**ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**Khoa công nghệ thông tin**

**ĐỀ TÀI:**

**CÔNG NGHỆ ORACLE GOLDENGATE TRONG ĐỒNG BỘ DỮ LIỆU**

**MỘT CHIỀU TỪ NHIỀU SITE VỀ MỘT SITE**

**SV Thực hiện: Đặng Tiến Mạnh**

**Hà Nội, 02/2018**

Mục lục

[**Mở đầu** 3](#_Toc506473079)

[**Chương 1: Tổng quan Kiến trúc Oracle Database** 6](#_Toc506473080)

[**1.1.** **Giới thiệu Oracle Database** 6](#_Toc506473081)

[**1.2.** **Kiến trúc Oracle Database Server** 6](#_Toc506473082)

[**1.2.1.** **Kiến trúc vật lý** 8](#_Toc506473083)

[1.2.1.1. Datafiles 8](#_Toc506473084)

[1.2.1.2. Control Files 9](#_Toc506473085)

[1.2.1.3. Redo Log Files 9](#_Toc506473086)

[1.2.1.4. Archived Redo Log Files 10](#_Toc506473087)

[1.2.1.5. Parameter Files 10](#_Toc506473088)

[1.2.1.6. Password files. 10](#_Toc506473089)

[1.2.2. **Kiến trúc logic** 10](#_Toc506473090)

[1.2.2.1. Data blocks 11](#_Toc506473091)

[1.2.2.2. Extents 11](#_Toc506473092)

[1.2.2.3. Segments 11](#_Toc506473093)

[**1.2.3.** **Kiến trúc bộ nhớ** 12](#_Toc506473094)

[1.2.3.1.1. System Global Area(SGA) 12](#_Toc506473095)

[1.2.3.2. PGA 13](#_Toc506473096)

[**1.2.4.** **Kiến trúc proccess** 13](#_Toc506473097)

[1.2.4.1. Database Writer Process (DBWn) 14](#_Toc506473098)

[1.2.4.2. Log Writer Process (LGWR) 14](#_Toc506473099)

[1.2.4.3. Archiver Process (ARCn) 14](#_Toc506473100)

[1.2.4.4. Checkpoint Process (CKPT) 14](#_Toc506473101)

[1.2.4.5. System Monitor Process (SMON) 14](#_Toc506473102)

[1.2.4.6. Process Monitor Process (PMON) 15](#_Toc506473103)

[**Chương 3: Công nghệ Oracle GoldenGate** 16](#_Toc506473104)

[**3.1.** **Khái niệm Oracle Golden Gate** 16](#_Toc506473105)

[**3.2.** **Lợi ích của Oracle GoldenGate** 16](#_Toc506473106)

[**3.3.** **Kiến trúc của Oracle GoldenGate** 18](#_Toc506473107)

[3.3.1. Manager 18](#_Toc506473108)

[3.3.2. Extract 18](#_Toc506473109)

[3.3.3. Data Pump 18](#_Toc506473110)

[3.3.4. Replicat 19](#_Toc506473111)

[3.3.5. Collector 19](#_Toc506473112)

[3.3.6. Trail files 19](#_Toc506473113)

[**Chương 4: Mô hình đồng bộ dữ liệu từ nhiều site về một site.** 20](#_Toc506473114)

[**Chương 5: Xây dựng mô hình đồng bộ dữ liệu điểm thi.** 21](#_Toc506473115)

[**1.** **Mô hình cơ sở dữ liệu trên server nguồn và đích** 21](#_Toc506473116)

[**2.** **Mô hình cài đặt** 21](#_Toc506473117)

[**3.** **Các bước thực hiện cầu hình đồng bộ dữ liệu** 22](#_Toc506473118)

[3.1. Yêu cầu 22](#_Toc506473119)

[3.2. Cài đặt 22](#_Toc506473120)

[**Phụ lục** 35](#_Toc506473121)

[Overview of Checkpoints 35](#_Toc506473122)

[About the Commit Sequence Number 35](#_Toc506473123)

[**Tài liệu tham khảo** 35](#_Toc506473124)

Danh mục hình ảnh

[Hình 1. Kiến trúc Oralce Database Server 8](#_Toc506390568)

[Hình 2. Kiến trúc Oracle Instance 9](#_Toc506390569)

[Hình 3. Kiến trúc vật lý Oracle Database 10](#_Toc506390570)

[Hình 4. Kiến trúc logic Oracle Database 12](#_Toc506390571)

[Hình 5 Các mô hình đồng bộ cơ bản Oracle GoldenGate 16](#_Toc506390572)

[Hình 6. Kiến trúc logic Oracle GoldenGate 17](#_Toc506390573)

[Hình 7. Mô hình dồng bộ điểm thi 21](#_Toc506390574)

# **Mở đầu**

1. **Lý do chọn đề tài**

Trong thời đại của xã hội thông tin và nền kinh tế trí thức, mọi hoạt động của các tổ chức muốn đạt hiệu quả cao, giành được thắng lợi trong thế cạnh tranh gay gắt thì nhất thiết phải có những phương pháp để có được những thông tin, tri thức cần thiết một cách nhanh và chính xác.

Yêu cầu có được thông tin nhanh chóng, chính xác phục vụ cho công việc không dễ gì có được bởi vì dữ liệu ngày một nhiều, lưu trữ phân tán ở nhiều nơi (phù hợp với tổ chức phân cấp của các đơn vị), ở nhiều dạng không tương thích với nhau..

Những năm gần đây, hệ cơ sở dữ liệu phân tán được phát triển mạnh trong thực tế dựa trên sự phát triển của cơ sở dữ liệu(CSDL) và mạng máy tính. Vì yêu cầu chung của các tổ chức kinh tế xã hội và vấn đề làm sao nắm bắt thông tin nhanh nhất để tổ chức công việc hiệu quả nhất trong khi các trụ sở của những tổ chức này là ở những vị trí địa lý khác nhau cho nên xây dựng một hệ thống thông tin làm việc trên cở sở dữ liệu phân tán là phù hợp với xu hướng hiện nay. CSDL phân tán ra đời dựa trên nền tảng chính của CSDL tập chung nhưng với đặc điểm riêng biệt là thiết kế theo phương pháp phân mảnh dữ liệu và sử dụng các bản sao, dữ liệu phân tán sẽ được cập nhật và lưu tại các nút (site), mỗi nút tương ứng với một database server và chỉ thực hiện một số chức năng nhất định. Việc ứng dụng CSDL phân tán có thể khắc phục được các nhược điểm vốn có của CSDL tập chung. Ngoài ra, việc kết nối giữa các CSDL từ xa được thực hiện qua kết nối database link sẽ đáp ứng được một số tiêu chuẩn cơ bản như cải thiện năng suất thực hiện công việc, đảm bảo an toàn dữ liệu, giảm giá thành truyền thống, bảo mật cao, dễ dàng thay đổi, phát triển, …

Việc giải quyết bài toán đồng bộ trược đây thường sử dụng giải pháp trích xuất và nhập liệu một cách thủ công hoặc theo một số mô đun chuyên biệt phát triển trong phần mềm hay có thể sử dụng công cụ có sẵn của Oracle như Import/Export. Tuy nhiên, với giải pháp này thường gây tổn thương đến CSDL được đồng bộ. Hiện nay có rất nhiều giải pháp đồng bộ CSDL khác nhau được phát triển từ các hãng phần mềm nổi tiếng nhứ: Streams, Advanced Replication, Oracle GoldenGate,… mỗi loại đều có những ưu nhược điểm riêng. Trong phạm vi đề tài này sẽ giới thiệu kỹ thuật đồng bộ CSDL Oracle GoldenGate của Oracle

1. **Mục tiêu của đề tài**

Nghiên cứu kiến trúc Oracle Database và công nghệ Oracle GoldenGate trong đồng bộ và trên cơ sở lý thuyết đó xây dựng mô hình đồng bộ dữ liệu điểm thi THPT quốc gia cụ thể là: Đồng bộ một chiều dữ liệu điểm thi ở 2 cơ sở khác nhau đến một hệ thống chung.

