# Oracle Database Document training

Quá trình họp tập đào tạo để quản trị Oracle Database sẽ phải trải qua 04 giai đoạn như sau:

1. SQL: Nắm vững ngôn ngữ SQL;
2. Oracle Administration Workshop I.rar: Là nội dung cơ bản về kiến trúc, các thành phần Oracle; Cài đặt Single và các thao tác Quản trị database cơ bản;
3. Oracle Administration Workshop II.rar: Là nội dung liên quan đến các Task quản trị nâng cao: như backup, restore, performance, moving data ….
4. Oracle RAC Administration 11g.rar: Nội dung nâng cao liên quan đến quản trị Oracle Cluster RAC;

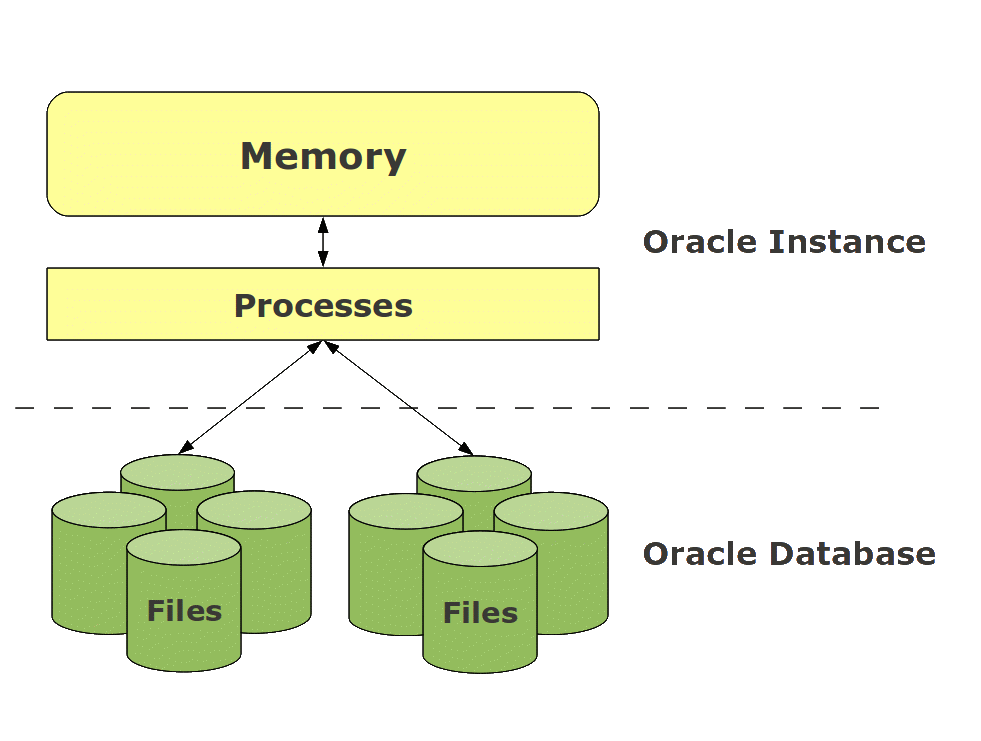
* Can bộ sẽ đọc phần bên dưới của tài liệu này sau đó đọc lần lượt và thực hành các bài trong các tài liệu 2,3,4; Còn về SQL thì mọi người tự học

# Oracle Basics 1 – Oracle Database vs Oracle Instance

Khi bắt đầu làm việc với Oracle, một trong những sự nhầm lẫn phổ biến nhất là giữa Oracle Database và Oracle Instance. Thuật ngữ Cơ sở dữ liệu Oracle chủ yếu được sử dụng để chỉ toàn bộ kiến trúc Oracle RDBMS. Tuy nhiên, xem xét kỹ hơn về Oracle RDBMS Server, chúng ta có thể thấy rằng nó bao gồm hai thực thể là Cơ sở dữ liệu và Instance. Điều gì thực sự là sự khác biệt giữa chúng? Hãy cùng tìm hiểu.

Bắt đầu làm việc với Oracle cũng là đối mặt với khá nhiều khái niệm mới. Hầu hết những người bắt đầu với Oracle đều có cùng một câu hỏi. Mấy năm trước mình cũng vậy :D. Vì vậy, tôi quyết định viết một số bài báo ngắn, nhằm trả lời một số câu hỏi cơ bản một cách ngắn gọn và đơn giản.

Đối với phần đầu tiên này, hãy quay lại các khái niệm Oracle Database và Oracle Instance của chúng ta. Hãy xem sơ đồ bên dưới.



Như chúng ta có thể thấy ở trên, kiến trúc Oracle RDBMS trải rộng trên 2 mặt phẳng: File và memory (bao gồm cả các tiến trình). Đó là nơi chúng ta thấy sự tách biệt giữa Oracle Instance và Oracle Database.

Oracle Instance bao gồm phần bộ nhớ và các tiến trình của Máy chủ RDBMS, trong khi Cơ sở dữ liệu Oracle bao gồm tất cả các File vật lý thuộc máy chủ (chính xác hơn là trên hệ thống lưu trữ).

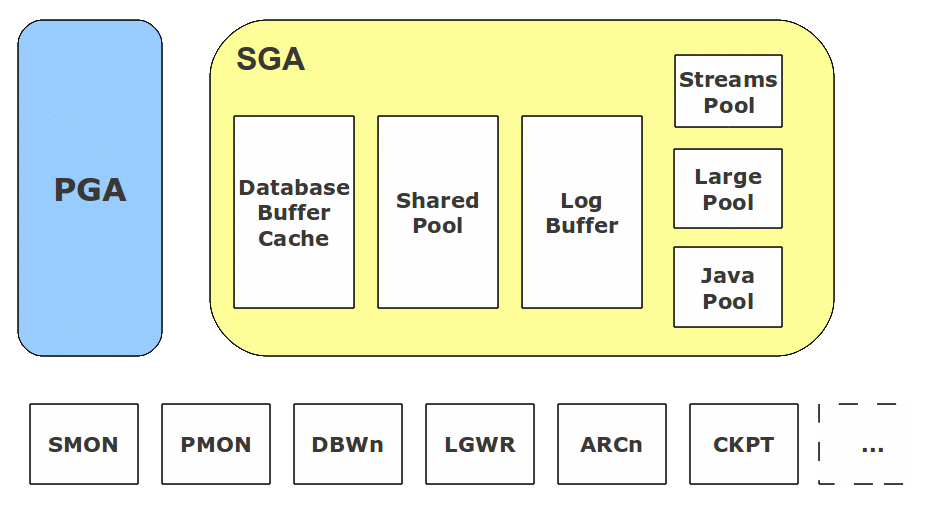
**What exactely is the Instance?**

Bây giờ chúng ta đã thấy sự khác biệt giữa Instance và Database, hãy xem chính xác Oracle Instance là gì.

