

QUẢN LÝ MỘT ORACLE INSTANCE

1. Mục đích

- Sau khi hoàn thành bài học này, bạn có thể:
 - Tạo và quản lý các file tham số.
 - Khởi động và tắt một instance.
 - Theo dõi và sử dụng các diagnostic file.

2. Nội dung bài học

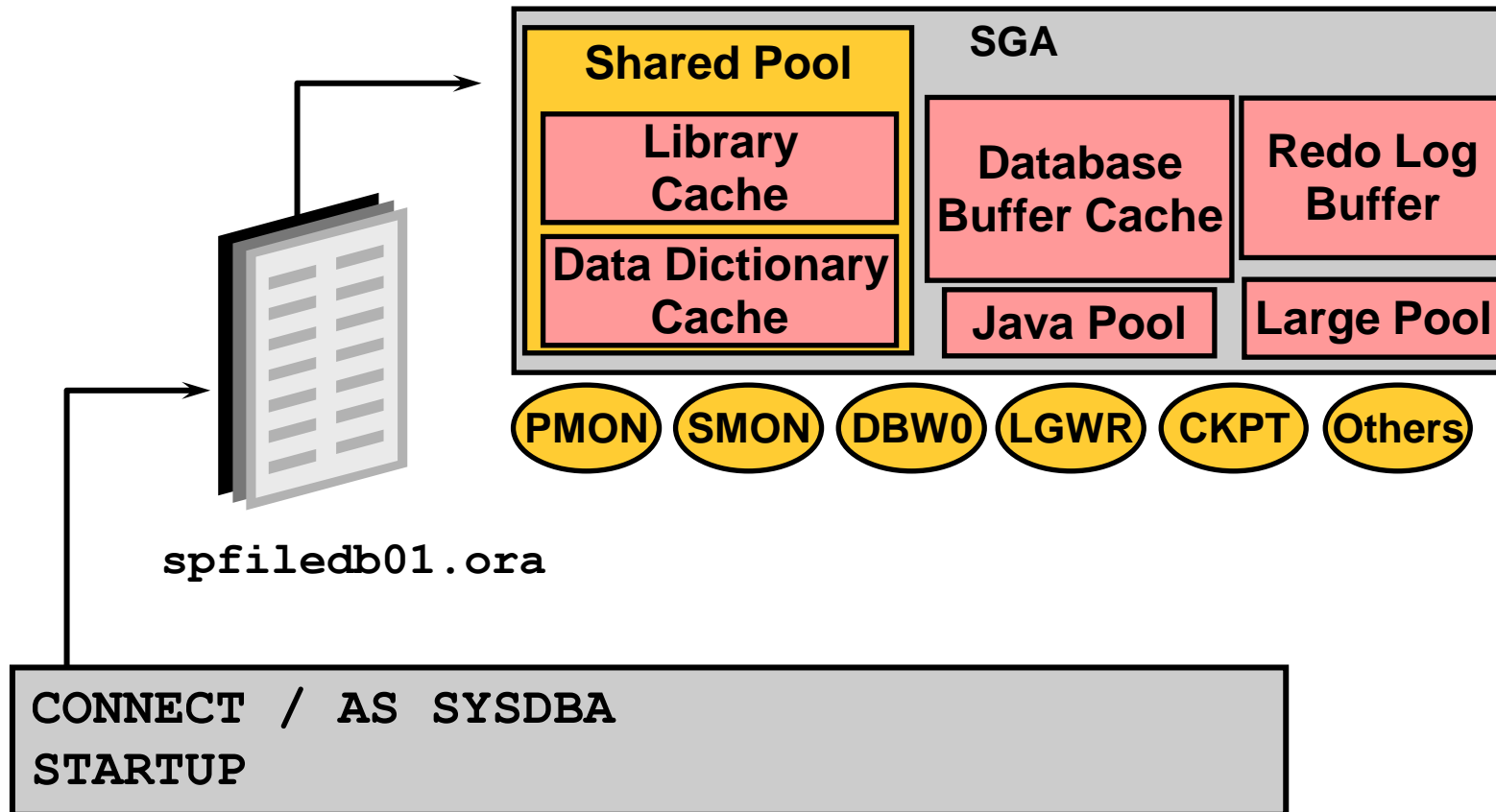
I. FILE THAM SỐ KHỞI TẠO ORACLE INSTANCE

II. KHỞI ĐỘNG VÀ TẮT MỘT ORACLE INSTANCE

III. TỔNG QUAN DIAGNOSTIC FILES

IV. THỰC HÀNH

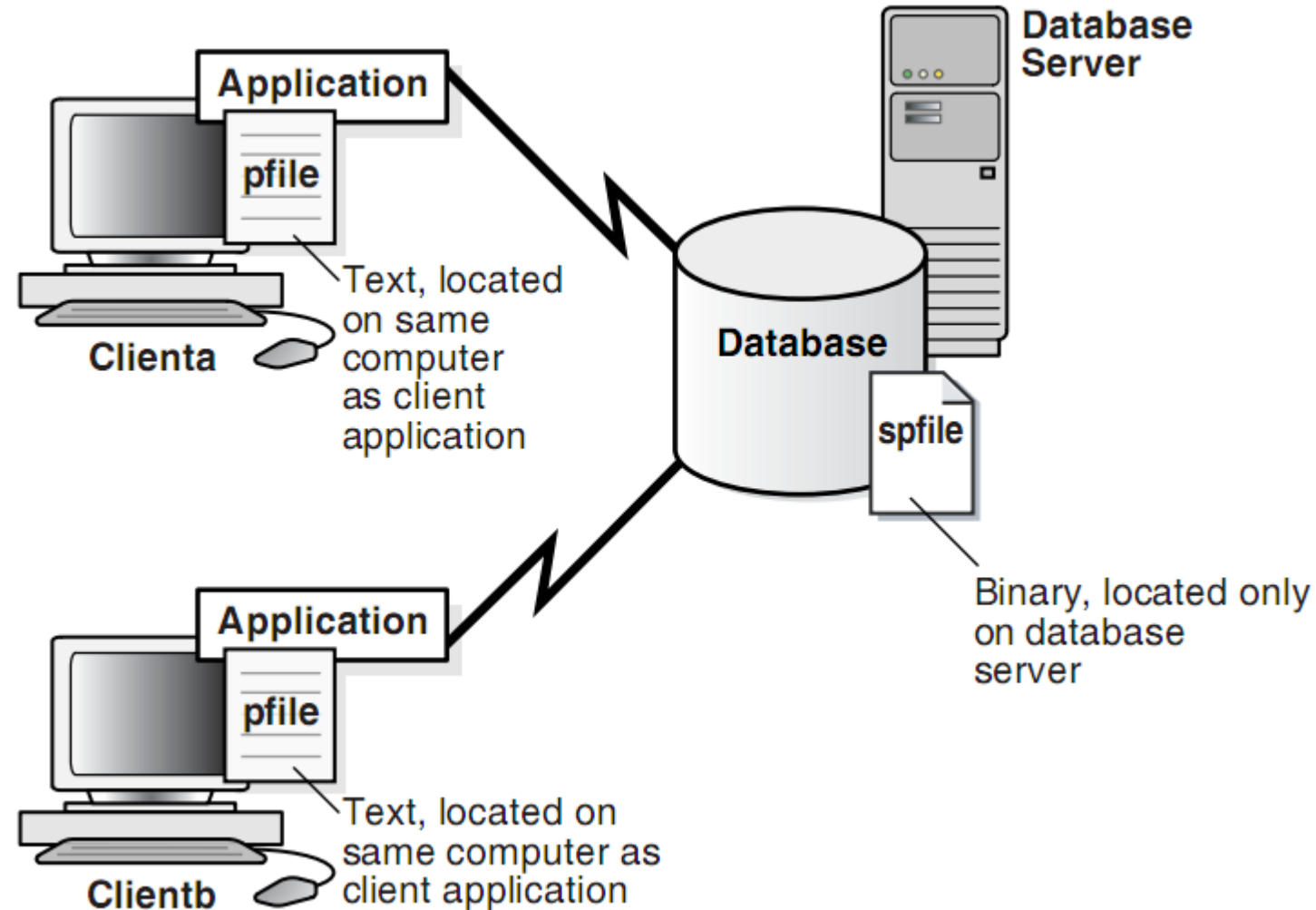
I. FILE THAM SỐ KHỞI TẠO ORACLE INSTANCE



1. Các loại file tham số khởi tạo

Để khởi động một Oracle Instance, Oracle server sẽ đọc thông tin các tham số trong file tham số khởi tạo, có 2 loại file tham số khởi tạo là:

- PFILE
- SPFILE



2. Nội dung trong file tham số

- **Tập hợp các tham số của instance như:**
 - Tên của instance
 - Kích thước bộ nhớ các thành phần trong SGA
 - Tên và vị trí control files
- **Các tham số trong file tham số được lưu trữ thành các mục (entry)**
 - Entries : Tên tham số = Giá trị.
 - 2 loại tham số:
 - Cụ thể: Có một entry trong file.
 - Ẩn: Không có entry trong file nhưng chọn các giá trị mặc định của Oracle.

