracle/fast\_recovery\_area/ORCL/controlfile/o1\_mf\_jngnomvm\_.ctl

rm /u01/app/oracle/oradata/ORCL/controlfile/o1\_mf\_jngnomtw\_.ctl ;

rm /u01/app/oracle/fast\_recovery\_area/ORCL/controlfile/o1\_mf\_jngnomvm\_.ctl

[oracle@linux7 oaz]$ more /u01/app/oracle/diag/rdbms/orcl/orcl/trace/orcl\_ora\_4799.trc

Trace file /u01/app/oracle/diag/rdbms/orcl/orcl/trace/orcl\_ora\_4799.trc

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

Build label: RDBMS\_19.3.0.0.0DBRU\_LINUX.X64\_190417

ORACLE\_HOME: /u01/app/oracle/product/19.0.0/dbhome\_1

System name: Linux

Node name: linux7

Release: 4.14.35-1902.300.11.el7uek.x86\_64

Version: #2 SMP Tue Mar 17 17:11:47 PDT 2020

Machine: x86\_64

Instance name: orcl

Redo thread mounted by this instance: 1

Oracle process number: 48

Unix process pid: 4799, image: oracle@linux7 (TNS V1-V3)

\*\*\* 2022-07-11T20:27:03.357710+07:00 (CDB$ROOT(1))

\*\*\* SESSION ID:(30.12819) 2022-07-11T20:27:03.357777+07:00

\*\*\* CLIENT ID:() 2022-07-11T20:27:03.357795+07:00

\*\*\* SERVICE NAME:(SYS$USERS) 2022-07-11T20:27:03.357878+07:00

\*\*\* MODULE NAME:(rman@linux7 (TNS V1-V3)) 2022-07-11T20:27:03.357898+07:00

\*\*\* ACTION NAME:(0000001 FINISHED70) 2022-07-11T20:27:03.357914+07:00

\*\*\* CLIENT DRIVER:() 2022-07-11T20:27:03.357928+07:00

\*\*\* CONTAINER ID:(1) 2022-07-11T20:27:03.357941+07:00

krsv\_proc\_add: Request to add process to V$MANAGED\_STANDBY [krse.c:2735]

krsv\_proc\_rem: Request to remove process from V$MANAGED\_STANDBY [krse.c:2794]

\*\*\* 2022-07-11T20:27:08.077718+07:00 (CDB$ROOT(1))

\*\*\* ACTION NAME:(0000003 FINISHED66) 2022-07-11T20:27:08.078496+07:00

RECID #17 Recno 17

Internal Status: 6

External Status: 2

Command id: 2022-07-11T20:26:41

Operation: backup

Start and End time 07/11/22 20:27:04 07/11/22 20:27:08

\*\*\* 2022-07-11T20:27:31.787796+07:00 (CDB$ROOT(1))

\*\*\* ACTION NAME:(0000007 FINISHED66) 2022-07-11T20:27:31.793068+07:00

RECID #18 Recno 18

Internal Status: 6

External Status: 2

Command id: 2022-07-11T20:26:41

Operation: backup

Start and End time 07/11/22 20:27:08 07/11/22 20:27:31

krsv\_proc\_add: Request to add process to V$MANAGED\_STANDBY [krse.c:2735]

Log read is SYNCHRONOUS though disk\_asynch\_io is enabled!

krsv\_proc\_rem: Request to remove process from V$MANAGED\_STANDBY [krse.c:2794]

\*\*\* 2022-07-11T20:27:34.163256+07:00 (CDB$ROOT(1))

\*\*\* ACTION NAME:(0000009 FINISHED66) 2022-07-11T20:27:34.163308+07:00

RECID #19 Recno 19

Internal Status: 6

External Status: 2

Command id: 2022-07-11T20:26:41

Operation: backup

Start and End time 07/11/22 20:27:31 07/11/22 20:27:34

\*\*\* 2022-07-11T20:27:35.301331+07:00 (CDB$ROOT(1))

RECID #20 Recno 20

Internal Status: 6

External Status: 2

Command id: 2022-07-11T20:26:41

Operation: Control File and SPFILE Autoback

Start and End time 07/11/22 20:27:34 07/11/22 20:27:35

\*\*\* ACTION NAME:(0000023 FINISHED129) 2022-07-11T20:28:06.535721+07:00

\*\*\* 2022-07-11T20:28:06.535687+07:00 (CDB$ROOT(1))

-- The following are current System-scope REDO Log Archival related

-- parameters and can be included in the database initialization file.

--

-- LOG\_ARCHIVE\_DEST=''

-- LOG\_ARCHIVE\_DUPLEX\_DEST=''

--

-- LOG\_ARCHIVE\_FORMAT=%t\_%s\_%r.arch

--

-- DB\_UNIQUE\_NAME="orcl"

--

-- LOG\_ARCHIVE\_CONFIG='SEND, RECEIVE, NODG\_CONFIG'

-- LOG\_ARCHIVE\_MAX\_PROCESSES=4

-- STANDBY\_FILE\_MANAGEMENT=MANUAL

-- FAL\_CLIENT=''

-- FAL\_SERVER=''

--

-- LOG\_ARCHIVE\_DEST\_1='LOCATION=USE\_DB\_RECOVERY\_FILE\_DEST'

-- LOG\_ARCHIVE\_DEST\_1='MANDATORY REOPEN=300 NODELAY'

-- LOG\_ARCHIVE\_DEST\_1='ARCH NOAFFIRM NOVERIFY SYNC'

-- LOG\_ARCHIVE\_DEST\_1='NOREGISTER'

-- LOG\_ARCHIVE\_DEST\_1='NOALTERNATE'

-- LOG\_ARCHIVE\_DEST\_1='NODEPENDENCY'

-- LOG\_ARCHIVE\_DEST\_1='NOMAX\_FAILURE NOQUOTA\_SIZE NOQUOTA\_USED NODB\_UNIQUE\_NAME'

-- LOG\_ARCHIVE\_DEST\_1='VALID\_FOR=(PRIMARY\_ROLE,ONLINE\_LOGFILES)'

-- LOG\_ARCHIVE\_DEST\_STATE\_1=ENABLE

--

-- Below are two sets of SQL statements, each of which creates a new

-- control file and uses it to open the database. The first set opens

-- the database with the NORESETLOGS option and should be used only if

-- the current versions of all online logs are available. The second

-- set opens the database with the RESETLOGS option and should be used

-- if online logs are unavailable.

-- The appropriate set of statements can be copied from the trace into

-- a script file, edited as necessary, and executed when there is a

-- need to re-create the control file.

--

-- Set #1. NORESETLOGS case

--

-- The following commands will create a new control file and use it

-- to open the database.

-- Data used by Recovery Manager will be lost.

-- Additional logs may be required for media recovery of offline

-- Use this only if the current versions of all online logs are

-- available.

-- After mounting the created controlfile, the following SQL

-- statement will place the database in the appropriate

-- protection mode:

-- ALTER DATABASE SET STANDBY DATABASE TO MAXIMIZE PERFORMANCE

STARTUP NOMOUNT

CREATE CONTROLFILE REUSE DATABASE "ORCL" NORESETLOGS ARCHIVELOG

MAXLOGFILES 16

MAXLOGMEMBERS 3

MAXDATAFILES 1024

MAXINSTANCES 8

MAXLOGHISTORY 292

LOGFILE

GROUP 1 (

'/u01/app/oracle/oradata/ORCL/onlinelog/o1\_mf\_1\_jngnoq26\_.log',

'/u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/o1\_mf\_1\_jngnotg1\_.log'

) SIZE 200M BLOCKSIZE 512,

GROUP 2 (

'/u01/app/oracle/oradata/ORCL/onlinelog/o1\_mf\_2\_jngnoq3d\_.log',

'/u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/o1\_mf\_2\_jngnotfp\_.log'

) SIZE 200M BLOCKSIZE 512,

GROUP 3 (

'/u01/app/oracle/oradata/ORCL/onlinelog/o1\_mf\_3\_jngnoq50\_.log',

'/u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/o1\_mf\_3\_jngnotg1\_.log'

) SIZE 200M BLOCKSIZE 512

-- STANDBY LOGFILE

DATAFILE

'/u01/app/oracle/oradata/ORCL/datafile/o1\_mf\_system\_jngnlyw0\_.dbf',

'/u01/app/oracle/oradata/ORCL/datafile/o1\_mf\_sysaux\_jngnmr4x\_.dbf',

'/u01/app/oracle/oradata/ORCL/datafile/o1\_mf\_undotbs1\_jngnn7b8\_.dbf',

'/u01/app/oracle/oradata/ORCL/datafile/o1\_mf\_system\_jngomqr0\_.dbf',

'/u01/app/oracle/oradata/ORCL/datafile/o1\_mf\_sysaux\_jngomqr2\_.dbf',

'/u01/app/oracle/oradata/ORCL/datafile/o1\_mf\_users\_jngnn8g7\_.dbf',

'/u01/app/oracle/oradata/ORCL/datafile/o1\_mf\_undotbs1\_jngomqr3\_.dbf',

'/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/o1\_mf\_system\_jngplfv4\_.dbf',

'/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/o1\_mf\_sysaux\_jngplfv9\_.dbf',

'/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/o1\_mf\_undotbs1\_jngplfv9\_.dbf',

'/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/o1\_mf\_users\_jngpltl1\_.dbf'

CHARACTER SET AL32UTF8

;

-- Commands to re-create incarnation table

-- Below log names MUST be changed to existing filenames on

-- disk. Any one log file from each branch can be used to

-- re-create incarnation records.

-- ALTER DATABASE REGISTER LOGFILE '/u01/app/oracle/fast\_recovery\_area/ORCL/archivelog/2022\_07\_11/o1\_mf\_1\_1\_%u\_.arc';

-- ALTER DATABASE REGISTER LOGFILE '/u01/app/oracle/fast\_recovery\_area/ORCL/archivelog/2022\_07\_11/o1\_mf\_1\_1\_%u\_.arc';

-- Recovery is required if any of the datafiles are restored backups,

-- or if the last shutdown was not normal or immediate.

RECOVER DATABASE

-- All logs need archiving and a log switch is needed.

ALTER SYSTEM ARCHIVE LOG ALL;

-- Database can now be opened normally.

ALTER DATABASE OPEN;

-- Open all the PDBs.

ALTER PLUGGABLE DATABASE ALL OPEN;

-- Commands to add tempfiles to temporary tablespaces.

-- Online tempfiles have complete space information.

-- Other tempfiles may require adjustment.

ALTER TABLESPACE TEMP ADD TEMPFILE '/u01/app/oracle/oradata/ORCL/datafile/o1\_mf\_temp\_jngnp437\_.tmp'

SIZE 81788928 REUSE AUTOEXTEND ON NEXT 655360 MAXSIZE 32767M;

ALTER TABLESPACE TEMP ADD TEMPFILE '/u01/app/oracle/oradata/ORCL/datafile/o2\_mf\_temp\_jngnp437\_.tmp'

SIZE 58720256 REUSE AUTOEXTEND ON NEXT 1048576 MAXSIZE 32767M;

ALTER SESSION SET CONTAINER = "PDB$SEED";

ALTER TABLESPACE TEMP ADD TEMPFILE '/u01/app/oracle/oradata/ORCL/datafile/temp012021-09-19\_22-26-02-682-PM.dbf'

SIZE 37748736 REUSE AUTOEXTEND ON NEXT 655360 MAXSIZE 32767M;

ALTER SESSION SET CONTAINER = "PDB1";

ALTER TABLESPACE TEMP ADD TEMPFILE '/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/o1\_mf\_temp\_jngplfvb\_.dbf'

SIZE 37748736 REUSE AUTOEXTEND ON NEXT 655360 MAXSIZE 32767M;

ALTER SESSION SET CONTAINER = "CDB$ROOT";

-- End of tempfile additions.

--

-- Set #2. RESETLOGS case

--

-- The following commands will create a new control file and use it

-- to open the database.

-- Data used by Recovery Manager will be lost.

-- The contents of online logs will be lost and all backups will

-- be invalidated. Use this only if online logs are damaged.

-- After mounting the created controlfile, the following SQL

-- statement will place the database in the appropriate

-- protection mode:

-- ALTER DATABASE SET STANDBY DATABASE TO MAXIMIZE PERFORMANCE

STARTUP NOMOUNT

CREATE CONTROLFILE REUSE DATABASE "ORCL" RESETLOGS ARCHIVELOG

MAXLOGFILES 16

MAXLOGMEMBERS 3

MAXDATAFILES 1024

MAXINSTANCES 8

MAXLOGHISTORY 292

LOGFILE

GROUP 1 (

'/u01/app/oracle/oradata/ORCL/onlinelog/o1\_mf\_1\_jngnoq26\_.log',

'/u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/o1\_mf\_1\_jngnotg1\_.log'

) SIZE 200M BLOCKSIZE 512,

GROUP 2 (

'/u01/app/oracle/oradata/ORCL/onlinelog/o1\_mf\_2\_jngnoq3d\_.log',

'/u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/o1\_mf\_2\_jngnotfp\_.log'

) SIZE 200M BLOCKSIZE 512,

GROUP 3 (

'/u01/app/oracle/oradata/ORCL/onlinelog/o1\_mf\_3\_jngnoq50\_.log',

'/u01/app/oracle/fast\_recovery\_area/ORCL/onlinelog/o1\_mf\_3\_jngnotg1\_.log'

) SIZE 200M BLOCKSIZE 512

-- STANDBY LOGFILE

DATAFILE

'/u01/app/oracle/oradata/ORCL/datafile/o1\_mf\_system\_jngnlyw0\_.dbf',

'/u01/app/oracle/oradata/ORCL/datafile/o1\_mf\_sysaux\_jngnmr4x\_.dbf',

'/u01/app/oracle/oradata/ORCL/datafile/o1\_mf\_undotbs1\_jngnn7b8\_.dbf',

'/u01/app/oracle/oradata/ORCL/datafile/o1\_mf\_system\_jngomqr0\_.dbf',

'/u01/app/oracle/oradata/ORCL/datafile/o1\_mf\_sysaux\_jngomqr2\_.dbf',

'/u01/app/oracle/oradata/ORCL/datafile/o1\_mf\_users\_jngnn8g7\_.dbf',

'/u01/app/oracle/oradata/ORCL/datafile/o1\_mf\_undotbs1\_jngomqr3\_.dbf',

'/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/o1\_mf\_system\_jngplfv4\_.dbf',

'/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/o1\_mf\_sysaux\_jngplfv9\_.dbf',

'/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/o1\_mf\_undotbs1\_jngplfv9\_.dbf',

'/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/o1\_mf\_users\_jngpltl1\_.dbf'

CHARACTER SET AL32UTF8

;

-- Commands to re-create incarnation table

-- Below log names MUST be changed to existing filenames on

-- disk. Any one log file from each branch can be used to

-- re-create incarnation records.

-- ALTER DATABASE REGISTER LOGFILE '/u01/app/oracle/fast\_recovery\_area/ORCL/archivelog/2022\_07\_11/o1\_mf\_1\_1\_%u\_.arc';

-- ALTER DATABASE REGISTER LOGFILE '/u01/app/oracle/fast\_recovery\_area/ORCL/archivelog/2022\_07\_11/o1\_mf\_1\_1\_%u\_.arc';

-- Recovery is required if any of the datafiles are restored backups,

-- or if the last shutdown was not normal or immediate.

RECOVER DATABASE USING BACKUP CONTROLFILE

-- Database can now be opened zeroing the online logs.

ALTER DATABASE OPEN RESETLOGS;

-- Open all the PDBs.

ALTER PLUGGABLE DATABASE ALL OPEN;

-- Commands to add tempfiles to temporary tablespaces.

-- Online tempfiles have complete space information.

-- Other tempfiles may require adjustment.

ALTER TABLESPACE TEMP ADD TEMPFILE '/u01/app/oracle/oradata/ORCL/datafile/o1\_mf\_temp\_jngnp437\_.tmp'

SIZE 81788928 REUSE AUTOEXTEND ON NEXT 655360 MAXSIZE 32767M;

ALTER TABLESPACE TEMP ADD TEMPFILE '/u01/app/oracle/oradata/ORCL/datafile/o2\_mf\_temp\_jngnp437\_.tmp'

SIZE 58720256 REUSE AUTOEXTEND ON NEXT 1048576 MAXSIZE 32767M;

ALTER SESSION SET CONTAINER = "PDB$SEED";

ALTER TABLESPACE TEMP ADD TEMPFILE '/u01/app/oracle/oradata/ORCL/datafile/temp012021-09-19\_22-26-02-682-PM.dbf'

SIZE 37748736 REUSE AUTOEXTEND ON NEXT 655360 MAXSIZE 32767M;

ALTER SESSION SET CONTAINER = "PDB1";

ALTER TABLESPACE TEMP ADD TEMPFILE '/u01/app/oracle/oradata/ORCL/CC5C01A97A924B63E05380BDA8C055CE/datafile/o1\_mf\_temp\_jngplfvb\_.dbf'

SIZE 37748736 REUSE AUTOEXTEND ON NEXT 655360 MAXSIZE 32767M;

ALTER SESSION SET CONTAINER = "CDB$ROOT";

-- End of tempfile additions.

--

\*\*\* 2022-07-11T20:30:34.518943+07:00 (CDB$ROOT(1))

RECID #16 Recno 16

Internal Status: 34

External Status: 2

Command id: 2022-07-11T20:26:41

Operation: RMAN

Start and End time 07/11/22 20:26:43 07/11/22 20:30:34

KJHA:2phase 001 prepare ctx action:1 env usecnt:1 ctx usecnt:1

KJHA:2phase 215 usecnt:1 tha stop prepare

2022-07-11 20:30:34.534 : [GIPCXCPT][GIPCXCPT] gipcInternalConnectSync: failed sync request, addr 0x188cd4c0 [0000000000000014] { gipcAddress : name 'ipc://scls\_opct\_linux7', objFlags 0x0,

addrFlags 0x4 }, ret gipcretConnectionRefused (29)

2022-07-11 20:30:34.537 : [GIPCXCPT][GIPCXCPT] gipcInternalConnectSync: failed sync request, addr 0x188cdd90 [000000000000001c] { gipcAddress : name 'ipc://scls\_opct\_linux7', objFlags 0x0,

addrFlags 0x4 }, ret gipcretConnectionRefused (29)

2022-07-11 20:30:34.537 : [GIPCXCPT][GIPCXCPT] gipcInternalConnectSync: failed sync request, addr 0x188cddd0 [0000000000000024] { gipcAddress : name 'ipc://scls\_opct\_linux7', objFlags 0x0,

addrFlags 0x4 }, ret gipcretConnectionRefused (29)

[ default][ default]u\_set\_comp\_error: comptype '103' : error '29'

2022-07-11 20:30:34.539 : [GIPCXCPT][GIPCXCPT] gipcInternalConnectSync: failed sync request, addr 0x188d6e70 [0000000000000035] { gipcAddress : name 'clsc://(ADDRESS=(PROTOCOL=ipc)(KEY=CRSD

\_UI\_SOCKET)(GIPCID=00000000-00000000-0))', objFlags 0x0, addrFlags 0x4 }, ret gipcretConnectionRefused (29)

2022-07-11 20:30:34.539 : [GIPCXCPT][GIPCXCPT] gipcConnectSyncF [clscrsconGipcConnect : clscrscon.c : 686]: EXCEPTION[ ret gipcretConnectionRefused (29) ] failed sync connect endp 0x188d7e

e0 [0000000000000032] { gipcEndpoint : localAddr 'clsc://(ADDRESS=(PROTOCOL=ipc)(KEY=)(GIPCID=00000000-00000000-0))', remoteAddr 'clsc://(ADDRESS=(PROTOCOL=ipc)(KEY=CRSD\_UI\_SOCKET)(GIPCID=0

0000000-00000000-0))', numPend 0, numReady 0, numDone 0, numDead 0, numTransfer 0, objFlags 0x0, pidPeer 0, readyRef (nil), ready 0, wobj 0x188cdf70, sendp 0x188d7430 status 13flags 0xa1080

71a, flags-2 0x0, usrFlags 0x0 }, addr 0x188d6e70 [0000000000000035] { gipcAddress : name 'clsc://(ADDRESS=(PROTOCOL=ipc)(KEY=CRSD\_UI\_SOCKET)(GIPCID=00000000-00000000-0))', objFlags 0x0, ad

drFlags 0x4 }, flags 0x0

2022-07-11 20:30:34.540 : [GIPCXCPT][GIPCXCPT] gipcInternalConnectSync: failed sync request, addr 0x188ce340 [000000000000003a] { gipcAddress : name 'ipc://scls\_opct\_linux7', objFlags 0x0,

addrFlags 0x4 }, ret gipcretConnectionRefused (29)

2022-07-11 20:30:34.540 : [GIPCXCPT][GIPCXCPT] gipcInternalConnectSync: failed sync request, addr 0x188cc8a0 [0000000000000042] { gipcAddress : name 'ipc://scls\_opct\_linux7', objFlags 0x0,

addrFlags 0x4 }, ret gipcretConnectionRefused (29)

[ default][ default]u\_set\_comp\_error: comptype '103' : error '29'

2022-07-11 20:30:34.542 : [GIPCXCPT][GIPCXCPT] gipcInternalConnectSync: failed sync request, addr 0x188cd340 [0000000000000053] { gipcAddress : name 'clsc://(ADDRESS=(PROTOCOL=ipc)(KEY=CRSD

\_UI\_SOCKET)(GIPCID=00000000-00000000-0))', objFlags 0x0, addrFlags 0x4 }, ret gipcretConnectionRefused (29)

2022-07-11 20:30:34.542 : [GIPCXCPT][GIPCXCPT] gipcConnectSyncF [clscrsconGipcConnect : clscrscon.c : 686]: EXCEPTION[ ret gipcretConnectionRefused (29) ] failed sync connect endp 0x188cf3

90 [0000000000000050] { gipcEndpoint : localAddr 'clsc://(ADDRESS=(PROTOCOL=ipc)(KEY=)(GIPCID=00000000-00000000-0))', remoteAddr 'clsc://(ADDRESS=(PROTOCOL=ipc)(KEY=CRSD\_UI\_SOCKET)(GIPCID=0

0000000-00000000-0))', numPend 0, numReady 0, numDone 0, numDead 0, numTransfer 0, objFlags 0x0, pidPeer 0, readyRef (nil), ready 0, wobj 0x188cba50, sendp 0x188d6520 status 13flags 0xa1080

71a, flags-2 0x0, usrFlags 0x0 }, addr 0x188cd340 [0000000000000053] { gipcAddress : name 'clsc://(ADDRESS=(PROTOCOL=ipc)(KEY=CRSD\_UI\_SOCKET)(GIPCID=00000000-00000000-0))', objFlags 0x0, ad

drFlags 0x4 }, flags 0x0

KJHA: kjha\_doDBAction2 400 clsrapi\_stop\_db prepare ret:184

KJHA: kjha\_doDBAction2 990 clscrs\_stat:184

KJHA:2phase 660 clscrs\_stat statret:184 usecnt:1 env id: ctx id:

\*\*\* 2022-07-11T20:30:35.492296+07:00 (CDB$ROOT(1))

Stopping background process SMCO

\*\*\* 2022-07-11T20:30:36.624308+07:00 (CDB$ROOT(1))

kwsbgshms: Shutdown all masters complete

\*\*\* 2022-07-11T20:30:36.624570+07:00 (CDB$ROOT(1))

Stopping background process MMNL

\*\*\* 2022-07-11T20:30:37.661270+07:00 (CDB$ROOT(1))

Stopping background process MMON

\*\*\* 2022-07-11T20:30:39.762718+07:00 (PDB1(3))

JIT: pid 4799 requesting stop

\*\*\* 2022-07-11T20:30:40.117514+07:00 (PDB$SEED(2))

JIT: pid 4799 requesting stop

\*\*\* 2022-07-11T20:30:40.198203+07:00 (CDB$ROOT(1))

ksukia: Starting kill, flags = 1

ksukia: Attempt 1 to kill process oracle@linux7 (TNS V1-V3), OS id=5088

ksukia: killed 1 out of 1 processes.

\*\*\* 2022-07-11T20:30:41.245096+07:00 (CDB$ROOT(1))

ksukia: Starting kill, flags = 1

ksukia: killed 0 out of 0 processes.

\*\*\* 2022-07-11T20:30:42.269410+07:00 (CDB$ROOT(1))

KJHA:2phase 003 finish ctx opi\_error:0 action:1 env usecnt:1 id: ctx usecnt:1 id: envp:0

KJHA:2phase 215 usecnt:2 tha stop finish

\*\*\* 2022-07-11T20:30:42.707355+07:00 (CDB$ROOT(1))

kwsbgshms: Shutdown all masters complete

Shutting down AQ Process Coordinator

\*\*\* 2022-07-11T20:30:42.707674+07:00 (CDB$ROOT(1))

Stopping background process AQPC

\*\*\* 2022-07-11T20:30:50.898427+07:00 (CDB$ROOT(1))

kwsbgshms: Shutdown all masters complete

Shutting down AQ Process Coordinator

\*\*\* 2022-07-11T20:30:58.024112+07:00 (CDB$ROOT(1))

\*\*\* 2022-07-11 20:30:58.024098 [krsv.c:2729]

krso\_proc\_stop: Clearing start ARCH error state

DDE rules only execution for: ORA 202

----- START Event Driven Actions Dump ----

---- END Event Driven Actions Dump ----

----- START DDE Actions Dump -----

Executing SYNC actions

----- START DDE Action: 'DB\_STRUCTURE\_INTEGRITY\_CHECK' (Async) -----

Successfully dispatched

----- END DDE Action: 'DB\_STRUCTURE\_INTEGRITY\_CHECK' (SUCCESS, 0 csec) -----

Executing ASYNC actions

----- END DDE Actions Dump (total 0 csec) -----

\*\*\* 2022-07-11T20:31:45.414926+07:00

DDE: Problem Key 'ORA 202' was flood controlled (0x1) (no incident)

ORA-00202: control file: '/u01/app/oracle/oradata/ORCL/controlfile/o1\_mf\_jngnomtw\_.ctl'

ORA-27041: unable to open file

Linux-x86\_64 Error: 2: No such file or directory

Additional information: 3

DDE: Problem Key 'ORA 202' was flood controlled (0x1) (no incident)

ORA-00202: control file: '/u01/app/oracle/oradata/ORCL/controlfile/o1\_mf\_jngnomtw\_.ctl'

ORA-27041: unable to open file

Linux-x86\_64 Error: 2: No such file or directory

Additional information: 3

DDE: Problem Key 'ORA 202' was flood controlled (0x1) (no incident)

ORA-00202: control file: '/u01/app/oracle/oradata/ORCL/controlfile/o1\_mf\_jngnomtw\_.ctl'

ORA-27041: unable to open file

Linux-x86\_64 Error: 2: No such file or directory

Additional information: 3

DDE: Problem Key 'ORA 202' was flood controlled (0x1) (no incident)

ORA-00202: control file: '/u01/app/oracle/oradata/ORCL/controlfile/o1\_mf\_jngnomtw\_.ctl'

ORA-27041: unable to open file

Linux-x86\_64 Error: 2: No such file or directory

ORA-00202: control file: '/u01/app/oracle/oradata/ORCL/controlfile/o1\_mf\_jngnomtw\_.ctl'

ORA-27041: unable to open file

Linux-x86\_64 Error: 2: No such file or directory

Additional information: 3

DDE: Problem Key 'ORA 202' was flood controlled (0x1) (no incident)

ORA-00202: control file: '/u01/app/oracle/oradata/ORCL/controlfile/o1\_mf\_jngnomtw\_.ctl'

ORA-27041: unable to open file

Linux-x86\_64 Error: 2: No such file or directory

Additional information: 3

DDE: Problem Key 'ORA 202' was flood controlled (0x1) (no incident)

ORA-00202: control file: '/u01/app/oracle/oradata/ORCL/controlfile/o1\_mf\_jngnomtw\_.ctl'

ORA-27041: unable to open file

Linux-x86\_64 Error: 2: No such file or directory

Additional information: 3

DDE: Problem Key 'ORA 202' was flood controlled (0x1) (no incident)

ORA-00202: control file: '/u01/app/oracle/oradata/ORCL/controlfile/o1\_mf\_jngnomtw\_.ctl'

ORA-27041: unable to open file

Linux-x86\_64 Error: 2: No such file or directory

Additional information: 3

DDE: Problem Key 'ORA 202' was flood controlled (0x1) (no incident)

ORA-00202: control file: '/u01/app/oracle/oradata/ORCL/controlfile/o1\_mf\_jngnomtw\_.ctl'

ORA-27041: unable to open file

Linux-x86\_64 Error: 2: No such file or directory

Additional information: 3

\*\*\* 2022-07-11T20:31:45.460220+07:00 (CDB$ROOT(1))

USER (ospid: 4799): terminating the instance

## Tài liệu tham khảo:

<https://www.tranvanbinh.vn/2020/12/cac-cau-lenh-recovery-du-lieu.html>

<https://www.tranvanbinh.vn/2021/01/thu-tuc-khoi-phuc-co-so-du-lieu-oracle.html>

<https://www.tranvanbinh.vn/2020/08/khoi-phuc-co-so-du-lieu-oracle-11g-sang.html>

<https://www.tranvanbinh.vn/2018/08/muc-ich-can-restore-lai-du-lieu-commit.html>

<https://www.tranvanbinh.vn/2022/06/tao-lai-control-file-trong-oracle.html>

<https://www.tranvanbinh.vn/2020/07/flashback-lai-package-spec-hoac-package.html>

<https://www.tranvanbinh.vn/2022/02/khoi-phuc-du-lieu-khi-drop-table.html>

<https://www.tranvanbinh.vn/2021/04/huong-dan-su-dung-crontab-linuxunix.html>

<https://www.tranvanbinh.vn/2018/09/chuyen-du-lieu-tu-tablespace-khac-sang.html>

# BÀI 16: CHUYỂN DỮ LIỆU

--0. Check

--OS:

ps -ef |grep expdp

ps -ef |grep impdp

--DB

-- job data pump: SYS SYS\_EXPORT\_FULL\_01 EXPORT FULL EXECUTING 8 1 10

select \* from DBA\_DATAPUMP\_JOBS;

-- session

select \* from dba\_datapump\_sessions;

-- longops

col table\_name format a30

select substr(sql\_text, instr(sql\_text,'"')+1,

instr(sql\_text,'"', 1, 2)-instr(sql\_text,'"')-1)

table\_name,

rows\_processed,

round((sysdate

- to\_date(first\_load\_time,'yyyy-mm-dd hh24:mi:ss'))

\*24\*60, 1) minutes,

trunc(rows\_processed /

((sysdate-to\_date(first\_load\_time,'yyyy-mm-dd hh24:mi:ss'))

\*24\*60)) rows\_per\_min

from

v$sqlarea

where

--upper(sql\_text) like 'INSERT % INTO "%' and

command\_type = 2

and

open\_versions > 0;

select

sid,

serial#

from

v$session s,

dba\_datapump\_sessions d

where

s.saddr = d.saddr;

select

sid,

serial#,

sofar,

totalwork

from v$session\_longops;

--1. Tao thu muc

--Tren OS tao:

$ mkdir /home/oracle/oaz

--PDB

sqlplus / as sysdba

alter session set container=pdb1;

select \* from dba\_directories;

cd

desc dba\_directories;

col DIRECTORY\_NAME format a30;

col directory\_path format a70;

set linesize 200;

select DIRECTORY\_NAME, directory\_path from dba\_directories;

-- drop directory oaz;

create directory oaz as '/home/oracle/oaz';

select DIRECTORY\_NAME, directory\_path from dba\_directories where directory\_name='OAZ';

--2.Export

-- 2.1.Export 1 schema (tab18) cua pdb1

alter session set container=pdb1;

create directory oaz as '/home/oracle/oaz';

create user oaz19 identified by oracle;

grant connect, resource, dba to oaz19;

create table oaz19.tab\_objects as select \* from dba\_objects ;

select \* from oaz19.tab\_objects;

--expdp \"sys/oracle@service as sysdba\"

expdp "' sys/oracle@pdb1 as sysdba'" DIRECTORY=oaz DUMPFILE=oaz19%U.dmp logfile=oaz19.log SCHEMAS=oaz19 COMPRESSION=ALL CONTENT=ALL PARALLEL=4;

-- 2.2.Export full DB

sqlplus / as sysdba

create directory oaz as '/home/oracle/oaz';

expdp "' / as sysdba'" DIRECTORY=oaz DUMPFILE=orcl%U.dmp logfile=orcl.log COMPRESSION=ALL CONTENT=ALL FULL=y PARALLEL=8;

-- Go mat khau: oracle

--1.8GB

select sum(bytes)/1024/1024/1024 "GB" from dba\_data\_files;

. . exported "oaz19"."TAB\_OBJECTS" 6.679 KB 9 rows

--3.Xoa bang tab\_objects di

select \* from oaz19.tab\_objects;

drop table oaz19.tab\_objects;

--Check tu tablespace: oaz19 BIN$/7+dIk6yGvXgU4C2qMBFow==$0 TABLE 12582912 65536 1048576 27 2147483645

--OAZ19 BIN$BliaFOlBUzTgY4C2qMDthQ==$0 TAB\_OBJECTS DROP TABLE USERS 2023-09-27:20:56:44 2023-09-27:20:59:37 7364590 YES YES 80995 80995 80995 1536

select \* from dba\_recyclebin;

select \* from oaz19.tab\_objects;

select \* from oaz19."BIN$BliaFOlBUzTgY4C2qMDthQ==$0";

flashback table oaz19.tab\_objects to before drop;

select \* from oaz19.tab\_objects;

drop table oaz19.tab\_objects purge;

--Kiem tra tablespace: ko co

select \* from dba\_recyclebin;

--[Error] Execution (3: 1): ORA-38305: object not in RECYCLE BIN

flashback table oaz19.tab\_objects to before drop;

--4.Import lai

--impdp "' / as sysdba'" DIRECTORY=binhdir DUMPFILE=orcl%U.dmp TABLES=C##BINHTV.TAB\_OBJECTS;

impdp oaz19/oracle@pdb1 DIRECTORY=oaz DUMPFILE=oaz19%U.dmp logfile=impdp.tab\_objects.log TABLES=oaz19.tab\_objects parallel=4;

impdp oaz19/oracle@pdb1 DIRECTORY=oaz DUMPFILE=oaz19%U.dmp logfile=impdp.tab\_objects.log TABLES=oaz19.tab\_objects REMAP\_TABLE=oaz19.tab\_objects:tab\_objects\_new parallel=4;

impdp "' sys/oracle@pdb1 as sysdba '" directory=oaz dumpfile=oaz19%u.dmp logfile=impdp.tabl\_objects.log tables=oaz19.tab\_objects parallel=4;

impdp "' sys/oracle@pdb1 as sysdba '" directory=oaz dumpfile=oaz19%u.dmp logfile=impdp.tabl\_objects.log tables=oaz19.tab\_objects REMAP\_TABLE=oaz19.tab\_objects:tab\_objects\_new parallel=4;

select \* from oaz19.tab\_objects\_new;

--select \* from oaz19.tab\_objects;

-- Test them

expdp "'/ as sysdba'" ESTIMATE\_ONLY=y full=y;

expdp "'/ as sysdba'" DIRECTORY=oaz ESTIMATE\_ONLY=y COMPRESSION=ALL

expdp tab18/oracle@pdb1 DIRECTORY=oaz ESTIMATE\_ONLY=y COMPRESSION=ALL;

## Tài liệu tham khảo:

<https://www.tranvanbinh.vn/2020/11/cac-lenh-hay-su-dung-khi-chuyen-du-lieu.html>

<https://www.tranvanbinh.vn/2020/01/su-dung-sql-loader-trong-oracle-e-load.html>

<https://www.tranvanbinh.vn/2020/02/tutorial-dba-co-ban-chuong-16-nap-va-to.html>

# BÀI 17: BẢO TRÌ CỚ SỞ DỮ LIỆU

## Tài liệu tham khảo:

<https://www.tranvanbinh.vn/2022/06/cach-lay-awr-report-trong-oracle.html>

<https://www.tranvanbinh.vn/2022/10/active-session-history-ash-trong-oracle.html>

<https://www.tranvanbinh.vn/2022/06/cach-lay-awr-report-trong-oracle.html>

<https://www.tranvanbinh.vn/2022/09/cach-oc-phan-tich-bao-cao-awr-trong.html>

<https://www.tranvanbinh.vn/p/tong-hop-bai-viet-ve-toi-uu-oracle.html>

# BÀI 18: QUẢN LÝ HIỆU NĂNG

## Tài liệu tham khảo:

<https://www.tranvanbinh.vn/2018/05/cau-lenh-monitor-csdl-oracle.html>

<https://www.tranvanbinh.vn/2022/09/tong-hop-cac-cau-lenh-check-sql-chiem.html>

<https://www.tranvanbinh.vn/2022/03/top-15-cau-lenh-e-montor-oracle-database.html>

<https://www.tranvanbinh.vn/2020/01/van-hanh-co-so-du-lieu-oracle-3huong.html>

<https://www.tranvanbinh.vn/2021/12/checklist-kiem-tra-am-bao-oracle.html>

<https://www.tranvanbinh.vn/2021/09/ung-dung-tu-nhien-chay-cham-check-nhu.html>

<https://www.tranvanbinh.vn/2022/09/vvip5-1-cau-lenh-kem-chiem-15tb-99-temp.html>

<https://www.tranvanbinh.vn/p/tong-hop-bai-viet-ve-toi-uu-oracle.html>

<https://www.tranvanbinh.vn/2020/08/gather-analyze-co-so-du-lieu-oracle.html>

<https://www.tranvanbinh.vn/2022/09/su-dung-cong-cu-sql-tunning-advisor-e.html>

<https://www.tranvanbinh.vn/2020/01/huong-dan-su-dung-cong-cu-oracle-sql.html>

<https://www.tranvanbinh.vn/2019/11/quan-ly-du-lieu-lob-rat-phuc-tap-nhung.html>

<https://www.tranvanbinh.vn/2022/09/cach-oc-phan-tich-bao-cao-awr-trong.html>

<https://www.tranvanbinh.vn/2022/06/cach-lay-awr-report-trong-oracle.html>

<https://www.tranvanbinh.vn/2022/10/active-session-history-ash-trong-oracle.html>

<https://www.tranvanbinh.vn/2022/10/automatic-database-diagnostic-monitor.html>

# BÀI 19: TỐI ƯU SQL

## Statistic

--[VIP] Gather, Analyze Cơ sở dữ liệu Oracle từ A-Z

--I. Tầm quan trọng của Optimizer Statistics

--Nếu thông tin gather không đầy đủ thì Plan câu lệnh SQL sai --> Hiệu năng DB kém

--Sau khi có thông tin gather thì cơ bản Oracle sẽ xác định Plan chuẩn --> Tăng hiệu năng của cơ sở dữ liệu

--Các thông tin sau khi gather:

--• Cỡ của bảng và index trong

--• Số row

--• Cỡ trung bình của row và các chain

--• Số row đã bị xóa trong lá của index

--Khi dữ liệu được insert, delete hay chỉnh sửa thì thông tin thực tế đã thay đổi so với lần gather gần nhất.

--

--Để không bị ghi đè thông tin statistic thì lock statistics của object lại.

--

-- II. Lưu ý khi gather

-- Check stale

select \* from dba\_tab\_statistics where owner='OAZ17' --and table\_name='TAB1'

and stale\_stats='YES';

----Estimate\_Percent

----AUTO\_SAMPLE\_SIZE

----DBMS\_STATS.AUTO\_SAMPLE\_SIZE

----10: Thay đổi 10% mới gather

----method\_opt

----FOR ALL COLUMNS SIZE AUTO: Tất cả các cột

----FOR ALL COLUMNS SIZE AUTO FOR COLUMNS SIZE 1 IMAGE\_COLUM: Bỏ qua cột IMAGE\_COLUM

----Degree

----4

----INCREMENTAL

----= true (nếu false mà bảng lớn sẽ tốn tài nguyên

----PUBLISH

----= true (if partitioned table)

----GRANULARITY

----AUTO

----PARTITION

----granularity=>'APPROX\_GLOBAL AND PARTITION'

----+gather 1 part nó nội suy cả bảng

----+nhưng không hay lắm nếu minh đang ngồi trên DB Core

--

--Cascade

--FALSE --> Khong gather index

BEGIN

SYS.DBMS\_STATS.GATHER\_TABLE\_STATS (

OwnName => 'TEST\_OWNER'

,TabName => 'SUBS'

,Estimate\_Percent => AUTO\_SAMPLE\_SIZE

,Method\_Opt => 'FOR ALL COLUMNS SIZE 1'

,Degree => 4

,Cascade => FALSE --> Khong gather index

,No\_Invalidate => FALSE);

END;

/

BEGIN

SYS.DBMS\_STATS.GATHER\_TABLE\_STATS (

OwnName => 'TEST\_OWNER'

,TabName => 'SUBS'

,Estimate\_Percent => 10

,Method\_Opt => 'FOR ALL COLUMNS SIZE 1'

,Degree => 4

,Cascade => TRUE --> Treo do gather cả index

,No\_Invalidate => FALSE);

END;

/

--III. Check last gathered

· DBA\_TABLES

· DBA\_TAB\_STATISTICS

· DBA\_TAB\_PARTITIONS

· DBA\_TAB\_SUB\_PARTITIONS

· DBA\_TAB\_COLUMNS

· DBA\_TAB\_COL\_STATISTICS

· DBA\_PART\_COL\_STATISTICS

· DBA\_SUBPART\_COL\_STATISTICS

· DBA\_INDEXES

· DBA\_IND\_STATISTICS

· DBA\_IND\_PARTITIONS

· DBA\_IND\_SUBPARTIONS

· DBA\_TAB\_HISTOGRAMS

· DBA\_PART\_HISTOGRAMS

· DBA\_SUBPART\_HISTOGRAMS

--a. Khi nào cần gather lại:

select owner, table\_name ,partition\_name, subpartition\_name, object\_type, num\_rows, sample\_size, last\_analyzed, global\_stats, stale\_stats

from dba\_tab\_statistics

where owner = 'SCOTT'

and table\_name = 'DEPT'

and (stale\_stats='YES' or stale\_stats is null) ;

--Note: Nếu stale\_stats=YES (thông tin cũ) hoặc trống (chưa có thông tin): Cần gather lại (còn No thì không cần)

--b. Table

--· Bảng partition: (trong TOAD dùng F4 --> partition để check)

select table\_name, partition\_name, last\_analyzed from DBA\_TAB\_PARTITIONS where table\_owner='OAZ17' and table\_name like upper('TABLE1')

--and trunc(last\_analyzed) > sysdate-7

order by last\_analyzed desc;

select owner, table\_name, partition\_name,last\_analyzed from DBA\_TAB\_STATISTICS where table\_owner='OAZ17' and table\_name like upper('TABLE1')

and trunc(last\_analyzed) > sysdate-7

order by last\_analyzed desc;

--· Bảng ko partition:

select owner, table\_name,last\_analyzed from dba\_tables where owner='OAZ17'

--and table\_name like upper('TEST\_TAB1')

--and trunc(last\_analyzed) > sysdate-7

order by last\_analyzed desc;

select owner, table\_name, partition\_name,last\_analyzed from DBA\_TAB\_STATISTICS where table\_owner='OAZ17' and table\_name like upper('TABLE1')

and trunc(last\_analyzed) > sysdate-7

order by last\_analyzed desc;

--c. Index

--· Index partition

select index\_owner, index\_name, partition\_name, last\_analyzed from dba\_ind\_partitions

where index\_owner=‘TEST\_OWNER’

and trunc(last\_analyzed) > sysdate-7

order by last\_analyzed desc;

· Index ko partition

select owner,table\_name,index\_name,num\_rows,last\_analyzed from dba\_indexes

where owner=‘TEST\_OWNER’

and trunc(last\_analyzed) > sysdate-7

order by last\_analyzed desc;

--II. Gather

--Các tham số:

--INCREMENTAL = true

--PUBLISH = true (if partitioned table)

--ESTIMATE\_PERCENT=AUTO\_SAMPLE\_SIZE

--GRANULARITY=AUTO

--1. Gather

--a. Database

-- Recommend: estimate\_percent => DBMS\_STATS.AUTO\_SAMPLE\_SIZE, method\_opt => 'FOR ALL COLUMNS SIZE AUTO'

EXEC DBMS\_STATS.gather\_database\_stats(estimate\_percent => DBMS\_STATS.AUTO\_SAMPLE\_SIZE,Method\_Opt => 'FOR ALL COLUMNS SIZE AUTO',degree=>8, cascade => TRUE);

BEGIN

SYS.DBMS\_STATS.gather\_database\_stats (

Estimate\_Percent => DBMS\_STATS.AUTO\_SAMPLE\_SIZE

,Method\_Opt => 'FOR ALL COLUMNS SIZE AUTO'

,Degree => 16

,Cascade => FALSE

,No\_Invalidate => FALSE);

END;

/

-- Thủ tục gather cả Db

CREATE OR REPLACE PACKAGE BODY TEST\_OWNER.pck\_gather\_table\_stats

IS

p\_error VARCHAR2 (1000);

p\_gather\_date\_par DATE := SYSDATE;

p\_gather\_date DATE := SYSDATE;

PROCEDURE exec\_gather\_tables

IS

BEGIN

-- Bảng unpartitioned dữ

IF (TO\_NUMBER(TO\_CHAR (SYSDATE, 'dd')) IN (1,5,11,15,21,25) and (to\_number(to\_char(sysdate,'hh24')) < 7 or to\_number(to\_char(sysdate,'hh24')) > 22))

THEN

pck\_gather\_table\_stats.gather\_unpartitioned\_tables;

END IF;

pck\_gather\_table\_stats.gather\_partitioned\_tables();

END;

PROCEDURE gather\_partitioned\_tables

IS

CURSOR c\_partitioned\_tables

IS

SELECT table\_owner, table\_name, partition\_name, last\_analyzed,

'begin

dbms\_stats.gather\_table\_stats

(ownname=>''' || TABLE\_OWNER || ''',

tabname=>''' || table\_name || ''',

partname=>''' || partition\_name || ''',

estimate\_percent => DBMS\_STATS.AUTO\_SAMPLE\_SIZE,

method\_opt => ''FOR ALL COLUMNS SIZE AUTO'',

cascade=>true,

degree=>10);

end;

'

script

FROM all\_tab\_partitions

WHERE

table\_owner='TEST\_OWNER'

and table\_name not like '%XXX%' and table\_name not like '%TMP%' and table\_name not like '%TEMP%' and table\_name not like '%TEST%' and table\_name not like '%$%'

AND (last\_analyzed is null or num\_rows is null)

and ((length(partition\_name) = 12 AND to\_date(substr(partition\_name,5,8),'YYYYMMDD')<trunc(sysdate)

AND to\_date(substr(partition\_name,5,8),'YYYYMMDD')>trunc(sysdate)-60

)

or (length(partition\_name) =10 AND to\_date(substr(partition\_name,5,6),'YYYYMM')<trunc(sysdate)

AND to\_date(substr(partition\_name,5,6),'YYYYMM')>=add\_months(trunc(sysdate,'month'),-2)

)

)

and partition\_name like '%2021%'

ORDER BY table\_name, partition\_name;

v\_table\_name VARCHAR2 (100);

v\_par\_name VARCHAR2 (100);

BEGIN

FOR v\_partitioned\_tables IN c\_partitioned\_tables

LOOP

IF (TO\_NUMBER (TO\_CHAR (SYSDATE, 'dd')) NOT IN(0)) and (to\_number(to\_char(sysdate,'hh24')) < 7 or to\_number(to\_char(sysdate,'hh24')) > 22)

THEN

v\_table\_name := v\_partitioned\_tables.table\_name;

v\_par\_name := v\_partitioned\_tables.partition\_name;

EXECUTE IMMEDIATE v\_partitioned\_tables.script;

-- pr\_insert\_log\_gather(v\_partitioned\_tables.table\_owner, v\_partitioned\_tables.table\_name, v\_partitioned\_tables.partition\_name,v\_partitioned\_tables.script);

end if;

END LOOP;

EXCEPTION

WHEN OTHERS

THEN

null;

END;

-- Gather stats for unpartitioned tables

-- Input:

-- p\_Owner: Owner of tables

-- p\_Ignored\_Tables\_List: List of Tables that will not be gathered

-- Output

-- p\_Gathered\_Tables: Tables already gathered

-- p\_Error: Errors descriptions(if any), NULL value means no error occurred

PROCEDURE gather\_unpartitioned\_tables

IS

CURSOR c\_tables

IS

SELECT owner, table\_name, last\_analyzed,

'begin

dbms\_stats.gather\_table\_stats

(ownname=>''' || OWNER || ''',

tabname=>''' || table\_name || ''',

estimate\_percent => DBMS\_STATS.AUTO\_SAMPLE\_SIZE,

method\_opt => ''FOR ALL COLUMNS SIZE AUTO'',

cascade=>true,

degree=>10);

end;

'

script

FROM dba\_tables

WHERE partitioned = 'NO'

AND tablespace\_name IS NOT NULL

AND table\_name NOT LIKE '%$%'

AND table\_name not like '%XX%'

and table\_name not like '%BK%'

and owner in ('TEST\_OWNER')

--AND NVL (last\_analyzed, SYSDATE - 15) < p\_gather\_date

AND (last\_analyzed is null or num\_rows is null)

ORDER BY last\_analyzed;

v\_table\_name VARCHAR2 (100);

BEGIN

FOR v\_tables IN c\_tables

LOOP

v\_table\_name := v\_tables.table\_name;

EXECUTE IMMEDIATE v\_tables.script;

-- pr\_insert\_log\_gather(v\_tables.owner, v\_tables.table\_name, null,v\_tables.script);

END LOOP;

EXCEPTION

WHEN OTHERS

THEN

null;

END;

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

END;

/

b. Schema

EXEC DBMS\_STATS.gather\_schema\_stats('TEST\_OWNER', estimate\_percent => 10,degree=>8, cascade => TRUE);

-- All partition của SCHEMA

-- Dùng incremetal=true để không quét cả bảng

SELECT table\_owner,table\_name, partition\_name, last\_analyzed,

'exec dbms\_stats.gather\_table\_stats(''' || table\_owner||''',''' || table\_name ||''',partname=>'''

|| PARTITION\_NAME || ''',granularity=>''partition'',cascade=> TRUE,force=>TRUE,estimate\_percent=>DBMS\_STATS.AUTO\_SAMPLE\_SIZE,'

|| 'method\_opt=>''FOR ALL COLUMNS SIZE AUTO'',degree => 8);' script

FROM dba\_tab\_partitions

WHERE table\_owner in ('USER1')

and ( (partition\_name LIKE

'DATA'

|| TO\_CHAR (trunc(sysdate) + 30, 'YYYYMM')

|| '%')

OR (partition\_name LIKE

'DATA'

|| TO\_CHAR (trunc(sysdate) + 60, 'YYYYMM')

|| '%'))

and ((length(partition\_name) = 12 AND to\_date(substr(partition\_name,5,8),'YYYYMMDD')<= trunc(sysdate)

)

or (length(partition\_name) =10 AND to\_date(substr(partition\_name,5,6),'YYYYMM')<=trunc(sysdate)

)

)

--AND (last\_analyzed is null or nvl(last\_analyzed,sysdate- 15) < trunc(sysdate))

AND (last\_analyzed is null or num\_rows is null)

ORDER BY table\_owner, table\_name, partition\_name;

-- Bảng non-partition

declare

--p\_gather\_date DATE := sysdate;

CURSOR c\_tables

IS

SELECT table\_name, last\_analyzed,

'begin

dbms\_stats.gather\_table\_stats

(ownname => '''

|| 'TEST\_OWNER'

|| ''',

tabname => '''

|| table\_name

|| ''',

cascade => true,

estimate\_percent => 10,

degree => 10);

end;

'

script

FROM dba\_tables

WHERE partitioned = 'NO' and

owner='TEST\_OWNER'

AND tablespace\_name IS NOT NULL

AND table\_name NOT LIKE '%$%'

--AND NVL (last\_analyzed, SYSDATE - 15) < trunc(sysdate)

AND (last\_analyzed is null or num\_rows is null)

ORDER BY last\_analyzed;

v\_table\_name VARCHAR2 (100);

BEGIN

--send\_sms('TEST\_OWNER Started gathering unpartitioned\_tables at ' || to\_char(sysdate,'dd/mm/yyy hh24:mm:ss'));

FOR v\_tables IN c\_tables

LOOP

v\_table\_name := v\_tables.table\_name;

EXECUTE IMMEDIATE v\_tables.script;

dbms\_output.put\_line(v\_tables.script);

END LOOP;

EXCEPTION

WHEN OTHERS

THEN

null;

END;

c. Table

Cả table (non-partition || partition): Bang partition nen theo từng partition, không nguy cơ sẽ treo

execute dbms\_stats.gather\_table\_stats(ownname => 'SCOTT', tabname =>

'DEPT', estimate\_percent => DBMS\_STATS.AUTO\_SAMPLE\_SIZE,

method\_opt => 'FOR ALL COLUMNS SIZE AUTO',cascade => true,degree => 10);

exec dbms\_stats.gather\_table\_stats (ownname => 'TEST\_OWNER',tabname => 'STOCK\_ISDN',cascade => true,estimate\_percent => 10,degree => 10);

BEGIN

SYS.DBMS\_STATS.GATHER\_TABLE\_STATS (

OwnName => 'TEST\_OWNER

,TabName => ‘TAB1'

,Estimate\_Percent => 10

,Method\_Opt => 'FOR ALL COLUMNS SIZE 1'

,Degree => 4

,Cascade => TRUE

,No\_Invalidate => FALSE);

END;

BEGIN

SYS.DBMS\_STATS.GATHER\_TABLE\_STATS (

OwnName => 'TEST\_OWNER

,TabName => ‘TAB1'

,Estimate\_Percent => 10

,Method\_Opt => 'DBMS\_STATS.AUTO\_SAMPLE\_SIZE'

,Degree => 4

,Cascade => TRUE

,No\_Invalidate => FALSE);

END;

-- Script All Partition 1 bang partition

SELECT table\_name, partition\_name, last\_analyzed,

'exec dbms\_stats.gather\_table\_stats(ownname =>'''||table\_owner||''',tabname =>'''||table\_name||''',partname'||'=>'''

|| PARTITION\_NAME

|| ''',granularity=>''partition'',cascade=> TRUE,force=>TRUE,estimate\_percent=>10,'

|| 'method\_opt=>''FOR ALL COLUMNS SIZE AUTO'',degree => 8);'

script

FROM dba\_tab\_partitions

WHERE table\_name in ('TAB1,'TAB2')

and table\_owner in ('TEST\_OWNER')

AND (last\_analyzed is null or num\_rows is null)

and partition\_name like '%2018%'

--AND and ((length(partition\_name) = 12 AND to\_date(substr(partition\_name,5,8),'YYYYMMDD')<= trunc(sysdate)

)

or (length(partition\_name) =10 AND to\_date(substr(partition\_name,5,6),'YYYYMM')<=trunc(sysdate)

)

)

ORDER BY table\_name, partition\_name;

Exec sys.send\_sms\_binhtv

Thủ tục

declare

--p\_gather\_date DATE := sysdate;

CURSOR c\_partitions

IS

SELECT table\_name, partition\_name, last\_analyzed,

'exec dbms\_stats.gather\_table\_stats(ownname =>'''||table\_owner||''',tabname =>'''||table\_name||''',partname'||'=>'''

|| PARTITION\_NAME

|| ''',granularity=>''partition'',cascade=> TRUE,force=>TRUE,estimate\_percent=>10,'

|| 'method\_opt=>''FOR ALL COLUMNS SIZE AUTO'',degree => 8);'

script

FROM all\_tab\_partitions

WHERE table\_name NOT LIKE '%$%'

and table\_owner in (‘TEST\_OWNER’)

AND (last\_analyzed is null or num\_rows is null)

--AND NVL (last\_analyzed, SYSDATE - 15) < trunc(sysdate)

and ((length(partition\_name) = 12 AND to\_date(substr(partition\_name,5,8),'YYYYMMDD')<= trunc(sysdate)

)

or (length(partition\_name) =10 AND to\_date(substr(partition\_name,5,6),'YYYYMM')<=trunc(sysdate)

)

)

ORDER BY last\_analyzed;

v\_table\_name VARCHAR2 (100);

BEGIN

--send\_sms('TEST\_OWNER Started gathering unpartitioned\_tables at ' || to\_char(sysdate,'dd/mm/yyy hh24:mm:ss'));

FOR v\_partitions IN c\_partitions

LOOP

v\_partition\_name := v\_partitions.table\_name;

EXECUTE IMMEDIATE v\_partitions.script;

dbms\_output.put\_line(v\_partitions.script);

END LOOP;

EXCEPTION

WHEN OTHERS

THEN

null;

END;

Script gather bảng non-partition STALE

SELECT /\* GATHER TABLE NON-PARTITION STALE \*/ table\_name, partition\_name, last\_analyzed,

'begin

dbms\_stats.gather\_table\_stats

(ownname=>''' || OWNER || ''',

tabname=>''' || table\_name || ''',

estimate\_percent => DBMS\_STATS.AUTO\_SAMPLE\_SIZE,

method\_opt => ''FOR ALL COLUMNS SIZE AUTO'',

cascade=>true,

degree=>10);

end;' script

from dba\_tab\_statistics a

where owner = 'BINHTV'

and table\_name not like 'XXX%' and table\_name not like 'TMP%' and (stale\_stats is null or stale\_stats = 'YES') and object\_type = 'TABLE'

--and table\_name in ('TAB1','TAB2')

and not exists (select 1 from dba\_tab\_statistics where owner = a.owner and table\_name = a.table\_name and object\_type = 'PARTITION' and rownum < 2)

order by 1,2 desc;

Script gather bảng partition STALE 2021

SELECT table\_name, partition\_name, last\_analyzed,num\_rows,

'begin dbms\_stats.gather\_table\_stats(ownname =>'''||owner||''',tabname =>'''||table\_name||''',partname'||'=>'''

|| PARTITION\_NAME

|| ''',granularity=>''partition'',cascade=> TRUE,force=>TRUE,estimate\_percent=>10,'

|| 'method\_opt=>''FOR ALL COLUMNS SIZE AUTO'',degree => 8); end;'

script

FROM dba\_tab\_statistics

where

owner='USER1'

--and table\_name in ('TAB1')

--and num\_rows<10000000

and partition\_name is not null

and partition\_name like '%2021%'

and ((length(partition\_name) = 12 AND to\_date(substr(partition\_name,5,8),'YYYYMMDD')<= trunc(sysdate)

)

or (length(partition\_name) =10 AND to\_date(substr(partition\_name,5,6),'YYYYMM')<=trunc(sysdate)

)

)

and (stale\_stats='YES' or stale\_stats is null);

SELECT /\* GATHER TABLE PARTITION STABLE \*/ table\_name, partition\_name, last\_analyzed,num\_rows,

'begin dbms\_stats.gather\_table\_stats(ownname =>'''||owner||''',tabname =>'''||table\_name||''',partname'||'=>'''

|| PARTITION\_NAME

|| ''',granularity=>''partition'',cascade=> TRUE,force=>TRUE,estimate\_percent=>10,'

|| 'method\_opt=>''FOR ALL COLUMNS SIZE AUTO'',degree => 8); end;'

script

FROM dba\_tab\_statistics

where

((owner='USER2' and table\_name in ('TAB5'))

or

owner='USER1' and table\_name in ('TAB1','TAB2','TAB3','TAB4')

)

and table\_name not like '%XXX%' and table\_name not like '%TMP%' and table\_name not like '%TEMP%' and table\_name not like '%TEST%' and table\_name not like '%$%'

--and table\_name in ('TAB1','','TAB2','TAB3')

-- and table\_name='TAB5'

--and num\_rows<10000000

and partition\_name is not null

and partition\_name like '%2021%'

and ((length(partition\_name) = 12 AND to\_date(substr(partition\_name,5,8),'YYYYMMDD')<= trunc(sysdate)

)

or (length(partition\_name) =10 AND to\_date(substr(partition\_name,5,6),'YYYYMM')<=trunc(sysdate)

)

)

and to\_date(substr(partition\_name,5,8),'yyyymmdd')> sysdate-30

and to\_date(substr(partition\_name,5,8),'yyyymmdd') < sysdate

and (stale\_stats='YES' or stale\_stats is null);

Tips: Gather tự động cho các partition:

begin

dbms\_stats.gather\_table\_stats(ownname=>'BINH\_OWNER',

tabname=> 'TRÁN',

partname=>'DATA20210304',

degree=>dbms\_stats.auto\_degree,

granularity=>'APPROX\_GLOBAL AND PARTITION') ;

end;

d. Partition

-- 1 partition

begin

dbms\_stats.gather\_table\_stats(ownname =>'BINHTV',

tabname =>'TAB1',

partname=>'DATA20161130',

granularity=>'partition',

cascade=> FALSE,

force=>TRUE,

estimate\_percent=>DBMS\_STATS.AUTO\_SAMPLE\_SIZE,

method\_opt=>'FOR ALL COLUMNS SIZE AUTO',

degree => 4);

end;

-- Cả bảng

SELECT table\_name, partition\_name, last\_analyzed,

'exec dbms\_stats.gather\_table\_stats(ownname =>'''||table\_owner||''',tabname =>'''||table\_name||''',partname'||'=>'''

|| PARTITION\_NAME

|| ''',granularity=>''partition'',cascade=> TRUE,force=>TRUE,estimate\_percent=>DBMS\_STATS.AUTO\_SAMPLE\_SIZE,'

|| 'method\_opt=>''FOR ALL COLUMNS SIZE AUTO'',degree => 8);'

script

FROM dba\_tab\_partitions

WHERE table\_name in ('TAB1')

and table\_owner in ('BINHTV')

--AND (last\_analyzed is null or num\_rows is null)

and num\_rows is null

--and partition\_name like '%2018%'

--AND to\_date(substr(partition\_name,5,8),'YYYYMMDD')>sysdate and ((length(partition\_name) = 12 AND to\_date(substr(partition\_name,5,8),'YYYYMMDD')<= trunc(sysdate)

)

or (length(partition\_name) =10 AND to\_date(substr(partition\_name,5,6),'YYYYMM')<=trunc(sysdate)

)

)

ORDER BY table\_name, partition\_name desc;

SELECT table\_name, partition\_name, last\_analyzed,

'exec dbms\_stats.gather\_table\_stats(ownname =>'''||table\_owner||''',tabname =>'''||table\_name||''',partname'||'=>'''

|| PARTITION\_NAME

|| ''',granularity=>''partition'',cascade=> TRUE,force=>TRUE,estimate\_percent=>DBMS\_STATS.AUTO\_SAMPLE\_SIZE,'

|| 'method\_opt=>''FOR ALL COLUMNS SIZE AUTO'',degree => 8);'

script

FROM dba\_tab\_partitions

WHERE table\_name in ('TAB1','TAB2')

and table\_owner in ('BINHTV')

AND (last\_analyzed is null or num\_rows is null)

and partition\_name like '%2018%'

--AND to\_date(substr(partition\_name,5,8),'YYYYMMDD')>sysdate and ((length(partition\_name) = 12 AND to\_date(substr(partition\_name,5,8),'YYYYMMDD')<= trunc(sysdate)

)

or (length(partition\_name) =10 AND to\_date(substr(partition\_name,5,6),'YYYYMM')<=trunc(sysdate)

)

)

ORDER BY table\_name, partition\_name;

-- Tự động tạo thông tin gather cho các partition

begin

dbms\_stats.gather\_table\_stats(ownname=>'BINHTV',

tabname=> 'TAB1',

partname=>'DATA202011',

degree=>dbms\_stats.auto\_degree,

granularity=>'APPROX\_GLOBAL AND PARTITION') ;

end;

e. Gather Index: Không lock index --> Ít dùng vì gather partition || table thì index đi kèm sẽ được gather theo

-- Index non-partition

EXEC DBMS\_STATS.gather\_index\_stats('SCOTT', 'EMPLOYEES\_PK', estimate\_percent => 15,degree => 8);

-- Index partition

EXEC DBMS\_STATS.gather\_index\_stats('SCOTT', 'EMPLOYEES\_I1', estimate\_percent => 15,degree => 8);

--+ Tuong minh

BEGIN

SYS.DBMS\_STATS.GATHER\_INDEX\_STATS (

OwnName => ‘TEST\_OWNER’

,IndName => 'TABLE1\_I1'

,Estimate\_Percent => 10

,Degree => 4

,No\_Invalidate => FALSE);

END;

/

f. Gather dictionary statistic

Non-CDB: SQL> EXEC DBMS\_STATS.GATHER\_DICTIONARY\_STATS;

CDB:

+ All PDB:

$ORACLE\_HOME/perl/bin/perl $ORACLE\_HOME/rdbms/admin/catcon.pl -l /tmp -b gatherstats -- --x"exec dbms\_stats.gather\_dictionary\_stats"

+ PDB cụ thể:

$ORACLE\_HOME/perl/bin/perl $ORACLE\_HOME/rdbms/admin/catcon.pl -l /tmp -c

'SALES1' -b gatherstats -- --x"exec dbms\_stats.gather\_dictionary\_stats"

f. Gather fixed object statistic

SQL> execute dbms\_stats.gather\_fixed\_objects\_stats;

2. Set Gather

exec DBMS\_STATS.SET\_TABLE\_STATS ( ownname=>'TEST\_OWNER’,tabname=> 'TAB1’, partname=>'DATA20110524', numrows=>56000000, numblks=>1000000);

exec DBMS\_STATS.SET\_INDEX\_STATS ( ownname=>'TEST\_OWNER’,indname=> 'TAB1\_I1', partname=>'DATA20121124', numrows=>37037005, indlevel=>3,numdist=>2337312);

exec DBMS\_STATS.SET\_TABLE\_STATS ( ownname=>'TEST\_OWNER',tabname=> 'TAB1', numrows=>5600000, numblks=>100000);

3. Delete gather

EXEC DBMS\_STATS.delete\_database\_stats;

EXEC DBMS\_STATS.delete\_schema\_stats('SCOTT');

EXEC DBMS\_STATS.delete\_table\_stats('SCOTT', 'EMP');

EXEC DBMS\_STATS.delete\_column\_stats('SCOTT', 'EMP', 'EMPNO');

EXEC DBMS\_STATS.delete\_index\_stats('SCOTT', 'EMP\_PK');

EXEC DBMS\_STATS.delete\_dictionary\_stats;

III. ANALYZE

1. Analyze database: Không hỗ trợ

2. ANALYZE SCHEMA

· Cả schema bang partition

SELECT table\_name, partition\_name, last\_analyzed, 'ANALYZE TABLE ' ||table\_owner|| '.'|| table\_name|| ' partition ('|| partition\_name || ') ESTIMATE STATISTICS SAMPLE 10 PERCENT' script

FROM all\_tab\_partitions

WHERE partition\_name LIKE

'DATA' || TO\_CHAR (sysdate, 'YYYYMM') || '%'

AND table\_name NOT LIKE '%$%'

and table\_owner in ('TEST\_OWNER', 'TEST2\_OWNER')

AND NVL (last\_analyzed, SYSDATE - 15) < sysdate

AND NVL (last\_analyzed, SYSDATE - 15) < sysdate

AND to\_date(substr(partition\_name,5,8),'YYYYMMDD')<sysdate

AND to\_date(substr(partition\_name,5,8),'YYYYMMDD')>sysdate-4

AND (last\_analyzed is null or num\_rows is null)

and ((length(partition\_name) = 12 AND to\_date(substr(partition\_name,5,8),'YYYYMMDD')<= trunc(sysdate)

)

or (length(partition\_name) =10 AND to\_date(substr(partition\_name,5,6),'YYYYMM')<=trunc(sysdate)

)

)

ORDER BY table\_name, partition\_name;

declare

CURSOR c\_partitioned\_tables

IS

SELECT table\_name, partition\_name, last\_analyzed,

'ANALYZE TABLE ' ||table\_owner|| '.'

|| table\_name

|| ' partition ('

|| partition\_name

|| ') ESTIMATE STATISTICS SAMPLE 10 PERCENT'

script

FROM dba\_tab\_partitions

WHERE table\_name NOT LIKE '%$%'

and table\_owner in ('TEST\_OWNER')

AND (last\_analyzed is null or num\_rows is null)

and ((length(partition\_name) = 12 AND to\_date(substr(partition\_name,5,8),'YYYYMMDD')<= trunc(sysdate)

)

or (length(partition\_name) =10 AND to\_date(substr(partition\_name,5,6),'YYYYMM')<=trunc(sysdate)

)

)

ORDER BY table\_name, partition\_name;

v\_table\_name VARCHAR2 (100);

v\_par\_name VARCHAR2 (100);

BEGIN

FOR v\_partitioned\_tables IN c\_partitioned\_tables

LOOP

v\_table\_name := v\_partitioned\_tables.table\_name;

v\_par\_name := v\_partitioned\_tables.partition\_name;

EXECUTE IMMEDIATE v\_partitioned\_tables.script;

--dbms\_output.put\_line(v\_partitioned\_tables.script);

END LOOP;

EXCEPTION

WHEN OTHERS

THEN

null;

END;

· Cả schema bang non-partition

3. ANALYZE TABLE

· Table non-partition

analyze table mc\_subscriber estimate statistics sample 10 percent

· Table partition

ANALYZE TABLE TAB1 partition (DATA20120701) ESTIMATE STATISTICS SAMPLE 10 PERCENT

+ All partition của 1 bảng:

select 'analyze table ' || table\_owner ||'.' || table\_name || ' partition(' || partition\_name || ') estimate statistics sample 10 percent;' from DBA\_TAB\_PARTITIONS where table\_owner=’TEST\_OWNER’ and table\_name='TAB1’ and last\_analyzed is null

and ((length(partition\_name) = 12 AND to\_date(substr(partition\_name,5,8),'YYYYMMDD')<= trunc(sysdate)

)

or (length(partition\_name) =10 AND to\_date(substr(partition\_name,5,6),'YYYYMM')<=trunc(sysdate)

)

);

+ 1 partition:

ANALYZE TABLE TEST\_OWNER.TAB1 partition (DATA20160809) ESTIMATE STATISTICS SAMPLE 10 PERCENT

Ghi chú:

ANALYZE TABLE test VALIDATE STRUCTURE CASCADE ONLINE;

4. Analyze Index:

-- Index non-partition

-- Tương đương câu GATHER ở trên

ANALYZE INDEX index\_name COMPUTE STATISTICS;

-- VALIDATE cấu trúc --> THẬN TRỌNG KHI CHẠY!!!

--+ Gây LOCK index IDX2 do đó các thao tác DML vào bảng bị wait, sau khi validate xong mới được thực hiện --> Gây lock, active session tăng cao

ANALYZE INDEX TEST\_OWNER.IDX2 VALIDATE STRUCTURE;

--+ Index partition

ANALYZE INDEX TEST\_OWNER.TAB1 \_IDX1 partition(DATA20141224) VALIDATE STRUCTURE;

--+ Sau khi chạy được validate check xem độ phân mảnh của index (nếu del\_lf\_row/lf\_rows > 30%)

SELECT blocks, pct\_used, distinct\_keys

lf\_rows, del\_lf\_rows

FROM index\_stats;

5. SET ANALYZE

Đang cập nhật.....

6. DELETE ANALYZE

ANALYZE TABLE TAB1 DELETE STATISTICS;

7. MONITOR KHI GATHER HOẶC ANALYZE

Nếu trong giờ hành chính có thể gây treo DB Core khi analyze bảng core 1 partition hoăc 1 bảng lớn (>50GB), cần kill ngay:

SELECT /\*1.ActiveSession\*/ distinct s.inst\_id i#, s.username, s.SID SID, s.osuser, s.machine,DECODE(S.WAIT\_TIME, 0, S.EVENT, 'CPU') ACTION,

s.sql\_id, SUBSTR(DECODE(SS.SQL\_TEXT, NULL, AA.NAME, SS.SQL\_TEXT), 1, 1000) SQLTEXT,s.logon\_time,s.p1text, S.P1, s.p2text, S.P2, s.p3text, S.P3

FROM GV$SESSION S, GV$SQLSTATS SS, AUDIT\_ACTIONS AA

WHERE S.STATUS = 'ACTIVE' AND S.SQL\_ID = SS.SQL\_ID (+) AND AA.ACTION = S.COMMAND and s.type != 'BACKGROUND' AND S.TYPE = 'USER'

--and s.username NOT in ('SYS','SYSMAN','DBSNMP','GGATE','GOLDENGATE')

AND username like 'SYS%'

--and DECODE(S.WAIT\_TIME, 0, S.EVENT, 'CPU') like '%cell single block physical read%'

and (lower(ss.sql\_text) like lower('%alter table%') or lower(ss.sql\_text) like lower('%analyz%') )

and lower(ss.sql\_text) not like lower('%ACTIVE, LOCK%')

--and s.sid=4588

--and s.machine like '%BINHTV%'

--and s.sql\_id ='ccwg0nqr1zbu7'

SELECT /\*5.SID\*/ 'kill -9 ' || spid a, a.INST\_ID,A.SQL\_ID,A.SID, A.SERIAL#, a.USERNAME, a.STATUS,A.SCHEMANAME,a.OSUSER,A.MACHINE,A.PROGRAM,A.TYPE,A.LOGON\_TIME,a.prev\_exec\_start,BACKGROUND

FROM gv$session a, gv$process b

WHERE b.addr = a.paddr

AND a.inst\_id=b.inst\_id

--and b.inst\_id=2

AND a.sid in (

select sid from (SELECT /\*1.ActiveSession\*/ distinct s.inst\_id i#, s.username, s.SID SID, s.osuser, s.machine,DECODE(S.WAIT\_TIME, 0, S.EVENT, 'CPU') ACTION,

s.sql\_id, SUBSTR(DECODE(SS.SQL\_TEXT, NULL, AA.NAME, SS.SQL\_TEXT), 1, 1000) SQLTEXT,s.logon\_time,s.p1text, S.P1, s.p2text, S.P2, s.p3text, S.P3

FROM GV$SESSION S, GV$SQLSTATS SS, AUDIT\_ACTIONS AA

WHERE S.STATUS = 'ACTIVE' AND S.SQL\_ID = SS.SQL\_ID (+) AND AA.ACTION = S.COMMAND and s.type != 'BACKGROUND' AND S.TYPE = 'USER'

--and s.username NOT in ('SYS','SYSMAN','DBSNMP','GGATE','GOLDENGATE')

AND username like 'SYS%'

--and DECODE(S.WAIT\_TIME, 0, S.EVENT, 'CPU') like '%cell single block physical read%'

and (lower(ss.sql\_text) like lower('%alter table%') or lower(ss.sql\_text) like lower('%analyz%') )

and lower(ss.sql\_text) not like lower('%ACTIVE, LOCK%'))

--and s.sid=4588

--and s.machine like '%BINHTV%'

--and s.sql\_id ='ccwg0nqr1zbu7'

)

and type='USER'

order by A.LOGON\_TIME;

MỘT SỐ LỖI HAY GẶP:

Lỗi 1: Gather dữ liệu offline

begin

dbms\_stats.gather\_table\_stats

(ownname=>'USER1',

tabname=>'TAB1',

partname=>'DATA20210609',

estimate\_percent => 10,

method\_opt => 'FOR ALL COLUMNS SIZE AUTO',

cascade=>true,

degree=>10);

end;

-- Cần datafile từ rất lâu rồi (other201812)

ORA-00376: file 13836 cannot be read at this time

ORA-01110: data file 13836: '+DATA/dbaviet/datafile/other201812.724.995998009'

ORA-06512: at "SYS.DBMS\_STATS", line 24281

ORA-06512: at "SYS.DBMS\_STATS", line 24332

ORA-06512: at line 2

Giải pháp:

ANALYZE TABLE USER1.TAB1 partition (DATA20210622) ESTIMATE STATISTICS SAMPLE 10 PERCENT;

BEGIN

SYS.DBMS\_STATS.GATHER\_TABLE\_STATS (

OwnName => 'USER1'

,TabName => 'TAB1'

,PartName => 'DATA20210609'

,Granularity => 'PARTITION'

,Estimate\_Percent => 10

,Method\_Opt => 'FOR ALL COLUMNS SIZE AUTO'

,Degree => 4

,Cascade => TRUE

,No\_Invalidate => FALSE);

END;

Lỗi 2: Nghiệp vụ đang chạy DML nhiều thì không gather đựợc

[1]: ORA-04021: timeout occurred while waiting to lock object

[1]: ORA-06512: at "SYS.DBMS\_STATS", line 24281

[1]: ORA-06512: at "SYS.DBMS\_STATS", line 24332

[1]: ORA-06512: at line 2

--> Giải pháp: Dừng nghiệp vụ mới gather được hoặc set tĩnh

## Tài liệu tham khảo:

<https://www.tranvanbinh.vn/2022/09/su-dung-cong-cu-sql-tunning-advisor-e.html>

<https://www.tranvanbinh.vn/2020/01/huong-dan-su-dung-cong-cu-oracle-sql.html>

<https://www.tranvanbinh.vn/2022/09/huong-dan-chay-dbmssqltune-voi-cau-lenh.html>

# BÀI 20: RESOURCE MANAGER

## Tài liệu tham khảo:

<https://www.tranvanbinh.vn/2020/12/thu-tuc-cau-hinh-resource-manager.html>

<https://www.tranvanbinh.vn/2020/12/thu-tuc-cau-hinh-resource-manager.html>

<https://www.tranvanbinh.vn/2020/08/gioi-han-active-session-paralel-cua.html>

<https://www.tranvanbinh.vn/2020/12/quan-ly-scheduler-job-trong-oracle.html>

<https://www.tranvanbinh.vn/2020/12/quan-ly-scheduler-job-trong-oracle.html>

<https://www.tranvanbinh.vn/2020/12/quan-ly-job-trong-oracle-database.html>

<https://www.tranvanbinh.vn/2022/03/thu-tuc-add-datafile-tu-ong-tren-oracle.html>

<https://www.tranvanbinh.vn/2020/12/script-add-partition-tu-ong.html>

<https://www.tranvanbinh.vn/2021/09/tu-ong-drop-cac-partition-cu-trong.html>

<https://www.tranvanbinh.vn/2020/01/thu-tuc-gui-email-nhieu-nguoi-tren-co.html>

<https://www.tranvanbinh.vn/2021/08/tu-ong-kill-uer-ung-dung-khi-vuot-qua.html>

<https://www.tranvanbinh.vn/2020/01/mau-bao-cao-co-so-du-lieu-may-chu.html>

<https://www.tranvanbinh.vn/2020/01/mau-bao-cao-co-so-du-lieu-may-chu.html>

<https://www.tranvanbinh.vn/2022/11/goi-chup-tai-inh-ky-dbact.html>

<https://www.tranvanbinh.vn/2022/11/goi-canh-bao-hieu-nang-dbawn.html>

<https://www.tranvanbinh.vn/2022/11/goi-tu-ong-hoa-van-hanh-dbaop.html>

<https://www.tranvanbinh.vn/2022/11/goi-bao-cao-tu-ong-dbarp.html>

<https://www.tranvanbinh.vn/2022/11/goi-bao-cao-tu-ong-dbarp.html>

<https://www.tranvanbinh.vn/2020/01/he-lo-bi-mat-giam-sat-co-so-du-lieu-chu.html>

<https://www.tranvanbinh.vn/2022/03/bao-cao-hien-trang-co-so-du-lieu-oracle.html>

# BÀI 21: SCHEDULER

## Tạo job add datafile tự động

--Thủ tục add datafile TỰ ĐỘNG trên Oracle Database ASM

--Mục đích: add partition tự động của DB lớn (không phải mọi tablespace đều add partition) trên Oracle Database

--BƯỚC 1: TẠO CÁC BẢNG LƯU LOG:

CREATE TABLE oaz19.dbamf\_log\_op

(id NUMBER,

name VARCHAR2(1000 BYTE),

event\_date TIMESTAMP (6) DEFAULT sysdate,

type VARCHAR2(1000 BYTE),

note VARCHAR2(1000 BYTE))

PCTFREE 10

INITRANS 1

MAXTRANS 255

TABLESPACE users

STORAGE (

INITIAL 65536

NEXT 1048576

MINEXTENTS 1

MAXEXTENTS 2147483645

)

NOCACHE

MONITORING

NOPARALLEL

LOGGING

/

CREATE SEQUENCE oaz19.DBAMF\_LOG\_OP\_SEQ

START WITH 1

MAXVALUE 9999999999999999999999999999

MINVALUE 1

NOCYCLE

CACHE 20

NOORDER;

CREATE TABLE oaz19.DBAMF\_LOG\_JOBS

(

ID NUMBER,

NAME VARCHAR2(4000 BYTE),

STATUS VARCHAR2(50 BYTE),

EVENT\_DATE DATE DEFAULT sysdate,

NOTE VARCHAR2(2000 BYTE)

)

TABLESPACE USERS

PCTUSED 0

PCTFREE 10

INITRANS 1

MAXTRANS 255

LOGGING

NOCOMPRESS

NOCACHE

NOPARALLEL

MONITORING;

--DROP SEQUENCE oaz19.DBAMF\_LOG\_JOBS\_SEQ;

CREATE SEQUENCE oaz19.DBAMF\_LOG\_JOBS\_SEQ

START WITH 1

MAXVALUE 9999999999999999999999999999

MINVALUE 1

NOCYCLE

CACHE 20

NOORDER;

select \* from oaz19.DBAMF\_LOG\_OP;

select oaz19.DBAMF\_LOG\_OP\_SEQ.nextval from dual;

--BƯỚC 2: TẠO THỦ TỤC TRÊN SYS

create or replace PROCEDURE sys.auto\_extend\_space

IS

v\_err varchar2(1000):='';

free\_space\_low\_level NUMBER := 50000;

CURSOR c\_free\_space -- get tablespace free left 200MB.