Trong khi làm việc với Máy chủ Oracle RDBMS, Instance sẽ cung cấp giao diện giữa người dùng và dữ liệu mà anh ta muốn thao tác. Để làm như vậy, Instance cung cấp các tiến trình để liên lạc với client cũng như để truy cập dữ liệu. Tuy nhiên, các tiến trình này, còn được gọi là tiến trình nền (background), sẽ không đủ để cung cấp dịch vụ một cách hiệu quả và cing cấp đâì đủ các các chức năng của RDBMS như nguyên tắc ACID (Tính nguyên tử, Tính nhất quán, Cách ly và Độ bền). Do đó, Instance cũng sẽ có một số bộ nhớ đệm và bộ đệm.

Về cơ bản một instance bộ đệm của nó có thể được chia thành 3 phần:

* System Global Area (SGA)
* Program Global Area (PGA)
* Background processes



**SGA structure**

SGA là cấu trúc bộ nhớ dùng chung được cấp phát khi khởi động Instance và giải phóng khi shutdown instance. Ở mức tối thiểu, SGA sẽ chứa 3 cấu trúc dữ liệu sau:

* Database Buffer Cache(Bộ nhớ đệm cơ sở dữ liệu): Bộ nhớ cache của bộ đệm cơ sở dữ liệu là khu vực làm việc của Oracle. Khi thao tác với dữ liệu (update, insert…) Oracle thường không truy cập và chỉnh sửa dữ liệu trên disk. Đầu tiên, các khối dữ liệu được sao chép trong bộ nhớ cache của bộ đệm cơ sở dữ liệu và sau đó có thể được truy cập và/hoặc sửa đổi trong đó.
* Redo Log Buffer (Bộ đệm nhật ký làm lại): Bộ đệm nhật ký là khu vực tổ chức ngắn hạn cho các vectơ thay đổi trước khi chúng được ghi vào redo log. Vectơ thay đổi là một sửa đổi được áp dụng cho dữ liệu. Tất cả thông tin redo được ghi vào đĩa khi một phiên thực hiện các thay đổi của nó (và trong một số điều kiện khác). Các câu lệnh DML (Ngôn ngữ thao tác dữ liệu) (Insert, update, delete) của tất cả các phiên được lưu trong Redo Log Buffer để tránh tranh chấp I/O và các phiên chờ truy cập các file redo log (khái niệm về “piggy backing”): Khi các khối dữ liệu (data block – đơn vị lưu trữ nhỏ nhất trong cơ sở dữ liệu) được cập nhật, Oracle sẽ ghi lại những thay đổi dưới dạng các vector, các vector này được gọi là "redo record".  
  Redo log file là một file chứa các redo record, ghi lại những thay đổi của cơ sở dữ liệu. Redo log file sẽ bảo vệ tính toàn vẹn dữ liệu trong trường hợp cơ sở dữ liệu gặp trục trặc như: mất điện, hỏng ổ đĩa, … Cụ thể, sau khi cơ sở dữ liệu được phục hồi nhờ bản backup tại một thời điểm bất kỳ, Oracle sẽ apply các redo log file đã lưu trữ trong khoảng thời gian từ thời điểm đó đến lúc gặp trục trặc, giúp trạng thái của cơ sở dữ liệu đồng bộ với trạng thái ngay trước lúc xảy ra sự cố.
* Shared Pool: Là nhóm chia sẻ là cấu trúc SGA phức tạp nhất. Nó được chia thành nhiều cấu trúc con, được quản lý nội bộ bởi Máy chủ Oracle. Về cơ bản, cấu trúc của nhóm chia sẻ chứa thông tin như SQL được thực thi gần đây hoặc cấu trúc đối tượng được sử dụng gần đây.

Bên cạnh 3 nhóm bắt buộc, SGA cũng có thể chứa một số cấu trúc tùy chọn:

* Stream Pool: Stream Pool được dành riêng cho các chức năng của Oracle Stream. Thông tin thêm về Oracle Streams có thể được tìm thấy ở [đây](https://download.oracle.com/docs/cd/B19306_01/server.102/b14220/integrat.htm#sthref3334).
* Java Pool: Chỉ được yêu cầu nếu ứng dụng sử dụng cơ sở dữ liệu sẽ sử dụng các thủ tục lưu trữ Java trong cơ sở dữ liệu (Java stored procedures).
* Large Pool: Là một khu vực tùy chọn, sẽ tự động được sử dụng bởi một số tiến trình mà nếu không sẽ lấy bộ nhớ từ nhóm dùng chung. Large Pool sẽ được sử dụng cho các hoạt động lớn như tiến trình máy chủ dùng chung hoặc hoạt động RMAN.

**PGA structure**

Không giống như SGA có sẵn cho tất cả các tiến trình, PGA là một phần bộ nhớ riêng dành cho một phiên. Lưu ý mỗi tiến trình nền cũng có PGA riêng. PGA được phân bổ khi tạo phiên và được giải phóng sau khi phiên bị hủy.

**Background processes**

Ngoài cấu trúc bộ nhớ, Instance cũng có một số tiến trình với mức sử dụng chuyên dụng. Oracle có một danh sách dài các tiến trình nền, nhưng những tiến trình chính như sau:

• SMON

SMON chủ yếu chịu trách nhiệm gắn và mở cơ sở dữ liệu. Nó gắn kết cơ sở dữ liệu bằng cách định vị và xác thực các tệp điều khiển, đồng thời mở cơ sở dữ liệu bằng cách làm như vậy với các tệp dữ liệu và redo log. Ngoài ra, SMON còn phụ trách một số nhiệm vụ quản lý như giải phóng không gian trống trong các tệp dữ liệu. Nếu Instance gặp sự cố trong lần sử dụng cuối cùng, thì SMON được sử dụng để khôi phục cơ sở dữ liệu (quá trình khôi phục và khôi phục).