3. PFILE

- File dạng text
- Điều chỉnh bởi chương trình soạn thảo của HĐH
- Các điều chỉnh được thực hiện bằng tay
- Các thay đổi có hiệu lực vào lần khởi động kế tiếp
- Chỉ mở khi instance khởi động
- Vị trí mặc định

%ORACLE_HOME%/database với Windown

\$ORACLE_HOME\dbs với Unix

- Định dạng tên: **initSID.ora**

3. PFILE

Ví dụ về PFILE

```
# Initialization Parameter File: initdba01.ora
db_name                = dba01
instance_name          = dba01
control_files           = (
                          /home/dba01/ORADATA/u01/control01dba01.ctl,
                          /home/dba01/ORADATA/u02/control01dba02.ctl)
db_block_size           = 4096
db_cache_size           = 4M
shared_pool_size        = 50000000
java_pool_size          = 50000000
max_dump_file_size      = 10240
background_dump_dest    = /home/dba01/ADMIN/BDUMP
user_dump_dest          = /home/dba01/ADMIN/UDUMP
core_dump_dest          = /home/dba01/ADMIN/CDUMP
undo_management         = AUTO
undo_tablespace         = UNDOTBS
```


4. SPFILE

- **Binary file.**
- **Được quản lý bởi Oracle Server.**
- **Luôn luôn đặt ở máy chủ.**
- **Có khả năng tạo ra các thay đổi mà không phải tắt và khởi động lại database.**
- **Định dạng tên: spfileSID.ora**

4. SPFILE

❖ Tạo một SPFILE

- Được tạo ra từ file PFILE:

```
SQL> CREATE SPFILE [= 'SPFILE-NAME' ] FROM  
PFILE [= 'PFILE-NAME' ] ;
```

- Có thể được thực hiện trước hoặc sau khi khởi động instance.
- VD: *create spfile='C:\spfileDB1.ora' from pfile;*

❖ Xuất nội dung của SPFILE lưu trữ vào PFILE

```
SQL> CREATE PFILE [= 'PFILE-NAME' ] FROM  
SPFILE [= 'SPFILE-NAME' ] ;
```

- Có thể được thực hiện trước hoặc sau khi khởi động instance.

4. SPFILE

❖ Điều chỉnh các tham số trong SPFILE

Cú pháp:

```
ALTER SYSTEM {SET|RESET}parameter_name = parameter_value  
[SCOPE = {MEMORY|SPFILE|BOTH}] [SID= 'sid'|'*'];
```

Trong đó:

- **SET|RESET:** Thiết lập hoặc thiết lập lại
- **parameter_name:** Tên tham số cần thay đổi
- **parameter_value:** Giá trị tham số
- **SCOPE:** Xác định hiệu lực của sự thay đổi
- **MEMORY:** Sự thay đổi chỉ có hiệu lực ở Instance hiện tại. Thiết lập này là mặc định nếu instance được khởi động bằng Pfile.
- **SPFILE:** Thay đổi giá trị tham số chỉ trong spfile, có hiệu lực khi khởi động lại Instance.
- **BOTH:** Thay đổi giá trị tham số ở cả SPFILE và Instance hiện tại