1. **Đối tượng và phạm vi nghiên cứu**

Đối tượng nghiên cứu: Oralce Database, Oracle GoldenGate

Phạm vi nghiên cứu: ….

1. **Phương pháp nghiên cứu**

Nghiên cứu trên hệ quản trị cơ sở dữ liệu Oralce (Các sách đã xuất bản, tài liệu trên mạng).

Nghiên cứu giải pháp đồng bộ dữ liệu sử dụng Oralce GoldenGate.

1. **Ý nghĩa của đề tài**

Đề tài giúp hiểu rõ các lý thuyết về cơ sở dữ liệu, có thể giúp người đọc nắm được các thành phần, Kiến trúc của hệ quản trị cơ sở dữ liệu. Ngoài ra, đề tài phần giúp người đọc hiểu thành phần, cơ chế hoạt động của giải pháp đồng bộ dữ liệu Oracle GoldenGate từ đó xây dựng được một mô hình đồng bộ dữ liệu.

1. **Bố cục của báo cáo.**

Nội dung chính của đề tài gồm 4 chương

Chương 1: Tổng quan Kiến trúc Oracle Database

Chương 2: Công nghệ Oralce GoldenGate

Chương 3: Mô hình đồng bộ dữ liệu một chiều từ nhiều site về một site

Chương 4: Xây dựng mô hình đồng bộ điểm thi

**Phụ lục**

**Tài liệu tham khảo**

# **Chương 1: Tổng quan Kiến trúc Oracle Database**

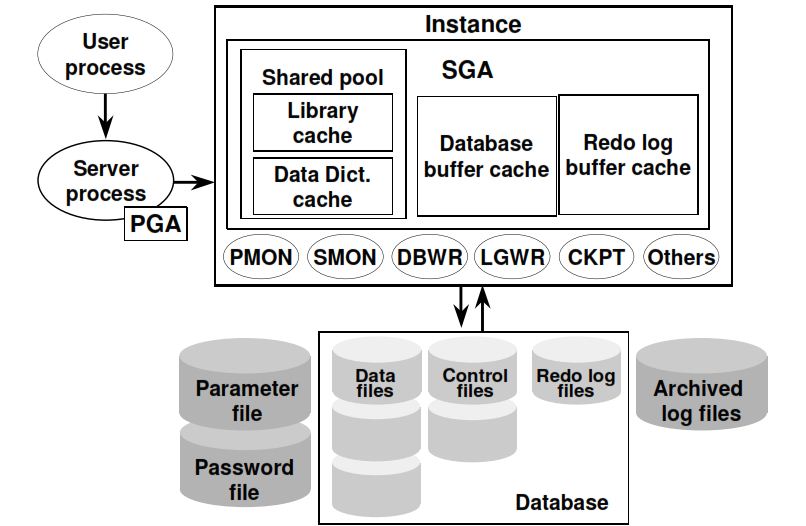
## **Giới thiệu Oracle Database**

Oracle Database là một hệ quản trị cơ sở dữ liệu, được phát triển và phân phối bởi tập đoàn Oracle.

Ở Việt nam, Oracle được sử dụng phổ biến ở các doanh nghiệp viễn thông như Mobiphone, Vinaphone, Viettel,… trong khối ngân hàng như Vietcombank, Techcombank, … hay trong khối nhà nước như Tổng cục Thuế, Tổng cục Hải Quan, ….

## **Kiến trúc Oracle Database Server**

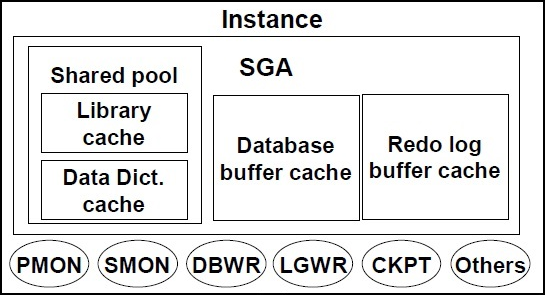
Kiến trúc Oracle Database Server là một hệ thống quản trị cơ sở dữ liệu cho phép quản lý thông tin một cách toàn diện. Oracle Database Server bao gồm hai thành phần chính: Oracle Instance, Oracle database.



Hình 1. Kiến trúc Oralce Database Server

* Oracle Instance

Oracle Instance là tập hợp Kiến trúc bộ nhớ và các tiến trình nền để quản lí các file dữ liệu của Oracle database.



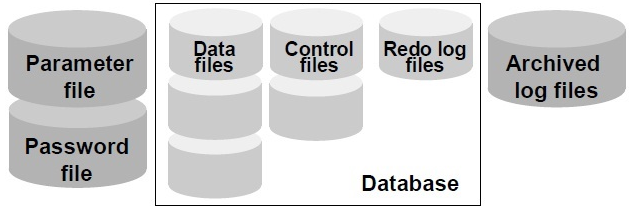
Hình 2. Kiến trúc Oracle Instance

* Oracle Database

Oracle Database dùng để lưu trữ và cung cấp thông tin cho người dùng. Việc quản lý dữ liêu trong Oracle Database thông qua Kiến trúc lưu trữ logic và vật lý.

### **Kiến trúc vật lý**

Kiến trúc vật lý của Oracle bao gồm tập hợp các tập tin: datafiles, control files, redo log files, archived redo log files, parameter files, password files, …



Hình 3. Kiến trúc vật lý Oracle Database

* + - 1. Datafiles

Datafiles là các file vật lý được lưu trữ trên ổ đĩa. Những file này chứa tất cả dữ liệu của database. Các dữ liệu thuộc Kiến trúc logic của database như tables, indexes … được lưu trữ dưới dạng vật lý trong các datafiles của database.

Tính chất của datafiles:

* Một datafile chỉ có thể liên kết với một database.
* Datafile có thể tự động tăng kích thước mỗi khi database sử dụng hết dung lượng lưu trữ.
* Một hoặc nhiều datafile tạo thành một đơn vị lưu trữ logic của Oracle được gọi là tablespace
* Một datafile chỉ thuộc duy nhất một tablespace
  + - 1. Control Files

Control files là tập tin quan trọng của Oracle lưu trữ các thông tin về Kiến trúc vật lý của cơ sở dữ liệu như:

* Tên cơ sở dữ liệu
* Thời gian tạo
* Tên và địa chỉ các datafile và redolog files
* Thông tin về tablespace

Khi một instance của Oracle database được mở, control file được sử dụng để xác định datafiles và redo log files. Khi thêm mới hay sử đổi datafile, redo log file … thì control file được tự động thay đổi tương ứng.