• PMON

Phiên người dùng bao gồm một tiến trình người dùng, ở phía máy Client và một tiến trình máy chủ ở phía máy chủ. PMON chịu trách nhiệm giám sát tất cả các tiến trình của máy chủ. Nếu nó phát hiện một phiên kết thúc bất ngờ, nó sẽ xử lý việc tắt tiến trình máy chủ, giải phóng bộ nhớ PGA và khôi phục giao dịch đang chờ xử lý.

• DBWn

Quá trình Database Writer quản lý quyền truy cập ghi vào các tệp dữ liệu. Nó sẽ ghi các khối dữ liệu từ bộ nhớ cache của bộ đệm cơ sở dữ liệu vào đĩa. Tùy thuộc vào tải của cơ sở dữ liệu, một số tiến trình DBWn có thể chạy song song.

• LGWR

Là DBWn, chịu trách nhiệm quản lý I/O với các tệp dữ liệu, LGWR quản lý chúng cho nhật ký làm lại trực tuyến. Nó truyền nội dung của bộ đệm redo log tới redo log online.

• ARCn

Tiến Trình lưu trữ sao chép redo log vào bộ lưu trữ được chỉ định sau mỗi lần chuyển đổi log (switch log) dưới dạng tệp nhật ký được lưu trữ. Một phiên bản có thể có tối đa 10 tiến trình lưu trữ. Các tiến trình lưu trữ chỉ tồn tại nếu cơ sở dữ liệu ở chế độ archive log.

• CKPT

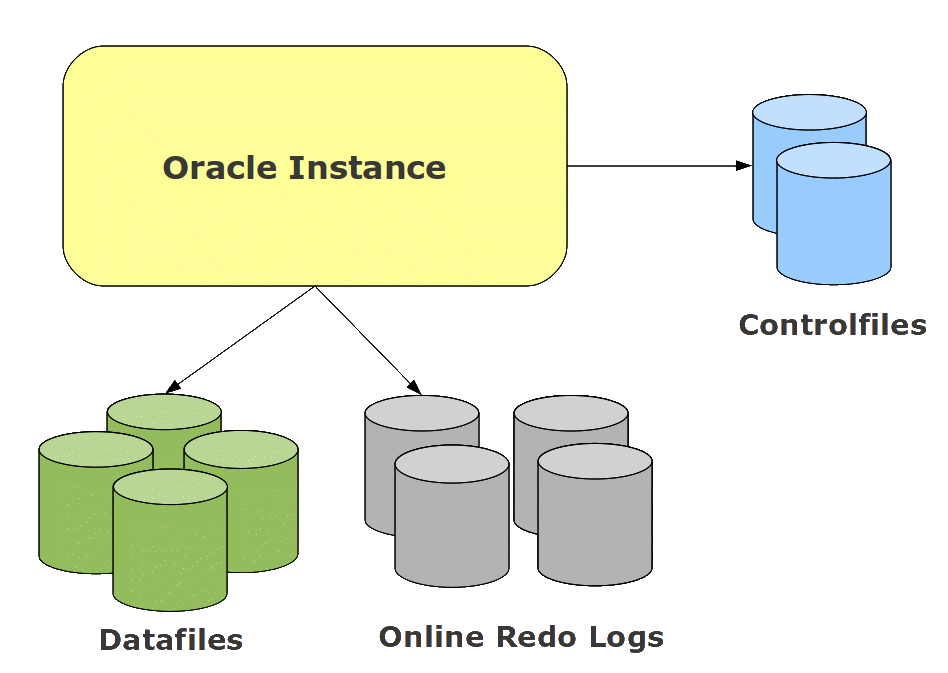
Về lý thuyết các khối đã sửa đổi thường xuyên được DBWn ghi trên các tệp dữ liệu. CKPT này được gọi là điểm kiểm tra. Quá trình điểm kiểm tra chịu trách nhiệm hướng dẫn DBWn tại điểm kiểm tra và cập nhật tiêu đề tệp điều khiển và tệp dữ liệu với điểm kiểm tra gần đây nhất (check point)

**What exactely is the Database?**

Cơ sở dữ liệu Oracle là phần lưu trữ của máy chủ RDBMS. Người ta phân biệt giữa lưu trữ vật lý và lưu trữ logic.

Cơ sở dữ liệu bao gồm 3 loại tệp:

* tập tin điều khiển (controlfile)
* nhật ký làm lại trực tuyến (online redo log)
* tập tin dữ liệu (datafile)



**Controlfiles**

Tệp điều khiển nhỏ nhưng rất quan trọng! Nó trỏ tới tất cả các phần tử cơ sở dữ liệu khác như tệp dữ liệu và online redo log. Nó cũng chứa thông tin để duy trì tính toàn vẹn của cơ sở dữ liệu (Số thay đổi hệ thống – SCN, dấu thời gian,…) và tất cả thông tin sao lưu và phục hồi, như nhật ký lưu trữ và sao lưu cơ sở dữ liệu mới nhất. Để bảo mật cơ sở dữ liệu, tệp điều khiển phải được ghép kênh. Nó có thể xử lý lên đến 8 bản sao.

**Online Redo Logs**

Online redo log lưu trữ tất cả các vectơ thay đổi chạy trên cơ sở dữ liệu theo chuỗi thời gian. Thông tin này là cần thiết để xây dựng lại db (=làm lại) tất cả các sửa đổi được thực hiện. Nó sẽ được sử dụng trong trường hợp datafile hoặc khôi phục cơ sở dữ liệu hoàn chỉnh. Cơ sở dữ liệu Oracle chạy ít nhất với 2 Online redo log, nhưng đối với tệp điều khiển, đáng để ghép chúng lại. Online redo log được sắp xếp theo nhóm, trong đó LGWR ghi trên cơ sở tuần hoàn (circular basis). Dịch vụ dbi khuyên nên sử dụng ít nhất 3 nhóm với 2 thành viên mỗi nhóm. Điểm mà LGWR di chuyển từ nhóm này sang nhóm khác được gọi là Log Switch. Nếu cơ sở dữ liệu ở chế độ archivelog, thì Online redo log hiện tại được lưu trữ bởi các quy trình ARCn tại mỗi chuyển đổi nhật ký.