IS

SELECT a.tablespace\_name,100 - ROUND ( (NVL (b.bytes\_free, 0) / a.bytes\_alloc) \* 100) "%Usage",

ROUND (a.bytes\_alloc / 1024 / 1024) "Size MB",

ROUND (a.bytes\_alloc / 1024 / 1024)- ROUND (NVL (b.bytes\_free, 0) / 1024 / 1024) "Used MB",

ROUND (NVL (b.bytes\_free, 0) / 1024 / 1024) "Free MB",

--ROUND ( (NVL (b.bytes\_free, 0) / a.bytes\_alloc) \* 100) "%Free",

ROUND (maxbytes / 1048576) "Max MB",

round(maxbytes/1048576-(ROUND (a.bytes\_alloc / 1024 / 1024)- ROUND (NVL (b.bytes\_free, 0) / 1024 / 1024)),0) "Free\_MB\_Max",

ROUND (ROUND ( (a.bytes\_alloc - NVL (b.bytes\_free, 0)) / 1024 / 1024)/ ROUND (maxbytes / 1048576) \* 100) "%Used of Max"

FROM (SELECT f.tablespace\_name, SUM (f.bytes) bytes\_alloc, SUM (DECODE (f.autoextensible, 'YES', f.maxbytes, 'NO', f.bytes)) maxbytes

FROM dba\_data\_files f

GROUP BY tablespace\_name) a,

(SELECT f.tablespace\_name, SUM (f.bytes) bytes\_free FROM dba\_free\_space f GROUP BY tablespace\_name) b

WHERE a.tablespace\_name = b.tablespace\_name(+)

--and ( a.tablespace\_name not in ('DUMP'))

-- and "Free\_MB\_Max" < 50000

and round(maxbytes/1048576-(ROUND (a.bytes\_alloc / 1024 / 1024)- ROUND (NVL (b.bytes\_free, 0) / 1024 / 1024)),0) < 100000

order by "%Used of Max" desc;

v\_sql VARCHAR2 (200);

msg VARCHAR2 (1000);

next\_datafile VARCHAR2 (1000);

BEGIN

insert into oaz19.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(oaz19.dbamf\_log\_jobs\_seq.nextval,'Starting sys.dba\_op.extend\_space',1,sysdate,'oaz19.dbamf\_log\_jobs, oaz19.dbamf\_log\_op');

commit;

FOR v\_free\_space IN c\_free\_space -- tablespace < 200MB

LOOP

BEGIN

v\_sql :=

'ALTER TABLESPACE '

|| v\_free\_space.tablespace\_name

|| ' ADD DATAFILE size 10m autoextend on next 1m'; --- 1G, autoextend 100m;

--DBMS\_OUTPUT.put\_line (v\_sql);

EXECUTE IMMEDIATE v\_sql;

insert into oaz19.dbamf\_log\_op (name, type, note) values (v\_sql, 'df','Add a new datafile');

commit;

EXCEPTION

WHEN OTHERS

THEN

v\_err := substr(SQLERRM,1,200);

insert into oaz19.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(oaz19.dbamf\_log\_jobs\_seq.nextval,'Error sys.dba\_op.extend\_space',-1,sysdate,'Error sys.dba\_op.extend\_spac, '||v\_err);

commit;

END;

END LOOP;

insert into oaz19.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(oaz19.dbamf\_log\_jobs\_seq.nextval,'Completed sys.dba\_op.extend\_space',1,sysdate,'oaz19.dbamf\_log\_jobs');

commit;

EXCEPTION

WHEN others THEN

v\_err := substr(SQLERRM,1,200);

insert into oaz19.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(oaz19.dbamf\_log\_jobs\_seq.nextval,'Error sys.dba\_op.extend\_space',-1,sysdate,'Error sys.dba\_op.extend\_spac, '||v\_err);

commit;

END;

--BƯỚC 3: Tạo job từ sched.Jobs: 10 phút chạy 1 lần

--Vào giao diện hoặc chạy câu lệnh

--BEGIN

-- SYS.DBMS\_SCHEDULER.DROP\_JOB

-- (job\_name => 'SYS.AUTO\_ADD\_DATAFILE');

--END;

--/

BEGIN

SYS.DBMS\_SCHEDULER.DROP\_JOB

(job\_name => 'SYS.AUTO\_ADD\_DATAFILE');

END;

/

BEGIN

SYS.DBMS\_SCHEDULER.CREATE\_JOB

(

job\_name => 'SYS.AUTO\_ADD\_DATAFILE'

,start\_date => TO\_TIMESTAMP\_TZ('2023/07/21 20:53:02.109327 +07:00','yyyy/mm/dd hh24:mi:ss.ff tzh:tzm')

,repeat\_interval => 'FREQ=HOURLY;INTERVAL=1;BYMINUTE=0;BYSECOND=0'

,end\_date => NULL

,job\_class => 'DEFAULT\_JOB\_CLASS'

,job\_type => 'STORED\_PROCEDURE'

,job\_action => 'SYS.AUTO\_EXTEND\_SPACE'

,comments => 'AUTO\_ADD\_DATAFILE'

);

SYS.DBMS\_SCHEDULER.SET\_ATTRIBUTE

( name => 'SYS.AUTO\_ADD\_DATAFILE'

,attribute => 'RESTARTABLE'

,value => FALSE);

SYS.DBMS\_SCHEDULER.SET\_ATTRIBUTE\_NULL

( name => 'SYS.AUTO\_ADD\_DATAFILE'

,attribute => 'MAX\_FAILURES');

SYS.DBMS\_SCHEDULER.SET\_ATTRIBUTE\_NULL

( name => 'SYS.AUTO\_ADD\_DATAFILE'

,attribute => 'MAX\_RUNS');

SYS.DBMS\_SCHEDULER.SET\_ATTRIBUTE

( name => 'SYS.AUTO\_ADD\_DATAFILE'

,attribute => 'STOP\_ON\_WINDOW\_CLOSE'

,value => FALSE);

SYS.DBMS\_SCHEDULER.SET\_ATTRIBUTE

( name => 'SYS.AUTO\_ADD\_DATAFILE'

,attribute => 'JOB\_PRIORITY'

,value => 3);

SYS.DBMS\_SCHEDULER.SET\_ATTRIBUTE\_NULL

( name => 'SYS.AUTO\_ADD\_DATAFILE'

,attribute => 'SCHEDULE\_LIMIT');

SYS.DBMS\_SCHEDULER.SET\_ATTRIBUTE

( name => 'SYS.AUTO\_ADD\_DATAFILE'

,attribute => 'AUTO\_DROP'

,value => FALSE);

SYS.DBMS\_SCHEDULER.SET\_ATTRIBUTE

( name => 'SYS.AUTO\_ADD\_DATAFILE'

,attribute => 'RESTART\_ON\_RECOVERY'

,value => FALSE);

SYS.DBMS\_SCHEDULER.SET\_ATTRIBUTE

( name => 'SYS.AUTO\_ADD\_DATAFILE'

,attribute => 'RESTART\_ON\_FAILURE'

,value => FALSE);

SYS.DBMS\_SCHEDULER.SET\_ATTRIBUTE

( name => 'SYS.AUTO\_ADD\_DATAFILE'

,attribute => 'STORE\_OUTPUT'

,value => TRUE);

SYS.DBMS\_SCHEDULER.ENABLE

(name => 'SYS.AUTO\_ADD\_DATAFILE');

END;

/

select \* from oaz19.dbamf\_log\_jobs order by id;

--BƯỚC 4: MONITOR JOBS:

-- Check job chạy lâu nhất, job chạy gần nhất cách đây 60 ngày

select owner,job\_name,job\_creator,state,job\_type,last\_run\_duration,job\_action,schedule\_type,start\_date,repeat\_interval,last\_start\_date,next\_run\_date from dba\_scheduler\_jobs

where --owner not like 'SYS'

--and

state!='DISABLED'

--and last\_start\_date > sysdate-60

order by last\_run\_duration desc;

SELECT \* FROM dba\_scheduler\_running\_jobs WHERE job\_name = 'AUTO\_ADD\_DATAFILE';

SELECT \* FROM dba\_scheduler\_jobs WHERE job\_name = 'AUTO\_ADD\_DATAFILE';

SELECT \* FROM dba\_scheduler\_job\_log WHERE job\_name = 'AUTO\_ADD\_DATAFILE';

--Hoặc kiểm tra JOB của schema khác

SELECT \* FROM user\_scheduler\_jobs WHERE job\_name = 'ADD\_DATA\_FILE';

SELECT \* FROM user\_scheduler\_job\_log WHERE job\_name = 'ADD\_DATA\_FILE';

--Hoặc

select \* from oaz19.dbamf\_log\_op;

select \* from oaz19.dbamf\_log\_jobs ;

## DBA\_CT

Mục đích: Gói chụp tải định kỳ DBA\_CT

create user binhtv identified by oracle;

grant connect, resource, dba to binhtv;

--DROP TABLE BINHTV.DBAMF\_LOG\_JOBS CASCADE CONSTRAINTS;

CREATE TABLE BINHTV.DBAMF\_LOG\_JOBS

(

ID NUMBER,

NAME VARCHAR2(4000 BYTE),

STATUS VARCHAR2(50 BYTE),

EVENT\_DATE DATE DEFAULT sysdate,

NOTE VARCHAR2(2000 BYTE)

)

TABLESPACE USERS

PCTUSED 0

PCTFREE 10

INITRANS 1

MAXTRANS 255

STORAGE (

INITIAL 64K

NEXT 1M

MINEXTENTS 1

MAXEXTENTS UNLIMITED

PCTINCREASE 0

BUFFER\_POOL DEFAULT

)

LOGGING

NOCOMPRESS

NOCACHE

MONITORING;

-- Start of DDL Script for Table BINHTV.DBAMF\_CT\_TBS\_SIZE

-- Generated 5-Nov-2020 16:12:10 from BINHTV@vietdba

-- Drop the old instance of DBAMF\_CT\_TBS\_SIZE

DROP TABLE binhtv.dbamf\_ct\_tbs\_size

/

CREATE TABLE binhtv.dbamf\_ct\_tbs\_size

(id NUMBER,

tbs\_name VARCHAR2(100 BYTE),

size\_gb NUMBER,

event\_date DATE DEFAULT sysdate,

note VARCHAR2(1000 BYTE))

PCTFREE 10

INITRANS 1

MAXTRANS 255

TABLESPACE users

STORAGE (

INITIAL 65536

NEXT 1048576

MINEXTENTS 1

MAXEXTENTS 2147483645

)

NOCACHE

MONITORING

NOPARALLEL

LOGGING

/

DROP SEQUENCE BINHTV.DBAMF\_LOG\_JOBS\_SEQ;

CREATE SEQUENCE BINHTV.DBAMF\_LOG\_JOBS\_SEQ

START WITH 2156723

MAXVALUE 9999999999999999999999999999

MINVALUE 1

NOCYCLE

CACHE 20

NOORDER;

-- End of DDL Script for Table BINHTV.DBAMF\_CT\_TBS\_SIZE

-- Start of DDL Script for Table BINHTV.DBAMF\_CT\_USER\_SIZE

-- Generated 5-Nov-2020 16:12:14 from BINHTV@vietdba

-- Drop the old instance of DBAMF\_CT\_USER\_SIZE

DROP TABLE binhtv.dbamf\_ct\_user\_size

/

CREATE TABLE binhtv.dbamf\_ct\_user\_size

(id NUMBER,

user\_name VARCHAR2(100 BYTE),

size\_mb NUMBER,

event\_date DATE DEFAULT sysdate,

note VARCHAR2(1000 BYTE))

PCTFREE 10

INITRANS 1

MAXTRANS 255

TABLESPACE users

STORAGE (

INITIAL 65536

NEXT 1048576

MINEXTENTS 1

MAXEXTENTS 2147483645

)

NOCACHE

MONITORING

NOPARALLEL

LOGGING

/

-- End of DDL Script for Table BINHTV.DBAMF\_CT\_USER\_SIZE

-- Start of DDL Script for Table BINHTV.DBAMF\_CT\_VMSTAT

-- Generated 5-Nov-2020 16:12:18 from BINHTV@vietdba

-- Drop the old instance of DBAMF\_CT\_VMSTAT

DROP TABLE binhtv.dbamf\_ct\_vmstat

/

CREATE TABLE binhtv.dbamf\_ct\_vmstat

(event\_date DATE,

duration NUMBER,

server\_name VARCHAR2(30 BYTE),

runque\_waits NUMBER,

swap\_in NUMBER,

swap\_out NUMBER,

cpu\_user NUMBER,

cpu\_sys NUMBER,

cpu\_idle NUMBER,

cpu\_wait NUMBER,

swapd NUMBER,

ram\_free\_k NUMBER)

PCTFREE 10

INITRANS 1

MAXTRANS 255

TABLESPACE users

STORAGE (

INITIAL 65536

NEXT 1048576

MINEXTENTS 1

MAXEXTENTS 2147483645

)

NOCACHE

MONITORING

NOPARALLEL

LOGGING

/

-- Grants for Table

GRANT INSERT ON binhtv.dbamf\_ct\_vmstat TO monitor

/

-- End of DDL Script for Table BINHTV.DBAMF\_CT\_VMSTAT

-- Start of DDL Script for Table BINHTV.DBAMF\_DISK\_USAGE

-- Generated 5-Nov-2020 16:12:22 from BINHTV@vietdba

-- Drop the old instance of DBAMF\_DISK\_USAGE

DROP TABLE binhtv.dbamf\_disk\_usage

/

CREATE TABLE binhtv.dbamf\_disk\_usage

(host\_name VARCHAR2(40 BYTE),

disk\_name VARCHAR2(50 BYTE),

disk\_size VARCHAR2(20 BYTE),

disk\_free VARCHAR2(20 BYTE),

disk\_percen VARCHAR2(10 BYTE),

disk\_mount VARCHAR2(50 BYTE),

db\_ip VARCHAR2(20 BYTE),

event\_date DATE DEFAULT sysdate)

PCTFREE 10

INITRANS 1

MAXTRANS 255

TABLESPACE users

STORAGE (

INITIAL 65536

NEXT 1048576

MINEXTENTS 1

MAXEXTENTS 2147483645

)

NOCACHE

MONITORING

NOPARALLEL

LOGGING

/

-- End of DDL Script for Table BINHTV.DBAMF\_DISK\_USAGE

-- Start of DDL Script for Table BINHTV.DBAMF\_LOG\_GG

-- Generated 5-Nov-2020 16:12:26 from BINHTV@vietdba

-- Drop the old instance of DBAMF\_LOG\_GG

DROP TABLE binhtv.dbamf\_log\_gg

/

CREATE TABLE binhtv.dbamf\_log\_gg

(id NUMBER,

program VARCHAR2(20 BYTE),

status VARCHAR2(50 BYTE),

gg\_group VARCHAR2(100 BYTE),

lag VARCHAR2(50 BYTE),

time\_since\_chkpt VARCHAR2(50 BYTE),

event\_date DATE DEFAULT sysdate)

PCTFREE 10

INITRANS 1

MAXTRANS 255

TABLESPACE users

STORAGE (

INITIAL 65536

NEXT 1048576

MINEXTENTS 1

MAXEXTENTS 2147483645

)

NOCACHE

MONITORING

NOPARALLEL

LOGGING

/

-- End of DDL Script for Table BINHTV.DBAMF\_LOG\_GG

-- Start of DDL Script for Table BINHTV.DBAMF\_LOG\_JOBS

-- Generated 5-Nov-2020 16:12:31 from BINHTV@vietdba

-- Drop the old instance of DBAMF\_LOG\_JOBS

DROP TABLE binhtv.dbamf\_log\_jobs

/

CREATE TABLE binhtv.dbamf\_log\_jobs

(id NUMBER,

name VARCHAR2(4000 BYTE),

status VARCHAR2(50 BYTE),

event\_date DATE DEFAULT sysdate,

note VARCHAR2(2000 BYTE))

PCTFREE 10

INITRANS 1

MAXTRANS 255

TABLESPACE users

STORAGE (

INITIAL 65536

NEXT 1048576

MINEXTENTS 1

MAXEXTENTS 2147483645

)

NOCACHE

MONITORING

NOPARALLEL

LOGGING

/

-- End of DDL Script for Table BINHTV.DBAMF\_LOG\_JOBS

-- Start of DDL Script for Table BINHTV.DBAMF\_LOG\_OP

-- Generated 5-Nov-2020 16:12:34 from BINHTV@vietdba

-- Drop the old instance of DBAMF\_LOG\_OP

DROP TABLE binhtv.dbamf\_log\_op

/

CREATE TABLE binhtv.dbamf\_log\_op

(id NUMBER,

msg VARCHAR2(1000 BYTE),

event\_date TIMESTAMP (6) DEFAULT sysdate,

type VARCHAR2(1000 BYTE),

note VARCHAR2(1000 BYTE))

PCTFREE 10

INITRANS 1

MAXTRANS 255

TABLESPACE users

STORAGE (

INITIAL 65536

NEXT 1048576

MINEXTENTS 1

MAXEXTENTS 2147483645

)

NOCACHE

MONITORING

NOPARALLEL

LOGGING

/

CREATE SEQUENCE BINHTV.DBAMF\_LOG\_OP\_SEQ

START WITH 1

MAXVALUE 9999999999999999999999999999

MINVALUE 1

NOCYCLE

CACHE 20

NOORDER;

-- End of DDL Script for Table BINHTV.DBAMF\_LOG\_OP

-- Start of DDL Script for Table BINHTV.DBAMF\_LOG\_WN

-- Generated 5-Nov-2020 16:12:37 from BINHTV@vietdba

-- Drop the old instance of DBAMF\_LOG\_WN

DROP TABLE binhtv.dbamf\_log\_wn

/

CREATE TABLE binhtv.dbamf\_log\_wn

(id NUMBER,

msg VARCHAR2(1000 BYTE),

event\_date TIMESTAMP (6) DEFAULT sysdate,

type VARCHAR2(1000 BYTE),

note VARCHAR2(1000 BYTE))

PCTFREE 10

INITRANS 1

MAXTRANS 255

TABLESPACE users

STORAGE (

INITIAL 65536

NEXT 1048576

MINEXTENTS 1

MAXEXTENTS 2147483645

)

NOCACHE

MONITORING

NOPARALLEL

LOGGING

/

-- End of DDL Script for Table BINHTV.DBAMF\_LOG\_WN

-- Start of DDL Script for Table BINHTV.DBAMF\_TAB\_CYCLE

-- Generated 5-Nov-2020 16:12:42 from BINHTV@vietdba

-- Drop the old instance of DBAMF\_TAB\_CYCLE

DROP TABLE binhtv.dbamf\_tab\_cycle

/

CREATE TABLE binhtv.dbamf\_tab\_cycle

(id NUMBER,

owner VARCHAR2(50 BYTE),

table\_name VARCHAR2(200 BYTE),

purpose VARCHAR2(1000 BYTE),

cycle\_days NUMBER,

rw\_days NUMBER,

readonly\_days NUMBER,

size\_mb NUMBER,

num\_rows NUMBER,

note VARCHAR2(1000 BYTE))

PCTFREE 10

INITRANS 1

MAXTRANS 255

TABLESPACE users

STORAGE (

INITIAL 65536

NEXT 1048576

MINEXTENTS 1

MAXEXTENTS 2147483645

)

NOCACHE

MONITORING

NOPARALLEL

LOGGING

/

-- End of DDL Script for Table BINHTV.DBAMF\_TAB\_CYCLE

-- Start of DDL Script for Table BINHTV.DBAMF\_CT\_SQL

-- Generated 5-Nov-2020 16:12:06 from BINHTV@vietdba

-- Drop the old instance of DBAMF\_CT\_SQL

DROP TABLE binhtv.dbamf\_ct\_sql

/

CREATE TABLE binhtv.dbamf\_ct\_sql

(inst\_id NUMBER,

username VARCHAR2(30 BYTE),

sid NUMBER,

osuser VARCHAR2(30 BYTE),

machine VARCHAR2(64 BYTE),

event VARCHAR2(64 BYTE),

sql\_id VARCHAR2(13 BYTE),

sql\_text VARCHAR2(3000 BYTE),

logon\_time DATE,

event\_date DATE DEFAULT sysdate)

PCTFREE 10

INITRANS 1

MAXTRANS 255

TABLESPACE users

STORAGE (

INITIAL 65536

NEXT 1048576

MINEXTENTS 1

MAXEXTENTS 2147483645

)

NOCACHE

MONITORING

NOPARALLEL

LOGGING

/

-- End of DDL Script for Table BINHTV.DBAMF\_CT\_SQL

-- Start of DDL Script for Table BINHTV.DBAMF\_CT\_SESSION

-- Generated 5-Nov-2020 16:12:01 from BINHTV@vietdba

-- Drop the old instance of DBAMF\_CT\_SESSION

DROP TABLE binhtv.dbamf\_ct\_session

/

CREATE TABLE binhtv.dbamf\_ct\_session

(username VARCHAR2(50 BYTE),

status VARCHAR2(10 BYTE),

count\_low VARCHAR2(5 BYTE),

count\_avg VARCHAR2(5 BYTE),

count\_high VARCHAR2(5 BYTE),

count\_critical VARCHAR2(5 BYTE),

notes VARCHAR2(1000 BYTE),

event\_date DATE DEFAULT sysdate,

count NUMBER)

PCTFREE 10

INITRANS 1

MAXTRANS 255

TABLESPACE users

STORAGE (

INITIAL 65536

NEXT 1048576

MINEXTENTS 1

MAXEXTENTS 2147483645

)

NOCACHE

MONITORING

NOPARALLEL

LOGGING

/

-- End of DDL Script for Table BINHTV.DBAMF\_CT\_SESSION

-- Start of DDL Script for Table BINHTV.DBAMF\_CT\_OBJ\_SIZE

-- Generated 5-Nov-2020 16:11:56 from BINHTV@vietdba

-- Drop the old instance of DBAMF\_CT\_OBJ\_SIZE

DROP TABLE binhtv.dbamf\_ct\_obj\_size

/

CREATE TABLE binhtv.dbamf\_ct\_obj\_size

(id NUMBER,

object\_owner VARCHAR2(50 BYTE),

object\_name VARCHAR2(200 BYTE),

size\_gb NUMBER,

event\_date DATE DEFAULT sysdate,

note VARCHAR2(1000 BYTE))

PCTFREE 10

INITRANS 1

MAXTRANS 255

TABLESPACE users

STORAGE (

INITIAL 65536

NEXT 1048576

MINEXTENTS 1

MAXEXTENTS 2147483645

)

NOCACHE

MONITORING

NOPARALLEL

LOGGING

/

-- End of DDL Script for Table BINHTV.DBAMF\_CT\_OBJ\_SIZE

-- Start of DDL Script for Table BINHTV.DBAMF\_CT\_DISK\_USAGE

-- Generated 5-Nov-2020 16:11:52 from BINHTV@vietdba

-- Drop the old instance of DBAMF\_CT\_DISK\_USAGE

DROP TABLE binhtv.dbamf\_ct\_disk\_usage

/

CREATE TABLE binhtv.dbamf\_ct\_disk\_usage

(host\_name VARCHAR2(40 BYTE),

disk\_name VARCHAR2(50 BYTE),

disk\_size VARCHAR2(20 BYTE),

disk\_free VARCHAR2(20 BYTE),

disk\_percen VARCHAR2(10 BYTE),

disk\_mount VARCHAR2(50 BYTE),

db\_ip VARCHAR2(20 BYTE),

event\_date DATE DEFAULT sysdate)

PCTFREE 10

INITRANS 1

MAXTRANS 255

TABLESPACE users

STORAGE (

INITIAL 65536

NEXT 1048576

MINEXTENTS 1

MAXEXTENTS 2147483645

)

NOCACHE

MONITORING

NOPARALLEL

LOGGING

/

-- End of DDL Script for Table BINHTV.DBAMF\_CT\_DISK\_USAGE

-- Start of DDL Script for Table BINHTV.DBAMF\_CPU\_MEM\_USAGE

-- Generated 5-Nov-2020 16:11:45 from BINHTV@vietdba

-- Drop the old instance of DBAMF\_CPU\_MEM\_USAGE

DROP TABLE binhtv.dbamf\_cpu\_mem\_usage

/

CREATE TABLE binhtv.dbamf\_cpu\_mem\_usage

(id NUMBER,

pct\_usage NUMBER,

type VARCHAR2(50 BYTE),

event\_date DATE DEFAULT sysdate,

note VARCHAR2(100 BYTE))

PCTFREE 10

INITRANS 1

MAXTRANS 255

TABLESPACE users

STORAGE (

INITIAL 65536

NEXT 1048576

MINEXTENTS 1

MAXEXTENTS 2147483645

)

NOCACHE

MONITORING

NOPARALLEL

LOGGING

/

-- End of DDL Script for Table BINHTV.DBAMF\_CPU\_MEM\_USAGE

-- Start of DDL Script for Package SYS.DBA\_CT

-- Drop the old instance of DBA\_CT

DROP PACKAGE sys.dba\_ct

/

CREATE OR REPLACE

PACKAGE sys.dba\_ct

IS

procedure ct\_session;

procedure ct\_sql;

procedure ct\_obj\_size;

procedure ct\_user\_size;

procedure ct\_tbs\_size;

END; -- Package spec

/

CREATE OR REPLACE

PACKAGE BODY sys.dba\_ct

IS

free\_space\_low\_level NUMBER := 80000; --2500MB

procedure ct\_session

is

v\_err varchar2(1000);

begin

-- Check 9h, 11h, 15h, 17h, 18h thi chay cau lenh

--if (to\_char(sysdate,'hh24')) in ('08','09','11','15','16','17','18') then

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Starting sys.dba\_ct.ct\_session',1,sysdate,'binhtv.dbamf\_ct\_session');

commit;

insert into binhtv.dbamf\_ct\_session(username ,status,count)

select /\* count , status\*/ username,status, count(\*) from gv$session group by username,status order by count(\*) desc;

commit;

--sys.send\_sms\_binhtv('sys.dba\_ct.ct\_session:' || to\_char(sysdate,'dd/mm/yyyy hh24:mi:ss'));

commit;

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Completed sys.dba\_ct.ct\_session',1,sysdate,'binhtv.dbamf\_ct\_session');

commit;

--end if;

EXCEPTION

WHEN others THEN

--send\_sms\_binhtv('Error sys.dba\_ct.ct\_session:' || SQLERRM);

v\_err := substr(SQLERRM,1,200);

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Error sys.dba\_ct.ct\_session',-1,sysdate,'Error sys.dba\_ct.ct\_session, '||v\_err);

commit;

send\_sms\_binhtv('Error sys.dba\_ct.ct\_session, '||v\_err);

commit;

END;

procedure ct\_sql

is

v\_err varchar2(1000);

begin

-- Check 9h, 11h, 15h, 17h, 18h thi chay cau lenh

--if (to\_char(sysdate,'hh24')) in ('08','09','11','15','17','18') then

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Starting sys.dba\_ct.ct\_sql',1,sysdate,'binhtv.dbamf\_ct\_sql');

commit;

insert into binhtv.dbamf\_ct\_sql(inst\_id ,username,sid,osuser,machine,EVENT,sql\_id,SQL\_TEXT,logon\_time)

SELECT distinct s.inst\_id inst\_id, s.username, s.SID SID, s.osuser, s.machine,DECODE(S.WAIT\_TIME, 0, S.EVENT, 'CPU') EVENT,

s.sql\_id, SUBSTR(DECODE(SS.SQL\_TEXT, NULL, AA.NAME, SS.SQL\_TEXT), 1, 1000) SQL\_TEXT,s.logon\_time

FROM GV$SESSION S, GV$SQLSTATS SS, AUDIT\_ACTIONS AA

WHERE S.STATUS = 'ACTIVE' AND S.SQL\_ID = SS.SQL\_ID (+) AND AA.ACTION = S.COMMAND and s.type != 'BACKGROUND' AND S.TYPE = 'USER'

and s.username NOT in ('SYS','SYSMAN','DBSNMP','GGATE','GOLDENGATE')

--AND username in 'SYS'

--and DECODE(S.WAIT\_TIME, 0, S.EVENT, 'CPU') like '%cell single block physical read%'

--and lower(ss.sql\_text) like lower('%parallel%')

--and s.sid=1234

--and s.machine like '%BINHTV%'

--and s.sql\_id ='ccwg0nqr1zbu7'

ORDER BY username,sql\_id;

commit;

--sys.send\_sms\_binhtv('sys.dba\_ct.ct\_sql' || to\_char(sysdate,'dd/mm/yyyy hh24:mi:ss'));

commit;

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Completed sys.dba\_ct.ct\_sql',1,sysdate,'binhtv.dbamf\_ct\_sql');

commit;

--end if;

EXCEPTION

WHEN others THEN

--send\_sms\_binhtv('Error sys.dba\_ct.ct\_sql: ' || SQLERRM);

v\_err := substr(SQLERRM,1,200);

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Error sys.dba\_ct.ct\_sql',-1,sysdate,'Error sys.dba\_ct.ct\_sql'||v\_err);

commit;

send\_sms\_binhtv('Error sys.dba\_ct.ct\_sql, '||v\_err);

END;

procedure ct\_obj\_size

is

v\_err varchar2(1000);

begin

-- Check 9h, 11h, 15h, 17h, 18h thi chay cau lenh

--if (to\_char(sysdate,'hh24')) in ('08','09','11','15','17','18') then

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Starting sys.dba\_ct.ct\_obj\_size',1,sysdate,'binhtv.dbamf\_obj\_size');

commit;

insert into binhtv.dbamf\_ct\_obj\_size(object\_owner,object\_name,size\_gb)

select owner, segment\_name, round(sum(bytes)/1024/1024/1024,2) from dba\_segments

where segment\_name not like '%$%'

and segment\_name not like '%XX%'

and segment\_name not like '%TMP%'

group by owner, segment\_name

having round(sum(bytes)/1024/1024/1024) >1

order by sum(bytes)desc, owner, segment\_name

;

commit;

--sys.send\_sms\_binhtv('sys.dba\_ct.ct\_obj\_size' || to\_char(sysdate,'dd/mm/yyyy hh24:mi:ss'));

commit;

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Completed sys.dba\_ct.ct\_obj\_size',1,sysdate,'binhtv.dbamf\_obj\_size');

commit;

--end if;

EXCEPTION

WHEN others THEN

v\_err := substr(SQLERRM,1,200);

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Error sys.dba\_ct.ct\_obj\_size',-1,sysdate,'Error sys.dba\_ct.ct\_obj\_size'||v\_err);

commit;

send\_sms\_binhtv('Error sys.dba\_ct.ct\_obj\_size: ' || v\_err);

END;

procedure ct\_user\_size

is

begin

-- Check 9h, 11h, 15h, 17h, 18h thi chay cau lenh

--if (to\_char(sysdate,'hh24')) in ('08','09','11','15','17','18') then

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Starting sys.dba\_ct.ct\_user\_size',1,sysdate,'binhtv.dbamf\_user\_size');

commit;

insert into binhtv.dbamf\_ct\_user\_size(user\_name, size\_mb)

select owner, round(sum(bytes)/1024/1024,2) from dba\_segments

group by owner

order by round(sum(bytes)/1024/1024)desc, owner;

commit;

--sys.send\_sms\_binhtv('sys.dba\_ct.ct\_user\_size' || to\_char(sysdate,'dd/mm/yyyy hh24:mi:ss'));

commit;

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Completed sys.dba\_ct.ct\_user\_size',1,sysdate,'binhtv.dbamf\_user\_size');

commit;

--end if;

EXCEPTION

WHEN others THEN

send\_sms\_binhtv('Error sys.dba\_ct.ct\_user\_size: ' || SQLERRM);

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Error sys.dba\_ct.ct\_user\_size' ,-1,sysdate,'Error sys.dba\_ct.ct\_user\_size' );

commit;

END;

procedure ct\_tbs\_size

is

begin

-- Check 9h, 11h, 15h, 17h, 18h thi chay cau lenh

--if (to\_char(sysdate,'hh24')) in ('08','09','11','15','17','18') then

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Starting sys.dba\_ct.ct\_tbs\_size',1,sysdate,'binhtv.dbamf\_ct\_tbs\_size');

commit;

insert into binhtv.dbamf\_ct\_tbs\_size(tbs\_name, size\_gb)

select tablespace\_name, round(sum(bytes)/1024/1024/1024,2) from dba\_segments

group by tablespace\_name

order by round(sum(bytes)/1024/1024/1024)desc, tablespace\_name;

commit;

--sys.send\_sms\_binhtv('sys.dba\_ct.ct\_tbs\_size' || to\_char(sysdate,'dd/mm/yyyy hh24:mi:ss'));

commit;

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Completed sys.dba\_ct.ct\_tbs\_size',1,sysdate,'binhtv.dbamf\_ct\_tbs\_size');

commit;

--end if;

EXCEPTION

WHEN others THEN

send\_sms\_binhtv('Error sys.dba\_ct.ct\_tbs\_size: ' || SQLERRM);

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Error sys.dba\_ct.ct\_tbs\_size' ,-1,sysdate,'Error sys.dba\_ct.ct\_tbs\_size' );

commit;

END;

END;

/

-- End of DDL Script for Package SYS.DBA\_CT

## DBA\_WN

-- Start of DDL Script for Package SYS.DBA\_WN

-- Drop the old instance of DBA\_WN

DROP PACKAGE sys.dba\_wn

/

CREATE OR REPLACE

PACKAGE sys.dba\_wn

IS

PROCEDURE pro\_active\_load;

PROCEDURE TBS\_WARNING;

END;

/

CREATE OR REPLACE

PACKAGE BODY sys.dba\_wn

IS

time\_retry number:=1.2;

/\*

PROCEDURE fix\_lock\_dblinks

IS

CURSOR c1

IS

SELECT local\_tran\_id

FROM DBA\_2PC\_PENDING

where (retry\_time-fail\_time)\*24\*60>time\_retry AND

STATE='committed'

and db\_user not in ('BINHTVK'); --waiting for longer 4 min

BEGIN

FOR r1 IN c1

LOOP

EXECUTE IMMEDIATE 'begin

dbms\_transaction.purge\_lost\_db\_entry('''

|| r1.local\_tran\_id

|| ''');

commit;

end;';

COMMIT;

END LOOP;

END; --\*/

PROCEDURE pro\_active\_load

IS

nsession NUMBER := 0;

nsession\_all NUMBER := 0;

nlock NUMBER := 0;

nlocklks NUMBER := 0;

mlocklks number := 0;

n\_diskfree number := 0;

nsize\_dbfs number := 0;

v\_err varchar2(1000):='';

BEGIN

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Starting sys.dba\_wn.pro\_active\_load',1,sysdate,'binhtv.dbamf\_log\_jobs, binhtv.dbamf\_log\_wn, binhtv.pending\_trans\_log\_vi');

commit ;

for r1 in (select name,trunc(usable\_file\_mb/1024) gb from v$asm\_diskgroup where name not in('CRS')) loop

if (r1.name='DATA' and r1.gb < 1000) then

--send\_sms\_all('DBAVIETDB, vung DATA con: ' || r1.gb || ' GB');

----send\_sms\_binhtv('DBAVIETDB, vung '||r1.name||' con: ' || r1.gb || ' GB');

insert into binhtv.dbamf\_log\_wn (msg, type, note) values ('DBAVIETDB, vung '||r1.name||' con: ' || r1.gb || ' GB', 'ASM','ASM');

commit;

elsif (r1.name='RECO' and r1.gb < 1000) then

--send\_sms\_all('DBAVIETDB, vung '||r1.name||' con: ' || r1.gb || ' GB');

----send\_sms\_binhtv('DBAVIETDB, vung '||r1.name||' con: ' || r1.gb || ' GB');

insert into binhtv.dbamf\_log\_wn (msg, type, note) values ('DBAVIETDB, vung '||r1.name||' con: ' || r1.gb || ' GB', 'ASM','ASM');

commit;

end if;

end loop;

SELECT COUNT ( \* ) INTO nsession FROM gv$session;

--IF nsession > 6000, comment ngay 20/12/2016

IF nsession > 1000

THEN

--send\_sms\_binhtv('DBAVIETDB, session:' || nsession);

----send\_sms\_binhtv('sys.dba\_mn.check\_pef: ' || nsession);

insert into binhtv.dbamf\_log\_wn (msg, type, note) values ('DBAVIETDB, session:' || nsession, 'nsession','nsession');

commit;

END IF;

SELECT COUNT ( \* ) INTO nsession\_all FROM gv$session where status like 'ACTIVE';

select count(\*) into nsession from (SELECT /\*1.ActiveSession\*/ distinct s.inst\_id i#, s.username, s.SID SID, s.osuser, s.machine,DECODE(S.WAIT\_TIME, 0, S.EVENT, 'CPU') ACTION,

s.sql\_id, SUBSTR(DECODE(SS.SQL\_TEXT, NULL, AA.NAME, SS.SQL\_TEXT), 1, 1000) SQLTEXT,s.logon\_time,s.p1text, S.P1, s.p2text, S.P2, s.p3text, S.P3

FROM GV$SESSION S, GV$SQLSTATS SS, AUDIT\_ACTIONS AA

WHERE S.STATUS = 'ACTIVE' AND S.SQL\_ID = SS.SQL\_ID (+) AND AA.ACTION = S.COMMAND and s.type != 'BACKGROUND' AND S.TYPE = 'USER'

and s.username NOT in ('SYS','SYSMAN','DBSNMP','GGATE','GOLDENGATE'));

IF nsession > 150

--IF nsession > 700

THEN

--send\_sms\_all('DBAVIETDB, active session:' || nsession || '/'||nsession\_all);

----send\_sms\_binhtv('sys.dba\_mn.check\_pef: ' || nsession);

insert into binhtv.dbamf\_log\_wn (msg, type, note) values ('DBAVIETDB, active session:' || nsession|| '/'||nsession\_all, 'nsession','nsession');

commit;

END IF;

----send\_sms\_sos

IF nsession > 200

--IF nsession > 700

THEN

--send\_sms\_sos('DBAVIETDB, active session:' || nsession || '/'||nsession\_all);

----send\_sms\_binhtv('sys.dba\_mn.check\_pef: ' || nsession);

insert into binhtv.dbamf\_log\_wn (msg, type, note) values ('DBAVIETDB, active session:' || nsession|| '/'||nsession\_all, 'nsession','nsession');

commit;

END IF;

SELECT COUNT ( \* ) INTO nlock FROM gv$session

WHERE blocking\_session IS NOT NULL;

--IF (nlock > 10)

IF (nlock > 10)

THEN

--send\_sms\_all('DBAVIETDB, LOCK:' || nlock);

----send\_sms\_binhtv('sys.dba\_mn.check\_pef: ' || nlock);

insert into binhtv.dbamf\_log\_wn (msg, type, note) values ('DBAVIETDB, LOCK:' || nlock, 'nlock','nlock');

commit;

END IF;

SELECT COUNT ( \* ) INTO nlocklks FROM dba\_2pc\_pending

where (retry\_time-fail\_time)\*24\*60>time\_retry

and db\_user not in ('SONTQ','BINHVT','CDR\_OWNER','CUS\_OWNER','CLOSECYCLE','GIANGNT','TRIEUNV','KHUYENMAI','THAMSOHOA','PCAT\_OWNER'); --waiting for longer 4 min;

IF (nlocklks>0 ) then

/\*Ghi log lai cac transaction log\*/

insert into binhtv.pending\_trans\_log\_vi select \* from dba\_2pc\_pending

where (retry\_time-fail\_time)\*24\*60>time\_retry AND STATE='committed';

SELECT COUNT ( \* ) INTO mlocklks FROM dba\_2pc\_pending

where (retry\_time-fail\_time)\*24\*60>time\_retry

and db\_user not in ('SONTQ','BINHVT','CDR\_OWNER','CUS\_OWNER','CLOSECYCLE','GIANGNT','TRIEUNV','KHUYENMAI','THAMSOHOA','INTERCHANGE','PCAT\_OWNER') --waiting for longer 4 min;

AND STATE='committed';

IF (mlocklks>0 ) then

send\_msg('DBAVIETDB, dblink LOCK:' || nlocklks);

insert into binhtv.dbamf\_log\_wn (msg, type, note) values ('DBAVIETDB, dblink LOCK:' || nlocklks, 'lock\_dblink','lock\_dblink');

commit;

null;

END IF;

commit;

sys.dba\_op.fix\_lock\_dblinks;

END IF;

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Completed sys.dba\_wn.pro\_active\_load',1,sysdate,'binhtv.dbamf\_log\_jobs, binhtv.dbamf\_log\_wn, binhtv.pending\_trans\_log\_vi');

commit ;

EXCEPTION

WHEN others THEN

v\_err := substr(SQLERRM,1,200);

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Error sys.dba\_wn.pro\_active\_load',-1,sysdate,'Error sys.dba\_wn.pro\_active\_load, ' ||v\_err);

commit;

--send\_sms\_binhtv('Error sys.dba\_wn.pro\_active\_load, ' ||v\_err);

END;

PROCEDURE TBS\_WARNING

IS

msg VARCHAR2(1000):='';

v\_err varchar2(1000) :='';

CURSOR c\_gold

IS

SELECT a.tablespace\_name , ROUND (b.BYTES / 1024 / 1024, 2) AS mb\_free

FROM (SELECT tablespace\_name, SUM (BYTES) BYTES

FROM dba\_data\_files

GROUP BY tablespace\_name) a,

(SELECT tablespace\_name, SUM (BYTES) BYTES, MAX (BYTES) largest

FROM dba\_free\_space

GROUP BY tablespace\_name) b

WHERE a.tablespace\_name = b.tablespace\_name

AND (lower(a.tablespace\_name) in ('data','indx','undotbs1','undotbs2','regb\_req\_data01','regb\_req\_data02','regb\_req\_data03','regb\_req\_data04','regb\_req\_data05','regb\_req\_data06','data','indx' || to\_char(sysdate,'YYYYMM'),'data' || to\_char(sysdate,'YYYYMM'),'logs' || to\_char(sysdate,'YYYYMM')))

AND (ROUND (b.BYTES / 1024 / 1024, 2) < 500)

ORDER BY ((a.BYTES - b.BYTES) / a.BYTES) DESC;

TYPE tbs\_type IS RECORD (tablespace\_name dba\_data\_files.tablespace\_name%TYPE,

mb\_free dba\_free\_space.bytes%TYPE);

rec tbs\_type;

BEGIN

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Starting sys.dba\_wn.TBS\_WARNING',1,sysdate,'binhtv.dbamf\_log\_jobs');

commit;

OPEN c\_gold;

LOOP

FETCH c\_gold INTO rec.tablespace\_name, rec.mb\_free;

EXIT WHEN c\_gold%NOTFOUND;

--msg:=msg || ';' || rec.tablespace\_name ||':'|| rec.mb\_free;

--dbms\_output.put\_line(rec.tablespace\_name ||':'|| rec.mb\_free);

END loop;

CLOSE c\_gold;

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Completed sys.dba\_wn.TBS\_WARNING',1,sysdate,'binhtv.dbamf\_log\_jobs');

commit;

EXCEPTION

WHEN others THEN

v\_err := substr(SQLERRM,1,200);

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Error sys.dba\_wn.TBS\_WARNING',-1,sysdate,'Error sys.dba\_wn.TBS\_WARNING, ' ||v\_err);

commit;

--send\_sms\_binhtv('Error sys.dba\_wn.TBS\_WARNING, ' ||v\_err);

END;

END;

/

-- End of DDL Script for Package SYS.DBA\_WN

## DBA\_OP

CREATE OR REPLACE PACKAGE SYS.dba\_op

IS

PROCEDURE auto\_drop\_partitions(v\_date date);

PROCEDURE fix\_lock\_dblinks;

PROCEDURE fix\_lock\_dblinks1;

PROCEDURE auto\_extend\_space;

procedure auto\_kill\_direct;

procedure auto\_kill\_sql\_id;

procedure auto\_kill\_lock\_mem;

procedure auto\_kill\_all;

PROCEDURE auto\_gather\_tables ;

PROCEDURE gather\_partitioned\_tables (p\_month DATE);

PROCEDURE gather\_unpartitioned\_tables ;

END; -- Package spec

/

CREATE OR REPLACE PACKAGE BODY SYS.dba\_op

IS

free\_space\_low\_level NUMBER := 50000; --1000MB

p\_error VARCHAR2 (1000);

p\_gather\_date DATE := SYSDATE ;

procedure auto\_drop\_partitions(v\_date date)

is

CURSOR c\_partition

IS --

SELECT table\_owner,table\_name, partition\_name

FROM dba\_tab\_partitions

WHERE (table\_owner='TEST\_OWNER')

and (sysdate - to\_date(SUBSTR(partition\_name,length(partition\_name)-7,8),'yyyy/mm/dd') > 365 AND table\_name IN ('MC\_IN\_LOG\_VIEW','MC\_HLR\_LOG\_VIEW'))

order by 1,2;

v\_sql\_command VARCHAR2 (2400);

--date\_num12t INT := 365; -- Chi luu giu 40 ngay, PROM\_CHARGE\_DAILY\_HIS,..

v\_err varchar2(1000):='';

--v\_date date := sysdate;

BEGIN

-- add partitions

--dbms\_output.put\_line('0');

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Starting sys.dba\_op.auto\_drop\_partitions',1,sysdate,'binhtv.dbamf\_log\_jobs');

commit;

--dbms\_output.put\_line('1');

FOR v\_data IN c\_partition

LOOP

--dbms\_output.put\_line('1.loop');

v\_sql\_command := 'alter table '

|| v\_data.table\_owner ||'.'

|| v\_data.table\_name

|| ' drop partition '

|| v\_data.partition\_name;

--dbms\_output.put\_line(v\_sql\_command);

EXECUTE IMMEDIATE v\_sql\_command;

--dbms\_output.put\_line('1.loop.IMMEDIATE');

--dbms\_output.put\_line(v\_sql\_command);

END LOOP;

--dbms\_output.put\_line('2');

--sys.send\_sms\_binhtv('Completed c\_partition at: ' || sysdate);

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Completed sys.dba\_op.auto\_drop\_partitions',1,sysdate,'binhtv.dbamf\_log\_jobs');

commit;

--dbms\_output.put\_line('3');

send\_sms\_binhtv('Completed sys.dba\_op.auto\_drop\_partitions');

EXCEPTION

WHEN others THEN

send\_sms\_binhtv('Error sys.dba\_op.auto\_drop\_partitions: ' || SQLERRM);

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Error sys.dba\_op.auto\_drop\_partitions',-1,sysdate,'Error sys.dba\_op.auto\_drop\_partitions');

commit;

END;

PROCEDURE fix\_lock\_dblinks

IS

CURSOR c1

IS

SELECT local\_tran\_id

FROM DBA\_2PC\_PENDING

where (retry\_time-fail\_time)\*24\*60>1.5

AND STATE='committed'

and (db\_user not in ('BINHTV')

or (db\_user is null and host like '%KMTD%')); --waiting for longer 4 min

BEGIN

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Starting sys.dba\_op.fix\_lock\_dblinks',1,sysdate,'binhtv.dbamf\_log\_jobs');

commit;

FOR r1 IN c1

LOOP

EXECUTE IMMEDIATE 'begin

dbms\_transaction.purge\_lost\_db\_entry('''

|| r1.local\_tran\_id

|| ''');

commit;

end;';

COMMIT;

END LOOP;

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Completed sys.dba\_op.fix\_lock\_dblinks',1,sysdate,'binhtv.dbamf\_log\_jobs');

commit;

EXCEPTION

WHEN others THEN

send\_sms\_binhtv('Error sys.dba\_op.fix\_lock\_dblinks: ' || SQLERRM);

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Error sys.dba\_op.fix\_lock\_dblinks',-1,sysdate,'Error sys.dba\_op.fix\_lock\_dblinks');

commit;

END;

PROCEDURE fix\_lock\_dblinks1

IS

v\_err varchar2(1000);

CURSOR c1

IS

SELECT local\_tran\_id, state

FROM DBA\_2PC\_PENDING

where (retry\_time-fail\_time)\*24\*60>1.5; --waiting for longer 4 min

BEGIN

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Starting sys.dba\_op.fix\_lock\_dblinks1',1,sysdate,'binhtv.dbamf\_log\_jobs');

/\*FOR r1 IN c1

LOOP

dbms\_output.put\_line('1');

EXECUTE IMMEDIATE 'rollback force '''

|| r1.local\_tran\_id

|| '''';

commit;

end loop;\*/

FOR r2 IN c1

LOOP

if r2.state in ('committed') then

EXECUTE IMMEDIATE 'begin

dbms\_transaction.purge\_lost\_db\_entry('''

|| r2.local\_tran\_id

|| ''');

commit;

end;';

commit;

elsif r2.state='prepared' then

EXECUTE IMMEDIATE 'rollback force '''

|| r2.local\_tran\_id

|| '''';

commit;

EXECUTE IMMEDIATE 'begin

dbms\_transaction.purge\_lost\_db\_entry('''

|| r2.local\_tran\_id

|| ''');

commit;

end;';

commit;

else

EXECUTE IMMEDIATE 'begin

dbms\_transaction.purge\_lost\_db\_entry('''

|| r2.local\_tran\_id

|| ''');

commit;

end;';

commit;

end if;

end loop;

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Completed sys.dba\_op.fix\_lock\_dblinks1',1,sysdate,'binhtv.dbamf\_log\_jobs');

commit;

EXCEPTION

WHEN others THEN

v\_err := substr(SQLERRM,1,200);

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Error sys.dba\_op.fix\_lock\_dblinks1',-1,sysdate,'Error sys.dba\_op.fix\_lock\_dblinks1, ' ||v\_err);

commit;

send\_sms\_binhtv('Error sys.dba\_op.fix\_lock\_dblinks1, ' ||v\_err);

END;

PROCEDURE auto\_extend\_space

IS

CURSOR c\_free\_space -- get tablespace free left 200MB.

IS

/\* SELECT tablespace\_name, SUM (BYTES) / 1024 / 1024

FROM SYS.dba\_free\_space

WHERE (lower(tablespace\_name) in ('regb\_req\_data01','regb\_req\_data06','data','indx','users','indx' || to\_char(sysdate,'YYYY'),'indx' || to\_char(sysdate,'YYYYMM'),'data' || to\_char(sysdate,'YYYY'),'data' || to\_char(sysdate,'YYYYMM'),'mclo' || to\_char(sysdate,'YYYYMM')))

--WHERE tablespace\_name='INDX2014'

GROUP BY tablespace\_name

HAVING SUM (BYTES) / 1024 / 1024 < free\_space\_low\_level;\*/

SELECT

a.tablespace\_name,

ROUND (a.bytes\_alloc / 1024 / 1024) megs\_alloc,

ROUND (NVL (b.bytes\_free, 0) / 1024 / 1024) megs\_free,

ROUND (maxbytes / 1048576) MAX,

ROUND (maxbytes / 1048576)-ROUND ( (a.bytes\_alloc - NVL (b.bytes\_free, 0)) / 1024 / 1024) free\_tbs

FROM ( SELECT f.tablespace\_name,

SUM (f.bytes) bytes\_alloc,

SUM (

DECODE (f.autoextensible,

'YES', f.maxbytes,

'NO', f.bytes))

maxbytes

FROM dba\_data\_files f

GROUP BY tablespace\_name) a,

( SELECT f.tablespace\_name, SUM (f.bytes) bytes\_free

FROM dba\_free\_space f

GROUP BY tablespace\_name) b

WHERE a.tablespace\_name = b.tablespace\_name(+)

AND (lower(a.tablespace\_name) in ('regb\_req\_data01','regb\_req\_data06','data','indx','users','indx' || to\_char(sysdate,'YYYY'),'indx' || to\_char(sysdate,'YYYYMM'),'data' || to\_char(sysdate,'YYYY'),'data' || to\_char(sysdate,'YYYYMM'),'data' || to\_char(sysdate,'YYYY')||'\_rw','indx' || to\_char(sysdate,'YYYY')||'\_rw','mclo' || to\_char(sysdate,'YYYYMM')))

AND ROUND (maxbytes / 1048576)-ROUND ( (a.bytes\_alloc - NVL (b.bytes\_free, 0)) / 1024 / 1024) < free\_space\_low\_level

order by ROUND (maxbytes / 1048576)-ROUND ( (a.bytes\_alloc - NVL (b.bytes\_free, 0)) / 1024 / 1024) desc;

v\_sql varchar2(1000);

msg varchar2(1000);

v\_err varchar2(1000);

BEGIN

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Starting sys.dba\_op.auto\_extend\_space',1,sysdate,'binhtv.dbamf\_log\_jobs');

commit;

for v\_free\_space in c\_free\_space

loop

begin

v\_sql:='ALTER TABLESPACE '

|| v\_free\_space.tablespace\_name

|| ' ADD DATAFILE size 8G autoextend OFF';

EXECUTE IMMEDIATE v\_sql;

--dbms\_output.put\_line('sql: ' || v\_sql);

send\_sms\_binhtv('Card: v\_sql ' || v\_sql);

end;

end loop;

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Completed sys.dba\_op.auto\_extend\_space',1,sysdate,'binhtv.dbamf\_log\_jobs');

commit;

--send\_sms\_binhtv('Completed sys.dba\_op.auto\_extend\_space');

EXCEPTION

WHEN others THEN

v\_err := substr(SQLERRM,1,200);

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Error sys.dba\_op.auto\_extend\_space',-1,sysdate,'Error sys.dba\_op.auto\_extend\_space, ' ||v\_err);

commit;

send\_sms\_binhtv('Error sys.dba\_op.auto\_extend\_space, ' ||v\_err);

END;

-- Job kill tien trinh quet full chiem nhieu IO

procedure auto\_kill\_direct

is

v\_err varchar2(1000);

cursor c1 is SELECT 'ALTER SYSTEM KILL SESSION '''||s.sid||','||s.serial#||',@'||s.inst\_id||''' immediate'sqltext

FROM GV$SESSION S, GV$SQLSTATS SS, AUDIT\_ACTIONS AA

WHERE s.type != 'BACKGROUND' AND S.STATUS = 'ACTIVE' AND S.SQL\_ID = SS.SQL\_ID (+) AND AA.ACTION = S.COMMAND AND S.TYPE = 'USER'

AND s.sql\_id <> (select sql\_id from v$session where sid=(select sid from v$mystat where rownum=1)) and username NOT in ('SYS','BINHVT')

and DECODE(S.WAIT\_TIME, 0, S.EVENT, 'CPU') IN ('direct path read','db file scattered read','PX Deq Credit: send blkd') ;

nsession number;

begin

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Starting sys.dba\_op.auto\_kill\_direct',1,sysdate,'binhtv.dbamf\_log\_jobs');

commit;

SELECT count(\*) into nsession

FROM GV$SESSION S, GV$SQLSTATS SS, AUDIT\_ACTIONS AA

WHERE s.type != 'BACKGROUND' AND S.STATUS = 'ACTIVE' AND S.SQL\_ID = SS.SQL\_ID (+) AND AA.ACTION = S.COMMAND AND S.TYPE = 'USER'

AND s.sql\_id <> (select sql\_id from v$session where sid=(select sid from v$mystat where rownum=1)) and username not in ('SYS','SYSMAN');

--IF nsession>150 THEN

IF nsession>300 THEN

for r1 in c1 loop

begin

execute immediate r1.sqltext;

exception

when others then null;

end;

end loop;

end if;

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Completed sys.dba\_op.auto\_kill\_direct',1,sysdate,'binhtv.dbamf\_log\_jobs');

commit;

EXCEPTION

WHEN others THEN

v\_err := substr(SQLERRM,1,200);

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Error sys.dba\_op.auto\_kill\_direct',-1,sysdate,'Error sys.dba\_op.auto\_kill\_direct, ' ||v\_err);

commit;

send\_sms\_binhtv('Error sys.dba\_op.auto\_kill\_direct, ' ||v\_err);

END;

-- Job kill cac cau lenh bat thuong gay tang tai

procedure auto\_kill\_sql\_id

is

cursor c2 is SELECT 'ALTER SYSTEM KILL SESSION '''||s.sid||','||s.serial#||',@'||s.inst\_id||''' immediate'sqltext

FROM GV$SESSION S, GV$SQLSTATS SS, AUDIT\_ACTIONS AA

WHERE s.type != 'BACKGROUND' AND S.STATUS = 'ACTIVE' AND S.SQL\_ID = SS.SQL\_ID (+) AND AA.ACTION = S.COMMAND AND S.TYPE = 'USER'

AND s.sql\_id <> (select sql\_id from v$session where sid=(select sid from v$mystat where rownum=1))

and s.sql\_id in ('dqb9hg546pk2c','8yun477ph5333','9m4fha58ngchf','77v3nwwph3pdv','2x2qbhtvr6yu2','gydasbrnsyxa1','ckj83ua4gk847','b49mktdg1ddsg','azxdzad5pku35','b49mktdg1ddsg','1xxb5zghqfnnd','dqb9hg546pk2c','azxdzad5pku35','bm72hcznbd433','6z9rabgwm8v70','562xv25t83b2u','drgbfu8y0fs3v','4c24awr2jkv1p','guz6xw11t4f6s','6z9rabgwm8v70','147z1raw5hsfq','6nxb8p7s91gs9','592x919qtvz06','bt05z7yfpyxk6','bgp6ja1spjrh3','g2c9zn8rqwwcf','bt05z7yfpyxk6','dqb9hg546pk2c','2f6jtz620h7m4','1fkjrpkyp5cgw','7470q698pzjs9','9k7fx6x81qvsb','gf5tujm2aa9dh','1uu005ppxhsqw','2kxybw5awdvfn','cgzc3ut2nrhff','gbmyfjjdcyk75','2541vzdnptncb')

and s.username NOT in ('SYS') ;

nsession number;

v\_err varchar2(1000);

begin

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Starting sys.dba\_op.auto\_kill\_sql\_id',1,sysdate,'binhtv.dbamf\_log\_jobs');

commit;

SELECT count(\*) into nsession

FROM GV$SESSION S, GV$SQLSTATS SS, AUDIT\_ACTIONS AA

WHERE s.type != 'BACKGROUND' AND S.STATUS = 'ACTIVE' AND S.SQL\_ID = SS.SQL\_ID (+) AND AA.ACTION = S.COMMAND and s.type != 'BACKGROUND' AND S.TYPE = 'USER'

AND s.sql\_id <> (select sql\_id from v$session where sid=(select sid from v$mystat where rownum=1)) and username NOT in ('SYS','SYSMAN','DBSNMP','GGATE','GOLDENGATE');

--IF nsession>300 THEN: cu

-- Cap nhat ngay 27/08/2020

--IF nsession>40 THEN

for r2 in c2 loop

begin

execute immediate r2.sqltext;

exception

when others then null;

end;

end loop;

--end if;

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Completed sys.dba\_op.auto\_kill\_sql\_id',1,sysdate,'binhtv.dbamf\_log\_jobs');

commit;

EXCEPTION

WHEN others THEN

v\_err := substr(SQLERRM,1,200);

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Error sys.dba\_op.auto\_kill\_sql\_id',-1,sysdate,'Error sys.dba\_op.auto\_kill\_sql\_id, ' ||v\_err);

commit;

send\_sms\_binhtv('Error sys.dba\_op.auto\_kill\_sql\_id, ' ||v\_err);

END;

-- Job kill cac cau lenh gay lock

procedure auto\_kill\_lock\_mem

is

v\_err varchar2(1000);

cursor c1 is SELECT 'ALTER SYSTEM KILL SESSION '''||s.sid||','||s.serial#||',@'||s.inst\_id||''' immediate;' sqltext

FROM GV$SESSION S, GV$SQLSTATS SS, AUDIT\_ACTIONS AA

WHERE s.type != 'BACKGROUND' AND S.STATUS = 'ACTIVE' AND S.SQL\_ID = SS.SQL\_ID (+) AND AA.ACTION = S.COMMAND AND S.TYPE = 'USER' and s.username not in ('SYS','SYSMAN','BINHVT')

and s.event in ('library cache lock','gc buffer busy acquire')

and s.username NOT in ('SYS');

nsession number;

begin

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Starting sys.dba\_op.auto\_kill\_lock\_mem',1,sysdate,'binhtv.dbamf\_log\_jobs');

commit;

SELECT count(\*) into nsession

FROM GV$SESSION S, GV$SQLSTATS SS, AUDIT\_ACTIONS AA

WHERE s.type != 'BACKGROUND' AND S.STATUS = 'ACTIVE' AND S.SQL\_ID = SS.SQL\_ID (+) AND AA.ACTION = S.COMMAND AND S.TYPE = 'USER'

AND s.sql\_id <> (select sql\_id from v$session where sid=(select sid from v$mystat where rownum=1)) and username not in ('SYS','SYSMAN');

IF nsession>300 THEN

for r1 in c1 loop

begin

execute immediate r1.sqltext;

exception

when others then null;

end;

end loop;

end if;

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Completed sys.dba\_op.auto\_kill\_lock\_mem',1,sysdate,'binhtv.dbamf\_log\_jobs');

commit;

EXCEPTION

WHEN others THEN

v\_err := substr(SQLERRM,1,200);

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Error sys.dba\_op.auto\_extend\_space',-1,sysdate,'Error sys.dba\_op.auto\_extend\_space, ' ||v\_err);

commit;

send\_sms\_binhtv('Error sys.dba\_op.auto\_extend\_space, ' ||v\_err);

END;

procedure auto\_kill\_all

is

v\_err VARCHAR2(1000);

cursor c1 is SELECT 'ALTER SYSTEM KILL SESSION '''||s.sid||','||s.serial#||',@'||s.inst\_id||''' immediate' sqltext

FROM GV$SESSION S, GV$SQLSTATS SS, AUDIT\_ACTIONS AA

WHERE s.type != 'BACKGROUND' AND S.STATUS = 'ACTIVE' AND S.SQL\_ID = SS.SQL\_ID (+) AND AA.ACTION = S.COMMAND AND S.TYPE = 'USER'

AND s.sql\_id <> (select sql\_id from v$session where sid=(select sid from v$mystat where rownum=1)) and s.USERNAME not in ('SYS','SYSTEM','GGATE','GOLDENGATE','BINHVT');

nsession number;

begin

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Starting sys.dba\_op.auto\_kill\_all',1,sysdate,'binhtv.dbamf\_log\_jobs');

commit;

select count(\*) into nsession from gv$session s where s.status='ACTIVE' group by status ;

IF nsession > 600 THEN

for r1 in c1 loop

begin

execute immediate r1.sqltext;

exception

when others then null;

end;

end loop;

end if;

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Completed sys.dba\_op.auto\_kill\_all',1,sysdate,'binhtv.dbamf\_log\_jobs');

commit;

EXCEPTION

WHEN others THEN

v\_err := substr(SQLERRM,1,200);

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Error sys.dba\_op.auto\_extend\_space',-1,sysdate,'Error sys.dba\_op.auto\_extend\_space, ' ||v\_err);

commit;

send\_sms\_binhtv('Error sys.dba\_op.auto\_extend\_space, ' ||v\_err);

END;

PROCEDURE auto\_gather\_tables

IS

BEGIN

send\_sms\_binhtv('TEST\_OWNER: Gather Starting at ' || to\_char(sysdate,'dd/mm/yyyy hh24:mm:ss'));

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Starting sys.dba\_op.auto\_gather\_tables',1,sysdate,'binhtv.dbamf\_log\_jobs');

commit;

gather\_partitioned\_tables (TRUNC (SYSDATE));

cdr\_monitor.send\_email\_html@vms2('tranbinh48ca@gmail.vn','TEST\_OWNER Completed gathering partitioned\_tables', 'TEST\_OWNER Completed gathering partitioned\_tables at ' || to\_char(sysdate,'dd/mm/yyy hh24:mm:ss'));

IF (TO\_NUMBER (TO\_CHAR (SYSDATE, 'dd')) IN (8, 18, 28) and (TO\_NUMBER (TO\_CHAR (SYSDATE, 'hh24')) < 8 or TO\_NUMBER (TO\_CHAR (SYSDATE, 'hh24')) > 21 ))

THEN

gather\_unpartitioned\_tables;

binhtv.send\_email\_html('tranbinh48ca@gmail.vn','TEST\_OWNER Completed gathering unpartitioned\_tables', 'TEST\_OWNER Completed gathering unpartitioned\_tables at ' || to\_char(sysdate,'dd/mm/yyy hh24:mm:ss'));

END IF;

send\_sms\_binhtv('TEST\_OWNER: Gather success at ' || to\_char(sysdate,'dd/mm/yyyy hh24:mm:ss'));

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Completed sys.dba\_op.auto\_gather\_tables',1,sysdate,'binhtv.dbamf\_log\_jobs');

commit;

--null;

EXCEPTION

WHEN others THEN

send\_sms\_binhtv('Error sys.dba\_op.auto\_gather\_tables: ' || SQLERRM);

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Error sys.dba\_op.exec\_gather\_tables\_cus',-1,sysdate,'Error sys.dba\_op.auto\_gather\_tables');

commit;

END;

PROCEDURE gather\_partitioned\_tables (p\_month DATE)

IS

CURSOR c\_partitioned\_tables

IS

SELECT table\_name, partition\_name, last\_analyzed,

'ANALYZE TABLE '

|| table\_owner

|| '.'

|| table\_name

|| ' partition ('

|| partition\_name

|| ') ESTIMATE STATISTICS SAMPLE 10 PERCENT'

script

FROM dba\_tab\_partitions

WHERE (( (partition\_name LIKE

'DATA' || TO\_CHAR (p\_month, 'YYYYMM') ||'%'

AND LENGTH(partition\_name)>10

AND TO\_DATE(SUBSTR(partition\_name,5,8),'YYYYMMDD')>=SYSDATE-7

AND TO\_DATE(SUBSTR(partition\_name,5,8),'YYYYMMDD')<SYSDATE+3

)

OR partition\_name =

'DATA' || TO\_CHAR (p\_month, 'YYYY')

) or table\_name in ('MC\_SUBSCRIBER'))

AND table\_name NOT LIKE '%$%'

AND table\_owner IN ('TEST\_OWNER')

AND (NVL (last\_analyzed, SYSDATE - 15) < p\_gather\_date OR num\_rows=0)

ORDER BY table\_name, last\_analyzed, partition\_name;

/\*

SELECT table\_name, partition\_name, last\_analyzed,

'begin

dbms\_stats.gather\_table\_stats

(ownname=>''' || TABLE\_OWNER || ''',

tabname=>''' || table\_name || ''',

partname=>''' || partition\_name || ''',

estimate\_percent => DBMS\_STATS.AUTO\_SAMPLE\_SIZE,

method\_opt => ''FOR ALL COLUMNS SIZE AUTO'',

cascade=>true,

degree=>10);

end;

' script

FROM all\_tab\_partitions

WHERE ((length(partition\_name) = 12 AND to\_date(substr(partition\_name,5,8),'YYYYMMDD')<sysdate

AND to\_date(substr(partition\_name,5,8),'YYYYMMDD')>sysdate-5)

or (length(partition\_name) =10 AND to\_date(substr(partition\_name,5,6),'YYYYMM')<sysdate

AND to\_date(substr(partition\_name,5,6),'YYYYMM')>=add\_months(trunc(sysdate,'month'),-1))

or (length(partition\_name) =8 AND to\_date(substr(partition\_name,5,4),'YYYY')<sysdate

AND to\_date(substr(partition\_name,5,4),'YYYY')>=trunc(sysdate,'year'))

)

AND table\_name NOT LIKE '%$%'

AND table\_owner IN ('TEST\_OWNER')

AND (NVL (last\_analyzed, SYSDATE - 15) < p\_gather\_date OR num\_rows=0)

ORDER BY table\_name, last\_analyzed, partition\_name;

\*/

v\_table\_name VARCHAR2 (100);

v\_par\_name VARCHAR2 (100);

BEGIN

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Starting sys.dba\_op.gather\_partitioned\_tables',1,sysdate,'binhtv.dbamf\_log\_jobs');

commit;

FOR v\_partitioned\_tables IN c\_partitioned\_tables

LOOP

v\_table\_name := v\_partitioned\_tables.table\_name;

v\_par\_name := v\_partitioned\_tables.partition\_name;

if (TO\_NUMBER (TO\_CHAR (SYSDATE, 'hh24')) < 8 or TO\_NUMBER (TO\_CHAR (SYSDATE, 'hh24')) > 21 )

then

EXECUTE IMMEDIATE v\_partitioned\_tables.script;

end if;

END LOOP;

--EXECUTE IMMEDIATE 'analyze table mc\_subscriber estimate statistics sample 10 percent';

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Completed sys.dba\_op.gather\_partitioned\_tables',1,sysdate,'binhtv.dbamf\_log\_jobs');

commit;

--null;

EXCEPTION

WHEN others THEN

send\_sms\_binhtv('Error sys.dba\_op.auto\_gather\_tables: ' || SQLERRM);

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Error sys.dba\_op.gather\_partitioned\_tables',-1,sysdate,'Error sys.dba\_op.gather\_partitioned\_tables');

commit;

END;

-- Gather stats for unpartitioned tables

-- Input:

-- p\_Owner: Owner of tables

-- p\_Ignored\_Tables\_List: List of Tables that will not be gathered

-- Output

-- p\_Gathered\_Tables: Tables already gathered

-- p\_Error: Errors descriptions(if any), NULL value means no error occurred

PROCEDURE gather\_unpartitioned\_tables

IS

CURSOR c\_tables

IS

SELECT owner, table\_name, last\_analyzed,

'begin

dbms\_stats.gather\_table\_stats

(ownname => '''

|| owner

|| ''',

tabname => '''

|| table\_name

|| ''',

cascade => true,

estimate\_percent => 10,

degree => 10);

end;

'

/\* 'begin

dbms\_stats.gather\_table\_stats

(ownname=>''' || OWNER || ''',

tabname=>''' || table\_name || ''',

estimate\_percent => DBMS\_STATS.AUTO\_SAMPLE\_SIZE,

method\_opt => ''FOR ALL COLUMNS SIZE AUTO'',

cascade=>true,

degree=>10);

end;

'

\*/ script

FROM dba\_tables

WHERE partitioned = 'NO'

AND owner IN ('TEST\_OWNER')

and table\_name not like 'XX%'

AND tablespace\_name IS NOT NULL

AND table\_name NOT LIKE '%$%'

AND NVL (last\_analyzed, SYSDATE - 15) < p\_gather\_date

ORDER BY last\_analyzed;

v\_table\_name VARCHAR2 (100);

BEGIN

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Starting sys.dba\_op.gather\_unpartitioned\_tables',1,sysdate,'binhtv.dbamf\_log\_jobs');

commit;

--TEST\_OWNER.send\_sms ('902912888', 'TEST\_OWNER: Gather unpartitioned\_tables started at ' || to\_char(sysdate,'dd/mm/yyyy hh24:mm:ss'));

FOR v\_tables IN c\_tables

LOOP

v\_table\_name := v\_tables.table\_name;

if (TO\_NUMBER (TO\_CHAR (SYSDATE, 'hh24')) < 8 or TO\_NUMBER (TO\_CHAR (SYSDATE, 'hh24')) > 21 )

then

EXECUTE IMMEDIATE v\_tables.script;

end if;

END LOOP;

--TEST\_OWNER.send\_sms ('902912888', 'TEST\_OWNER: Gather unpartitioned\_tables successed at ' || to\_char(sysdate,'dd/mm/yyyy hh24:mm:ss'));

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Completed sys.dba\_op.gather\_unpartitioned\_tables',1,sysdate,'binhtv.dbamf\_log\_jobs');

commit;

--null;

EXCEPTION

WHEN others THEN

send\_sms\_binhtv('Error sys.dba\_op.gather\_unpartitioned\_tables: ' || SQLERRM);

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Error sys.dba\_op.gather\_unpartitioned\_tables',-1,sysdate,'Error sys.dba\_op.gather\_unpartitioned\_tables');

commit;