Ngoài ra, Controlfile cũng được sử dụng trong việc khôi phục dữ liệu.

* + - 1. Redo Log Files

Mỗi Oracle database đều có tối thiểu 2 redo log files. Tập hợp các redo log files được gọi là redo log của database. Redo log có nhiệm vụ ghi lại tất cả thay đổi dữ liệu trong database. Redo log được sử dụng để bảo vệ database trong trường hợp có sự cố.

Oracle cho phép sử dụng cùng một lúc nhiều redo log để cùng lưu trữ các bản sao của redo log trên các ổ đĩa khác nhau. Các thông tin trong redo log được sử dụng để khôi phục dữ liệu đã mất trong trường hợp hệ thống gặp sự cố khi ghi dữ liệu lên datafiles. (ví dụ trong trường hợp mất điện bất chợt khiến dữ liệu trong bộ nhớ đệm chưa được ghi vào datafile và gây ra mất dữ liệu).

Log switch là thời điểm mà database dừng ghi vào một redo log file và bắt đầu ghi vào redo log file khác. Thông thường, một log switch xảy ra khi redo log file hiện tại đầy và phải ghi vào file kế tiếp.

* + - 1. Archived Redo Log Files

Oracle có thể lưu trữ thông tin trong redo log file tới một hoặc nhiều nguồn offile được gọi là archived log (trong trường hợp database chạy ở chế độ archivelog).

Archived redo log files là một bản copy của một hoặc nhiều redo log files. Khi redo log files đã đầy hoặc khi xảy ra log switch thì dữ liệu từ redo log files sẽ được ghi ra archive log files Các archived log files được dùng với các bản backup sẽ đảm bảo có thể khôi phục lại hệ thống đến một thời điểm xác định.

* + - 1. Parameter Files

Là những tập tin chứa danh sách tham số cấu hình của Oracle, như kích thước bộ nhớ, vị trí các control files, redo log files, … được dùng khi khởi động database.

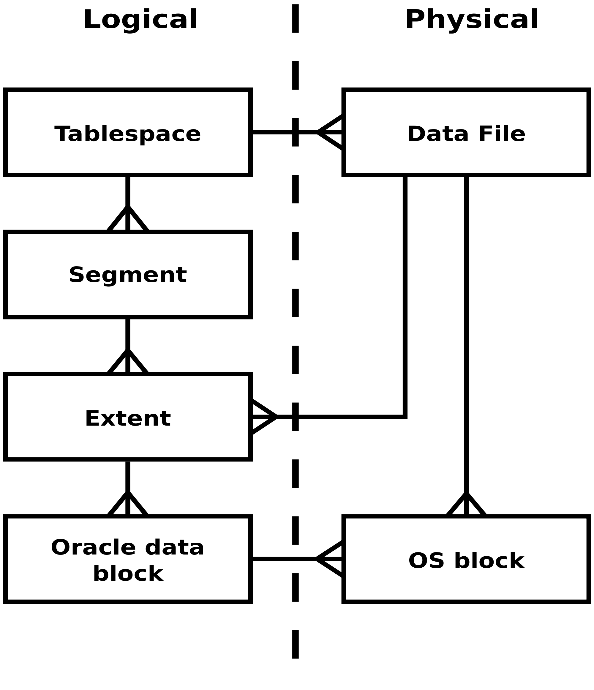
Có hai loại file tham số là pfile và spfile

* pfile: là file text có tên là init{SID}.ora
* spfile: là file nhị phân có tên là spfile{SID}.ora
  + - 1. Password files.

Password files được sử dụng để xác định quyền của từng user trong database. Cho phép người sử dụng có thể khởi động hoặc tắt một Oracle Instance.

* + 1. **Kiến trúc logic**

Kiến trúc lưu trữ logic bao gồm: tablespaces, data blocks, extents và segments



Hình 4. Kiến trúc logic Oracle Database

* + - 1. Data blocks

Oracle database quản lý không gian lưu trữ trong datafiles bằng các đơn vị được gọi là data blocks. Data block là **cấp độ lưu trữ logic nhỏ nhất** trong Oracle database. Thường một Oracle data block sẽ **gồm nhiều OS block** để tăng I/O..

Kích thước block được chỉ định bởi tham số DB\_BLOCK\_SIZE. Kích thước data block trong Oracle là bội số của kích thước block trong hệ điều hành để tránh I/O không cần thiết.

* + - 1. Extents

Extent là mức độ lưu trữ cao hơn**, bao gồm nhiều data block liền kề** (về mặt logical, còn về mặt physical có thể là các block rải rác trên disk). Khi 1 object cần tăng thêm kích thước, Oracle sẽ cấp phát thêm ở mức độ extent.

* + - 1. Segments

Segment là đơn vị lưu trữ logic bao gồm nhiều exten**t**. Có 4 loại segment chính:

* **Data segment**: là segment tạo nên table. Mỗi table được lưu hoàn toàn trong 1 segment (với partitioned table thì mỗi partition là 1 segment)
* **Index segment**: là segment tạo nên index.  Mỗi index được lưu hoàn toàn trong 1 segment (với partitioned index thì mỗi partition là 1 segment)
* **Undo segment**: các segment lưu trữ thông tin undo trong undo tablespace
* **Temporary segment**: các segment lưu trữ thông tin tạm thời trong temporary tablespace
  + - 1. Tablespaces

Việc lưu trữ nhìn từ trong database, ở mức độ cao nhất sẽ là các tablespace.

Tablespace là tập hợp các data files. Có 3 loại tablespace lưu trữ:

* **Data tablespace**: tablespace chứa dữ liệu. Khi cài đặt thường có sẵn tablespace SYSTEM và SYSAUX, là các tablespace hệ thống chứa dữ liệu của Oracle, tablespace USERS để chứa dữ liệu của user/ứng dụng.
* **Undo tablespace**: là các tablespace chứa dữ liệu undo, liên quan đến hoạt động của database, do Oracle quản lý. Chỉ có 1 undo tablespace trong database
* **Temporary tablespace**: là các tablespace chứa dữ liệu tạm thời, liên quan đến hoạt động của database, cũng do Oracle quản lý. Có thể có nhiều temporary tablespace trong database
  + 1. **Kiến trúc bộ nhớ**
       - 1. System Global Area(SGA)

SGA là vùng nhớ chia sẻ chứa thông tin dữ liệu và thông tin điều khiển một Oracle Instance. Nếu nhiều người dùng cùng kết nối tới một instance, dữ liệu trong SGA được chia sẻ giữa các người dùng. Do đó, Đôi khi SGA còn được gọi là vùng nhớ chia sẻ.