**Datafiles**

Các tệp dữ liệu là kho lưu trữ dữ liệu. Cơ sở dữ liệu Oracle yêu cầu ít nhất 2 tệp dữ liệu, tệp này phải được tạo khi tạo cơ sở dữ liệu. Đây là khi liên kết giữa bộ nhớ logic và bộ nhớ vật lý được tạo. Các tệp dữ liệu là các cấu trúc vật lý mà quản trị viên hệ thống có thể nhìn thấy. Về mặt logic, dữ liệu người dùng được lưu trữ trong các không gian bảng, được tổ chức theo các phân đoạn, thường là các bảng, chỉ mục và các đối tượng lớn (nhị phân hoặc ký tự). Các tệp dữ liệu có thể được đổi tên, thay đổi kích thước, thêm, di chuyển hoặc xóa bất kỳ lúc nào, nhưng hãy nhớ rằng một số thao tác yêu cầu thời gian chết.

# Oracle Basics 2 – Data Storage

Cuối cùng thì mục tiêu chính của cơ sở dữ liệu là gì? Lưu trữ dữ liệu

Câu hỏi đặt ra là Oracle tạo ra nó như thế nào. Đối với nhiều người mới bắt đầu, một chủ đề khó hiểu là hiểu được bức tranh giữa bộ nhớ vật lý và bộ nhớ logic. Vì vậy, làm thế nào nó hoạt động?

Việc lưu trữ trong cơ sở dữ liệu Oracle có thể được nhìn thấy từ 2 quan điểm:

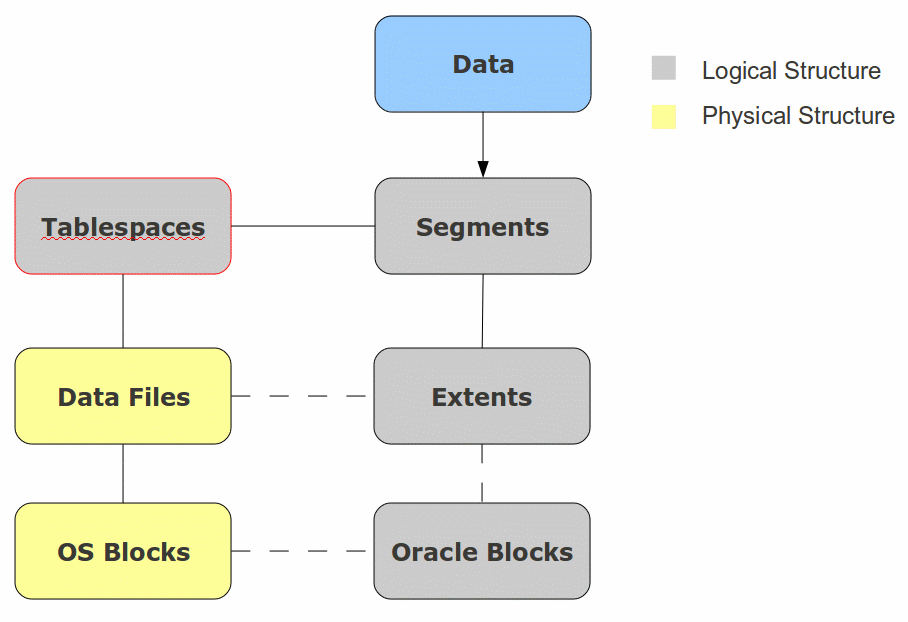
• Người dùng cơ sở dữ liệu

• Quản trị hệ thống

Một mặt, bạn có những người dùng xem bảng, index, v.v. Điều này có thể được tóm tắt và gọi chung là các đối tượng hoặc, được gọi là phân đoạn (Segment) trong Oracle, Mặt khác, có những người quản trị hệ thống (Unix hoặc Windows, không quan trọng…), những người xem các tệp và blocks.

Với vai trò là DBA, ta sẽ là liên kết còn thiếu giữa 2 quan điểm này

Vì vậy, làm thế nào để chúng ta đi xuống từ một Đối tượng, Phân đoạn, đến các tệp và khối hệ điều hành? Hãy xem phần bên dưới:



Như chúng ta đã nói, dữ liệu được lưu trữ trong Segment (Bảng, Index…). Các Segment được lưu trữ trong Table space và đây là cầu nối giữa lưu trữ logic và lưu trữ vật lý.

Một mặt, Table space là tập hợp vật lý của các tệp dữ liệu, được tạo từ các khối disk block trên hệ điều hành. Mặt khác, các Table space là tập hợp logic của các Segment (Bảng, chỉ mục…). Gần như có thể so sánh chúng với các thư mục nơi bạn sắp xếp dữ liệu của mình.

Các Segment bao gồm các Mức độ, là các phần dữ liệu. Vì Segment có thể sẽ phát triển theo thời gian nên Extents sẽ được thêm vào để tuân theo sự gia tăng kích thước của Segment. Extents đưcọ thực hiện ở bức độ Oracle data block

Cuối cùng, chúng tôi đạt đến cấp độ thấp nhất, trong đó Oracle Blocks là các nhóm Operating System Blocks. Kích thước khối Oracle được xác định ở cấp cơ sở dữ liệu nhưng cũng có thể được tùy chỉnh trên mỗi vùng bảng. Kích thước khối Oracle phải lớn hơn hoặc bằng và là bội số của các khối OS. Đây là một điểm cơ bản khi nói về hiệu suất của Oracle. Nhiều thực hành tốt nhất có thể được tìm thấy trên Internet về nó.

Hãy lấy một ví dụ nhỏ:

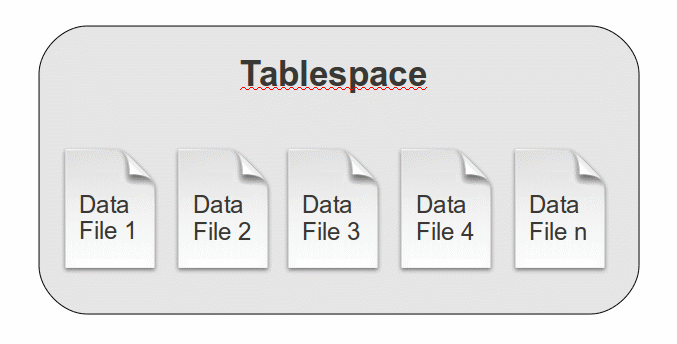
1 Cơ sở dữ liệu với Khối Oracle 8KB trên Hệ thống Tệp có khối 4KB.