END;

END;

/

## DBA\_RP

CREATE OR REPLACE PACKAGE SYS.dba\_rp

IS

PROCEDURE hc\_all;

PROCEDURE hc\_sms;

PROCEDURE session\_sms;

PROCEDURE hc\_storage;

PROCEDURE hc\_backup;

PROCEDURE hc\_du;

PROCEDURE hc\_os;

--PROCEDURE hc\_ddl;

procedure hc\_ddl\_log;

procedure hc\_fga\_log;

END; -- Package spec

/

CREATE OR REPLACE PACKAGE BODY SYS.dba\_rp

IS

-- Ngay chay 2 lan 8h, 16h cac thong tin: 1.Tai DB, 2.Backup, 3.Storage, 3.1.User su dung, 4.Object invalid, 5.Index Non-partition, parttion Unuable, 6.Archive log: Hoi it

--

procedure hc\_all

is

tAll varchar2(32700):='';

tAll\_New varchar2(10000):='';

tDBName varchar2(20):='';

v\_err varchar2(1000):='';

--1.Tai DB

nActiveSession number;

nTotalActiveSession number; -- Ca background

nTotalInactiveSession number;

nTotalSession number;

nLock number;

nLockDBLink number;

CURSOR cTotalUserSession

IS

select /\* count , status\*/ username,status, count(\*) num from gv$session group by username,status

having (count(\*)>400 and status='INACTIVE') or (count(\*)>20 and status='ACTIVE')

order by status,count(\*) desc,username;

tTotalUserSession varchar2(30000):='';

cursor c\_cpu\_ram

is

select event\_date, duration, server\_name, cpu\_idle, round(ram\_free\_k/1024/1024,0) "GB"

from BINHTV.dbamf\_ct\_vmstat

where event\_date > sysdate-1/24

and (cpu\_idle <= 20 or round(ram\_free\_k/1024/1024,0) <=70)

order by event\_date desc;

v\_cpu\_ram varchar2(30000):='';

--2.Backup

CURSOR cBackup

IS

select command\_id, start\_time, end\_time, status,INPUT\_TYPE, input\_bytes\_display, output\_bytes\_display, time\_taken\_display, round(compression\_ratio,0) RATIO , input\_bytes\_per\_sec\_display, output\_bytes\_per\_sec\_display

from v$rman\_backup\_job\_details

where trunc(end\_time)>=trunc(sysdate-7)

order by end\_time desc;

tBackup varchar2(30000):='';

--3.Storage

cursor cASM is

select name, state, type, round(total\_mb/1024) total\_gb, round(usable\_file\_mb/1024) usable\_file\_gb from v$asm\_diskgroup;

tASM varchar2(30000) := '';

cursor cTBS is

SELECT a.tablespace\_name,100 - ROUND ( (NVL (b.bytes\_free, 0) / a.bytes\_alloc) \* 100) pct\_usage, ROUND(a.bytes\_alloc / 1024 / 1024/1024) size\_gb, ROUND (NVL (b.bytes\_free, 0) / 1024 / 1024/1024) free\_gb,

(ROUND (a.bytes\_alloc / 1024 / 1024/1024)- ROUND (NVL (b.bytes\_free, 0) / 1024 / 1024/1024)) used\_gb, ROUND ( (NVL (b.bytes\_free, 0) / a.bytes\_alloc) \* 100) pct\_free, ROUND (maxbytes / 1048576/1024) max\_gb,

ROUND (ROUND ( (a.bytes\_alloc - NVL (b.bytes\_free, 0)) / 1024 / 1024)/ ROUND (maxbytes / 1048576) \* 100) as pct\_used\_max

FROM ( SELECT f.tablespace\_name, SUM (f.bytes) bytes\_alloc, SUM (DECODE (f.autoextensible, 'YES', f.maxbytes, 'NO', f.bytes)) maxbytes

FROM dba\_data\_files f

GROUP BY tablespace\_name) a,

( SELECT f.tablespace\_name, SUM (f.bytes) bytes\_free FROM dba\_free\_space f GROUP BY tablespace\_name) b

WHERE a.tablespace\_name = b.tablespace\_name(+)

--and a.tablespace\_name in ('DATA','INDX')

and ROUND (ROUND ( (a.bytes\_alloc - NVL (b.bytes\_free, 0)) / 1024 / 1024)/ ROUND (maxbytes / 1048576) \* 100) > 90

order by pct\_used\_max desc;

tTBS varchar2(30000) := '';

cursor c\_user\_usage is

select owner,round(sum(bytes)/1024/1024/1024,0) "GB" from dba\_segments

group by owner

having round(sum(bytes)/1024/1024/1024,0) > 50

order by "GB" desc;

v\_user\_usage varchar2(30000) := '';

cursor c\_object\_usage is

select owner,segment\_name,round(sum(bytes)/1024/1024/1024,0) "GB" from dba\_segments

group by owner, segment\_name

having round(sum(bytes)/1024/1024/1024,0) > 200

order by "GB" desc;

v\_object\_usage varchar2(3000) := '';

-- Cac tablespace Read Only:Chay lau

/\* cursor cTBS\_RO is

select tablespace\_name, round(sum(bytes)/1024/1024/1024,0) "GB"

from dba\_segments

where tablespace\_name in (select name from v$tablespace where ts# in (select ts# from v$datafile where enabled='READ ONLY'))

group by tablespace\_name

order by tablespace\_name ;

tStorage varchar2(4000):='';\*/

cursor c\_du

is

select \* from binhtv.dbamf\_ct\_disk\_usage

where event\_date>sysdate-1

and nvl(substr(disk\_percen,-3,2),0)>70

order by disk\_percen desc;

v\_du varchar2(30000):='';

--4.Object invalid

nInvalidObject number;

--5.Index Non-partition, parttion Unuable

nUnuableIndNonPar number;

nUnuableIndPar number;

--6.Archive log

-- Theo doi archived log sinh ra

cursor cTotalArchivedLog is

select trunc(completion\_time), round(sum(blocks\*block\_size)/1024/1024/1024,0) "Archived Log GB" from V$ARCHIVED\_LOG

where trunc(completion\_time) >= trunc(sysdate-1)

--and trunc(completion\_time)>= to\_date(trunc(sysdate),'dd/mm/yyyy')

and dest\_id=1

group by trunc(completion\_time)

order by trunc(completion\_time) desc;

-- Archived log sinh ra theo gio

cursor cHourArchivedLog is

select to\_char(next\_time,'YYYY-MM-DD hh24') Hour, round(sum(size\_in\_byte)/1024/1024,0) as size\_in\_mb, count(\*) log\_switch from (

select thread# ,sequence#, FIRST\_CHANGE#,blocks\*BLOCK\_SIZE as size\_in\_byte, next\_time

from v$archived\_log where name is not null and completion\_time>sysdate-1 group by thread# ,sequence#, FIRST\_CHANGE#,blocks\*BLOCK\_SIZE, next\_time)

group by to\_char(next\_time,'YYYY-MM-DD hh24') order by 1 desc;

tArchivedLog varchar2(30000):='';

--7.Gather

tGather varchar2(4000):='';

tGatherTabPar varchar2(100) :='';

tGatherIndPar varchar2(100) :='';

tGatherTabNonPar varchar2(100) :='';

tGatherIndNonPar varchar2(100) :='';

BEGIN

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Starting sys.dba\_rp.hc\_all',1,sysdate,'sys.dba\_rp.hc\_all');

commit;

--dbms\_output.put\_line('Before if');

--if (to\_char(sysdate,'hh24') in ('00','08','13')) then

--dbms\_output.put\_line('After if');

select name into tDBName from v$database;

tAll := tAll ||'<h2>BAO CAO CSDL ' || tDBName ||' NGAY ' || to\_char(sysdate,'dd/mm/yyyy hh24:mm:ss') ||'</h2>';

--1.TAI DB

select count(\*) into nActiveSession from gv$session where username NOT in ('SYS','SYSMAN','DBSNMP','GGATE','DBAVIETENGATE') and status='ACTIVE';

select count(\*) into nTotalActiveSession from gv$session where status='ACTIVE';

select count(\*) into nTotalInactiveSession from gv$session where status='INACTIVE';

select count(\*) into nTotalSession from gv$session;

SELECT COUNT(\*) INTO nLock FROM gv$session WHERE blocking\_session IS NOT NULL;

sELECT COUNT ( \* ) INTO nLockDBLink FROM dba\_2pc\_pending where (retry\_time-fail\_time)\*24\*60>2;

tAll := tAll|| '<br><b>SUMMARY SESSION:</b><table width="600" border ="1" cellspacing="1" cellpadding="1"><tr bgcolor="#FFF1D9" style="font-weight:bold"><td>nActiveSession</td><td>nTotalActiveSession</td><td>nTotalInactiveSession</td><td>nTotalSession</td><td>nLock</td><td>nLockDBLink</td></tr>';

if nActiveSession > 200 then

tAll := tAll||'<tr>

<td><font color="red"><b>'||nActiveSession||'</b></font></td>

<td>'||nTotalActiveSession||'</td>

<td>'||nTotalInactiveSession||'</td>

<td>'||nTotalSession||'</td>

<td>'||nLock||'</td>

<td>'||nLockDBLink||'</td>

</tr>'||'</table>';

else

tAll := tAll||'<tr>

<td>'||nActiveSession||'</td>

<td>'||nTotalActiveSession||'</td>

<td>'||nTotalInactiveSession||'</td>

<td>'||nTotalSession||'</td>

<td>'||nLock||'</td>

<td>'||nLockDBLink||'</td>

</tr>'||'</table>';

end if;

--1.3.Total session theo tung user

tTotalUserSession :=tTotalUserSession||'

<br><b>USER SESSIONS DETAIL</b>:

<table width="600" border ="1" cellspacing="1" cellpadding="1">

<tr bgcolor="#FFF1D9" style="font-weight:bold">

<td>UserName</td>

<td>Status</td>

<td>Total Sesion</td>

</tr>';

FOR r1 IN cTotalUserSession

LOOP

tTotalUserSession:=tTotalUserSession||'<tr>

<td>'||r1.username||'</td>

<td>'||r1.status||' </td>

<td>'||r1.num||'</td>

</tr>';

END LOOP;

tTotalUserSession:=tTotalUserSession||'</table>';

tAll :=tAll || tTotalUserSession;

--cdr\_monitor.send\_email\_html\_m('binhtv@mobifone.vn;tranbinh48ca@gmail.com','DBA\_BILL\_DBA MONITOR\_' || sysdate, tAll);

dbms\_output.put\_line('1');

--CPU,RAM

v\_cpu\_ram:=v\_cpu\_ram||'<br><b>CPU, RAM USAGE:</b><table width="600" border ="1" cellspacing="1" cellpadding="1">

<tr bgcolor="#FFF1D9" style="font-weight:bold">

<td> event\_date</td>

<td> duration</td>

<td> server\_name</td>

<td> cpu\_idle</td>

<td> GB</td>

</tr>';

for r11 in c\_cpu\_ram loop

v\_cpu\_ram:=v\_cpu\_ram||'<tr>

<td>'||to\_char(r11.event\_date,'dd/mm/yyyy hh24:mm:ss')||'</td>

<td>'||r11.duration||'</td>

<td>'||r11.server\_name||'</td>

<td>'||r11.cpu\_idle||'</td>

<td>'||r11.GB||'</td>

</tr>' ;

end loop;

v\_cpu\_ram := v\_cpu\_ram || '</table>';

tAll := tAll || v\_cpu\_ram;

--dbms\_output.put\_line('2');

--2.BACKUP: Lay 1 tuan

tBackup:=tBackup||'<br><b>SUMMARY BACKUP:</b><table width="600" border ="1" cellspacing="1" cellpadding="1">

<tr bgcolor="#FFF1D9" style="font-weight:bold">

<td>command\_id</td>

<td>start\_time</td>

<td>end\_time</td>

<td>status</td>

<td>INPUT\_TYPE</td>

<td>input\_bytes\_display</td>

<td>output\_bytes\_display</td>

<td>time\_taken\_display</td>

<td>RATIO</td>

<td>input\_bytes\_per\_sec\_display</td>

<td>output\_bytes\_per\_sec\_display</td>

</tr>';

for r1 in cBackup loop

tBackup:=tBackup||'<tr>

<td>'||r1.command\_id||'</td>

<td>'||to\_char(r1.start\_time,'dd/mm/yyyy hh24:mm:ss')||'</td>

<td>'||to\_char(r1.end\_time,'dd/mm/yyyy hh24:mm:ss')||' </td>

<td>'||r1.status||' </td>

<td>'||r1.INPUT\_TYPE||'</td>

<td>'||r1.input\_bytes\_display||'</td>

<td>'||r1.output\_bytes\_display||'</td>

<td>'||r1.time\_taken\_display||'</td>

<td>'||r1.RATIO||'</td>

<td>'||r1.input\_bytes\_per\_sec\_display||'</td>

<td>'||r1.output\_bytes\_per\_sec\_display||'</td>

</tr>' ;

end loop;

tBackup := tBackup || '</table>';

tAll := tAll || tBackup;

--dbms\_output.put\_line('2');

--3.STORAGE

tASM :='<br><b>ASM DISKGROUP:</b><table width="600" border ="1" cellspacing="1" cellpadding="1">

<tr bgcolor="#FFF1D9" style="font-weight:bold">

<td>name</td>

<td>state</td>

<td>type</td>

<td>total\_gb</td>

<td>usable\_file\_gb</td>

</tr>';

for r2 in cASM loop

tASM:=tASM||'<tr>

<td>'||r2.name||'</td>

<td>'||r2.state||'</td>

<td>'||r2.type||'</td>

<td>'||r2.total\_gb||'</td>

<td>'||r2.usable\_file\_gb||'</td>

</tr>' ;

end loop;

tASM := tASM || '</table><br>';

tAll := tAll || tASM;

--dbms\_output.put\_line('3.tASM');

tTBS :='<br><b>DUNG LUONG TABLESPACE:</b><table width="600" border ="1" cellspacing="1" cellpadding="1">

<tr bgcolor="#FFF1D9" style="font-weight:bold">

<td> Tablespace Name </td>

<td> %Usage </td>

<td> Size GB </td>

<td> Free GB </td>

<td> Used GB </td>

<td> %Free </td>

<td> Max GB </td>

<td> %Used Max </td>

</tr>';

--dbms\_output.put\_line('0');

for r3 in cTBS loop

if r3.pct\_used\_max > 90 then

tTBS:=tTBS||'<tr>

<td><font color="red"><b>'||r3.tablespace\_name||'</b></font></td>

<td>'||r3.pct\_usage||'</td>

<td>'||r3.size\_gb||'</td>

<td>'||r3.free\_gb||'</td>

<td>'||r3.used\_gb||'</td>

<td>'||r3.pct\_free||'</td>

<td>'||r3.max\_gb||'</td>

<td><font color="red"><b>'||r3.pct\_used\_max||'</b></font></td>

</tr>' ;

else

tTBS:=tTBS||'<tr>

<td>'||r3.tablespace\_name||'</td>

<td>'||r3.pct\_usage||'</td>

<td>'||r3.size\_gb||'</td>

<td>'||r3.free\_gb||'</td>

<td>'||r3.used\_gb||'</td>

<td>'||r3.pct\_free||'</td>

<td>'||r3.max\_gb||'</td>

<td>'||r3.pct\_used\_max||'</td>

</tr>' ;

end if;

end loop;

--dbms\_output.put\_line('1:' || length(tTBS));

tTBS := tTBS || '</table><br>';

tAll := tAll || tTBS;

--dbms\_output.put\_line('3.TBS');

v\_user\_usage :='<br><b>DUNG LUONG THEO USER:</b><table width="600" border ="1" cellspacing="1" cellpadding="1">

<tr bgcolor="#FFF1D9" style="font-weight:bold">

<td> User </td>

<td> GB </td>

</tr>';

--dbms\_output.put\_line('for r31 in c\_user\_usage loop');

for r31 in c\_user\_usage loop

if r31."GB" > 100 then

v\_user\_usage:=v\_user\_usage||'<tr>

<td><font color="red"><b>'||r31.owner||'</b></font></td>

<td><font color="red"><b>'||r31."GB"||'</b></font></td>

</tr>' ;

else

v\_user\_usage:=v\_user\_usage||'<tr>

<td>'||r31.owner||'</td>

<td>'||r31."GB"||'</td>

</tr>' ;

end if;

end loop;

--dbms\_output.put\_line('end loop;');

v\_user\_usage := v\_user\_usage || '</table>';

tAll := tAll || v\_user\_usage;

--dbms\_output.put\_line('3.DUNG LUONG THEO USER');

v\_object\_usage :='<br><b>DUNG LUONG THEO OBJECT:</b><table width="600" border ="1" cellspacing="1" cellpadding="1">

<tr bgcolor="#FFF1D9" style="font-weight:bold">

<td> User</td>

<td> Segment\_name/td>

<td> GB </td>

</tr>';

for r32 in c\_object\_usage loop

if r32."GB" > 1000 then

v\_object\_usage:=v\_object\_usage||'<tr>

<td><font color="red"><b>'||r32.owner||'</b></font></td>

<td><font color="red"><b>'||r32.segment\_name||'</b></font></td>

<td><font color="red"><b>'||r32."GB"||'</b></font></td>

</tr>' ;

else

v\_object\_usage:=v\_object\_usage||'<tr>

<td>'||r32.owner||'</td>

<td>'||r32.segment\_name||'</td>

<td>'||r32."GB"||'</td>

</tr>';

end if;

end loop;

v\_object\_usage := v\_object\_usage || '</table>';

tAll := tAll || v\_object\_usage;

/\*dbms\_output.put\_line('3.DUNG LUONG THEO OBJECT');

--dbms\_output.put\_line('3');

v\_du :='<br><b>DUNG LUONG CAC PHAN VUNG OS:</b><table width="600" border ="1" cellspacing="1" cellpadding="1">

<tr bgcolor="#FFF1D9" style="font-weight:bold">

<td> Host\_Name</td>

<td> Name</td>

<td> Size </td>

<td> Free </td>

<td> Percent </td>

<td> Mount Point </td>

<td> Event\_Date </td>

<td> IP </td>

</tr>';

--dbms\_output.put\_line('v\_object\_usage');

for r33 in c\_du loop

if(substr(r33.Disk\_percen,-3,0) > 80) then

v\_du:=v\_du||'<tr>

<td><font color="red"><b>'||r33.host\_name||'</b></font></td>

<td><font color="red"><b>'||r33.disk\_name||'</b></font></td>

<td><font color="red"><b>'||r33.Disk\_size||'</b></font></td>

<td><font color="red"><b>'||r33.Disk\_free||'</b></font></td>

<td><font color="red"><b>'||r33.Disk\_percen||'</b></font></td>

<td><font color="red"><b>'||r33.Disk\_mount||'</b></font></td>

<td><font color="red"><b>'||to\_char(r33.event\_date,'dd/mm/yyyy hh24:mm:ss')||'</b></font></td>

<td><font color="red"><b>'||r33.db\_ip||'</b></font></td>

</tr>' ;

else

v\_du:=v\_du||'<tr>

<td>'||r33.host\_name||'</td>

<td>'||r33.disk\_name||'</td>

<td>'||r33.Disk\_size||'</td>

<td>'||r33.Disk\_free||'</td>

<td>'||r33.Disk\_percen||'</td>

<td>'||r33.Disk\_mount||'</td>

<td>'||to\_char(r33.event\_date,'dd/mm/yyyy hh24:mm:ss')||'</td>

<td>'||r33.db\_ip||'</td>

</tr>' ;

end if;

end loop;

v\_du := v\_du || '</table>';

tAll := tAll || v\_du; \*/

--dbms\_output.put\_line('3.DUNG LUONG THEO OS');

--dbms\_output.put\_line('3.Completed');

--4.Object invalid

select count(\*) into nInvalidObject from (select 'ALTER '||OBJECT\_TYPE||' '||OWNER||'.'||OBJECT\_NAME||' COMPILE;' from dba\_objects

where object\_type in ('PROCEDURE','FUNCTION','TRIGGER','PACKAGE') and status like 'INVALID'and OWNER in ('CDR\_OWNER','CUS\_OWNER','MC\_OWNER')

UNION ALL

select 'ALTER PACKAGE '||OWNER||'.'||OBJECT\_NAME||' COMPILE BODY;' from dba\_objects

where object\_type in ('PACKAGE BODY') and status like 'INVALID' and OWNER like 'CDR\_OWNER');

tAll := tAll||'<br><b>OBJECT INVALID:</b><table width="600" border ="1" cellspacing="1" cellpadding="1">

<tr bgcolor="#FFF1D9" style="font-weight:bold">

<td> So luong Object Invalid </td>

<td> So luong Index Partition UNUSABLE </td>

<td> So luong Index Non-Partition UNUSABLE </td>

</tr>';

--dbms\_output.put\_line('tAll: ' || length(tAll));

tAll := tAll||'<tr>

<td>'||nInvalidObject||'</td>' ;

--dbms\_output.put\_line('tAll: ' || length(tAll));

--Index invalid: Partition, non-partition

select count(\*) into nUnuableIndNonPar from dba\_indexes where status='UNUSABLE';

select count(\*) into nUnuableIndPar from dba\_ind\_partitions where status='UNUSABLE';

tAll := tAll||'

<td>'||nUnuableIndPar||'</td>

<td>'||nUnuableIndNonPar||'</td>

</tr>' ;

tAll := tAll || '</table>';

--dbms\_output.put\_line('4');

--5.Archive log trong 1 ngay hien tai

tAll := tAll||'<br><b>ARCHIVE LOG TRONG 7 NGAY:</b><table width="600" border ="1" cellspacing="1" cellpadding="1">

<tr bgcolor="#FFF1D9" style="font-weight:bold">

<td> Completion\_time </td>

<td> Archived\_Log\_GB </td>

</tr>';

for r1 in (select trunc(completion\_time) completion\_time, round(sum(blocks\*block\_size)/1024/1024/1024,0) as archived\_log\_gb from V$ARCHIVED\_LOG

where trunc(completion\_time) >= trunc(sysdate-3)

--and trunc(completion\_time)>= to\_date(trunc(sysdate),'dd/mm/yyyy')

and dest\_id=1

group by trunc(completion\_time)

order by trunc(completion\_time) desc) loop

if r1.archived\_log\_gb > 1000 then

tAll:=tAll||'<tr>

<td><font color="red"><b>'||r1.completion\_time||'</b></font></td>

<td><font color="red"><b>'||r1.archived\_log\_gb||'</b></font></td>

</tr>' ;

else

tAll:=tAll||'<tr>

<td>'||r1.completion\_time||'</td>

<td>'||r1.archived\_log\_gb||'</td>

</tr>' ;

end if;

end loop;

tAll:=tAll||'</table>';

--dbms\_output.put\_line('5');

--6.Gather

tAll := tAll||'<br><b>GATHER:</b><table width="600" border ="1" cellspacing="1" cellpadding="1">

<tr bgcolor="#FFF1D9" style="font-weight:bold">

<td> tGatherTabPar </td>

<td> tGatherIndPar </td>

<td> tGatherTabNonPar </td>

<td> tGatherIndNonPar </td>

</tr>';

select to\_date(max(last\_analyzed),'dd/mm/yyyy') into tGatherTabPar from dba\_tab\_partitions

where table\_owner in ('CDR\_OWNER','CUS\_OWNER','MC\_OWNER') and last\_analyzed<sysdate and last\_analyzed>sysdate-7;

select to\_date(max(last\_analyzed),'dd/mm/yyyy') into tGatherIndPar from dba\_ind\_partitions

where index\_owner in ('CDR\_OWNER','CUS\_OWNER','MC\_OWNER') and last\_analyzed<sysdate and last\_analyzed>sysdate-7;

-- Non-partition

select to\_date(max(last\_analyzed),'dd/mm/yyyy') into tGatherTabNonPar from dba\_tables

where owner in ('CDR\_OWNER','CUS\_OWNER','MC\_OWNER') and last\_analyzed<sysdate and last\_analyzed>sysdate-7 ;

select to\_date(max(last\_analyzed),'dd/mm/yyyy') into tGatherIndNonPar from dba\_indexes

where owner in ('CDR\_OWNER','CUS\_OWNER','MC\_OWNER') and last\_analyzed<sysdate and last\_analyzed>sysdate-7;

tAll:=tAll||'<tr>

<td>'||tGatherTabPar||'</td>

<td>'||tGatherIndPar||'</td>

<td>'||tGatherTabNonPar||'</td>

<td>'||tGatherIndNonPar||'</td>

</tr>' ;

tAll := tAll || '</table>';

--dbms\_output.put\_line('6');

--dbms\_output.put\_line('tAll: '||length(tAll));

--execute immediate 'truncate table tc\_monior\_lob';

--insert into tc\_monior\_lob(text) values(tAll);

--commit;

--select text into tAll\_New from tc\_monior\_lob;

IF nActiveSession >= 250 then

cdr\_monitor.send\_email\_html\_m('tranbinh48ca@gmail.com','HIGH - HC\_' || tDBName ||'\_ALL NGAY ' || to\_char(sysdate,'dd/mm/yyyy hh24:mm:ss'), tAll);

ELSIF nActiveSession > 200 and nActiveSession < 250 then

cdr\_monitor.send\_email\_html\_m('tranbinh48ca@gmail.com','MID - HC\_' || tDBName ||'\_ALL NGAY ' || to\_char(sysdate,'dd/mm/yyyy hh24:mm:ss'), tAll);

ELSE

cdr\_monitor.send\_email\_html\_m('tranbinh48ca@gmail.com','LOW - HC\_' || tDBName ||'\_ALL NGAY ' || to\_char(sysdate,'dd/mm/yyyy hh24:mm:ss'), tAll);

END IF;

--send\_sms\_binhtv('sys.dba\_rp.hc');

--end if;

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Completed sys.dba\_rp.hc\_all',1,sysdate,'sys.dba\_rp.hc\_all');

commit;

EXCEPTION

WHEN others THEN

send\_sms\_binhtv('Error sys.dba\_rp.hc: ' || SQLERRM);

v\_err := SUBSTR(SQLERRM, 1, 200);

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Error sys.dba\_rp.hc\_all',-1,sysdate,'Error sys.dba\_rp.hc\_all, '||v\_err);

commit;

END;

PROCEDURE hc\_backup

is

tAll varchar2(32700):='';

tAll\_New varchar2(10000):='';

tDBName varchar2(20):='';

v\_err varchar2(1000):='';

--1.Tai DB

nActiveSession number;

nTotalActiveSession number; -- Ca background

nTotalInactiveSession number;

nTotalSession number;

nLock number;

nLockDBLink number;

--2.Backup

CURSOR cBackup

IS

select command\_id, start\_time, end\_time, status,INPUT\_TYPE, input\_bytes\_display, output\_bytes\_display, time\_taken\_display, round(compression\_ratio,0) RATIO , input\_bytes\_per\_sec\_display, output\_bytes\_per\_sec\_display

from v$rman\_backup\_job\_details

where trunc(end\_time)>=trunc(sysdate-30)

order by end\_time desc;

tBackup varchar2(30000):='';

BEGIN

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Starting sys.dba\_rp.hc\_backup',1,sysdate,'sys.dba\_rp.hc\_backup');

commit;

--dbms\_output.put\_line('Before if');

--if (to\_char(sysdate,'hh24') in ('00','08','13')) then

--dbms\_output.put\_line('After if');

select name into tDBName from v$database;

tAll := tAll ||'<h2>BAO CAO BACKUP CSDL ' || tDBName ||' NGAY ' || to\_char(sysdate,'dd/mm/yyyy hh24:mm:ss') ||'</h2>';

--2.BACKUP: Lay 30 ngay

tBackup:=tBackup||'<br><b>SUMMARY BACKUP:</b><table width="600" border ="1" cellspacing="1" cellpadding="1">

<tr bgcolor="#FFF1D9" style="font-weight:bold">

<td>command\_id</td>

<td>start\_time</td>

<td>end\_time</td>

<td>status</td>

<td>INPUT\_TYPE</td>

<td>input\_bytes\_display</td>

<td>output\_bytes\_display</td>

<td>time\_taken\_display</td>

<td>RATIO</td>

<td>input\_bytes\_per\_sec\_display</td>

<td>output\_bytes\_per\_sec\_display</td>

</tr>';

for r1 in cBackup loop

tBackup:=tBackup||'<tr>

<td>'||r1.command\_id||'</td>

<td>'||to\_char(r1.start\_time,'dd/mm/yyyy hh24:mm:ss')||'</td>

<td>'||to\_char(r1.end\_time,'dd/mm/yyyy hh24:mm:ss')||' </td>

<td>'||r1.status||' </td>

<td>'||r1.INPUT\_TYPE||'</td>

<td>'||r1.input\_bytes\_display||'</td>

<td>'||r1.output\_bytes\_display||'</td>

<td>'||r1.time\_taken\_display||'</td>

<td>'||r1.RATIO||'</td>

<td>'||r1.input\_bytes\_per\_sec\_display||'</td>

<td>'||r1.output\_bytes\_per\_sec\_display||'</td>

</tr>' ;

end loop;

tBackup := tBackup || '</table>';

tAll := tAll || tBackup;

--dbms\_output.put\_line('2');

--dbms\_output.put\_line('6');

dbms\_output.put\_line('tAll: '||length(tAll));

--execute immediate 'truncate table tc\_monior\_lob';

--insert into tc\_monior\_lob(text) values(tAll);

--commit;

--select text into tAll\_New from tc\_monior\_lob;

cdr\_monitor.send\_email\_html\_m('tranbinh48ca@gmail.com','RP\_' || tDBName ||'\_BACKUP NGAY ' || to\_char(sysdate,'dd/mm/yyyy hh24:mm:ss'), tAll);

--send\_sms\_binhtv('sys.dba\_rp.hc\_backup');

--end if;

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Completed sys.dba\_rp.hc\_backup',1,sysdate,'sys.dba\_rp.hc\_backup');

commit;

EXCEPTION

WHEN others THEN

send\_sms\_binhtv('Error sys.dba\_rp.hc\_backup: ' || SQLERRM);

v\_err := SUBSTR(SQLERRM, 1, 200);

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Error sys.dba\_rp.hc\_backup',-1,sysdate,'Error sys.dba\_rp.hc\_backup, ' ||v\_err);

commit;

end;

--PROCEDURE hc\_ddl;

procedure hc\_os

is

tAll varchar2(32700):='';

tAll\_New varchar2(10000):='';

tDBName varchar2(20):='';

v\_err varchar2(1000):='';

cursor c\_cpu\_ram

is

select event\_date, duration, server\_name, cpu\_idle, round(ram\_free\_k/1024/1024,0) "GB"

from BINHTV.dbamf\_ct\_vmstat

where event\_date > sysdate-1/24

and (cpu\_idle <= 20 or round(ram\_free\_k/1024/1024,0) <=70)

order by event\_date desc;

v\_cpu\_ram varchar2(30000):='';

cursor c\_du

is

select \* from binhtv.dbamf\_ct\_disk\_usage

where event\_date>sysdate-1

and substr(disk\_percen,-3,0)>'50'

order by disk\_percen desc;

v\_du varchar2(30000):='';

BEGIN

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Starting sys.dba\_rp.hc\_os',1,sysdate,'sys.dba\_rp.hc\_os');

commit;

--CPU,RAM

v\_cpu\_ram:=v\_cpu\_ram||'<br><b>CPU, RAM USAGE:</b><table width="600" border ="1" cellspacing="1" cellpadding="1">

<tr bgcolor="#FFF1D9" style="font-weight:bold">

<td> event\_date</td>

<td> duration</td>

<td> server\_name</td>

<td> cpu\_idle</td>

<td> GB</td>

</tr>';

for r11 in c\_cpu\_ram loop

v\_cpu\_ram:=v\_cpu\_ram||'<tr>

<td>'||to\_char(r11.event\_date,'dd/mm/yyyy hh24:mm:ss')||'</td>

<td>'||r11.duration||'</td>

<td>'||r11.server\_name||'</td>

<td>'||r11.cpu\_idle||'</td>

<td>'||r11.GB||'</td>

</tr>' ;

end loop;

v\_cpu\_ram := v\_cpu\_ram || '</table>';

tAll := tAll || v\_cpu\_ram;

v\_du :='<br><b>DUNG LUONG CAC PHAN VUNG OS:</b><table width="600" border ="1" cellspacing="1" cellpadding="1">

<tr bgcolor="#FFF1D9" style="font-weight:bold">

<td> Host\_Name</td>

<td> Name</td>

<td> Size </td>

<td> Free </td>

<td> Percent </td>

<td> Mount Point </td>

<td> Event\_Date </td>

<td> IP </td>

</tr>';

--dbms\_output.put\_line('v\_object\_usage');

for r33 in c\_du loop

if(substr(r33.Disk\_percen,-3,0) > 80) then

v\_du:=v\_du||'<tr>

<td><font color="red"><b>'||r33.host\_name||'</b></font></td>

<td><font color="red"><b>'||r33.disk\_name||'</b></font></td>

<td><font color="red"><b>'||r33.Disk\_size||'</b></font></td>

<td><font color="red"><b>'||r33.Disk\_free||'</b></font></td>

<td><font color="red"><b>'||r33.Disk\_percen||'</b></font></td>

<td><font color="red"><b>'||r33.Disk\_mount||'</b></font></td>

<td><font color="red"><b>'||to\_char(r33.event\_date,'dd/mm/yyyy hh24:mm:ss')||'</b></font></td>

<td><font color="red"><b>'||r33.db\_ip||'</b></font></td>

</tr>' ;

else

v\_du:=v\_du||'<tr>

<td>'||r33.host\_name||'</td>

<td>'||r33.disk\_name||'</td>

<td>'||r33.Disk\_size||'</td>

<td>'||r33.Disk\_free||'</td>

<td>'||r33.Disk\_percen||'</td>

<td>'||r33.Disk\_mount||'</td>

<td>'||to\_char(r33.event\_date,'dd/mm/yyyy hh24:mm:ss')||'</td>

<td>'||r33.db\_ip||'</td>

</tr>' ;

end if;

end loop;

v\_du := v\_du || '</table>';

tAll := tAll || v\_du;

--dbms\_output.put\_line('6');

dbms\_output.put\_line('tAll: '||length(tAll));

--execute immediate 'truncate table tc\_monior\_lob';

--insert into tc\_monior\_lob(text) values(tAll);

--commit;

--select text into tAll\_New from tc\_monior\_lob;

cdr\_monitor.send\_email\_html\_m('tranbinh48ca@gmail.com','RP' || tDBName ||'\_OS NGAY ' || to\_char(sysdate,'dd/mm/yyyy hh24:mm:ss'), tAll);

--send\_sms\_binhtv('sys.dba\_rp.hc\_os');

--end if;

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Completed sys.dba\_rp.hc\_os',1,sysdate,'sys.dba\_rp.hc\_os');

commit;

EXCEPTION

WHEN others THEN

send\_sms\_binhtv('Error sys.dba\_rp.hc: ' || SQLERRM);

v\_err := SUBSTR(SQLERRM, 1, 200);

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Error sys.dba\_rp.hc\_os',-1,sysdate,'Error sys.dba\_rp.hc\_os, '||v\_err);

commit;

END;

PROCEDURE hc\_du

is

tAll varchar2(32700):='';

tAll\_New varchar2(10000):='';

tDBName varchar2(20):='';

v\_err varchar2(1000):='';

cursor c\_du

is

select \* from binhtv.dbamf\_ct\_disk\_usage

where event\_date>sysdate-1

--and substr(disk\_percen,-3,0)>'50'

order by host\_name asc,disk\_percen desc;

v\_du varchar2(30000):='';

BEGIN

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Starting sys.dba\_rp.hc\_du',1,sysdate,'sys.dba\_rp.hc\_du');

commit;

--dbms\_output.put\_line('Before if');

--if (to\_char(sysdate,'hh24') in ('00','08','13')) then

--dbms\_output.put\_line('After if');

select name into tDBName from v$database;

tAll := tAll ||'<h2>BAO CAO CSDL ' || tDBName ||'\_DU NGAY ' || to\_char(sysdate,'dd/mm/yyyy hh24:mm:ss') ||'</h2>';

v\_du :='<br><b>DUNG LUONG CAC PHAN VUNG OS:</b><table width="600" border ="1" cellspacing="1" cellpadding="1">

<tr bgcolor="#FFF1D9" style="font-weight:bold">

<td> Host\_Name</td>

<td> Name</td>

<td> Size </td>

<td> Free </td>

<td> Percent </td>

<td> Mount Point </td>

<td> Event\_Date </td>

<td> IP </td>

</tr>';

--dbms\_output.put\_line('v\_object\_usage');

for r33 in c\_du loop

if(substr(r33.Disk\_percen,-3,0) > 80) then

v\_du:=v\_du||'<tr>

<td><font color="red"><b>'||r33.host\_name||'</b></font></td>

<td><font color="red"><b>'||r33.disk\_name||'</b></font></td>

<td><font color="red"><b>'||r33.Disk\_size||'</b></font></td>

<td><font color="red"><b>'||r33.Disk\_free||'</b></font></td>

<td><font color="red"><b>'||r33.Disk\_percen||'</b></font></td>

<td><font color="red"><b>'||r33.Disk\_mount||'</b></font></td>

<td><font color="red"><b>'||to\_char(r33.event\_date,'dd/mm/yyyy hh24:mm:ss')||'</b></font></td>

<td><font color="red"><b>'||r33.db\_ip||'</b></font></td>

</tr>' ;

else

v\_du:=v\_du||'<tr>

<td>'||r33.host\_name||'</td>

<td>'||r33.disk\_name||'</td>

<td>'||r33.Disk\_size||'</td>

<td>'||r33.Disk\_free||'</td>

<td>'||r33.Disk\_percen||'</td>

<td>'||r33.Disk\_mount||'</td>

<td>'||to\_char(r33.event\_date,'dd/mm/yyyy hh24:mm:ss')||'</td>

<td>'||r33.db\_ip||'</td>

</tr>' ;

end if;

end loop;

v\_du := v\_du || '</table>';

tAll := tAll || v\_du;

--dbms\_output.put\_line('6');

--dbms\_output.put\_line('tAll: '||length(tAll));

--execute immediate 'truncate table tc\_monior\_lob';

--insert into tc\_monior\_lob(text) values(tAll);

--commit;

--select text into tAll\_New from tc\_monior\_lob;

cdr\_monitor.send\_email\_html\_m('tranbinh48ca@gmail.com','HC\_' || tDBName ||'\_DU NGAY ' || to\_char(sysdate,'dd/mm/yyyy hh24:mm:ss'), tAll);

--send\_sms\_binhtv('sys.dba\_rp.hc\_du');

--end if;

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Completed sys.dba\_rp.hc\_du',1,sysdate,'sys.dba\_rp.hc\_du');

commit;

EXCEPTION

WHEN others THEN

send\_sms\_binhtv('Error sys.dba\_rp.hc\_du: ' || SQLERRM);

--v\_err := SUBSTR(SQLERRM, 1, 200);

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Error sys.dba\_rp.hc\_du',-1,sysdate,'Error sys.dba\_rp.hc\_du, '||v\_err);

commit;

end;

PROCEDURE hc\_storage

is

tAll varchar2(32700):='';

tAll\_New varchar2(10000):='';

tDBName varchar2(20):='';

v\_err varchar2(1000):='';

--2.Backup

CURSOR cBackup

IS

select command\_id, start\_time, end\_time, status,INPUT\_TYPE, input\_bytes\_display, output\_bytes\_display, time\_taken\_display, round(compression\_ratio,0) RATIO , input\_bytes\_per\_sec\_display, output\_bytes\_per\_sec\_display

from v$rman\_backup\_job\_details

where trunc(end\_time)>=trunc(sysdate-3)

order by end\_time desc;

tBackup varchar2(30000):='';

--3.Storage

cursor cASM is

select name, state, type, total\_mb, usable\_file\_mb from v$asm\_diskgroup;

tASM varchar2(30000) := '';

cursor cTBS is

SELECT a.tablespace\_name,100 - ROUND ( (NVL (b.bytes\_free, 0) / a.bytes\_alloc) \* 100) pct\_usage, ROUND(a.bytes\_alloc / 1024 / 1024) size\_mb, ROUND (NVL (b.bytes\_free, 0) / 1024 / 1024) free\_mb,

(ROUND (a.bytes\_alloc / 1024 / 1024)- ROUND (NVL (b.bytes\_free, 0) / 1024 / 1024)) used\_mb, ROUND ( (NVL (b.bytes\_free, 0) / a.bytes\_alloc) \* 100) pct\_free, ROUND (maxbytes / 1048576) max\_mb,

ROUND (ROUND ( (a.bytes\_alloc - NVL (b.bytes\_free, 0)) / 1024 / 1024)/ ROUND (maxbytes / 1048576) \* 100) as pct\_used\_max

FROM ( SELECT f.tablespace\_name, SUM (f.bytes) bytes\_alloc, SUM (DECODE (f.autoextensible, 'YES', f.maxbytes, 'NO', f.bytes)) maxbytes

FROM dba\_data\_files f

GROUP BY tablespace\_name) a,

( SELECT f.tablespace\_name, SUM (f.bytes) bytes\_free FROM dba\_free\_space f GROUP BY tablespace\_name) b

WHERE a.tablespace\_name = b.tablespace\_name(+)

--and a.tablespace\_name in ('DATA','INDX')

and ROUND (ROUND ( (a.bytes\_alloc - NVL (b.bytes\_free, 0)) / 1024 / 1024)/ ROUND (maxbytes / 1048576) \* 100) > 85

order by pct\_used\_max desc;

tTBS varchar2(30000) := '';

cursor c\_user\_usage is

select owner,round(sum(bytes)/1024/1024/1024,0) "GB" from dba\_segments

group by owner

having round(sum(bytes)/1024/1024/1024,0) > 50

order by "GB" desc;

v\_user\_usage varchar2(30000) := '';

cursor c\_object\_usage is

select owner,segment\_name,round(sum(bytes)/1024/1024/1024,0) "GB" from dba\_segments

group by owner, segment\_name

having round(sum(bytes)/1024/1024/1024,0) > 100

order by "GB" desc;

v\_object\_usage varchar2(3000) := '';

-- Cac tablespace Read Only:Chay lau

/\* cursor cTBS\_RO is

select tablespace\_name, round(sum(bytes)/1024/1024/1024,0) "GB"

from dba\_segments

where tablespace\_name in (select name from v$tablespace where ts# in (select ts# from v$datafile where enabled='READ ONLY'))

group by tablespace\_name

order by tablespace\_name ;

tStorage varchar2(4000):='';\*/

cursor c\_du

is

select \* from binhtv.dbamf\_ct\_disk\_usage

where event\_date>sysdate-1

and substr(disk\_percen,-3,0)>'50'

order by disk\_percen desc;

v\_du varchar2(30000):='';

--6.Archive log

-- Theo doi archived log sinh ra

cursor cTotalArchivedLog is

select trunc(completion\_time), round(sum(blocks\*block\_size)/1024/1024/1024,0) "Archived Log GB" from V$ARCHIVED\_LOG

where trunc(completion\_time) >= trunc(sysdate-1)

--and trunc(completion\_time)>= to\_date(trunc(sysdate),'dd/mm/yyyy')

and dest\_id=1

group by trunc(completion\_time)

order by trunc(completion\_time) desc;

-- Archived log sinh ra theo gio

cursor cHourArchivedLog is

select to\_char(next\_time,'YYYY-MM-DD hh24') Hour, round(sum(size\_in\_byte)/1024/1024,0) as size\_in\_mb, count(\*) log\_switch from (

select thread# ,sequence#, FIRST\_CHANGE#,blocks\*BLOCK\_SIZE as size\_in\_byte, next\_time

from v$archived\_log where name is not null and completion\_time>sysdate-1 group by thread# ,sequence#, FIRST\_CHANGE#,blocks\*BLOCK\_SIZE, next\_time)

group by to\_char(next\_time,'YYYY-MM-DD hh24') order by 1 desc;

tArchivedLog varchar2(30000):='';

BEGIN

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Starting sys.dba\_rp.hc\_storage',1,sysdate,'sys.dba\_rp.hc\_storage');

commit;

--dbms\_output.put\_line('Before if');

--if (to\_char(sysdate,'hh24') in ('00','08','13')) then

--dbms\_output.put\_line('After if');

select name into tDBName from v$database;

tAll := tAll ||'<h2>BAO CAO STORAGE CSDL ' || tDBName ||' NGAY ' || to\_char(sysdate,'dd/mm/yyyy hh24:mm:ss') ||'</h2>';

--2.BACKUP: Lay 1 tuan

tBackup:=tBackup||'<br><b>SUMMARY BACKUP:</b><table width="600" border ="1" cellspacing="1" cellpadding="1">

<tr bgcolor="#FFF1D9" style="font-weight:bold">

<td>command\_id</td>

<td>start\_time</td>

<td>end\_time</td>

<td>status</td>

<td>INPUT\_TYPE</td>

<td>input\_bytes\_display</td>

<td>output\_bytes\_display</td>

<td>time\_taken\_display</td>

<td>RATIO</td>

<td>input\_bytes\_per\_sec\_display</td>

<td>output\_bytes\_per\_sec\_display</td>

</tr>';

for r1 in cBackup loop

tBackup:=tBackup||'<tr>

<td>'||r1.command\_id||'</td>

<td>'||to\_char(r1.start\_time,'dd/mm/yyyy hh24:mm:ss')||'</td>

<td>'||to\_char(r1.end\_time,'dd/mm/yyyy hh24:mm:ss')||' </td>

<td>'||r1.status||' </td>

<td>'||r1.INPUT\_TYPE||'</td>

<td>'||r1.input\_bytes\_display||'</td>

<td>'||r1.output\_bytes\_display||'</td>

<td>'||r1.time\_taken\_display||'</td>

<td>'||r1.RATIO||'</td>

<td>'||r1.input\_bytes\_per\_sec\_display||'</td>

<td>'||r1.output\_bytes\_per\_sec\_display||'</td>

</tr>' ;

end loop;

tBackup := tBackup || '</table>';

tAll := tAll || tBackup;

--dbms\_output.put\_line('2');

--3.STORAGE

tASM :='<br><b>ASM DISKGROUP:</b><table width="600" border ="1" cellspacing="1" cellpadding="1">

<tr bgcolor="#FFF1D9" style="font-weight:bold">

<td>name</td>

<td>state</td>

<td>type</td>

<td>total\_mb</td>

<td>usable\_file\_mb</td>

</tr>';

for r2 in cASM loop

tASM:=tASM||'<tr>

<td>'||r2.name||'</td>

<td>'||r2.state||'</td>

<td>'||r2.type||'</td>

<td>'||r2.total\_mb||'</td>

<td>'||r2.usable\_file\_mb||'</td>

</tr>' ;

end loop;

tASM := tASM || '</table><br>';

tAll := tAll || tASM;

--dbms\_output.put\_line('3.tASM');

tTBS :='<br><b>TABLESPACE:</b><table width="600" border ="1" cellspacing="1" cellpadding="1">

<tr bgcolor="#FFF1D9" style="font-weight:bold">

<td> Tablespace Name </td>

<td> %Usage </td>

<td> Size MB </td>

<td> Free MB </td>

<td> Used MB </td>

<td> %Free </td>

<td> Max MB </td>

<td> %Used Max </td>

</tr>';

--dbms\_output.put\_line('0');

for r3 in cTBS loop

tTBS:=tTBS||'<tr>

<td>'||r3.tablespace\_name||'</td>

<td>'||r3.pct\_usage||'</td>

<td>'||r3.size\_mb||'</td>

<td>'||r3.free\_mb||'</td>

<td>'||r3.used\_mb||'</td>

<td>'||r3.pct\_free||'</td>

<td>'||r3.max\_mb||'</td>

<td>'||r3.pct\_used\_max||'</td>

</tr>' ;

end loop;

--dbms\_output.put\_line('1:' || length(tTBS));

tTBS := tTBS || '</table><br>';

tAll := tAll || tTBS;

--dbms\_output.put\_line('3.TBS');

v\_user\_usage :='<br><b>DUNG LUONG THEO USER:</b><table width="600" border ="1" cellspacing="1" cellpadding="1">

<tr bgcolor="#FFF1D9" style="font-weight:bold">

<td> User </td>

<td> GB </td>

</tr>';

--dbms\_output.put\_line('for r31 in c\_user\_usage loop');

for r31 in c\_user\_usage loop

if r31."GB" > 100 then

v\_user\_usage:=v\_user\_usage||'<tr>

<td><font color="red"><b>'||r31.owner||'</b></font></td>

<td><font color="red"><b>'||r31."GB"||'</b></font></td>

</tr>' ;

else

v\_user\_usage:=v\_user\_usage||'<tr>

<td>'||r31.owner||'</td>

<td>'||r31."GB"||'</td>

</tr>' ;

end if;

end loop;

--dbms\_output.put\_line('end loop;');

v\_user\_usage := v\_user\_usage || '</table>';

tAll := tAll || v\_user\_usage;

v\_object\_usage :='<br><b>DUNG LUONG THEO OBJECT:</b><table width="600" border ="1" cellspacing="1" cellpadding="1">

<tr bgcolor="#FFF1D9" style="font-weight:bold">

<td> User</td>

<td> Segment\_name/td>

<td> GB </td>

</tr>';

--dbms\_output.put\_line('v\_object\_usage');

for r32 in c\_object\_usage loop

if r32."GB" > 1000 then

v\_object\_usage:=v\_object\_usage||'<tr>

<td><font color="red"><b>'||r32.owner||'</b></font></td>

<td><font color="red"><b>'||r32.segment\_name||'</b></font></td>

<td><font color="red"><b>'||r32."GB"||'</b></font></td>

</tr>' ;

else

v\_object\_usage:=v\_object\_usage||'<tr>

<td>'||r32.owner||'</td>

<td>'||r32.segment\_name||'</td>

<td>'||r32."GB"||'</td>

</tr>';

end if;

end loop;

v\_object\_usage := v\_object\_usage || '</table>';

tAll := tAll || v\_object\_usage;

--dbms\_output.put\_line('1:' || length(v\_user\_usage));

--dbms\_output.put\_line('3');

v\_du :='<br><b>DUNG LUONG CAC PHAN VUNG OS:</b><table width="600" border ="1" cellspacing="1" cellpadding="1">

<tr bgcolor="#FFF1D9" style="font-weight:bold">

<td> Host\_Name</td>

<td> Name</td>

<td> Size </td>

<td> Free </td>

<td> Percent </td>

<td> Mount Point </td>

<td> Event\_Date </td>

<td> IP </td>

</tr>';

--dbms\_output.put\_line('v\_object\_usage');

for r33 in c\_du loop

if(substr(r33.Disk\_percen,-3,0) > 80) then

v\_du:=v\_du||'<tr>

<td><font color="red"><b>'||r33.host\_name||'</b></font></td>

<td><font color="red"><b>'||r33.disk\_name||'</b></font></td>

<td><font color="red"><b>'||r33.Disk\_size||'</b></font></td>

<td><font color="red"><b>'||r33.Disk\_free||'</b></font></td>

<td><font color="red"><b>'||r33.Disk\_percen||'</b></font></td>

<td><font color="red"><b>'||r33.Disk\_mount||'</b></font></td>

<td><font color="red"><b>'||to\_char(r33.event\_date,'dd/mm/yyyy hh24:mm:ss')||'</b></font></td>

<td><font color="red"><b>'||r33.db\_ip||'</b></font></td>

</tr>' ;

else

v\_du:=v\_du||'<tr>

<td>'||r33.host\_name||'</td>

<td>'||r33.disk\_name||'</td>

<td>'||r33.Disk\_size||'</td>

<td>'||r33.Disk\_free||'</td>

<td>'||r33.Disk\_percen||'</td>

<td>'||r33.Disk\_mount||'</td>

<td>'||to\_char(r33.event\_date,'dd/mm/yyyy hh24:mm:ss')||'</td>

<td>'||r33.db\_ip||'</td>

</tr>' ;

end if;

end loop;

v\_du := v\_du || '</table>';

tAll := tAll || v\_du;

--5.Archive log trong 1 ngay hien tai

tAll := tAll||'<br><b>ARCHIVE LOG TRONG 7 NGAY:</b><table width="600" border ="1" cellspacing="1" cellpadding="1">

<tr bgcolor="#FFF1D9" style="font-weight:bold">

<td> Completion\_time </td>

<td> Archived\_Log\_GB </td>

</tr>';

for r1 in (select trunc(completion\_time) completion\_time, round(sum(blocks\*block\_size)/1024/1024/1024,0) as archived\_log\_gb from V$ARCHIVED\_LOG

where trunc(completion\_time) >= trunc(sysdate-7)

--and trunc(completion\_time)>= to\_date(trunc(sysdate),'dd/mm/yyyy')

and dest\_id=1

group by trunc(completion\_time)

order by trunc(completion\_time) desc) loop

if r1.archived\_log\_gb > 1000 then

tAll:=tAll||'<tr>

<td><font color="red"><b>'||r1.completion\_time||'</b></font></td>

<td><font color="red"><b>'||r1.archived\_log\_gb||'</b></font></td>

</tr>' ;

else

tAll:=tAll||'<tr>

<td>'||r1.completion\_time||'</td>

<td>'||r1.archived\_log\_gb||'</td>

</tr>' ;

end if;

end loop;

tAll:=tAll||'</table>';

--dbms\_output.put\_line('5');

--dbms\_output.put\_line('6');

dbms\_output.put\_line('tAll: '||length(tAll));

--execute immediate 'truncate table tc\_monior\_lob';

--insert into tc\_monior\_lob(text) values(tAll);

--commit;

--select text into tAll\_New from tc\_monior\_lob;

cdr\_monitor.send\_email\_html\_m('tranbinh48ca@gmail.com','HC\_' || tDBName ||'\_STORAGE NGAY ' || to\_char(sysdate,'dd/mm/yyyy hh24:mm:ss'), tAll);

send\_sms\_binhtv('sys.dba\_rp.hc\_storage');

--end if;

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Completed sys.dba\_rp.hc\_storage',1,sysdate,'sys.dba\_rp.hc\_storage');

commit;

EXCEPTION

WHEN others THEN

send\_sms\_binhtv('Error sys.dba\_rp.hc\_storage: ' || SQLERRM);

v\_err := SUBSTR(SQLERRM, 1, 200);

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Error sys.dba\_rp.hc\_storage',-1,sysdate,'Error sys.dba\_rp.hc\_storage, '||v\_err);

commit;

end;

PROCEDURE hc\_sms

is

v\_active\_session number :=0;

v\_total\_active\_session number := 0;

v\_total\_session number := 0;

v\_lock number := 0;

v\_backup\_status varchar2(100) := 'Fail' ;

v\_count\_backup number;

v\_tbl\_par\_date1 varchar2(100) :='';

v\_tbl\_non\_par\_date1 varchar2(100) :='';

v\_all varchar2(1000) := '';

v\_tbs varchar2(20) := '';

v\_used\_max number := 0;

V\_TBL\_NON\_PAR\_DATE varchar2(50) :='';

V\_TBL\_PAR\_DATE varchar2(50) := '';

V\_INVALID\_OBJ number := 0;

v\_err varchar2(1000):='';

cursor c\_tbs is

SELECT a.tablespace\_name,ROUND (ROUND ( (a.bytes\_alloc - NVL (b.bytes\_free, 0)) / 1024 / 1024)/ ROUND (maxbytes / 1048576) \* 100) as pct\_used\_max

FROM ( SELECT f.tablespace\_name, SUM (f.bytes) bytes\_alloc, SUM (DECODE (f.autoextensible, 'YES', f.maxbytes, 'NO', f.bytes)) maxbytes

FROM dba\_data\_files f

GROUP BY tablespace\_name) a,

( SELECT f.tablespace\_name, SUM (f.bytes) bytes\_free FROM dba\_free\_space f GROUP BY tablespace\_name) b

WHERE a.tablespace\_name = b.tablespace\_name(+)

and ROUND (ROUND ( (a.bytes\_alloc - NVL (b.bytes\_free, 0)) / 1024 / 1024)/ ROUND (maxbytes / 1048576) \* 100)>96;

cursor c\_archived\_log is

select trunc(completion\_time) as completion\_time, round(sum(blocks\*block\_size)/1024/1024/1024,0) as archived\_log from V$ARCHIVED\_LOG

where trunc(completion\_time) >= trunc(sysdate-1)

--and trunc(completion\_time)>= to\_date(trunc(sysdate),'dd/mm/yyyy')

and dest\_id=1

group by trunc(completion\_time)

order by trunc(completion\_time) desc;

BEGIN

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Starting sys.dba\_rp.hc\_sms',1,sysdate,'binhtv.dbamf\_log\_jobs');

commit;

--select to\_char(sysdate,'hh') from dual;

--if (to\_char(sysdate,'hh24') in ('08','17')) then

select count(\*) into v\_active\_session from gv$session where status='ACTIVE' and username NOT in ('SYS','SYSMAN','DBSNMP','GGATE','DBAVIETENGATE') and type != 'BACKGROUND' AND TYPE = 'USER' ;

select count(\*) into v\_total\_active\_session from gv$session where status='ACTIVE';

select count(\*) into v\_total\_session from gv$session;

Select count(\*) into v\_lock From gv$session where blocking\_session is not NULL and type not like 'BACKGROUND';

v\_all := v\_all || 'DBAVIET\_HC\_' || to\_char(sysdate,'dd/mm hh24:mi')||':Active ('||v\_active\_session || '),TAS('||v\_total\_active\_session||'),TS('||v\_total\_session||'),Lock('||v\_lock||')';

--dbms\_output.put\_line('1.'||v\_all);

--2.BACKUP

-- select command\_id, start\_time, end\_time, status,INPUT\_TYPE, input\_bytes\_display, output\_bytes\_display, time\_taken\_display, round(compression\_ratio,0) RATIO , input\_bytes\_per\_sec\_display, output\_bytes\_per\_sec\_display

-- from v$rman\_backup\_job\_details

-- where trunc(end\_time)>=trunc(sysdate)

-- order by end\_time desc;

-- select substr(output\_bytes\_display,1,8) from v$rman\_backup\_job\_details

-- where trunc(end\_time)>=trunc(sysdate)

-- order by end\_time desc;

select count(1) into v\_count\_backup

from v$rman\_backup\_job\_details

where trunc(end\_time)>=trunc(sysdate)

and to\_number(trunc(substr(output\_bytes\_display,1,8)),'999')>0

order by end\_time desc;

if v\_count\_backup > 0 then

v\_backup\_status := substr(v\_backup\_status,1,3);

end if;

v\_all := v\_all ||',Backup('|| v\_backup\_status ||')';

--dbms\_output.put\_line('2.'||v\_all);

--3.STORAGE --19s

for r\_tbs in c\_tbs loop

v\_all := v\_all || ',Storage('||r\_tbs.tablespace\_name || ':'||r\_tbs.pct\_used\_max ||'%';

end loop;

v\_all := v\_all || '),';

--dbms\_output.put\_line('3.'||v\_all);

--4.ARCHIVE\_LOG

v\_all := v\_all || 'ARC(';

for r\_archived\_log in c\_archived\_log loop

v\_all := v\_all ||to\_char(r\_archived\_log.completion\_time,'dd/mm')||':'||r\_archived\_log.archived\_log ||',';

end loop;

v\_all := v\_all || '),';

--dbms\_output.put\_line('4.'||v\_all);

--5.ANALYZE

select last\_analyzed into v\_tbl\_par\_date1 from dba\_tab\_partitions

where table\_owner='CUS\_OWNER' and last\_analyzed>sysdate-1

and rownum=1

order by last\_analyzed desc;

--dbms\_output.put\_line('5.1\_Pre'||v\_all);

v\_all := v\_all || 'Table par:' ||v\_tbl\_par\_date1 || ',';

--dbms\_output.put\_line('5.1\_Post'||v\_all);

select last\_analyzed into v\_tbl\_non\_par\_date1 from dba\_tables

where owner='CUS\_OWNER' and last\_analyzed>sysdate-1

and rownum=1

order by last\_analyzed desc;

v\_all := v\_all || 'Table non\_par:' ||v\_tbl\_non\_par\_date1 ||',';

--dbms\_output.put\_line('5.2'||v\_all);

--6.INVALID OBJ

select count(\*) into v\_invalid\_obj from (select 'ALTER '||OBJECT\_TYPE||' '||OWNER||'.'||OBJECT\_NAME||' COMPILE;' from dba\_objects

where object\_type in ('PROCEDURE','FUNCTION','TRIGGER','PACKAGE') and status like 'INVALID'and OWNER like 'CUS\_OWNER'

UNION ALL

select 'ALTER PACKAGE '||OWNER||'.'||OBJECT\_NAME||' COMPILE BODY;' from dba\_objects

where object\_type in ('PACKAGE BODY') and status like 'INVALID' and OWNER like 'CUS\_OWNER');

v\_all := v\_all || 'Invalid Obj:'|| v\_invalid\_obj;

--dbms\_output.put\_line('6.'||v\_all);

send\_sms\_binhtv(v\_all);

--send\_sms\_binhtv('sys.dba\_rp.hc\_sms');

--end if;

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Completed sys.dba\_rp.hc\_sms',1,sysdate,'binhtv.dbamf\_log\_jobs');

commit;

EXCEPTION

WHEN others THEN

send\_sms\_binhtv('Error sys.dba\_rp.hc\_sms: ' || SQLERRM);

v\_err := SUBSTR(SQLERRM, 1, 200);

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Error sys.dba\_rp.hc\_sms',-1,sysdate,'Error sys.dba\_rp.hc\_sms, '||v\_err);

commit;

END;

PROCEDURE session\_sms

is

nActiveSession number;

nTotalActiveSession number; -- Ca background

nTotalInactiveSession number;

nTotalSession number;

nLock number;

v\_err varchar2(1000);

--strBackup varchar2(50);

BEGIN

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Starting sys.dba\_rp.session\_sms',1,sysdate,'binhtv.dbamf\_log\_jobs');

commit;

--1.SESSION

/\*SELECT distinct s.inst\_id i#, s.username, s.SID SID, s.osuser, s.machine,DECODE(S.WAIT\_TIME, 0, S.EVENT, 'CPU') ACTION,

s.sql\_id, SUBSTR(DECODE(SS.SQL\_TEXT, NULL, AA.NAME, SS.SQL\_TEXT), 1, 1000) SQLTEXT,s.logon\_time,s.p1text, S.P1, s.p2text, S.P2, s.p3text, S.P3

FROM GV$SESSION S, GV$SQLSTATS SS, AUDIT\_ACTIONS AA

WHERE S.STATUS = 'ACTIVE' AND S.SQL\_ID = SS.SQL\_ID (+) AND AA.ACTION = S.COMMAND and s.type != 'BACKGROUND' AND S.TYPE = 'USER'

and s.username NOT in ('SYS','SYSMAN','DBSNMP','GGATE','DBAVIETENGATE')

--AND username in 'SYS'

--and DECODE(S.WAIT\_TIME, 0, S.EVENT, 'CPU') like '%cell single block physical read%'

--and lower(ss.sql\_text) like lower('%parallel%')

--and s.sid=1234

--and s.machine like '%BINHTV%'

--and s.sql\_id ='ccwg0nqr1zbu7'

ORDER BY username,sql\_id; \*/

if (to\_char(sysdate,'hh24') in ('07','08','09','10','11','14','15''16','17','18','19','20','21')) then

select count(\*) into nActiveSession from gv$session where status='ACTIVE' and username NOT in ('SYS','SYSMAN','DBSNMP','GGATE','DBAVIETENGATE') and type != 'BACKGROUND' AND TYPE = 'USER' ;

select count(\*) into nTotalActiveSession from gv$session where status='ACTIVE';

select count(\*) into nTotalSession from gv$session;

Select count(\*) into nLock From gv$session where blocking\_session is not NULL and type not like 'BACKGROUND';

send\_sms\_all('DBA\_DBAVIET','DBAVIET\_HeathCheck: ActiveSession ('||nActiveSession || '),TotalActiveSession('||nTotalActiveSession||'),TotalSession('||nTotalSession||'),Lock('||nLock||'),' ||to\_char(sysdate,'dd/mm/yyyy hh24:mi:ss'));

--sys.send\_sms\_ht('DBA\_DBAVIET','DBAVIET\_HeathCheck:' || chr(10) || 'ActiveSession: ('||nActiveSession || ');' || chr(10) ||'TotalActiveSession:('||nTotalActiveSession||');' || chr(10) ||'TotalSession:('||nTotalSession||');' || chr(10) ||'Lock:('||nLock||');' || chr(10) || 'Report time:' ||to\_char(sysdate,'dd/mm/yyyy hh24:mi:ss'));

--send\_sms\_binhtv('DBAVIET\_HeathCheck:' || chr(10) || 'ActiveSession: ('||nActiveSession || ');' || chr(10) ||'TotalActiveSession:('||nTotalActiveSession||');' || chr(10) ||'TotalSession:('||nTotalSession||');' || chr(10) ||'Lock:('||nLock||');' || chr(10) || 'Report time:' ||to\_char(sysdate,'dd/mm/yyyy hh24:mi:ss'));

--dbms\_output.put\_line('DBAVIET\_HeathCheck:' || chr(10) || 'ActiveSession: ('||nActiveSession || ');' || chr(10) ||'TotalActiveSession:('||nTotalActiveSession||');' || chr(10) ||'TotalSession:('||nTotalSession||');' || chr(10) ||'Lock:('||nLock||');' || chr(10) || 'Report time:' ||to\_char(sysdate,'dd/mm/yyyy hh24:mi:ss'));

end if;

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Completed sys.dba\_rp.session\_sms',1,sysdate,'binhtv.dbamf\_log\_jobs');

commit;

EXCEPTION

WHEN others THEN

send\_sms\_binhtv('Error sys.dba\_rp.session\_sms: ' || SQLERRM);

v\_err := SUBSTR(SQLERRM, 1, 200);

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Error sys.dba\_rp.session\_sms',-1,sysdate,'Error sys.dba\_rp.session\_sms, '||v\_err);

commit;

END;

-- Chay cuoi ngay, Bao cao object sai quy dinh, quyen tac dong, ddl

procedure hc\_ddl\_log

is

v\_all varchar2(30000):='';

v\_db\_name varchar2(20):='';

v\_err varchar2(1000):='';

--1.Thong ke all theo user\_name

cursor c\_all

is

select trunc(ddl\_date) "event\_date",count(\*) num from ddl\_log

where ddl\_date >= sysdate-7

--and object\_name not in ('GLOBAL\_B4\_SUB\_USAGE\_ITEM','RP\_VERIFY\_NEW\_SUBS','RP\_CUSTS\_NOT\_PAID\_ACT')

--and owner='CUS\_OWNER'

group by trunc(ddl\_date)

order by trunc(ddl\_date) desc;

--2.Th?ng k?ƒÆ’?†â€™?ƒâ€ ?¢â‚¬â„¢?ƒÆ’?¢â‚¬Â ?ƒÂ¢?¢â€?Â¬?¢â€?Â¢?ƒÆ’?†â€™?ƒÂ¢?¢â€?Â¬?‚Â ?ƒÆ’?‚Â¢?ƒÂ¢?¢â‚¬Å¡?‚Â¬?ƒÂ¢?¢â‚¬Å¾?‚Â¢?ƒÆ’?†â€™?ƒâ€ ?¢â‚¬â„¢?ƒÆ’?‚Â¢?ƒÂ¢?¢â‚¬Å¡?‚Â¬?ƒâ€¦?‚Â¡?ƒÆ’?†â€™?ƒÂ¢?¢â€?Â¬?…Â¡?ƒÆ’?¢â‚¬Å¡?ƒâ€??‚Âª theo user\_name theo ng?ƒÆ’?†â€™?ƒâ€ ?¢â‚¬â„¢?ƒÆ’?¢â‚¬Â ?ƒÂ¢?¢â€?Â¬?¢â€?Â¢?ƒÆ’?†â€™?ƒÂ¢?¢â€?Â¬?‚Â ?ƒÆ’?‚Â¢?ƒÂ¢?¢â‚¬Å¡?‚Â¬?ƒÂ¢?¢â‚¬Å¾?‚Â¢?ƒÆ’?†â€™?ƒâ€ ?¢â‚¬â„¢?ƒÆ’?‚Â¢?ƒÂ¢?¢â‚¬Å¡?‚Â¬?ƒâ€¦?‚Â¡?ƒÆ’?†â€™?ƒÂ¢?¢â€?Â¬?…Â¡?ƒÆ’?¢â‚¬Å¡?ƒâ€??‚Â y CUS\_OWNER

cursor c\_cus\_user

is

select trunc(ddl\_date) "event\_date",user\_name,count(\*) num from ddl\_log

where ddl\_date >= sysdate-1

--and object\_name not in ('GLOBAL\_B4\_SUB\_USAGE\_ITEM','RP\_VERIFY\_NEW\_SUBS','RP\_CUSTS\_NOT\_PAID\_ACT')

and owner='CUS\_OWNER'

group by trunc(ddl\_date),user\_name

order by trunc(ddl\_date) desc, num desc;

--3.Th?ng k?ƒÆ’?†â€™?ƒâ€ ?¢â‚¬â„¢?ƒÆ’?¢â‚¬Â ?ƒÂ¢?¢â€?Â¬?¢â€?Â¢?ƒÆ’?†â€™?ƒÂ¢?¢â€?Â¬?‚Â ?ƒÆ’?‚Â¢?ƒÂ¢?¢â‚¬Å¡?‚Â¬?ƒÂ¢?¢â‚¬Å¾?‚Â¢?ƒÆ’?†â€™?ƒâ€ ?¢â‚¬â„¢?ƒÆ’?‚Â¢?ƒÂ¢?¢â‚¬Å¡?‚Â¬?ƒâ€¦?‚Â¡?ƒÆ’?†â€™?ƒÂ¢?¢â€?Â¬?…Â¡?ƒÆ’?¢â‚¬Å¡?ƒâ€??‚Âª theo user\_name theo ng?ƒÆ’?†â€™?ƒâ€ ?¢â‚¬â„¢?ƒÆ’?¢â‚¬Â ?ƒÂ¢?¢â€?Â¬?¢â€?Â¢?ƒÆ’?†â€™?ƒÂ¢?¢â€?Â¬?‚Â ?ƒÆ’?‚Â¢?ƒÂ¢?¢â‚¬Å¡?‚Â¬?ƒÂ¢?¢â‚¬Å¾?‚Â¢?ƒÆ’?†â€™?ƒâ€ ?¢â‚¬â„¢?ƒÆ’?‚Â¢?ƒÂ¢?¢â‚¬Å¡?‚Â¬?ƒâ€¦?‚Â¡?ƒÆ’?†â€™?ƒÂ¢?¢â€?Â¬?…Â¡?ƒÆ’?¢â‚¬Å¡?ƒâ€??‚Â y CUS\_OWNER tr? ('GLOBAL\_B4\_SUB\_USAGE\_ITEM','RP\_VERIFY\_NEW\_SUBS','RP\_CUSTS\_NOT\_PAID\_ACT')

cursor c\_cus\_user\_min

is

select trunc(ddl\_date) "event\_date",user\_name,count(\*) num from ddl\_log

where ddl\_date >= trunc(sysdate-1)

and object\_name not in ('GLOBAL\_B4\_SUB\_USAGE\_ITEM','RP\_VERIFY\_NEW\_SUBS','RP\_CUSTS\_NOT\_PAID\_ACT')

and owner='CUS\_OWNER'

group by trunc(ddl\_date),user\_name

order by trunc(ddl\_date), num desc;

--4.Thong ke theo object\_type tr? ('GLOBAL\_B4\_SUB\_USAGE\_ITEM','RP\_VERIFY\_NEW\_SUBS','RP\_CUSTS\_NOT\_PAID\_ACT')

-- Chi CUS\_OWNER

cursor c\_cus\_obj\_type

is

select trunc(ddl\_date) "event\_date",object\_type,count(\*) num from ddl\_log

where ddl\_date >= trunc(sysdate-1)

and object\_name not in ('GLOBAL\_B4\_SUB\_USAGE\_ITEM','RP\_VERIFY\_NEW\_SUBS','RP\_CUSTS\_NOT\_PAID\_ACT')

and owner='CUS\_OWNER'

group by trunc(ddl\_date),object\_type

order by trunc(ddl\_date), num desc;

--5..Thong ke theo object\_type tr? ('GLOBAL\_B4\_SUB\_USAGE\_ITEM','RP\_VERIFY\_NEW\_SUBS','RP\_CUSTS\_NOT\_PAID\_ACT')

-- Chi CUS\_OWNER

-- NOT: ('CALLCENTER\_BEA','PAYMENT\_GATEWAY','PAYMENT')

cursor c\_cus\_obj\_type\_min

is

select trunc(ddl\_date) "event\_date",object\_type,count(\*) num from ddl\_log

where ddl\_date >= trunc(sysdate-1)

and object\_name not in ('GLOBAL\_B4\_SUB\_USAGE\_ITEM','RP\_VERIFY\_NEW\_SUBS','RP\_CUSTS\_NOT\_PAID\_ACT')

and owner='CUS\_OWNER'

and user\_name not in ('CALLCENTER\_BEA','PAYMENT\_GATEWAY','PAYMENT')

group by trunc(ddl\_date),object\_type

order by trunc(ddl\_date), num desc;

BEGIN

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Starting sys.dba\_rp.hc\_ddl\_log',1,sysdate,'sys.dba\_rp.hc\_ddl\_log');

commit;

--dbms\_output.put\_line('length:'||length(v\_all));

--if (to\_char(sysdate,'hh24') in ('08','17')) then

select name into v\_db\_name from v$database;

v\_all := v\_all ||'<h2>BAO CAO DDL\_LOG ' || v\_db\_name ||' NGAY ' || to\_char(sysdate,'dd/mm/yyyy hh24:mm:ss') ||'</h2>';

--1.ALL

v\_all:=v\_all||'<br><b>1.DDL LOG TRONG 7 NGAY GAN NHAT:</b><table width="600" border ="1" cellspacing="1" cellpadding="1">

<tr bgcolor="#FFF1D9" style="font-weight:bold">

<td>Ngay thong ke</td>

<td>So luong</td>

</tr>';

for r1 in c\_all loop

v\_all:=v\_all||'<tr>

<td>'||to\_char(r1."event\_date",'dd/mm/yyyy hh24:mm:ss')||'</td>

<td>'||r1.num||'</td>

</tr>' ;

end loop;

v\_all := v\_all || '</table>';

--dbms\_output.put\_line('1.length:'||length(v\_all));

--2.c\_cus\_user

v\_all:=v\_all||'<br><b>2.DDL LOG theo ngay, user\_name theo ng?ƒÆ’?†â€™?ƒâ€ ?¢â‚¬â„¢?ƒÆ’?¢â‚¬Â ?ƒÂ¢?¢â€?Â¬?¢â€?Â¢?ƒÆ’?†â€™?ƒÂ¢?¢â€?Â¬?‚Â ?ƒÆ’?‚Â¢?ƒÂ¢?¢â‚¬Å¡?‚Â¬?ƒÂ¢?¢â‚¬Å¾?‚Â¢?ƒÆ’?†â€™?ƒâ€ ?¢â‚¬â„¢?ƒÆ’?‚Â¢?ƒÂ¢?¢â‚¬Å¡?‚Â¬?ƒâ€¦?‚Â¡?ƒÆ’?†â€™?ƒÂ¢?¢â€?Â¬?…Â¡?ƒÆ’?¢â‚¬Å¡?ƒâ€??‚Â y cua CUS\_OWNER:</b><table width="600" border ="1" cellspacing="1" cellpadding="1">

<tr bgcolor="#FFF1D9" style="font-weight:bold">

<td>Ngay thong ke</td>

<td>User\_name</td>

<td>So luong</td>

</tr>';

for r2 in c\_cus\_user loop

v\_all:=v\_all||'<tr>

<td>'||to\_char(r2."event\_date",'dd/mm/yyyy hh24:mm:ss')||'</td>

<td>'||r2.user\_name||'</td>

<td>'||r2.num||'</td>

</tr>' ;

end loop;

v\_all := v\_all || '</table>';

--dbms\_output.put\_line('2.length:'||length(v\_all));

--3.c\_cus\_user\_min

v\_all:=v\_all||'<br><b>3.DDL LOG theo user\_name theo ng?ƒÆ’?†â€™?ƒâ€ ?¢â‚¬â„¢?ƒÆ’?¢â‚¬Â ?ƒÂ¢?¢â€?Â¬?¢â€?Â¢?ƒÆ’?†â€™?ƒÂ¢?¢â€?Â¬?‚Â ?ƒÆ’?‚Â¢?ƒÂ¢?¢â‚¬Å¡?‚Â¬?ƒÂ¢?¢â‚¬Å¾?‚Â¢?ƒÆ’?†â€™?ƒâ€ ?¢â‚¬â„¢?ƒÆ’?‚Â¢?ƒÂ¢?¢â‚¬Å¡?‚Â¬?ƒâ€¦?‚Â¡?ƒÆ’?†â€™?ƒÂ¢?¢â€?Â¬?…Â¡?ƒÆ’?¢â‚¬Å¡?ƒâ€??‚Â y cua CUS\_OWNER, Tru (GLOBAL\_B4\_SUB\_USAGE\_ITEM,RP\_VERIFY\_NEW\_SUBS,RP\_CUSTS\_NOT\_PAID\_ACT):</b><table width="600" border ="1" cellspacing="1" cellpadding="1">