SGA gồm các vùng nhớ chính:

* Database buffer cache

- Lưu trữ các dữ liệu được lấy ra từ datafiles

- Được quản lý thông qua thuật toán LRU **(Least Recently Used). Những dữ liệu trong buffer it được sử dụng sẽ được thay thế cho dữ liêu mới cần thiết.**

- Khi người dùng thự hiện câu truy vấn dữ liệu Oracle sẽ kiểm tra xem trong Database buffer cache có tồn tại dữ liệu cần lấy không ? Trong trường hợp các dữ liệu cần thiết không được tìm thấy. Oracle sẽ đọc dữ liệu từ datafiles và ghi vào Database buffer cache

* Redo log buffer

- Là phần bộ nhớ chứa những thay đổi trên CSDL, do các câu lệnh DML, DDL. Khi redo log buffer đầy 1/3 hoặc cứ sau mỗi 3 giây, Log write proccess sẽ ghì vào redo log files để lấy chỗ cho những mội dung mới.

- Do Redo log buffer chứa những thay đổi trên CSDL, nên để đảm bảo toàn vẹn dữ liệu, 1 transaction được coi là đã commit khi đã ghi những thay đổi trong Redo log buffer vào redo log files.

* Shared pool

- Là vùng nhớ lưu trữ diễn giải của câu lệnh SQL và PL/SQL. Điều này cho phép thực thì các câu lệnh SQL hoặc PL/SQL mà không tốn thêm tài nguyên.

* + - 1. PGA

Là phần bộ nhớ riêng cho mỗi server process (user session) hay background process.

* + 1. **Kiến trúc proccess**

Khi user mở một tool ứng dụng lên, chẳng hạn như SQL\*Plus, Toad… và kết nối đến database, ta gọi đó là **user process**. Phần process chúng ta sẽ tìm hiểu ở đây là các process chạy trên instance, bao gồm **server process** và các **background process.**

* + - 1. **Database Writer Process (DBWn)**

DBW có nhiệm vụ **ghi những dữ liệu đã thay đổi (dirty buffer) xuống disk.** Có thể cấu hình tối đa 20 process DBW (từ DBW0 -> DBW9 và DBWa -> DBWj).

Thường DBW sẽ ghi dữ liệu khi Database buffer cache gần đầy, thiếu chỗ trống, trong 1 số trường hợp như khi alter tablespace offline hoặc read only, hoặc ghi để gọi checkpoint. DBW sẽ ghi dữ liệu theo batch (ghi multiblock) để tăng hiệu năng

* + - 1. **Log Writer Process (LGWR)**

Log Writer Process chịu trách nhiệm **ghi dữ liệu từ redo log buffer cache vào redo log file**. LGWR là 1 trong những process hoạt động nhiều nhất, do phải ghi dữ liệu liên tục. LGWR ghi dữ liệu mỗi 3 giây, hoặc khi:

- Redo log buffer đầy 1/3

- User commit transaction, hoặc trước khí DBWn ghi dữ liệu đã thay đỏi vào disk(để đảm bảo không bị mất dữ liệu)

* + - 1. **Archiver Process (ARCn)**

Process này chỉ có nếu database chạy ở chế độ ARCHIVE LOG. Sau khi switch log, process này chịu trách nhiệm **ghi redo log files ra archive log files để lưu trữ**. Ngoài ra process này còn thực hiện chuyển dữ liệu qua hệ thống standby nếu có cấu hình.

* + - 1. **Checkpoint Process (CKPT)**

Process này **cập nhật checkpoint (thông tin về System Change Number (SCN)) trong control file và data header**. CKPT cập nhật checkpoint khi redo log đầy, hoặc trước khi DBW ghi dữ liệu, để đánh dấu vị trí cần khôi phục khi instance crash (đảm bảo tất cả dữ liệu trước SCN đó đã ghi vào disk, đánh dấu điểm thực hiện quá trình instance recovery).

* + - 1. **System Monitor Process (SMON)**

Khi hệ thống bị crash (vd cúp điện, 1 process khác chết làm instance crash) và được start lại, SMON sẽ thực hiện **recovery**, cập nhật dữ liệu từ redo log file vào datafile. Ngoài ra SMON còn dọn dẹp temporary segment trong các tablespace nếu không còn sử dụng nữa.

* + - 1. **Process Monitor Process (PMON)**

PMON làm nhiệm vụ**theo dõi các user process**. Nếu 1 user process không còn kết nối đến database nữa, nó sẽ rollback transaction của user nếu còn transaction dang dở, release lock của user nếu có, “thả” những block của user trong buffer cache ra, và xóa process ID khỏi danh sách active process.

PMON còn **theo dõi dispatcher và server process**, restart lại nếu chúng bị lỗi.

Ngoài ra, PMON **đăng ký thông tin dispatcher, instance với listener** để phục vụ cho việc nhận kết nối từ user.

# **Chương 3: Công nghệ Oracle GoldenGate**

## **3.1. Khái niệm Oracle Golden Gate**

Oracle GoldenGate là một giải pháp phần mềm cho phép đồng bộ và tích hợp dữ liệu theo thời gian thực, đảm bảo tính sẵn sàng cao đối với các ứng dụng quan trọng.

Oracle GoldenGate cung cấp khả năng đồng bộ và phân phối dữ liệu giữa các môi trường cơ sở dữ liệu, hệ điều hành khác nhau một cách ổn đinh, nhanh chóng và chính xác.

* 1. **Lợi ích của Oracle GoldenGate**

Những lợi ích của Oracle GoldenGate:

* Dữ liệu được gửi đi theo thời gian thực.
* Chỉ những transaction đã commit được gửi đi giúp tăng hiệu năng và duy trì tính nhất quán dữ liệu. Những transaction chưa commit cũng được thu thập bởi GoldenGate những bị loại bỏ khi nhận được lệnh Rollback.
* Hỗ trợ các phiên bản khác nhau của Oracle và nhiều cở sở dữ liệu khác trên các nền tảng hệ điều hành khác nhau.
* Hiệu năng cao trong khi tác động lên cơ sở dữ liệu rất nhỏ.
* Kiến trúc và cấu hình GoldenGate rất đơn giản, dễ hiểu khiến GoldenGate trở thành một cung cụ hiệu quả mà dễ học và dễ thực hiện.
* Có cơ chế phục hồi dữ liệu để giải quyết các loại lỗi khác nhau: lỗi site, lỗi kết nối mạng …
* Sử dụng các tệp định dạng riêng để giữ lại các dữ liệu được commit từ cở sở dữ liệu nguồn và hoàn toàn độc lập với cơ sở dữ liệu đang sử dụng.
* Sử dụng mạng chuẩn giữa CSDL nguồn và CSDL mà không dựa vào dịch vụ mạng của Oracle Database. Do đó dữ liệu được truyền đi mà không ảnh hưởng đến hiệu suất mạng trên CSDL nguồn và CSDL đích.
* Có cơ chế riêng để theo dõi số lượng transaction đã được hoàn thành và số transaction đang chờ xử lý. Cơ chế này độc lập với cơ sở dữ liệu nên đảm bảo rằng dữ liệu không bị mất hoặc có transaction bị thực hiện nhiều lần.