==> 1 Khối Oracle = 2 Khối hệ điều hành

Bây giờ chúng ta có thể xem xét sâu hơn một chút về tất cả các yếu tố này

**Tablespaces and Data Files**

Như đã thấy, Tablespaces là các thùng chứa logic bao gồm một hoặc nhiều tệp dữ liệu.



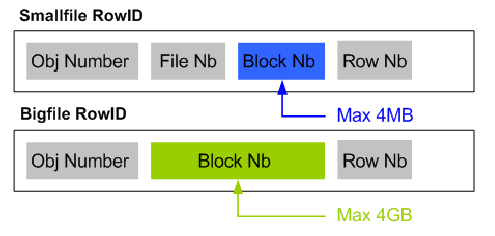
Oracle cung cấp 2 loại Tablespace:

• Smallfile tablespaces – được sử dụng phổ biến nhất

• Bigfile Tablespace

Các Tablespace Smallfile bao gồm một số tệp dữ liệu, trong khi Bigfiles bao gồm một tệp dữ liệu duy nhất. Một trong những lý do tại sao Oracle giới thiệu Bigfile Tablespace lớn là giới hạn về số lượng tệp dữ liệu cấu thành cơ sở dữ liệu (thường là 64 nghìn). Trong các Bigfile Tablespace, giới hạn kích thước tối đa của tệp dữ liệu cao hơn nhiều so với các Smallfile tablespaces.

Làm thế nào nó hoạt động? Bạn có thể đã nghe nói về ROWID, đây là một ID duy nhất cho mỗi hàng trong cơ sở dữ liệu. Trên thực tế, ROWID không chỉ là ID, nó còn là con trỏ tới vị trí hàng. Giữa các vùng bảng Smallfiles và Bigfiles, ROWID hơi khác một chút.



Theo định nghĩa, các không gian bảng Bigfile được tạo từ một tệp duy nhất, không cần Nb tệp trong ROWID. Do đó, Khối Nb được mã hóa trên 4GB thay vì 4kB.

Nếu chúng ta lấy lại ví dụ về cơ sở dữ liệu 8kB, chúng ta có những điều sau:

Kích thước tối đa của tệp nhỏ: khối 4kB X 8kB (kích thước của mỗi khối) = 32 GB mỗi tệp

Kích thước tối đa của Bigfile: khối 4GB X 8kB (kích thước của mỗi khối) = 32 TB mỗi tệp

Việc quản lý tablespaces có thể được thực hiện trong Từ điển dữ liệu Oracle hoặc, kể từ 10g, được quản lý cục bộ. Ngày nay, hầu hết tất cả các tablespaces đều được cấu hình là tablespaces được quản lý cục bộ (LMTS). Điều này có nghĩa là việc quản lý tablespaces được thực hiện trong một bitmap được lưu trữ trong Segment Header của tablespace. LMTS có ưu điểm là giảm – nhưng không tránh được – sự phân mảnh và cải thiện hiệu suất DDL.

Trong khi tạo tệp dữ liệu, có thể thực hiện được 2 giải pháp. Có thể tạo chúng với kích thước cố định sẽ được phân bổ khi tạo tệp hoặc có thể tạo chúng ở chế độ tự động mở rộng. Các tệp dữ liệu Autoextend được tạo với kích thước ban đầu và sau đó sẽ tự động tăng lên cho đến kích thước tối đa được xác định. Nếu không có kích thước tối đa nào được cung cấp khi tạo tệp dữ liệu, thì giới hạn được xác định bởi Kích thước khối Oracle (xem ở trên về ROWID và kích thước tệp dữ liệu tối đa) sẽ được áp dụng.

Đây là 2 view chính trong oracle để lấy thông tin về tablespaces và data files: dba\_tablespaces, dba\_data\_files

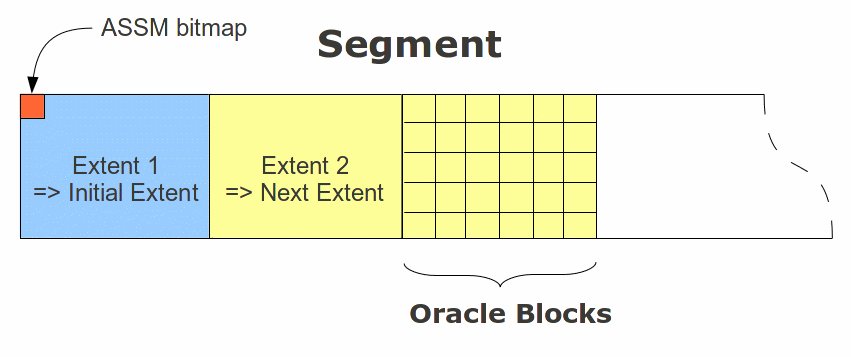
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43 | SQL> desc dba\_tablespaces;  Name                       Null?         Type  -------------------------- ------------- ------------------------------------  TABLESPACE\_NAME            NOT NULL      VARCHAR2(30)  BLOCK\_SIZE                 NOT NULL      NUMBER  INITIAL\_EXTENT                           NUMBER  NEXT\_EXTENT                              NUMBER  MIN\_EXTENTS                NOT NULL      NUMBER  MAX\_EXTENTS                              NUMBER  MAX\_SIZE                                 NUMBER  PCT\_INCREASE                             NUMBER  MIN\_EXTLEN                               NUMBER  STATUS                                   VARCHAR2(9)  CONTENTS                                 VARCHAR2(9)  LOGGING                                  VARCHAR2(9)  FORCE\_LOGGING                            VARCHAR2(3)  EXTENT\_MANAGEMENT                        VARCHAR2(10)  ALLOCATION\_TYPE                          VARCHAR2(9)  PLUGGED\_IN                               VARCHAR2(3)  SEGMENT\_SPACE\_MANAGEMENT                 VARCHAR2(6)  DEF\_TAB\_COMPRESSION                      VARCHAR2(8)  RETENTION                                VARCHAR2(11)  BIGFILE                                  VARCHAR2(3)  PREDICATE\_EVALUATION                     VARCHAR2(7)  ENCRYPTED                                VARCHAR2(3)  COMPRESS\_FOR                             VARCHAR2(12)  SQL> desc dba\_data\_files;   Name                       Null?    Type   -------------------------- -------- -------------------------------   FILE\_NAME                           VARCHAR2(513)   FILE\_ID                             NUMBER   TABLESPACE\_NAME                     VARCHAR2(30)   BYTES                               NUMBER   BLOCKS                              NUMBER   STATUS                              VARCHAR2(9)   RELATIVE\_FNO                        NUMBER   AUTOEXTENSIBLE                      VARCHAR2(3)   MAXBYTES                            NUMBER   MAXBLOCKS                           NUMBER   INCREMENT\_BY                        NUMBER   USER\_BYTES                          NUMBER   USER\_BLOCKS                         NUMBER   ONLINE\_STATUS                       VARCHAR2(7) |

**Segments and Extents**

Phân đoạn (Segment) là tên chung được sử dụng trong cơ sở dữ liệu Oracle để đại diện cho các đối tượng như bảng, chỉ mục hoặc phân vùng. Chúng được lưu trữ trong Tệp dữ liệu theo từng phần được gọi là Extents.