<tr bgcolor="#FFF1D9" style="font-weight:bold">

<td>Ngay thong ke</td>

<td>User\_name</td>

<td>So luong</td>

</tr>';

for r3 in c\_cus\_user\_min loop

v\_all:=v\_all||'<tr>

<td>'||to\_char(r3."event\_date",'dd/mm/yyyy hh24:mm:ss')||'</td>

<td>'||r3.user\_name||'</td>

<td>'||r3.num||'</td>

</tr>' ;

end loop;

v\_all := v\_all || '</table>';

--dbms\_output.put\_line('3.length:'||length(v\_all));

--4.c\_cus\_obj\_type

v\_all:=v\_all||'<br><b>4.DDL LOG theo object\_type cua CUS\_OWNER, Tru (GLOBAL\_B4\_SUB\_USAGE\_ITEM,RP\_VERIFY\_NEW\_SUBS,RP\_CUSTS\_NOT\_PAID\_ACT):</b><table width="600" border ="1" cellspacing="1" cellpadding="1">

<tr bgcolor="#FFF1D9" style="font-weight:bold">

<td>Ngay thong ke</td>

<td>Object\_type</td>

<td>So luong</td>

</tr>';

for r4 in c\_cus\_obj\_type loop

v\_all:=v\_all||'<tr>

<td>'||to\_char(r4."event\_date",'dd/mm/yyyy hh24:mm:ss')||'</td>

<td>'||r4.object\_type||'</td>

<td>'||r4.num||'</td>

</tr>' ;

end loop;

v\_all := v\_all || '</table>';

--dbms\_output.put\_line('4.length:'||length(v\_all));

--5.c\_cus\_obj\_type\_min

v\_all:=v\_all||'<br><b>4.DDL LOG theo object\_type cua CUS\_OWNER, Tru (GLOBAL\_B4\_SUB\_USAGE\_ITEM,RP\_VERIFY\_NEW\_SUBS,RP\_CUSTS\_NOT\_PAID\_ACT), Tru user\_name tac dong (CALLCENTER\_BEA,PAYMENT\_GATEWAY,PAYMENT):</b><table width="600" border ="1" cellspacing="1" cellpadding="1">

<tr bgcolor="#FFF1D9" style="font-weight:bold">

<td>Ngay thong ke</td>

<td>Object\_type</td>

<td>So luong</td>

</tr>';

for r5 in c\_cus\_obj\_type\_min loop

v\_all:=v\_all||'<tr>

<td>'||to\_char(r5."event\_date",'dd/mm/yyyy hh24:mm:ss')||'</td>

<td>'||r5.object\_type||'</td>

<td>'||r5.num||'</td>

</tr>' ;

end loop;

v\_all := v\_all || '</table>';

--dbms\_output.put\_line('5.length:'||length(v\_all));

cdr\_monitor.send\_email\_html\_m('tranbinh48ca@gmail.com','HC\_' || v\_db\_name ||'\_DDL\_LOG NGAY ' || to\_char(sysdate,'dd/mm/yyyy hh24:mm:ss'), v\_all);

--sys.dba\_rp.send\_sms\_binhtv('sys.dba\_rp.rpt\_ddl\_log');

--end if;

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Completed sys.dba\_rp.hc\_ddl\_log',1,sysdate,'sys.dba\_rp.hc\_ddl\_log');

commit;

EXCEPTION

WHEN others THEN

send\_sms\_binhtv('Error sys.dba\_rp.hc\_ddl\_log: ' || SQLERRM);

v\_err := substr(SQLERRM,1,200);

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Error sys.dba\_rp.hc\_ddl\_log',-1,sysdate,'Error sys.dba\_rp.hc\_ddl\_log, '||v\_err);

commit;

END;

PROCEDURE hc\_fga\_log

is

tAll varchar2(32700):='';

tAll\_New varchar2(10000):='';

tDBName varchar2(20):='';

v\_err varchar2(1000):='';

--1.Tai DB

ndb\_user varchar2(500);

nstatement\_type varchar2(500);

nCount number;

--2.Backup

CURSOR cFGA

IS

select db\_user, statement\_type,count(\*) "Count" from DBA\_FGA\_AUDIT\_TRAIL

where timestamp >= sysdate-1

group by db\_user, statement\_type

order by "Count" desc,statement\_type, db\_user;

tFGA varchar2(30000):='';

BEGIN

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Starting sys.dba\_rp.hc\_fga',1,sysdate,'sys.dba\_rp.hc\_fga');

commit;

select name into tDBName from v$database;

--dbms\_output.put\_line('Before if');

--if (to\_char(sysdate,'hh24') in ('00','08','13')) then

--dbms\_output.put\_line('After if');

select name into tDBName from v$database;

tAll := tAll || '<h2>BAO CAO TRUY CAP CSDL ' || tDBName || ' NGAY ' || to\_char(sysdate,'dd/mm/yyyy hh24:mm:ss') || '</h2>';

--2.BACKUP: Lay 30 ngay

tFGA:=tFGA||'<br><table width="600" border ="1" cellspacing="1" cellpadding="1">

<tr bgcolor="#FFF1D9" style="font-weight:bold">

<td>User\_name</td>

<td>Cau lenh</td>

<td>So luong</td>

</tr>';

for r1 in cFGA loop

tFGA:=tFGA||'<tr>

<td>'||r1.db\_user||'</td>

<td>'||r1.statement\_type||'</td>

<td>'||r1."Count"||' </td>

' ;

end loop;

--tFGA := tFGA || '</table>';

tAll := tAll || tFGA;

--dbms\_output.put\_line('2');

--dbms\_output.put\_line('6');

dbms\_output.put\_line('tAll: '||length(tAll));

--execute immediate 'truncate table tc\_monior\_lob';

--insert into tc\_monior\_lob(text) values(tAll);

--commit;

--select text into tAll\_New from tc\_monior\_lob;

CUS\_OWNER.send\_email\_html\_clob('tranbinh48ca@gmail.com', 'BAO CAO TRUY CAP CSDL ' ||tDBName ||' ' || to\_char(sysdate,'dd/mm/yyyy hh24:mm:ss'), tAll);

--send\_sms\_binhtv('sys.dba\_rp.hc\_backup');

--end if;

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Completed sys.dba\_rp.hc\_fga',1,sysdate,'sys.dba\_rp.hc\_fga');

commit;

EXCEPTION

WHEN others THEN

send\_sms\_binhtv('Error sys.dba\_rp.hc\_fga: ' || SQLERRM);

v\_err := SUBSTR(SQLERRM, 1, 200);

insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)

values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Error sys.dba\_rp.hc\_fga',-1,sysdate,'Error sys.dba\_rp.hc\_fga, ' ||v\_err);

commit;

end;

end;

/

## Tài liệu tham khảo:

# PARTITION

## Tạo bảng partition:

--Thủ tục tạo bảng partition theo ngày trong Oracle Database

--Thủ tục tạo partition theo ngày trong Oracle Database

--drop table oaz20.table1;

create user oaz20 identified by oracle;

grant connect, resource, dba to oaz20;

-- 07.2.1.Tao bang

CREATE TABLE oaz20.table1

(id VARCHAR2(15) NOT NULL,

create\_datetime DATE NOT NULL,

col2 VARCHAR2(20),

col3 NUMBER(10,2)

)

TABLESPACE DATA

PARTITION BY RANGE (create\_datetime)

(

PARTITION DATA20220101 VALUES LESS THAN (TO\_DATE('2022-01-02 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))

TABLESPACE DATA2022

);

--07.2.2.Add them partition

--Add them partition 2022: Bôi đen và ấn F5 --> Vào DBMS Ouput copy nội dung và mở Tab mới để chạy

DECLARE

v\_nam NUMBER (4) := 2022; --2014

v\_tablename VARCHAR2 (50) := 'table1';

v\_date\_from date := to\_date('02/01/2022','dd/mm/yyyy');

v\_date\_to date := to\_date('31/12/2022','dd/mm/yyyy');

v\_numday number(5);

v\_tablespace varchar2(50):='DATA2022';

BEGIN

DBMS\_OUTPUT.ENABLE (buffer\_size => NULL);

v\_numday:=v\_date\_to-v\_date\_from;

FOR i IN 0 .. v\_numday

LOOP

DBMS\_OUTPUT.put\_line ('alter table oaz20.'|| v\_tablename || ' add PARTITION DATA'||to\_char(v\_date\_from+i,'YYYYMMDD')||' VALUES LESS THAN (TO\_DATE('''|| to\_char(v\_date\_from+i+1,'YYYY-MM-DD')||' 00:00:00'', ''SYYYY-MM-DD HH24:MI:SS'', ''NLS\_CALENDAR=GREGORIAN'')) LOGGING TABLESPACE '||v\_tablespace||';');

END LOOP;

END;

-- 07.2.3.Add them partition 2023: Bôi đen rồi F5 --> Vào DBMS Ouput copy nội dung và mở Tab mới để chạy

DECLARE

v\_nam NUMBER (4) := 2023;

v\_tablename VARCHAR2 (50) := 'table1';

v\_date\_from date := to\_date('01/01/2023','dd/mm/yyyy');

v\_date\_to date := to\_date('31/12/2023','dd/mm/yyyy');

v\_numday number(5);

v\_tablespace varchar2(50):='DATA2023';

BEGIN

DBMS\_OUTPUT.ENABLE (buffer\_size => NULL);

v\_numday:=v\_date\_to-v\_date\_from;

FOR i IN 0 .. v\_numday

LOOP

DBMS\_OUTPUT.put\_line ('alter table oaz20.'|| v\_tablename || ' add PARTITION DATA'||to\_char(v\_date\_from+i,'YYYYMMDD')||' VALUES LESS THAN (TO\_DATE('''|| to\_char(v\_date\_from+i+1,'YYYY-MM-DD')||' 00:00:00'', ''SYYYY-MM-DD HH24:MI:SS'', ''NLS\_CALENDAR=GREGORIAN'')) LOGGING TABLESPACE '||v\_tablespace||';');

END LOOP;

END;

--table1

-- Check lai cac partition vua tao

select \* from dba\_tab\_partitions where table\_owner ='OAZ20'

and table\_name='TABLE1'

order by partition\_name;

-- Hoac dua chuot vao table1 an nut F4

## --3.Tao index

select \* from oaz20.table1 where create\_datetime>sysdate-2

and id='1';

-- Go ctrl+P sẽ là explain plan

-- local: index partition

create index oaz20.table1\_I1 on oaz20.table1(id) LOCAL parallel 8 nologging online;

alter index oaz20.table1\_I1 noparallel;

--Ctr+E: Explain Plan

select \* from oaz20.table1 where create\_datetime>sysdate-2

and id='1';

-- global non-partition

create index oaz20.table1\_I2 on oaz20.table1(col2) parallel 8 nologging online;

alter index oaz20.table1\_I2 noparallel;

-- Them du lieu 01/01/2022, 02/01/2022

insert into oaz20.table1 values(5,to\_date('01/01/2022','dd/mm/yyyy'),2,3);

commit;

select \* from oaz20.table1;

alter table oaz20.table1 truncate partition data20220101; -- Khi do index global table1\_i2 unusable

insert into oaz20.table1 values(5,to\_date('01/01/2022','dd/mm/yyyy'),2,3);

commit;

insert into oaz20.table1 values(5,to\_date('02/01/2022','dd/mm/yyyy'),2,3);

commit;

insert into oaz20.table1 values(5,to\_date('03/01/2022','dd/mm/yyyy'),2,3);

commit;

select \* from oaz20.table1;

alter table oaz20.table1 drop partition data20220101;

select \* from oaz20.table1 where col2 = 3;

delete oaz20.table1 where col2=2;

commit;

--oaz20 TABLE1\_I2 UNUSABLE

select owner, index\_name, status from dba\_indexes where index\_name='TABLE1\_I2';

--insert into oaz20.table1 values(5,to\_date('01/05/2021','dd/mm/yyyy'),2,3);

--commit;

--

--update oaz20.table1 set col2='col2';

--commit;

-- Cay index table1\_i2 unusable: DML loi

alter index oaz20.table1\_I2 rebuild online;

select \* from oaz20.table1 where col2 = 3;

select \* from dba\_indexes where owner='OAZ20'

and table\_name='TABLE1';

select \* from dba\_ind\_partitions where index\_owner='OAZ20'

and index\_name like 'TABLE1%'

order by 1,2,4;

insert into oaz20.table1 values(5,to\_date('02/01/2022','dd/mm/yyyy'),2,3);

commit;

alter table oaz20.table1 truncate partition data20220102 update global indexes;

alter table oaz20.table1 drop partition data20220102 update global indexes;

alter table oaz20.table1 drop partition data20220103;

select \* from dba\_ind\_partitions

where index\_owner='OAZ20'

and index\_name='TABLE1\_I1'

--and status!='USABLE';

alter index oaz20.table1\_I1 nologging noparallel;

--3.Rebuild index ve tablespace INDX2022, INDX2023: Bôi đen đoạn dưới và ấn F5

--2022, 2023

DECLARE

v\_date\_from date := to\_date('01/01/2022','dd/mm/yyyy');

v\_date\_to date := to\_date('31/12/2023','dd/mm/yyyy');

v\_numday number;

v\_tablespace varchar2(50):='INDX';

cursor c1 is

select a.\* from DBA\_PART\_INDEXES a, DBA\_TAB\_PARTITIONS b where a.owner=B.TABLE\_OWNER and a.table\_name=B.TABLE\_NAME and b.table\_owner='OAZ20' and b.table\_name='TABLE1' and a.index\_name not like '%$%' and b.partition\_name like '%20231231' order by a.owner,a.index\_name;

BEGIN

v\_numday:=v\_date\_to-v\_date\_from;

FOR i1 in c1

LOOP

FOR i IN 0 .. v\_numday

LOOP

DBMS\_OUTPUT.put\_line ('alter index '||i1.owner||'.'||i1.index\_name || ' REBUILD PARTITION DATA'||to\_char(v\_date\_from+i,'YYYYMMDD')||' TABLESPACE '||v\_tablespace||to\_char(v\_date\_from+i,'YYYY')||' nologging parallel 8 online;');

END LOOP;

END LOOP;

END;

-- Set nologging noparallel

DECLARE

cursor c1 is

select distinct a.index\_owner, a.index\_name from DBA\_ind\_partitions a where a.index\_owner='OAZ20' and a.index\_name not like '%$%' and a.partition\_name like '%20231231'

order by a.index\_owner,a.index\_name;

BEGIN

FOR i1 in c1

LOOP

DBMS\_OUTPUT.put\_line ('alter index '||i1.index\_owner||'.'||i1.index\_name || ' nologging noparallel;');

END LOOP;

END;

alter index oaz20.TABLE1\_I1 nologging noparallel;

## Tạo dữ liệu 1 triệu row theo ngày

--drop table bang\_to purge;

-oaz18

CREATE TABLE oaz18.bang\_to (

bang\_toID int,

LastName varchar(255),

FirstName varchar(255),

Address varchar(255),

City varchar(255),

create\_date date

)

TABLESPACE DATA

PARTITION BY RANGE (create\_date)

(

PARTITION DATA20230101 VALUES LESS THAN (TO\_DATE('2023-01-02 00:00:00', 'YYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))

TABLESPACE DATA2023);

-- Add partition het nam 2023, boi den xong an F5

DECLARE

v\_nam NUMBER (4) := 2023;

v\_owner varchar2 (50) := 'oaz18';

v\_tablename VARCHAR2 (50) := 'bang\_to';

v\_date\_from date := to\_date('02/01/2023','dd/mm/yyyy');

v\_date\_to date := to\_date('31/12/2023','dd/mm/yyyy');

v\_numday number(5);

v\_tablespace varchar2(50):='DATA';

BEGIN

v\_numday:=v\_date\_to-v\_date\_from;

FOR i IN 0 .. v\_numday

LOOP

DBMS\_OUTPUT.put\_line ('alter table '||v\_owner||'.'|| v\_tablename || ' add PARTITION DATA' ||to\_char(v\_date\_from+i,'YYYYMMDD')||' VALUES LESS THAN (TO\_DATE('''|| to\_char(v\_date\_from+i+1,'YYYY-MM-DD')||' 00:00:00'',''SYYYY-MM-DD HH24:MI:SS'', ''NLS\_CALENDAR=GREGORIAN'')) LOGGING TABLESPACE DATA'|| to\_char(v\_date\_from+i,'YYYY')||';');

END LOOP;

END;

-- Them du lieu deu co cac partition theo ngay

declare

v\_commit number;

v\_reset\_date number;

v\_start date := to\_date('01/01/2023','dd/mm/yyyy');

BEGIN

FOR v\_LoopCounter IN 1..1000000 LOOP

if v\_start = to\_date('31/12/2023','dd/mm/yyyy') then

v\_start := to\_date('01/01/2023','dd/mm/yyyy');

end if;

INSERT INTO bang\_to (bang\_toID, LastName, FirstName, Address, City,create\_date)

VALUES (TO\_CHAR(v\_LoopCounter),'Name'||v\_LoopCounter,'FirstName'||v\_LoopCounter,'HANOI','HANOI',v\_start);

v\_start := v\_start+1;

v\_commit := v\_commit + 1;

if v\_commit=1000 then

commit;

v\_commit :=1;

end if;

END LOOP;

END;

--Them 10.000 ro cho ngay hien tai (vd 20/06/2023)

BEGIN

FOR v\_LoopCounter IN 1..100000 LOOP

INSERT INTO bang\_to (bang\_toID, LastName, FirstName, Address, City, create\_date)

VALUES (TO\_CHAR(v\_LoopCounter),'Name'||v\_LoopCounter,'FirstName'||v\_LoopCounter,'HANOI','HANOI', sysdate);

END LOOP;

COMMIT;

END;

--Them 10.000 ro cho ngay hien tai (vd 20/06/2023)

BEGIN

FOR v\_LoopCounter IN 1..1000 LOOP

INSERT INTO bang\_to (bang\_toID, LastName, FirstName, Address, City, create\_date)

VALUES (TO\_CHAR(v\_LoopCounter),'Name'||v\_LoopCounter,'FirstName'||v\_LoopCounter,'HANOI','HANOI', sysdate);

END LOOP;

COMMIT;

END;

select \* from bang\_to;

Plan

SELECT STATEMENT ALL\_ROWSCost: 99,273 Bytes: 56,610,000 Cardinality: 1,110,000

2 PARTITION RANGE ALL Cost: 99,273 Bytes: 56,610,000 Cardinality: 1,110,000 Partition #: 1 Partitions accessed #1 - #365

1 TABLE ACCESS FULL TABLE OAZ18.BANG\_TO Cost: 99,273 Bytes: 56,610,000 Cardinality: 1,110,000 Partition #: 1 Partitions accessed #1 - #365

-- TAO GLOBAL INDEX

create index bang\_to\_ind1 on bang\_to(bang\_toID) tablespace INDX;

select \* from bang\_to where bang\_toID=1234;

Plan

SELECT STATEMENT ALL\_ROWSCost: 6

2 TABLE ACCESS BY GLOBAL INDEX ROWID BATCHED TABLE OAZ18.BANG\_TO Cost: 6 Bytes: 153 Cardinality: 3 Partition #: 1 Partition access computed by row location

1 INDEX RANGE SCAN INDEX OAZ18.BANG\_TO\_IND1 Cost: 3 Cardinality: 3

alter table bang\_to truncate partition data20230101;

--Sau do index bang\_to\_ind1 unusable do la global index

select \* from bang\_to where bang\_toID=1234;

Plan

SELECT STATEMENT ALL\_ROWSCost: 98,999

2 PARTITION RANGE ALL Cost: 98,999 Bytes: 153 Cardinality: 3 Partition #: 1 Partitions accessed #1 - #365

1 TABLE ACCESS FULL TABLE OAZ18.BANG\_TO Cost: 98,999 Bytes: 153 Cardinality: 3 Partition #: 1 Partitions accessed #1 - #365

-- Xu ly

alter index bang\_to\_ind1 rebuild online;

-- Muon khong bi unusable phai update global index

alter table bang\_to truncate partition data20230102 update global indexes;

-- TAO LOCAL INDEX

create index bang\_to\_ind1 on bang\_to(bang\_toID) tablespace INDX local parallel 8 online ;

--Rebuild partition index ve tablespace tuong ung

DECLARE

v\_date\_from date := to\_date('01/01/2023','dd/mm/yyyy');

v\_date\_to date := to\_date('31/12/2023','dd/mm/yyyy');

v\_numday number;

v\_tablespace varchar2(50):='INDX';

cursor c1 is

select a.\* from DBA\_PART\_INDEXES a, DBA\_TAB\_PARTITIONS b where a.owner=B.TABLE\_OWNER and a.table\_name=B.TABLE\_NAME and b.table\_owner='OAZ18' and b.table\_name='BANG\_TO' and a.index\_name not like '%$%' and b.partition\_name like '%20231231' order by a.owner,a.index\_name;

BEGIN

v\_numday:=v\_date\_to-v\_date\_from;

FOR i1 in c1

LOOP

FOR i IN 0 .. v\_numday

LOOP

DBMS\_OUTPUT.put\_line ('alter index '||i1.owner||'.'||i1.index\_name || ' REBUILD PARTITION DATA'||to\_char(v\_date\_from+i,'YYYYMMDD')||' TABLESPACE '||v\_tablespace||to\_char(v\_date\_from+i,'YYYY')||' nologging parallel 8 online;');

END LOOP;

END LOOP;

END;

select \* from bang\_to where bang\_toID=1234 and create\_date>=to\_date('21/06/2023','dd/mm/yyyy') and create\_date<to\_date('22/06/2023','dd/mm/yyyy');

select \* from bang\_to where lastname like 'Binh%' and create\_date>=to\_date('21/06/2023','dd/mm/yyyy') and create\_date<to\_date('22/06/2023','dd/mm/yyyy');

Plan

SELECT STATEMENT ALL\_ROWSCost: 276 Bytes: 50 Cardinality: 1

2 PARTITION RANGE SINGLE Cost: 276 Bytes: 50 Cardinality: 1 Partition #: 1 Partitions accessed #172

1 TABLE ACCESS FULL TABLE OAZ18.BANG\_TO Cost: 276 Bytes: 50 Cardinality: 1 Partition #: 2 Partitions accessed #172

create index OAZ18.IDX$$\_00220001 on OAZ18.BANG\_TO("LASTNAME") tablespace INDX local parallel 8 online;

alter index OAZ18.IDX$$\_00220001 noparallel ;

-- Cau lenh quet ko co partiiton\_key

select \* from bang\_to where lastname like 'Binh%';

Plan

SELECT STATEMENT ALL\_ROWSCost: 98,730 Bytes: 153 Cardinality: 3

2 PARTITION RANGE ALL Cost: 98,730 Bytes: 153 Cardinality: 3 Partition #: 1 Partitions accessed #1 - #365

1 TABLE ACCESS FULL TABLE OAZ18.BANG\_TO Cost: 98,730 Bytes: 153 Cardinality: 3 Partition #: 1 Partitions accessed #1 - #365

--Khuyen nghi Oracle

create index OAZ18.IDX$$\_001F0001 on OAZ18.BANG\_TO("LASTNAME");

select \* from bang\_to where lastname like 'Binh%' and create\_date>=to\_date('21/05/2023','dd/mm/yyyy') and create\_date<to\_date('22/06/2023','dd/mm/yyyy');

## Tài liệu tham khảo:

<https://www.tranvanbinh.vn/2020/12/quan-ly-bang-partitionfull.html>

<https://www.tranvanbinh.vn/2019/12/lam-nao-tao-partion-het-nam-2020-cho-ca.html>

<https://www.tranvanbinh.vn/2020/12/tao-bang-hash-partition.html>

<https://www.tranvanbinh.vn/2020/08/gather-analyze-co-so-du-lieu-oracle.html>

<https://www.tranvanbinh.vn/2020/01/quan-tri-co-so-du-lieu-oracle-9van-hanh.html>

<https://www.tranvanbinh.vn/2019/11/quan-ly-du-lieu-lob-rat-phuc-tap-nhung.html>

<https://www.tranvanbinh.vn/2019/12/lam-nao-tao-partion-het-nam-2020-cho-ca.html>

<https://tranvanbinh.vn/2020/12/thu-tuc-tao-bang-partition-theo-ngay.html>

<https://www.tranvanbinh.vn/2021/04/tao-partition-tu-ong-voi-tinh-nang.html>

<https://www.tranvanbinh.vn/2019/11/chuyen-bang-tu-non-partition-sang.html>

<https://www.tranvanbinh.vn/2020/12/script-add-partition-tu-ong.html+>

<https://www.tranvanbinh.vn/2020/12/chuyen-bang-non-partition-sang.html>

# INDEX

## Tài liệu tham khảo:

<https://www.tranvanbinh.vn/2020/11/quan-ly-index.html>

<https://www.tranvanbinh.vn/2022/02/hieu-ve-index-partition-phan-biet-local.html>

<https://www.tranvanbinh.vn/2020/12/co-che-lam-viec-cua-index.html>

<https://www.tranvanbinh.vn/2019/11/khi-nao-can-tao-index.html>

<https://www.tranvanbinh.vn/2021/02/tao-index-bang-core-bang-cuc-lon-2-15.html>

# ORACLE RAC

## Cài đặt:

<https://www.tranvanbinh.vn/2022/07/thu-tuc-cai-at-oracle-rac-19c-tren.html>

<https://www.tranvanbinh.vn/2021/02/cai-at-oracle-database-12c-release-2.html>

## 1. Startup/Shutdown Clusterware Oracle RAC

Hệ thống Clusterware sẽ được khởi động cùng với quá trình hệ điều hành khởi động, tuy nhiên có nhiều trường hợp người quản trị sẽ phải khởi động Clusterware bằng tay. Khi đó thực hiện như sau:

Login vào máy chủ database với user “root” trên từng node cần startup Clusterware, chạy lệnh sau:

$ cd /u01/app/12.2.0/grid/bin

* **Startup Clusterware:**

$ ./crsctl start crs

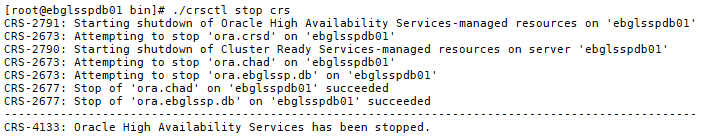
Kết quả trả về “*Oracle High Availability Services has been started*” là startup clusterware thành công



* Để Shutdown Clusterware:

$./crsctl stop crs -f

Kết quả trả về “*Oracle High Availability Services has been stopped*” là startup clusterware thành công



## 2. Startup/Shutdown Oracle database

Login vào máy chủ database với user “oracle”, chạy lệnh sau:

$ cd $ORACLE\_HOME/bin

* **Startup database:**
* Chạy lệnh theo cú pháp sau để startup tất cả Oracle RAC instances cho 1 database, thay tham số <**DB name**> tương ứng theo mục “4. Danh sách máy chủ và quản trị”

$ srvctl start database -d <**DB name**>

* Kiểm tra trạng thái của database sau khi bật, chạy lệnh theo cú pháp. Trạng thái trả về “*running on*” là thành công

$ srvctl status database -d <**DB name>**

**Ví dụ:** Với database CRM



* Hoặc truy nhập vào tất cả các máy chủ của database đó và chạy lệnh sau với kết quả trả về “*Oracle instance started*”

$ sqlplus /as sysdba

SQL>startup



* **Startup 1 instances:**
* Chạy lệnh theo cú pháp sau và thay <**DB Name**>, <**Instance Name**> giá trị tương ứng:

$ srvctl start instance -d <**DB name**> -i <**instance name**>

* Kiểm tra trạng thái của database sau khi bật, chạy lệnh theo cú pháp. Trạng thái trả về “*running on*” là thành công

$ srvctl status instance -d <**DB name> -**i <**instance name**>

**Ví dụ:** Với database: ebglssp và instance: ebglssp1



* Hoặc chạy lệnh sau, với kết quả trả về “*Oracle instance started*”

$ sqlplus /as sysdba

SQL> startup



* **Shutdown tất cả Oracle instances:**
* Chạy lệnh theo cú pháp sau và thay **<DB Name>** giá trị tương ứng:

$ srvctl stop database -d <**DB name**>

* Kiểm tra trạng thái của database, chạy lệnh theo cú pháp. Trạng thái trả về “*is not running on*” là thành công

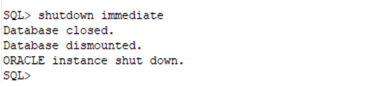
$ srvctl status database -d <**DB name>**



* Hoặc truy nhập vào tất cả các máy chủ của database đó và chạy lệnh, với kết quả trả về “*Oracle instance shut down*” là thành công

$sqlplus / as sysdba

SQL> shutdown immediate;



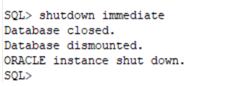
* **Shutdown 1 instance:**
* Chạy lệnh theo cú pháp sau và thay **<DB Name>, <Instance Name>** giá trị tương ứng:

$ srvctl stop instance -d <**DB\_name**> -i <**instance\_name**>

* Kiểm tra trạng thái của database, chạy lệnh theo cú pháp. Trạng thái trả về “*is not running on*” là thành công
* Hoặc chạy lệnh, với kết quả trả về “*Oracle instance shut down*” là thành công

$sqlplus / as sysdba

SQL>Shutdown immediate



## 3. Startup/Shutdown ASM

Login vào máy chủ database với user “grid”, chạy lệnh sau:

$ cd $ORACLE\_HOME/bin

* Startup ASM:
* Chạy lệnh sau:

$ srvctl start asm

* Để kiểm tra kết quả ta chạy lệnh theo sau với kết quả trả về “*ASM is running on*” là thành công

$ srvctl status asm

**Ví dụ:**



* Hoặc chạy lệnh sau, với kết quả trả về “*ASM diskgroups mounted*” là thành công

$ sqlplus /as sysasm

SQL>startup

SQL> startup

ASM instance started

Total System Global Area 1140850688 bytes

Fixed Size 8629704 bytes

Variable Size 1107055160 bytes

ASM Cache 25165824 bytes

ASM diskgroups mounted

* Startup ASM cho 1 instance:
* Chạy lệnh sau:

$ srvctl start asm -node <server\_name>

**Ví dụ:**

$ srvctl start asm -node csdl01

* Để kiểm tra kết quả ta chạy lệnh theo sau với kết quả trả về “*ASM is running on”* là thành công



* Hoặc chạy lệnh sau:

$ sqlplus /as sysasm

SQL>startup

* Shutdown ASM:
* Chạy lệnh:

$ srvctl stop asm

* Để kiểm tra kết quả ta chạy lệnh theo sau với kết quả trả về “*ASM is not running*” là thành công
* Hoặc truy nhập vào tất cả các máy chủ của database đó và chạy lệnh:

$ sqlplus /as sysasm

SQL>shutdown immediate

*ASM diskgroups dismounted*

*ASM instance shutdown*

* Shutdown 1 ASM instance:
* Chạy lệnh sau:

$ srvctl stop asm -node <server\_name>

**Ví dụ:**

$ srvctl stop asm -node cdl01

* Hoặc có thể chạy lệnh sau:

$ sqlplus /as sysasm

SQL>shutdown immediate

*ASM diskgroups dismounted*

*ASM instance shutdown*

## 4. Start/Stop Oracle Listener

Login vào máy chủ database với user “grid”, chạy các lệnh sau:

* Start listener của toàn bộ database:
* Chạy lệnh theo cú pháp sau và thay tham số tương ứng

$ srvctl start listener

* Kiểm tra kết quả ta dùng lệnh sau với kết quả trả về: “*LISTENER is running on node*” là thành công

$ srvctl status listener

* Hoặc vào từng máy chủ database chạy lệnh sau, với kết quả trả về “*The command completed successfully*” là thành công

$ lsnrctl start

* Start listener trên 1 instance:
* Chạy lệnh theo cú pháp sau và thay tham số tương ứng

$ srvctl start listener -n <**DB Server**>

**Ví dụ** :

$ srvctl start listener -n csdl01

* Kiểm tra kết quả ta dùng lênh sau với kết quả trả về: “LISTENER is running on node”

$ srvctl status listener

* Hoặc chạy lệnh sau:

$ lsnrctl start

* Stop listener của toàn bộ database:
* Chạy lệnh theo cú pháp sau và thay tham số tương ứng

$ srvctl stop listener

* Kiểm tra kết quả ta dùng lênh sau với kết quả trả về: “*LISTENER is not running*” là thành công

$ srvctl status listener

* Hoặc vào từng máy chủ database chạy lệnh sau với kết quả trả về “*The command completed successfully*” là thành công

$ lsnrctl stop

* Stop listener trên 1 instance:
* Chạy lệnh theo cú pháp sau và thay tham số tương ứng

$ srvctl stop listener -n <DB Server>

**Ví dụ :**

$ srvctl stop listener -n csdl01

* Kiểm tra kết quả ta dùng lệnh sau với kết quả trả về: “*LISTENER is running on node*” là thành công
* Hoặc chạy lệnh sau:

$ lsnrctl stop

## 5. Start/Stop Oracle Resource

Login vào máy chủ database với user “root”, chạy các lệnh sau:

* **Start Resource:**

$ cd /u01/app/12.2.0/grid/bin

* Thực hiện lệnh theo cú pháp sau và thay tham số tương ứng:

$ ./ crsctl start resource <resource> -n <**db name**>

* Trường hợp muốn start toàn bộ các resource

$ ./ crsctl start resource -all

* Kiểm tra kết quả ta dùng lệnh sau với kết quả trả về: “*ONLINE”* là thành công

$ ./ crsctl status resource

* **Stop Resource:**
* Thực hiện lệnh theo cú pháp sau và thay tham số tương ứng:

$ ./ crsctl stop resource <resource> -n <db name>

* Trường hợp muốn start toàn bộ các resource

$ ./ crsctl stop resource -all

* Kiểm tra kết quả ta dùng lệnh sau với kết quả trả về: “*OFFLINE”* là thành công

$ ./ crsctl status resource

## Tài liệu tham khảo:

<https://www.tranvanbinh.vn/2022/07/thu-tuc-cai-at-oracle-rac-19c-tren.html>

<https://www.tranvanbinh.vn/2021/02/cai-at-oracle-database-12c-release-2.html>

<https://www.tranvanbinh.vn/p/tong-hop-cac-bai-viet-ve-oracle-rac.html>

<https://www.tranvanbinh.vn/2023/06/oracle-rac-la-gi-va-kien-truc-oracle-rac.html>

<https://www.tranvanbinh.vn/2022/07/cac-giai-phap-luu-tru-cho-oracle-rac.html>

# ORACLE DATAGUARD

## Cài đặt:

<https://www.tranvanbinh.vn/2021/08/cai-at-cau-hinh-quan-tri-oracle-data.html>

## 1. HƯỚNG DẪN GIÁM SÁT, BẬT, TẮT DATAGUARD

**Giám sát**

**Sử dụng Oracle Broker kiểm tra trạng thái cấu hình DataGuard:**

dgmgrl /

show configuration;

show database verbose dbaviet

**Hoặc sử dụng SQL kiểm tra apply lag trên Standby:**

SELECT NAME, VALUE FROM V$DATAGUARD\_STATS WHERE NAME='apply lag';

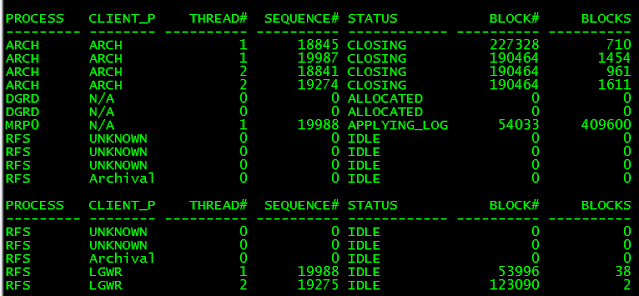
[](https://lh3.googleusercontent.com/-nCCLo1fDh5c/YGrJbyO-pRI/AAAAAAAAA3Q/hTyAV2zFoVc8uR3z_iHmqOzS7-XIt0zSQCLcBGAsYHQ/image.png)

Nếu lag =0 thì OK, còn > 5 phút cần check lại.

**Kiểm tra trạng thái các process sử dụng cho đồng bộ dữ liệu trên Standby**

ALTER SESSION SET nls\_date\_format='DD-MON-YYYY HH24:MI:SS'

SELECT PROCESS, CLIENT\_PROCESS, THREAD#, SEQUENCE#, STATUS, BLOCK#, BLOCKS FROM GV$MANAGED\_STANDBY order by PROCESS, thread#, SEQUENCE#;

[](https://lh3.googleusercontent.com/-mx5ZXhvDb74/YGrJpSr0OMI/AAAAAAAAA3U/d-RGSDrUrloXqY2w77vDNPcnNrJWDm1TACLcBGAsYHQ/image.png)

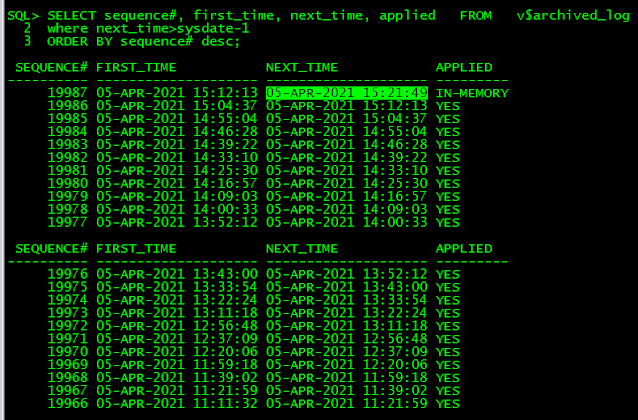
**Check apply archived log:**

SQL> !date

Mon Apr  5 15:23:15 +07 2021

SELECT sequence#, first\_time, next\_time, applied  FROM   v$archived\_log

where next\_time>sysdate-1 ;

[](https://lh3.googleusercontent.com/-dIsOCuJ9ZTQ/YGrKBsGQ2zI/AAAAAAAAA3g/1LHezXWp0uwbwWwwjFWXH3uWKCxgif5HQCLcBGAsYHQ/image.png)

(Như này là real time, sequence cuối cùng 19987 là IN-MEMORY, còn nếu nhiều sequence mà APPLIED=NO là GAP (trễ) ở đó)

***-- Đảm bảo không có row nào***

SELECT NAME, thread#, sequence#, next\_time, first\_change#, creator, fal,

 registrar, archived, applied, deleted,

 TRUNC (blocks \* block\_size / 1024 / 1024) || 'M' arc

 FROM v$archived\_log

 WHERE thread# = 1 AND sequence# >= (SELECT MAX (sequence#)

 FROM v$archived\_log

 WHERE applied = 'YES' AND thread# = 1)

 and dest\_id=2

UNION ALL

SELECT NAME, thread#, sequence#, next\_time, first\_change#, creator, fal,

 registrar, archived, applied, deleted,

 TRUNC (blocks \* block\_size / 1024 / 1024) || 'M' arc

 FROM v$archived\_log

 WHERE thread# = 2 AND sequence# >= (SELECT MAX (sequence#)

 FROM v$archived\_log

 WHERE applied = 'YES' AND thread# = 2)

  and dest\_id=2

ORDER BY thread#, sequence# DESC;

**Kiểm tra transfer được archive log từ Primary sang Standby**

**§ Primary:** Tạo log switch

ALTER SESSION SET nls\_date\_format='DD-MON-YYYY HH24:MI:SS';

-- Kiểm tra sequence# mới nhất: ví dụ **1131595** 1/29/2021 2:16:20 PM 1/29/2021 3:32:26 PM

SELECT sequence#, first\_time, next\_time

FROM   v$archived\_log;

ALTER SYSTEM SWITCH LOGFILE;

-- Kiểm tra sequence# hiện tại mới sinh ra: ví dụ **1131596** 1/29/2021 3:32:26 PM 1/29/2021 4:47:06 PM

SELECT sequence#, first\_time, next\_time

FROM   v$archived\_log;

**§ Standby:**Kiểm tra xem archive log mới sinh ra đã sang standby và đã apply chưa

ALTER SESSION SET nls\_date\_format='DD-MON-YYYY HH24:MI:SS';

-- Nếu equence# là **131596**thì OK

SELECT sequence#, first\_time, next\_time

FROM   v$archived\_log;

**Note**: Nếu không transfer được archive log từ primary sang standby hãy kiểm tra:

- Đường mạng giữa primary & standby

- Listener trên standby

- Standby có đang mount hoặc open hay không.

- Vùng chứa archive log (+RECO) trên standby có bị đầy hay không

**2.START/STOP**

**2.1. Start/Stop broker**

alter system set dg\_broker\_start=true sid='\*'

alter system set dg\_broker\_start=false sid='\*';

2.2.**Start/Stop chế độ apply trên Primary**

**Cách 1: Dùng Broker**

***--Khởi động truyền log/archive trên 1 node database dbaviet***

dgmgrl /

edit database dbavietset state='TRANSPORT-ON';

***--Dừng truyền log/archive trên database dbaviet***

dgmgrl /

edit database dbavietset state='TRANSPORT-OFF';

Check lai:

DGMGRL> SHOW DATABASE dbaviet;

**Cách 2: Dùng SQL**

-- dbaviet

SQL> show parameter archive\_dest\_state\_2

-- Enable

ALTER SYSTEM SET log\_archive\_dest\_state\_2='ENABLE' SCOPE=MEMORY SID='\*';

--Disable đẩy archivelog sang dest\_2

ALTER SYSTEM SET log\_archive\_dest\_state\_2='DEFER' SCOPE=BOTH SID='\*';

Khi tắt sẽ có log:

*Fri Jul 16 10:31:10 2021*

*FAL[server, ARC3]: FAL archive failed, see trace file.*

*ARCH: FAL archive failed. Archiver continuing*

*ORACLE Instance dbaviet\_dg1 - Archival Error. Archiver continuing.*

*FAL[server, ARC3]: FAL archive failed, see trace file.*

*ARCH: FAL archive failed. Archiver continuing*

*ORACLE Instance dbaviet\_dg1 - Archival Error. Archiver continuing.*

*FAL[server, ARC3]: FAL archive failed, see trace file.*

*ARCH: FAL archive failed. Archiver continuing*

*ORACLE Instance dbaviet\_dg1 - Archival Error. Archiver continuing.*

*FAL[server, ARC3]: FAL archive failed, see trace file.*

**2.3. Start/Stop chế độ apply trên standby**

**-- Cách 1: Dùng Broker**

1. Bật apply trên database dbaviet\_standby

dgmgrl /

edit database dbaviet\_standbyset state='APPLY-ON';

show database dbaviet\_standby

2. Dừng apply trên database dbaviet\_standby

dgmgrl /

edit database dbaviet\_standbyset state='APPLY-OFF';

**-- Cách 2: Dùng SQL**

-- Start

alter database recover managed standby database using current logfile disconnect;

hoặc:

alter database recover managed standby database disconnect from session;

-- Stop

alter database recover managed standby database cancel;

**2.4. Tự động, manual thêm datafile**

alter system set standby\_file\_management=AUTO;

alter system set standby\_file\_management=MANUAL;

## 2. Cách xử lý GAPS trong standby database (Data Guard) sử dụng backup RMAN incremental

Các bước trong bài đăng này có thể được sử dụng để giải quyết sự cố nếu cơ sở dữ liệu physical sandby bị mất hoặc bị hỏng archived redo data hoặc thiếu archived log (gap)1. Đảm bảo rằng bạn dừng tiên trình khôi phục được quản lý (MRP) trước khi tiếp tục các bước tiếp theo.

-- STANBY

SQL> **ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;**

2. Trên **STANDBY DATABASE (cơ sở dữ liệu dự phòng)** , tìm SCN sẽ được sử dụng để incremental backup tại cơ sở dữ liệu chính (Primary Database). Bạn cần sử dụng 'SCN thấp nhất' từ 3 truy vấn dưới đây:

--STANDBY

SQL> **SELECT CURRENT\_SCN FROM V$DATABASE;**

CURRENT\_SCN

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

3164433

SQL> **select min(fhscn) from x$kcvfh;**

MIN(FHSCN)

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

**3162298**

SQL> **select min(f.fhscn) from x$kcvfh f, v$datafile d**

**where f.hxfil =d.file#**

**and d.enabled != 'READ ONLY' ;**

MIN(F.FHSCN)

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

**3162298**

3. Bạn cần sử dụng ' **SCN thấp nhất** ' từ 3 truy vấn, ở đây là -> SCN: **3162298**. Trong RMAN, hãy kết nối với cơ sở dữ liệu **PRIMARY**và tạo bản incremental backup từ SCN được dẫn xuất ở bước trước:

--PRIMARY

RMAN> **BACKUP INCREMENTAL FROM SCN 3162298 DATABASE FORMAT '/tmp/ForStandby\_%U' tag 'FORSTANDBY';**

4. Chuyển tất cả các bộ backup được tạo trên hệ thống chính sang hệ thống dự phòng.

--PRIMARY

primary $ **scp /tmp/ForStandby\_\* standby:/tmp**

Trên  **STANDBY**catalog các backup pieces để nó nhận được bản backup (thông tin này được ghi vào control file của Standby database):

--STANDBY

RMAN> **CATALOG START WITH '/tmp/ForStandby';**

using target database control file instead of recovery catalog

searching for all files that match the pattern /tmp/ForStandby

List of Files Unknown to the Database

=====================================

File Name: /tmp/ForStandby\_2lkglss4\_1\_1

File Name: /tmp/ForStandby\_2mkglst8\_1\_1

Do you really want to catalog the above files (enter YES or NO)? YES

cataloging files...

cataloging done

List of Cataloged Files

=======================

File Name: /tmp/ForStandby\_2lkglss4\_1\_1

File Name: /tmp/ForStandby\_2mkglst8\_1\_1

5. Khôi phục cơ sở dữ liệu **STANDBY** với incremental backup đã được catalog:

--STANDBY

RMAN> **RECOVER DATABASE NOREDO;**

starting recover at 03-JUN-09

allocated channel: ORA\_DISK\_1

channel ORA\_DISK\_1: sid=28 devtype=DISK

channel ORA\_DISK\_1: starting incremental datafile backupset restore

channel ORA\_DISK\_1: specifying datafile(s) to restore from backup set

destination for restore of datafile 00001: +DATA/mystd/datafile/system.297.688213333

destination for restore of datafile 00002: +DATA/mystd/datafile/undotbs1.268.688213335

destination for restore of datafile 00003: +DATA/mystd/datafile/sysaux.267.688213333

channel ORA\_DISK\_1: reading from backup piece /tmp/ForStandby\_2lkglss4\_1\_1

channel ORA\_DISK\_1: restored backup piece 1

piece handle=/tmp/ForStandby\_2lkglss4\_1\_1 tag=FORSTANDBY

channel ORA\_DISK\_1: restore complete, elapsed time: 00:00:02

Finished recover at 03-JUN-09

6. Trong RMAN, kết nối với cơ sở dữ liệu **PRIMARY**và tạo bản sao lưu standby control file:

--PRIMARY

RMAN> **BACKUP CURRENT CONTROLFILE FOR STANDBY FORMAT '/tmp/ForStandbyCTRL.bck';**

7. Sao chép bản sao standby control file vào hệ thống **STANDBY** .

--PRIMARY

primary $ **scp /tmp/ForStandbyCTRL.bck standby:/tmp**

8. Sao lưu vị trí của datafile ở**standby** trong trường hợp tên / vị trí datafile khác với tên / vị trí tệp Primary.

--standby

SQL> spool datafile\_names\_step8.txt

set lines 200

col name format a60

**select file#, name from v$datafile order by file# ;**

spool off

9. Từ RMAN, kết nối với cơ sở dữ liệu **STANDBY** và khôi phục standby control file:

--STANDBY

RMAN> **SHUTDOWN IMMEDIATE ;**

RMAN> **STARTUP NOMOUNT;**

RMAN> **RESTORE STANDBY CONTROLFILE FROM '/tmp/ForStandbyCTRL.bck';**

Starting restore at 03-JUN-09

using target database control file instead of recovery catalog

allocated channel: ORA\_DISK\_1

channel ORA\_DISK\_1: sid=36 devtype=DISK

channel ORA\_DISK\_1: restoring control file

channel ORA\_DISK\_1: restore complete, elapsed time: 00:00:07

output filename=+DATA/mystd/controlfile/current.257.688583989

Finished restore at 03-JUN-09

10. Tắt cơ sở dữ liệu **STANDBY** và startup mount:

--STANDBY

SQL> **SHUTDOWN;**

SQL> **STARTUP MOUNT;**

11. Vì control file được khôi phục từ PRIMARY, các vị trí datafile trong STANDBY controlfile  sẽ giống với cơ sở dữ liệu PRIMARY, vì vậy các catalog datafile trong STANDBY sẽ thực hiện các thao tác đổi tên cần thiết. Thực hiện bước dưới đây trong STANDBY cho mỗi diskgroup trong đó cấu trúc thư mục datafile giữa chế độ primary và chế độ standby là khác nhau.

--STANDBY

RMAN> **CATALOG START WITH '+DATA/mystd/datafile/';**

List of Files Unknown to the Database

=====================================

File Name: +data/mystd/DATAFILE/SYSTEM.309.685535773

File Name: +data/mystd/DATAFILE/SYSAUX.301.685535773

File Name: +data/mystd/DATAFILE/UNDOTBS1.302.685535775

File Name: +data/mystd/DATAFILE/SYSTEM.297.688213333

File Name: +data/mystd/DATAFILE/SYSAUX.267.688213333

File Name: +data/mystd/DATAFILE/UNDOTBS1.268.688213335

Do you really want to catalog the above files (enter YES or NO)? YES

cataloging files...

cataloging done

List of Cataloged Files

=======================

File Name: +data/mystd/DATAFILE/SYSTEM.297.688213333

File Name: +data/mystd/DATAFILE/SYSAUX.267.688213333

File Name: +data/mystd/DATAFILE/UNDOTBS1.268.688213335

12. Chuyển các datafile sang tên chính xác của nó tại standby :

--STANDBY

RMAN> **SWITCH DATABASE TO COPY;**

datafile 1 switched to datafile copy "+DATA/mystd/datafile/system.297.688213333"

datafile 2 switched to datafile copy "+DATA/mystd/datafile/undotbs1.268.688213335"

datafile 3 switched to datafile copy "+DATA/mystd/datafile/sysaux.267.688213333"

13. Trên cơ sở dữ liệu ở standby, xóa tất cả các standby redo log group:

--STANDBY

SQL> **ALTER DATABASE CLEAR LOGFILE GROUP [group number];**

**Xác minh:**  
Chạy lại các truy vấn từ bước 2 để xác nhận rằng incremental apply đã chuyển các datafile forwared. SCN bây giờ phải lớn hơn giá trị ban đầu của nó:

--STANDBY

SQL> **SELECT CURRENT\_SCN FROM V$DATABASE;**

CURRENT\_SCN

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

SQL> **select min(fhscn) from x$kcvfh;**

MIN(FHSCN)

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

SQL> **select min(f.fhscn) from x$kcvfh f, v$datafile d**

**where f.hxfil =d.file#**

**and d.enabled != 'READ ONLY' ;**

MIN(F.FHSCN)

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

14. Bắt đầu quá trình MRP ở **standby**

--**standby**

SQL> **ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT LOGFILE DISCONNECT FROM SESSION;**

Bạn cũng có thể lấy và thực hiện incremental backup các datafile từ SCN ở standby ​​trên cơ sở dữ liệu chính. Bằng cách đó, bạn có thể giảm thời gian sao lưu và khôi phục.

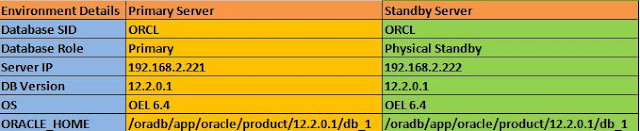
## 3. Thủ tục switchover Oracle DataGuard

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Công việc** | **Câu lệnh tại máy primay 192.168.0.22** | **Câu lệnh tại máy standby 192.168.0.11** |
| 1 | Kiểm tra đồng bộ log giữa 2 máy | ALTER SESSION SET nls\_date\_format='DD-MON-YYYY HH24:MI:SS';  SELECT sequence#, first\_time, next\_time  FROM   v$archived\_log  ORDER BY sequence#;  kết quả:  3289 29-MAY-2017 16:55:53 29-MAY-2017 17:20:49 | ALTER SESSION SET nls\_date\_format='DD-MON-YYYY HH24:MI:SS';  SELECT sequence#, first\_time, next\_time, applied  FROM   v$archived\_log  ORDER BY sequence#;  kết quả:  3289 29-MAY-2017 16:55:53 29-MAY-2017 17:20:49 YES |
| 2 | Nếu trên máy standby trạng thái NO  thì  chuyển về Yes |  | recover standby database; |
| 3 | Tắt database trên máy 11 |  | sqlplus / as sysdba  shutdown immediate |
| 4 | Bật lại database máy 11 |  | sqlplus / as sysdba  startup mount |
| 5 | apply log trên máy standby 11 |  | alter database recover managed standby database   disconnect from session; |
| 6 | Thực hiện switchover tư máy  22 sáng máy 11 thực hiên trên máy 22  trước sau đó thực thi trên máy 11 | sqlplus / as sysdba  alter database commit to switchover to standby;  -- Mount old primary database to standby database  alter database mount standby database;  alter database recover managed standby database   disconnect from session; | sqlplus  / as sysdba  alter database commit to switchover to primary;  --Shutdown standby database  shutdown immediate  -- Open old standby database to primary  startup; |
| 7 | Tắt Database máy 22 | sqlplus / as sysdba  shutdown immediate |  |
| 8 | Sau khi thêm tài nguyên thì bật lên | sqlplus / as sysdba  startup mount |  |
| 9 | apply log trên máy standby 22 | alter database recover managed standby database   disconnect from session; |  |
| 6 | Thực hiện switchover từ  máy 11 sáng máy 22 thực hiện trên  máy 22 trước rồi mới thực hiện máy 11 | sqlplus  / as sysdba  alter database commit to switchover to primary;  --Shutdown standby database  shutdown immediate  -- Open old standby database to primary  startup; | sqlplus / as sysdba  alter database commit to switchover to standby;  -- Mount old primary database to standby database  alter database mount standby database;  alter database recover managed standby database   disconnect from session; |

## 4. Thủ tục Failover Oracle Data Guard

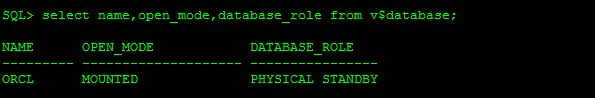
Khi site chính gặp sự cố đột  ngột không khôi phục được thì buộc phải bật con Physical Standby thành Primary, hay còn gọi là Failover

**Chi tiết cấu hình Dataguard**

**[](https://lh3.googleusercontent.com/-sl1bRRbYBf0/YH7_QG0vohI/AAAAAAAAA5Q/v-W5uwv94FkBimXQjge3pL3ZQjyZ0S8FwCLcBGAsYHQ/image.png)**

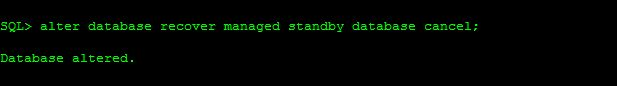
**Login in vào Standby Server:-**

**Step:1 Kiểm tra** database role và open\_mode  
select name,open\_mode,database\_role from v$database;

[](https://lh3.googleusercontent.com/-8QSmjGYRIB8/YH7_X22WxMI/AAAAAAAAA5U/dN_BnaIGZKspJ9SpBpOA1oysQcsQuhecwCLcBGAsYHQ/image.png)

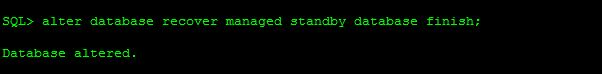
**Step:2** Cancel tiến trình MRP  
alter database recover managed standby database cancel;

[](https://lh3.googleusercontent.com/-VftiCxfUp1I/YH7_jJvNcWI/AAAAAAAAA5c/AmIulW21kZcwbISXqk--xwfziocj-4cBwCLcBGAsYHQ/image.png)

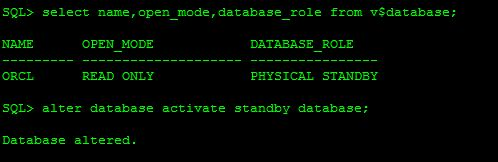
[](https://lh3.googleusercontent.com/-et7YxFrRFzM/YH7_kd8tRPI/AAAAAAAAA5g/uj68FA_hrpkxwOSH6mEPWuMb_Ie9zvooACLcBGAsYHQ/image.png)

**Step:3**Chuyển  standby thành primary

alter database recover managed standby database finish;

[](https://lh3.googleusercontent.com/-y56Phk6b0Us/YH7_rsnP2qI/AAAAAAAAA5o/22fRhlYKHOADkciP2x0hvXNsB8QFUgFQgCLcBGAsYHQ/image.png)

[alter database activate standby database;](https://oracledbwr.com/wp-content/uploads/2018/06/5-2.jpg)

[](https://lh3.googleusercontent.com/-jZ2B0dIwmCM/YH7_ynbPY_I/AAAAAAAAA5s/DubvWPxN50kIXNeQZ6x6sxNrcYQKKQP1wCLcBGAsYHQ/image.png)

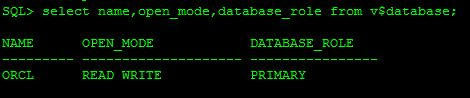
Managed recovery process đã tạm dừng giữa primary và standby database và standby trở thành primary database.

**Step:4**shutdown và kiểm tra database name, open mode, role.

shutdown immediate;

Startup

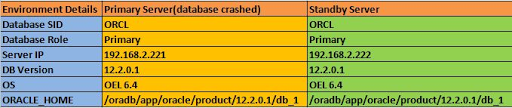
**Step:5**  
select name,open\_mode,database\_role from v$database;

[](https://lh3.googleusercontent.com/-oEhnt4UZMJY/YH8AH_JaikI/AAAAAAAAA58/xQHgFMSaXDUMfTqO9NXonui579RxG5W2QCLcBGAsYHQ/image.png)

Note:

Nên backup full 1 bản ở Primary chính để đảm bảo an toàn vì hiện tại không có dự phòng gì ngoài bản backup full vừa thực hiện (chú ý bản backup phải nằm trên SAN khác với SAN chứa datafile)

Sau khi failover:

[](https://lh3.googleusercontent.com/-YT9MCXTIA_o/YH8AU1tY1tI/AAAAAAAAA6A/tLKZZpmiOkswbjby3bGXN_nSc37vFDj7ACLcBGAsYHQ/image.png)

## Tài liệu tham khảo:

<https://www.tranvanbinh.vn/p/ong-bo-dataguard.html>

<https://www.tranvanbinh.vn/2021/08/cai-at-cau-hinh-quan-tri-oracle-data.html>

# ORACLE GOLDENGATE

## Cài đặt:

<https://www.tranvanbinh.vn/2022/09/vvip-thu-tuc-cai-at-oracle-golden-gate.html>

<https://www.tranvanbinh.vn/2023/05/oracle-goldengate-19c-cach-cau-hinh.html>

## 1. GIÁM SÁT ORACLE GOLDENGATE

**1.CHECK**

--Log: /u01/app/oracle/goldengate/ggserr.log

1. Check all processes

GGSCI (db1) 8> info all

GGSCI (db1) 8> status all

**2. View thông tin manager**

GGSCI (app-prepaid1) 4> info manager

**3. View thông tin extract**

- View thông tin đọc trail file:

GGSCI (db1) 40> info hr\_rep1

REPLICAT   HR\_REP1   Last Started 2013-01-08 08:08   Status RUNNING

Checkpoint Lag       07:43:40 (updated 00:00:09 ago)

Log Read Checkpoint  File /u05/oracle/goldengate/dirdat/hr\_rmt1/rt001601

                     2013-01-08 00:25:32.562485  RBA 19334029

- View quá trình xử lý trail file:

GGSCI (app-prepaid1) 3> info replicat rep1, detail

- info extract HR\_PUMP6, SHOWCH

- Lấy thông tin xử lý trail file: Processing data hay At EOF để xác định lỗi từ đâu

GGSCI () 2> send CUSPUMP3, status

Sending STATUS request to EXTRACT CUSPUMP3 ...

EXTRACT CUSPUMP3 (PID 10701)

  Current status: Recovery complete: Processing data

  Current read position:

  Sequence #: 13458

  RBA: 99269918

  Timestamp: 2012-12-03 08:23:44.000000

  Extract Trail: /dbfs\_direct/shared\_files/goldengate/dirdat/cusext3/lt

  Current write position:

  Sequence #: 13394

  RBA: 85181329

  Timestamp: 2012-12-03 08:44:27.105848

  Extract Trail: /u05/oracle/goldengate/dirdat/cusrmt3/rt

- Log Report:

GGSCI> view report EXTG\_MF  (file tương ứng trong dirrpt)

- View các tham số của group

GG> view params EXT1

- Thống kê giao dịch theo thời gian, ngày,…:

GGSCI (devu007) 33> stats extract ext1

## 2. Cấu hình thêm/bớt bảng trong Oracle GoldenGate

1. Thêm một bảng cho một group

Ví dụ thêm một bảng mới OWNER\_TEST.TABLE\_TEST cho group OTHEXT1, và đã có cấu trúc trên database

#1.Trên db1\_source

$GGATE/ggsci

DBLOGIN USERID ggate, PASSWORD xxxxx,ENCRYPTKEY DEFAULT

ADD TRANDATA OWNER\_TEST.TABLE\_TEST

#2. Trên db1\_source: STOP EXTRACT & PUMP

STOP OTHEXT1

STOP OTHPUMP1

#3. Trên db1\_source: Thêm tham số bên dưới vào parameter file của 2 group OTHEXT1 & OTHPUMP1 (cú pháp sửa parameter theo mục 5.2.2.3)

TABLE OWNER\_TEST.TABLE\_TEST;

#4. Nếu bảng không có dữ liệu thì sang bước 5. Sử dụng datapump Export dữ liệu trên bảng OWNER\_TEST.TABLE\_TEST trên db1\_source ra dump file, copy dumpfile sang db1\_source-HCM và import dữ liệu từ dump file

#5. Nếu bảng đã có dữ liệu thì bỏ qua bước này. Trên db1\_source-HCM Tạo cấu trúc dữ bảng giống hệt với bên db1\_source – sử dụng công cụ TOAD để lấy script

#6. Trên db1\_source-HCM : STOP REPLICAT

STOP  OTHREP1

#7. Trên db1\_source-HCM: Thêm tham số vào parameter file của group OTHREP1

#7.1.Nếu bảng chưa có dữ liệu dùng tham số

MAP OWNER\_TEST.TABLE\_TEST, TARGET OWNER\_TEST.TABLE\_TEST;

#7.2.Nếu bảng có dữ liệu dùng tham số

MAP OWNER\_TEST.TABLE\_TEST, TARGET OWNER\_TEST.TABLE\_TEST, HANDLECOLLISIONS;

#8. Trên db1\_source:  start các tiến trình OTHEXT1 & OTHPUMP1

START OTHEXT1

START OTHPUMP1

#9. Trên db1\_source-HCM:  START tiến trình OTHREP1

START OTHREP1

#.10. Nếu bảng không có dữ liệu bỏ qua bước này. Sau khi các tiến trình đồng bộ OTHEXT1 & OTHPUMP1 & OTHREP1 đã hết lag. Sửa lại tham số cho group OTHREP1

STOP OTHREP1

#Sửa tham số: MAP OWNER\_TEST.TABLE\_TEST, TARGET OWNER\_TEST.TABLE\_TEST, HANDLECOLLISIONS;

Thành MAP OWNER\_TEST.TABLE\_TEST, TARGET OWNER\_TEST.TABLE\_TEST;

START OTHREP;

2. Loại bỏ một bảng cho một group

Ví dụ loại bỏ một bảng OWNER\_TEST.TABLE\_TEST cho group OTHEXT

#1. Trên db1\_source

$GGATE/ggsci

GGATE> STOP OTHEXT1

#2. Thêm tham số tableexclude vào parameter file của group OTHEXT1 (cú pháp sửa parameter theo mục 5.2.2.3)

TABLEEXCLUDE OWNER\_TEST.TABLE\_TEST;

#3. Trên db1\_source:  start tiến trình OTHEXT1

START OTHEXT1

3. Skip một tiến trình đang mining

#Trên db1\_source

$GGATE/ggsci

GGATE> SEND extract\_name SHOWTRANS

GGATE> SEND extract\_name SKIPTRANS trans\_id

4. Thay đổi tham số replicate

GGSCI> edit param REP1

5. Login

dblogin userid gg\_owner

## 3. Xử lý lỗi WARNING OGG-01519 GoldenGate bị abend trên tiến trình Extract, Pump, Replicat

**1. Lỗi: WARNING OGG-01519 GoldenGate**

2021-02-19 23:55:48  WARNING OGG-01519  Oracle GoldenGate Capture for Oracle, pump1.prm:  Waiting at EOF on input trail file /u01/oracle/goldengate/dirdat/ext1/lt248294, which is not marked as complete; but succeeding trail file /u01/oracle/goldengate/dirdat/ext1/lt248295 exists. If ALTER ETROLLOVER has been performed on source extract, ALTER EXTSEQNO must be performed on each corresponding downstream reader.

2021-02-20 00:03:44  INFO    OGG-00957  Oracle GoldenGate Manager for Oracle, mgr.prm:  Purged old extract file/u01/oracle/goldengate/dirdat/cusext2/lt408637, applying UseCheckPoints purge rule: Oldest Chkpt Seqno 408808 > 408637.

**2. Nguyên nhân**

- File  /u01/oracle/goldengate/dirdat/ext1/lt248294 không có tín hiệu EOF (End of file) mở mãi không đóng mặc dù kết thúc rồi.

**3. Giải pháp**

**-- SOURCE**

***-- Extract  ext1***

GGSCI> stop ext1

(nếu không được thì GGSCI> kill extract ext1)

GGSCI> info ext1, detail

GGSCI> alter extract ext1 etrollover

--GGSCI> alter ext1 begin 2021-02-19 23:55:00

--GGSCI> alter ext1 begin now

GGSCI> alter ext1 extseqno 248295 extrba 0

GGSCI> Start ext1

GGSCI> Info ext1

--+ file moi tao la 248295

[oracle@dbaviet03 ext1]$ ls -lt

total 46243

-rw-rw-rw- 1 oracle oinstall   29631 Feb 19 23:46 lt248296

-rw-rw-rw- 1 oracle oinstall    1066 Feb 19 23:34 lt248295

-rw-rw-rw- 1 oracle oinstall 7321619 Feb 19 23:23 lt248294

-rw-rw-rw- 1 oracle oinstall 9999895 Feb 19 19:53 lt248293

-rw-rw-rw- 1 oracle oinstall 9999732 Feb 19 19:53 lt248292

-rw-rw-rw- 1 oracle oinstall 9999787 Feb 19 19:53 lt248291

-rw-rw-rw- 1 oracle oinstall 9998739 Feb 19 19:53 lt248290

***-- Pump***pump1

GGSCI> stop pump1

(nếu không được thì GGSCI> kill extract pump1)

GGSCI> alter pump1, extseqno 248295 extrba0

GGSCI> start pump1

***--DEST (Replicat): Nếu cần thì thực hiện***

GGSCI> stop rep1

(nếu không được thì GGSCI> kill extract rep1)

GGSCI> alter rep1, extseqno 088955 extrba0

GGSCI> start rep1

--+ File trail bên dest:

[oracle@dbavietDEST dirdat]$ ls -lt

total 14336

-rw-r-----. 1 oracle oinstall   31607 Feb 19 23:52 t1088957

-rw-r-----. 1 oracle oinstall    1531 Feb 19 23:39 t1088956

-rw-r-----. 1 oracle oinstall    1473 Feb 19 23:37 t1088955

-rw-r--r--. 1 oracle oinstall 7318678 Feb 19 23:21 t1088954

-rw-r--r--. 1 oracle oinstall 7318678 Feb 19 22:28 lt248294

## Tài liệu tham khảo:

<https://www.tranvanbinh.vn/p/ong-bo-goldengate.html>

# HƯỚNG DẪN GIÁM SÁT, VẬN HÀNH SQL SERVER

Các CSDL SQL Server tải thấp, do vậy chỉ cần đảm bảo an toàn về mặt dữ liệu, backup đầy đủ, dọn dẹp định kỳ; chu kỳ backup:

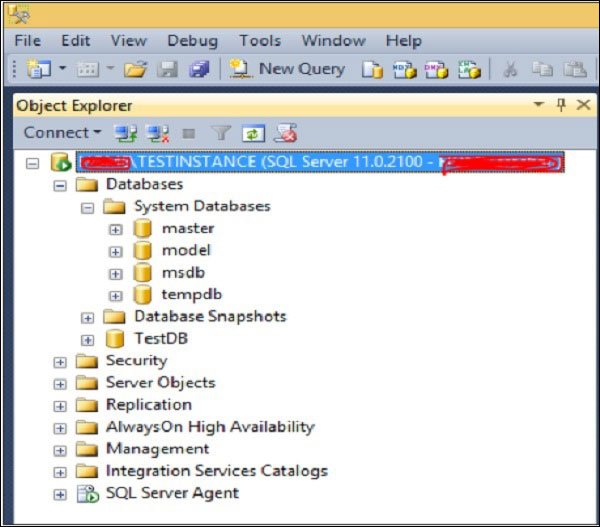
* Backup full: ngày 2 lần vào lúc 0h và 12h
* Backup transaction log: 20 phút backup 1 lần

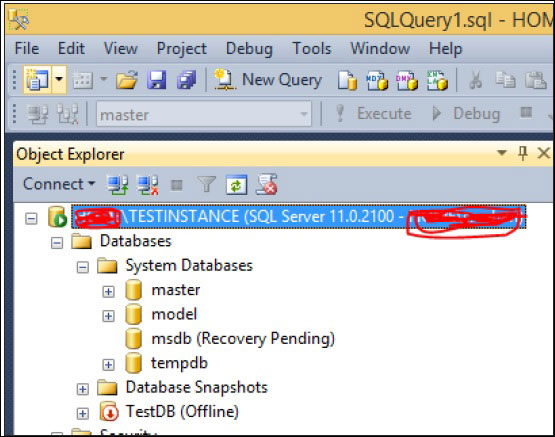
## 1. Giám sát cơ sở dữ liệu

Giám sát dùng để chỉ việc kiểm tra trạng thái cơ sở dữ liệu (CSDL), thiết lập tên người sở hữu, tên tập tin, kích thước tập tin, lịch trình sao lưu…

Việc giám sát CSDL trên SQL Server có thể thực hiện chủ yếu qua SQL Server Management Studio hoặc T-SQL, và cũng có thể kiểm soát qua nhiều cách khác như các công cụ của bên thứ 3, tạo Job trong Server Agent…

Có thể kiểm tra trạng thái của CSDL xem là đang kết nối (Online) như trong hình dưới đây hoặc đang ở trạng thái khác (Offiline) như trong hình bên dưới.

  
CSDL đang trong trạng thái kết nối

  
TestDB hiện đang Offline

## 2. Vận hành CSDL

### a. Cách khởi động và dừng các dịch vụ trong MS SQL Server

MS SQL Server mang tới 2 dịch vụ chính để tạo và duy trì cơ sở dữ liệu (CSDL). Ngoài ra còn có các dịch vụ bổ sung phục vụ các mục đích khác.

Hai dịch vụ chính của MS SQL Server gồm:

* SQL Server
* SQL Server Agent

Các dịch vụ bổ sung của MS SQL Server gồm:

* SQL Server Browser
* SQL Server FulL Text Search
* SQL Server Integration Services
* SQL Server Reporting Services
* SQL Server Analysis Services

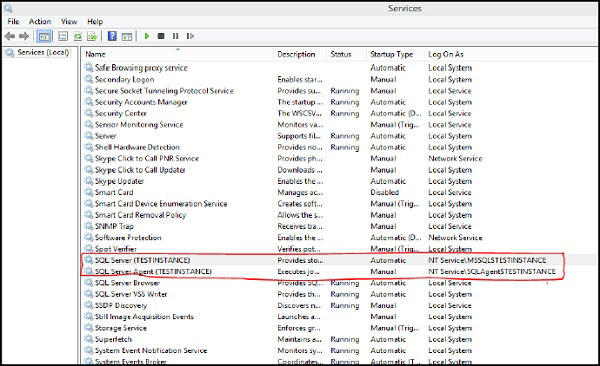
Bạn có thể bắt đầu hoặc tạm ngưng sử dụng các dịch vụ nói trên bằng nhưng cách dưới đây.

Khởi động dịch vụ trong MS SQL Server

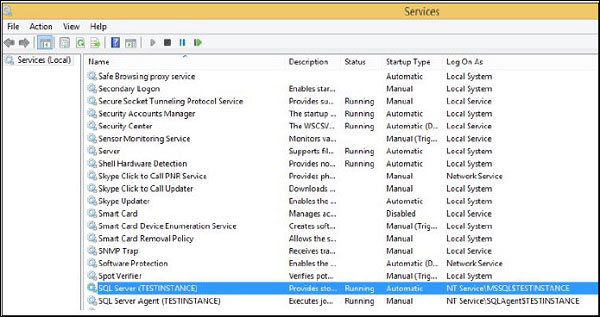
Để khởi động bất kì dịch vụ nào trong MS SQL Server, bạn có thể dùng 1 trong 2 cách dưới đây.

**Cách 1: Sử dụng Services.msc**

**Bước 1**: Đi tới hộp thoại **Run,** gõ **services.msc** và click **OK.** Màn hình dưới đây sẽ hiện ra.

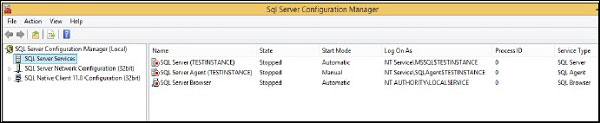
  
Tìm dịch vụ trong hộp thoại Run của Windows

**Bước 2**: Để khởi động, click chuột phải vào dịch vụ bạn cần và chọn **Start.** Khi đó dịch vụ sẽ khởi động như trong hình dưới đây.

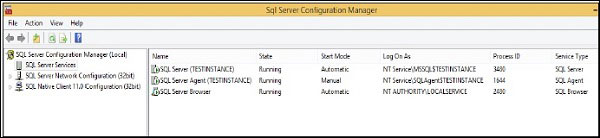
  
Đã khởi động dịch vụ được chọn

**Cách 2: Dùng SQL Server Configuration Manager**

**Bước 1:** Mở trình quản lý bằng các bước **Start** > **All Programs** > **MS SQL Server 2012** > **Configuration Tools** > **SQL Server Configuration Manager.**

  
Danh sách dịch vụ hiện ra khi đã mở được trình quản lý

**Bước 2:** Chọn tên dịch vụ, click chuột phải và chọn **Start.** Dịch vụ khởi động thành công như hình bên dưới.

  
Khởi động thành công dịch vụ trong MS SQL Server

### b. Dừng dịch vụ trong MS SQL Server

Để dừng dịch vụ trong MS SQL Server, sử dụng 1 trong 3 cách sau.

**Cách 1: Dùng Services.msc**

**Bước 1**: Đi tới hộp thoại **Run,** gõ **services.msc** và click **OK.** như trong Bước 1 - cách 1 khi khởi động dịch vụ.

**Bước 2:** Để dừng dịch vụ, click chuột phải và chọn **Stop.** Dịch vụ vừa chọn sẽ được dừng lại như trong hình bên dưới.