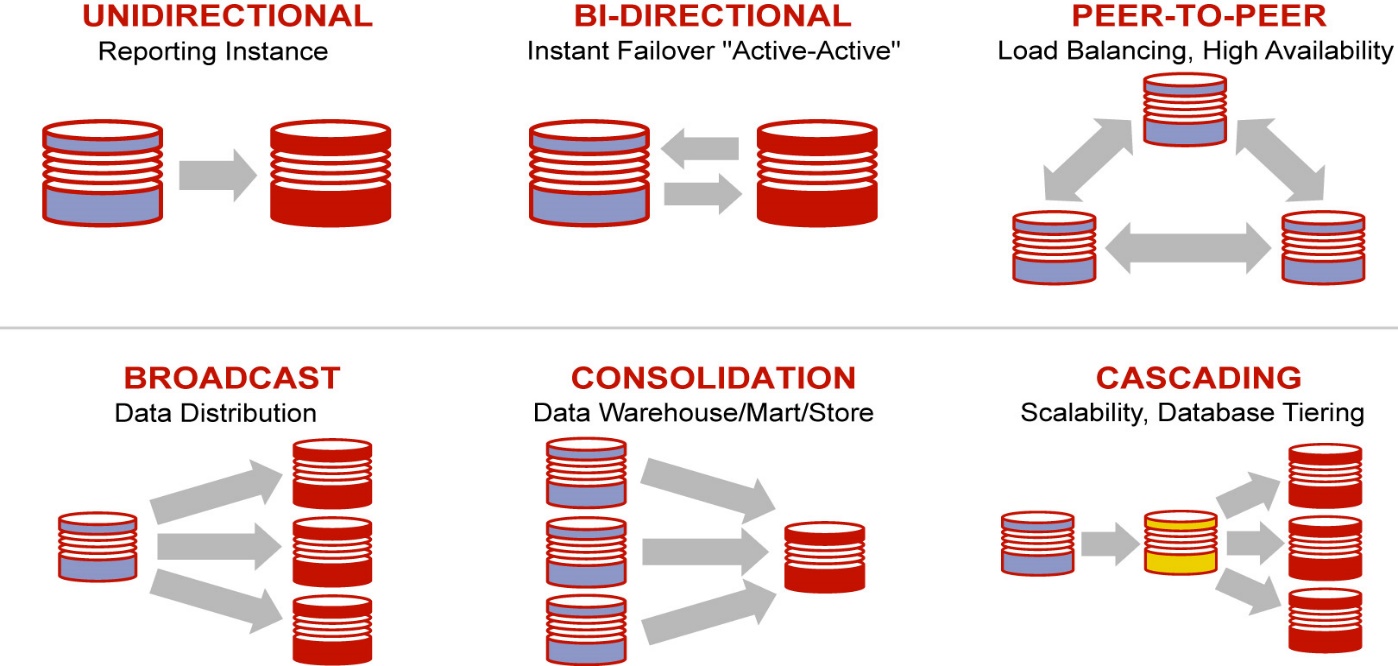
***Với những lợi ích nêu trên Oracle GoldenGate cung cấp các giải pháp:***

* Sẵn sàng cao

Với Oracle GoldenGate ta có thể có một CSDL phòng được liên tục đồng bộ với CSDL chính. Trong trường hợp CSDL chính gặp sự cố có thể ngay lập tức chuyển đổi.

* Nâng cấp và chuyển đổi CSDL không có thời gian chết
* Báo cáo trực tuyến

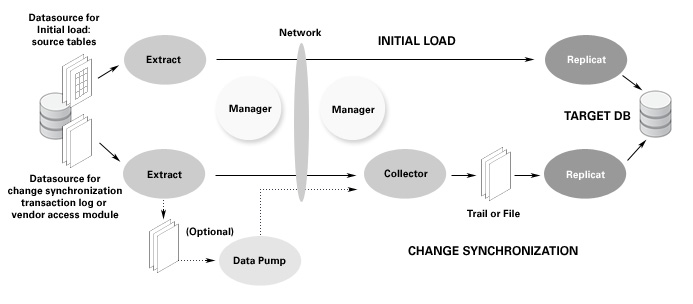
Tạo một hệ thống riêng biệt để sử dụng cho việc tạo báo cáo từ đó làm giảm tải trên CSDL nguồn.



Hình 5 Các mô hình đồng bộ cơ bản Oracle GoldenGate

* 1. **Kiến trúc của Oracle GoldenGate**

Kiến trúc của Oracle GoldenGate bao gồm các tiến trình: Manager, Extract, Data Pump, Replicat, Collector và các files: Trail files, parameter files, report files.



Hình 6. Kiến trúc logic Oracle GoldenGate

* + 1. Manager

Là tiến trình chạy trên cả hệ thống nguồn và hệ thống đích với nhiệm vụ:

* Khởi động hoặc khởi động lại các tiến trình khác của Oracle GoldenGate
* Quản lý cổng kết nối của các tiến trình
* Quản lý các tập tin Trail
* Quản lý các sự kiện, lỗi và tạo báo cáo.
  + 1. Extract

Extract là tiến trình chạy trên hệ thống nguồn. Trong khi Extract đang hoạt động sẽ liên tục quét transaction log (redo log nếu là oracle) để thu thập những những thay đổi dữ liệu của database trên hệ thống nguồn và ghi vào trail file.

* + 1. Data Pump

Data Pump là tiến trình chạy trên hệ thống nguồn. Data Pump liên tục quét thông tin trong trail file. Khi có dữ liêu mới được phát hiện trong trail file tiến trình này sẽ đóng gói và chuyển dữ liệu này đến remote trail file trên hệ thống đích thông qua giao thức TCP/IP. Ngoài ra, Data Pump cho phép mã hóa và nén dữ liệu trong gói tin TCP/IP.

Lợi ích sử dụng Data Pump:

* Phục hồi dữ liệu trong trường hợp lỗi mạng hoặc lỗi trên hệ thống đích.
* Có thể thực hiện lọc và chuyển đổi dữ liệu.
* Đồng bộ dữ liệu từ nhiều database nguồn đến một database đích.
* Đồng bộ dữ liệu từ một database nguồn đến nhiều database đích, khi gửi dữ liệu tới nhiều database đích. Có thể cấu hình Data Pump trên database nguồn cho mỗi database đích. Nếu mất kết nối tới một database đích nào đó thì dữ liệu vẫn có thể gửi tới các database đích khác.
  + 1. Replicat

Replicat là tiến trình chạy dến hệ thống đích, được sử đụng để đọc Remote Trail và cấu trúc lại thao tác DML, DDL sau đó tích hợp vào CSDL đích.

* + 1. Collector

Collector là tiến trình có sẵn trên hệ thống đích. Collector nhận dữ liệu đã commit được gửi bởi Extract hoặc Data Pump và ghi dữ liệu này vào Remote Trail.