Phân đoạn có thể ở chế độ THỦ CÔNG hoặc ở chế độ TỰ ĐỘNG – Automatic Segment Space Management (ASSM)

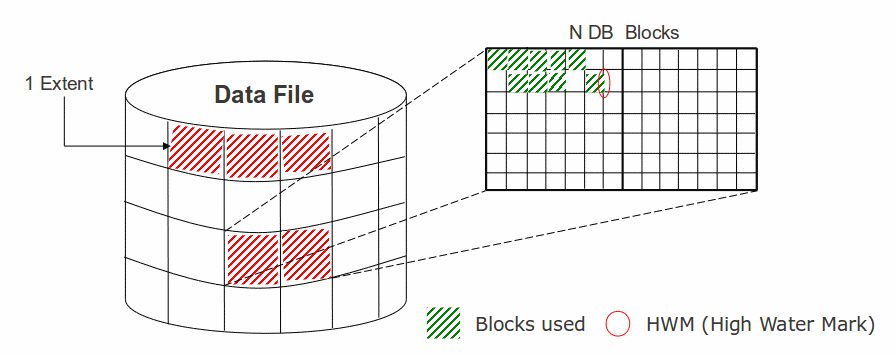
Trong phiên bản Oracle trước đó, chế độ MANUAL quản lý các khối trống và không gian trống trong Danh sách Free List lưu trữ Từ điển dữ liệu, Nó làm quá tải tablespace hệ thống. Kể từ Oracle 10g và ASSM, các khối trống và không gian trống được quản lý trong một bitmap trong Tiêu đề phân đoạn của mỗi Tablespace.



The 2 main views to find segments and extents information are: dba\_segments and dba\_extents

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42 | SQL> desc dba\_segments  Name                       Null?           Type  -------------------------- --------------- -----------------------------------  OWNER                                      VARCHAR2(30)  SEGMENT\_NAME                               VARCHAR2(81)  PARTITION\_NAME                             VARCHAR2(30)  SEGMENT\_TYPE                               VARCHAR2(18)  SEGMENT\_SUBTYPE                            VARCHAR2(10)  TABLESPACE\_NAME                            VARCHAR2(30)  HEADER\_FILE                                NUMBER  HEADER\_BLOCK                               NUMBER  BYTES                                      NUMBER  BLOCKS                                     NUMBER  EXTENTS                                    NUMBER  INITIAL\_EXTENT                             NUMBER  NEXT\_EXTENT                                NUMBER  MIN\_EXTENTS                                NUMBER  MAX\_EXTENTS                                NUMBER  MAX\_SIZE                                   NUMBER  RETENTION                                  VARCHAR2(7)  MINRETENTION                               NUMBER  PCT\_INCREASE                               NUMBER  FREELISTS                                  NUMBER  FREELIST\_GROUPS                            NUMBER  RELATIVE\_FNO                               NUMBER  BUFFER\_POOL                                VARCHAR2(7)  FLASH\_CACHE                                VARCHAR2(7)  CELL\_FLASH\_CACHE                           VARCHAR2(7)  SQL> desc dba\_extents  Name                       Null?           Type  -------------------------- --------------- ------------------------------------  OWNER                                      VARCHAR2(30)  SEGMENT\_NAME                               VARCHAR2(81)  PARTITION\_NAME                             VARCHAR2(30)  SEGMENT\_TYPE                               VARCHAR2(18)  TABLESPACE\_NAME                            VARCHAR2(30)  EXTENT\_ID                                  NUMBER  FILE\_ID                                    NUMBER  BLOCK\_ID                                   NUMBER  BYTES                                      NUMBER  BLOCKS                                     NUMBER  RELATIVE\_FNO                               NUMBER |

Một khái niệm quan trọng khác cần hiểu, trong trường hợp phân đoạn Bảng (Table segments), là “Hình mờ cao” (HWM). Nó xác định vị trí của khối được hình thành cuối cùng cho segment. Điều đó có nghĩa là trong trường hợp Quét toàn bộ bảng (FTS – tức là chọn \* từ bảng1;) Oracle sẽ đi qua tất cả các khối của phân đoạn cho đến vị trí HWM.



Chìa khóa ở đây là hãy nhớ rằng việc xóa dữ liệu, nghĩa là các khối Free, KHÔNG di chuyển vị trí HWM. Điều này có thể có tác động khá lớn đến hiệu suất yêu cầu.

Cách duy nhất để di chuyển xuống HWM là cắt bớt bảng (truncate table), nhưng điều này cũng có nghĩa là mất tất cả dữ liệu hoặc thực hiện sắp xếp lại bảng. Tuy nhiên, điều này hoàn toàn nằm ngoài chủ đề hiện tại của chúng ta (Vì vậy, ở đây chúng tôi giới thiệu phần giới thiệu về lưu trữ dữ liệu Oracle)

**Oracle Basics** 3 **– Database startup and shutdown**

Startup and shutdown processes within oracle database servers are split in different steps (for startup) and modes (for shutdown) and even if these principles are quite trivial, some confusions between these steps and/or modes can easily be done. So I’m going to give a picture of both of these processes and their variations.

Before shutting any database server down, it needs first to be started, so let’s begin with the startup process…

Note that to avoid confusion, I use the term database server to designate the entity composed from both the Database and the Instance.