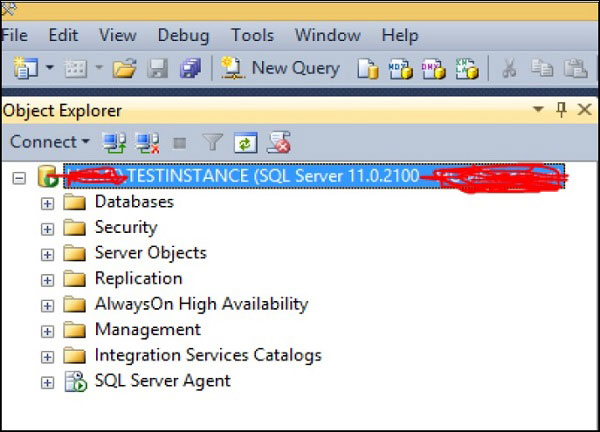
**Cách 2: Dùng SQL Server Configuration Manager**

**Bước 1:** Mở **Configuration Manager** bằng các bước như trong bước 2 - cách 2 khi khởi động dịch vụ.

**Bước 2:** Chọn dịch vụ cần dừng, click chuột phải và chọn **Stop.** Dịch vụ được chọn sẽ dừng lại.

**Cách 3: Dùng SQL Server Management Studio SSMS**

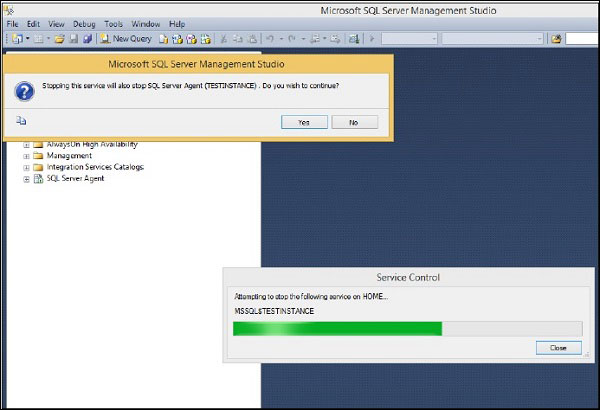
**Bước 1:** Kết nối tới bản cài của SQL Server như trong hình dưới.

  
Bản TESTINSTANCE của SQL Server

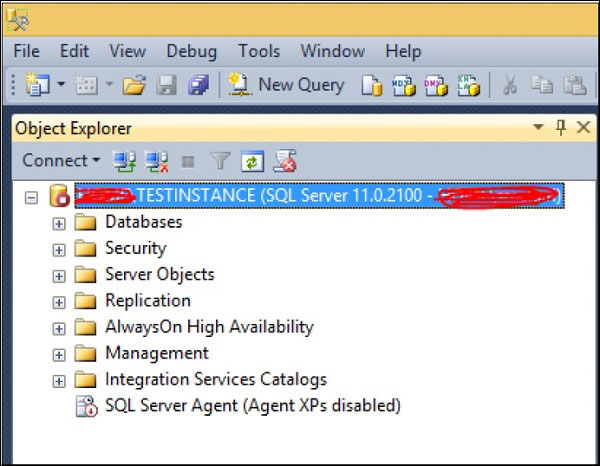
**Bước 2:** Click chuột phải vào tên của bản cài và chọn **Stop.** Màn hình dưới đây sẽ hiện ra.

  
Hộp thoại khi chọn Stop bản cài SQL Server

**Bước 3:** Chọn **Yes,** màn hình dưới đây sẽ hiện ra.

  
Xác nhận lại việc dừng dịch vụ

**Bước 4:** Chọn **Yes** để đồng ý dừng dịch vụ SQL Server Agent. Dịch vụ này sẽ tạm ngừng như trong hình bên dưới.

  
Trạng thái SQL Server Agent là Disabled

### c. Backup CSDL

Backup là một bản sao lưu dữ liệu/cơ sở dữ liệu. Việc sao lưu CSDL trong MS SQL Server là rất quan trọng để bảo vệ dữ liệu trước việc mất CSDL. Có 3 hình thức sao lưu chính trong MS SQL Server là **Full/Database, Differential/Incremental, Transactional Log/Log.**

Dưới đây là 2 cách để tạo bản sao lưu CSDL trong MS SQL Server Management Studio.

**CÁCH 1: SAO LƯU DATABASE BẰNG T-SQL**

**Full/Database**

Backup database <TEN DATABASE> to disk = '<DUONG DAN FILE BACKUP + TEN FILE>'

**Differential/Incremental**

Backup database <TEN DATABASE> to

disk = '<DUONG DAN FILE BACK UP + TEN FILE>' with differential

**Transactional Log/Log**

Backup log <TEN DATABASE> to disk = '<DUONG DAN FILE BACKUP + TEN FILE>'

Ví dụ: Lệnh dưới đây được dùng để sao lưu toàn bộ CSDL (Full/Database) có tên QTM tới địa chỉ **D:\** với tên của bản sao lưu là **QTM.bak**

Backup database QTM to disk = 'D:\QTM.bak'

Lệnh dưới đây được dùng để sao lưu những thay đổi kể từ lần sao lưu gần nhất trên CSDL (Differential/Incremental) có tên QTM tới địa chỉ **D:\** với tên của bản sao lưu là: **TestDB\_Full.bak**

Backup database QTM to disk = 'D:\QTM.bak' with differential

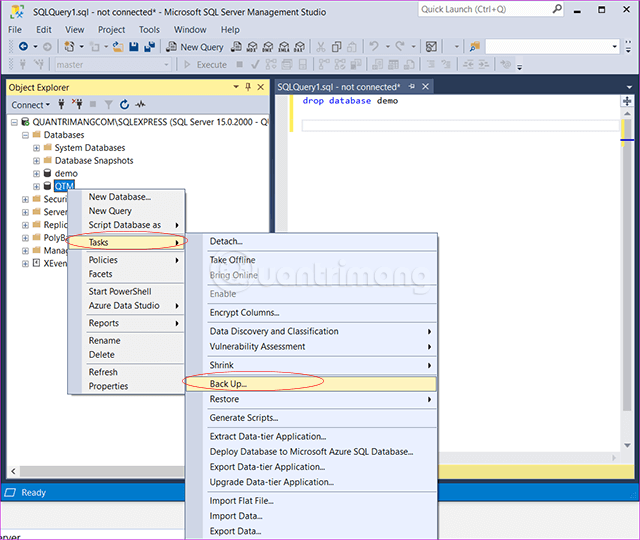
Lệnh dưới đây được dùng để sao lưu một cách thứ tự tất cả giao dịch kể từ lần thực hiện sao lưu Transaction Log gần nhất trên CSDL (Transaction Log/Log) có tên QTM tới địa chỉ **D:\** với tên của bản sao lưu là **QTM.bak**

Backup log QTM to disk = 'D:\QTM\_log.trn'

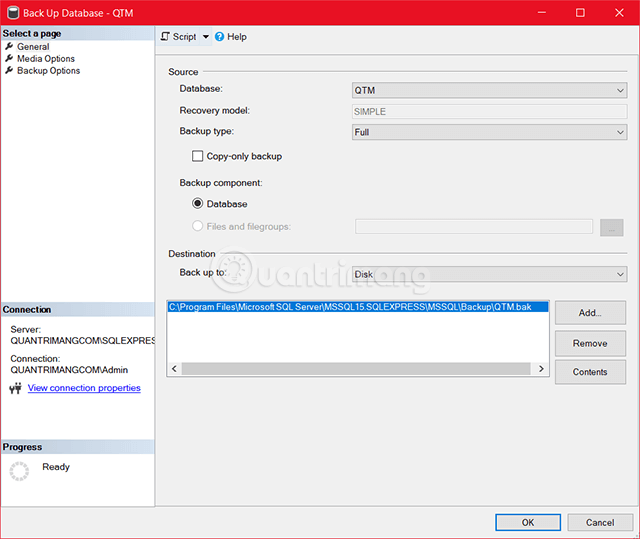
**CÁCH 2: DÙNG SQL SERVER MANAGEMENT STUDIO (SSMS)**

Giao diện MS SQL Server Management Studio có thể khác nhau một chút trên các phiên bản, hướng dẫn dưới đây được thực hiện trên phiên bản 18, ở các bản cũ hơn bạn chỉ cần chuột phải vào CSDL cần sao lưu chọn Backup là được.

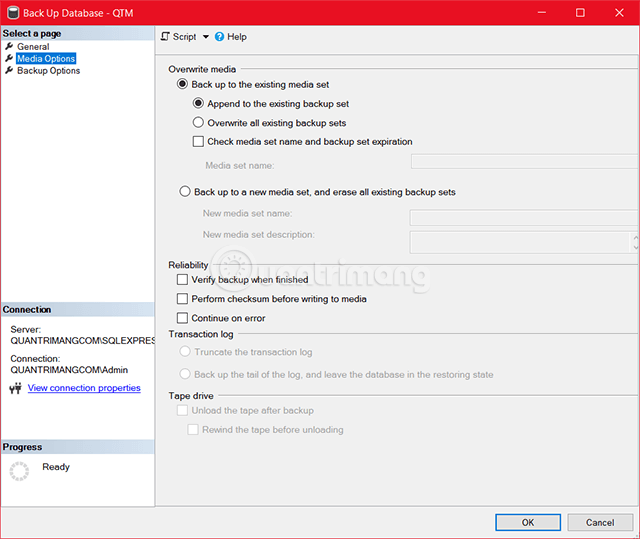
**Bước 1**: Kết nối tới bản cài CSDL và mở thư mục CSDL như trong hình dưới đây. Click chuột phải vào CSDL QTM và chọn **Tasks** > **Back Up...**:

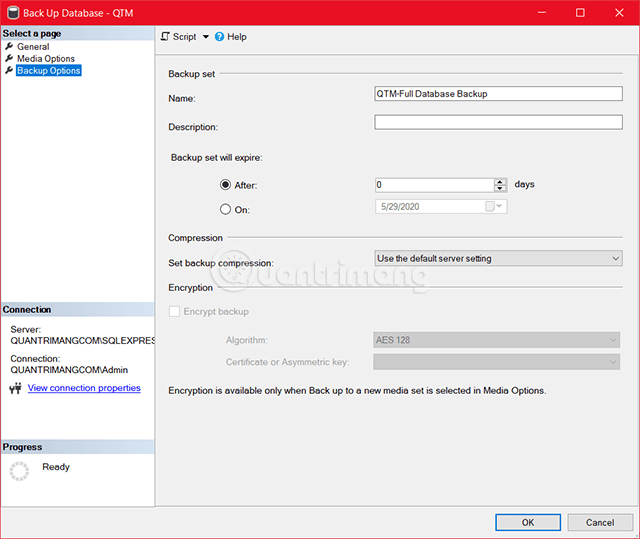
  
Mở thư mục CSDL

**Bước 2:** Khi đó màn hình dưới đây sẽ xuất hiện.

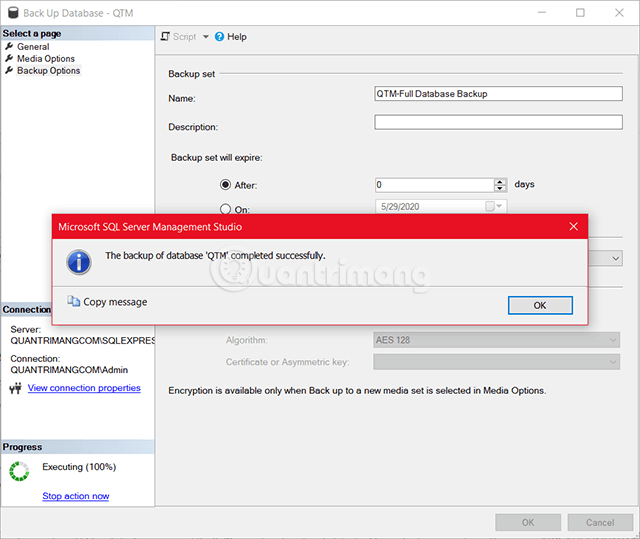
  
Các thông tin lựa chọn hình thức sao lưu cho CSDL

**Bước 3:** Lựa chọn loại sao lưu CSDL và chọn đúng đường dẫn cho nơi lưu bản sao. Chọn **Options** ở góc trên bên trái màn hình, khi đó hộp thoại sau sẽ hiện ra.

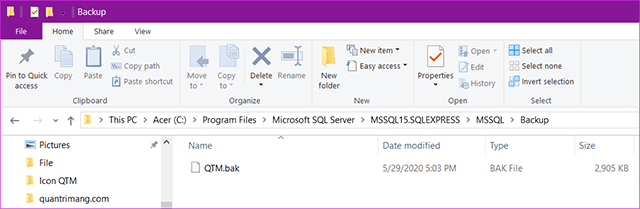
  
Lựa chọn sao lưu CSDL khác



**Bước 4:** Chọn **OK** để tạo bản sao lưu cho QTM như trong hình dưới đây.

  
Tạo bản sao lưu thành công

Kiểm tra trong đường dẫn thư mục vừa chọn, bạn sẽ thấy file sao lưu.

  
File sao lưu của CSDL QTM vừa được tạo

### b. Restore CSDL

Khôi phục là quá trình sao chép dữ liệu đã sao lưu và đưa các giao dịch được ghi lại vào dữ liệu của MS SQL Server. Hiểu đơn giản, đây là quá trình lấy file sao lưu và đưa nó trở lại CSDL.

Khôi phục CSDL có thể được thực hiện bằng 2 cách.

**Cách 1: Sử dụng T-SQL**

Cú pháp dưới đây được dùng để khôi phục CSDL.

Restore database <Your database name> from disk = '<Backup file location + file name>'

Ví dụ

Lệnh sau đây được dùng để khôi phục lại CSDL có tên TestDB đã có tập tin sao lưu tên là **TestDB\_Full.bak** nằm tại địa chỉ **D:\** nếu bạn chọn viết đè lên CSDL hiện tại.

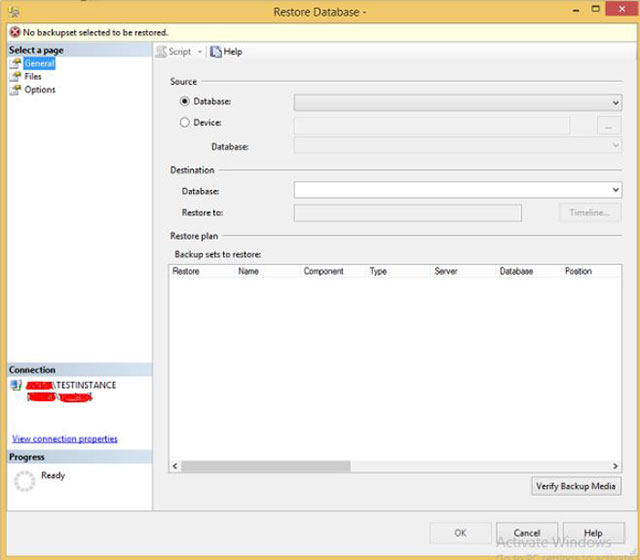
Restore database TestDB from disk = ' D:\TestDB\_Full.bak' with replace

Nếu bạn tạo CSDL mới bằng lệnh khôi phục này và không có đường dẫn, file ghi lại lịch sử trên máy chủ nói trên, hãy dùng lệnh dưới đây. Đảm bảo là có tồn tại đường dẫn **D:\Data.**

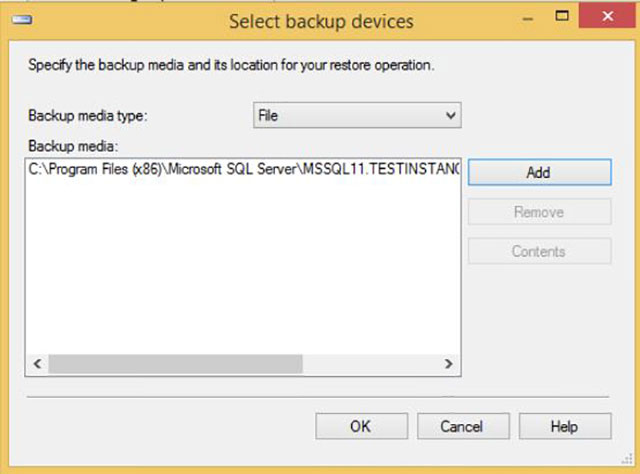
RESTORE DATABASE TestDB FROM DISK = 'D:\ TestDB\_Full.bak' WITH MOVE 'TestDB' TO 'D:\Data\TestDB.mdf', MOVE 'TestDB\_Log' TO 'D:\Data\TestDB\_Log.ldf'

**Cách 2: Sử dụng SQL Server Management Studio SSMS**

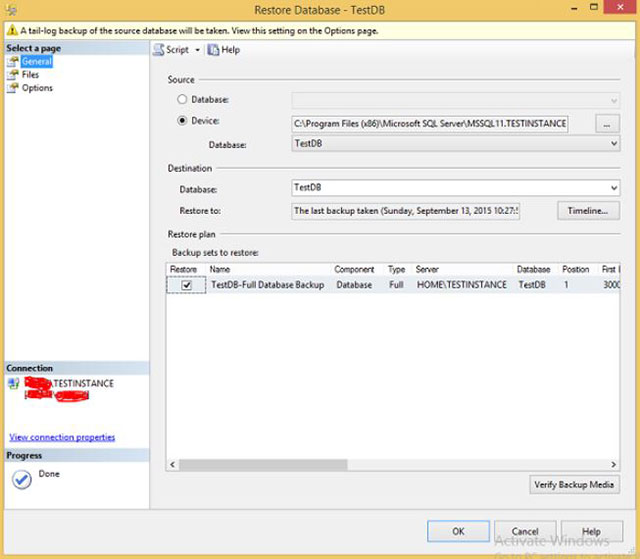
**Bước 1**: Kết nối tới CSDL có tên **TESTINSTANCE** và click chuột phải vào thư mục, chọn **Restore Database** để hiện ra như hình dưới đây.

  
Khôi phục CSDL bằng lựa chọn sau khi click chuột phải

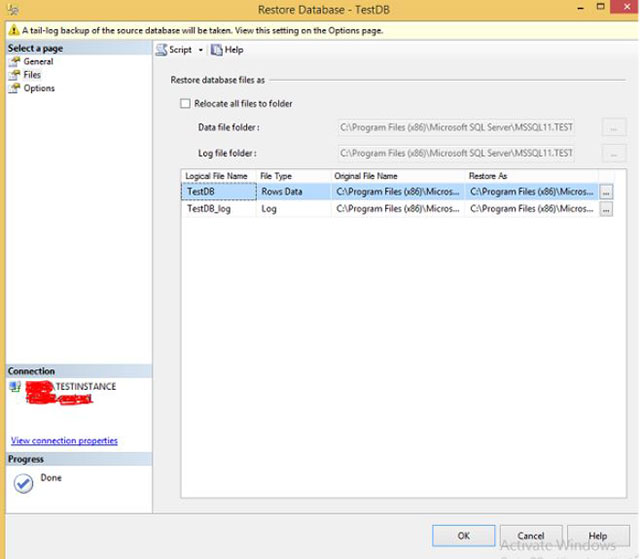
**Bước 2**: Chọn **Device** và chọn đường dẫn để mở tập tin sao lưu như trong hình dưới đây.

  
Tìm nơi đã lưu file sao lưu khi tạo trước đây

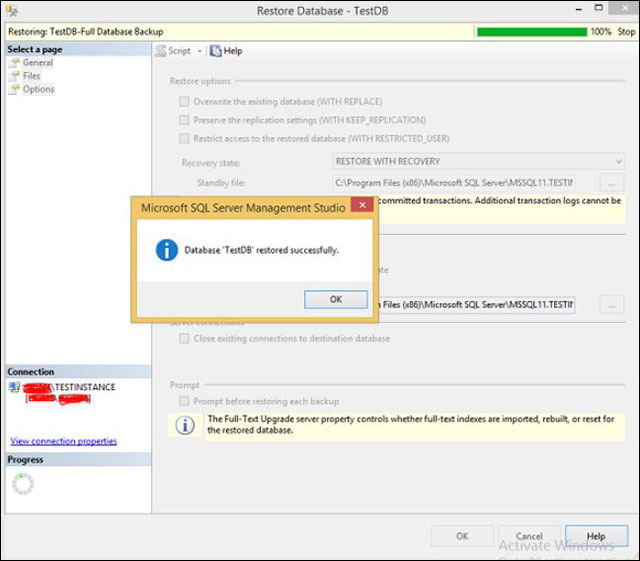
**Bước 3**: Click **OK** và màn hình dưới đây sẽ hiện ra.

  
Chọn CSDL cần khôi phục và tập tin sao lưu đã tạo trước đó

**Bước 4**: Chọn **Files** ở góc bên trái màn hình, hộp thoại dưới đây sẽ hiện ra.

  
Thông tin các file cho quá trình khôi phục CSDL

**Bước 5**: Chọn **Options** ở góc trái và click **OK** để bắt đầu khôi phục CSDL TestDB như trong hình dưới đây.

  
Khôi phục thành công CSDL TestDB

## Tài liệu tham khảo:

<https://www.tranvanbinh.vn/p/sql-server.html>

# HƯỚNG DẪN GIÁM SÁT, VẬN HÀNH POSTGRESQL

## 1. Giám sát CSDL

Khi DB mức độ online 24/7 cần thực hiện các công việc kiểm tra đầu ngày và cuối ngày đầu ngày (8h) và cuối ngày (16h30) làm việc

### a. Kiểm tra không gian trống

|  |
| --- |
| Bước 1: Đăng nhập OS bằng tài khoản **postgres**  su - postgres |
| Bước 2: Kiểm tra dung lượng trống của các ổ đĩa bằng cách thực hiện câu lệnh  df -g hoặc df -h |
| Bước 3: : Xem cột %Used cho biết % không gian đã dùng của ổ đĩa. Nếu %Used ở mức cảnh báo Warning (85%) hoặc mức Critical (95%) cần thông báo cho nhân viên quản trị hệ thống để xử lý. |
| Bước 4: Ghi nhận kết quả kiểm tra vào *PL1\_Checklist vận hành PostgreSQL* |

### Kiểm tra alert log file

|  |
| --- |
| Hành động |
| Bước 1: Đăng nhập OS bằng tài khoản **postgres**  su - postgres |
| Bước 2: Đi đến thư mục /u03/pgAAA/log  cd /u03/pgAAA/log |
| Bước 3: Mở file và tìm “posgresql\_“ hoặc “error” từ dưới lên của file alert log trong một ngày làm việc tính từ thời điểm kiểm tra về trước. Nếu có dữ liệu logs error, cần thông báo cho nhân viên quản trị CSDL (theo Danh sách người quản trị CSDL ở Mục 5) để xử lý. |
| Bước 4: Ghi nhận kết quả kiểm tra vào *PL1\_Checklist vận hành PostgreSQL* |

### c. Kiểm tra trạng thái hoạt động của dịch vụ PostgreSQL

|  |
| --- |
| Bước 1: Đăng nhập OS bằng tài khoản **postgres**  su - postgres |
| Bước 2: Kiểm tra trạng thái hoạt động của dịch vụ PostgreSQL  pg\_ctl -D $PGDATA status  *- Trạng thái bình thường:*    *- Trạng thái không bình thường:*    Khi xuất hiện trạng thái không bình thường cần báo ngay cho nhân viên quản trị CSDL (theo Danh sách người quản trị CSDL ở Mục 5) để xử lý.  *- Các thao tác xử lý khi xuất hiện trạng thái không bình thường:*  Kiểm tra dữ liệu logs của dịch vụ PostgreSQL:  tail –n 100 –f $PGLOG/postgresql\_\*.log  tail –n 100 –f $PGLOG/logfile\_start  Phân tích dữ liệu logs của dịch vụ PostgreSQL, xác định nguyên nhân, tình trạng lỗi và xử lý (tham khảo *PL2\_Hướng dẫn phân tích dữ liệu logs PostgreSQL và xử lý lỗi)*  Thực hiện lại các Bước 1, Bước 2 để Kiểm tra trạng thái hoạt động của dịch vụ PostgreSQL |

### Kiểm tra trạng thái đồng bộ dữ liệu sang máy chủ dự phòng PostgreSQL

|  |
| --- |
| Hành động |
| Bước 1: Đăng nhập OS bằng tài khoản **postgres**  su - postgres |
| Bước 2: Kiểm tra trạng thái đồng bộ dữ liệu sang máy chủ dự phòng PostgreSQL  psql -x -c "select \* from pg\_stat\_replication;"  *- Trạng thái bình thường:*    *- Trạng thái không bình thường:*    Khi xuất hiện trạng thái không bình thường cần báo ngay cho nhân viên quản trị CSDL (theo Danh sách người quản trị CSDL ở Mục 5) để xử lý.  *- Các thao tác xử lý khi xuất hiện trạng thái không bình thường:*  Kiểm tra dữ liệu logs của dịch vụ PostgreSQL:  tail –n 100 –f $PGLOG/postgresql\_\*.log  tail –n 100 –f $PGLOG/logfile\_start  Kiểm tra trạng thái hoạt động của các máy chủ Slave theo mục **7.1.3**  Phân tích dữ liệu logs và trạng thái hoạt động của các máy chủ Slave để xác định nguyên nhân, tình trạng lỗi và xử lý (tham khảo *PL2\_Hướng dẫn phân tích dữ liệu logs PostgreSQL và xử lý lỗi)*  Thực hiện lại các Bước 1, Bước 2 để Kiểm tra trạng thái đồng bộ dữ liệu của dịch vụ PostgreSQL |

### Hướng dẫn phân tích dữ liệu logs PostgreSQL và xử lý lỗi

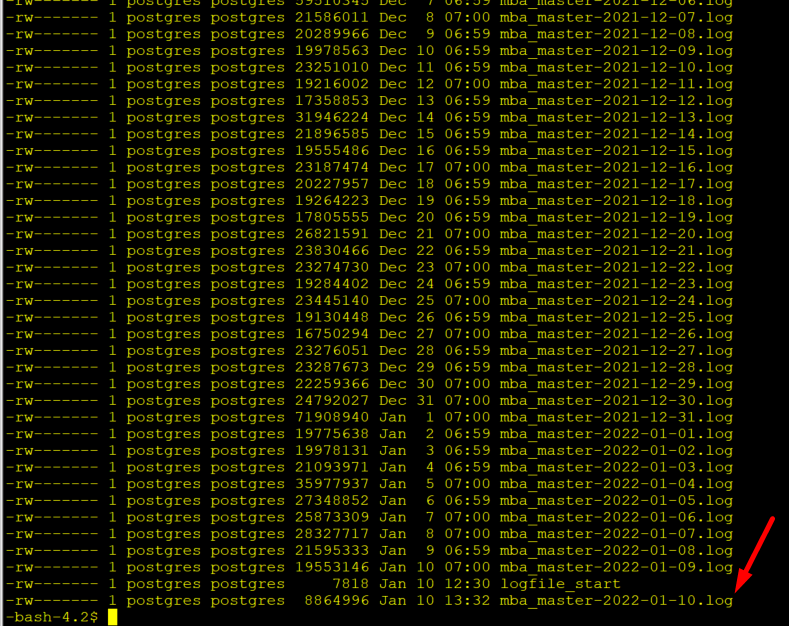
* **Cách đọc log trong postgres**

B1: Login vào server với user postgres

su – postgres

cd $PGLOG

ls -ltr



tail -f mba\_master-year-month-day.log

**Xác định sự cố, lỗi trong thời gian nào đó**

Copy log về máy windown

## 2. Vận hành CSDL

### a. Khỏi động dịch vụ PostgreSQL

|  |
| --- |
| Bước 1: Đăng nhập vào OS bằng tài khoản **postgres**  su - postgres |
| Bước 2: Khởi động dịch vụ PostgreSQL  pg\_ctl -D $PGDATA -l $PGLOG/logfile\_start start |
| Bước 3: Kiểm tra trạng thái hoạt động của dịch vụ PostgreSQL sau khi khởi động  pg\_ctl -D $PGDATA status  *- Trạng thái bình thường:*    *- Trạng thái không bình thường:*    Khi xuất hiện trạng thái không bình thường cần báo ngay cho nhân viên quản trị (theo Danh sách người quản trị ở Mục 5) để xử lý.  *- Các thao tác xử lý khi xuất hiện trạng thái không bình thường:*  Kiểm tra dữ liệu logs của dịch vụ PostgreSQL:  tail –n 100 –f $PGLOG/postgresql\_\*.log  tail –n 100 –f $PGLOG/logfile\_start  Phân tích dữ liệu logs của dịch vụ PostgreSQL và xác định nguyên nhân, trạng trái lỗi theo *PL2\_Hướng dẫn phân tích dữ liệu logs PostgreSQL và xử lý lỗi*  Sửa lỗi dựa trên quá trình phân dữ liệu logs và xác định nguyên nhân (Tham khảo *PL2\_Hướng dẫn phân tích dữ liệu logs PostgreSQL và xử lý lỗi*)  Thực hiện lại các Bước 2, Bước 3 để khởi động lại dịch vụ PostgreSQL và kiểm tra lại trạng thái. |

### b. Tắt dịch vụ PostgreSQL

|  |
| --- |
| Bước 1: Đăng nhập vào OS bằng tài khoản **postgres**  su - postgres |
| Bước 2: Tắt dịch vụ PostgreSQL  pg\_ctl -D $PGDATA stop -mf |
| Bước 3: Kiểm tra trạng thái hoạt động của dịch vụ PostgreSQL sau khi tắt  pg\_ctl -D $PGDATA status  *- Trạng thái bình thường:*    *- Trạng thái không bình thường:*    Khi xuất hiện trạng thái không bình thường cần  báo ngay cho nhân viên quản trị (theo Danh sách người quản trị ở Mục 5)  *- Các thao tác xử lý khi xuất hiện trạng thái không bình thường:*  Kiểm tra dữ liệu logs của dịch vụ PostgreSQL:  tail –n 100 –f $PGLOG/postgresql\_\*.log  tail –n 100 –f $PGLOG/logfile\_start  Phân tích dữ liệu logs của dịch vụ PostgreSQL và xác định nguyên nhân, tình trạng lỗi theo *PL2\_Hướng dẫn phân tích dữ liệu logs PostgreSQL và xử lý lỗi*  Sửa lỗi dựa trên quá trình phân dữ liệu logs và xác định nguyên nhân (Tham khảo *PL2\_Hướng dẫn phân tích dữ liệu logs PostgreSQL và xử lý lỗi*)  Thực hiện lại các Bước 2, Bước 3 để tắt dịch vụ PostgreSQL và kiểm tra lại trạng thái. |

### c. Khỏi động lại dịch vụ PostgreSQL

|  |
| --- |
| Bước 1: Đăng nhập vào OS bằng tài khoản **postgres**  su – postgres |
| Bước 2: Khởi động lại dịch vụ PostgreSQL  pg\_ctl -D $PGDATA restart |
| Bước 3: Kiểm tra trạng thái hoạt động của dịch vụ PostgreSQL sau khi khởi động lại  pg\_ctl -D $PGDATA status  *- Trạng thái bình thường:*    *- Trạng thái không bình thường:*    Khi xuất hiện trạng thái không bình thường cần báo ngay cho nhân viên quản trị (theo Danh sách người quản trị ở Mục 5)  *- Các thao tác xử lý khi xuất hiện trạng thái không bình thường:*  Kiểm tra dữ liệu logs của dịch vụ PostgreSQL:  tail –n 100 –f $PGLOG/postgresql\_\*.log  tail –n 100 –f $PGLOG/logfile\_start  Phân tích dữ liệu logs của dịch vụ PostgreSQL, xác định nguyên nhân, tình trạng lỗi và xử lý (Tham khảo *PL2\_Hướng dẫn phân tích dữ liệu logs PostgreSQL và xử lý lỗi)*  Thực hiện lại các Bước 2, Bước 3 để khởi động lại dịch vụ PostgreSQL và kiểm tra lại trạng thái. |

### d. Sao lưu dữ liệu

|  |
| --- |
| Bước 1: Đăng nhập OS bằng tài khoản **postgres**  su - postgres |
| Bước 2: Thực hiện sao lưu database  pg\_dump -Fc pgsqldbname > $PG\_BACKUP\_DATA\_DIR/full\_pgsqldbname\_YYYYMMDD\_HH\_mm.tar  *Sao lưu thành công:*    *Sao lưu không thành công:*    Xuất hiện lỗi error bên dưới thông báo sao lưu. Khi đó cần báo ngay cho nhân viên quản trị (theo Danh sách người quản trị ở Mục 5) để xử lý. |

### Khôi phục dữ liệu

|  |
| --- |
| Bước 1: Đăng nhập OS bằng tài khoản **postgres**  su - postgres |
| Bước 2: Thực hiện khôi phục database  pg\_restore -C -d pgsqldbname > $PG\_BACKUP\_DATA\_DIR/full\_pgsqldbname\_YYYYMMDD\_HH\_mm.tar  *Phục hồi thành công:*    *Phục hồi không thành công:*    Xuất hiện lỗi error bên dưới thông báo phục hồi. Khi đó cần báo ngay cho nhân viên quản trị (theo Danh sách người quản trị ở Mục 5) để xử lý. |

## 3. Các câu lệnh khác

**- Stop/Start:**

service postgresql-12 status

service postgresql-12 start

**- Connect:**

psql -h localhost -p 5432

psql -h localhost -p 9999

psql -h localhost -p 9999 -c "show pool\_nodes;"

**- Backup/Restore:**

pg\_basebackup -h 192.168.254.31 -p 5432 -U replication -D /data01/pgdata/ -Fp -Xs -P -R -T '/data01/pgdata/sea\_tblspc\_api=/data01/pgdata/sea\_tblspc\_api\_stb' -T '/data01/pgdata/sea\_tblspc\_swib=/data01/pgdata/sea\_tblspc\_swib\_stb'

Tạo job crontab trên OS định kỳ 00h00 backup hàng ngày.

**- Clean archive log:**

pg\_archivecleanup -d /data01/pgdata/pg\_wal 00000002000000110000009C

/usr/pgsql-12/bin/pg\_archivecleanup -d /pgdata01/wal 000000020000001F000000B1

**- List user:**

postgres=# \du

SELECT usename AS role\_name, CASE WHEN usesuper AND usecreatedb THEN CAST('superuser, create database' AS pg\_catalog.text) WHEN usesuper THEN CAST('superuser' AS pg\_catalog.text) WHEN usecreatedb THEN CAST('create database' AS pg\_catalog.text) ELSE CAST('' AS pg\_catalog.text) END role\_attributes FROM pg\_catalog.pg\_user ORDER BY role\_name desc;

**- Create user:**

CREATE USER binhtv4 WITH PASSWORD 'Cognac#12345';

CREATE USER pcidss\_user WITH PASSWORD 'ank#123' VALID UNTIL '2050-01-01';

GRANT CONNECT ON DATABASE API TO pcidss\_user;

**- List db:**

postgres=# \l

**- Connect db:**

postgres=# \c API

**- List table:**

postgres=# \dt

**- Create role:**

CREATE ROLE grp\_ewallet WITH

NOLOGIN

NOSUPERUSER

INHERIT

NOCREATEDB

NOCREATEROLE

NOREPLICATION

VALID UNTIL 'infinity';

**- Alter user binhtv4 with superuser;**

- ALTER GROUP grp\_superadmin ADD USER binhtv4;

**-- Check lock:**

SELECT \* FROM pg\_locks pl LEFT JOIN pg\_stat\_activity psa

ON pl.pid = psa.pid where usename='siem';

SELECT blocked\_locks.pid AS blocked\_pid,

blocked\_activity.usename AS blocked\_user,

blocking\_locks.pid AS blocking\_pid,

blocking\_activity.usename AS blocking\_user,

blocked\_activity.query AS blocked\_statement,

blocking\_activity.query AS current\_statement\_in\_blocking\_process

FROM pg\_catalog.pg\_locks blocked\_locks

JOIN pg\_catalog.pg\_stat\_activity blocked\_activity ON blocked\_activity.pid = blocked\_locks.pid

JOIN pg\_catalog.pg\_locks blocking\_locks

ON blocking\_locks.locktype = blocked\_locks.locktype

AND blocking\_locks.database IS NOT DISTINCT FROM blocked\_locks.database

AND blocking\_locks.relation IS NOT DISTINCT FROM blocked\_locks.relation

AND blocking\_locks.page IS NOT DISTINCT FROM blocked\_locks.page

AND blocking\_locks.tuple IS NOT DISTINCT FROM blocked\_locks.tuple

AND blocking\_locks.virtualxid IS NOT DISTINCT FROM blocked\_locks.virtualxid

AND blocking\_locks.transactionid IS NOT DISTINCT FROM blocked\_locks.transactionid

AND blocking\_locks.classid IS NOT DISTINCT FROM blocked\_locks.classid

AND blocking\_locks.objid IS NOT DISTINCT FROM blocked\_locks.objid

AND blocking\_locks.objsubid IS NOT DISTINCT FROM blocked\_locks.objsubid

AND blocking\_locks.pid != blocked\_locks.pid

JOIN pg\_catalog.pg\_stat\_activity blocking\_activity ON blocking\_activity.pid = blocking\_locks.pid

WHERE NOT blocked\_locks.granted;

**--Check câu lệnh chạy lâu:**

SELECT pid, now() - pg\_stat\_activity.query\_start AS duration, state,

query FROM pg\_stat\_activity WHERE state ='active' AND query NOT ILIKE '%pg\_stat\_activity%'

AND (now() - pg\_stat\_activity.query\_start) > interval '30 seconds'

ORDER BY duration desc;

**-- Kill:**

select pg\_cancel\_backend(9722);

**-- Check replicat:**

On master:

select \* from pg\_stat\_replication;

On replica:

select \* from pg\_stat\_wal\_receiver;

postgres=# select pg\_is\_in\_recovery();

pg\_is\_in\_recovery

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

t

(1 row)

postgres=# select pg\_last\_xlog\_receive\_location();

pg\_last\_xlog\_receive\_location

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

0/540C1DB8

postgres=# select pg\_last\_xlog\_replay\_location();

pg\_last\_xlog\_replay\_location

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

0/540C1DB8

(1 row)

postgres=# SELECT CASE WHEN pg\_last\_xlog\_receive\_location() = pg\_last\_xlog\_replay\_location()

THEN 0

ELSE EXTRACT (EPOCH FROM now() - pg\_last\_xact\_replay\_timestamp())

END AS log\_delay;

log\_delay

-----------

0

(1 row)

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

**-- export:**

time pg\_dumpall -r -l API > /backupfs/dump/role\_api.sql

time pg\_dumpall -t -l API > /backupfs/dump/tablespace\_api.sql

time pg\_dump -Fc API -n api\_ewallet > /backupfs/dump/api.dump

**-- import:**

time psql -f /tmp/role\_api.sql postgres

time psql -f tablespace\_api.sql postgres

time pg\_restore -d API /tmp/api.dump

**-- authen md5 pgpool:**

# cd /data01/pgpool/bin/

# ./pg\_md5 --md5auth --username=u\_ewallet Ewal

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

**Golive chuyển db lên cloud:**

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

time pg\_dumpall -r -l API > /tmp/role\_api.sql

time pg\_dumpall -t -l API > /tmp/tablespace\_api.sql

pg\_dump -s -t \_api.auth\_request\_in\_new4\_id\_seq -t \_api.auth\_request\_out\_new4\_id\_seq -t \_api.auth\_request\_in -t \_api.auth\_request\_out\_new4\_id\_seq -t \_api.auth\_request\_out -t \_api.sms\_tbl\_command\_id\_seq -t \_api.sms\_tbl\_command -t \_api.sms\_tbl\_error\_id\_seq -t \_api.sms\_tbl\_error -t \_api.sms\_tbl\_request\_in\_id\_new1\_seq -t \_api.sms\_tbl\_request\_in -f /tmp/sequence.sql API

pg\_dump -s -t \_api.auth\_context -t \_api.auth\_context\_activity -t \_api.auth\_location\_activity -t \_api.auth\_locking\_user -t \_api.auth\_login\_fail -t \_api.auth\_otp -t \_api.auth\_request\_arc -t \_api.auth\_request\_error -t \_api.auth\_request\_in -t \_api.auth\_request\_out -t \_api.auth\_reset\_pass -t \_api.auth\_session -t \_api.cust\_info\_account -t \_api.cust\_info\_account\_addinfo -t \_api.cust\_info\_contracts -t \_api.cust\_info\_customer -t \_api.cust\_info\_customer\_corp\_addinfo -t \_api.cust\_info\_customer\_corp\_session -t \_api.cust\_info\_customer\_ind\_addinfo -t \_api.cust\_info\_customer\_ind\_session -t \_api.cust\_info\_loan\_products -t \_api.cust\_info\_loan\_products\_addinfo -t \_api.cust\_info\_request\_arc -t \_api.cust\_info\_request\_error -t \_api.cust\_info\_request\_in -t \_api.cust\_info\_request\_out -t \_api.cust\_info\_request\_out\_20210701 -t \_api.cust\_info\_user\_profile -t \_api.sms\_tbl\_command -t \_api.sms\_tbl\_error -t \_api.sms\_tbl\_request\_arc -t \_api.sms\_tbl\_request\_arc\_old -t \_api.sms\_tbl\_request\_error -t \_api.sms\_tbl\_request\_in -t \_api.sms\_tbl\_request\_in\_old -t \_api.sms\_tbl\_request\_out -t \_api.sms\_tbl\_request\_out\_arc -t \_api.sms\_tbl\_request\_out\_arc\_old -t \_api.sms\_tbl\_sms\_template -f /tmp/table\_metadata.sql API

pg\_dump --insert -t \_api.\_api\_channel\_config -t \_api.\_api\_config -t \_api.\_api\_config\_activity -t \_api.\_api\_error -f /tmp/table\_config.sql API

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

time psql -f /tmp/role\_api.sql postgres

time psql -f /tmp/tablespace\_api.sql postgres

time pg\_restore -d API /tmp/table\_metadata.dump

time pg\_restore -d API /tmp/table\_config.dump

 time psql --dbname=API -U postgres -f /tmp/table\_config.sql

time psql --dbname=API -U postgres -f /tmp/table\_metadata.sql

time psql --dbname=API -U postgres -f /tmp/sequence.sql

SELECT \* FROM pg\_catalog.pg\_tables where schemaname='\_api' order by tablename;

 SELECT \* FROM pg\_catalog.pg\_tables where schemaname='\_api' and tablename like 'sms\_%' order by tablename;

SELECT \* FROM pg\_catalog.pg\_tables where schemaname='\_api' and tablename like 'cust\_info\_%' order by tablename;

SELECT \* FROM pg\_catalog.pg\_tables where schemaname='\_api' and tablename like 'auth\_%' order by tablename;

select \* from \_api.\_api\_channel\_config;

select \* from \_api.\_api\_config;

select \* from \_api.\_api\_config\_activity;

select \* from \_api.\_api\_channel\_config;

 ----------------

 pg\_md5 --md5auth --username=lg\_apiauthen xxx

pg\_md5 --md5auth --username=lg\_custinfo xxx

pg\_md5 --md5auth --username=lg\_apisms xxx

GRANT ALL ON SCHEMA \_api TO lg\_apisms;

ALTER DEFAULT PRIVILEGES IN SCHEMA \_api

GRANT INSERT, SELECT, UPDATE, DELETE ON TABLES TO lg\_apisms;

ALTER DEFAULT PRIVILEGES IN SCHEMA \_api

GRANT ALL ON TABLES TO lg\_apisms;

GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO lg\_swib\_website;

GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA public TO lg\_swib\_website;

ALTER TABLE api\_ewallet.wallet\_his OWNER TO grp\_ewallet;

ALTER TABLE api\_ewallet.wallet\_addition\_type SET TABLESPACE sea\_tblspc\_api;

ALTER SCHEMA eba OWNER TO u\_nextgen\_pilot;

**--grant all schema**

SELECT format(

'GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA %I TO u\_nextgen\_pilot;',

nspname

)

FROM pg\_namespace

WHERE nspname NOT in ('pg\_catalog','information\_schema')

\gexec

   -**-check size:**

select table\_schema, table\_name, pg\_relation\_size('"'||table\_schema||'"."'||table\_name||'"')/1024/1024/1024 as "table\_size (GB)"

from information\_schema.tables

where

table\_schema not in ('pg\_catalog','information\_schema')

and table\_type = 'BASE TABLE'

and table\_name like 'sms\_tbl%'

order by 3 desc;

**--Create new table (include new index) + sequence: done**

SET search\_path TO \_api;

ALTER TABLE test\_out RENAME TO test\_out\_old1;

ALTER TABLE test\_out\_new RENAME TO test\_out;

**--Archive data (90 days):**

select \* from public.tbl\_data\_in where dayid > '20211019';

COPY (SELECT \* FROM public.tbl\_data\_in WHERE dayid > '20210720') TO '/u02/backup/export/tbl\_data\_in.csv';

SET search\_path TO public;

ALTER TABLE tbl\_data\_in RENAME TO tbl\_data\_in\_20211021;

=> create new table tbl\_data\_in

COPY public.tbl\_data\_in FROM '/u02/backup/export/tbl\_data\_in.csv';

**-- Table size:**

select table\_catalog, table\_schema, table\_name, pg\_relation\_size(table\_schema||'.'||table\_name) from information\_schema.tables order by 3;

 -**-API**

 select schemaname as table\_schema, relname as table\_name, pg\_size\_pretty(pg\_relation\_size(relid)) as data\_size from pg\_catalog.pg\_statio\_user\_tables order by pg\_relation\_size(relid) desc;

## Tài liệu tham khảo:

<https://www.tranvanbinh.vn/p/tong-hop-cac-bai-viet-ve-postgresql.html>

# HƯỚNG DẪN GIÁM SÁT, VẬN HÀNH MYSQL/MARIADB

 Hầu hết chúng ta đều quen quản lý cơ sở dữ liệu MySQL/MariaDB thông qua công cụ phpMyAdmin. Thực tế công cụ phpMyAdmin quá phổ biến giúp người dùng có thể quản lý cơ sở dữ liệu một cách nhanh chóng, hiện quả thông qua giao diện trực quan. Nên dùng phpMyAdmin để quản trị cho dễ dàng, ngoài ra có thể dùng dòng lệnh

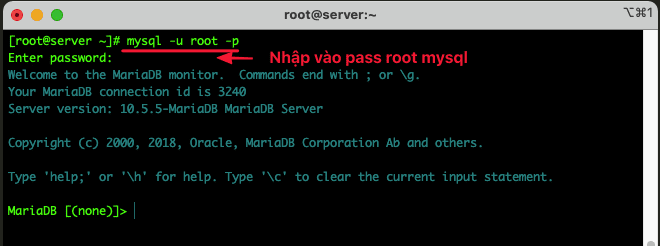
## 1. Các câu lệnh cơ bản

### a. Đăng nhập vào MariaDB

Đâu tiên cần đăng nhập vào MySQL/MariaDB user root.

Bạn nhập vào cú pháp mysql -u root -p khi đó MySQL sẽ yêu cầu bạn nhập vào Password của MySQL. Đây là password cao nhất quản lý toàn bộ cơ sở dữ liệu.

mysql -u root -p



### b. Tạo Database

Để tạo database bạn nhập vào dòng sau. Mình sẽ chú thích các lệnh sau là:

* CREATE DATABASE: Lệnh tạo database
* database1: Tên database cần tạo, bạn hãy đổi bằng tên bạn cần tạo

MariaDB [(none)]> CREATE DATABASE database1;

Các lệnh quản lý cơ sở dữ liệu MySQL/MariaDB

### c. Tạo User Password

Tạo User và password cho user bạn nhập vào dòng lệnh bên dưới, trong đó.

* CREATE USER: Lệnh tạo user
* database1: Tên user, bạn hãy thay đổi bằng tên user của bạn
* IDENTIFIED BY ‘my-password’: Thay my-password bằng password của bạn cần đặt

MariaDB [(none)]> CREATE USER 'database1'@'localhost' IDENTIFIED BY 'my-password';



### d. Gán quyền User Database

Khi đã tạo database và user xong bạn sẽ cần gán quyền để thực thi các truy vấn. Để gán bạn sử dụng cú pháp sau.

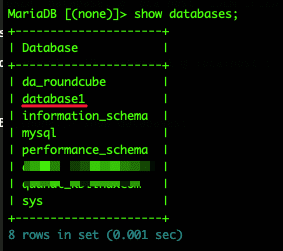
MariaDB [(none)]> GRANT ALL PRIVILEGES ON database1.\* TO 'database1'@'localhost';



### e. Show Database

Để hiển thị tất cả database bạn sử dụng lệnh show databases. Với lệnh này bạn sẽ thấy tất cả các database hiện có.

MariaDB [(none)]> MariaDB [(none)]> show databases;



### f. Truy cập vào Database

Để truy cập vào một database nào đó. Bạn sử dụng use ten-database để truy cập vào. Bạn xem

MariaDB [(none)]> use database1;

### g. Show Table Database

MariaDB [database1]> show tables;



## 2. Backup, restore

### a. Backup

Sử dụng lệnh sau

mysqldump --opt -u [uname] -p [dbname] > [backupfile.sql]

Trong đó:

**[uname]** : user của database

**[dbname] :** Tên của database

**[backupfile.sql] :** Tên file backup muốn lưu

**[–opt]** : Các tùy chọn mysqldump

Ví dụ:

mysqldump -u root -p hocvps > db\_hocvps.sql

**Backup database đồng thời nén gzip**

mysqldump -u [uname] -p [dbname] | gzip -9 > [backupfile.sql.gz]

### b. Restore

Sử dụng lệnh sau

mysql -u [uname] -p [dbname] < [backupfile.sql]

Trong đó:

**[uname]** : user của database

**[dbname]** : Tên của database

**[backupfile.sql]** : Tên file backup đã lưu (file backup muốn phục hồi)

Ví dụ:

mysql -u root -p hocvps < db\_hocvps.sql

**Phục hồi database được nén**

gunzip < [backupfile.sql.gz] | mysql -u [uname] -p [dbname]

Nếu bạn muốn phục hồi một database đã tạo ra trước đó bạn sử dụng lệnh mysqlimport. Cấu trúc lệnh như sau:

mysqlimport -u [uname] -p [dbname] [backupfile.sql]

## Tài liệu tham khảo:

# QUY TRÌNH PHỐI HỢP CÔNG VIỆC MẢNG CSDL GIỮA CÁC ĐƠN VỊ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Loại yêu cầu** | **Phương pháp thực hiện (dự kiến)** | **Ưu điểm** | **Hạn chế** | **Rủi ro** | **Cấp phê duyệt** |
| 1 | Yêu cầu tạo, khóa user | - Phòng Ứng dụng gửi văn bản đề nghị phối hợp sang phòng Hạ tầng. - Lãnh đạo Phòng Hạ tầng phân công xử lý theo văn bản | - Lãnh đạo Trung tâm nắm được công việc phát sinh. - Có hồ sơ lưu trữ |  |  | -Tạo user mới phải được LĐTCT phê duyệt trên HT NGHIỆP VỤ CORE (vì theo quy trình NGHIỆP VỤ CORE),các hệ thống khác thèo quy trình riêng của hệ thống đó |
| 2 | Yêu cầu cấp quyền Database, thêm, bớt IP kết nối | - Quản trị ứng dụng gửi đề xuất cho Lãnh đạo Phòng Ứng dụng qua trang PAKH - Lãnh đạo Phòng chuyển tiếp sang cho Lãnh đạo Phòng Hạ tầng. - Lãnh đạo Phòng Hạ tầng chuyển tiếp cho tổ trưởng tổ Database - Tổ trưởng tổ Database xử lý hoặc chuyển tiếp cho chuyên viên quản lý DB tương ứng thực hiện và báo lại. - Sau khi quản trị ứng dụng thực hiện xong thì thông báo lại để thu hồi quyền | - Lãnh đạo nắm được công việc và kiểm soát được ảnh hưởng | - Qua nhiều bước, dẫn tới bị chậm. - Phát sinh nhiều làm tăng công việc của DBA. - Khó khăn trong việc thống kê công việc | - Cấp quyền insert, update trực tiếp vào Database có thể dẫn tới dữ liệu bị tác động thay đổi ngoài mong muốn. - Việc thu hồi quyền thường bị quên dẫn tới tình trạng dư thừa quyền (Privilege creepy) - Việc cấp quyền có thể không được ghi log đầy đủ, khó kiểm tra khi có vấn đề về sau | - Quyền truy cập vào các bảng dữ liệu "nhạy cảm" yêu cầu phải có LĐTCT phê duyệt (vì theo quy trình NGHIỆP VỤ CORE), các hệ thống khác thèo quy trình riêng của hệ thống đó |
| 3 | Yêu cầu cập nhật bảng, thủ tục | - Quản trị ứng dụng gửi đề xuất cho Lãnh đạo Phòng Ứng dụng. - Lãnh đạo Phòng chuyển tiếp sang cho Lãnh đạo Phòng Hạ tầng. - Lãnh đạo Phòng Hạ tầng chuyển tiếp cho tổ trưởng tổ Database - Tổ trưởng tổ Database xử lý hoặc chuyển tiếp cho chuyên viên quản lý DB tương ứng thực hiện và báo lại. - Quản trị ứng dụng test và thông báo lại | - Lãnh đạo nắm được công việc và kiểm soát được ảnh hưởng - DBA thực hiện nên kiểm soát được ảnh hưởng. | - Qua nhiều bước, dẫn tới bị chậm. - Phát sinh nhiều làm tăng công việc của DBA. - Khó khăn trong việc thống kê công việc |  | Xin GL LĐTT phê duyệt, những ảnh hưởng > 5 phút trình LĐTCT phê duyệt |
| 4 | Yêu cầu tối ưu câu lệnh, thủ tục, object | - Quản trị ứng dụng gửi đề xuất cho Lãnh đạo Phòng Ứng dụng. - Lãnh đạo Phòng chuyển tiếp sang cho Lãnh đạo Phòng Hạ tầng. - Lãnh đạo Phòng Hạ tầng chuyển tiếp cho tổ trưởng tổ Database - Tổ trưởng tổ Database xử lý hoặc chuyển tiếp cho chuyên viên quản lý DB tương ứng thực hiện và báo lại. - Quản trị ứng dụng test và thông báo lại | - Lãnh đạo nắm được công việc và kiểm soát được ảnh hưởng - DBA thực hiện nên kiểm soát được ảnh hưởng. | - Qua nhiều bước, dẫn tới bị chậm. - Phát sinh nhiều làm tăng công việc của DBA. - Khó khăn trong việc thống kê công việc |  | Xin GL LĐTT phê duyệt, những ảnh hưởng > 5 phút trình LĐTCT phê duyệt |
| 5 | Yêu cầu bổ sung đồng bộ dữ liệu Golden Gate | - Quản trị ứng dụng gửi đề xuất cho Lãnh đạo Phòng Ứng dụng. - Lãnh đạo Phòng chuyển tiếp sang cho Lãnh đạo Phòng Hạ tầng. - Lãnh đạo Phòng Hạ tầng chuyển tiếp cho tổ trưởng tổ Database - Tổ trưởng tổ Database xử lý hoặc chuyển tiếp cho chuyên viên quản lý DB tương ứng thực hiện và báo lại. - Quản trị ứng dụng test và thông báo lại | - Lãnh đạo nắm được công việc và kiểm soát được ảnh hưởng | - Khó khăn trong việc thống kê công việc |  | Xin GL LĐTT phê duyệt  Có thể gây tăng tải cho CSDL |
| 6 | Yêu cầu giải phóng session | - Quản trị ứng dụng gửi đề xuất cho tổ trưởng tổ Database, cc lãnh đạo 2 phòng - Tổ trưởng tổ Database xử lý hoặc chuyển tiếp cho chuyên viên quản lý DB tương ứng thực hiện và báo lại. - Quản trị ứng dụng test và thông báo lại | - Lãnh đạo nắm được công việc | - Phát sinh nhiều làm tăng công việc của DBA. - Khó khăn trong việc thống kê công việc |  |  |
| 7 | Yêu cầu phối hợp kiểm tra lỗi, chậm, treo | - Quản trị ứng dụng gửi đề xuất cho tổ trưởng tổ Database, cc lãnh đạo 2 phòng - Tổ trưởng tổ Database xử lý hoặc chuyển tiếp cho chuyên viên quản lý DB tương ứng thực hiện và báo lại. - Quản trị ứng dụng test và thông báo lại | - Lãnh đạo nắm được công việc và kiểm soát được ảnh hưởng | - Khó khăn trong việc thống kê công việc |  | Khi treo, ảnh hưởng đến nhiều dịch vu, nhiều phòng thì cần thông báo, xử lý theo quy trình, quy định |
| 8 | Các yêu cầu đặc thù khác: Cài đặt database, dựng môi trường test, phối hợp triển khai dự án | - Phòng Ứng dụng gửi văn bản đề nghị phối hợp sang phòng Hạ tầng. - Lãnh đạo Phòng Hạ tầng phân công xử lý theo văn bản | - Lãnh đạo Trung tâm nắm được công việc phát sinh |  |  | Xin GL LĐTT phê duyệt vì nội dung này mất nhiều nguồn lực, tài nguyên nên LĐTT phải đồng ý chủ trương và phân bổ nguồn lực (con người, phần cứng,…) |

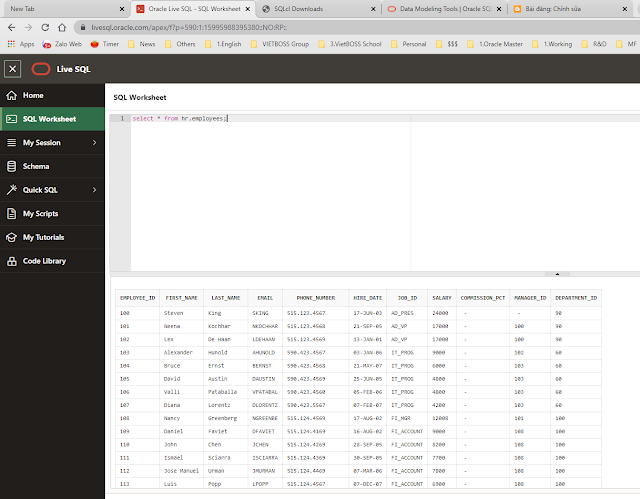
## Tài liệu tham khảo:

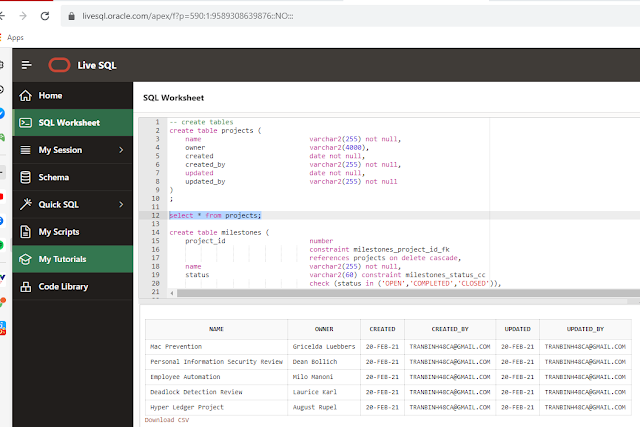
# SQL

## Lab SQL Online và Kho tài nguyên Code Libary của Oracle

Mục đích: Giúp thực hành trên trực tiếp giao diện Web mà không cần cài đặt trên PC cá nhân

Link Live sQL (SQL Worksheet): <https://livesql.oracle.com/apex/f?p=590:1:9589308639876::NO:::>

[](https://lh3.googleusercontent.com/-6IBLQa4IfDc/YLTFoZF7woI/AAAAAAAABJM/o9SyLU76lPga4hNVdxOcR_IOUxuCC2FZACLcBGAsYHQ/image.png)

[](https://lh3.googleusercontent.com/-7KroVVQa68w/YDDX_Bc5tTI/AAAAAAAAAq0/H12PkKdEbpUXG2mQ67a4btrvjt2gmLTBQCLcBGAsYHQ/image.png)

Link kho tài nguyên : <https://livesql.oracle.com/apex/f?p=590:49:9589308639876::NO:::>\

## SQL Tutorial

### SQL Tutorial - Bài 1: SELECT

Các nội dung sẽ đề cập trong

* Thiết lập môi trường
* Ký tự đại diện "\*"
* Cột (column)
* Bí danh cột
* Bí danh bảng
* Hàm
* Biểu thức
* Truy vấn vô hướng

THIẾT LẬP

Bạn có thể thực hiện tất cả các truy vấn trực tuyến miễn phí bằng [SQL Fiddle](http://sqlfiddle.com/#!4) .

Các ví dụ trong bài viết này yêu cầu phải có các bảng sau đây.

-DROP TABLE employees PURGE;

--DROP TABLE departments PURGE;

CREATE TABLE departments (

department\_id NUMBER(2) CONSTRAINT departments\_pk PRIMARY KEY,

department\_name VARCHAR2(14),

location VARCHAR2(13)

);

INSERT INTO departments VALUES (10,'ACCOUNTING','NEW YORK');

INSERT INTO departments VALUES (20,'RESEARCH','DALLAS');

INSERT INTO departments VALUES (30,'SALES','CHICAGO');

INSERT INTO departments VALUES (40,'OPERATIONS','BOSTON');

COMMIT;

CREATE TABLE employees (

employee\_id NUMBER(4) CONSTRAINT employees\_pk PRIMARY KEY,

employee\_name VARCHAR2(10),

job VARCHAR2(9),

manager\_id NUMBER(4),

hiredate DATE,

salary NUMBER(7,2),

commission NUMBER(7,2),

department\_id NUMBER(2) CONSTRAINT emp\_department\_id\_fk REFERENCES departments(department\_id)

);

INSERT INTO employees VALUES (7369,'SMITH','CLERK',7902,to\_date('17-12-1980','dd-mm-yyyy'),800,NULL,20);

INSERT INTO employees VALUES (7499,'ALLEN','SALESMAN',7698,to\_date('20-2-1981','dd-mm-yyyy'),1600,300,30);

INSERT INTO employees VALUES (7521,'WARD','SALESMAN',7698,to\_date('22-2-1981','dd-mm-yyyy'),1250,500,30);

INSERT INTO employees VALUES (7566,'JONES','MANAGER',7839,to\_date('2-4-1981','dd-mm-yyyy'),2975,NULL,20);

INSERT INTO employees VALUES (7654,'MARTIN','SALESMAN',7698,to\_date('28-9-1981','dd-mm-yyyy'),1250,1400,30);

INSERT INTO employees VALUES (7698,'BLAKE','MANAGER',7839,to\_date('1-5-1981','dd-mm-yyyy'),2850,NULL,30);

INSERT INTO employees VALUES (7782,'CLARK','MANAGER',7839,to\_date('9-6-1981','dd-mm-yyyy'),2450,NULL,10);

INSERT INTO employees VALUES (7788,'SCOTT','ANALYST',7566,to\_date('13-JUL-87','dd-mm-rr')-85,3000,NULL,20);

INSERT INTO employees VALUES (7839,'KING','PRESIDENT',NULL,to\_date('17-11-1981','dd-mm-yyyy'),5000,NULL,10);

INSERT INTO employees VALUES (7844,'TURNER','SALESMAN',7698,to\_date('8-9-1981','dd-mm-yyyy'),1500,0,30);

INSERT INTO employees VALUES (7876,'ADAMS','CLERK',7788,to\_date('13-JUL-87', 'dd-mm-rr')-51,1100,NULL,20);

INSERT INTO employees VALUES (7900,'JAMES','CLERK',7698,to\_date('3-12-1981','dd-mm-yyyy'),950,NULL,30);

INSERT INTO employees VALUES (7902,'FORD','ANALYST',7566,to\_date('3-12-1981','dd-mm-yyyy'),3000,NULL,20);

INSERT INTO employees VALUES (7934,'MILLER','CLERK',7782,to\_date('23-1-1982','dd-mm-yyyy'),1300,NULL,10);

COMMIT;

Các bảng này là một biến thể của các bảng EMP và DEPT từ lược đồ SCOTT khi cài đặt sample của Oracle. Bạn sẽ thấy rất nhiều ví dụ của Oracle trên internet bằng cách sử dụng các bảng từ lược đồ SCOTT. Bạn có thể tìm thấy các định nghĩa bảng gốc trong tập lệnh "$ORACLE\_HOME/rdbms/admin/utlsampl.sql".

KÝ TỰ ĐẠI DIỆN (WILDCARD ) "\*"

SELECT \*

FROM employees

ORDER BY employee\_id;

EMPLOYEE\_ID EMPLOYEE\_N JOB MANAGER\_ID HIREDATE SALARY COMMISSION DEPARTMENT\_ID

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

7369 SMITH CLERK 7902 17-DEC-1980 00:00:00 800 20

7499 ALLEN SALESMAN 7698 20-FEB-1981 00:00:00 1600 300 30

7521 WARD SALESMAN 7698 22-FEB-1981 00:00:00 1250 500 30

7566 JONES MANAGER 7839 02-APR-1981 00:00:00 2975 20

7654 MARTIN SALESMAN 7698 28-SEP-1981 00:00:00 1250 1400 30

7698 BLAKE MANAGER 7839 01-MAY-1981 00:00:00 2850 30

7782 CLARK MANAGER 7839 09-JUN-1981 00:00:00 2450 10

7788 SCOTT ANALYST 7566 19-APR-1987 00:00:00 3000 20

7839 KING PRESIDENT 17-NOV-1981 00:00:00 5000 10

7844 TURNER SALESMAN 7698 08-SEP-1981 00:00:00 1500 0 30

7876 ADAMS CLERK 7788 23-MAY-1987 00:00:00 1100 20

7900 JAMES CLERK 7698 03-DEC-1981 00:00:00 950 30

7902 FORD ANALYST 7566 03-DEC-1981 00:00:00 3000 20

7934 MILLER CLERK 7782 23-JAN-1982 00:00:00 1300 10

14 rows selected.

SQL>

Bạn có thể thấy từ ví dụ trên, khi chúng tôi thực hiện câu lệnh, tất cả dữ liệu từ tất cả các cột được hiển thị.

Ký tự đại diện cũng có thể được sử dụng cho nhiều bảng trong SELECT. Trong ví dụ sau, chúng tôi có hai bảng (EMPLOYEES and DEPARTMENTS) được nối (join) với nhau. Chúng tôi đã cung cấp cho họ cả bí danh bảng (e và d) và sử dụng các bí danh đó trong SELECTd. Khi truy vấn được thực thi, chúng tôi nhận được tất cả các cột từ cả hai bảng.

SELECT e.\*, d.\*

FROM employees e

JOIN departments d ON d.department\_id = e.department\_id

ORDER BY e.employee\_id;

EMPLOYEE\_ID EMPLOYEE\_N JOB MANAGER\_ID HIREDATE SALARY COMMISSION DEPARTMENT\_ID DEPARTMENT\_ID DEPARTMENT\_NAM LOCATION

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

7369 SMITH CLERK 7902 17-DEC-1980 00:00:00 800 20 20 RESEARCH DALLAS

7499 ALLEN SALESMAN 7698 20-FEB-1981 00:00:00 1600 300 30 30 SALES CHICAGO

7521 WARD SALESMAN 7698 22-FEB-1981 00:00:00 1250 500 30 30 SALES CHICAGO

7566 JONES MANAGER 7839 02-APR-1981 00:00:00 2975 20 20 RESEARCH DALLAS

7654 MARTIN SALESMAN 7698 28-SEP-1981 00:00:00 1250 1400 30 30 SALES CHICAGO

7698 BLAKE MANAGER 7839 01-MAY-1981 00:00:00 2850 30 30 SALES CHICAGO

7782 CLARK MANAGER 7839 09-JUN-1981 00:00:00 2450 10 10 ACCOUNTING NEW YORK

7788 SCOTT ANALYST 7566 19-APR-1987 00:00:00 3000 20 20 RESEARCH DALLAS

7839 KING PRESIDENT 17-NOV-1981 00:00:00 5000 10 10 ACCOUNTING NEW YORK

7844 TURNER SALESMAN 7698 08-SEP-1981 00:00:00 1500 0 30 30 SALES CHICAGO

7876 ADAMS CLERK 7788 23-MAY-1987 00:00:00 1100 20 20 RESEARCH DALLAS

7900 JAMES CLERK 7698 03-DEC-1981 00:00:00 950 30 30 SALES CHICAGO

7902 FORD ANALYST 7566 03-DEC-1981 00:00:00 3000 20 20 RESEARCH DALLAS

7934 MILLER CLERK 7782 23-JAN-1982 00:00:00 1300 10 10 ACCOUNTING NEW YORK

14 rows selected.

SQL>

CỘT (COLUMNS)

Trong nhiều trường hợp, chúng tôi sẽ tránh các ký tự đại diện và chỉ định các cột theo tên. Trong ví dụ sau, chúng tôi sẽ lấy ra hai cột (EMPLOYEE\_ID và EMPLOYEE\_NAME) từ bảng. Khi chúng tôi thực hiện truy vấn, chúng tôi chỉ có thể thấy hai cột đó được trả về.

SELECT employee\_id, employee\_name

FROM employees

ORDER BY employee\_id;

EMPLOYEE\_ID EMPLOYEE\_N

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

7369 SMITH

7499 ALLEN

7521 WARD

7566 JONES

7654 MARTIN

7698 BLAKE

7782 CLARK

7788 SCOTT

7839 KING

7844 TURNER

7876 ADAMS

7900 JAMES

7902 FORD

7934 MILLER

14 rows selected.

SQL>

BÍ DANH CỘT (ALIAS)

Chúng ta có thể thay đổi tên của các cột được trả về trong tập kết quả bằng cách sử dụng các bí danh cột. Trong ví dụ sau, chúng tôi có hai loại khác nhau. Bí danh đầu tiên không phân biệt chữ hoa chữ thường, nhưng chữ cái thứ hai, sử dụng dấu ngoặc kép, buộc trường hợp của tên cột. Bạn có thể thấy tên cột đã thay đổi trong đầu ra bên dưới.

SELECT employee\_id AS employee\_no, employee\_name AS "Name"

FROM employees

ORDER BY employee\_id;

EMPLOYEE\_NO Name

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

7369 SMITH

7499 ALLEN

7521 WARD

7566 JONES

7654 MARTIN

7698 BLAKE

7782 CLARK

7788 SCOTT

7839 KING

7844 TURNER

7876 ADAMS

7900 JAMES

7902 FORD

7934 MILLER

14 rows selected.

SQL>

BÍ DANH BẢNG

Các bí danh bảng làm cho nó dễ dàng hơn nhiều để xem các cột đã đến từ đâu. Trong ví dụ sau, chúng tôi đã sử dụng "e" làm bí danh cho bảng NHÂN VIÊN và "d" làm bí danh cho bảng SỞ HỮU. Các bí danh bảng cũng ngăn chặn các tham chiếu mơ hồ đến các cột. Cả hai bảng EMPLOYEES  và DEPARTMENTS  đều có một cột có tên là DEPARTMENT\_ID. Sử dụng bí danh cho thấy chính xác bảng mà chúng ta mong đợi nó đến từ đâu.

SELECT e.employee\_id,

e.employee\_name,

d.department\_id,

d.department\_name

FROM employees e

JOIN departments d ON e.department\_id = d.department\_id

ORDER BY e.employee\_id;

EMPLOYEE\_ID EMPLOYEE\_N DEPARTMENT\_ID DEPARTMENT\_NAM

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

7369 SMITH 20 RESEARCH

7499 ALLEN 30 SALES

7521 WARD 30 SALES

7566 JONES 20 RESEARCH

7654 MARTIN 30 SALES

7698 BLAKE 30 SALES

7782 CLARK 10 ACCOUNTING

7788 SCOTT 20 RESEARCH

7839 KING 10 ACCOUNTING

7844 TURNER 30 SALES

7876 ADAMS 20 RESEARCH

7900 JAMES 30 SALES

7902 FORD 20 RESEARCH

7934 MILLER 10 ACCOUNTING

14 rows selected.

SQL>

Chúng ta có thể thấy từ đầu ra ở trên, bí danh bảng không được bao gồm trong tên cột. Nếu bạn cần bất kỳ dấu hiệu nào của bảng nơi cột xuất phát, bạn nên sử dụng bí danh cột.

Một số người thích đủ điều kiện tất cả các cột bằng cách sử dụng tên bảng và tên cột cùng nhau. Điều đó ổn đối với tên bảng nhỏ, nhưng tôi nghĩ nó không ổn đối với tên bảng lớn.

SELECT employees.employee\_id,

employees.employee\_name,

departments.department\_id,

departments.department\_name

FROM employees

JOIN departments ON employees.department\_id = departments.department\_id

ORDER BY employees.employee\_id;

EMPLOYEE\_ID EMPLOYEE\_N DEPARTMENT\_ID DEPARTMENT\_NAM

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

7369 SMITH 20 RESEARCH

7499 ALLEN 30 SALES

7521 WARD 30 SALES

7566 JONES 20 RESEARCH

7654 MARTIN 30 SALES

7698 BLAKE 30 SALES

7782 CLARK 10 ACCOUNTING

7788 SCOTT 20 RESEARCH

7839 KING 10 ACCOUNTING

7844 TURNER 30 SALES

7876 ADAMS 20 RESEARCH

7900 JAMES 30 SALES

7902 FORD 20 RESEARCH

7934 MILLER 10 ACCOUNTING

14 rows selected.

SQL>

Giống như các bí danh bảng, tên bảng không được bao gồm trong tên cột được trả về.

HÀM

Các SELECT cũng có thể bao gồm các cuộc gọi chức năng. Chúng có thể là các hàm dựng sẵn hoặc các hàm PL / SQL mà bạn tự mã hóa. Trong ví dụ sau, chúng tôi đã sử dụng hàm  UPPER để chuyển đổi một chuỗi ký tự chứa văn bản chữ thường thành văn bản chữ hoa.

SELECT UPPER('lowercase text') AS text

FROM dual;

TEXT

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

LOWERCASE TEXT

1 row selected.

SQL>

Oracle chứa nhiều hàm dựng sẵn để xử lý chuỗi, số và ngày, v.v ... Mọi thảo luận về chúng đều nằm ngoài phạm vi của bài viết này.

BIỂU THỨC

Các SELECT có thể bao gồm một loạt các biểu thức, trong đó có các hoạt động toán học.

SELECT 1+2 AS addition

FROM dual;

ADDITION

----------

3

1 row selected.

SQL>

TRUY VẤN VÔ HƯỚNG

Có thể nguồn dữ liệu cho một cột bằng cách sử dụng truy vấn con vô hướng. Truy vấn con vô hướng là một truy vấn trả về một hàng của một cột hoặc đối tượng. Ví dụ dưới đây sử dụng truy vấn con vô hướng để trả về số lượng nhân viên cho mỗi bộ phận.

SELECT d.department\_id, d.department\_name,

(SELECT COUNT(\*) FROM employees e WHERE e.department\_id = d.department\_id) AS emp\_count

FROM departments d

ORDER BY d.department\_id;

DEPARTMENT\_ID DEPARTMENT\_NAM EMP\_COUNT

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

10 ACCOUNTING 3

20 RESEARCH 5

30 SALES 6

40 OPERATIONS 0

4 rows selected.

SQL>

Có thể sử dụng phép join:

SELECT d.department\_id, d.department\_name, COUNT(e.employee\_id) AS emp\_count

FROM departments d

LEFT OUTER JOIN employees e ON d.department\_id = e.department\_id

GROUP BY d.department\_id, d.department\_name

ORDER BY d.department\_id;

DEPARTMENT\_ID DEPARTMENT\_NAM EMP\_COUNT

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

10 ACCOUNTING 3

20 RESEARCH 5

30 SALES 6

40 OPERATIONS 0

4 rows selected.

SQL>

### SQL Tutorial - Bài 2: Mệnh đề FROM

Nội dung:

* Thiết lập môi trường
* Các bảng
* View inline
* Mệnh đề WITH
* View

THIẾT LẬP MÔI TRƯỜNG

Bạn có thể thực hiện tất cả các truy vấn trực tuyến miễn phí bằng [SQL Fiddle](http://sqlfiddle.com/#!4) .

Các ví dụ trong bài viết này yêu cầu phải có các bảng sau đây.