4. SPFILE

❖ Điều chỉnh các tham số trong SPFILE

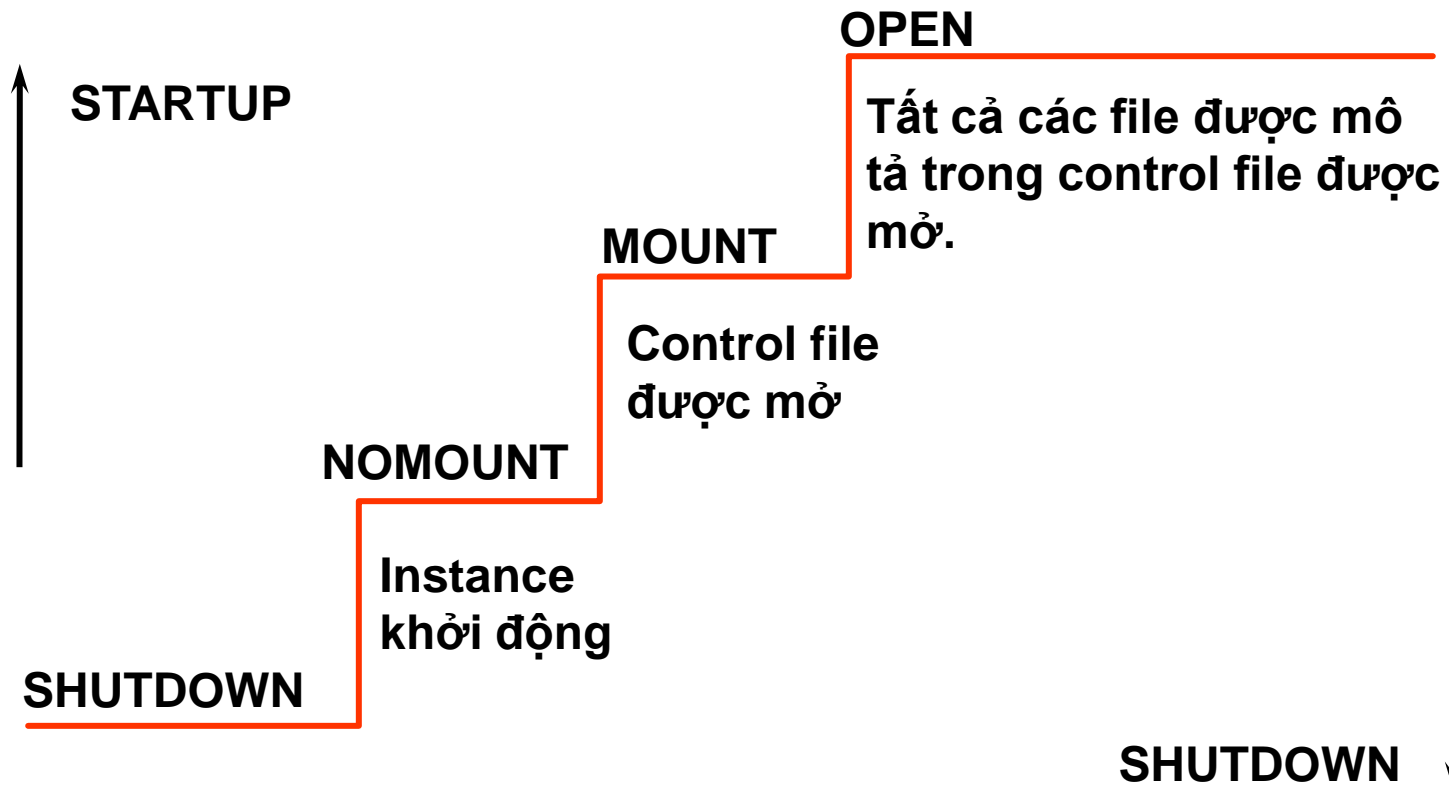
VD:

SQL> show parameters open_cursors;//Hiển thị giá trị open_cursors hiện tại.

SQL> ALTER SYSTEM SET open_cursors=301 SCOPE=BOTH;

II. KHỞI ĐỘNG VÀ TẮT MỘT ORACLE INSTANCE

1. Khởi động một CSDL



1. Khởi động một CSDL

Đặc điểm Lệnh STARTUP

- Thư mục mặc định của file tham số:

%Oracle_home%\database

- Thứ tự khi startup:

- spfileSID.ora.
- Default SPFILE. (spfile.ora)
- initSID.ora.

- Chỉ rõ khởi động bằng PFILE:

```
STARTUP PFILE = 'C:\initORCL.ora' ;
```

- Trong PFILE có thể chỉ cần chỉ ra sử dụng spfile:

```
SPFILE = 'C:\spfileORCL.ora' ;
```

1. Khởi động một CSDL

Cú pháp:

```
STARTUP [FORCE][RESTRICT] [PFILE=filename] [MOUNT] |  
[OPEN [open_options]] | [NOMOUNT]
```

Trong đó:

FORCE: Khởi động ngay lập tức, nếu instance đang hoạt động thì sẽ bị shutdown ở chế độ abort.