**Starting a database server: a 3 steps process**

Before understanding the process itself, the first question is “How to start a database server?” The answer is that you need to connect locally to it and to be a SYSDBA!  
So then the next question is “How to connect to something which is NOT running yet?” To do so you need 3 environment variables:

ORACLE\_HOME –> Gives the Oracle Binaries Path  
ORACLE\_SID  –> Gives the Instance SID to connect to  
PATH –> Must contain $ORACLE\_HOME/bin or %ORACLE\_HOME%bin (Windows)

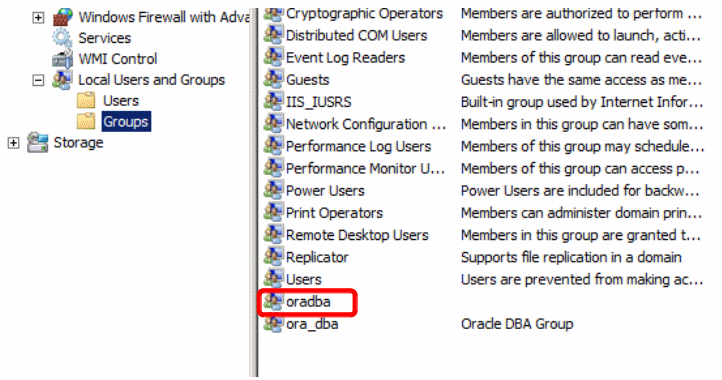
|  |  |
| --- | --- |
| 1  2  3  4 | oracle@vmoratest1:/home/oracle/ [TYRION] echo $ORACLE\_HOME  /u00/app/oracle/product/11.2.0/db\_1  oracle@vmoratest1:/home/oracle/ [TYRION] echo $ORACLE\_SID  TYRION |

Then you have to be on the server and be part of the DBA Operating System group. In UNIX world this corresponds to the group “dba”, while in Windows it is “oradba”.

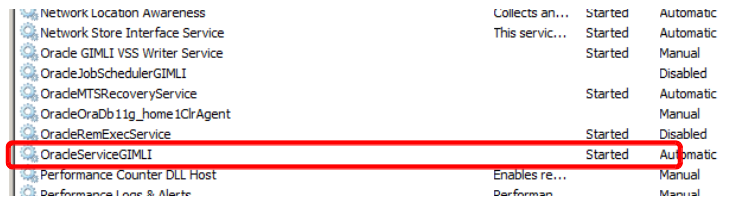
**Unix**:

|  |  |
| --- | --- |
| 1  2 | oracle@vmoratest1:/home/oracle/ [TYRION] id  uid=500(oracle) gid=501(dba) groups=501(dba),502(oinstall) |

**Windows**:



In Windows environment, you need also that the Instance Service is started.



Finally, start SQLPLUS and simply connect as internal: sqlplus / as sysdba    where the / means to connect using OS authentication (DBA group) on the Instance defined by the variable ORACLE\_SID.

|  |  |
| --- | --- |
| 1  2  3  4 | oracle@vmoratest1:/home/oracle/ [TYRION] sqlplus / as sysdba  SQL\*Plus: Release 11.2.0.1.0 Production on Wed May 18 09:02:07 2011  Copyright (c) 1982, 2009, Oracle.  All rights reserved.  Connected to an idle instance. |

Once connected, simply run the command: startup

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | <SQL> startup  ORACLE instance started.  Total System Global Area  732352512 bytes  Fixed Size            1339036 bytes  Variable Size          440402276 bytes  Database Buffers      285212672 bytes  Redo Buffers            5398528 bytes  Database mounted.  Database opened.  SQL> select open\_mode from v$database;  OPEN\_MODE  --------------------  READ WRITE |

This is it! you database server is started   
So now let’s have a looking what is going done backstage.

Basically starting an Oracle database is divided in 3 steps:

* NOMOUNT
* MOUNT
* OPEN

For each of them there is a corresponding startup command:

* NOMOUNT => startup nomount
* MOUNT   => startup mount
* OPEN    => startup

As you can see it the startup command automatically takes the database in Open mode. However some operations, like restores and recovers require to decompose the startup process. In this case swithing up from one mode to the next one is done using an “alter database” command.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | SQL> startup nomount  ORACLE instance started.  Total System Global Area  732352512 bytes  Fixed Size            1339036 bytes  Variable Size          440402276 bytes  Database Buffers      285212672 bytes  Redo Buffers            5398528 bytes  SQL> alter database mount;  Database altered.  SQL> alter database open;  Database altered.  SQL> select open\_mode from v$database;  OPEN\_MODE  --------------------  READ WRITE |

Note that you can NOT switch a database one mode/step down. This requires a shutdown!

Here we need to open a small bracket for the databases running on Windows servers.  
As said above, on Windows the service OracleService must be running to connect to the database. However starting this service will fire up a full startup of the database, which means that a healthy database will be fully open once the service is started. On databases requiring restore/recover operations, you will basically also need to start the service and let him failing to open the database before you can go ahead.

In order to really understand the logic behind the startup process, you need to keep in mind the difference between the Oracle Instance and the Oracle Database. You can have a look to my [first article](https://blog.dbi-services.com/oracle-basics-1-oracle-database-vs-oracle-instance) on Oracle Basis to get a refresh on it.

**Step 1 – NOMOUNT**

In the NOMOUNT level, only the Oracle INSTANCE is started. To do so Oracle looks for the Instance spfile or pfile and starts all Processes and Memory Structures.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | SQL> startup nomount  ORACLE instance started.  Total System Global Area  732352512 bytes  Fixed Size            1339036 bytes  Variable Size          440402276 bytes  Database Buffers      285212672 bytes  Redo Buffers            5398528 bytes |

Let’s have a look on the running processes:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | oracle@vmoratest1:/home/oracle/ [TYRION] ps -ef |grep TYRION  oracle    3999     1  0 09:12 ?        00:00:00 ora\_pmon\_TYRION  oracle    4001     1  0 09:12 ?        00:00:00 ora\_vktm\_TYRION  oracle    4005     1  0 09:12 ?        00:00:00 ora\_gen0\_TYRION  oracle    4007     1  0 09:12 ?        00:00:00 ora\_diag\_TYRION  oracle    4009     1  0 09:12 ?        00:00:00 ora\_dbrm\_TYRION  oracle    4011     1  0 09:12 ?        00:00:00 ora\_psp0\_TYRION  oracle    4013     1  0 09:12 ?        00:00:00 ora\_dia0\_TYRION  oracle    4015     1  1 09:12 ?        00:00:00 ora\_mman\_TYRION  oracle    4017     1  0 09:12 ?        00:00:00 ora\_dbw0\_TYRION  oracle    4019     1  0 09:12 ?        00:00:00 ora\_lgwr\_TYRION  oracle    4021     1  0 09:12 ?        00:00:00 ora\_ckpt\_TYRION  oracle    4023     1  0 09:12 ?        00:00:00 ora\_smon\_TYRION  oracle    4025     1  0 09:12 ?        00:00:00 ora\_reco\_TYRION  oracle    4027     1  0 09:12 ?        00:00:00 ora\_mmon\_TYRION  oracle    4029     1  0 09:12 ?        00:00:00 ora\_mmnl\_TYRION  oracle    4031     1  0 09:12 ?        00:00:00 ora\_d000\_TYRION  oracle    4033     1  0 09:12 ?        00:00:00 ora\_s000\_TYRION  oracle    4106  3356  0 09:12 pts/1    00:00:00 grep TYRION |