--DROP TABLE employees PURGE;

--DROP TABLE departments PURGE;

CREATE TABLE departments (

department\_id NUMBER(2) CONSTRAINT departments\_pk PRIMARY KEY,

department\_name VARCHAR2(14),

location VARCHAR2(13)

);

INSERT INTO departments VALUES (10,'ACCOUNTING','NEW YORK');

INSERT INTO departments VALUES (20,'RESEARCH','DALLAS');

INSERT INTO departments VALUES (30,'SALES','CHICAGO');

INSERT INTO departments VALUES (40,'OPERATIONS','BOSTON');

COMMIT;

CREATE TABLE employees (

employee\_id NUMBER(4) CONSTRAINT employees\_pk PRIMARY KEY,

employee\_name VARCHAR2(10),

job VARCHAR2(9),

manager\_id NUMBER(4),

hiredate DATE,

salary NUMBER(7,2),

commission NUMBER(7,2),

department\_id NUMBER(2) CONSTRAINT emp\_department\_id\_fk REFERENCES departments(department\_id)

);

INSERT INTO employees VALUES (7369,'SMITH','CLERK',7902,to\_date('17-12-1980','dd-mm-yyyy'),800,NULL,20);

INSERT INTO employees VALUES (7499,'ALLEN','SALESMAN',7698,to\_date('20-2-1981','dd-mm-yyyy'),1600,300,30);

INSERT INTO employees VALUES (7521,'WARD','SALESMAN',7698,to\_date('22-2-1981','dd-mm-yyyy'),1250,500,30);

INSERT INTO employees VALUES (7566,'JONES','MANAGER',7839,to\_date('2-4-1981','dd-mm-yyyy'),2975,NULL,20);

INSERT INTO employees VALUES (7654,'MARTIN','SALESMAN',7698,to\_date('28-9-1981','dd-mm-yyyy'),1250,1400,30);

INSERT INTO employees VALUES (7698,'BLAKE','MANAGER',7839,to\_date('1-5-1981','dd-mm-yyyy'),2850,NULL,30);

INSERT INTO employees VALUES (7782,'CLARK','MANAGER',7839,to\_date('9-6-1981','dd-mm-yyyy'),2450,NULL,10);

INSERT INTO employees VALUES (7788,'SCOTT','ANALYST',7566,to\_date('13-JUL-87','dd-mm-rr')-85,3000,NULL,20);

INSERT INTO employees VALUES (7839,'KING','PRESIDENT',NULL,to\_date('17-11-1981','dd-mm-yyyy'),5000,NULL,10);

INSERT INTO employees VALUES (7844,'TURNER','SALESMAN',7698,to\_date('8-9-1981','dd-mm-yyyy'),1500,0,30);

INSERT INTO employees VALUES (7876,'ADAMS','CLERK',7788,to\_date('13-JUL-87', 'dd-mm-rr')-51,1100,NULL,20);

INSERT INTO employees VALUES (7900,'JAMES','CLERK',7698,to\_date('3-12-1981','dd-mm-yyyy'),950,NULL,30);

INSERT INTO employees VALUES (7902,'FORD','ANALYST',7566,to\_date('3-12-1981','dd-mm-yyyy'),3000,NULL,20);

INSERT INTO employees VALUES (7934,'MILLER','CLERK',7782,to\_date('23-1-1982','dd-mm-yyyy'),1300,NULL,10);

COMMIT;

Các bảng này là một biến thể của các bảng EMP và DEPT từ lược đồ SCOTT. Bạn sẽ thấy rất nhiều ví dụ của Oracle trên internet bằng cách sử dụng các bảng từ lược đồ SCOTT. Bạn có thể tìm thấy các định nghĩa bảng gốc trong tập lệnh "$ORACLE\_HOME/rdbms/admin/utlsampl.sql".

BẢNG

Thông thường mệnh đề FROM liệt kê các bảng và điều kiện nối được sử dụng trong truy vấn. Hình thức đơn giản nhất của nó là khi chỉ có một bảng duy nhất được truy cập, như hiển thị bên dưới. Trong ví dụ này, tôi đã yêu cầu dữ liệu từ tất cả các cột từ bảng EMPLOYEES  và chúng tôi đã yêu cầu dữ liệu được sắp xếp theo cột EMPLOYEE\_ID. Vì tất cả dữ liệu đến từ một bảng duy nhất, chúng tôi chỉ có một bảng được tham chiếu trong mệnh đề FROM.

SELECT \*

FROM employees

ORDER BY employee\_id;

EMPLOYEE\_ID EMPLOYEE\_N JOB MANAGER\_ID HIREDATE SALARY COMMISSION DEPARTMENT\_ID

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

7369 SMITH CLERK 7902 17-DEC-1980 00:00:00 800 20

7499 ALLEN SALESMAN 7698 20-FEB-1981 00:00:00 1600 300 30

7521 WARD SALESMAN 7698 22-FEB-1981 00:00:00 1250 500 30

7566 JONES MANAGER 7839 02-APR-1981 00:00:00 2975 20

7654 MARTIN SALESMAN 7698 28-SEP-1981 00:00:00 1250 1400 30

7698 BLAKE MANAGER 7839 01-MAY-1981 00:00:00 2850 30

7782 CLARK MANAGER 7839 09-JUN-1981 00:00:00 2450 10

7788 SCOTT ANALYST 7566 19-APR-1987 00:00:00 3000 20

7839 KING PRESIDENT 17-NOV-1981 00:00:00 5000 10

7844 TURNER SALESMAN 7698 08-SEP-1981 00:00:00 1500 0 30

7876 ADAMS CLERK 7788 23-MAY-1987 00:00:00 1100 20

7900 JAMES CLERK 7698 03-DEC-1981 00:00:00 950 30

7902 FORD ANALYST 7566 03-DEC-1981 00:00:00 3000 20

7934 MILLER CLERK 7782 23-JAN-1982 00:00:00 1300 10

14 rows selected.

SQL>

Đầu ra ở trên hiển thị tất cả dữ liệu từ tất cả các cột trong bảng EMPLOYEES , như mong đợi.

Trong nhiều trường hợp, chúng ta cần nguồn dữ liệu từ nhiều bảng cùng một lúc. Điều này được thực hiện bằng cách sử dụng tham gia. Trong ví dụ sau, chúng tôi đang sử dụng phép nối kiểu ANSI giữa các bảng EMPLOYEES và DEPARTMENTS . Việc nối (join) hoặc khớp được thực hiện bằng cột DEPARTMENT\_ID trong mỗi bảng. Khi chúng ta thực hiện phép nối kiểu ANSI, tất cả các bảng và điều kiện nối của chúng được liệt kê trong mệnh đề FROM. Khi chúng tôi thực hiện truy vấn sau, chúng tôi sẽ thấy mỗi tên nhân viên (EMPPLOYEE.EMPLOYEE\_NAME) và tên bộ phận của họ (DEPARTMENT.DEPARTMENT\_NAME).

SELECT e.employee\_name, d.department\_name

FROM employees e

JOIN departments d ON d.department\_id = e.department\_id

ORDER BY e.employee\_name;

EMPLOYEE\_N DEPARTMENT\_NAM

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

ADAMS RESEARCH

ALLEN SALES

BLAKE SALES

CLARK ACCOUNTING

FORD RESEARCH

JAMES SALES

JONES RESEARCH

KING ACCOUNTING

MARTIN SALES

MILLER ACCOUNTING

SCOTT RESEARCH

SMITH RESEARCH

TURNER SALES

WARD SALES

14 rows selected.

SQL>

Nhiều nhà phát triển Oracle thích sử dụng cú pháp không phải ANSI để tham gia. Các bảng được tham chiếu như một danh sách được phân tách bằng dấu phẩy trong mệnh đề FROM và các điều kiện nối là các vị từ trong mệnh đề WHERE. Câu lệnh sau tương đương về chức năng với câu lệnh trước.

SELECT e.employee\_name, d.department\_name

FROM employees e, departments d

WHERE d.department\_id = e.department\_id

ORDER BY e.employee\_name;

EMPLOYEE\_N DEPARTMENT\_NAM

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

ADAMS RESEARCH

ALLEN SALES

BLAKE SALES

CLARK ACCOUNTING

FORD RESEARCH

JAMES SALES

JONES RESEARCH

KING ACCOUNTING

MARTIN SALES

MILLER ACCOUNTING

SCOTT RESEARCH

SMITH RESEARCH

TURNER SALES

WARD SALES

14 rows selected.

SQL>

Có một số biến thể của các phép nối, được mô tả ở đây.

VIEWS NỘI TUYẾN (INLINE VIEW)

Bạn sẽ thường thấy các truy vấn con trong mệnh đề FROM. Chúng được gọi là inline views. Sau khi được xác định và đưa ra một bí danh, chúng có thể được nối và tham chiếu như một bảng thông thường. Câu lệnh sau đây có chức năng tương đương với hai ví dụ trước, nhưng phép nối đã được chuyển sang dạng inline view.

SELECT ed.employee\_name, ed.department\_name

FROM (SELECT e.employee\_name, d.department\_name

FROM employees e, departments d

WHERE d.department\_id = e.department\_id) ed

ORDER BY ed.employee\_name;

EMPLOYEE\_N DEPARTMENT\_NAM

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

ADAMS RESEARCH

ALLEN SALES

BLAKE SALES

CLARK ACCOUNTING

FORD RESEARCH

JAMES SALES

JONES RESEARCH

KING ACCOUNTING

MARTIN SALES

MILLER ACCOUNTING

SCOTT RESEARCH

SMITH RESEARCH

TURNER SALES

WARD SALES

14 rows selected.

SQL>

MỆNH ĐỀ WITH

Một thay thế cho một inline view là để di chuyển truy vấn con ra mệnh đề WITH. Sau đó chúng ta có thể tham chiếu truy vấn có tên này trong mệnh đề FROM của câu lệnh SELECT chính . Điều này có thể được sử dụng để làm cho một mệnh đề FROM rất phức tạp trông đơn giản hơn nhiều.

WITH emp\_dept\_join AS (

SELECT e.employee\_name, d.department\_name

FROM employees e, departments d

WHERE d.department\_id = e.department\_id

)

SELECT ed.employee\_name, ed.department\_name

FROM emp\_dept\_join ed

ORDER BY ed.employee\_name;

EMPLOYEE\_N DEPARTMENT\_NAM

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

ADAMS RESEARCH

ALLEN SALES

BLAKE SALES

CLARK ACCOUNTING

FORD RESEARCH

JAMES SALES

JONES RESEARCH

KING ACCOUNTING

MARTIN SALES

MILLER ACCOUNTING

SCOTT RESEARCH

SMITH RESEARCH

TURNER SALES

WARD SALES

14 rows selected.

SQL>

Mệnh đề WITH được thảo luận chi tiết hơn ở đây .

VIEWS

Một cách khác là tạo ra một khung nhìn (view) thông thường dựa trên truy vấn con.

CREATE OR REPLACE VIEW emp\_dept\_join\_v AS

SELECT e.employee\_name, d.department\_name

FROM employees e, departments d

WHERE d.department\_id = e.department\_id;

View ẩn một số cột quan trọng hoặc 1 số điều kiện.

SELECT ed.employee\_name, ed.department\_name

FROM emp\_dept\_join\_v ed

ORDER BY ed.employee\_name;

EMPLOYEE\_N DEPARTMENT\_NAM

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

ADAMS RESEARCH

ALLEN SALES

BLAKE SALES

CLARK ACCOUNTING

FORD RESEARCH

JAMES SALES

JONES RESEARCH

KING ACCOUNTING

MARTIN SALES

MILLER ACCOUNTING

SCOTT RESEARCH

SMITH RESEARCH

TURNER SALES

WARD SALES

14 rows selected.

SQL>

### SQL Tutorial - Bài 3: Mệnh đề WHERE

Nội dung

* Thiết lập môi trường
* Giới thiệu
* Điều kiện đẳng thức và bất đẳng thức
* Điều kiện IN và NOT IN trong
* Điều kiện EXISTS và NOT EXISTS
* Điều kiện BETWEEN và NOT BETWEEN
* Điều kiện LIKE và NOT LIKE
* Điều kiện OR
* Phần kết luận

THIẾT LẬP MÔI TRƯỜNG

Bạn có thể thực hiện tất cả các truy vấn trực tuyến miễn phí bằng [SQL Fiddle](http://sqlfiddle.com/#!4) .

Các ví dụ trong bài viết này yêu cầu phải có các bảng sau đây.

--DROP TABLE employees PURGE;

--DROP TABLE departments PURGE;

CREATE TABLE departments (

department\_id NUMBER(2) CONSTRAINT departments\_pk PRIMARY KEY,

department\_name VARCHAR2(14),

location VARCHAR2(13)

);

INSERT INTO departments VALUES (10,'ACCOUNTING','NEW YORK');

INSERT INTO departments VALUES (20,'RESEARCH','DALLAS');

INSERT INTO departments VALUES (30,'SALES','CHICAGO');

INSERT INTO departments VALUES (40,'OPERATIONS','BOSTON');

COMMIT;

CREATE TABLE employees (

employee\_id NUMBER(4) CONSTRAINT employees\_pk PRIMARY KEY,

employee\_name VARCHAR2(10),

job VARCHAR2(9),

manager\_id NUMBER(4),

hiredate DATE,

salary NUMBER(7,2),

commission NUMBER(7,2),

department\_id NUMBER(2) CONSTRAINT emp\_department\_id\_fk REFERENCES departments(department\_id)

);

INSERT INTO employees VALUES (7369,'SMITH','CLERK',7902,to\_date('17-12-1980','dd-mm-yyyy'),800,NULL,20);

INSERT INTO employees VALUES (7499,'ALLEN','SALESMAN',7698,to\_date('20-2-1981','dd-mm-yyyy'),1600,300,30);

INSERT INTO employees VALUES (7521,'WARD','SALESMAN',7698,to\_date('22-2-1981','dd-mm-yyyy'),1250,500,30);

INSERT INTO employees VALUES (7566,'JONES','MANAGER',7839,to\_date('2-4-1981','dd-mm-yyyy'),2975,NULL,20);

INSERT INTO employees VALUES (7654,'MARTIN','SALESMAN',7698,to\_date('28-9-1981','dd-mm-yyyy'),1250,1400,30);

INSERT INTO employees VALUES (7698,'BLAKE','MANAGER',7839,to\_date('1-5-1981','dd-mm-yyyy'),2850,NULL,30);

INSERT INTO employees VALUES (7782,'CLARK','MANAGER',7839,to\_date('9-6-1981','dd-mm-yyyy'),2450,NULL,10);

INSERT INTO employees VALUES (7788,'SCOTT','ANALYST',7566,to\_date('13-JUL-87','dd-mm-rr')-85,3000,NULL,20);

INSERT INTO employees VALUES (7839,'KING','PRESIDENT',NULL,to\_date('17-11-1981','dd-mm-yyyy'),5000,NULL,10);

INSERT INTO employees VALUES (7844,'TURNER','SALESMAN',7698,to\_date('8-9-1981','dd-mm-yyyy'),1500,0,30);

INSERT INTO employees VALUES (7876,'ADAMS','CLERK',7788,to\_date('13-JUL-87', 'dd-mm-rr')-51,1100,NULL,20);

INSERT INTO employees VALUES (7900,'JAMES','CLERK',7698,to\_date('3-12-1981','dd-mm-yyyy'),950,NULL,30);

INSERT INTO employees VALUES (7902,'FORD','ANALYST',7566,to\_date('3-12-1981','dd-mm-yyyy'),3000,NULL,20);

INSERT INTO employees VALUES (7934,'MILLER','CLERK',7782,to\_date('23-1-1982','dd-mm-yyyy'),1300,NULL,10);

COMMIT;

Các bảng này là một biến thể của các bảng EMP và DEPT từ lược đồ SCOTT. Bạn sẽ thấy rất nhiều ví dụ của Oracle trên internet bằng cách sử dụng các bảng từ lược đồ SCOTT. Bạn có thể tìm thấy các định nghĩa bảng gốc trong tập lệnh "$ORACLE\_HOME/rdbms/admin/utlsampl.sql".

GIỚI THIỆU

Mệnh đề WHERE được sử dụng để lọc các tập kết quả. Nếu các điều kiện lọc trong mệnh đề WHERE ước tính thành FALSE cho một hàng cụ thể, hàng đó sẽ không được trả về trong tập kết quả cuối cùng. Nhiều điều kiện lọc có thể được nhóm lại với nhau bằng cách sử dụng ANDvà ORđiều kiện.

Khi sử dụng cú pháp nối không ANSI, mệnh đề WHERE chứa cả điều kiện nối và điều kiện lọc.

SELECT e.employee\_name, e.salary, d.department\_id, d.department\_name

FROM employees e, departments d

**WHERE e.department\_id = d.department\_id -- Điều kiện join**

**AND d.department\_id = 20 -- Lọc**

**AND e.salary >= 2000 -- Lọc**

ORDER BY e.employee\_name;

EMPLOYEE\_N SALARY DEPARTMENT\_ID DEPARTMENT\_NAM

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

FORD 3000 20 RESEARCH

JONES 2975 20 RESEARCH

SCOTT 3000 20 RESEARCH

3 rows selected.

SQL>

Khi sử dụng cú pháp nối ANSI, mệnh đề WHERE chỉ chứa các điều kiện lọc.

SELECT e.employee\_name, e.salary, d.department\_id, d.department\_name

FROM employees e

JOIN departments d ON e.department\_id = d.department\_id

WHERE **d.department\_id = 20**   **-- Lọc**

**AND e.salary >= 2000 -- Lọc**

ORDER BY e.employee\_name;

EMPLOYEE\_N SALARY DEPARTMENT\_ID DEPARTMENT\_NAM

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

FORD 3000 20 RESEARCH

JONES 2975 20 RESEARCH

SCOTT 3000 20 RESEARCH

3 rows selected.

SQL>

Trong phần còn lại của bài viết này, chúng tôi sẽ xem xét các loại điều kiện bạn có thể thấy trong mệnh đề WHERE. Hãy nhớ rằng, đây không phải là một danh sách đầy đủ, chỉ là một số trong những điều phổ biến nhất. Để giữ cho mọi thứ đơn giản, các ví dụ sau sẽ sử dụng cú pháp nối ANSI, vì vậy mệnh đề WHERE chỉ chứa các điều kiện lọc.

ĐIỀU KIỆN BÌNH ĐẲNG VÀ BẤT BÌNH ĐẲNG

Điều kiện lọc có thể sử dụng các bài kiểm tra đẳng thức và bất đẳng thức cơ bản.

* =: Bằng nhau.
* <> hoặc! =: Không bằng nhau.
* >: Lớn hơn.
* <: Ít hơn.
* > =: Tuyệt vời hơn hoặc bằng.
* <=: Nhỏ hơn hoặc bằng.

SELECT e.employee\_name, e.department\_id, e.salary

FROM employees e

WHERE **e.department\_id = 20**

AND **e.salary >= 2000**

ORDER BY e.employee\_name;

EMPLOYEE\_N DEPARTMENT\_ID SALARY

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

FORD 20 3000

JONES 20 2975

SCOTT 20 3000

3 rows selected.

SQL>

ĐIỀU KIỆN IN VÀ NOT IN

Điều kiện IN ước tính thành TRUE cho các giá trị cột trong danh sách đã chỉ định.

SELECT e.department\_id, e.employee\_id, e.employee\_name

FROM employees e

WHERE **e.department\_id IN (10, 20)**

ORDER BY e.department\_id, e.employee\_id;

DEPARTMENT\_ID EMPLOYEE\_ID EMPLOYEE\_N

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

10 7782 CLARK

10 7839 KING

10 7934 MILLER

20 7369 SMITH

20 7566 JONES

20 7788 SCOTT

20 7876 ADAMS

20 7902 FORD

8 rows selected.

SQL>

Điều kiện NOT IN ước tính thành TRUE cho các giá trị cột không có trong danh sách đã chỉ định.

SELECT e.department\_id, e.employee\_id, e.employee\_name

FROM employees e

WHERE **e.department\_id NOT IN (10, 20)**

ORDER BY e.department\_id, e.employee\_id;

DEPARTMENT\_ID EMPLOYEE\_ID EMPLOYEE\_N

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

30 7499 ALLEN

30 7521 WARD

30 7654 MARTIN

30 7698 BLAKE

30 7844 TURNER

30 7900 JAMES

6 rows selected.

SQL>

Các điều kiện IN và NOT IN có thể được đánh giá theo các giá trị được trả về bởi một truy vấn con.

SELECT e.department\_id, e.employee\_id, e.employee\_name

FROM employees e

WHERE **e.department\_id IN (SELECT d.department\_id**

**FROM departments d**

**WHERE d.department\_id < 30)**

ORDER BY e.department\_id, e.employee\_id;

DEPARTMENT\_ID EMPLOYEE\_ID EMPLOYEE\_N

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

10 7782 CLARK

10 7839 KING

10 7934 MILLER

20 7369 SMITH

20 7566 JONES

20 7788 SCOTT

20 7876 ADAMS

20 7902 FORD

8 rows selected.

SQL>

ĐIỀU KIỆN EXISTS VÀ NOT EXISTS

Điều kiện EXISTS ước tính thành TRUE nếu truy vấn con trả về một hoặc nhiều hàng.

SELECT d.department\_id, d.department\_name

FROM departments d

WHERE EXISTS (SELECT 1

FROM employees e

WHERE d.department\_id = e.department\_id)

ORDER BY d.department\_id;

DEPARTMENT\_ID DEPARTMENT\_NAM

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

10 ACCOUNTING

20 RESEARCH

30 SALES

3 rows selected.

SQL>

Điều kiện NOT EXISTS ước tính thành TRUE nếu truy vấn con trả về 0 hàng.

SELECT d.department\_id, d.department\_name

FROM departments d

WHERE **NOT EXISTS (SELECT 1**

**FROM employees e**

**WHERE d.department\_id = e.department\_id)**

ORDER BY d.department\_id;

DEPARTMENT\_ID DEPARTMENT\_NAM

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

40 OPERATIONS

1 row selected.

SQL>

ĐIỀU KIỆN BETWEEN VÀ NOT BETWEEN

Điều kiện BETWEEN ước tính thành TRUE cho các giá trị cột trong phạm vi được chỉ định, bao gồm các ranh giới.

SELECT d.department\_id, d.department\_name

FROM departments d

WHERE **department\_id BETWEEN 20 AND 40**

ORDER BY d.department\_id;

DEPARTMENT\_ID DEPARTMENT\_NAM

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

20 RESEARCH

30 SALES

40 OPERATIONS

3 rows selected.

Điều kiện NOT BETWEEN ước tính thành TRUE cho các giá trị cột không nằm trong phạm vi được chỉ định, bao gồm các ranh giới.

SELECT d.department\_id, d.department\_name

FROM departments d

WHERE **department\_id NOT BETWEEN 20 AND 40**

ORDER BY d.department\_id;

DEPARTMENT\_ID DEPARTMENT\_NAM

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

10 ACCOUNTING

1 row selected.

SQL>

ĐIỀU KIỆN LIKE VÀ NOT LIKE

Điều kiện LIKE ước tính thành TRUE nếu có mẫu khớp. '%' Là ký tự đại diện có chiều dài thay đổi. '\_' Là ký tự đại diện một ký tự.

SELECT d.department\_id, d.department\_name

FROM departments d

WHERE **department\_name LIKE '%O%'**

ORDER BY d.department\_id;

DEPARTMENT\_ID DEPARTMENT\_NAM

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

10 ACCOUNTING

40 OPERATIONS

2 rows selected.

SQL>

Điều kiện NOT LIKE ước tính thành TRUE nếu không có mẫu khớp.

SELECT d.department\_id, d.department\_name

FROM departments d

WHERE **department\_name NOT LIKE '%O%'**

ORDER BY d.department\_id;

DEPARTMENT\_ID DEPARTMENT\_NAM

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

20 RESEARCH

30 SALES

2 rows selected.

SQL>

ĐIỀU KIỆN OR

Hãy cẩn thận khi liên kết các điều kiện với nhau trong mệnh đề WHERE bằng cách sử dụng điều kiện OR. Thật dễ dàng để làm rối logic của bạn. Trong ví dụ sau tôi nghĩ rằng tôi đã yêu cầu các nhà quản lý và thư ký ở department\_id=20.

SELECT e.employee\_id, e.employee\_name, e.department\_id, e.salary, e.job

FROM employees e

WHERE e.department\_id = 20

AND e.job = 'MANAGER'

OR e.job = 'CLERK'

ORDER BY e.employee\_id;

EMPLOYEE\_ID EMPLOYEE\_N DEPARTMENT\_ID SALARY JOB

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

7369 SMITH 20 800 CLERK

7566 JONES 20 2975 MANAGER

7876 ADAMS 20 1100 CLERK

7900 JAMES 30 950 CLERK

7934 MILLER 10 1300 CLERK

5 rows selected.

SQL>

Chúng ta có thể thấy từ đầu ra, chúng tôi đã trả lại nhân viên từ bộ phận 30 và 10. Điều này là do điều kiện OR đã hủy bỏ điều kiện bộ phận nếu nhân viên là nhân viên bán hàng.

Chúng ta nên sử dụng dấu ngoặc đơn để đảm bảo rằng chúng ta đã hỏi đúng câu hỏi.

SELECT e.employee\_id, e.employee\_name, e.department\_id, e.salary, e.job

FROM employees e

WHERE e.department\_id = 20

**AND (e.job = 'MANAGER' OR e.job = 'CLERK')**

ORDER BY e.employee\_id;

EMPLOYEE\_ID EMPLOYEE\_N DEPARTMENT\_ID SALARY JOB

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

7369 SMITH 20 800 CLERK

7566 JONES 20 2975 MANAGER

7876 ADAMS 20 1100 CLERK

3 rows selected.

SQL>

PHẦN KẾT LUẬN

Có nhiều chức năng, điều kiện và toán tử mà bạn sẽ tiếp xúc trong suốt quá trình làm việc với SQL, nhưng tất cả chúng đều là những bổ sung trên nền tảng này. Hiểu những điều cơ bản và những thứ khác sẽ rơi vào vị trí theo thời gian.

Người mới bắt đầu thường thấy các phép nối (join) khá khó hiểu và có xu hướng hấp dẫn các điều kiện IN và EXISTS để lọc dữ liệu, ngay cả trong trường hợp sẽ hiệu quả hơn khi tham gia vào bảng cụ thể và sử dụng bộ lọc đơn giản để chống lại dùng join. Trong những ngày đầu, bạn có thể sẽ tập trung vào việc chỉ cố gắng để có được kết quả chính xác, nhưng để phát triển lâu dài, bạn nên thử càng nhiều phương pháp khác nhau để có kết quả giống nhau nhất có thể và so sánh hiệu suất.

Bạn cũng cần lưu ý rằng những gì bạn viết không phải lúc nào cũng là những gì Oracle chạy. Trong một số trường hợp, trình tối ưu hóa (optimizer) sẽ chuyển đổi (viết lại) câu lệnh của bạn. Ví dụ, một điều kiện IN sử dụng danh sách có thể được mở rộng thành một nhóm các điều kiện bình đẳng được liên kết với các điều kiện OR. Lúc đầu có thể gây bỡ ngỡ cho bạn, nhưng khi bạn làm lâu về sau, bạn sẽ cần bắt đầu xem xét điều này khi chọn điều kiện lọc để sử dụng.

### SQL Tutorial - Bài 4: Mệnh đề ORDER BY

Trong bài viết này, chúng ta sẽ xem làm thế nào bạn có thể ảnh hưởng đến thứ tự của dữ liệu được trả về bởi các truy vấn.

* Thiết lập môi trường
* Giới thiệu
* Xác định các Cột: Biểu thức, Vị trí và Bí danh Cột
* Thứ tự tăng dần (ASC) và giảm dần (DESC)
* Xử lý NULL: NULLS FIRST và NULLS LAST

THIẾT LẬP MÔI TRƯỜNG

Bạn có thể thực hiện tất cả các truy vấn trực tuyến miễn phí bằng [SQL Fiddle](http://sqlfiddle.com/#!4) .

Các ví dụ trong bài viết này yêu cầu phải có các bảng sau đây.

--DROP TABLE employees PURGE;

--DROP TABLE departments PURGE;

CREATE TABLE departments (

department\_id NUMBER(2) CONSTRAINT departments\_pk PRIMARY KEY,

department\_name VARCHAR2(14),

location VARCHAR2(13)

);

INSERT INTO departments VALUES (10,'ACCOUNTING','NEW YORK');

INSERT INTO departments VALUES (20,'RESEARCH','DALLAS');

INSERT INTO departments VALUES (30,'SALES','CHICAGO');

INSERT INTO departments VALUES (40,'OPERATIONS','BOSTON');

COMMIT;

CREATE TABLE employees (

employee\_id NUMBER(4) CONSTRAINT employees\_pk PRIMARY KEY,

employee\_name VARCHAR2(10),

job VARCHAR2(9),

manager\_id NUMBER(4),

hiredate DATE,

salary NUMBER(7,2),

commission NUMBER(7,2),

department\_id NUMBER(2) CONSTRAINT emp\_department\_id\_fk REFERENCES departments(department\_id)

);

INSERT INTO employees VALUES (7369,'SMITH','CLERK',7902,to\_date('17-12-1980','dd-mm-yyyy'),800,NULL,20);

INSERT INTO employees VALUES (7499,'ALLEN','SALESMAN',7698,to\_date('20-2-1981','dd-mm-yyyy'),1600,300,30);

INSERT INTO employees VALUES (7521,'WARD','SALESMAN',7698,to\_date('22-2-1981','dd-mm-yyyy'),1250,500,30);

INSERT INTO employees VALUES (7566,'JONES','MANAGER',7839,to\_date('2-4-1981','dd-mm-yyyy'),2975,NULL,20);

INSERT INTO employees VALUES (7654,'MARTIN','SALESMAN',7698,to\_date('28-9-1981','dd-mm-yyyy'),1250,1400,30);

INSERT INTO employees VALUES (7698,'BLAKE','MANAGER',7839,to\_date('1-5-1981','dd-mm-yyyy'),2850,NULL,30);

INSERT INTO employees VALUES (7782,'CLARK','MANAGER',7839,to\_date('9-6-1981','dd-mm-yyyy'),2450,NULL,10);

INSERT INTO employees VALUES (7788,'SCOTT','ANALYST',7566,to\_date('13-JUL-87','dd-mm-rr')-85,3000,NULL,20);

INSERT INTO employees VALUES (7839,'KING','PRESIDENT',NULL,to\_date('17-11-1981','dd-mm-yyyy'),5000,NULL,10);

INSERT INTO employees VALUES (7844,'TURNER','SALESMAN',7698,to\_date('8-9-1981','dd-mm-yyyy'),1500,0,30);

INSERT INTO employees VALUES (7876,'ADAMS','CLERK',7788,to\_date('13-JUL-87', 'dd-mm-rr')-51,1100,NULL,20);

INSERT INTO employees VALUES (7900,'JAMES','CLERK',7698,to\_date('3-12-1981','dd-mm-yyyy'),950,NULL,30);

INSERT INTO employees VALUES (7902,'FORD','ANALYST',7566,to\_date('3-12-1981','dd-mm-yyyy'),3000,NULL,20);

INSERT INTO employees VALUES (7934,'MILLER','CLERK',7782,to\_date('23-1-1982','dd-mm-yyyy'),1300,NULL,10);

COMMIT;

Các bảng này là một biến thể của các bảng EMP và DEPT từ lược đồ SCOTT. Bạn sẽ thấy rất nhiều ví dụ của Oracle trên internet bằng cách sử dụng các bảng từ lược đồ SCOTT. Bạn có thể tìm thấy các định nghĩa bảng gốc trong tập lệnh "$ORACLE\_HOME/rdbms/admin/utlsampl.sql".

GIỚI THIỆU

Nơi tốt nhất để bắt đầu, là với trích dẫn này từ tài liệu ( [ở đây](https://docs.oracle.com/database/121/SQLRF/statements_10002.htm#i2171079) ).

"Sử dụng mệnh đề ORDER BY để sắp xếp các hàng được trả về bởi câu lệnh. Nếu không có order\_by\_clause, không có đảm bảo nào tồn tại rằng cùng một truy vấn được thực hiện nhiều lần sẽ truy xuất các hàng theo cùng một thứ tự."

Khi bạn bắt đầu học SQL, rất dễ quên điều này và bắt đầu tin rằng có một mẫu có thể dự đoán được với đầu ra của dữ liệu. Có một vài lý do điều này có thể xảy ra.

* Nếu bạn đang sử dụng một lượng nhỏ dữ liệu thử nghiệm, tất cả các row có thể được tải vào một block, do đó, các row có thể được trả về theo cách có thể dự đoán được. Khi bạn làm việc vỡi liệu hệ thống production, giả định của bạn có thể được chứng minh là sai.
* Có thể có một loại ngầm định được thực hiện bởi một hoặc nhiều thao tác trong truy vấn của bạn, làm cho đầu ra có thể dự đoán được, nhưng đó chỉ có thể là phiên bản cụ thể và có thể là phiên bản vá bạn đang chạy. Oracle có thể thay đổi thuật toán tại bất kỳ thời điểm nào trong tương lai, điều này có thể "phá vỡ" ứng dụng của bạn. Oracle đã nói với bạn cách bảo vệ bản thân trong tuyên bố trên.

Các tài liệu sử dụng mô tả văn bản sau đây cho các mệnh đề ORDER BY.

ORDER [ SIBLINGS ] BY

{ expr | position | c\_alias }

[ ASC | DESC ]

[ NULLS FIRST | NULLS LAST ]

[, { expr | position | c\_alias }

[ ASC | DESC ]

[ NULLS FIRST | NULLS LAST ]

]...

Các phần sau đây sẽ trình bày một số tùy chọn này.

XÁC ĐỊNH CÁC CỘT: BIỂU THỨC, VỊ TRÍ VÀ BÍ DANH CỘT

Có ba cách chính để xác định các cột có trong hoạt động sắp xếp. Có lẽ biểu thức phổ biến nhất là chỉ định một hoặc nhiều cột trong danh sách được phân tách bằng dấu phẩy. Trong ví dụ sau, các kết quả được sắp xếp theo các cột SALARY và COMMISSION.

SELECT e.salary, e.commission, e.employee\_name

FROM employees e

WHERE department\_id = 30

ORDER BY **e.salary, e.commission;**

SALARY COMMISSION EMPLOYEE\_N

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

950 JAMES

1250 500 WARD

1250 1400 MARTIN

1500 0 TURNER

1600 300 ALLEN

2850 BLAKE

6 rows selected.

SQL>

Trong số những thứ khác, biểu thức có thể bao gồm nhiều cột được kết hợp với nhau. Trong ví dụ sau, các kết quả được sắp xếp theo tổng của các cột SALARY và COMMISSION. Các hàm NVL chuyển đổi bất kỳ giá trị NULL trong cột COMMISSION bằng không, để làm cho kết quả của việc bổ sung rõ ràng hơn.

SELECT e.salary, e.commission, e.employee\_name

FROM employees e

WHERE department\_id = 30

ORDER BY e.salary + NVL(e.commission,0);

SALARY COMMISSION EMPLOYEE\_N

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

950 JAMES

1500 0 TURNER

1250 500 WARD

1600 300 ALLEN

1250 1400 MARTIN

2850 BLAKE

6 rows selected.

SQL>

Các cột có thể được tham chiếu bởi vị trí cột của chúng. Hãy nhớ rằng, nếu bạn thay đổi SELECT, bạn cũng sẽ phải sửa đổi mệnh đề ORDER BY.

SELECT e.salary, e.commission, e.employee\_name

FROM employees e

WHERE department\_id = 30

ORDER BY 1;

SALARY COMMISSION EMPLOYEE\_N

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

950 JAMES

1250 500 WARD

1250 1400 MARTIN

1500 0 TURNER

1600 300 ALLEN

2850 BLAKE

6 rows selected.

SQL>

Các cột cũng có thể được tham chiếu bằng bí danh cột của chúng. Trong ví dụ sau, cột SALARY được đặt bí danh là SAL, được sử dụng trong mệnh đề ORDER BY.

SELECT e.salary **AS sal**, e.commission, e.employee\_name

FROM employees e

WHERE department\_id = 30

ORDER BY **sal**;

SAL COMMISSION EMPLOYEE\_N

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

950 JAMES

1250 500 WARD

1250 1400 MARTIN

1500 0 TURNER

1600 300 ALLEN

2850 BLAKE

6 rows selected.

SQL>

THỨ TỰ TĂNG DẦN (ASC) VÀ GIẢM DẦN (DESC)

Thứ tự mặc định là**tăng dần**, vì vậy các câu lệnh sau là tương đương về chức năng.

-- Tăng đần (ASC) là mặc định.

SELECT e.salary AS sal, e.commission, e.employee\_name

FROM employees e

WHERE department\_id = 30

ORDER BY **e.salary**;

SAL COMMISSION EMPLOYEE\_N

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

950 JAMES

1250 500 WARD

1250 1400 MARTIN

1500 0 TURNER

1600 300 ALLEN

2850 BLAKE

6 rows selected.

SQL>

-- Tường minh ASC.

SELECT e.salary AS sal, e.commission, e.employee\_name

FROM employees e

WHERE department\_id = 30

ORDER BY **e.salary ASC**;

SAL COMMISSION EMPLOYEE\_N

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

950 JAMES

1250 500 WARD

1250 1400 MARTIN

1500 0 TURNER

1600 300 ALLEN

2850 BLAKE

6 rows selected.

SQL>

Để chuyển sang giảm dần, sử dụng từ khóa DESC.

SELECT e.salary AS sal, e.commission, e.employee\_name

FROM employees e

WHERE department\_id = 30

ORDER BY **e.salary DESC;**

SAL COMMISSION EMPLOYEE\_N

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

2850 BLAKE

1600 300 ALLEN

1500 0 TURNER

1250 500 WARD

1250 1400 MARTIN

950 JAMES

6 rows selected.

SQL>

Mỗi cột trong mệnh đề ORDER BY có thể có một thứ tự khác nhau.

SELECT e.salary, e.commission, e.employee\_name

FROM employees e

WHERE department\_id = 30

ORDER BY **e.salary ASC, e.commission DESC;**

SALARY COMMISSION EMPLOYEE\_N

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

950 JAMES

1250 1400 MARTIN

1250 500 WARD

1500 0 TURNER

1600 300 ALLEN

2850 BLAKE

6 rows selected.

SQL>

XỬ LÝ NULL: NULLS FIRST VÀ NULLS LAST

Một thứ tự tăng dần giả định NULLS LAST. Bạn có thể chỉ định nó một cách rõ ràng nếu bạn muốn.

SELECT e.commission, e.employee\_name

FROM employees e

WHERE department\_id = 30

ORDER BY e.commission ASC;

COMMISSION EMPLOYEE\_N

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

0 TURNER

300 ALLEN

500 WARD

1400 MARTIN

JAMES

BLAKE

6 rows selected.

SQL>

Mặc định có thể được thay đổi bằng cách chỉ định NULLS FIRST.

SELECT e.commission, e.employee\_name

FROM employees e

WHERE department\_id = 30

ORDER BY **e.commission ASC NULLS FIRST;**

COMMISSION EMPLOYEE\_N

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

JAMES

BLAKE

0 TURNER

300 ALLEN

500 WARD

1400 MARTIN

6 rows selected.

SQL>

Một thứ tự giảm dần giả định NULLS FIRST. Bạn có thể chỉ định nó một cách rõ ràng nếu bạn muốn.

SELECT e.commission, e.employee\_name

FROM employees e

WHERE department\_id = 30

ORDER BY **e.commission DESC;**

COMMISSION EMPLOYEE\_N

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

BLAKE

JAMES

1400 MARTIN

500 WARD

300 ALLEN

0 TURNER

6 rows selected.

SQL>

Mặc định có thể được thay đổi bằng cách chỉ định NULLS LAST.

SELECT e.commission, e.employee\_name

FROM employees e

WHERE department\_id = 30

ORDER BY **e.commission DESC NULLS LAST;**

COMMISSION EMPLOYEE\_N

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

1400 MARTIN

500 WARD

300 ALLEN

0 TURNER

JAMES

BLAKE

6 rows selected.

SQL>

Mỗi cột trong mệnh đề ORDER BY có thể có cách xử lý NULL khác nhau.

SELECT e.salary, e.commission, e.employee\_name

FROM employees e

WHERE department\_id = 30

ORDER BY **e.salary ASC NULLS FIRST, e.commission DESC NULLS LAST;**

SALARY COMMISSION EMPLOYEE\_N

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

950 JAMES

1250 1400 MARTIN

1250 500 WARD

1500 0 TURNER

1600 300 ALLEN

2850 BLAKE

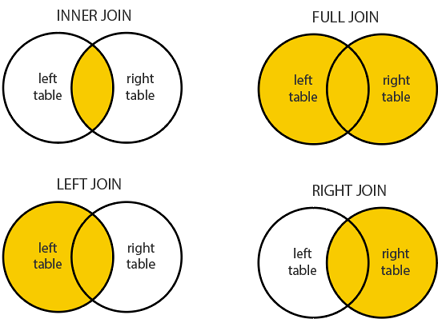
6 rows selected.

SQL>

### SQL Tutorial - Bài 5: JOIN

Trong bài viết này, chúng tôi sẽ xem xét một số liên kết phổ biến, cả ANSI và không ANSI, có sẵn trong SQL.

* Thiết lập môi trường
* Giới thiệu
* [INNER] JOIN ... ON
* LEFT [OUTER] JOIN
* RIGHT [OUTER] JOIN
* FULL [OUTER] JOIN
* CROSS JOIN
* NATURAL JOIN
* [INNER] JOIN ... USING

[](https://lh3.googleusercontent.com/-n76c6dtr5sw/YBO5d-3PzGI/AAAAAAAAAh8/xiT6YIzePLEXArb8uU1f1vgg8JRXYeg8ACLcBGAsYHQ/image.png)

THIẾT LẬP MÔI TRƯỜNG

Bạn có thể thực hiện tất cả các truy vấn trực tuyến miễn phí bằng [SQL Fiddle](http://sqlfiddle.com/#!4) .

Các ví dụ trong bài viết này yêu cầu phải có các bảng sau đây.

--DROP TABLE employees PURGE;

--DROP TABLE departments PURGE;

CREATE TABLE departments (

department\_id NUMBER(2) CONSTRAINT departments\_pk PRIMARY KEY,

department\_name VARCHAR2(14),

location VARCHAR2(13)

);

INSERT INTO departments VALUES (10,'ACCOUNTING','NEW YORK');

INSERT INTO departments VALUES (20,'RESEARCH','DALLAS');

INSERT INTO departments VALUES (30,'SALES','CHICAGO');

INSERT INTO departments VALUES (40,'OPERATIONS','BOSTON');

COMMIT;

CREATE TABLE employees (

employee\_id NUMBER(4) CONSTRAINT employees\_pk PRIMARY KEY,

employee\_name VARCHAR2(10),

job VARCHAR2(9),

manager\_id NUMBER(4),

hiredate DATE,

salary NUMBER(7,2),

commission NUMBER(7,2),

department\_id NUMBER(2) CONSTRAINT emp\_department\_id\_fk REFERENCES departments(department\_id)

);

INSERT INTO employees VALUES (7369,'SMITH','CLERK',7902,to\_date('17-12-1980','dd-mm-yyyy'),800,NULL,20);

INSERT INTO employees VALUES (7499,'ALLEN','SALESMAN',7698,to\_date('20-2-1981','dd-mm-yyyy'),1600,300,30);

INSERT INTO employees VALUES (7521,'WARD','SALESMAN',7698,to\_date('22-2-1981','dd-mm-yyyy'),1250,500,30);

INSERT INTO employees VALUES (7566,'JONES','MANAGER',7839,to\_date('2-4-1981','dd-mm-yyyy'),2975,NULL,20);

INSERT INTO employees VALUES (7654,'MARTIN','SALESMAN',7698,to\_date('28-9-1981','dd-mm-yyyy'),1250,1400,30);

INSERT INTO employees VALUES (7698,'BLAKE','MANAGER',7839,to\_date('1-5-1981','dd-mm-yyyy'),2850,NULL,30);

INSERT INTO employees VALUES (7782,'CLARK','MANAGER',7839,to\_date('9-6-1981','dd-mm-yyyy'),2450,NULL,10);

INSERT INTO employees VALUES (7788,'SCOTT','ANALYST',7566,to\_date('13-JUL-87','dd-mm-rr')-85,3000,NULL,20);

INSERT INTO employees VALUES (7839,'KING','PRESIDENT',NULL,to\_date('17-11-1981','dd-mm-yyyy'),5000,NULL,10);

INSERT INTO employees VALUES (7844,'TURNER','SALESMAN',7698,to\_date('8-9-1981','dd-mm-yyyy'),1500,0,30);

INSERT INTO employees VALUES (7876,'ADAMS','CLERK',7788,to\_date('13-JUL-87', 'dd-mm-rr')-51,1100,NULL,20);

INSERT INTO employees VALUES (7900,'JAMES','CLERK',7698,to\_date('3-12-1981','dd-mm-yyyy'),950,NULL,30);

INSERT INTO employees VALUES (7902,'FORD','ANALYST',7566,to\_date('3-12-1981','dd-mm-yyyy'),3000,NULL,20);

INSERT INTO employees VALUES (7934,'MILLER','CLERK',7782,to\_date('23-1-1982','dd-mm-yyyy'),1300,NULL,10);

COMMIT;

Các bảng này là một biến thể của các bảng EMP và DEPT từ lược đồ SCOTT. Bạn sẽ thấy rất nhiều ví dụ của Oracle trên internet bằng cách sử dụng các bảng từ lược đồ SCOTT. Bạn có thể tìm thấy các định nghĩa bảng gốc trong tập lệnh "$ORACLE\_HOME/rdbms/admin/utlsampl.sql".

GIỚI THIỆU

Các phép nối (join) được sử dụng để kết hợp dữ liệu từ nhiều bảng để tạo thành một tập kết quả duy nhất. Oracle cung cấp hai cách tiếp cận để tham gia các bảng, cú pháp nối không ANSI và cú pháp nối ANSI, trông khá khác nhau.

Cú pháp tham gia không phải ANSI trong lịch sử là cách bạn thực hiện tham gia trong Oracle và ngày nay nó vẫn rất phổ biến. Các bảng được nối được liệt kê trong  mệnh đềFROM và các điều kiện nối được định nghĩa là các vị từ trong mệnh đề WHERE. Ngay cả khi bạn không thích nó, bạn sẽ phải làm quen với nó vì có rất nhiều mã ngoài đó vẫn sử dụng nó. Nếu bạn không quen với cú pháp, bạn sẽ phải vật lộn để sửa lỗi bất kỳ mã hiện có nào và một số ví dụ trên internet sẽ trông khá bí ẩn đối với bạn.

Cú pháp tham gia ANSI được giới thiệu trong Oracle 9i . Nó có một số lợi thế so với cú pháp ban đầu.

* Nó đọc giống tiếng Anh hơn, vì vậy nó rõ ràng hơn nhiều.
* Các bảng và điều kiện nối đều được giữ cùng nhau trong mệnh đề FROM, vì vậy mệnh đề WHEREchỉ chứa các bộ lọc, không chứa các điều kiện nối.
* Cú pháp gây khó khăn, nếu không nói là không thể bao gồm điều kiện nối.
* Các bộ lọc trên các cột từ các bảng tham gia bên ngoài được xử lý theo cách rõ ràng hơn nhiều.
* Nó dễ mang theo hơn, được hỗ trợ bởi một số công cụ cơ sở dữ liệu quan hệ.
* Nó cung cấp một số chức năng không được hỗ trợ trực tiếp bởi cú pháp tham gia không phải ANSI, mà không cần sử dụng nhiều nỗ lực hơn.

Bất chấp tất cả những lợi thế này, nhiều nhà phát triển của Oracle vẫn sử dụng cú pháp tham gia không phải ANSI. Một phần điều này chỉ vì thói quen. Một phần điều này là do trình tối ưu hóa Oracle chuyển đổi hầu hết cú pháp nối ANSI thành cú pháp nối không ANSI tương đương trước khi nó được thực thi.

Đối với người mới bắt đầu, ý kiến ​​cá nhân của tôi là bạn nên tập trung vào cú pháp tham gia ANSI, nhưng lưu ý về tương đương không phải ANSI. Trong bài viết này tôi sẽ chỉ ra cú pháp ANSI và không phải ANSI cho mỗi ví dụ, khi có liên quan.

Một số phương thức tham gia phổ biến hơn các phương pháp khác, vì vậy ban đầu tập trung sự chú ý của bạn vào những phương pháp bạn có thể thấy nhất. Các phép nối phổ biến nhất mà bạn có thể thấy trong mã là như sau.

* [INNER] JOIN ... ON
* LEFT [OUTER] JOIN
* RIGHT [OUTER] JOIN

Sau đây là ít phổ biến hơn.

* FULL [OUTER] JOIN
* CROSS JOIN
* NATURAL JOIN
* [INNER] JOIN ... USING

Sau đây là rất hiếm tại thời điểm này.

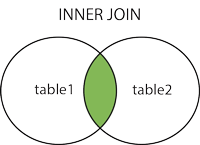
* CROSS APPLY
* OUTER APPLY

Nếu một từ được bao quanh bởi "[]" thì có nghĩa đó là một từ khóa tùy chọn. Không có bất kỳ vòng loại nào khác, tham gia là một tham gia bên trong, vì vậy sử dụng từ khóa INNER là không cần thiết. Nếu liên kết bao gồm các từ LEFT, RIGHT hoặc FULL, theo định nghĩa là join ngoài, vì vậy từ khóa OUTER là dư thừa. Lựa chọn bao gồm hoặc loại trừ những từ này thực sự là sở thích cá nhân, vì vậy hãy tuân theo tiêu chuẩn trong công ty của bạn hoặc làm những gì cảm thấy phù hợp với bạn.

Với tất cả những gì trong tâm trí, chúng ta hãy xem xét một số ví dụ.

[INNER] JOIN ... ON

Một dữ liệu INNER JOIN kết hợp từ hai bảng trong đó có một kết quả khớp trên (các) cột tham gia trong cả hai bảng.

**[](https://lh3.googleusercontent.com/-__gzvvkn3HU/YBO6XISBa_I/AAAAAAAAAiI/rEJTnazfgw8W0daijzXaNCAsMMaLgtljACLcBGAsYHQ/image.png)**

Hãy nhớ rằng, từ khóa INNER là tùy chọn. Trong các ví dụ bên dưới, chúng tôi sẽ trả lại DEPARTMENT\_NAME và EMPLOYEE\_NAME cho mỗi nhân viên. Bộ phận OPERATIONS có DEPARTMENT\_ID là 40, do đó, nó không bị xóa bởi điều kiện bộ lọc, nhưng không có nhân viên nào trong bộ phận này, do đó không có kết quả trùng khớp và nó không được trả về trong tập kết quả.

Dưới đây là một ví dụ về ANSI INNER JOIN.

SELECT d.department\_name,

e.employee\_name

FROM departments d

JOIN employees e ON d.department\_id = e.department\_id

WHERE d.department\_id >= 30

ORDER BY d.department\_name;

DEPARTMENT\_NAM EMPLOYEE\_N

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

SALES ALLEN

SALES BLAKE

SALES JAMES

SALES MARTIN

SALES TURNER

SALES WARD

6 rows selected.

SQL>

Đây là tương đương non-ANSI của câu lệnh trước.

SELECT d.department\_name,

e.employee\_name

FROM  **departments d, employees e**

**WHERE d.department\_id = e.department\_id**

AND d.department\_id >= 30

ORDER BY d.department\_name;

DEPARTMENT\_NAM EMPLOYEE\_N

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

SALES ALLEN

SALES BLAKE

SALES JAMES

SALES MARTIN

SALES TURNER

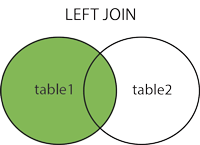
SALES WARD

6 rows selected.

SQL>

LEFT [OUTER] JOIN (JOIN TRÁI)

A LEFT [OUTER] JOIN trả về tất cả các hàng hợp lệ từ bảng ở phía bên trái của từ khóa JOIN, cùng với các giá trị từ bảng ở phía bên phải hoặc NULL nếu một hàng phù hợp không tồn tại.

[](https://lh3.googleusercontent.com/-FPK3p64Cmtw/YBO6Z0kvEDI/AAAAAAAAAiM/wKDPVTms95QfkdHrhHHM86LrIBJZ4IRRQCLcBGAsYHQ/image.png)

Sử dụng ví dụ trước, nhưng chuyển sang một LEFT OUTER JOIN, chúng ta sẽ thấy bộ phận OPERATIONS , mặc dù nó không có nhân viên.

Dưới đây là một ví dụ về ANSI LEFT OUTER JOIN.

SELECT d.department\_name,

e.employee\_name

FROM departments d

LEFT OUTER JOIN employees e ON d.department\_id = e.department\_id

WHERE d.department\_id >= 30

ORDER BY d.department\_name, e.employee\_name;

DEPARTMENT\_NAM EMPLOYEE\_N

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

OPERATIONS

SALES ALLEN

SALES BLAKE

SALES JAMES

SALES MARTIN

SALES TURNER

SALES WARD

7 rows selected.

SQL>

Đây là tương đương non-ANSI của câu lệnh trước. Lưu ý "(+)" được sử dụng để chỉ ra phía của điều kiện join có thể bị thiếu. Đối với điều kiện nối nhiều cột, mỗi cột phải có "(+)". Không giống như cú pháp nối ANSI, cú pháp nối non-ANSI không bị ảnh hưởng bởi thứ tự của các bảng.

SELECT d.department\_name,

e.employee\_name

FROM departments d, employees e

WHERE d.department\_id = e.department\_id (+)

AND d.department\_id >= 30

ORDER BY d.department\_name, e.employee\_name;

DEPARTMENT\_NAM EMPLOYEE\_N

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

OPERATIONS

SALES ALLEN

SALES BLAKE

SALES JAMES

SALES MARTIN

SALES TURNER

SALES WARD

7 rows selected.

SQL>

Thêm bộ lọc (filter) vào các cột được trả về từ bảng đã tham gia bên ngoài (outer) là nguyên nhân phổ biến gây nhầm lẫn. Nếu bạn kiểm tra một giá trị cụ thể, ví dụ "salary> = 2000", nhưng giá trị cho cột SALARY là NULL vì hàng bị thiếu, một điều kiện thông thường trong mệnh đề WHERE sẽ ném hàng đi, do đó đánh bại đối tượng của outer join. Cả hai phương pháp ANSI và non-ANSI đều có cách xử lý vấn đề này.

Sử dụng cú pháp  ANSI join, các bộ lọc trên các cột từ bảng nối ngoài được bao gồm trong chính phép nối, thay vì được đặt trong mệnh đề WHERE.

SELECT d.department\_name,

e.employee\_name

FROM departments d

LEFT OUTER JOIN employees e ON d.department\_id = e.department\_id AND e.salary >= 2000

WHERE d.department\_id >= 30

ORDER BY d.department\_name, e.employee\_name;

DEPARTMENT\_NAM EMPLOYEE\_N

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

OPERATIONS

SALES BLAKE

2 rows selected.

SQL>

Sử dụng cú pháp nối non-ANSI, "(+)" được sử dụng để chỉ ra một cột có thể có giá trị NULL là kết quả của phép outer join.

SELECT d.department\_name,

e.employee\_name

FROM departments d, employees e

WHERE d.department\_id = e.department\_id (+)

AND e.salary (+) >= 2000

AND d.department\_id >= 30

ORDER BY d.department\_name, e.employee\_name;

DEPARTMENT\_NAM EMPLOYEE\_N

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

OPERATIONS

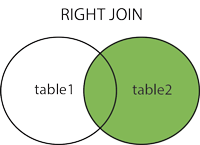
SALES BLAKE

2 rows selected.

SQL>

RIGHT [OUTER] JOIN (JOIN PHẢI)

Sự RIGHT [OUTER] JOIN đối lập của LEFT [OUTER] JOIN. Nó trả về tất cả các hàng hợp lệ từ bảng ở phía bên phải của từ khóa JOIN, cùng với các giá trị từ bảng ở phía bên trái hoặc NULL nếu một hàng phù hợp không tồn tại. Tất cả các điểm nêu ra trong phần trước cũng áp dụng ở đây.

[](https://lh3.googleusercontent.com/-nFkgYRqHv_A/YBO6gBMY9uI/AAAAAAAAAiQ/PZAn_D9LMtImlYSbdIh1PCqxPeWCBA2HgCLcBGAsYHQ/image.png)

Ví dụ sau đây đã thay đổi thứ tự của các bảng để RIGHT [OUTER] JOIN bây giờ bắt buộc phải có.

SELECT d.department\_name,

e.employee\_name

FROM employees e

RIGHT OUTER JOIN departments d ON e.department\_id = d.department\_id

WHERE d.department\_id >= 30

ORDER BY d.department\_name, e.employee\_name;

DEPARTMENT\_NAM EMPLOYEE\_N

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

OPERATIONS

SALES ALLEN

SALES BLAKE

SALES JAMES

SALES MARTIN

SALES TURNER

SALES WARD

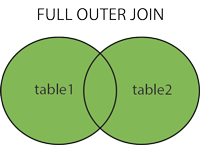
7 rows selected.

SQL>

Hãy nhớ rằng, cú pháp nối ngoài non-ANSI không phụ thuộc vào thứ tự bảng, do đó không có khái niệm thực sự về các phép nối ngoài phải hoặc trái, chỉ là các phép nối ngoài.

FULL [OUTER] JOIN (JOIN ĐẦY ĐỦ)

A FULL [OUTER] JOIN kết hợp tất cả các hàng từ các bảng ở bên trái và bên phải của phép join. Nếu một trong hai bên bị thiếu dữ liệu, nó sẽ được thay thế bằng NULL, thay vì bỏ hàng đi.

[](https://lh3.googleusercontent.com/-Easp0p_HABU/YBO6j0GSpGI/AAAAAAAAAiU/gBmLAtNJQT8rR-9jkddYWfeiQmNxO4edQCLcBGAsYHQ/image.png)

Để xem một ví dụ hoạt động, chúng ta cần thêm một nhân viên khác không được chỉ định vào một bộ phận.

INSERT INTO employees VALUES (8888,'JONES','DBA',null,to\_date('02-1-1982','dd-mm-yyyy'),1300,NULL,NULL);

COMMIT;

Dưới đây là một ví dụ về ANSI FULL OUTER JOIN.

SELECT d.department\_name,

e.employee\_name

FROM **employees e**

**FULL OUTER JOIN departments d ON e.department\_id = d.department\_id**

ORDER BY d.department\_name, e.employee\_name;

DEPARTMENT\_NAM EMPLOYEE\_N

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

ACCOUNTING CLARK

ACCOUNTING KING

ACCOUNTING MILLER

OPERATIONS

RESEARCH ADAMS

RESEARCH FORD

RESEARCH JONES

RESEARCH SCOTT

RESEARCH SMITH

SALES ALLEN

SALES BLAKE

SALES JAMES

SALES MARTIN

SALES TURNER

SALES WARD

JONES

16 rows selected.

SQL>

Không có tương đương trực tiếp với phép nối ngoài đầy đủ bằng cú pháp nối non-ANSI, nhưng chúng ta có thể tạo lại nó bằng cách kết hợp hai truy vấn nối ngoài bằng cách sử dụng a UNION ALL, như được hiển thị bên dưới.

SELECT d.department\_name,

e.employee\_name

FROM employees e, departments d

WHERE e.department\_id = d.department\_id (+)

UNION ALL

SELECT d.department\_name,

e.employee\_name

FROM departments d, employees e

WHERE d.department\_id = e.department\_id (+)

AND e.employee\_name IS NULL

ORDER BY 1, 2;

DEPARTMENT\_NAM EMPLOYEE\_N

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

ACCOUNTING CLARK

ACCOUNTING KING

ACCOUNTING MILLER

OPERATIONS

RESEARCH ADAMS

RESEARCH FORD

RESEARCH JONES

RESEARCH SCOTT

RESEARCH SMITH

SALES ALLEN

SALES BLAKE

SALES JAMES

SALES MARTIN

SALES TURNER

SALES WARD

JONES

16 rows selected.

SQL>

Thật thú vị, khi bạn chạy ANSI FULL OUTER JOIN, trình tối ưu hóa của Oracle viết lại nó thành tương đương non-ANSI, do đó không có cải tiến hiệu suất liên quan đến nó. Nó chỉ dễ nhìn hơn.

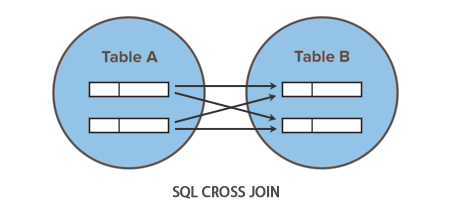
Hãy loại bỏ nhân viên bổ sung đó để nó không ảnh hưởng đến bất kỳ ví dụ nào khác.

DELETE FROM employees WHERE employee\_id = 8888;

COMMIT;

CROSS JOIN

CROSS JOIN là sự tạo ra có chủ ý của một sản phẩm của Cartesian. Không có cột tham gia nào được chỉ định, do đó, mọi sự kết hợp có thể của các hàng giữa hai bảng được tạo ra.

[](https://lh3.googleusercontent.com/-zWTe4L1eUl0/YBO6ohIICVI/AAAAAAAAAiY/kEcSgtQ1KawdJG_4pUXYWhkHkrHGMxYEgCLcBGAsYHQ/image.png)

Dưới đây là một ví dụ về ANSI CROSS JOIN.

SELECT e.employee\_name,

d.department\_name

FROM employees e

**CROSS JOIN departments d**

ORDER BY e.employee\_name, d.department\_name;

EMPLOYEE\_N DEPARTMENT\_NAM

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

ADAMS ACCOUNTING

ADAMS OPERATIONS

ADAMS RESEARCH

ADAMS SALES

... Output amended for brevity ...

WARD ACCOUNTING

WARD OPERATIONS

WARD RESEARCH

WARD SALES

**56 rows selected.**

SQL>

Đây là tương đương non-ANSI của câu lệnh trước. Lưu ý, không có điều kiện tham gia trong mệnh đề WHERE.

SELECT e.employee\_name,

d.department\_name

FROM **employees e, departments d**

ORDER BY e.employee\_name, d.department\_name;

EMPLOYEE\_N DEPARTMENT\_NAM

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

ADAMS ACCOUNTING

ADAMS OPERATIONS

ADAMS RESEARCH

ADAMS SALES

... Output amended for brevity ...

WARD ACCOUNTING

WARD OPERATIONS

WARD RESEARCH

WARD SALES

**56 rows selected.**

SQL>

NATURAL JOIN (JOIN TỰ NHIÊN)

A NATURAL JOIN là một biến thể trên một INNER JOIN. Các cột tham gia được xác định ngầm, dựa trên các tên cột. Bất kỳ cột nào có cùng tên giữa hai bảng được coi là cột join. Dưới đây là một ví dụ sử dụng cú pháp join ANSI.

ELECT e.employee\_name,

d.department\_name

FROM **employees e**

**NATURAL JOIN departments d**

ORDER BY e.employee\_name, d.department\_name;

EMPLOYEE\_N DEPARTMENT\_NAM

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

ADAMS RESEARCH

ALLEN SALES

BLAKE SALES

CLARK ACCOUNTING

FORD RESEARCH

JAMES SALES

JONES RESEARCH

KING ACCOUNTING

MARTIN SALES

MILLER ACCOUNTING

SCOTT RESEARCH

SMITH RESEARCH

TURNER SALES

WARD SALES

14 rows selected.

SQL>

Không có tương đương non-ANSI về điều này, vì tất cả các điều kiện tham gia phải được chỉ định.

Sử dụng một NATURAL JOIN là một ý tưởng tồi. Nếu ai đó thêm một cột mới vào một trong các bảng có cùng tên với một cột trong bảng khác, họ có thể phá vỡ mọi liên kết tự nhiên hiện có. Nó thực sự là một lỗi đang chờ để xảy ra.

Bạn không thể áp dụng bất kỳ bộ lọc bí danh nào cho các cột được sử dụng trong các phép join tự nhiên, như trong ví dụ sau.

SELECT e.employee\_name,

d.department\_name

FROM employees e

NATURAL JOIN departments **d**

**WHERE** d.department\_id = 20

ORDER BY e.employee\_name;

**WHERE d.department\_id = 20**

**\***

**ERROR at line 5:**

**ORA-25155: column used in NATURAL join cannot have qualifier**

SQL>

Thay vào đó, bạn phải xóa bí danh, trong các trường hợp khác sẽ dẫn đến lỗi tham chiếu mơ hồ.

SELECT e.employee\_name,

d.department\_name

FROM employees e

NATURAL JOIN departments d

WHERE **department\_id = 20**

ORDER BY e.employee\_name;

EMPLOYEE\_N DEPARTMENT\_NAM

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

ADAMS RESEARCH

FORD RESEARCH

JONES RESEARCH

SCOTT RESEARCH

SMITH RESEARCH

5 rows selected.

SQL>

[INNER] JOIN ... USING

INNER JOIN ... USING gần như là một ngôi nhà nửa đường giữa thông thường INNER JOIN và NATURAL JOIN. Phép nối được tạo bằng các cột có tên trùng khớp trong mỗi bảng, nhưng bạn phải chỉ định các cột sẽ được sử dụng, không phải toàn bộ điều kiện. Điều này cho phép bạn tham gia vào một tập hợp con của các cột chung cho cả hai bảng.

SELECT e.employee\_name,

d.department\_name

FROM **employees e**

**JOIN departments d USING (department\_id)**

ORDER BY e.employee\_name;

EMPLOYEE\_N DEPARTMENT\_NAM

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

ADAMS RESEARCH

ALLEN SALES

BLAKE SALES

CLARK ACCOUNTING

FORD RESEARCH

JAMES SALES

JONES RESEARCH

KING ACCOUNTING

MARTIN SALES

MILLER ACCOUNTING

SCOTT RESEARCH

SMITH RESEARCH

TURNER SALES

WARD SALES

14 rows selected.

SQL>

Đây là một cú pháp join an toàn vì nó không thể bị ảnh hưởng bởi việc thêm các cột vào một trong hai bảng. Tương tự như NATURAL JOIN, bạn không thể áp dụng bất kỳ bộ lọc bí danh nào cho các cột được sử dụng trong liên kết, nhưng nếu bạn xóa bí danh thì nó hoạt động.

ELECT e.employee\_name,

d.department\_name

FROM employees e

JOIN departments d USING (department\_id)

WHERE  **d.department\_id = 20**

ORDER BY e.employee\_name;

**WHERE d.department\_id = 20**

**\***

**ERROR at line 5:**

**ORA-25154: column part of USING clause cannot have qualifier**

SQL>

SELECT e.employee\_name,

d.department\_name

FROM employees e

JOIN departments d USING (department\_id)

WHERE **department\_id = 20**

ORDER BY e.employee\_name;

EMPLOYEE\_N DEPARTMENT\_NAM

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

ADAMS RESEARCH

FORD RESEARCH

JONES RESEARCH

SCOTT RESEARCH

SMITH RESEARCH

5 rows selected.

SQL>

### SQL Tutorial - Bài 6: Toán tử (Operators)

Trong bài viết này, chúng ta hãy xem các toán tử tập hợp SQL có sẵn trong Oracle.

* Thiết lập môi trường
* Giới thiệu
* UNION
* UNION ALL
* INTERSECT
* MINUS
* ORDER BY
* Data Types

THIẾT LẬP MÔI TRƯỜNG

Bạn có thể thực hiện tất cả các truy vấn trực tuyến miễn phí bằng [SQL Fiddle](http://sqlfiddle.com/#!4) .

Các ví dụ trong bài viết này yêu cầu phải có các bảng sau đây.

--DROP TABLE employees PURGE;

--DROP TABLE departments PURGE;

CREATE TABLE departments (

department\_id NUMBER(2) CONSTRAINT departments\_pk PRIMARY KEY,

department\_name VARCHAR2(14),

location VARCHAR2(13)

);

INSERT INTO departments VALUES (10,'ACCOUNTING','NEW YORK');

INSERT INTO departments VALUES (20,'RESEARCH','DALLAS');

INSERT INTO departments VALUES (30,'SALES','CHICAGO');

INSERT INTO departments VALUES (40,'OPERATIONS','BOSTON');

COMMIT;

CREATE TABLE employees (

employee\_id NUMBER(4) CONSTRAINT employees\_pk PRIMARY KEY,

employee\_name VARCHAR2(10),

job VARCHAR2(9),

manager\_id NUMBER(4),

hiredate DATE,

salary NUMBER(7,2),

commission NUMBER(7,2),

department\_id NUMBER(2) CONSTRAINT emp\_department\_id\_fk REFERENCES departments(department\_id)

);

INSERT INTO employees VALUES (7369,'SMITH','CLERK',7902,to\_date('17-12-1980','dd-mm-yyyy'),800,NULL,20);

INSERT INTO employees VALUES (7499,'ALLEN','SALESMAN',7698,to\_date('20-2-1981','dd-mm-yyyy'),1600,300,30);

INSERT INTO employees VALUES (7521,'WARD','SALESMAN',7698,to\_date('22-2-1981','dd-mm-yyyy'),1250,500,30);

INSERT INTO employees VALUES (7566,'JONES','MANAGER',7839,to\_date('2-4-1981','dd-mm-yyyy'),2975,NULL,20);

INSERT INTO employees VALUES (7654,'MARTIN','SALESMAN',7698,to\_date('28-9-1981','dd-mm-yyyy'),1250,1400,30);

INSERT INTO employees VALUES (7698,'BLAKE','MANAGER',7839,to\_date('1-5-1981','dd-mm-yyyy'),2850,NULL,30);

INSERT INTO employees VALUES (7782,'CLARK','MANAGER',7839,to\_date('9-6-1981','dd-mm-yyyy'),2450,NULL,10);

INSERT INTO employees VALUES (7788,'SCOTT','ANALYST',7566,to\_date('13-JUL-87','dd-mm-rr')-85,3000,NULL,20);

INSERT INTO employees VALUES (7839,'KING','PRESIDENT',NULL,to\_date('17-11-1981','dd-mm-yyyy'),5000,NULL,10);

INSERT INTO employees VALUES (7844,'TURNER','SALESMAN',7698,to\_date('8-9-1981','dd-mm-yyyy'),1500,0,30);

INSERT INTO employees VALUES (7876,'ADAMS','CLERK',7788,to\_date('13-JUL-87', 'dd-mm-rr')-51,1100,NULL,20);

INSERT INTO employees VALUES (7900,'JAMES','CLERK',7698,to\_date('3-12-1981','dd-mm-yyyy'),950,NULL,30);

INSERT INTO employees VALUES (7902,'FORD','ANALYST',7566,to\_date('3-12-1981','dd-mm-yyyy'),3000,NULL,20);

INSERT INTO employees VALUES (7934,'MILLER','CLERK',7782,to\_date('23-1-1982','dd-mm-yyyy'),1300,NULL,10);

COMMIT;

Các bảng này là một biến thể của các bảng EMP và DEPT từ lược đồ SCOTT. Bạn sẽ thấy rất nhiều ví dụ của Oracle trên internet bằng cách sử dụng các bảng từ lược đồ SCOTT. Bạn có thể tìm thấy các định nghĩa bảng gốc trong tập lệnh "$ ORACLE\_HOME/rdbms/admin/utlsampl.sql".

GIỚI THIỆU

Toán tử tập hợp cho phép bạn kết hợp các kết quả của nhiều truy vấn riêng biệt thành một tập kết quả duy nhất.

Hai truy vấn sau đây sẽ được sử dụng cho hầu hết các ví dụ trong bài viết này. Cái đầu tiên trả về các bộ phận 10, 20 và 30. Thứ hai trả về các bộ phận 20, 30 và 40. Như bạn có thể thấy, các bộ phận 20 và 30 là chung cho cả hai bộ kết quả.

-- Department 10, 20 và 30.

SELECT department\_id, department\_name

FROM departments

WHERE department\_id <= 30;

DEPARTMENT\_ID DEPARTMENT\_NAM

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

10 ACCOUNTING

20 RESEARCH

30 SALES

3 rows selected.

SQL>

--Department 20, 30 và 40.

SELECT department\_id, department\_name

FROM departments

WHERE department\_id >= 20;

DEPARTMENT\_ID DEPARTMENT\_NAM

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

20 RESEARCH

30 SALES

40 OPERATIONS

3 rows selected.

SQL>

UNION

Toán tử UNION trả về tất cả các hàng riêng biệt bởi một trong hai câu hỏi được chọn. Điều đó có nghĩa là bất kỳ hàng **trùng lặp sẽ được loại bỏ**.

Trong ví dụ dưới đây, thông báo chỉ có một hàng duy nhất cho mỗi bộ phận 20 và 30, thay vì hai hàng.

SELECT department\_id, department\_name

FROM departments

WHERE department\_id <= 30

UNION

SELECT department\_id, department\_name

FROM departments

WHERE department\_id >= 20

ORDER BY 1;

DEPARTMENT\_ID DEPARTMENT\_NAM

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

10 ACCOUNTING

20 RESEARCH

30 SALES

40 OPERATIONS

4 rows selected.

SQL>

Việc loại bỏ các bản sao đòi hỏi phải xử lý thêm, vì vậy bạn nên cân nhắc sử dụng UNION ALL nếu có thể.

UNION ALL

Toán tử UNION ALL trả về tất cả các hàng bởi một trong hai câu hỏi được chọn. Điều đó có nghĩa là bất kỳ bản sao nào sẽ vẫn còn trong tập kết quả cuối cùng.

Trong ví dụ dưới đây, chú ý có hai hàng cho mỗi bộ phận 20 và 30.

SELECT department\_id, department\_name

FROM departments

WHERE department\_id <= 30

**UNION ALL**

SELECT department\_id, department\_name

FROM departments

WHERE department\_id >= 20

ORDER BY 1;

DEPARTMENT\_ID DEPARTMENT\_NAM

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

10 ACCOUNTING

**20 RESEARCH**

**20 RESEARCH**

**30 SALES**

**30 SALES**

40 OPERATIONS

6 rows selected.

SQL>

INTERSECT

INTERSECT trả về tất cả các hàng riêng biệt bởi cả hai truy vấn lựa chọn. Điều đó có nghĩa là chỉ những hàng chung cho cả hai truy vấn sẽ có mặt trong tập kết quả cuối cùng.

Trong ví dụ dưới đây, chú ý có một hàng cho các bộ phận 20 và 30, vì cả hai hàng này đều xuất hiện trong tập kết quả cho các truy vấn tương ứng của chúng.

SELECT department\_id, department\_name

FROM departments

WHERE department\_id <= 30

**INTERSECT**

SELECT department\_id, department\_name

FROM departments

WHERE department\_id >= 20

ORDER BY 1;

DEPARTMENT\_ID DEPARTMENT\_NAM

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

20 RESEARCH

30 SALES

2 rows selected.

SQL>

MINUS

MINUS trả về tất cả các hàng riêng biệt bằng cách truy vấn đầu tiên nhưng không phải là thứ hai được chọn. Đây là chức năng tương đương với toán tử ANSI set EXCEPT DISTINCT.

Trong ví dụ dưới đây, truy vấn đầu tiên sẽ trả về các bộ phận 10, 20, 30, nhưng các bộ phận 20 và 30 bị xóa vì chúng được trả về bởi truy vấn thứ hai. Điều này để lại một hàng duy nhất cho bộ phận 10.

SELECT department\_id, department\_name

FROM departments

WHERE department\_id <= 30

**MINUS**

SELECT department\_id, department\_name

FROM departments

WHERE department\_id >= 20

ORDER BY 1;

DEPARTMENT\_ID DEPARTMENT\_NAM

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

10 ACCOUNTING

1 row selected.

SQL>

ORDER BY

 ORDER BY được áp dụng cho tất cả các hàng trả lại trong tập kết quả cuối cùng. Các cột trong mệnh đề ORDER BY có thể được tham chiếu bằng tên cột hoặc bí danh cột có trong truy vấn đầu tiên của câu lệnh, vì chúng được chuyển đến tập kết quả cuối cùng. Thông thường, bạn sẽ thấy mọi người sử dụng vị trí cột vì nó ít gây nhầm lẫn khi dữ liệu được lấy từ các vị trí khác nhau cho mỗi khối truy vấn.

-- Tên cột

SELECT **employee\_id**, employee\_name

FROM employees

WHERE department\_id = 10

UNION ALL

SELECT department\_id, department\_name

FROM departments

WHERE department\_id >= 20

ORDER BY **employee\_id**;

EMPLOYEE\_ID EMPLOYEE\_NAME

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

20 RESEARCH

30 SALES

40 OPERATIONS

7782 CLARK

7839 KING

7934 MILLER

6 rows selected.

SQL>

-- Bí danh cột

SELECT employee\_id **AS emp\_id**, employee\_name

FROM employees

WHERE department\_id = 10

UNION ALL

SELECT department\_id, department\_name

FROM departments

WHERE department\_id >= 20

ORDER BY **emp\_id**;

EMP\_ID EMPLOYEE\_NAME

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

20 RESEARCH

30 SALES

40 OPERATIONS

7782 CLARK

7839 KING

7934 MILLER

6 rows selected.

SQL>

-- Vị trí cột

SELECT employee\_id, employee\_name

FROM employees

WHERE department\_id = 10

UNION ALL

SELECT department\_id, department\_name

FROM departments

WHERE department\_id >= 20

ORDER BY **1**;

EMPLOYEE\_ID EMPLOYEE\_NAME

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

20 RESEARCH

30 SALES

40 OPERATIONS

7782 CLARK

7839 KING

7934 MILLER

6 rows selected.

SQL>

Tham chiếu tên cột hoặc bí danh cột trong các truy vấn sau này không được phép.

-- Tên cột sai.