* + 1. Trail files

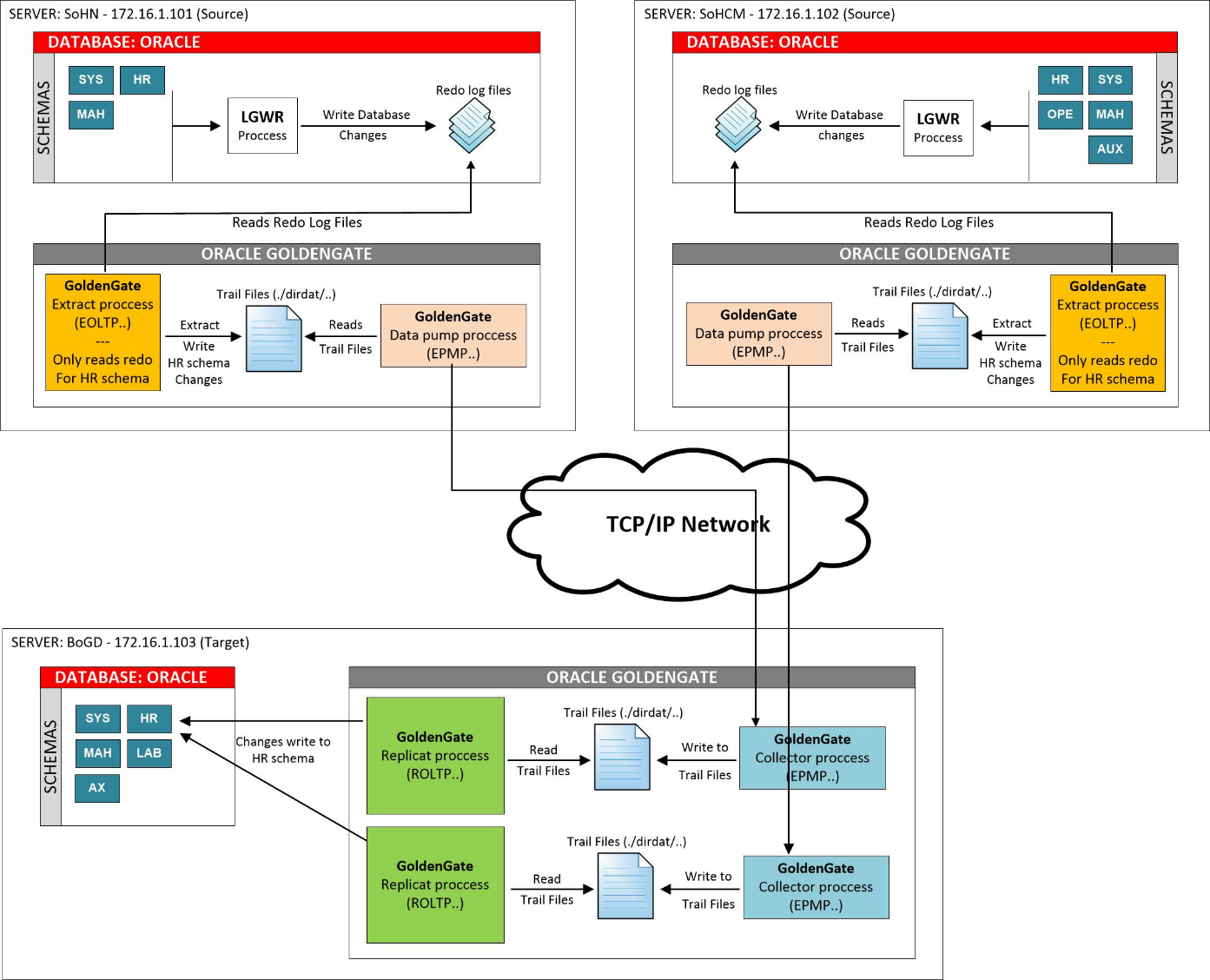
Để hỗ trợ việc liên tục thu thập và sao chép những thay đổi của database, Oracle GoldenGate lưu trữ bản ghi về những thay đổi này trên ở đĩa trong một loại file được gọi là Trail.

Với việc sử dụng trail để lưu trữ. Oracle GoldenGate cung cấp tính chính xác của dữ liệu và khả năng chịu lỗi. Với việc sử dụng Trail cho phép quá trình thu thập và sao chép dữ liệu diễn ra độc lập.

**Chương 4: Mô hình đồng bộ dữ liệu từ nhiều site về một site.**

# **Chương 5: Xây dựng mô hình đồng bộ dữ liệu điểm thi.**

1. **Mô hình cơ sở dữ liệu trên server nguồn và đích**
2. **Mô hình cài đặt**



Hình 7. Mô hình dồng bộ điểm thi

|  |  |  |  |
| --- | --- | --- | --- |
|  | Server nguồn | | Server đích |
| Host Name | SoHN  192.168.1.101 | SoHCM  192.168.1.102 | BoGD  192.168.1.103 |
| Database Software | Oracle Database 11g R2 | Oracle Database 11g R2 | Oracle Database 11g R2 |
| OS | RedHat 6.6 64-bit | Oracle Linux R6 64-bit | RedHat 6.6 64-bit |
| Oracle SID | SoHN | SoHCM | BoGD |
| Sync Table | HR.THISINH  HR.DIADIEM  HR.CUMTHI  HR.DIEMTHI  HR.PHONGTHI  … | HR.THISINH  HR.DIADIEM  HR.CUMTHI  HR.DIEMTHI  HR.PHONGTHI  … | HR.THISINH  HR.DIADIEM  HR.CUMTHI  HR.DIEMTHI  HR.PHONGTHI  … |

1. **Các bước thực hiện cầu hình đồng bộ dữ liệu**
   1. Yêu cầu

* Server nguồn và đích đã được cài đặt Oracle Database
* Download bộ phần mềm Oracle GoldenGate tương ứng với OS, database của môi trường đồng bộ sau đó thực hiện unzip– (trên cả server nguồn và đích).
* Chỉnh sửa file host với nội dung như sau(trên cả server nguồn và đích)

192.168.1.101 SoHN

192.168.1.102 SoHCM

192.168.1.103 BoGD

* 1. Cài đặt

**Bước 1**: Tạo user, gán các quyền cho user trên database nguồn và đích.

|  |
| --- |
| [oracle@dbserver1 gg]$ sqlplus '/as sysdba'  sql> Create user ggs\_admin identified by ggs123  sql> Grant create session, dba to ggs\_admin |

**Bước 2**: Thực hiện enable supplemental, chuyển redo log file trên database nguồn

|  |
| --- |
| Sql> Alter database add supplemental log data;  sql> Alter system switch logfile;  ###Kiểm tra  sql> Select supplemental\_log\_data\_min from v$database |

**Bước 3**: Thực hiện cấu hình đồng bộ.

**Bước 3.1**:

+Thực thi file ggsci (trên cả server nguồn và đích)

|  |
| --- |
| $ cd /u02/gg  $ ./ggsci |

+Xuất hiện màn hình command line: Tạo các thư mục mặc định cho các tiến trình đồng bộ (trên cả server nguồn và đích)

|  |
| --- |
| ggsci> create subdirs |

**Bước 3.2**: Cấu hình Manager proccess (trên cả server nguồn và đích)

|  |
| --- |
| ggsci> dblogin userid ggs\_admin, password ggs123  ggsic> edit params mgr  PORT 7809  PURGEOLDEXTRACTS ./dirdat/\*, USECHECKPOINTS, MINKEEPHOURS 120  AUTOSTART ER \*  AUTORESTART ER \*, WAITMINUTES 10, RETRIES 20 |