RESTRICT: Khởi động ở chế độ hạn chế, chỉ có user có quyền admin mới đăng nhập được.

open_options: READ {ONLY | WRITE [RECOVER]} | RECOVER

2. Thay đổi trạng thái CSDL

Lệnh ALTER DATABASE

- Thay đổi trạng thái của CSDL từ NOMOUNT sang MOUNT hoặc từ MOUNT sang OPEN:

```
ALTER DATABASE db01 MOUNT|OPEN;
```

- Mở CSDL ở chế độ read-only: (chỉ thực hiện được khi database ở chế độ MOUNT)

```
ALTER DATABASE db01 OPEN READ ONLY;
```

- Bật/tắt 1 instance ở chế độ giới hạn truy cập:

```
ALTER SYSTEM ENABLE/DISABLE RESTRICTED  
SESSION;
```


3. Tắt một CSDL

Cú pháp: SQL> SHUTDOWN [NORMAL | TRANSACTIONAL |
IMMEDIATE | ABORT]

Shutdown Mode	ABORT	IMMEDIATE	TRANSACTIONAL	NORMAL
Cho phép tạo các kết nối mới	NO	NO	NO	NO
Đợi các phiên hiện thời kết thúc	NO	NO	NO	YES
Đợi các giao dịch hiện thời kết thúc	NO	NO	YES	YES
T.hiện một checkpoint, đóng các file	NO	YES	YES	YES

III. TỔNG QUAN DIAGNOSTIC FILES

- Diagnostic files
 - Chứa các thông tin về những sự kiện quan trọng mà Database gặp phải.
 - Được sử dụng để giải quyết vấn đề.
- Một số loại file:
 - alertSID.log file.
 - Background trace file.
 - User trace file.

1. Alert Log File

- File alertSID.log:
 - Ghi lại các lệnh.
 - Ghi lại kết quả của các sự kiện chính.
 - Được sử dụng cho thông tin hoạt động hàng ngày.
 - Được sử dụng cho việc chẩn đoán các lỗi CSDL.
- Mỗi entry có một mốc thời gian liên quan tới nó.
- Được quản lý bởi DBA.
- Vị trí đặt file này được xác định thông qua BACKGROUND_DUMP_DEST.

2. Background Trace File

- Background trace file:
 - Ghi chép các lỗi phát hiện bởi các background process.
 - Được sử dụng để chẩn đoán và sửa các lỗi.
- File này được tạo ra khi background process gặp lỗi.
- Vị trí đặt file được xác định thông qua `BACKGROUND_DUMP_DEST`.

3. User Trace File

- User trace files
 - Tạo ra bởi user process.
 - Cũng có thể được tạo ra bởi server process.
 - Chứa số liệu thống kê cho các lệnh SQL.
 - Chứa các thông báo lỗi của người sử dụng.
- Được tạo ra khi user session của người sử dụng gặp lỗi.
- Vị trí đặt file được xác định bởi USER_DUMP_DEST.
- Kích thước file được xác định thông qua MAX_DUMP_FILE_SIZE.

Thực hành

1. Đăng nhập vào user sys và shutdown database.
2. Sau khi shutdown database, tạo spfile từ pfile. Spfile đặt ở thư mục mặc định.
3. Xem nội dung SPFILE.
4. Đăng nhập vào user sys và startup database từ spfile.
5. Mở khóa user HR và thay đổi mật khẩu là **hr** như sau:

SQL> alter user HR account unlock identified by hr;

6. Shutdown database và mở lại ở chế độ read-only.
7. Đăng nhập vào user HR và thực hiện insert vào bảng REGIONS như sau:

INSERT INTO regions VALUES (5, 'Mars');

Điều gì sẽ xảy ra?

8. Chuyển database sang chế độ read-write, thực hiện insert lại vào bảng REGIONS nhưng chưa commit;
9. Mở 1 session mới và đăng nhập vào user sys, thực hiện Shutdown database ở chế độ TRANSACTIONAL.

Điều gì sẽ xảy ra ở phiên làm việc của user sys?

10. Rollback dữ liệu vừa insert vào bảng HR, điều gì sẽ xảy ra?
11. Tắt 2 session và tạo 1 session mới với user sys và startup database.
12. Mở 1 phiên làm việc mới với user HR. Ở phiên của sys, bật chức năng hạn chế truy cập. Ở user HR, có thực hiện truy vấn dữ liệu trong bản REGIONS được không?
13. Logout user hr, sau đó login lại, hỏi có thể login được không?
14. Tắt chế độ hạn chế truy cập và đăng nhập lại user hr.