SELECT employee\_id, employee\_name

FROM employees

WHERE department\_id = 10

UNION ALL

SELECT **department\_id**, department\_name

FROM departments

WHERE department\_id >= 20

ORDER BY department\_id;

**ORDER BY department\_id**

**\***

**ERROR at line 8:**

**ORA-00904: "DEPARTMENT\_ID": invalid identifier**

SQL>

-- Wrong column alias

SELECT employee\_id, employee\_name

FROM employees

WHERE department\_id = 10

UNION ALL

SELECT department\_id **AS dept\_id**, department\_name

FROM departments

WHERE department\_id >= 20

ORDER BY dept\_id;

**ORDER BY dept\_id**

**\***

**ERROR at line 8:**

**ORA-00904: "DEPT\_ID": invalid identifier**

SQL>

LOẠI DỮ LIỆU

Các kiểu dữ liệu phải khớp với cùng một vị trí cột trong mỗi truy vấn. Một số chuyển đổi ngầm sẽ có thể giữa một số loại dữ liệu, ví dụ CHAR/ VARCHAR2 hoặc chuyển đổi giữa một số loại số.

Trong ví dụ sau, chúng tôi chuyển đổi cột EMPLOYEE\_ID sang VARCHAR2 sử dụng hàm TO\_CHAR, ngắt câu lệnh.

SELECT **TO\_CHAR(employee\_id)** AS employee\_id, employee\_name

FROM employees

WHERE department\_id = 10

UNION ALL

SELECT department\_id, department\_name

FROM departments

WHERE department\_id >= 20

ORDER BY 1;

**SELECT TO\_CHAR(employee\_id) AS employee\_id, employee\_name**

**\***

**ERROR at line 1:**

**ORA-01790: expression must have same datatype as corresponding expression**

SQL>

### SQL Tutorial - Bài 7: GROUP BY và HAVING

Trong bài viết này, chúng ta hãy xem mệnh đề **GROUP BY** và **HAVING** trong SQL.

* Thiết lập môi trường
* GROUP BY
* Joins
* HAVING

THIẾT LẬP MÔI TRƯỜNG

Bạn có thể thực hiện tất cả các truy vấn trực tuyến miễn phí bằng [SQL Fiddle](http://sqlfiddle.com/#!4) .

Các ví dụ trong bài viết này yêu cầu phải có các bảng sau đây.

--DROP TABLE employees PURGE;

--DROP TABLE departments PURGE;

CREATE TABLE departments (

department\_id NUMBER(2) CONSTRAINT departments\_pk PRIMARY KEY,

department\_name VARCHAR2(14),

location VARCHAR2(13)

);

INSERT INTO departments VALUES (10,'ACCOUNTING','NEW YORK');

INSERT INTO departments VALUES (20,'RESEARCH','DALLAS');

INSERT INTO departments VALUES (30,'SALES','CHICAGO');

INSERT INTO departments VALUES (40,'OPERATIONS','BOSTON');

COMMIT;

CREATE TABLE employees (

employee\_id NUMBER(4) CONSTRAINT employees\_pk PRIMARY KEY,

employee\_name VARCHAR2(10),

job VARCHAR2(9),

manager\_id NUMBER(4),

hiredate DATE,

salary NUMBER(7,2),

commission NUMBER(7,2),

department\_id NUMBER(2) CONSTRAINT emp\_department\_id\_fk REFERENCES departments(department\_id)

);

INSERT INTO employees VALUES (7369,'SMITH','CLERK',7902,to\_date('17-12-1980','dd-mm-yyyy'),800,NULL,20);

INSERT INTO employees VALUES (7499,'ALLEN','SALESMAN',7698,to\_date('20-2-1981','dd-mm-yyyy'),1600,300,30);

INSERT INTO employees VALUES (7521,'WARD','SALESMAN',7698,to\_date('22-2-1981','dd-mm-yyyy'),1250,500,30);

INSERT INTO employees VALUES (7566,'JONES','MANAGER',7839,to\_date('2-4-1981','dd-mm-yyyy'),2975,NULL,20);

INSERT INTO employees VALUES (7654,'MARTIN','SALESMAN',7698,to\_date('28-9-1981','dd-mm-yyyy'),1250,1400,30);

INSERT INTO employees VALUES (7698,'BLAKE','MANAGER',7839,to\_date('1-5-1981','dd-mm-yyyy'),2850,NULL,30);

INSERT INTO employees VALUES (7782,'CLARK','MANAGER',7839,to\_date('9-6-1981','dd-mm-yyyy'),2450,NULL,10);

INSERT INTO employees VALUES (7788,'SCOTT','ANALYST',7566,to\_date('13-JUL-87','dd-mm-rr')-85,3000,NULL,20);

INSERT INTO employees VALUES (7839,'KING','PRESIDENT',NULL,to\_date('17-11-1981','dd-mm-yyyy'),5000,NULL,10);

INSERT INTO employees VALUES (7844,'TURNER','SALESMAN',7698,to\_date('8-9-1981','dd-mm-yyyy'),1500,0,30);

INSERT INTO employees VALUES (7876,'ADAMS','CLERK',7788,to\_date('13-JUL-87', 'dd-mm-rr')-51,1100,NULL,20);

INSERT INTO employees VALUES (7900,'JAMES','CLERK',7698,to\_date('3-12-1981','dd-mm-yyyy'),950,NULL,30);

INSERT INTO employees VALUES (7902,'FORD','ANALYST',7566,to\_date('3-12-1981','dd-mm-yyyy'),3000,NULL,20);

INSERT INTO employees VALUES (7934,'MILLER','CLERK',7782,to\_date('23-1-1982','dd-mm-yyyy'),1300,NULL,10);

COMMIT;

Các bảng này là một biến thể của các bảng EMP và DEPT từ lược đồ SCOTT. Bạn sẽ thấy rất nhiều ví dụ của Oracle trên internet bằng cách sử dụng các bảng từ lược đồ SCOTT. Bạn có thể tìm thấy các định nghĩa bảng gốc trong tập lệnh "$ ORACLE\_HOME / rdbms / admin / utlsampl.sql".

MỆNH ĐỀ GROUP BY

SQL bao gồm nhiều Hàm tổng hợp (Aggregate functions) , lấy thông tin từ nhiều hàng và tổng hợp lại để tạo ra ít hàng hơn trong tập kết quả cuối cùng. Các mã sau đây có chứa một ví dụ đơn giản rằng việc sử dụng COUNT, AVGvà SUM tổng hợp. Chúng tôi có 14 hàng trong EMPLOYEES, nhưng truy vấn tạo ra một hàng duy nhất với số lượng các hàng trong bảng, cũng như trung bình và tổng số cột SALARY cho tất cả các hàng trong bảng.

SELECT **COUNT(\*)** AS employee\_count,

**AVG(e.salary)** AS avg\_salary,

**SUM(e.salary)** AS sum\_salary

FROM employees e;

EMPLOYEE\_COUNT AVG\_SALARY SUM\_SALARY

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

14 2073.21429 29025

1 row selected.

SQL>

Không có mệnh đề GROUP BY, toàn bộ tập kết quả được coi là một nhóm duy nhất, do đó, các hàm tổng hợp hoạt động trên toàn bộ tập kết quả. Việc thêm GROUP BYsẽ phân chia kết quả thành các nhóm hàng, với các hàm tổng hợp được áp dụng trên cơ sở nhóm. Ví dụ sau đây nhóm các hàng theo DEPARTMENT\_ID, do đó các tổng hợp trên cơ sở từng bộ phận.

SELECT e.department\_id,

COUNT(\*) AS employee\_count,

AVG(e.salary) AS avg\_salary,

SUM(e.salary) AS sum\_salary

FROM employees e

**GROUP BY e.department\_id**

ORDER BY e.department\_id;

DEPARTMENT\_ID EMPLOYEE\_COUNT AVG\_SALARY SUM\_SALARY

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

10 3 2916.66667 8750

20 5 2175 10875

30 6 1566.66667 9400

3 rows selected.

SQL>

Càng nhiều cột trong GROUP BYmệnh đề, các tổng hợp càng chi tiết. Các nhóm ví dụ sau theo các cột DEPARTMENT\_ID và JOB.

SELECT e.department\_id,

e.job,

COUNT(\*) AS employee\_count,

AVG(e.salary) AS avg\_salary,

SUM(e.salary) AS sum\_salary

FROM employees e

**GROUP BY e.department\_id, e.job**

ORDER BY e.department\_id, e.job;

DEPARTMENT\_ID JOB EMPLOYEE\_COUNT AVG\_SALARY SUM\_SALARY

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

10 CLERK 1 1300 1300

10 MANAGER 1 2450 2450

10 PRESIDENT 1 5000 5000

20 ANALYST 2 3000 6000

20 CLERK 2 950 1900

20 MANAGER 1 2975 2975

30 CLERK 1 950 950

30 MANAGER 1 2850 2850

30 SALESMAN 4 1400 5600

9 rows selected.

SQL>

Hãy nhớ rằng, tất cả các cột không tổng hợp phải được bao gồm trong GROUP BY.

JOINS

Một truy vấn được nhóm có thể tham gia vào các bảng khác để cung cấp thêm thông tin. Trong ví dụ sau, chúng tôi muốn báo cáo một số dữ liệu tổng hợp dựa trên bộ phận, nhưng chúng tôi muốn hiển thị DEPARTMENT\_NAME, thay vì DEPARTMENT\_ID, vì vậy chúng tôi phải join vào các bảng DEPARTMENTS .

SELECT d.department\_name,

COUNT(\*) AS employee\_count,

AVG(e.salary) AS avg\_salary,

SUM(e.salary) AS sum\_salary

FROM **departments d**

**JOIN employees e ON d.department\_id = e.department\_id**

GROUP BY d.department\_name

ORDER BY d.department\_name;

DEPARTMENT\_NAM EMPLOYEE\_COUNT AVG\_SALARY SUM\_SALARY

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

**ACCOUNTING 3 2916.66667 8750**

**RESEARCH 5 2175 10875**

**SALES 6 1566.66667 9400**

3 rows selected.

SQL>

Điều này hoạt động tốt, nhưng chúng ta đang mất một phần thông tin quan trọng. Bộ phận OPERATIONS  không có nhân viên, nhưng chúng tôi không hiển thị nó vì chúng tôi đã sử dụng một INNER JOIN. Chuyển sang một LEFT OUTER JOIN xuất hiện để khắc phục vấn đề đó, nhưng nó báo cáo số lượng nhân viên sai.

SELECT d.department\_name,

COUNT(\*) AS employee\_count,

AVG(e.salary) AS avg\_salary,

SUM(e.salary) AS sum\_salary

FROM departments d

LEFT OUTER JOIN employees e ON d.department\_id = e.department\_id

GROUP BY d.department\_name

ORDER BY d.department\_name;

DEPARTMENT\_NAM EMPLOYEE\_COUNT AVG\_SALARY SUM\_SALARY

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

ACCOUNTING 3 2916.66667 8750

**OPERATIONS 1**

RESEARCH 5 2175 10875

SALES 6 1566.66667 9400

4 rows selected.

SQL>

Bây giờ chúng tôi có một hàng cho bộ phận OPERATIONS , COUNT(\*)cuộc gọi đã trả về đúng giá trị 1, nhưng đây không phải là số lượng nhân viên chính xác. Đây không phải là lỗi của GROUP BY, nhưng cách thức COUNT(\*)hoạt động gọi. Thay vào đó, nếu chúng ta tham chiếu một cột bắt buộc từ bảng join ngoài trong COUNT gọi, chúng ta sẽ nhận được kết quả chính xác, vì các giá trị NULL không được tính. Ví dụ sau đây cho thấy làm thế nào.

SELECT d.department\_name,

**COUNT(e.employee\_id)** AS employee\_count,

AVG(e.salary) AS avg\_salary,

SUM(e.salary) AS sum\_salary

FROM departments d

LEFT OUTER JOIN employees e ON d.department\_id = e.department\_id

GROUP BY d.department\_name

ORDER BY d.department\_name;

DEPARTMENT\_NAM EMPLOYEE\_COUNT AVG\_SALARY SUM\_SALARY

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

ACCOUNTING 3 2916.66667 8750

**OPERATIONS 0**

RESEARCH 5 2175 10875

SALES 6 1566.66667 9400

4 rows selected.

SQL>

MỆNH ĐỀ HAVING

Chúng tôi đã đề cập trước đây, có nhiều cột được tham chiếu trong mệnh đề GROUP BY cho chúng tôi nhiều thông tin chi tiết hơn. Ví dụ sau đây nhóm dữ liệu theo DEPARTMENT\_NAME và JOB.

SELECT d.department\_name, e.job,

COUNT(e.employee\_id) AS employee\_count,

AVG(e.salary) AS avg\_salary,

SUM(e.salary) AS sum\_salary

FROM departments d

LEFT OUTER JOIN employees e ON d.department\_id = e.department\_id

GROUP BY d.department\_name, e.job

ORDER BY d.department\_name, e.job;

DEPARTMENT\_NAM JOB EMPLOYEE\_COUNT AVG\_SALARY SUM\_SALARY

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

ACCOUNTING CLERK 1 1300 1300

ACCOUNTING MANAGER 1 2450 2450

ACCOUNTING PRESIDENT 1 5000 5000

OPERATIONS 0

RESEARCH ANALYST 2 3000 6000

RESEARCH CLERK 2 950 1900

RESEARCH MANAGER 1 2975 2975

SALES CLERK 1 950 950

SALES MANAGER 1 2850 2850

SALES SALESMAN 4 1400 5600

10 rows selected.

SQL>

Đôi khi chúng tôi chỉ quan tâm đến thông tin từ các nhóm dữ liệu cụ thể. HAVING cho phép chúng tôi lọc ra các nhóm không đáp ứng yêu cầu cụ thể. Bạn có thể nghĩ về nó như một mệnh đề  WHERE cho mệnh đề GROUP BY. Trong ví dụ sau, chúng tôi chỉ trả lại thông tin cho các nhóm có nhiều hơn 1 nhân viên.

SELECT d.department\_name, e.job,

COUNT(e.employee\_id) AS employee\_count,

AVG(e.salary) AS avg\_salary,

SUM(e.salary) AS sum\_salary

FROM departments d

LEFT OUTER JOIN employees e ON d.department\_id = e.department\_id

GROUP BY d.department\_name, e.job

**HAVING COUNT(e.employee\_id) > 1**

ORDER BY d.department\_name, e.job;

DEPARTMENT\_NAM JOB EMPLOYEE\_COUNT AVG\_SALARY SUM\_SALARY

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

RESEARCH ANALYST 2 3000 6000

RESEARCH CLERK 2 950 1900

SALES SALESMAN 4 1400 5600

3 rows selected.

SQL>

Chỉ là một sự ngớ ngẩn sang một bên, có thể có một mệnh đề HAVING mà không có một mệnh đề GROUP BY, như trong ví dụ sau đây. Toàn bộ tập kết quả được coi là một nhóm duy nhất, vì vậy bộ lọc  mệnh đề HAVING vẫn có thể được áp dụng.

SELECT COUNT(e.employee\_id) AS employee\_count,

AVG(e.salary) AS avg\_salary,

SUM(e.salary) AS sum\_salary

FROM departments d

LEFT OUTER JOIN employees e ON d.department\_id = e.department\_id

HAVING COUNT(e.employee\_id) > 2

ORDER BY d.department\_name, e.job;

EMPLOYEE\_COUNT AVG\_SALARY SUM\_SALARY

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

14 2073.21429 29025

1 row selected.

SQL>

### SQL Tutorial - Bài 8: Câu lệnh INSERT

Trong bài viết này, chúng tôi sẽ xem xét một số biến thể của câu lệnh **INSERT**.

* Thiết lập môi trường
* COMMIT và ROLLBACK
* INSERT cơ bản
* INSERT vào View
* INSERT ... SELECT
* Subqueries

THIẾT LẬP MÔI TRƯỜNG

Bạn có thể thực hiện tất cả các truy vấn trực tuyến miễn phí bằng [SQL Fiddle](http://sqlfiddle.com/#!4) .

Các ví dụ trong bài viết này yêu cầu phải có các bảng sau đây.

--DROP TABLE employees PURGE;

--DROP TABLE departments PURGE;

CREATE TABLE departments (

department\_id NUMBER(2) CONSTRAINT departments\_pk PRIMARY KEY,

department\_name VARCHAR2(14),

location VARCHAR2(13)

);

INSERT INTO departments VALUES (10,'ACCOUNTING','NEW YORK');

INSERT INTO departments VALUES (20,'RESEARCH','DALLAS');

INSERT INTO departments VALUES (30,'SALES','CHICAGO');

INSERT INTO departments VALUES (40,'OPERATIONS','BOSTON');

COMMIT;

CREATE TABLE employees (

employee\_id NUMBER(4) CONSTRAINT employees\_pk PRIMARY KEY,

employee\_name VARCHAR2(10),

job VARCHAR2(9),

manager\_id NUMBER(4),

hiredate DATE,

salary NUMBER(7,2),

commission NUMBER(7,2),

department\_id NUMBER(2) CONSTRAINT emp\_department\_id\_fk REFERENCES departments(department\_id)

);

INSERT INTO employees VALUES (7369,'SMITH','CLERK',7902,to\_date('17-12-1980','dd-mm-yyyy'),800,NULL,20);

INSERT INTO employees VALUES (7499,'ALLEN','SALESMAN',7698,to\_date('20-2-1981','dd-mm-yyyy'),1600,300,30);

INSERT INTO employees VALUES (7521,'WARD','SALESMAN',7698,to\_date('22-2-1981','dd-mm-yyyy'),1250,500,30);

INSERT INTO employees VALUES (7566,'JONES','MANAGER',7839,to\_date('2-4-1981','dd-mm-yyyy'),2975,NULL,20);

INSERT INTO employees VALUES (7654,'MARTIN','SALESMAN',7698,to\_date('28-9-1981','dd-mm-yyyy'),1250,1400,30);

INSERT INTO employees VALUES (7698,'BLAKE','MANAGER',7839,to\_date('1-5-1981','dd-mm-yyyy'),2850,NULL,30);

INSERT INTO employees VALUES (7782,'CLARK','MANAGER',7839,to\_date('9-6-1981','dd-mm-yyyy'),2450,NULL,10);

INSERT INTO employees VALUES (7788,'SCOTT','ANALYST',7566,to\_date('13-JUL-87','dd-mm-rr')-85,3000,NULL,20);

INSERT INTO employees VALUES (7839,'KING','PRESIDENT',NULL,to\_date('17-11-1981','dd-mm-yyyy'),5000,NULL,10);

INSERT INTO employees VALUES (7844,'TURNER','SALESMAN',7698,to\_date('8-9-1981','dd-mm-yyyy'),1500,0,30);

INSERT INTO employees VALUES (7876,'ADAMS','CLERK',7788,to\_date('13-JUL-87', 'dd-mm-rr')-51,1100,NULL,20);

INSERT INTO employees VALUES (7900,'JAMES','CLERK',7698,to\_date('3-12-1981','dd-mm-yyyy'),950,NULL,30);

INSERT INTO employees VALUES (7902,'FORD','ANALYST',7566,to\_date('3-12-1981','dd-mm-yyyy'),3000,NULL,20);

INSERT INTO employees VALUES (7934,'MILLER','CLERK',7782,to\_date('23-1-1982','dd-mm-yyyy'),1300,NULL,10);

COMMIT;

Các bảng này là một biến thể của các bảng EMP và DEPT từ lược đồ SCOTT. Bạn sẽ thấy rất nhiều ví dụ của Oracle trên internet bằng cách sử dụng các bảng từ lược đồ SCOTT. Bạn có thể tìm thấy các định nghĩa bảng gốc trong tập lệnh "$ORACLE\_HOME/rdbms/admin/utlsampl.sq".

Trong bài viết này, chúng tôi cũng sẽ cần bản sao sau của bảng employees  . Lưu ý rằng chúng tôi đã bắt buộc cột EMPLOYEE\_NAME và JOB. Cột EMPLOYEE\_ID hoàn toàn bắt buộc, do đó nó là khóa chính.

-- DROP TABLE employees2 PURGE;

CREATE TABLE employees2 (

employee\_id NUMBER(4) CONSTRAINT employees2\_pk PRIMARY KEY,

employee\_name VARCHAR2(10) NOT NULL,

job VARCHAR2(9) NOT NULL,

manager\_id NUMBER(4),

hiredate DATE,

salary NUMBER(7,2),

commission NUMBER(7,2),

department\_id NUMBER(2)

);

COMMIT VÀ ROLLBACK

Tất cả các thay đổi của ngôn ngữ dữ liệu (DML - Data Manipulation Language) được thực hiện như một phần của giao dịch. Khi thực hiện câu lệnh COMMIT thì dữ liệu được sẵn sàng để lưu vào datafile. Để lấy lại dữ liệu cũ đã COMMIT thì cần flashback lại.

Các lệnh Ngôn ngữ định nghĩa dữ liệu (DDL) thực hiện một commit ngầm định, cũng xác nhận tất cả các thay đổi DML trong phiên hiện tại.

Nếu bạn quyết định không muốn giữ một số thay đổi chưa commit, bạn có thể loại bỏ chúng bằng cách sử dụng câu lệnh ROLLBACK. Nhiều ví dụ trong bài viết này sẽ đưa ra các câu lệnh ROLLBACK sau khi thử nghiệm, để phục hồi dữ liệu về trạng thái ban đầu.

Một số công cụ và ngôn ngữ lập trình mặc định là tự động commit, vì vậy chúng tự động đưa ra câu lệnh COMMIT sau mỗi câu lệnh DML mà chúng xử lý. Đừng để điều này đánh lừa bạn nghĩ rằng đây là hành vi mặc định. Không phải vậy.

INSERT CƠ BẢN

**INSERT INTO employees2 VALUES (8888,'JONES','CLERK',7902,to\_date('17-12-1980','dd-mm-yyyy'),800,NULL,20);**

SELECT \*

FROM employees2

ORDER BY employee\_id;

EMPLOYEE\_ID EMPLOYEE\_N JOB MANAGER\_ID HIREDATE SALARY COMMISSION DEPARTMENT\_ID

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

8888 JONES CLERK 7902 17-DEC-1980 00:00:00 800 20

1 row selected.

SQL>

ROLLBACK;

Chỉ định danh sách cột an toàn hơn nhiều vì câu lệnh sẽ không bị hỏng nếu các cột tùy chọn được thêm vào bảng.

INSERT INTO employees2 (**employee\_id, employee\_name, job, manager\_id, hiredate, salary, commission, department\_id**)

VALUES (8888,'JONES','CLERK',7902,to\_date('17-12-1980','dd-mm-yyyy'),800,NULL,20);

SELECT \*

FROM employees2

ORDER BY employee\_id;

EMPLOYEE\_ID EMPLOYEE\_N JOB MANAGER\_ID HIREDATE SALARY COMMISSION DEPARTMENT\_ID

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

8888 JONES CLERK 7902 17-DEC-1980 00:00:00 800 20

1 row selected.

SQL>

ROLLBACK;

Danh sách cột rõ ràng cũng cho phép bạn xóa các cột tùy chọn khỏi câu lệnh và hoán đổi vị trí của các cột trong câu lệnh. Trong ví dụ sau, các cột ManageER\_ID, COMMISSION và DEPARTMENT\_ID đã bị xóa và cột EMPLOYEE\_ID đã được chuyển đến cuối danh sách cột và danh sách giá trị.

INSERT INTO employees2 (**employee\_name, job, hiredate, salary, employee\_id**)

VALUES ('JONES','CLERK',to\_date('17-12-1980','dd-mm-yyyy'),800,8888);

SELECT \*

FROM employees2

ORDER BY employee\_id;

EMPLOYEE\_ID EMPLOYEE\_N JOB MANAGER\_ID HIREDATE SALARY COMMISSION DEPARTMENT\_ID

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

8888 JONES CLERK 7902 17-DEC-1980 00:00:00 800 20

1 row selected.

SQL>

ROLLBACK;

Hãy nhớ rằng, bỏ lỡ các cột bắt buộc là không được phép, trừ khi bạn có các giá trị hoặc kích hoạt mặc định của cột tạo ra chúng. Trong ví dụ sau, cột JOB bị bỏ qua, đây là cột bắt buộc trong bảng EMPLOYEES2.

INSERT INTO employees2 (employee\_id, employee\_name, manager\_id, hiredate, salary, commission, department\_id)

VALUES (8888,'JONES',7902,to\_date('17-12-1980','dd-mm-yyyy'),800,NULL,20);

**INSERT INTO employees2 (employee\_id, employee\_name, manager\_id, hiredate, salary, commission, department\_id)**

**\***

**ERROR at line 1:**

**ORA-01400: cannot insert NULL into ("TEST"."EMPLOYEES2"."JOB")**

SQL>

INSERT INTO VIEWS

Có thể chèn vào bảng cơ sở được liên kết với chế độ xem. Có một số hạn chế liên quan đến điều này, nhưng chúng hơi ngoài phạm vi cho một bài viết cấp độ mới bắt đầu. Trong ví dụ sau, chúng ta tạo một dạng xem đơn giản trên bảng EMPLOYEES2, sau đó chèn vào đó.

CREATE OR REPLACE VIEW employees2\_v AS

SELECT \* FROM employees2;

INSERT INTO **employees2\_v** VALUES (8888,'JONES','CLERK',7902,to\_date('17-12-1980','dd-mm-yyyy'),800,NULL,20);

SELECT \*

FROM employees2

ORDER BY employee\_id;

EMPLOYEE\_ID EMPLOYEE\_N JOB MANAGER\_ID HIREDATE SALARY COMMISSION DEPARTMENT\_ID

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

8888 JONES CLERK 7902 17-DEC-1980 00:00:00 800 20

1 row selected.

SQL>

ROLLBACK;

Bạn sẽ không thấy nó rất thường xuyên, nhưng bạn cũng có thể chèn vào inline view. Danh sách cột rõ ràng.

INSERT INTO (**SELECT employee\_id, employee\_name, job, hiredate, salary**

**FROM employees2**)

VALUES (8888,'JONES','CLERK',to\_date('17-12-1980','dd-mm-yyyy'),800);

SELECT \*

FROM employees2

ORDER BY employee\_id;

EMPLOYEE\_ID EMPLOYEE\_N JOB MANAGER\_ID HIREDATE SALARY COMMISSION DEPARTMENT\_ID

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

8888 JONES CLERK 7902 17-DEC-1980 00:00:00 800 20

1 row selected.

SQL>

ROLLBACK;

INSERT ... SELECT

Có thể tạo nhiều hàng trong một câu lệnh INSERT bằng cách sử dụng cú pháp INSERT ... SELECT. Ví dụ sau sao chép tất cả dữ liệu từ bảng NHÂN VIÊN vào bảng EMPLOYEES2.

-- Xóa trắng bảng này

TRUNCATE TABLE employees2;

**INSERT INTO employees2**

**SELECT \* FROM employees;**

COMMIT;

SELECT \*

FROM employees2

ORDER BY employee\_id;

EMPLOYEE\_ID EMPLOYEE\_N JOB MANAGER\_ID HIREDATE SALARY COMMISSION DEPARTMENT\_ID

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

7369 SMITH CLERK 7902 17-DEC-1980 00:00:00 800 20

7499 ALLEN SALESMAN 7698 20-FEB-1981 00:00:00 1600 300 30

7521 WARD SALESMAN 7698 22-FEB-1981 00:00:00 1250 500 30

7566 JONES MANAGER 7839 02-APR-1981 00:00:00 2975 20

7654 MARTIN SALESMAN 7698 28-SEP-1981 00:00:00 1250 1400 30

7698 BLAKE MANAGER 7839 01-MAY-1981 00:00:00 2850 30

7782 CLARK MANAGER 7839 09-JUN-1981 00:00:00 2450 10

7788 SCOTT ANALYST 7566 19-APR-1987 00:00:00 3000 20

7839 KING PRESIDENT 17-NOV-1981 00:00:00 5000 10

7844 TURNER SALESMAN 7698 08-SEP-1981 00:00:00 1500 0 30

7876 ADAMS CLERK 7788 23-MAY-1987 00:00:00 1100 20

7900 JAMES CLERK 7698 03-DEC-1981 00:00:00 950 30

7902 FORD ANALYST 7566 03-DEC-1981 00:00:00 3000 20

7934 MILLER CLERK 7782 23-JAN-1982 00:00:00 1300 10

14 rows selected.

SQL>

Danh sách cột trong câu lệnh INSERT câu lệnh phải khớp với SELECT trong truy vấn.

TRUNCATE TABLE employees2;

INSERT INTO employees2 (**employee\_id, employee\_name, job, hiredate, salary)**

**SELECT employee\_id, employee\_name, job, hiredate, salary**

FROM employees;

COMMIT;

SELECT \*

FROM employees2

ORDER BY employee\_id;

EMPLOYEE\_ID EMPLOYEE\_N JOB MANAGER\_ID HIREDATE SALARY COMMISSION DEPARTMENT\_ID

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

7369 SMITH CLERK 17-DEC-1980 00:00:00 800

7499 ALLEN SALESMAN 20-FEB-1981 00:00:00 1600

7521 WARD SALESMAN 22-FEB-1981 00:00:00 1250

7566 JONES MANAGER 02-APR-1981 00:00:00 2975

7654 MARTIN SALESMAN 28-SEP-1981 00:00:00 1250

7698 BLAKE MANAGER 01-MAY-1981 00:00:00 2850

7782 CLARK MANAGER 09-JUN-1981 00:00:00 2450

7788 SCOTT ANALYST 19-APR-1987 00:00:00 3000

7839 KING PRESIDENT 17-NOV-1981 00:00:00 5000

7844 TURNER SALESMAN 08-SEP-1981 00:00:00 1500

7876 ADAMS CLERK 23-MAY-1987 00:00:00 1100

7900 JAMES CLERK 03-DEC-1981 00:00:00 950

7902 FORD ANALYST 03-DEC-1981 00:00:00 3000

7934 MILLER CLERK 23-JAN-1982 00:00:00 1300

14 rows selected.

SQL>

Các quy tắc thông thường áp dụng cho phần truy vấn của câu lệnh, vì vậy bạn có thể lọc dữ liệu hoặc join với các bảng khác để thu thập dữ liệu.

TRUNCATE TABLE employees2;

INSERT INTO employees2

SELECT \*

FROM employees

**WHERE department\_id = 20;**

COMMIT;

SELECT \*

FROM employees2

ORDER BY employee\_id;

EMPLOYEE\_ID EMPLOYEE\_N JOB MANAGER\_ID HIREDATE SALARY COMMISSION DEPARTMENT\_ID

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

7369 SMITH CLERK 7902 17-DEC-1980 00:00:00 800 20

7566 JONES MANAGER 7839 02-APR-1981 00:00:00 2975 20

7788 SCOTT ANALYST 7566 19-APR-1987 00:00:00 3000 20

7876 ADAMS CLERK 7788 23-MAY-1987 00:00:00 1100 20

7902 FORD ANALYST 7566 03-DEC-1981 00:00:00 3000 20

5 rows selected.

SQL>

TRUY VẤN CON VÔ HƯỚNG (SCALAR SUBQUERIES)

Các truy vấn con vô hướng có thể được sử dụng để tạo giá trị cho câu lệnh INSERT. Trong ví dụ dưới đây, cột SALARY được điền bởi một truy vấn vô hướng vô hướng, trả về SALARY tối đa từ bảng EMPLOYEES  và thêm 1000 vào nó.

TRUNCATE TABLE employees2;

INSERT INTO employees2 (employee\_id, employee\_name, job, hiredate, salary)

VALUES (8888, 'JONES','CLERK',to\_date('17-12-1980','dd-mm-yyyy'),(**SELECT MAX(salary)+1000 FROM employees)**);

SELECT \*

FROM employees2

ORDER BY employee\_id;

EMPLOYEE\_ID EMPLOYEE\_N JOB MANAGER\_ID HIREDATE SALARY COMMISSION DEPARTMENT\_ID

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

8888 JONES CLERK 17-DEC-1980 00:00:00 6000

1 row selected.

SQL>

### SQL Tutorial - Bài 9: Câu lệnh UPDATE

Trong bài viết này, chúng tôi sẽ xem xét một số biến thể của  câu lệnh **UPDATE**

* Thiết lập môi trường
* COMMIT và ROLLBACK
* UPDATE cơ bản
* UPDATE qua View
* Scalar Subqueries
* 0 Rows Updated

THIẾT LẬP

Bạn có thể thực hiện tất cả các truy vấn trực tuyến miễn phí bằng [SQL Fiddle](http://sqlfiddle.com/#!4) .

Các ví dụ trong bài viết này yêu cầu phải có các bảng sau đây.

--DROP TABLE employees PURGE;

--DROP TABLE departments PURGE;

CREATE TABLE departments (

department\_id NUMBER(2) CONSTRAINT departments\_pk PRIMARY KEY,

department\_name VARCHAR2(14),

location VARCHAR2(13)

);

INSERT INTO departments VALUES (10,'ACCOUNTING','NEW YORK');

INSERT INTO departments VALUES (20,'RESEARCH','DALLAS');

INSERT INTO departments VALUES (30,'SALES','CHICAGO');

INSERT INTO departments VALUES (40,'OPERATIONS','BOSTON');

COMMIT;

CREATE TABLE employees (

employee\_id NUMBER(4) CONSTRAINT employees\_pk PRIMARY KEY,

employee\_name VARCHAR2(10),

job VARCHAR2(9),

manager\_id NUMBER(4),

hiredate DATE,

salary NUMBER(7,2),

commission NUMBER(7,2),

department\_id NUMBER(2) CONSTRAINT emp\_department\_id\_fk REFERENCES departments(department\_id)

);

INSERT INTO employees VALUES (7369,'SMITH','CLERK',7902,to\_date('17-12-1980','dd-mm-yyyy'),800,NULL,20);

INSERT INTO employees VALUES (7499,'ALLEN','SALESMAN',7698,to\_date('20-2-1981','dd-mm-yyyy'),1600,300,30);

INSERT INTO employees VALUES (7521,'WARD','SALESMAN',7698,to\_date('22-2-1981','dd-mm-yyyy'),1250,500,30);

INSERT INTO employees VALUES (7566,'JONES','MANAGER',7839,to\_date('2-4-1981','dd-mm-yyyy'),2975,NULL,20);

INSERT INTO employees VALUES (7654,'MARTIN','SALESMAN',7698,to\_date('28-9-1981','dd-mm-yyyy'),1250,1400,30);

INSERT INTO employees VALUES (7698,'BLAKE','MANAGER',7839,to\_date('1-5-1981','dd-mm-yyyy'),2850,NULL,30);

INSERT INTO employees VALUES (7782,'CLARK','MANAGER',7839,to\_date('9-6-1981','dd-mm-yyyy'),2450,NULL,10);

INSERT INTO employees VALUES (7788,'SCOTT','ANALYST',7566,to\_date('13-JUL-87','dd-mm-rr')-85,3000,NULL,20);

INSERT INTO employees VALUES (7839,'KING','PRESIDENT',NULL,to\_date('17-11-1981','dd-mm-yyyy'),5000,NULL,10);

INSERT INTO employees VALUES (7844,'TURNER','SALESMAN',7698,to\_date('8-9-1981','dd-mm-yyyy'),1500,0,30);

INSERT INTO employees VALUES (7876,'ADAMS','CLERK',7788,to\_date('13-JUL-87', 'dd-mm-rr')-51,1100,NULL,20);

INSERT INTO employees VALUES (7900,'JAMES','CLERK',7698,to\_date('3-12-1981','dd-mm-yyyy'),950,NULL,30);

INSERT INTO employees VALUES (7902,'FORD','ANALYST',7566,to\_date('3-12-1981','dd-mm-yyyy'),3000,NULL,20);

INSERT INTO employees VALUES (7934,'MILLER','CLERK',7782,to\_date('23-1-1982','dd-mm-yyyy'),1300,NULL,10);

COMMIT;

Các bảng này là một biến thể của các bảng EMP và DEPT từ lược đồ SCOTT. Bạn sẽ thấy rất nhiều ví dụ của Oracle trên internet bằng cách sử dụng các bảng từ lược đồ SCOTT. Bạn có thể tìm thấy các định nghĩa bảng gốc trong tập lệnh "$ORACLE\_HOME/rdbms/admin/utlsampl.sql".

COMMIT VÀ ROLLBACK

Khi thay đổi dữ liệu để xác nhận thay đổi đó là đúng người dùng cần gõ lệnh COMMIT, khi commit dữ liệu đó sẽ được lưu vĩnh viễn trong datafile và người dùng khác có thể nhìn thấy dữ liệu thay đổi đó.

Khi chưa gõ commit nhưng có lệnh DDL thì sẽ được commit ngầm định, xác nhận tất cả các thay đổi DML nổi bật trong phiên hiện tại.

Nếu bạn quyết định không muốn giữ một số thay đổi chưa commit, bạn có thể loại bỏ chúng bằng cách sử dụng  câu lệnh ROLLBACK để hoàn lại dữ liệu về trạng thái ban đầu.

UPDATE CƠ BẢN

Câu lệnh UPDATE được sử dụng để thay đổi các giá trị cột trong một hàng hiện có. Không có mệnh đề WHERE, tất cả các hàng trong bảng được cập nhật bằng một câu lệnh.

Truy vấn sau đây liệt kê tất cả các giá trị SALARY trong bảng EMPLOYEE.

SELECT e.salary

FROM employees e

ORDER BY e.salary;

SALARY

----------

800

950

1100

1250

1250

1300

1500

1600

2450

2850

2975

3000

3000

5000

14 rows selected.

SQL>

Hãy cập nhật tất cả bằng câu lệnh sau:

UPDATE employees

SET salary = salary + 1;

**14 rows updated.**

SQL>

SELECT e.salary

FROM employees e

ORDER BY e.salary;

SALARY

----------

801

951

1101

1251

1251

1301

1501

1601

2451

2851

2976

3001

3001

5001

14 rows selected.

SQL>

Mệnh đề WHERE được sử dụng để giới hạn các dòng dữ liệu. Chúng ta có thể muốn cập nhật một số lượng nhỏ hàng hoặc thậm chí một hàng bằng cách sử dụng bộ lọc trên (các) cột khóa chính của bảng.

UPDATE employees

SET salary = salary - 1

WHERE salary >= 2000;

**6 rows updated.**

SQL>

SELECT e.salary

FROM employees e

ORDER BY e.salary;

SALARY

----------

801

951

1101

1251

1251

1301

1501

1601

2450

2850

2975

3000

3000

5000

14 rows selected.

SQL>

-- Revert all the changes.

ROLLBACK;

Nhiều giá trị cột có thể được sửa đổi trong một câu lệnh UPDATE.

UPDATE employees

SET **salary = 9999,**

**commission = 1000,**

**manager\_id = 7566**

WHERE employee\_id = 7369;

1 row updated.

SQL>

ROLLBACK;

UPDATE QUA VIEW

Có thể cập nhật bảng cơ sở được liên kết với một view.

CREATE OR REPLACE VIEW employees\_v AS

SELECT \* FROM employees;

UPDATE **employees\_v**

SET salary = 1000

WHERE employee\_id = 7369;

1 row updated.

SQL>

ROLLBACK;

Bạn sẽ không thấy nó rất thường xuyên, nhưng bạn cũng có thể cập nhật thông qua các inline view. Điều này có thể được sử dụng để kiểm soát số lượng hàng được cập nhật, thay vì sử dụng bộ lọc trong mệnh đề  WHERE của  câu lệnh UPDATE.

UPDATE (**SELECT employee\_id, salary**

**FROM employees**

**WHERE department\_id = 20**)

SET salary = 4000;

5 rows updated.

SQL>

ROLLBACK;

TRUY VẤN CON  VÔ HƯỚNG (SCALAR SUBQUERIES)

Giá trị cập nhật có thể đến từ một truy vấn con vô hướng. Phần sau đây UPDATE SALARY của nhân viên 7369, cho nhân viên được trả lương cao nhất.

UPDATE employees

SET salary = (SELECT MAX(salary) FROM employees)

WHERE employee\_id = 7369;

**1 row updated.**

SQL>

ROLLBACK;

CẬP NHẬT 0 HÀNG

Bản cập nhật không có hàng là bản cập nhật hợp lệ và không dẫn đến lỗi. Điều này có thể gây nhầm lẫn cho người mới bắt đầu.

UPDATE employees

SET salary = 10000

WHERE employee\_id = 9999;

0 rows updated.

SQL>

Do đó, bạn không thể kiểm tra lỗi không cập nhật các hàng bằng ngoại lệ NO\_DATA\_FOUND trong PL/SQL:

SET SERVEROUTPUT ON

BEGIN

UPDATE employees

SET salary = 10000

WHERE employee\_id = 9999;

DBMS\_OUTPUT.put\_line('NO\_DATA\_FOUND Not Raised');

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.put\_line('NO\_DATA\_FOUND Raised');

END;

/

**NO\_DATA\_FOUND Not Raised**

PL/SQL procedure successfully completed.

SQL>

Thay vào đó, bạn phải kiểm tra thủ công số hàng được cập nhật bằng cách sử dụng SQL%ROWCOUNT.

SET SERVEROUTPUT ON

BEGIN

UPDATE employees

SET salary = 10000

WHERE employee\_id = 9999;

**IF SQL%ROWCOUNT = 0 THEN**

**-- Manually raise the NO\_DATA\_FOUND exception.**

**RAISE NO\_DATA\_FOUND;**

**END IF;**

DBMS\_OUTPUT.put\_line('NO\_DATA\_FOUND Not Raised');

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.put\_line('NO\_DATA\_FOUND Raised');

END;

/

**NO\_DATA\_FOUND Raised**

PL/SQL procedure successfully completed.

SQL>

### SQL Tutorial - Bài 10: DELETE và TRUNCATE

Trong bài viết này, chúng ta hãy xem xét các Câu lệnh **DELETE**và **TRUNCATE TABLE**.

* Thiết lập môi trường
* COMMIT và ROLLBACK
* DELETE cơ bản
* DELETE qua View
* 0 Rows Deleted
* TRUNCATE TABLE

THIẾT LẬP MÔI TRƯỜNG

Bạn có thể thực hiện tất cả các truy vấn trực tuyến miễn phí bằng [SQL Fiddle](http://sqlfiddle.com/#!4) .

Các ví dụ trong bài viết này yêu cầu phải có các bảng sau đây.

--DROP TABLE employees PURGE;

--DROP TABLE departments PURGE;

CREATE TABLE departments (

department\_id NUMBER(2) CONSTRAINT departments\_pk PRIMARY KEY,

department\_name VARCHAR2(14),

location VARCHAR2(13)

);

INSERT INTO departments VALUES (10,'ACCOUNTING','NEW YORK');

INSERT INTO departments VALUES (20,'RESEARCH','DALLAS');

INSERT INTO departments VALUES (30,'SALES','CHICAGO');

INSERT INTO departments VALUES (40,'OPERATIONS','BOSTON');

COMMIT;

CREATE TABLE employees (

employee\_id NUMBER(4) CONSTRAINT employees\_pk PRIMARY KEY,

employee\_name VARCHAR2(10),

job VARCHAR2(9),

manager\_id NUMBER(4),

hiredate DATE,

salary NUMBER(7,2),

commission NUMBER(7,2),

department\_id NUMBER(2) CONSTRAINT emp\_department\_id\_fk REFERENCES departments(department\_id)

);

INSERT INTO employees VALUES (7369,'SMITH','CLERK',7902,to\_date('17-12-1980','dd-mm-yyyy'),800,NULL,20);

INSERT INTO employees VALUES (7499,'ALLEN','SALESMAN',7698,to\_date('20-2-1981','dd-mm-yyyy'),1600,300,30);

INSERT INTO employees VALUES (7521,'WARD','SALESMAN',7698,to\_date('22-2-1981','dd-mm-yyyy'),1250,500,30);

INSERT INTO employees VALUES (7566,'JONES','MANAGER',7839,to\_date('2-4-1981','dd-mm-yyyy'),2975,NULL,20);

INSERT INTO employees VALUES (7654,'MARTIN','SALESMAN',7698,to\_date('28-9-1981','dd-mm-yyyy'),1250,1400,30);

INSERT INTO employees VALUES (7698,'BLAKE','MANAGER',7839,to\_date('1-5-1981','dd-mm-yyyy'),2850,NULL,30);

INSERT INTO employees VALUES (7782,'CLARK','MANAGER',7839,to\_date('9-6-1981','dd-mm-yyyy'),2450,NULL,10);

INSERT INTO employees VALUES (7788,'SCOTT','ANALYST',7566,to\_date('13-JUL-87','dd-mm-rr')-85,3000,NULL,20);

INSERT INTO employees VALUES (7839,'KING','PRESIDENT',NULL,to\_date('17-11-1981','dd-mm-yyyy'),5000,NULL,10);

INSERT INTO employees VALUES (7844,'TURNER','SALESMAN',7698,to\_date('8-9-1981','dd-mm-yyyy'),1500,0,30);

INSERT INTO employees VALUES (7876,'ADAMS','CLERK',7788,to\_date('13-JUL-87', 'dd-mm-rr')-51,1100,NULL,20);

INSERT INTO employees VALUES (7900,'JAMES','CLERK',7698,to\_date('3-12-1981','dd-mm-yyyy'),950,NULL,30);

INSERT INTO employees VALUES (7902,'FORD','ANALYST',7566,to\_date('3-12-1981','dd-mm-yyyy'),3000,NULL,20);

INSERT INTO employees VALUES (7934,'MILLER','CLERK',7782,to\_date('23-1-1982','dd-mm-yyyy'),1300,NULL,10);

COMMIT;

Các bảng này là một biến thể của các bảng EMP và DEPT từ lược đồ SCOTT. Bạn sẽ thấy rất nhiều ví dụ của Oracle trên internet bằng cách sử dụng các bảng từ lược đồ SCOTT. Bạn có thể tìm thấy các định nghĩa bảng gốc trong tập lệnh "$ORACLE\_HOME/rdbms/admin/utlsampl.sql".

COMMIT VÀ ROLLBACK

Khi thay đổi dữ liệu để xác nhận thay đổi đó là đúng người dùng cần gõ lệnh COMMIT, khi commit dữ liệu đó sẽ được lưu vĩnh viễn trong datafile và người dùng khác có thể nhìn thấy dữ liệu thay đổi đó.

Khi chưa gõ commit nhưng có lệnh DDL thì sẽ được commit ngầm định, xác nhận tất cả các thay đổi DML nổi bật trong phiên hiện tại.

Nếu bạn quyết định không muốn giữ một số thay đổi chưa commit, bạn có thể loại bỏ chúng bằng cách sử dụng  câu lệnh ROLLBACK để hoàn nguyên dữ liệu về trạng thái ban đầu.

DELETE CƠ BẢN

Câu lệnh DELETE được sử dụng để xóa các row khỏi bảng. Không có mệnh đề WHERE tất cả các row trong bảng sẽ bị xóa bởi một câu lệnh.

Ví dụ sau đây xóa tất cả các hàng khỏi bảng EMPLOYEES, sau đó ROLLBACK để hủy xóa.

**DELETE FROM employees;**

14 rows deleted.

SQL>

ROLLBACK;

Mệnh đề WHERE cho phép bạn giới hạn các row bị xóa.

DELETE FROM employees

**WHERE employee\_id = 7369;**

1 row deleted.

SQL>

ROLLBACK;

DELETE QUA VIEW

Có thể xóa khỏi bảng cơ sở được liên kết với một view.  Trong ví dụ sau, chúng ta tạo một view đơn giản trên bảng EMPLOYEES, sau đó xóa nó.

CREATE OR REPLACE VIEW employees\_v AS

SELECT \* FROM employees;

DELETE FROM employees\_v

WHERE employee\_id = 7369;

1 row updated.

SQL>

ROLLBACK;

Bạn sẽ không thấy nó rất thường xuyên, nhưng bạn cũng có thể xóa qua chế độ xem nội tuyến. Điều này có thể được sử dụng để kiểm soát số hàng bị xóa, thay vì sử dụng bộ lọc trong WHEREmệnh đề của DELETEchính câu lệnh.

DELETE FROM (SELECT employee\_id, salary

FROM employees

WHERE department\_id = 20);

5 rows deleted.

SQL>

ROLLBACK;

0 ROWS DELETED

Việc xóa các row bằng 0 là việc xóa hợp lệ và không gây ra lỗi. Điều này có thể gây nhầm lẫn cho người mới bắt đầu.

DELETE FROM employees

WHERE employee\_id = 9999;

0 rows deleted.

SQL>

Do đó, bạn không thể kiểm tra việc không xóa các row bằng cách sử dụng ngoại lệ NO\_DATA\_FOUND trong PL/SQL:

SET SERVEROUTPUT ON

BEGIN

DELETE FROM employees

WHERE employee\_id = 9999;

DBMS\_OUTPUT.put\_line('NO\_DATA\_FOUND Not Raised');

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.put\_line('NO\_DATA\_FOUND Raised');

END;

/

**NO\_DATA\_FOUND Not Raised**

PL/SQL procedure successfully completed.

SQL>

Thay vào đó, bạn phải kiểm tra thủ công số lượng row đã xóa bằng cách sử dụng SQL%ROWCOUNT:

SET SERVEROUTPUT ON

BEGIN

DELETE FROM employees

WHERE employee\_id = 9999;

**IF SQL%ROWCOUNT = 0 THEN**

**-- Manually raise the NO\_DATA\_FOUND exception.**

**RAISE NO\_DATA\_FOUND;**

**END IF;**

DBMS\_OUTPUT.put\_line('NO\_DATA\_FOUND Not Raised');

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.put\_line('NO\_DATA\_FOUND Raised');

END;

/

**NO\_DATA\_FOUND Raised**

PL/SQL procedure successfully completed.

SQL>

TRUNCATE TABLE

Nếu bạn muốn xóa tất cả các row khỏi bảng, câu lệnh TRUNCATE TABLE này hiệu quả hơn nhiều so với câu lệnh DELETE. Câu lệnh TRUNCATE TABLE là một lệnh DDL, vì vậy nó bao gồm một lệnh ngầm COMMIT, vì vậy không có cách nào ROLLBACK lại dữ liệu khi đã gõ TRUNCATE TABLE.

Trong ví dụ sau, chúng ta kiểm tra số row trong bảng, đưa ra câu lệnh TRUNCATE TABLE, ngay lập tức ROLLBACK và kiểm tra lại số row trong bảng. Bạn sẽ thấy từ đầu ra, ROLLBACK không hủy bỏ được câu lệnh TRUNCATE TABLE nên **RẤT CẨN THẬN**khi gõ TRUNCATE

SELECT COUNT(\*)

FROM employees;

COUNT(\*)

----------

**14**

1 row selected.

SQL> **TRUNCATE TABLE employees;**

Table truncated.

SQL> **ROLLBACK;**

Rollback complete.

SQL>

SELECT COUNT(\*)

FROM employees;

COUNT(\*)

----------

**0**

1 row selected.

SQL>

Câu lệnh TRUNCATE TABLE có thể xóa dung lượng lưu trữ được liên kết với bảng hoặc để nó được sử dụng lại sau này.

-- Xóa luôn dung lượng lưu trữ.

TRUNCATE TABLE employees;

TRUNCATE TABLE employees DROP STORAGE;

-- Giữ lại dung lượng lưu trữ.

TRUNCATE TABLE employees REUSE STORAGE;

## Tài liệu tham khảo:

<https://www.tranvanbinh.vn/p/sql-sql-co-ban.html>

# PL/SQL

## PL/SQL là gì? Hiểu sâu về PL/SQL

Mục lục nội dung

1. PL/SQL là gì?

2. Đôi điều lưu ý về PL/SQL

2.1 Embedded language – Ngôn ngữ nhúng

2.2 Hiệu suất cao

3. Kiến trúc PL/SQL

4. Ưu nhược điểm của PL/SQL

4.1 Ưu điểm

4.2 Nhược điểm

5. Ví dụ

6. Tham khảo

Anh em có cơ hội làm việc với hệ cơ sở dữ liệu Oracle sẽ không còn lạ lẫm gì với PL/SQL. Nhưng kiến trúc cụ thể của PL/SQL là gì? PLSQL có ưu nhược điểm gì không?

Bài viết dành cho anh em mới tập tành sử dụng PL/SQL nhưng cũng vô cùng bổ ích cho anh em đã có thời gian dài sử dụng.

Ơ kìa, exception chỗ nào cũng cần được xử lý chứ

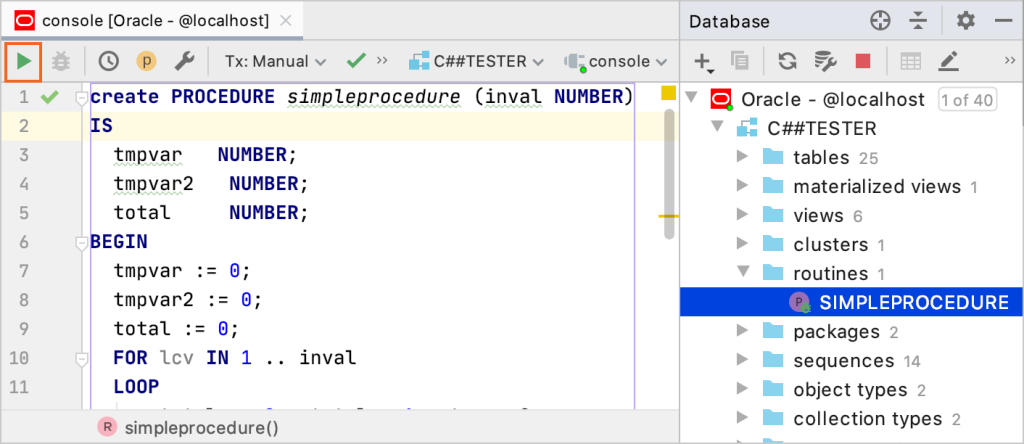
1. PL/SQL LÀ GÌ?

Về SQL thì anh em không còn lạ lẫm gì. SQL là viết tắt của Structural Query Language (ngôn ngữ truy vấn có cấu trúc). Vậy PL trong PL/SQL là gì?

PL ở đây là Procedural Language extensions (phần mở rộng ngôn ngữ thủ tục). Bản thân SQL là cách thức giao tiếp chuẩn để làm việc với các hệ cơ sở dữ liệu RDBMS.

PL/SQL là thành phần mở rộng thêm. Thành phần này giúp khắc phục một số hạn chế của SQL, bênh cạnh đó PL/SQL cũng cung cấp các giải pháp ngôn ngữ lập trình toàn diện hơn giúp xây dựng các ứng dụng dựa trên **hệ cơ sở dữ liệu Oracle**.

Chính bởi mục đích khắc phục một số hạn chế cố hữu của SQL nên PL/SQL là một ngôn ngữ bậc cao. Anh em có thể xem nó như là một **ngôn ngữ lập trình**.

Procedure của PL/SQL sử dụng keywords như các ngôn ngữ tự nhiên, sao lại không giống code =)))

2. ĐÔI ĐIỀU LƯU Ý VỀ PL/SQL

Sau khi đã nắm được PL/SQL là phần mở rộng hơn của SQL, sử dụng để giao tiếp với hệ cơ sở dữ liệu Oracle. Tiếp theo đây sẽ là một vài lưu ý nhỏ cho anh em khi sử dụng PL/SQL

2.1 Embedded language – Ngôn ngữ nhúng

PL/SQL là ngôn ngữ nhúng, bản thân nó chỉ có thể thực thi trong hệ cơ sở dữ liệu Oracle. Do không thể hoạt động độc lập, anh em đừng nhầm lẫn nó với các ngôn ngữ lập trình khác như Java, C++.

Cũng chính vì phụ thuộc vào hệ cơ sở dữ liệu Oracle nên anh em khi sử dụng RDBMS khác như Postgres hay Maria, PL SQL lúc đó không có ý nghĩa gì nữa đâu nha.

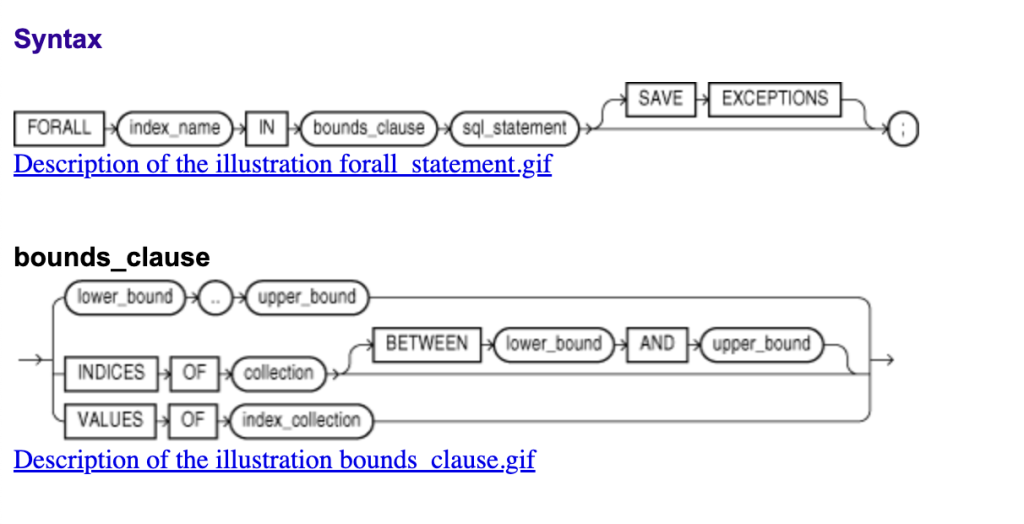


2.2 Hiệu suất cao

Bản thân PL/SQL là ngôn ngữ sinh ra để giao tiếp với hệ cơ sở dữ liệu. Nên so về hiệu suất, ở đây chưa bàn tới hiệu suất thực thi vì nó phụ thuộc vào hệ cơ sở dữ liệu.

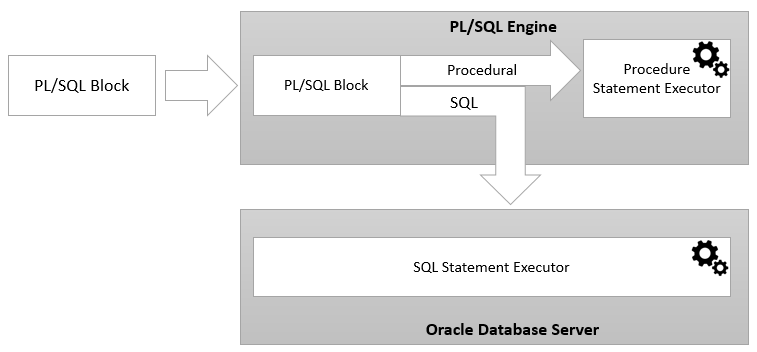
Nếu chỉ so sánh về hiệu suất cách viết, độ dễ để viết thì PL/SQL hơn hẳn các ngôn ngữ khác như Java, Python, C++. Mặc dù các ngôn ngữ này đều có thể thao tác với hệ cơ sở dữ liệu thông qua ODBC.

Ngoài ra, nếu nói về hiệu suất, PL/SQL cung cấp FORALL, sử dụng FORALL sẽ mang lại hiệu suất tốt hơn so với việc thao tác bằng các ngôn ngữ khác.

*The FORALL statement issues a series of INSERT, UPDATE, or DELETE statements, usually much faster than an equivalent FOR loop.*Câu lệnh forall đưa ra một loạt câu lệnh INSERT, UPDATE hoặc DELETE, thường nhanh hơn nhiều so với vòng lặp for tương đương

3. KIẾN TRÚC PL/SQL

Sơ bộ anh em có thể hình dung PL/SQL là ngôn ngữ hỗ trợ nằm ngoài Oracle, bằng cách nào đó ngôn ngữ viết bằng PL/SQL có thể được dịch ra thành SQL để thực thi ở hệ cơ sở dữ liệu Oracle. Vậy cụ thể kiến trúc PL/SQL như thế nào?

Kiến trúc PLSQL

Theo như hình kiến trúc phía trên, PL/SQL sẽ chịu trách nhiệm biên dịch code anh em viết thành byte-code. Một đoạn code từ PL/SQL block được gửi tới PL/SQL Engine, đoạn code sẽ được biên dịch và thực thi ở Oracle Database Server.

4. ƯU NHƯỢC ĐIỂM CỦA PL/SQL

4.1 Ưu điểm

* PL/SQL cung cấp hiệu suất tốt hơn (cái này trong một vài trường hợp cụ thể thôi nha, không phải lúc nào cũng thế)
* PL/SQL hỗ trợ Object-Oriented Programming concepts (nguyên lý lập trình hướng đối tượng), cho phép kế thừa, …) cái này tiện lợi cho anh em lập trình đã nắm vững nguyên lý OOP.
* PL/SQL hỗ trợ nhiều công cụ phát triển Web (Web application)

4.2 Nhược điểm

* Cái này thì chắc là xưa cũ rồi vì giờ máy anh em nào cũng mạnh. PL/SQL cần bộ nhớ nhiều (chạy tốn ram khá là khủng khiếp nha)
* Ở các procedures thì tính năng debug không có, viết ra anh em phải debug bằng mắt chứ PL/SQL không hỗ trợ.

5. VÍ DỤ

Giải thích rồi, kiến trúc rồi. Giờ tới lúc cho anh em ví dụ về PL/SQL nha. Vì PL/SQL ở trên đã nói với anh em là giống như ngôn ngữ lập trình nên có một vài cú pháp anh em bắt buộc phải theo nha

Ví dụ như := là gán, 1..10 là từ 1 tới 10. Kết thúc vòng lặp thì phải là END LOOP.

-- available online in file 'sample1'

// Khai báo biến x bằng 100

DECLARE

x NUMBER := 100;

BEGIN

// Lặp i từ 1 tới 10

FOR i IN 1..10 LOOP

// Nếu i chia hết cho 2, i là số chẵn

IF MOD(i,2) = 0 THEN -- i is even

// Thêm vào bảng tạm 1 dòng với cột thứ 3 kiểu chuỗi nói rằng i là số chẵn

INSERT INTO temp VALUES (i, x, 'i is even');

ELSE

// Ngược lại thì cũng insert một dòng như i là số lẻ

INSERT INTO temp VALUES (i, x, 'i is odd');

END IF;

// Tăng giá trị x thêm 100

x := x + 100;

END LOOP;

// Chạy hết tất cả trong vòng for rồi insert một lượt vào DB, kiểu thực hiện toàn bộ câu iinssert một lần

COMMIT;

END;

## PL/SQL Tutorial

### Bài Giới thiệu - Oracle PL/SQL cơ bản

**1.**  **Cấu trúc**

Cấu trúc đầy đủ của một khối lệnh PL/SQL bao gồm:

*DECLARE /\* Phần khai báo - Không bắt buộc \*/*

*Khai báo các biến sử dụng trong phần thân*

*BEGIN /\* Phần thân \*/*

*Đoạn lệnh thực hiện;*

*EXCEPTION /\* Phần xử lý lỗi - Không bắt buộc \*/*

*Xử lý lỗi xảy ra;*

*END;*

**Ví dụ1:**

*DECLARE*

*empno     NUMBER(4):=7788;*

*BEGIN*

*UPDATE emp*

*SET sal = 9000*

*WHERE empno = 0001;*

*END;*

**Ví dụ 2:**

*DECLARE*

*v\_deptno     NUMBER(2);*

*v\_loc     VARCHAR2(15);*

*BEGIN*

*SELECT deptno, loc INTO v\_deptno, v\_loc*

*FROM dept*

*WHERE dname = 'SALES';*

*EXCEPTION*

*WHEN others THEN*

*Message(SQLERRM);*

*END;*

**2.**  **Gọi 1 thủ tục**

         **Từ SQL\*Plus:**

*SQL> execute binhtv.send\_sms('Test');*

         **Từ Tool:**

*Begin*

*Binhtv.send\_sms('Test');*

*End;*

Hoặc từ SQL Navigator:

*execute binhtv.send\_sms('Test');*

         **Từ RMAN**

*SQL "BEGIN binhtv.send\_sms\_binhtv(''Backup Level 1 Success''); END; ";*

(    nháy kép                                    2 nháy đơn….              2 nháy đơn …. Nháy kép)

### Bài 1.PL/SQL cơ bản - Các yếu tố cơ bản, các phép toán

**1.**    **Comment:**

/\* … \*/

--

**2.**    **Khai báo biến**

|  |
| --- |
| SET SERVEROUTPUT ON  DECLARE  Myname VARCHAR2(20);  BEGIN  DBMS\_OUTPUT.PUT\_LINE('My name is: '||Myname);  Myname := 'John';  DBMS\_OUTPUT.PUT\_LINE('My name is: '||Myname);  END;  / |

         **INTO**: đổ dữ liệu vào biến

|  |
| --- |
| declare      result number;  begin      SELECT (HISAL\*12) INTO result FROM SCOTT.SALGRADE WHERE grade = 4;      DBMS\_OUTPUT.PUT\_LINE(result);  end;  / |

|  |
| --- |
| DECLARE  emp\_hiredate employees.hire\_date%TYPE;  emp\_salary employees.salary%TYPE;  BEGIN  SELECT hire\_date, salary  INTO emp\_hiredate, emp\_salary  FROM employees  WHERE employee\_id = 100;  END;  / |

         **Khai báo hằng**

**DECLARE**

**v\_char CONSTANT VARCHAR2(10) := 'Hello World';**

**v\_number CONSTANT NUMBER := 123;**

**BEGIN**

**NULL;**

**END;**

         **Toán tử:**

         **Ghép tiếp các cột dữ liệu**

Toán tử ghép tiếp chuỗi (||) cho phép ghép tiếp dữ liệu trong các cột khác nhau của cùng một dòng dữ liệu với nhau thành một chuỗi. Ta có thể có nhiều toán tử ghép chuỗi trong cùng một column alias.

Ví dụ:

SELECT empno||ename EMPLOYEE FROM emp;

         **Ghép tiếp chuỗi ký tự**

Trong mệnh đề SELECT, ta có thể thực hiện ghép tiếp bất kỳ ký tự nào, biểu thức hay số nào mà không phải là column hoặc column alias.

Ví dụ:

SELECT empno || ename || ‘ WORK IN DEPARTMENT ’

|| deptno ‘Employee Detail’

FROM emp;

         Để kiểm tra có phải NULL hay không dùng các toán tử IS NULL hoặc IS NOT NULL

The Relational Operators: =, <>, !=, ~=, <, >, <=, >=

|  |  |  |
| --- | --- | --- |
| **Operator** | **Example** | **Usage** |
| = | IF A = B THEN | The equality operator. |
| <> | IF A <> B THEN | The inequality operator. |
| != | IF A != B THEN | Another inequality operator, synonymous with <>. |
| ~= | IF A ~= B THEN | Another inequality operator, synonymous with <>. |
| < | IF A < B THEN | The less than operator. |
| > | IF A > B THEN | The greater than operator. |
| <= | IF A <= B THEN | The less than or equal to operator. |
| >= | IF A >= B THEN | The greater than or equal to operator. |
| LIKE | IF A LIKE B THEN | The pattern-matching operator. |
| BETWEEN | IF A BETWEEN B AND C THEN | Checks to see if a value lies within a specified range of values. |
| IN | IF A IN (B,C,D) THEN | Checks to see if a value lies within a specified list of values. |
| IS NULL | IF A IS NULL THEN | Checks to see if a value is null. |

|  |  |
| --- | --- |
| **True Expressions** | **False Expressions** |
| 5 = 5 | 5 = 3 |
| 'AAAA' = 'AAAA' | 'AAAA ' = 'AAAA' |
| 5 != 3 | 5 <> 5 |
| 'AAAA ' ~= 'AAAA' | 'AAAA' ~= 'AAAA' |
| 10 < 200 | 10.1 < 10.05 |
| 'Jeff' < 'Jenny' | 'jeff' < 'Jeff' |
| TO\_DATE('15-Nov-61' < '15-Nov-97') | TO\_DATE('1-Jan-97' < '1-Jan-96') |
| 10.1 <= 10.1 | 10 <= 20 |
| 'A' <= 'B' | 'B' <= 'A' |
| TO\_DATE('1-Jan-97') <= TO\_DATE('1-Jan-97) | TO\_DATE('15-Nov-61') <= TO\_DATE('15-Nov-60) |

Để tìm những nhân viên có tên có chuỗi 'A\_B'

SELECT ename FROM emp WHERE ename LIKE '%A\\_B%'; ESCAPE '\'

Vì ký hiệu "\_" dùng để đại diện cho một ký tự bất kỳ nên nếu không có mệnh đề ESCAPE, câu lệnh trên sẽ tìm tất cả các nhân viên tên AAB, ABB, ACB, v.v...

          **Toán tử tập hợp**

UNION: Kết hợp kết quả của nhiều câu hỏi với nhau, chỉ giữ lại một đại diện cho các mẫu tin trùng nhau.

UNION ALL: Kết hợp kết quả của nhiều câu hỏi với nhau, các mẫu tin trùng nhau cũng được lặp lại

INTERSET:Lấy phần giao các kết quả của nhiều câu hỏi

MINUS: Lấy kết quả có trong câu hỏi thứ nhất mà không có trong câu hỏi thứ hai (câu hỏi sau toán tử MINUS)

     **Toán tử khác:**

NOT IN: Không thuộc

ANY và SOME: So sánh một giá trị với mỗi giá trị trong một danh sách hay trong kết quả trả về của câu hỏi con, phải sau toán tử =

ALL:So sánh một giá trị với mọi giá trị trong danh sách hay trong kết quả trả về của câu hỏi con.

EXISTS: Trả về TRUE nếu có tồn tại.