Ghi chú|

* PURGEOLDEXTRACTS ./dirdat/\*, USECHECKPOINTS, MINKEEPHOURS 120: Tự động xóa các file trail cũ đã được sử dụng để đồng bộ sau 120 giờ.
* AUTOSTART: Tự động start các tiến trình Extract khi Manager được start
* AUTORESTART: Tự động restart các tiến trình bị ngắt

**Bước 3.3**: Cấu hình trên server nguồn

|  |
| --- |
| ggsci> dblogin userid ggs\_admin, password ggs12345  ggsci(SoHN)> add trandata HR.  ggsci(SoHN)> add trandata HR.  ggsci(SoHN)> add trandata HR.  ggsci(SoHN)> add trandata HR.  ggsci(SoHN)> add trandata HR.  ggsci(SoHN)> add trandata HR.  ggsci(SoHN)> add trandata HR.  ggsci(SoHN)> add trandata HR. |

Ghi chú

-Nếu có nhiều bảng cần đồng bộ sử dụng sql để sinh câu lệnh **add trandata**

sql> set pages 0

sql> select add trandata HR.'||tname from tab;

Thực hiện lệnh sau trên Server SoHN:

|  |
| --- |
| ggsci(SoHN)> add extract EOLTP01, tranlog, begin now  ggsci(SoHN)> add exttrail ./dirdat/aa, extract EOLTP01  ggsci(SoHN)> add extract EPMP01, exttrailsource ./dirdat/aa  ggsci(SoHN)> add rmttrail ./dirdat/aa, extract EPMP01  ggsci(SoHN)> add extract EOLTP02, tranlog, begin now  ggsci(SoHN)> add exttrail ./dirdat/ab, extract EOLTP02  ggsci(SoHN)> add extract EPMP02, exttrailsource ./dirdat/ab  ggsci(SoHN)> add rmttrail ./dirdat/ab, extract EPMP02  ggsci(SoHN)> add extract EOLTP03, tranlog, begin now  ggsci(SoHN)> add exttrail ./dirdat/ac, extract EOLTP03  ggsci(SoHN)> add extract EPMP03, exttrailsource ./dirdat/sc  ggsci(SoHN)> add rmttrail ./dirdat/ac, extract EPMP03 |

Thực hiện lệnh sau trên Server SoHCM:

|  |
| --- |
| ggsci(SoHN)> add extract EOLTP01, tranlog, begin now  ggsci(SoHN)> add exttrail ./dirdat/ba, extract EOLTP01  ggsci(SoHN)> add extract EPMP01, exttrailsource ./dirdat/ba  ggsci(SoHN)> add rmttrail ./dirdat/ba, extract EPMP01  ggsci(SoHN)> add extract EOLTP02, tranlog, begin now  ggsci(SoHN)> add exttrail ./dirdat/bb, extract EOLTP02  ggsci(SoHN)> add extract EPMP02, exttrailsource ./dirdat/bb  ggsci(SoHN)> add rmttrail ./dirdat/bb, extract EPMP02  ggsci(SoHN)> add extract EOLTP03, tranlog, begin now  ggsci(SoHN)> add exttrail ./dirdat/bc, extract EOLTP03  ggsci(SoHN)> add extract EPMP03, exttrailsource ./dirdat/bc  ggsci(SoHN)> add rmttrail ./dirdat/bc, extract EPMP03 |

Chú ý:

* Tranlog:

**Bước 3.4**: Cấu hình đồng bộ trên server đích

|  |
| --- |
| ggsci(BoGD)> dblogin userid ggs\_admin, password ggs12345  ggsci(BoGD)> add checkpointtable ggs.tbl\_chkpt  ggsci(BoGD)> add replicat ROLTP01, ExtTrail ./dirdat/aa, checkpoint ggs.tbl\_chkpt  ggsci(BoGD)> add replicat ROLTP02, ExtTrail ./dirdat/ab, checkpoint ggs.tbl\_chkpt  ggsci(BoGD)> add replicat ROLTP03, ExtTrail ./dirdat/ac, checkpoint ggs.tbl\_chkpt  ggsci(BoGD)> add replicat ROLTP04, ExtTrail ./dirdat/ba, checkpoint ggs.tbl\_chkpt  ggsci(BoGD)> add replicat ROLTP05, ExtTrail ./dirdat/bb, checkpoint ggs.tbl\_chkpt  ggsci(BoGD)> add replicat ROLTP06, ExtTrail ./dirdat/bc, checkpoint ggs.tbl\_chkpt |

Chú ý:

* Add checkpointtable ggs.tbl\_chkpt:

**Bước 3.5**: Tạo và cấu hình các file tham số trên server nguồn:

Thực hiện mã hóa mật khẩu ggs\_admin (sử dụng mã Blowfish với key mặc định):

|  |
| --- |
| ggsci> encrypt password ggs123 AES192 encryptkey 123456a@ |

Bước 3.5.1: Cấu hình Extract proccess (trên cả 2 server nguồn)

|  |
| --- |
| ggsci(SoHN)> edit param EOLTP01  EXTRACT EOLTP01  REPORTROLLOVER on monday  REPORTCOUNT every 1 hours, RATE  USERID ggs, PASSWORD goldengate  DISCARDFILE ./dirout/EOLTP01.dsc, APPEND  EXTTRAIL ./dirdat/sa  ENCRYPTTRAIL AES192 KEYNAME 123456a@  TABLE HR.  TABLE HR. |
| ggsci(SoHN)> edit param EOLTP02  EXTRACT EOLTP02  REPORTROLLOVER on monday  REPORTCOUNT every 1 hours, RATE  USERID ggs, PASSWORD goldengate  DISCARDFILE ./dirout/EOLTP02.dsc, APPEND  EXTTRAIL ./dirdat/sb  ENCRYPTTRAIL AES192 KEYNAME 123456a@  TABLE HR.  TABLE HR. |
| ggsci(SoHN)> edit param EOLTP03  EXTRACT EOLTP03  REPORTROLLOVER on monday  REPORTCOUNT every 1 hours, RATE  USERID ggs, PASSWORD goldengate  DISCARDFILE ./dirout/EOLTP03.dsc, APPEND  EXTTRAIL ./dirdat/sc  ENCRYPTTRAIL AES192 KEYNAME 123456a@  TABLE HR.  TABLE HR. |

Bước 3.5.1: Cấu hình Data pump proccess

**Trên server SoHN:**

|  |
| --- |
| ggsci(SoHN)> edit param EPMP01  EXTRACT EPMP01  UserId ggs, Password goldengate  DISCARDFILE ./dirout/EPMP01.dsc, APPEND  REPORTROLLOVER on monday  REPORTCOUNT every 1 hours, RATE  DECRYPTTRAIL AES192 KEYNAME 123456a@  RMTHOST BoGD, MGRPORT 7809  ENCRYPTTRAIL AES192 KEYNAME 123456aA@  RMTTRAIL ./dirdat/aa  PASSTHRU  TABLE HR.  TABLE HR. |
| ggsci(SoHN)> edit param EPMP02  EXTRACT EPMP02  UserId ggs, Password goldengate  DISCARDFILE ./dirout/EPMP02.dsc, APPEND  REPORTROLLOVER on monday  REPORTCOUNT every 1 hours, RATE  DECRYPTTRAIL AES192 KEYNAME 123456a@  RMTHOST BoGD, MGRPORT 7809  ENCRYPTTRAIL AES192 KEYNAME 123456aA@  RMTTRAIL ./dirdat/ab  PASSTHRU  TABLE HR.  TABLE HR. |
| ggsci(SoHN)> edit param EPMP03  EXTRACT EPMP03  UserId ggs, Password goldengate  DISCARDFILE ./dirout/EPMP03.dsc, APPEND  REPORTROLLOVER on monday  REPORTCOUNT every 1 hours, RATE  DECRYPTTRAIL AES192 KEYNAME 123456a@  RMTHOST BoGD, MGRPORT 7809  ENCRYPTTRAIL AES192 KEYNAME 123456aA@  RMTTRAIL ./dirdat/ac  PASSTHRU  TABLE HR.  TABLE HR. |

**Trên server soHCM:**

|  |
| --- |
| ggsci(SoHN)> edit param EPMP01  EXTRACT EPMP01  UserId ggs, Password goldengate  DISCARDFILE ./dirout/EPMP01.dsc, APPEND  REPORTROLLOVER on monday  REPORTCOUNT every 1 hours, RATE  DECRYPTTRAIL AES192 KEYNAME 123456a@  RMTHOST BoGD, MGRPORT 7809  ENCRYPTTRAIL AES192 KEYNAME 123456aA@  RMTTRAIL ./dirdat/ba  PASSTHRU  TABLE HR.  TABLE HR. |
| ggsci(SoHN)> edit param EPMP02  EXTRACT EPMP02  UserId ggs, Password goldengate  DISCARDFILE ./dirout/EPMP02.dsc, APPEND  REPORTROLLOVER on monday  REPORTCOUNT every 1 hours, RATE  DECRYPTTRAIL AES192 KEYNAME 123456a@  RMTHOST BoGD, MGRPORT 7809  ENCRYPTTRAIL AES192 KEYNAME 123456aA@  RMTTRAIL ./dirdat/bb  PASSTHRU  TABLE HR.  TABLE HR. |
| ggsci(SoHN)> edit param EPMP03  EXTRACT EPMP03  UserId ggs, Password goldengate  DISCARDFILE ./dirout/EPMP03.dsc, APPEND  REPORTROLLOVER on monday  REPORTCOUNT every 1 hours, RATE  DECRYPTTRAIL AES192 KEYNAME 123456a@  RMTHOST BoGD, MGRPORT 7809  ENCRYPTTRAIL AES192 KEYNAME 123456aA@  RMTTRAIL ./dirdat/bc  PASSTHRU  TABLE HR.  TABLE HR. |

Chú ý:

* Các file tham số cấu hình được lưu trong đường dẫn /u02/gg/dirrpm

**Bước 3.6**: Tạo và cầu hình các file tham số trên server đích

|  |
| --- |
| ggsci(BoGD)> edit param ROLTP01  REPLICAT ROLTP01  USERID ggs, PASSWORD goldengate  HANDLECOLLISIONS  ASSUMETARGETDEFS  DISCARDFILE ./dirrpt/ROLTP01.dsc, APPEND  DECRYPTTRAIL AES192 KEYNAME 123456aA@  MAP HR., TARGET HR.  MAP HR., TARGET HR. |
| ggsci(BoGD)> edit param ROLTP02  REPLICAT ROLTP02  USERID ggs, PASSWORD goldengate  HANDLECOLLISIONS  ASSUMETARGETDEFS  DISCARDFILE ./dirrpt/ROLTP02.dsc, APPEND  DECRYPTTRAIL AES192 KEYNAME 123456aA@  MAP HR., TARGET HR.  MAP HR., TARGET HR. |
| ggsci(BoGD)> edit param ROLTP03  REPLICAT ROLTP03  USERID ggs, PASSWORD goldengate  HANDLECOLLISIONS  ASSUMETARGETDEFS  DISCARDFILE ./dirrpt/ROLTP03.dsc, APPEND  DECRYPTTRAIL AES192 KEYNAME 123456aA@  MAP HR., TARGET HR.  MAP HR., TARGET HR. |
| ggsci(BoGD)> edit param ROLTP04  REPLICAT ROLTP04  USERID ggs, PASSWORD goldengate  HANDLECOLLISIONS  ASSUMETARGETDEFS  DISCARDFILE ./dirrpt/ROLTP04.dsc, APPEND  DECRYPTTRAIL AES192 KEYNAME 123456aA@  MAP HR., TARGET HR.  MAP HR., TARGET HR. |
| ggsci(BoGD)> edit param ROLTP05  REPLICAT ROLTP05  USERID ggs, PASSWORD goldengate  HANDLECOLLISIONS  ASSUMETARGETDEFS  DISCARDFILE ./dirrpt/ROLTP05.dsc, APPEND  DECRYPTTRAIL AES192 KEYNAME 123456aA@  MAP HR., TARGET HR.  MAP HR., TARGET HR. |
| ggsci(BoGD)> edit param ROLTP06  REPLICAT ROLTP06  USERID ggs, PASSWORD goldengate  HANDLECOLLISIONS  ASSUMETARGETDEFS  DISCARDFILE ./dirrpt/ROLTP06.dsc, APPEND  DECRYPTTRAIL AES192 KEYNAME 123456aA@  MAP HR., TARGET HR.  MAP HR., TARGET HR. |

Chú ý:

* Tham số HANDLECOLLISIONS được thiết lập có ý nghĩa: cho phép bỏ qua tất cả những TH bị collision, ví dụ: nếu server đích đang record có ID: 1078 rồi thì server nguồn khi đồng bộ sang sẽ bỏ qua record có ID này mà ko báo lỗi GG nữa.

**Bước 5**: Thực hiện quá trình Initload

Cách 1: Sử dụng công cụ import/export của Oracle database

-Thực hiện export dữ liệu trên database nguồn và import vào database đích.

Trước khi thực hiện export thực hiện cầu lệnh sau.

Trong trường hợp database nguồn có khóa phụ, trigger cần thực hiện loại bỏ khóa phụ và trigger

**Bước 6**: Khởi động các tiến trình (trên cả server nguồn và đích)

Trên server nguồn

|  |
| --- |
| ggsci> start mgr  ggsci> start E\*  ggsci> start P\* |

Trên server đích

|  |
| --- |
| ggsci> start mgr  ggsci> start R\* |

Kiểm tra trạng thái của các tiến trình

|  |
| --- |
| ggsci> info all |

# **Phụ lục**

## Overview of Checkpoints

## About the Commit Sequence Number

# **Tài liệu tham khảo**

Book: Oracle\_GoldenGate\_11g\_Implementer's\_guide

https://docs.oracle.com/cd/A81042\_01/DOC/server.816/a76965/c06memor.htm

<https://www.red-gate.com/simple-talk/sql/oracle/goldengate-an-introduction/>

<http://www.oracle.com/technetwork/middleware/goldengate/overview/index.html>

<https://docs.oracle.com/goldengate/c1230/gg-winux/GGCON/introduction-oracle-goldengate.htm>

<https://docs.oracle.com/goldengate/1212/gg-inux/GWUAD/wu_about_gg.htm>