SELECT \* FROM emp

WHERE sal = ANY (SELECT sal FROM emp WHERE deptno=30);

SELECT \* FROM emp

WHERE sal >= ALL (select distinct sal From emp

Where deptno =30)

Order by sal desc;

SELECT ENAME, SAL, JOB, DEPTNO

FROM EMP

WHERE SAL > SOME (SELECT DISTINCT SAL

FROM EMP

WHERE DEPTNO =30)

ORDER BY SAL DESC;

Tìm những người có nhân viên

SELECT EMPNO, ENAME, JOB, DEPTNO

FROM EMP E

WHERE EXISTS (SELECT EMPNO FROM EMP WHERE EMP.MGR = E.EMPNO);

         **Constant**

You cannot use SYSDATE in the condition of a CHECK constraint

**5.**    **Từ khóa**

         **DEFINE:**

         **Chuyển đổi kiểu dữ liệu (converision)**

|  |  |
| --- | --- |
| Explicit (Tường minh) | – TO\_CHAR  – TO\_DATE  – TO\_NUMBER  – TO\_TIMESTAMP |
| Implicit (internal, tự động) |  |
|  |  |

### Bài 2.PL/SQL cơ bản - Cách tổ chức dữ liệu

1.     **Kiểu DL cơ sở:**

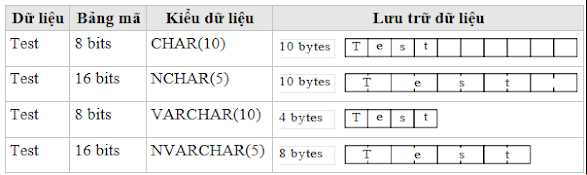
|  |  |
| --- | --- |
| **Basic Type** | **Specific Types** |
| Numeric | INTEGER, SHORTINTEGER, LONGINTEGER, DECIMAL, SHORTDECIMAL, NUMBER |
| Text | TEXT, NTEXT, ID |
| Boolean | BOOLEAN |
| Date | DATETIME, DATE |

**Character Datatypes**

|  |  |
| --- | --- |
| **Đúng** | **Sai** |
| CUSTNO NOT NULL CHAR(2)  SELECT transamt FROM cust\_trans WHERE custno > '11'; |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Type**  **Syntax** | **Oracle 9i** | **Oracle 10g** | **Oracle 11g** | **Explanation**  **(if applicable)** |
| Char[size] | Maximum size of 2000 bytes. | Maximum size of 2000 bytes. | Maximum size of 2000 bytes. | Where ***size*** is the number of characters to store. Fixed-length strings. Space padded.  Size: có thể có hoặc không (VD column1 char hoặc column1 char(10)) |
| nchar(size) | Maximum size of 2000 bytes. | Maximum size of 2000 bytes. | Maximum size of 2000 bytes. | Where ***size*** is the number of characters to store. Fixed-length NLS string Space padded. |
| nvarchar2(size) | Maximum size of 4000 bytes. | Maximum size of 4000 bytes. | Maximum size of 4000 bytes. | Where ***size*** is the number of characters to store. Variable-length NLS string. |
| varchar2(size) | Maximum size of 4000 bytes. | Maximum size of 4000 bytes. | Minimum size is 1; maximum size is 4,000.) | Where ***size*** is the number of characters to store. Variable-length string.  Size: bắt buộc (VD colum1 varchar2 sẽ báo lỗi) |
| long | Maximum size of 2GB. | Maximum size of 2GB. | Maximum size of 2GB. | Variable-length strings. (backward compatible)  An EMP\_IMAGE column cannot be included in the ORDER BY clause.  You cannot add a new column to the table with LONG as the data type.  You cannot alter the table to include the NOT NULL constraint on the EMP\_IMAGE column. |
| raw | Maximum size of 2000 bytes. | Maximum size of 2000 bytes. | Maximum size of 2000 bytes. | Variable-length binary strings |
| long raw | Maximum size of 2GB. | Maximum size of 2GB. | Maximum size of 2GB. | Variable-length binary strings. (backward compatible) |

Ví dụ minh hoạ việc lưu trữ dữ liệu xâu chữ giữa các kiểu dữ liệu khác nhau

[](https://blogger.googleusercontent.com/img/a/AVvXsEinfIWnl0j3EXSzTN8lUPaEBrkNQgMItRRx91bAdniV1eW8x2Si3ELHiU5UYFywXDWsvTYvGXrcOHqnOb_mOXxjhBkVk-8PkY3zIOBVrxpSR3DmPD6Gy38vlJycvQRcukaPxuKS0Tfjl7lplwFmrcw79S1ZDDVr2bXjXNuiYJxBQLKu_TnRssm1L5qf)

**Numeric Datatypes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Type**  **Syntax** | **Oracle 9i** | **Oracle 10g** | **Oracle 11g** | **Explanation**  **(if applicable)** |
| number(p,s) | Precision can range from 1 to 38.  Scale can range from -84 to 127. | Precision can range from 1 to 38.  Scale can range from -84 to 127. | Precision can range from 1 to 38.  Scale can range from -84 to 127. | Where ***p*** is the precision and ***s*** is the scale.  For example, number(7,2) is a number that has 5 digits before the decimal and 2 digits after the decimal.   |  |  |  | | --- | --- | --- | | **Input Data** | **Specified As** | **Stored As** | | 7,456,123.89 | **NUMBER** | 7456123.89 | | 7,456,123.89 | **NUMBER(\*,1)** | 7456123.9 | | 7,456,123.89 | **NUMBER(9)** | 7456124 | | 7,456,123.89 | **NUMBER(9,2)** | 7456123.89 | | 7,456,123.89 | **NUMBER(9,1)** | 7456123.9 | | 7,456,123.89 | **NUMBER(6)** | (not accepted, exceeds precision) | | 7,456,123.89 | **NUMBER(7,-2)** | 7456100 |   *From <*[*https://docs.oracle.com/cd/B28359\_01/server.111/b28318/datatype.htm*](https://docs.oracle.com/cd/B28359_01/server.111/b28318/datatype.htm)*>* |
| numeric(p,s) | Precision can range from 1 to 38. | Precision can range from 1 to 38. | Precision can range from 1 to 38. | Where ***p*** is the precision and ***s*** is the scale.  For example, numeric(7,2) is a number that has 5 digits before the decimal and 2 digits after the decimal. |
| float |  |  |  | CREATE TABLE test (col1 NUMBER(5,2), col2 FLOAT(5));  INSERT INTO test VALUES (1.23, 1.23); INSERT INTO test VALUES (7.89, 7.89); INSERT INTO test VALUES (12.79, 12.79); INSERT INTO test VALUES (123.45, 123.45);  SELECT \* FROM test;       COL1       COL2 ---------- ----------       1.23        1.2       7.89        7.9      12.79         13     123.45        120  *From <*[*https://docs.oracle.com/cd/B28359\_01/server.111/b28286/sql\_elements001.htm*](https://docs.oracle.com/cd/B28359_01/server.111/b28286/sql_elements001.htm)*>* |
| dec(p,s) | Precision can range from 1 to 38. | Precision can range from 1 to 38. | Precision can range from 1 to 38. | Where ***p*** is the precision and ***s*** is the scale.  For example, dec(3,1) is a number that has 2 digits before the decimal and 1 digit after the decimal. |
| decimal(p,s) | Precision can range from 1 to 38. | Precision can range from 1 to 38. | Precision can range from 1 to 38. | Where ***p*** is the precision and ***s*** is the scale.  For example, decimal(3,1) is a number that has 2 digits before the decimal and 1 digit after the decimal. |
| integer |  |  |  |  |
| int |  |  |  |  |
| smallint |  |  |  |  |
| real |  |  |  |  |
| double precision |  |  |  |  |

**Date/Time Datatypes**

|  |  |
| --- | --- |
| Đúng | Sai |
| inv\_date = '15-february-2008' : uses implicit conversion | inv\_date > '01-02-2008' : uses implicit conversion  --> Chuyển thành inv\_date>to\_date('01-02-2008','dd-mm-yyyy'), inv\_date kiểu date |
| inv\_no BETWEEN '101' AND '110' : uses implicit conversion |  |
| TRANSDATE DATE  SELECT transdate + '10' FROM cust\_trans; |  |
| SELECT SYSDATE - TO\_DATE('01/JANUARY/2007') FROM DUAL;  SELECT SYSDATE - TO\_DATE('01-JANUARY-2007') FROM DUAL; | SELECT SYSDATE - '01-JAN-2007' FROM DUAL;  SELECT TO\_CHAR(SYSDATE, 'DD-MON-YYYY') - '01-JAN-2007' FROM DUAL;  SELECT TO\_DATE(SYSDATE, 'DD/MONTH/YYYY') - '01/JANUARY/2007' FROM DUAL; |
| Elements of the Date Format Model  ---------------------------------------------------------------------  DY Three-letter abbreviation of the day of the week  DAY Full name of the day of the week  DD Numeric day of the month  MM Two-digit value for the month  MON Three-letter abbreviation of the month  MONTH Full name of the month  YYYY Full year in numbers  YEAR Year spelled out (in English) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Type**  **Syntax** | **Oracle 9i** | **Oracle 10g** | **Oracle 11g** | **Explanation**  **(if applicable)** |
| date | A date between Jan 1, 4712 BC and Dec 31, 9999 AD. | A date between Jan 1, 4712 BC and Dec 31, 9999 AD. | A date between Jan 1, 4712 BC and Dec 31, 9999 AD. |  |
| timestamp (*fractional seconds precision*) | ***fractional seconds precision*** must be a number between 0 and 9. (default is 6) | ***fractional seconds precision*** must be a number between 0 and 9. (default is 6) | ***fractional seconds precision*** must be a number between 0 and 9. (default is 6) | Includes year, month, day, hour, minute, and seconds.  For example:  timestamp(6) |
| timestamp (*fractional seconds precision*) with time zone | ***fractional seconds precision*** must be a number between 0 and 9. (default is 6) | ***fractional seconds precision*** must be a number between 0 and 9. (default is 6) | ***fractional seconds precision*** must be a number between 0 and 9. (default is 6) | Includes year, month, day, hour, minute, and seconds; with a time zone displacement value.  For example:  timestamp(5) with time zone |
| timestamp (*fractional seconds precision*) with local time zone | ***fractional seconds precision*** must be a number between 0 and 9. (default is 6) | ***fractional seconds precision*** must be a number between 0 and 9. (default is 6) | ***fractional seconds precision*** must be a number between 0 and 9. (default is 6) | Includes year, month, day, hour, minute, and seconds; with a time zone expressed as the session time zone.  For example:  timestamp(4) with local time zone |
| interval year  (*year precision*)  to month | ***year precision*** is the number of digits in the year. (default is 2) | ***year precision*** is the number of digits in the year. (default is 2) | ***year precision*** is the number of digits in the year. (default is 2) | Time period stored in years and months.  For example:  interval year(4) to month |
| interval day  (*day precision*)  to second (*fractional seconds precision*) | ***day precision*** must be a number between 0 and 9. (default is 2)  ***fractional seconds precision*** must be a number between 0 and 9. (default is 6) | ***day precision*** must be a number between 0 and 9. (default is 2)  ***fractional seconds precision*** must be a number between 0 and 9. (default is 6) | ***day precision*** must be a number between 0 and 9. (default is 2)  ***fractional seconds precision*** must be a number between 0 and 9. (default is 6) | Time period stored in days, hours, minutes, and seconds.  For example:  interval day(2) to second(6) |

**Large Object (LOB) Datatypes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Type**  **Syntax** | **Oracle 9i** | **Oracle 10g** | **Oracle 11g** | **Explanation**  **(if applicable)** |
| bfile | Maximum file size of 4GB. | Maximum file size of 232-1 bytes. | Maximum file size of 264-1 bytes. | File locators that point to a binary file on the server file system (outside the database). |
| blob | Store up to 4GB of binary data. | Store up to (4 gigabytes -1) \* (the value of the CHUNK parameter of LOB storage). | Store up to (4 gigabytes -1) \* (the value of the CHUNK parameter of LOB storage). | Stores unstructured binary large objects. |
| clob | Store up to 4GB of character data. | Store up to (4 gigabytes -1) \* (the value of the CHUNK parameter of LOB storage) of character data. | Store up to (4 gigabytes -1) \* (the value of the CHUNK parameter of LOB storage) of character data. | Stores single-byte and multi-byte character data. |
| nclob | Store up to 4GB of character text data. | Store up to (4 gigabytes -1) \* (the value of the CHUNK parameter of LOB storage) of character text data. | Store up to (4 gigabytes -1) \* (the value of the CHUNK parameter of LOB storage) of character text data. | Stores unicode data. |

**Rowid Datatypes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Type**  **Syntax** | **Oracle 9i** | **Oracle 10g** | **Oracle 11g** | **Explanation**  **(if applicable)** |
| rowid | The format of the rowid is: BBBBBBB.RRRR.FFFFF  Where BBBBBBB is the block in the database file;  RRRR is the row in the block;  FFFFF is the database file. | The format of the rowid is: BBBBBBB.RRRR.FFFFF  Where BBBBBBB is the block in the database file;  RRRR is the row in the block;  FFFFF is the database file. | The format of the rowid is: BBBBBBB.RRRR.FFFFF  Where BBBBBBB is the block in the database file;  RRRR is the row in the block;  FFFFF is the database file. | Fixed-length binary data. Every record in the database has a physical address or **rowid**. |
| urowid(size) |  |  |  | Universal rowid.  Where ***size*** is optional. |

*Pasted from <*[*http://www.techonthenet.com/oracle/datatypes.php*](http://www.techonthenet.com/oracle/datatypes.php)*>*

         **Convert dữ liệu**

|  |  |
| --- | --- |
| Implicit Conversion | Char --> Number  Number --> Char |
| Syntax of Explicit Data Type Conversion | Functions  TO\_NUMBER(char1, [format mask], [nls\_parameters]) = num1  TO\_CHAR(num1, [format mask], [nls\_parameters]) = char1  TO\_DATE(char1, [format mask], [nls\_parameters]) = date1  TO\_CHAR(date1, [format mask], [nls\_parameters]) = char1 |

**2.**   **Kiểu dữ liệu có cấu trúc**

         **Cursors**

Cursor là kiểu biến có cấu trúc, cho phép ta xử lý dữ liệu gồm nhiều dòng. Số dòng phụ thuộc vào câu lệnh truy vấn dữ liệu sau nó. Trong quá trình xử lý, ta thao tác với cursor thông qua từng dòng dữ liệu. Dòng dữ liệu này được định vị bởi một con trỏ. Với việc dịch chuyển con trỏ, ta có thể lấy được toàn bộ dữ liệu trả về

|  |
| --- |
| ***-- Thực hiện lệnh từ output của câu lệnh trước***  DECLARE  cursor c1 is  select 'alter index '||index\_owner||'.'||index\_name||' rebuild SUBpartition '||SUBpartition\_name||' tablespace pps\_indx parallel 8 online' name from dba\_ind\_SUBpartitions  where   status like 'UN%'  union all  select 'alter index '||index\_owner||'.'||index\_name||' rebuild partition '||partition\_name||' tablespace pps\_indx parallel 8 online' name from dba\_ind\_partitions  where  status like 'UN%';  begin      for r1 in c1      loop          execute immediate r1.name;      end loop;  end; |

         **Fetch 1 dòng dữ liệu từ cursor**: (phải mở, đóng cursor)

|  |  |
| --- | --- |
| SET SERVEROUTPUT ON  DECLARE  CURSOR emp\_cursor IS  SELECT employee\_id, last\_name FROM employees WHERE department\_id =30;  empno employees.employee\_id%TYPE;  lname employees.last\_name%TYPE;  BEGIN  OPEN emp\_cursor;  FETCH emp\_cursor INTO empno, lname;  DBMS\_OUTPUT.PUT\_LINE(empno ||' '||lname);  CLOSE emp\_cursor;  END;  / | Con trỏ emp\_cursor trả về 6 employees có department\_id=30 nhưng câu lệnh FETCH sẽ lấy về chỉ 1 row, muốn lấy tất cả các row thì phải dùng vòng lặp (for) |

         **Fetch lấy all row từ cursor:** LOOP...FETCH (phải mở, đóng cursor)

|  |  |
| --- | --- |
| DECLARE  -- Khai báo cursor để truy vấn dữ liệu  CURSOR c\_Emp IS  SELECT \*  FROM emp  WHERE dept\_id = 10;  -- Khai báo biến cursor tương ứng để chứa dòng dữ liệu  v\_Emp c\_EMP%rowtype;   -- v\_Emp giống như kiểu record  BEGIN  -- Mở cursor  OPEN c\_Emp;  LOOP  -- Lấy dòng dữ liệu từ cursor  FETCH c\_Emp INTO v\_Emp;  -- Thoát khỏi vòng lặp nếu đã lấy hết dữ liệu trong cursor  EXIT WHEN c\_Emp%notfound;  -- Bổ sung dữ liệu vào Emp\_ext với dữ liệu lấy được từ cursor  INSERT INTO Emp\_ext (empno, ename, job) VALUES (v\_Emp.empno, v\_Emp.ename, v\_Emp.job);  END LOOP;  -- Đóng cursor  CLOSE c\_Emp;  END; | [https://blogger.googleusercontent.com/img/a/AVvXsEjhUx8ApUuFxw5E1FtwpNSNW41AV2C-867Ho8o5t8UUQDiiEoW6ZULMi2kuOUnkJV6VaeyasQEhmzIaqXisrTRuCEQlqxRqItuhtb603HOnfElf__AECTGa8I3_TzWDWa8tEq3ql1TIl8DkPxVtorH5A93L9_hEm3eXV_EZqusee1wHm63Zm4F98Vp0=w262-h121](https://blogger.googleusercontent.com/img/a/AVvXsEjhUx8ApUuFxw5E1FtwpNSNW41AV2C-867Ho8o5t8UUQDiiEoW6ZULMi2kuOUnkJV6VaeyasQEhmzIaqXisrTRuCEQlqxRqItuhtb603HOnfElf__AECTGa8I3_TzWDWa8tEq3ql1TIl8DkPxVtorH5A93L9_hEm3eXV_EZqusee1wHm63Zm4F98Vp0) |

         **Cursor FOR.. LOOP**

|  |
| --- |
| SET SERVEROUTPUT ON  DECLARE  CURSOR c\_Emp IS  SELECT employee\_id, last\_name FROM employees  WHERE department\_id =30;  Emp\_record c\_Emp%ROWTYPE;  BEGIN  FOR emp\_record IN c\_Emp LOOP  DBMS\_OUTPUT.PUT\_LINE(emp\_record.employee\_id||' ' ||emp\_record.last\_name);  END LOOP;  END;  / |

         **%ROWCOUNT and %NOTFOUND**

|  |
| --- |
| SET SERVEROUTPUT ON  DECLARE  empno employees.employee\_id%TYPE;  ename employees.last\_name%TYPE;  CURSOR emp\_cursor IS SELECT employee\_id,  last\_name FROM employees;  BEGIN  OPEN emp\_cursor;  LOOP  FETCH emp\_cursor INTO empno, ename;  EXIT WHEN emp\_cursor%ROWCOUNT > 10 OR emp\_cursor%NOTFOUND;  DBMS\_OUTPUT.PUT\_LINE(TO\_CHAR(empno) ||' '|| ename);  END LOOP;  CLOSE emp\_cursor;  END ;  / |

         **Kiểu dữ liệu Record**

TYPE t\_Emp IS

RECORD OF (

empno     number(4) not null,

ename     char(10),

job     char(9),

mgr     number(4),

hiredate     date default sysdate,

sal     number(7,2),

comm     number(7,2),

deptno     number(2) not null);

v\_Emp\_record     t\_Emp;

+ Gán giá trị:

t\_Emp.empno := 10;

t\_Emp.ename := "BinhTV"

...

         **ROWTYPE: dữ liệu một dòng**

Bản ghi trong PL/SQL. là một biến có thể giữ nhiều giá trị và là một tập hợp các biến tương ứng với các trường trong table.

Khai báo kiểu dữ liệu bản ghi.

Tên biến     Tên bảng%ROWTYPE;

Ví dụ:

v\_Emp emp%ROWTYPE;

Truy nhập đến các trường trong dữ liệu bản ghi dùng giống như trong 1 dòng dữ liệu trả về. Ví dụ:

v\_Emp.empno, v\_Emp.sal, ...

|  |
| --- |
| ...  DEFINE employee\_number = 124  DECLARE  emp\_rec employees%ROWTYPE;  BEGIN  SELECT \* INTO emp\_rec FROM employees  WHERE employee\_id = &employee\_number;  INSERT INTO retired\_emps(empno, ename, job, mgr,hiredate, leavedate, sal, comm, deptno)  VALUES (emp\_rec.employee\_id, emp\_rec.last\_name,emp\_rec.job\_id,emp\_rec.manager\_id,emp\_rec.hire\_date, SYSDATE, emp\_rec.salary,emp\_rec.commission\_pct, emp\_rec.department\_id);  END;  / |
| SET SERVEROUTPUT ON  SET VERIFY OFF  DEFINE employee\_number = 124  DECLARE  emp\_rec retired\_emps%ROWTYPE;  BEGIN  SELECT \* INTO emp\_rec FROM retired\_emps;  emp\_rec.leavedate:=SYSDATE;  UPDATE retired\_emps SET ROW = emp\_rec WHERE  empno=&employee\_number;  END;  /  SELECT \* FROM retired\_emps; |
| DECLARE  v\_data abcd.efgh%TYPE;  v\_data\_row abcd%ROWTYPE;  BEGIN  EXECUTE IMMEDIATE 'SELECT efgh FROM abcd WHERE efgh = :text\_string'  INTO v\_data  USING 'ijkl';  DBMS\_OUTPUT.PUT\_LINE( 'Column Variable: ' || v\_data );  EXECUTE IMMEDIATE 'SELECT \* FROM abcd WHERE efgh = :text\_string'  INTO v\_data\_row  USING 'ijkl';  DBMS\_OUTPUT.PUT\_LINE( 'Row Variable: ' || v\_data\_row.efgh );  END;  INFO: Column Variable: ijkl  INFO: Row Variable: ijkl  *From <*[*http://www.dba-oracle.com/t\_oracle\_execute\_immediate.htm*](http://www.dba-oracle.com/t_oracle_execute_immediate.htm)*>* |

         **TYPE: Kiểu dữ liệu của một cột**

Cú pháp:

Tên biến     Tên cột dữ liệu%TYPE;

|  |
| --- |
| DECLARE  salary NUMBER(6):=6000;  sal\_hike VARCHAR2(5):='1000';  total\_salary **salary%TYPE**; // total\_salary trùng kiểu DL salary  BEGIN  total\_salary:=salary+sal\_hike;  dbms\_output.put\_line(total\_salary);  END;  / |

         **Kiểu dữ liệu Table**

Cú pháp:

TYPE Tên\_kiểu\_Table IS

TABLE OF Tên kiểu dữ liệu [NOT NULL] INDEX BY BINARY\_INTEGER;

Tên biến     Tên\_kiểu\_Table;

Ví dụ:

TYPE t\_Name IS

TABLE OF Emp.Ename%TYPE INDEX BY BINARY\_INTEGER;

v\_First\_name     t\_Name;

v\_Last\_name     t\_Name;

### Bài 3.PL/SQL cơ bản - Cấu trúc điều khiển (if, else, case, for, while)

**1.**    **Cấu trúc tuần tự**

Thực hiện lần lượt từng lệnh một.

dbms\_output.put\_line('Excellent 1');

dbms\_output.put\_line('Excellent 2');

dbms\_output.put\_line('Excellent 3'); 

**2.**    **Cấu trúc rẽ nhánh**

|  |  |
| --- | --- |
| **Rẽ nhánh không đầy đủ (if)** | IF ename = 'SCOTT' THEN  beam\_me\_up := 'YES';  COMMIT; |
| **Rẽ nhánh đầy đủ (if…else)** | IF sales > (quota + 200) THEN        bonus := (sales - quota)/4;      ELSE        bonus := 50;      END IF;  ----  IF monthly\_value <= 4000 THEN       ILevel := 'Low Income';  ELSIF monthly\_value > 4000 and monthly\_value <= 7000 THEN       ILevel := 'Avg Income';  ELSIF monthly\_value > 7000 and monthly\_value <= 15000 THEN       ILevel := 'Moderate Income';  ELSE       ILevel := 'High Income';  END IF; |
| **Case** | CASE v\_grade   WHEN 'A' THEN dbms\_output.put\_line('Excellent');   WHEN 'B' THEN dbms\_output.put\_line('Very Good');   WHEN 'C' THEN dbms\_output.put\_line('Good');   WHEN 'D' THEN dbms\_output.put\_line('Fair');   WHEN 'F' THEN dbms\_output.put\_line('Poor');   ELSE dbms\_output.put\_line('Error: No such grade'); END CASE |
|  | SELECT TRANS\_DATE,CASE  WHEN TRIM(TO\_CHAR(trans\_date,'DAY')) IN ('SATURDAY','SUNDAY') THEN 'weekend'  ELSE 'weekday'  END "Day Type"  FROM transactions; |
|  | [https://blogger.googleusercontent.com/img/a/AVvXsEg9X_j4hrddsjvTbCF7JkyXg1Ygb8ZI7829oLrJWznFI5R9Vf8NQ6ZsudbWmz6pW-LM7FywnLKJZKdSOZnBHaQp_sYJ9arg8PrS01wA_nBrod8nNLByj7YvhDB8mTvSxPS_5BAV9ig18qC9gKfCOnhKgIoffeyMzRKMzPweVhJ9UGrDjGkVhtfK4cEU=w572-h160](https://blogger.googleusercontent.com/img/a/AVvXsEg9X_j4hrddsjvTbCF7JkyXg1Ygb8ZI7829oLrJWznFI5R9Vf8NQ6ZsudbWmz6pW-LM7FywnLKJZKdSOZnBHaQp_sYJ9arg8PrS01wA_nBrod8nNLByj7YvhDB8mTvSxPS_5BAV9ig18qC9gKfCOnhKgIoffeyMzRKMzPweVhJ9UGrDjGkVhtfK4cEU) |
|  |  |
|  |  |

**3.**    **Cấu trúc lặp**

         **LOOP không định trước**

|  |  |
| --- | --- |
| x := 0;  y := 1000;  LOOP  x := x + 1;  y := y - x;  EXIT x > y;  END LOOP; |  |
|          **Fetch 1 dòng dữ liệu dau tien từ cursor**: (phải mở, đóng cursor)           SET SERVEROUTPUT ON  DECLARE  CURSOR emp\_cursor IS  SELECT employee\_id, last\_name FROM employees  WHERE department\_id =30;  empno employees.employee\_id%TYPE;  lname employees.last\_name%TYPE;  BEGIN  OPEN emp\_cursor;  FETCH emp\_cursor INTO empno, lname;  DBMS\_OUTPUT.PUT\_LINE(empno ||' '||lname);  CLOSE emp\_cursor;  END;  / | Con trỏ emp\_cursor trả về 6 employees có department\_id=30 nhưng câu lệnh FETCH sẽ lấy về chỉ 1 row, muốn lấy tất cả các row thì phải dùng vòng lặp (for) |
|          **Fetch lấy all row từ cursor:** LOOP...FETCH (phải mở, đóng cursor)  DECLARE  -- Khai báo cursor để truy vấn dữ liệu  CURSOR c\_Emp IS  SELECT \*  FROM emp  WHERE dept\_id = 10;  -- Khai báo biến cursor tương ứng để chứa dòng dữ liệu  v\_Emp c\_EMP%rowtype;   -- v\_Emp giống như kiểu record  BEGIN  -- Mở cursor  **OPEN** c\_Emp;  **LOOP**  -- Lấy dòng dữ liệu từ cursor  FETCH c\_Emp INTO v\_Emp;  -- Thoát khỏi vòng lặp nếu đã lấy hết dữ liệu trong cursor  EXIT WHEN c\_Emp%notfound;  -- Bổ sung dữ liệu vào Emp\_ext với dữ liệu lấy được từ cursor  INSERT INTO Emp\_ext (empno, ename, job) VALUES (v\_Emp.empno, v\_Emp.ename, v\_Emp.job);  **END LOOP**;  -- Đóng cursor  **CLOSE** c\_Emp;  END;  REM ===================================c  REM Grant on commit refresh system privilege to the owner of on  REM commit refresh materialized view which references tables  REM outside of the owner schema  REM ===================================  declare    owner varchar(30);    cursor c\_mv\_owner is       select distinct u.name          from sum$ s, sumdep$ d, obj$ o1, obj$ o2, user$ u          where s.obj# = d.sumobj# and                bitand(s.mflags, 65536) != 0 and                d.p\_obj# = o1.obj# and                s.obj# = o2.obj# and                o1.owner# != o2.owner# and                o2.owner# = u.user#;  begin     open c\_mv\_owner;     loop       fetch c\_mv\_owner into owner;       exit when c\_mv\_owner%NOTFOUND;       execute immediate 'GRANT ON COMMIT REFRESH TO ' ||                         dbms\_assert.enquote\_name(owner, FALSE);     end loop;     close c\_mv\_owner;  end;  / |  |

         **FOR-LOOP: Loop đã định trước**

|  |
| --- |
| declare      cursor c1 is select  distinct tablespace\_name  from dba\_data\_files  where   tablespace\_name like 'DATA200%' or tablespace\_name like 'INDX200%' order by tablespace\_name;  begin      for r1 in c1 loop          dbms\_output.put\_line('SQL '''||'alter tablespace ' || r1.tablespace\_name || ' read only'';');          dbms\_output.put\_line('SQL '''||'alter tablespace ' || r1.tablespace\_name || ' offline'';');      end loop;  end; |
| -- Switch all existing users to new temp tablespace. BEGIN   FOR cur\_user IN (SELECT username FROM dba\_users WHERE temporary\_tablespace = 'TEMP') LOOP     EXECUTE IMMEDIATE 'ALTER USER ' || cur\_user.username || ' TEMPORARY TABLESPACE temp2';   END LOOP; END;  *-- Không sử dụng CURSOR*  BEGIN  **FOR loop\_emp IN**  **(SELECT \***  **FROM**  **employees**  **WHERE salary>15000)**    LOOP      dbms\_output.put\_line('Number of rows processed: '||nvl(to\_char(sql%rowcount),'Null'));    END LOOP loop\_emp;  END;  *From <*[*http://www.dba-oracle.com/t\_adv\_plsql\_implicit\_cursor\_FOR\_loop.htm*](http://www.dba-oracle.com/t_adv_plsql_implicit_cursor_FOR_loop.htm)*>*   for r1 in (select trunc(completion\_time) completion\_time, round(sum(blocks\*block\_size)/1024/1024/1024,2) as archived\_log\_gb from V$ARCHIVED\_LOG                          where trunc(completion\_time) >= trunc(sysdate-7)                          --and trunc(completion\_time)>= to\_date(trunc(sysdate),'dd/mm/yyyy')                          and dest\_id=1                          group by trunc(completion\_time)                          order by trunc(completion\_time) desc) loop                    tAll:=tAll||'<tr>                              <td>'||r1.completion\_time||'</td>                              <td>'||r1.archived\_log\_gb||'</td>                           </tr>' ;             end loop;             tAll:=tAll|| |
| PROCEDURE           auto\_drop\_partition (v\_date DATE)  IS      date\_num1        INT := 40;                         -- Chi luu giu 40 ngay, PROM\_CHARGE\_DAILY\_HIS      date\_num2        INT := 70;                         -- Chi luu giu 40 ngay, LOG\_WS\_999      date\_num3        INT := 180;                        -- Chi luu giu 180 ngay, reg\_sms\_log      --date\_num4        INT := 300;                        --log dau noi, Dungnd      CURSOR c\_partition      IS          SELECT   table\_name, partition\_name            FROM   user\_tab\_partitions           WHERE   (sysdate - to\_date(SUBSTR(partition\_name,length(partition\_name)-7,8),'yyyy/mm/dd') > date\_num1                          AND table\_name IN                              ('PROM\_CHARGE\_DAILY\_HIS'))                         OR                          (sysdate - to\_date(SUBSTR(partition\_name,5,6),'yyyymm') > date\_num2                          AND table\_name IN                              ('LOG\_PROCESS'))                        OR                         (sysdate - to\_date(SUBSTR(partition\_name,length(partition\_name)-7,8),'yyyy/mm/dd') > date\_num3                          AND table\_name IN                              ('CP\_SUM\_LOG'))                           OR                          (sysdate - to\_date(SUBSTR(partition\_name,5,6),'yyyymm') > date\_num3                          AND table\_name IN                              ('REG\_SMS\_LOG'))                        OR                         (sysdate - to\_date(SUBSTR(partition\_name,length(partition\_name)-7,8),'yyyy/mm/dd') > 365                          AND table\_name IN                              ('LOG\_WS\_999'))                         OR                         (sysdate - to\_date(SUBSTR(partition\_name,length(partition\_name)-7,8),'yyyy/mm/dd') > 365                          AND table\_name IN                              ('SEND\_999\_LOG','SUB\_TRANSACTION\_LOG'))    ;        v\_sql\_command   VARCHAR2 (2400);  BEGIN      -- add partitions      FOR v\_data IN c\_partition      LOOP          BEGIN              EXECUTE IMMEDIATE   ' Alter table '                               || v\_data.table\_name                               || ' drop partition '                               || v\_data.partition\_name;          EXCEPTION              WHEN OTHERS              THEN                  DBMS\_OUTPUT.put\_line (SQLERRM);          END;      END LOOP;  END; |
| ***-- Chạy cả khi exception***  declare      cursor c1 is          SELECT a.INST\_ID,count(1) CountSession          FROM gv$session a, gv$process b          WHERE b.ADDR = a.paddr              AND a.inst\_id=b.inst\_id              and a.status='INACTIVE'              and A.USERNAME  in ('NEIF\_OWNER')              and prev\_exec\_start > sysdate - 5/24\*60              and type='USER'          group by a.inst\_id;      cursor c2 is          SELECT 'ALTER SYSTEM KILL SESSION '''||a.sid||','||a.serial#||',@'||a.inst\_id||''' immediate' sqltext, a.inst\_id          FROM gv$session a, gv$process b          WHERE b.ADDR = a.paddr              AND a.inst\_id=b.inst\_id              and a.status='INACTIVE'              and A.USERNAME  in ('USER1\_OWNER')              and prev\_exec\_start > sysdate - 5/24\*60              and type='USER' order by a.inst\_id;      vInst varchar2(20);      vSession number;  begin       for r1 in c1 loop           begin              if r1.CountSession > 100 then                  vInst:=r1.inst\_id;                  --dbms\_output.put\_line(vInst);                  for r2 in c2 loop                      begin                          if r2.inst\_id=vInst then                              dbms\_output.put\_Line(r2.sqltext);                              execute immediate r2.sqltext;                          end if;                      EXCEPTION                          WHEN OTHERS THEN                             dbms\_output.put\_Line('Error but CONTINUE');                             CONTINUE;                      end;                  end loop;              end if;           end;       end loop;         insert into dba\_tc\_logs (id, content) values (dba\_tc\_seq.nextval,'Complted to Kill NEIF\_OWNER INACTIVE sesssions after 5 minutes');       commit;  end;  ***-- Tạo dữ liệu test***  create table t (name1 varchar2(100), name2 varchar2(100));  alter table t rename to spct;  declare       t number;  begin       for t in 1..20000 loop            insert into spct values (Null,'a');            commit;       end loop;  end;  / |

         **WHILE-LOOP**

|  |
| --- |
| DECLARE  countryid locations.country\_id%TYPE := 'CA';  loc\_id locations.location\_id%TYPE;  new\_city locations.city%TYPE := 'Montreal';  counter NUMBER := 1;  BEGIN  SELECT MAX(location\_id) INTO loc\_id FROM locations  WHERE country\_id = countryid;  WHILE counter <= 3 LOOP  INSERT INTO locations(location\_id, city, country\_id)  VALUES((loc\_id + counter), new\_city, countryid);  counter := counter + 1;  END LOOP;  END;  / |

**Notes:**

         To exit a loop use the **EXIT WHEN** statement;

         To skip to the next iteration of the loop, use the **CONTINUE WHEN** statement ([Oracle 11g](http://www.orafaq.com/wiki/Oracle_11g) and above).

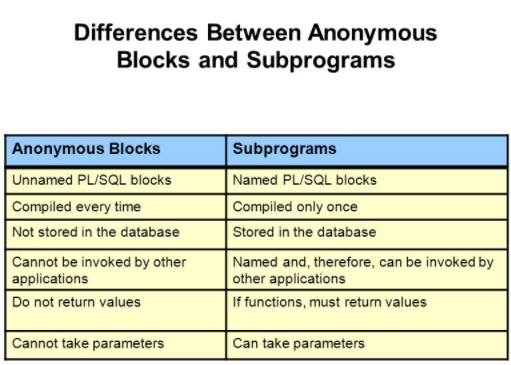
         **Ngoài ra:**

         **Goto:**

|  |
| --- |
| goto the\_end;  <<the\_end>> dbms\_output.put\_line ('The End.'); |

### Bài 4.PL/SQL cơ bản - Chương trình con

**1.**    **Sự khác nhau giữa  Anonymous Block và Subprograms**

**[](https://lh3.googleusercontent.com/-OWNKj6UkC8Y/YQ_6HgmnHPI/AAAAAAAABQI/A9qE6l_4n5UuIqTMo-S1qPj-ZPwUaxNlACLcBGAsYHQ/image.png)**

**2.**    **Anonymous Blocks (Oracle PL/SQL Blocks)**

*DECLARE  
   sql\_stmt    VARCHAR2(200);  
   plsql\_block VARCHAR2(500);  
   emp\_id      NUMBER(4) := 7566;  
   salary      NUMBER(7,2);  
   dept\_id     NUMBER(2) := 50;  
   dept\_name   VARCHAR2(14) := 'PERSONNEL';  
   location    VARCHAR2(13) := 'DALLAS';  
   emp\_rec     emp%ROWTYPE;  
BEGIN  
   EXECUTE IMMEDIATE 'CREATE TABLE bonus (id NUMBER, amt NUMBER)';  
   sql\_stmt := 'INSERT INTO dept VALUES (:1, :2, :3)';  
   EXECUTE IMMEDIATE sql\_stmt USING dept\_id, dept\_name, location;  
   sql\_stmt := 'SELECT \* FROM emp WHERE empno = :id';  
   EXECUTE IMMEDIATE sql\_stmt INTO emp\_rec USING emp\_id;  
   plsql\_block := 'BEGIN emp\_pkg.raise\_salary(:id, :amt); END;';  
   EXECUTE IMMEDIATE plsql\_block USING 7788, 500;  
   sql\_stmt := 'UPDATE emp SET sal = 2000 WHERE empno = :1  
      RETURNING sal INTO :2';  
   EXECUTE IMMEDIATE sql\_stmt USING emp\_id RETURNING INTO salary;  
   EXECUTE IMMEDIATE 'DELETE FROM dept WHERE deptno = :num'  
      USING dept\_id;  
   EXECUTE IMMEDIATE 'ALTER SESSION SET SQL\_TRACE TRUE';  
END;*

*declare*

*total number :=0;*

*begin*

*for rs in (select isdn,d\_tac,profile\_text,imsi   from user1.tab1 b where*

*file\_DATE >= trunc(p\_date)+1-- >=n+1*

*and  file\_DATE < trunc(p\_date)+2*

*AND d\_TAC = trunc(p\_date)*

*and (A\_CURSTA = 'ACTIF' OR (A\_CURSTA = 'INACT' AND d\_tin >  trunc(p\_date)))*

*and profile\_text not like 'TS%'*

*)*

*loop*

*insert into  user1.tab1\_tmp (isdn,d\_tac,profile\_text,imsi) values(rs.isdn,rs.d\_tac,rs.profile\_text,rs.imsi);*

*total :=total+1;*

*if(total = 100) then commit; total:=0; end if;*

*end loop;*

*commit;*

*EXCEPTION WHEN others  THEN*

*dbms\_output.put\_line('n\_dump\_data\_ptm\_tmp:'||SQLERRM);*

*end;*

**3.**    **Chương trình con (Subprograms)**

**3.1. Procedure:**

|  |
| --- |
| *PROCEDURE Transaction\_count\_alert*  *IS*  *CURSOR c1*  *IS*  *SELECT COUNT ( \* ) cnt, service\_code*  *FROM msg\_audit*  *WHERE issue\_datetime >= SYSDATE - 6 / 24 and response\_code = '00'*  *GROUP BY service\_code order by service\_code;*  *v1   c1%ROWTYPE;*  *msg varchar2(160);*  *BEGIN*  *msg:= 'Trans Count in last 6h:';*  *FOR v1 IN c1*  *LOOP*  *msg:= msg || 'Srv=' || v1.service\_code || ', Cnt =' || v1.cnt || ';';*  *END LOOP;*  *INSERT INTO THREAD\_SMTP\_QUEUE*  *(*  *batch\_id,*  *SOURCE,*  *message\_date,*  *MESSAGE*  *)*  *VALUES (*  *'2800',*  *'Transaction Count Alert',*  *SYSDATE,*  *msg*  *);*  *COMMIT;*  *END;* |

         **Gọi procedure**

|  |
| --- |
| *BEGIN*  *Transaction\_count\_alert;*  *END;*  */* |

**3.2. Function**

|  |
| --- |
| *CREATE FUNCTION check\_sal RETURN Boolean IS*  *dept\_id employees.department\_id%TYPE;*  *empno employees.employee\_id%TYPE;*  *sal employees.salary%TYPE;*  *avg\_sal employees.salary%TYPE;*  *BEGIN*  *empno:=205;*  *SELECT salary,department\_id INTO sal,dept\_id*  *FROM employees WHERE employee\_id= empno;*  *SELECT avg(salary) INTO avg\_sal FROM employees*  *WHERE department\_id=dept\_id;*  *IF sal > avg\_sal THEN*  *RETURN TRUE;*  *ELSE*  *RETURN FALSE;*  *END IF;*  *EXCEPTION*  *WHEN NO\_DATA\_FOUND THEN*  *RETURN NULL;*  *END;*  */* |

         **Truyền biến vào function:**

|  |
| --- |
| *DROP FUNCTION check\_sal;*  *CREATE FUNCTION check\_sal(empno employees.employee\_id%TYPE)*  *RETURN Boolean IS*  *dept\_id employees.department\_id%TYPE;*  *sal employees.salary%TYPE;*  *avg\_sal employees.salary%TYPE;*  *BEGIN*  *SELECT salary,department\_id INTO sal,dept\_id*  *FROM employees WHERE employee\_id=empno;*  *SELECT avg(salary) INTO avg\_sal FROM employees*  *WHERE department\_id=dept\_id;*  *IF sal > avg\_sal THEN*  *RETURN TRUE;*  *ELSE*  *RETURN FALSE;*  *END IF;*  *EXCEPTION ...*  *...* |

         **Gọi hàm:**

|  |
| --- |
| *BEGIN*  *DBMS\_OUTPUT.PUT\_LINE('Checking for employee with id 205');*  *IF (check\_sal(205) IS NULL) THEN*  *DBMS\_OUTPUT.PUT\_LINE('The function returned NULL due to exception');*  *ELSIF (check\_sal(205)) THEN*  *DBMS\_OUTPUT.PUT\_LINE('Salary > average');*  *ELSE*  *DBMS\_OUTPUT.PUT\_LINE('Salary < average');*  *END IF;*  *DBMS\_OUTPUT.PUT\_LINE('Checking for employee with id 70');*  *IF (check\_sal(70) IS NULL) THEN*  *DBMS\_OUTPUT.PUT\_LINE('The function returned NULL due to exception');*  *ELSIF (check\_sal(70)) THEN*  *...*  *END IF;*  *END;*  */* |

         **Trong TOAD, SQL Navigator chú ý thêm output để hiển thị nhiều hơn:**

*DECLARE  
   ... declarations ...  
BEGIN  
   DBMS\_OUTPUT.ENABLE (1000000);   // enable output voi 1000000 ky tu  
   ...  
END;*

         **Trong SQL\*Plus:**

*SQL> SET SERVEROUTPUT ON*

         **Hàm trong hàm:**

*DECLARE*

*outer\_variable VARCHAR2(20):='GLOBAL VARIABLE';*

*BEGIN*

*DECLARE*

*inner\_variable VARCHAR2(20):='LOCAL VARIABLE';*

*BEGIN*

*DBMS\_OUTPUT.PUT\_LINE(inner\_variable);*

*DBMS\_OUTPUT.PUT\_LINE(outer\_variable);*

*END;*

*DBMS\_OUTPUT.PUT\_LINE(outer\_variable);*

*END;*

*/*

**4. Packages**

|  |
| --- |
| *CREATE OR REPLACE PACKAGE test\_pkg AS*  *PROCEDURE add\_test (col1\_in NUMBER, col2\_in CHAR);*  *PROCEDURE del\_test (col1\_in NUMBER);*  *END test\_pkg;*  */*  *CREATE OR REPLACE PACKAGE BODY test\_pkg AS*  *PROCEDURE add\_test(col1\_in IN NUMBER,col2\_in IN CHAR) AS*  *BEGIN*  *INSERT INTO test VALUES (col1\_in,col2\_in);*  *END add\_test;*  *PROCEDURE del\_test(col1\_in IN NUMBER) AS*  *BEGIN*  *DELETE FROM test WHERE col1\_in = col1;*  *END del\_test;*  *END* test\_pkg;  / |

Gọi thủ tục trong package:

*Begin*

*Test\_pkg.add\_test(1, ‘col10’);*

*Test\_pkg.del\_test(1, ‘col10’);*

*End;*

### Bài 5.PL/SQL cơ bản - I/O dữ liệu

**1.**    **Console:**

DBMS\_OUTPUT.ENABLE (1000000);   // enable output voi 1000000 ky tu

dbms\_output.put\_line('Hi Binh, I can write PL/SQL');

**2.**    **Làm việc với File:**

|  |
| --- |
| --------------SYSDBA-----------------------  *SQL> CREATE DIRECTORY MYDIR AS 'C:\TESTLOC';*  Directory created.  *SQL> GRANT READ, WRITE ON DIRECTORY MYDIR TO SCOTT*  Grant succeeded.  *--------------SCOTT-------------------------*  *DECLARE*  *L\_HANDLER UTL\_FILE.FILE\_TYPE;*  *BEGIN*  *L\_HANDLER := UTL\_FILE.FOPEN('MYDIR', 'SYS.txt', 'W');*  *UTL\_FILE.PUTF(L\_HANDLER, 'UTL\_FILE write mode demonstration');*  *UTL\_FILE.FCLOSE(L\_HANDLER);*  *END;*  PL/SQL procedure successfully completed. |

**3.**    **Exception**

|  |
| --- |
| *SET SERVEROUTPUT ON*  ***Single Exception Handler for Multiple Exceptions***  *CREATE OR REPLACE PROCEDURE select\_item (   t\_column VARCHAR2,   t\_name   VARCHAR2 ) AUTHID DEFINER IS   temp VARCHAR2(30); BEGIN   temp := t\_column;  -- For error message if next SELECT fails    -- Fails if table t\_name does not have column t\_column:    SELECT COLUMN\_NAME INTO temp   FROM USER\_TAB\_COLS   WHERE TABLE\_NAME = UPPER(t\_name)   AND COLUMN\_NAME = UPPER(t\_column);    temp := t\_name;  -- For error message if next SELECT fails    -- Fails if there is no table named t\_name:    SELECT OBJECT\_NAME INTO temp   FROM USER\_OBJECTS   WHERE OBJECT\_NAME = UPPER(t\_name)   AND OBJECT\_TYPE = 'TABLE';  EXCEPTION****WHEN NO\_DATA\_FOUND THEN******DBMS\_OUTPUT.PUT\_LINE ('No Data found for SELECT on ' || temp);***  *WHEN others THEN*  *send\_sms\_binhtv('Error sys.dba\_op.auto\_drop\_partitions: ' || SQLERRM);*  *insert into binhtv.dbamf\_log\_jobs (id,name,status,event\_date, note)*  *values(binhtv.dbamf\_log\_jobs\_seq.nextval,'Error sys.dba\_ct.ct\_session',-1,sysdate,'1:OK');*  *commit;*  *DBMS\_OUTPUT.PUT\_LINE ('Unexpected error');*  *RAISE;*  *END;*  *END; /* |

## Tài liệu tham khảo:

<https://www.tranvanbinh.vn/p/sql-plsql-co-ban.html>

# LINUX

Tham khảo thêm: <https://www.tranvanbinh.vn/p/tong-hop-bai-viet-ve-system.html>

Bài viết này chứa một danh sách ngắn gọn các lệnh mà hầu hết các DBA sẽ thao tác trên OS LINUX, SOLARIS, AIX

**NỘI DUNG:**

* Thao tác với tập tin cơ bản
* Quyền file, thư mục
* Quản lý người dùng hệ điều hành
* Quản lý proces
* Tên người dùng và tên máy chủ
* Dòng lỗi trong tập tin
* Xóa các tập tin cũ
* Kiểm tra tệp tồn tại
* Xoay tệp nhật ký
* Tìm tập tin lớn
* Thực hiện hành động cho mọi tệp trong thư mục
* Thực hiện hành động cho mọi dòng trong một tệp
* Bí danh (alias)
* Xóa DOS CR / LFs (^ M)
* Chạy các lệnh như người dùng Oracle từ Toot
* Nén tập tin
* Giám sát hiệu suất
  + vmstat
  + free
  + iostat
  + top/topas
  + sar
  + mpstat
* Ẩn mật khẩu
* Tập lệnh khởi động tự động trên Linux
* CRON
* Cluster Wide CRON Jobs trên Tru64
* NFS Mount (Sun)
* NFS Mount (Tru64)
* Cài đặt profile hữu ích
* Tập tin hữu ích

THAO TÁC VỚI TẬP TIN CƠ BẢN

Lệnh "pwd" hiển thị thư mục hiện tại.

root> pwd

/u01/app/oracle/product/12.2.0.2.0

Lệnh "ls" liệt kê tất cả các tệp và thư mục trong thư mục được chỉ định. Nếu không có vị trí được xác định, nó hoạt động trên thư mục hiện tại.

root> ls

root> ls /u01

root> ls -al

Danh sách cờ "-a" bị ẩn "." các tập tin. Cờ "-l" liệt kê chi tiết tập tin.

Lệnh "cd" được sử dụng để thay đổi thư mục.

root> cd /u01/app/oracle

Lệnh "touch" được sử dụng để tạo một tệp trống mới với các quyền mặc định.

root> touch my.log

Lệnh "rm" được sử dụng để xóa các tập tin và thư mục.

root> rm my.log

root> rm -R /archive

Cờ "-R" cho biết lệnh lặp lại thông qua các thư mục con.

Lệnh "mv" được sử dụng để di chuyển hoặc đổi tên các tệp và thư mục.

root> mv [from] [to]

root> mv my.log my1.log

root> mv \* /archive

root> mv /archive/\* .

Các "." đại diện cho thư mục hiện tại.

Lệnh "cp" được sử dụng để sao chép các tập tin và thư mục.

root> cp [from] [to]

root> cp my.log my1.log

root> cp \* /archive

root> cp /archive/\* .

Lệnh "mkdir" được sử dụng để tạo các thư mục mới.

root> mkdir archive

Lệnh "rmdir" được sử dụng để xóa các thư mục.

root> rmdir archive

Lệnh "find" có thể được sử dụng để tìm vị trí của các tệp cụ thể.

root> find / -name dbmspool.sql

root> tìm / -print | grep -i dbmspool.sql

Cờ "/" đại diện cho thư mục nhìn chằm chằm cho tìm kiếm. Các ký tự đại diện như "dbms \*" có thể được sử dụng cho tên tệp.

Lệnh "which" có thể được sử dụng để tìm vị trí thực thi mà bạn đang sử dụng.

oracle> sqlplus

Lệnh "which" tìm kiếm cài đặt PATH của bạn cho các lần xuất hiện của tệp thực thi được chỉ định.

QUYỀN TẬP TIN

Quyền như sau

Owner Group World Permission

========= ========= ========= ======================

7 (u+rwx) 7 (g+rwx) 7 (o+rwx) read + write + execute

6 (u+rw) 6 (g+rw) 6 (o+rw) read + write

5 (u+rx) 5 (g+rx) 5 (o+rx) read + execute

4 (u+r) 4 (g+r) 4 (o+r) read only

2 (u+w) 2 (g+w) 2 (o+w) write only

1 (u+x) 1 (g+x) 1 (o+x) execute only

Lệnh hay dùng:

# chmod 777 \*.log -- Gán full quyền cho owner, group, group khác

# chmod 775 \*.log -- Gán full quyền cho owner, group của owner và quyền read+execute cho group khác

# chmod +x binh.sh -- Gán quyền thực thi file binh.sh

# chmod -R 775 /my/dir -- Gán full quyền các file, thư mục hiện tại, thư mục con cho owner, group của owner và quyền read+execute cho group khác

Lệnh "umask" có thể được sử dụng để đọc hoặc đặt quyền truy cập tệp mặc định cho người dùng hiện tại.

root> umask 022

Giá trị umask được trừ khỏi các quyền mặc định (666) để đưa ra quyền cuối cùng.

666: Quyền mặc định

022: - giá trị ô

644: sự cho phép cuối cùng

Lệnh "chmod" được sử dụng để thay đổi quyền của tệp sau khi tệp được tạo.

root> chmod 777 \* .log

Các eqivalents ký tự có thể được sử dụng trong lệnh chmod.

root> chmod o+rwx \*.log

root> chmod g+r \*.log

root> chmod -Rx \*.log

Lệnh "chown" được sử dụng để đặt lại quyền sở hữu các tệp sau khi tạo.

root> chown -R oinstall.dba \*

Cờ "-R"  là đệ quy (rescursive) khiến lệnh ăn vào các thư mục con nào.

QUẢN LÝ NGƯỜI DÙNG HỆ ĐIỀU HÀNH

Lệnh "useradd" được sử dụng để thêm người dùng hệ điều hành.

root> useradd -G oinstall -g dba -d /usr/users/my\_user -m -s /bin/ksh my\_user

* Cờ "-G" chỉ định nhóm chính.
* Cờ "-g" chỉ định nhóm phụ.
* Cờ "-d" chỉ định thư mục mặc định.
* Cờ "-m" tạo thư mục mặc định.
* Cờ "-s" chỉ định shell mặc định.

Lệnh "usermod" được sử dụng để sửa đổi cài đặt người dùng sau khi người dùng đã được tạo.

root> usermod -s /bin/csh my\_user

Lệnh "userdel" được sử dụng để xóa người dùng hiện tại.

root> userdel -r my\_user

Cờ "-r" xóa thư mục mặc định.

Lệnh "passwd" được sử dụng để đặt hoặc đặt lại mật khẩu đăng nhập của người dùng.

root> passwd my\_user

Lệnh "who" có thể được sử dụng để liệt kê tất cả người dùng có kết nối hệ điều hành.

root> who

root> who | head -5

root> who | tail -5

root> who | grep -i ora

root> who | wc -l

* Lệnh "head -5" giới hạn đầu ra trong 5 dòng đầu tiên của lệnh who.
* Lệnh "tail -5" giới hạn đầu ra ở 5 dòng cuối cùng của lệnh who.
* Lệnh "grep -i ora" giới hạn đầu ra ở các dòng có chứa "ora".
* Lệnh "wc -l" trả về số lượng dòng từ "ai" và do đó số lượng người dùng được kết nối.

QUẢN LÝ TIẾN TRÌNH (PROCESS)

Lệnh "ps" liệt kê thông tin quy trình hiện tại.

# ps

# ps -ef | grep -i ora

# ps -ef | grep -i ora | grep -v grep

# ps -ef | grep -i [o] ra

Các quy trình cụ thể có thể bị giết (kill) bằng cách chỉ định id tiến trình trong lệnh kill.

# kill 12345

# kill -9 12345

Bạn có thể tiêu diệt (kill) nhiều tiến trình bằng một lệnh đơn bằng cách kết hợp "giết" với các lệnh "ps" và "awk".

# kill -9 `ps -ef | grep ora | awk '{print $2}'`

TÊN VÀ TÊN MÁY CHỦ (HOSTNAME)

Các lệnh "uname" và "hostname" có thể được sử dụng để lấy thông tin về máy chủ.

root> uname -a

OSF1 oradb01.lynx.co.uk V5.1 2650 alpha

root> uname -a | awk '{ print $2 }'

oradb01.lynx.co.uk

root> hostname

oradb01.lynx.co.uk

DÒNG LỖI TRONG TẬP TIN

Bạn có thể trả về các dòng lỗi trong một tệp bằng cách sử dụng.

root> cat alert\_LIN1.log | grep -i ORA-

Lệnh "grep -i ORA-" giới hạn đầu ra ở các dòng có chứa "ORA-". Cờ "-i" làm cho trường hợp so sánh không phân biệt chữ hóa thường. Một số dòng lỗi có thể được trả về bằng lệnh "wc". Điều này thường đưa ra số đếm từ, nhưng cờ "-l" thay đổi từ đó để đếm số dòng.

root> cat alert\_LIN1.log | grep -i ORA- | wc -l

## XÓA CÁC TẬP TIN CŨ

Các lệnh  find có thể được sử dụng để cung cấp một danh sách các tập tin vào lệnh rm hoặc lệnh "-delete" có thể được sử dụng trực tiếp.

find /backup/logs/ -name daily\_backup\* -mtime +21 -exec rm -f {} ;

find /backup/logs/daily\_backup\* -mtime +5 -exec rm -f {} \;

find /backup/logs/daily\_backup\* -mtime +5 -delete;

## KIỂM TRA TỆP TỒN TẠI

Shell Bash cho phép bạn kiểm tra sự hiện diện của một tệp bằng cách sử dụng so sánh "[-e filepath]". Trong đoạn script sau, nhật ký sao lưu được đổi tên nếu nó xuất hiện và các tệp cũ hơn 30 ngày bị xóa sẽ bị xóa.

#!/bin/bash

if [ -e /tmp/backup.log ]; then

DATE\_SUFFIX=`date +"%Y"-"%m"-"%d"`

mv /tmp/backup.log /tmp/backup-$DATE\_SUFFIX.log

fi

# Xóa các tệp nhật ký cũ.

find /tmp/backup\*.log -mtime +30 -delete;

Đây là một ví dụ về luân chuyển nhật ký, trong đó nhật ký mới nhất không bao gồm ngày trong tên của nó.

## XOAY (ROTATE) TỆP NHẬT KÝ

Xem phần trước cho một biến thể khác về luân chuyển nhật ký (log rotation).

Kịch bản sau đây cung cấp một ví dụ về cách quản lý vòng quay nhật ký bằng cách sử dụng trình bao Bash. Tệp nhật ký bao gồm ngày trong tên tệp. Các tập tin cũ hơn 30 ngày sẽ bị xóa.

#!/bin/bash

DATE\_SUFFIX=`date +"%Y"-"%m"-"%d"`

LOG\_FILE=/tmp/backup-$DATE\_SUFFIX.log

# Làm một cái gì đó cần đăng nhập.

echo "Send this to log" >> $LOG\_FILE 2>&1

# Xóa các tệp nhật ký cũ.

find /tmp/backup\*.log -mtime +30 -delete;

## TÌM TẬP TIN LỚN

Tìm 20 tệp lớn nhất đệ quy từ thư mục này.

$ find . -type f -print0 | xargs -0 du -h | sort -hr | head -20

## THỰC HIỆN HÀNH ĐỘNG CHO MỌI TỆP TRONG THƯ MỤC

Các tập lệnh sau đây cho thấy hai phương thức để thực hiện một hành động cho mỗi tệp trong một thư mục.

#!/bin/bash

for FILE in `ls /tmp/`; do

# Do something with the file name.

echo $FILE;

done

# Or this.

for FILE in $( ls /tmp/ ); do

echo $FILE

done

## THỰC HIỆN HÀNH ĐỘNG CHO MỌI DÒNG TRONG MỘT TỆP

Các tập lệnh sau đây cho thấy một phương thức để thực hiện một hành động cho mỗi dòng trong một tệp.

#!/bin/bash

while read LINE; do

# Do something with the line.

echo $LINE;

done < /tmp/myfile.txt

## BÍ DANH (ALIAS)

Bí danh là một phím tắt được đặt tên cho một lệnh dài hơn sử dụng định dạng sau.

alias name='command'

Ví dụ: nếu bạn yêu cầu quyền truy cập sudo cho một lệnh cụ thể, bạn có thể muốn bao gồm lệnh này dưới dạng bí danh để bạn không phải nhớ nhập nó.

alias myscript='sudo -u oracle /path/to/myscript'

## XÓA DOS CR / LFS (^ M)

Xóa các ký tự CR / LF (^ M) kiểu DOS khỏi các tệp UNIX bằng cách sử dụng.

sed -e 's/^M$//' filename > tempfile

Tempfile mới được tạo nên loại bỏ ký tự ^ M.

Nếu có sẵn, có lẽ tốt hơn để sử dụng các lệnh dos2unixvà unix2dos.

## CHẠY CÁC LỆNH NHƯ NGƯỜI DÙNG ORACLE TỪ GỐC

Các tập lệnh sau đây cho thấy cách một số lệnh có thể được chạy với tư cách là người dùng "oracle" người dùng "root".

#!/bin/ksh

su - oracle <<EOF

ORACLE\_SID=LIN1; export ORACLE\_SID

rman catalog=rman/rman@w2k1 target=/ cmdfile=my\_cmdfile log=my\_logfile append

EOF

Điều này thường là cần thiết khi các công việc CRON được chạy từ người dùng root chứ không phải người dùng orory.

## NÉN TẬP TIN

Tạo các file, thư mục:

mkdir -p /tmp/test-dir/subdir1

mkdir -p /tmp/test-dir/subdir2

mkdir -p /tmp/test-dir/subdir3

mkdir -p /tmp/extract-dir

touch /tmp/test-dir/subdir1/file1.txt

touch /tmp/test-dir/subdir1/file2.txt

touch /tmp/test-dir/subdir2/file3.txt

touch /tmp/test-dir/subdir2/file4.txt

touch /tmp/test-dir/subdir3/file5.txt

touch /tmp/test-dir/subdir3/file6.txt

## TAR

Nén

# cd /tmp

# tar -cvf archive1.tar test-dir

Kiểm tra:

# tar -tvf /tmp/archive1.tar

Giải nén:

# cd /tmp/extract-dir

# tar -xvf /tmp/archive1.tar

## GZIP

Để tạo ra file .gz, cần nén fiel tar

# cd /tmp

# tar -cvf archive3.tar test-dir

# gzip archive3.tar --File tạo ra archive3.tar.gz

Để cho kết quả lẹnh trên ó thể dùng 1 lệnh:

# cd /tmp

# tar -cvzf archive3.tar.gz test-dir --File tạo ra archive3.tar.gz

Giải nén ra file archive3.tar :

# gunzip archive3.tar.gz -- Ra file archive3.tar

Giải nén trực tiếp ra thư mục archive3:

# cd /tmp/extract-dir

# tar -xvzf /tmp/archive3.tar.gz

## ZIP

Nén:

# cd /tmp

# zip -r archive5.zip test-dir

Check:

# unzip -l archive5.zip

Giải nén:

# cd /tmp/extract-dir

# unzip /tmp/archive5.zip

## CPIO

Nén:

# cd /tmp

# find test-dir | cpio -ov > archive6.cpio

Check:

# cpio -t < /tmp/archive6.cpio

Giải nén:

# cd /tmp/extract-dir

# cpio -idmv < /tmp/archive6.cpio

## GIÁM SÁT HIỆU NĂNG (PERFORMANCE)

## VMSTAT

Báo cáo thống kê bộ nhớ ảo (swap, free RAM, si, so, CPU)

# vmstat 5 3

procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu------

r b swpd free buff cache si so bi bo in cs us sy id wa st

0 0 0 1060608 24372 739080 0 0 1334 63 1018 1571 14 11 66 10 0

0 0 0 995244 24392 799656 0 0 6302 160 1221 1962 10 10 62 18 0

0 0 0 992376 24400 799784 0 0 1 28 992 1886 3 2 95 0 0

#

Xem [vmstat man page](http://linux.die.net/man/8/vmstat).

## FREE

Báo cáo việc sử dụng bộ nhớ hiện tại. Dòng "- / + bộ đệm / bộ đệm:" đại diện cho bộ nhớ thực được sử dụng và bộ nhớ trống, bỏ qua bộ đệm của hệ thống tệp Linux.

# free

total used free shared buffers cached

Mem: 8178884 4669760 3509124 0 324056 1717756

-/+ buffers/cache: 2627948 5550936

Swap: 10289148 0 10289148

#

## IOSTAT

Báo cáo thống kê I/O.

# iostat

Linux 3.2.10-3.fc16.x86\_64 (maggie.localdomain) 03/19/2012 \_x86\_64\_(4 CPU)

avg-cpu: %user %nice %system %iowait %steal %idle

2.02 0.23 0.51 0.78 0.00 96.46

Device: tps kB\_read/s kB\_wrtn/s kB\_read kB\_wrtn

sda 9.23 100.55 62.99 1796672 1125538

dm-0 13.60 100.31 62.99 1792386 1125524

dm-1 0.02 0.08 0.00 1432 0

# iostat 5 100 --5s chạy 1 lần lấy 100 mẫu

## TOP

Hiển thị các proces chiếm tải, CPU sử dụng, Chú ý cột 0 %id (là free), Mem sử dụng, Swap sử dụng

# top

top - 13:58:17 up 2 min, 1 user, load average: 2.54, 1.11, 0.41

Tasks: 160 total, 6 running, 154 sleeping, 0 stopped, 0 zombie

Cpu(s): 77.1%us, 22.6%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.3%hi, 0.0%si, 0.0%st

Mem: 2058872k total, 879072k used, 1179800k free, 23580k buffers

Swap: 4095992k total, 0k used, 4095992k free, 620116k cached

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND

2882 oracle 20 0 610m 64m 56m R 24.9 3.2 0:02.20 oracle

2927 root 20 0 90328 3832 2604 R 24.6 0.2 0:00.89 Xorg

2931 oracle 20 0 605m 34m 31m R 11.5 1.7 0:00.35 oracle

2933 oracle 20 0 605m 34m 30m S 9.8 1.7 0:00.30 oracle

2888 oracle 20 0 614m 52m 40m S 6.9 2.6 0:00.78 oracle

2935 oracle 20 0 604m 22m 20m S 6.2 1.1 0:00.19 oracle

2937 oracle 20 0 604m 19m 17m R 4.6 1.0 0:00.14 oracle

2688 oracle -2 0 603m 15m 13m S 4.3 0.8 0:01.08 oracle

2685 oracle 20 0 603m 15m 13m S 0.7 0.8 0:00.22 oracle

2939 oracle 20 0 217m 4084 3504 R 0.7 0.2 0:00.02 oracle

2698 oracle 20 0 604m 18m 16m S 0.3 0.9 0:00.17 oracle

2702 oracle 20 0 609m 22m 14m S 0.3 1.1 0:00.17 oracle

2704 oracle 20 0 618m 21m 19m S 0.3 1.1 0:00.21 oracle

2714 oracle 20 0 603m 20m 18m S 0.3 1.0 0:00.18 oracle

1 root 20 0 10364 704 588 S 0.0 0.0 0:00.36 init

2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd

3 root RT 0 0 0 0 S 0.0 0.0 0:00.00 migration/0

4 root 20 0 0 0 0 S 0.0 0.0 0:00.00 ksoftirqd/0

5 root RT 0 0 0 0 S 0.0 0.0 0:00.00 watchdog/0

6 root 20 0 0 0 0 S 0.0 0.0 0:00.03 events/0

7 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuset

8 root 20 0 0 0 0 S 0.0 0.0 0:00.00 khelper

9 root 20 0 0 0 0 S 0.0 0.0 0:00.00 netns

#

Các **PID** cột sau đó có thể được kết hợp với các cột  **SPID** trên view v$proces để cung cấp thêm thông tin về proces này.

SELECT a.username,

a.osuser,

a.program,

spid,

sid,

a.serial#

FROM v$session a,

v$process b

WHERE a.paddr = b.addr

AND spid = '&pid';

Xem [op man page](http://linux.die.net/man/1/top).

-- IDLE 99.6%

# top hoặc topas

top - 17:14:40 up 6:49, 1 user, load average: 0.00, 0.00, 0.00

Tasks: 104 total, 1 running, 103 sleeping, 0 stopped, 0 zombie

Cpu(s): 0.1%us, 0.1%sy, 0.0%ni, **99.6%id**, 0.2%wa, 0.0%hi, 0.0%si, 0.0%st

Mem: 2050836k total, 459132k used, 1591704k free, 44956k buffers

Swap: 4128760k total, 0k used, 4128760k free, 163880k cached

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND

3414 root 20 0 15084 1088 824 R 2.0 0.1 0:00.01 top

1 root 20 0 19396 1500 1192 S 0.0 0.1 0:00.71 init

2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd

3 root RT 0 0 0 0 S 0.0 0.0 0:00.00 migration/0

...

## SAR

Trên hệ thống Linux, [sar](http://linux.die.net/man/1/sar) (System Activity Reporter) có lẽ là một trong những công cụ đơn giản và linh hoạt nhất để sử dụng hệ thống báo cáo bao gồm CPU, bộ nhớ, đĩa và hoạt động mạng. Nó tự động thu thập số liệu thống kê hoạt động hệ thống khi được cài đặt bằng lệnh sau.

# yum install sysstat

Các sarcú pháp lệnh có dạng như sau.

# sar [options] [interval [count]]

Các tham số "tùy chọn" xác định những gì được báo cáo, sẽ được thảo luận sau. Tham số "khoảng" cho biết khoảng thời gian tính bằng giây giữa các mẫu. Tham số "đếm" cho biết số lượng mẫu sẽ được lấy trước khi lệnh kết thúc. Nếu "tính" bị bỏ qua, việc lấy mẫu sẽ tiếp tục vô thời hạn. Nếu cả "khoảng" và "đếm" bị bỏ qua, lệnh sẽ báo cáo các giá trị từ các mẫu 10 phút được lấy kể từ khi máy được khởi động lại lần cuối.

Như đã thấy trong trang [sar man page](http://linux.die.net/man/1/sar), có rất nhiều tùy chọn có sẵn, nhưng một số điểm bắt đầu bạn có thể thấy thú vị bao gồm:

* CPU:  
  + CPU cơ bản: sar [-u] [distance [Count]]
  + Tải trung bình: sar -q [khoảng [đếm]]
* sar:  
  + Phân trang hạt nhân: sar -B [distance [Count]]
  + Bộ nhớ không sử dụng: sar -r [distance [Count]]
  + Hoán đổi không gian: sar -S [khoảng [đếm]]
* Đĩa:  
  + I / O đĩa trung bình: sar -b [distance [Count]]
  + Đĩa I / O: sar -dp [distance [Count]]
* Mạng:  
  + Mạng: sar -n DEV [khoảng [đếm]]
  + Lỗi mạng: sar -n EDEV [khoảng [đếm]]

Dưới đây là một ví dụ về đầu ra từ một báo cáo CPU.

# sar -u 1 5

Linux 2.6.32-100.0.19.el5 (ol5-112.localdomain) 06/27/2011

03:10:07 PM CPU %user %nice %system %iowait %steal %idle

03:10:08 PM all 0.00 1.01 23.23 75.76 0.00 0.00

03:10:09 PM all 0.00 1.02 35.71 63.27 0.00 0.00

03:10:10 PM all 0.98 3.92 35.29 59.80 0.00 0.00

03:10:11 PM all 0.00 1.03 29.90 69.07 0.00 0.00

03:10:12 PM all 0.00 2.00 35.00 63.00 0.00 0.00

Average: all 0.20 1.81 31.85 66.13 0.00 0.00

#

## MPSTAT

Báo cáo bộ xử lý thống kê liên quan.

# mpstat 10 2

Linux 2.6.32-100.0.19.el5 (ol5-112.localdomain) 06/27/2011

01:59:57 PM CPU %user %nice %sys %iowait %irq %soft %steal %idle intr/s

02:00:07 PM all 1.21 0.00 0.90 0.20 0.00 0.00 0.00 97.69 980.50

02:00:17 PM all 0.70 0.00 0.40 0.00 0.00 0.10 0.00 98.79 973.77

Average: all 0.95 0.00 0.65 0.10 0.00 0.05 0.00 98.24 977.14

#

Xem  [mpstat man page](http://linux.die.net/man/1/mpstat).

## ẨN MẬT KHẨU

Bạn có thể được yêu cầu sử dụng mật khẩu trong các tập lệnh gọi các công cụ của Oracle, như SQL \* Plus, Xuất / Nhập và RMAN, v.v. Một phương pháp để xóa thông tin đăng nhập khỏi chính tập lệnh là tạo tệp thông tin xác thực để giữ chúng. Trong trường hợp này, tôi đang sử dụng "/home/oracle/.scottcred", có chứa phần sau đây.

scott/tiger

Thay đổi quyền để đảm bảo tệp chỉ hiển thị cho chủ sở hữu.

$ chmod 600 /home/oracle/.scottcred

Bây giờ thay thế các tham chiếu đến thông tin đăng nhập bằng nội dung của tệp.

$ expdp </home/oracle/.scottcred schemas = thư mục SCOTT = DATA\_PUMP\_DIR dumpfile = SCOTT.dmp logfile = expdpSCOTT.log

## TẬP LỆNH KHỞI ĐỘNG TỰ ĐỘNG TRÊN LINUX

CRON

Có hai phương pháp chỉnh sửa tập tin crontab. Trước tiên, bạn có thể sử dụng tùy chọn "crontab -l> filename" để liệt kê nội dung và dẫn nội dung này vào một tệp. Khi bạn đã chỉnh sửa tệp, bạn có thể áp dụng tệp đó bằng "tên tệp crontab".

* Đăng nhập bằng root
* crontab -l> newcron
* Chỉnh sửa tập tin newcron.
* crontab newcron

Ngoài ra, bạn có thể sử dụng tùy chọn "crontab -e" để chỉnh sửa trực tiếp tệp crontab.

Các mục có các yếu tố sau.

field allowed values

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

minute 0-59

hour 0-23

day of month 1-31

month 1-12

day of week 0-7 (both 0 and 7 are Sunday)

user Valid OS user

command Valid command or script.

5 trường đầu tiên có thể được chỉ định bằng cách sử dụng các quy tắc sau.

\* - Tất cả các giá trị khả dụng hoặc "đầu tiên cuối cùng".

3-4 - Một phạm vi duy nhất đại diện cho mỗi có thể từ đầu đến cuối phạm vi bao gồm.

1,2,5,6 - Một danh sách cụ thể của các giá trị.

1-3,5-8 - Một danh sách cụ thể của phạm vi.

0-23/2 - Mọi giá trị khác trong phạm vi được chỉ định.

Mục sau đây chạy một kịch bản dọn dẹp vào 01:00 mỗi Chủ nhật. Bất kỳ đầu ra hoặc lỗi từ tập lệnh đều được chuyển đến / dev / null để ngăn chặn sự tích tụ của thư đến root.

# crontab -e

0 1 \* \* 0 /u01/app/oracle/dba/weekly\_cleanup > /dev/null 2>&1

# crontab -l

## JOB CRON CỤM TRÊN TRU64

Trên hệ thống cụm cron là nốt cụ thể. Nếu bạn cần một công việc để bắn một lần trên mỗi cụm, thay vì một lần cho mỗi nút, bạn cần một cách tiếp cận khác cho công việc định kỳ tiêu chuẩn. Một cách tiếp cận được đưa ra trong tài liệu thực hành tốt nhất của HP ( Sử dụng cron trong Cụm máy chủ TruCluster ), nhưng theo tôi, một giải pháp thanh lịch hơn được đề xuất bởi Jason Orendorf của Nhóm doanh nghiệp HP Tru64 Unix ( TruCluster Clustercron ).

Trong giải pháp của mình, Jason tạo ra một tệp có tên / bin / cronrun với các nội dung sau.

#!/bin/ksh

set -- $(/usr/sbin/cfsmgr -F raw /)

shift 12

[[ "$1" = "$(/bin/hostname -s)" ]] && exit 0

exit 1

Kịch bản lệnh này chỉ trả về TRUE (0) trên nút là CFS phục vụ cluster\_root.

Tất cả các công việc toàn cụm nên có một mục crontab trên mỗi nút của cụm như thế nào.

5 \* \* \* /bin/cronrun && /usr/local/bin/myjob

Mặc dù công việc cron bắn vào tất cả các nút, phần "/ bin / cronrun &&" của mục nhập ngăn tập lệnh chạy trên tất cả các nút trừ CFS hiện đang phục vụ cluster\_root.

## NFS MOUNT (SUN)

Các deamon sau đây phải được chạy để chia sẻ cho PC.

* /usr/lib/nfs/nfsd -a
* /usr/lib/nfs/mountd
* /opt/SUNWpcnfs/sbin/rpc.pcnfsd

Để xem danh sách các ổ đĩa được gắn nfs đã có loại.

exportfs

Đầu tiên, điểm gắn kết phải được chia sẻ để có thể nhìn thấy bằng máy từ xa.

share -F nfs -o ro /cdrom

Tiếp theo chia sẻ có thể được gắn trên một máy từ xa bằng cách sử dụng **root** .

mkdir /cdrom#1

mount -o ro myhost:/cdrom /cdrom#1

## NFS MOUNT (TRU64)

Trên máy chủ, nếu NFS hiện không được thiết lập, hãy làm như sau.

* Trình quản lý ứng dụng -> Quản trị hệ thống -> Cấu hình -> NFS
* Chọn tùy chọn "Cấu hình hệ thống làm máy chủ NFS".
* Chấp nhận mọi mặc định.

Tạo thư mục điểm gắn kết.

mkdir /u04/backup

Nối các mục sau vào tệp "/ etc / export".

/u04/backup

Hãy chắc chắn rằng các quyền chính xác được cấp trên thư mục.

chmod -R 777 /u04/backup

Trên máy khách, nếu NFS hiện không được thiết lập, hãy làm như sau.

* Trình quản lý ứng dụng -> Quản trị hệ thống -> Cấu hình -> NFS
* Chọn tùy chọn "Cấu hình hệ thống dưới dạng máy khách NFS".
* Chấp nhận mọi mặc định.

Tạo thư mục điểm gắn kết.

mkdir /backup

Nối một mục sau vào tệp "/ etc / fstab".

nfs-server-name:/u04/backup /backup nfs rw,bg,intr 0 0

Cuối cùng, gắn (mount) tập tin.

mount /backup

Tại thời điểm này, bạn có thể bắt đầu sử dụng điểm gắn kết từ máy khách của mình.

## NÚI SAMBA / CIFS (LINUX)

Xem cấu hình Linux Samba .

Tạo một thư mục để sử dụng cho điểm gắn kết.

# mkdir /host

Thêm dòng sau vào tệp "/ etc / fstab".

//192.168.0.4/public /host cifs rw,credentials=/root/.smbcred,uid=500,guid=500 0 0

Tạo một tệp có tên "/root/.smbcred" với các nội dung sau.

username=myuser

password=mypassword

Thay đổi quyền trên tệp thông tin đăng nhập.

# chmod 600 /root/.smbcred

Gắn kết (mount) chia sẻ.

# mount /host

## CẤU HÌNH PC XSTATION

Tải xuống CygWin setup.exe từ http://www.cygwin.com .

Cài đặt, đảm bảo chọn tất cả các gói tùy chọn X11R6 (hoặc XFree86 trong các phiên bản cũ hơn).

Nếu bạn cần quyền truy cập root, hãy thêm mục sau vào tệp / etc / securettys trên mỗi máy chủ.

<client-name>:0

Từ lệnh quảng bá trên PC làm như sau.

set PATH=PATH;c:cygwinbin;c:cygwinusrX11R6bin

XWin.exe :0 -query <server-name>

Môi trường X sẽ bắt đầu trong một cửa sổ mới.

Nhiều bản phân phối Linux không khởi động XDMCP theo mặc định. Để cho phép truy cập XDMCP từ Cygwin chỉnh sửa tệp "/etc/X11/gdm/gdm.conf". Trong phần "[xdmcp]", đặt "Bật = đúng".

Nếu bạn đang bắt đầu bất kỳ ứng dụng X nào trong phiên, bạn sẽ cần đặt biến môi trường HIỂN THỊ. Hãy nhớ rằng, bạn đang hoạt động như một XStation chứ không phải chính máy chủ, vì vậy biến này phải được đặt như sau.

DISPLAY=<client-name>:0.0; export DISPLAY

## XAUTH (MAGIC COOKIE)

Truy cập vào máy chủ X có thể bị hỏng khi sử dụng suvà sudora lệnh. Các xauthlệnh cung cấp một giải pháp này. Quá trình này bao gồm các giai đoạn sau:

* Kiểm tra số hiển thị hiện tại của bạn.
* Sử dụng xauth list để có được một danh sách các cookie ma thuật.
* Chuyển sang người dùng mới.
* Sử dụng xauth addđể đặt cookie ma thuật cho số hiển thị của bạn.

Một ví dụ về điều này được hiển thị dưới đây.

$ echo $DISPLAY

localhost:12.0

$ xauth list

ol6.localdomain/unix:12 MIT-MAGIC-COOKIE-1 be64852468ca3c334720b10bb3c4d3cb

$ sudo su oracle

$ xauth add ol6.localdomain/unix:12 MIT-MAGIC-COOKIE-1 be64852468ca3c334720b10bb3c4d3cb

Bây giờ bạn sẽ có thể truy cập máy chủ X, giống như bạn có thể trước khi chuyển đổi người dùng.

## CÀI ĐẶT PROFILE HỮU ÍCH

 /etc/profile: Là profile mức system ảnh hưởng đến toàn bộ các user

Mỗi user có profile riêng:

# ls -al /etc/skel

total 36

drwxr-xr-x. 4 root root 4096 Feb 25 14:11 .

drwxr-xr-x. 113 root root 12288 Mar 1 14:38 ..

-rw-r--r--. 1 root root 18 Mar 29 2011 .bash\_logout

-rw-r--r--. 1 root root 176 Mar 29 2011 .bash\_profile

-rw-r--r--. 1 root root 124 Mar 29 2011 .bashrc

drwxr-xr-x. 2 root root 4096 Nov 20 2010 .gnome2

drwxr-xr-x. 4 root root 4096 Feb 25 14:06 .mozilla

#

Trong đó file .bash\_profile" cho user oracle có dạng:

ORACLE\_SID=ORCL; export ORACLE\_SID

ORACLE\_BASE=/u01/app/oracle; export ORACLE\_BASE

ORACLE\_HOME=$ORACLE\_BASE/product/11.2.0/db\_1; export ORACLE\_HOME

PATH=$ORACLE\_HOME/bin:$PATH; export PATH

## TẬP TIN HỮU ÍCH

Dưới đây là một số tập tin có thể được sử dụng.

|  |  |
| --- | --- |
| **Con đường** | **Nội dung** |
| /etc/passwd | Thiết lập người dùng |
| /etc/group | Cài đặt nhóm cho người dùng. |
| /etc/host | Thông tin tra cứu tên máy chủ. |
| /etc/system | Thông số hạt nhân cho Solaris. |
| /etc/sysconfigtab | Các tham số kernel cho Tru64. |
| /etc/sysctl.conf | Các tham số kernel cho Linux. |

Để biết thêm thông tin xem:

* [Oracle Database Administrator's Reference 11g Release 2 (11.2) for Linux and UNIX-Based Operating Systems](http://docs.oracle.com/cd/E11882_01/server.112/e10839/toc.htm)

## Tài liệu tham khảo:

<https://www.tranvanbinh.vn/p/tong-hop-bai-viet-ve-system.html>

<https://www.tranvanbinh.vn/2021/02/30-lenh-quan-trong-trong-linux-kem-vi-du.html>

<https://www.tranvanbinh.vn/2022/02/nhung-lenh-linux-co-ban-ai-cung-can-biet.html>

<https://www.tranvanbinh.vn/2022/04/su-khac-nhau-ve-cau-lenh-cua-linux.html>

<https://www.tranvanbinh.vn/p/tong-hop-bai-viet-ve-system.html>

<https://www.tranvanbinh.vn/2021/02/30-lenh-quan-trong-trong-linux-kem-vi-du.html>

<https://www.tranvanbinh.vn/2022/02/nhung-lenh-linux-co-ban-ai-cung-can-biet.html>

<https://www.tranvanbinh.vn/2022/04/su-khac-nhau-ve-cau-lenh-cua-linux.html>