Here we can recognize processes like SMON, LGWR, DBW or PMON

The question now is, how Oracle find the spfile?  
In fact Oracle goes automatically in the dbs (UNIX) or database (Windows) folder in ORACLE\_HOME and look for a spfile spfile.ora. If there is no spfile Oracle will the search for a pfile, init.ora, and finally take a default spfile if none are present. A good practice, OFA (Oracle Flexible Architecture), is to store the spfile in the admin directory of the database and to use a pfile to point to it.

Here is a small example:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | oracle@vmoratest1:/home/oracle/ [TYRION] cd $ORACLE\_HOME/dbs  oracle@vmoratest1:/u00/app/oracle/product/11.2.0/db\_1/dbs/ [TYRION] ls  hc\_DBUA0.dat  hc\_TYRION.dat  init.ora  initTYRION.ora  lkULTHAN\_SITE1  orapwTYRION  peshm\_DBUA0\_0  peshm\_ULTHAN\_SITE1\_0  oracle@vmoratest1:/u00/app/oracle/product/11.2.0/db\_1/dbs/ [TYRION] cat initTYRION.ora  SPFILE='/u00/app/oracle/admin/TYRION/pfile/spfileTYRION.ora'  oracle@vmoratest1:/u00/app/oracle/product/11.2.0/db\_1/dbs/ [TYRION] cd /u00/app/oracle/admin/TYRION/pfile/  oracle@vmoratest1:/u00/app/oracle/admin/TYRION/pfile/ [TYRION] ls  spfileTYRION.ora |

On a instance in NOMOUNT mode, it is only possible to access and interact to the spfile information

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | SQL> show parameter name  NAME                             TYPE       VALUE  -------------------------------- ---------- ------------------------------  db\_file\_name\_convert             string  db\_name                          string     ULTHAN  db\_unique\_name                   string     ULTHAN\_SITE1  global\_names                     boolean    FALSE  instance\_name                    string     TYRION  lock\_name\_space                  string  log\_file\_name\_convert            string  service\_names                    string     TYRION.it.dhu-domain.com  SQL> show parameter spfile  NAME                       TYPE       VALUE  ------------------------- ----------- ------------------------------  spfile                     string     /u00/app/oracle/admin/TYRION/pfile/spfileTYRION.ora |

and to access some instance views

|  |  |
| --- | --- |
| 1  2  3  4 | SQL> select INSTANCE\_NAME,HOST\_NAME,STATUS from v$instance;  INSTANCE\_NAME      HOST\_NAME                         STATUS  ----------------- --------------------------------- ------------  TYRION             vmoratest1.it.dhu-domain.com      STARTED |

The startup\_mode column permits to identify the current mode of the database server. Here STARTED means in NOMOUNT.

**Step 2 – MOUNT**

Once taking the database server from NOMOUNT to MOUNT, Oracle access the control files, as declared in the spfile, and mount the database.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | SQL> show parameter control\_files  NAME                TYPE          VALUE  ------------------- ------------- ----------------------------------------------------  control\_files       string        /u01/oradata/TYRION/control01TYRION.dbf, /u02/oradata                                    /TYRION, /control02TYRION.dbf,/u03/oradata/TYRION                                    /control03TYRION.dbf  SQL> alter database mount;  Database altered. |

Check the new status

|  |  |
| --- | --- |
| 1  2  3  4  5 | SQL> select INSTANCE\_NAME,HOST\_NAME,STATUS from v$instance;    INSTANCE\_NAME      HOST\_NAME                         STATUS  ----------------- --------------------------------- ------------  TYRION             vmoratest1.it.dhu-domain.com      MOUNTED |

Let’s check the control files usage:

|  |  |
| --- | --- |
| 1  2 | [root@vmoratest1 TYRION]# fuser -a ./\*  ./control01TYRION.dbf:  4017  4019  4021 |

At this point Oracle knows information like the files composing the databases and their location, the backup history, the SCN status aso… SYSDBA users are still the own one, who can access the database, but now they have access to several information stored in v$views (thanks to the control file).

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | SQL> select open\_mode from v$database;  OPEN\_MODE  --------------------  MOUNTED  SQL> select name from v$datafile;  NAME  --------------------------------------------------------------------------------  /u01/oradata/TYRION/system01TYRION.dbf  /u01/oradata/TYRION/sysaux01TYRION.dbf  /u01/oradata/TYRION/undotbs01TYRION.dbf  /u01/oradata/TYRION/users01TYRION.dbf  SQL> select member from v$logfile;  MEMBER  --------------------------------------------------------------------------------  /u01/oradata/TYRION/redog1m1TYRION.dbf  /u02/oradata/TYRION/redog1m2TYRION.dbf  /u01/oradata/TYRION/redog2m1TYRION.dbf  /u02/oradata/TYRION/redog2m2TYRION.dbf  /u01/oradata/TYRION/redog3m1TYRION.dbf  /u02/oradata/TYRION/redog3m2TYRION.dbf  6 rows selected. |

**Step 3 – OPEN**

The last step is to open the database, which means accessing the data files and redo logs. Once the database is OPEN, all users can log on again and go ahead with their normal activities.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | SQL> alter database open;  Database altered.  SQL> select status,open\_mode from v$instance,v$database;  STATUS         OPEN\_MODE  -------------- --------------------  OPEN           READ WRITE |

**Some variations**

For some maintenance or administration activities, it may be necessary to open the database but to avoid any user to access it. The best solution is then to open the database in RESTRICT mode. This means that only SYSDBA users will be allowed to log in.

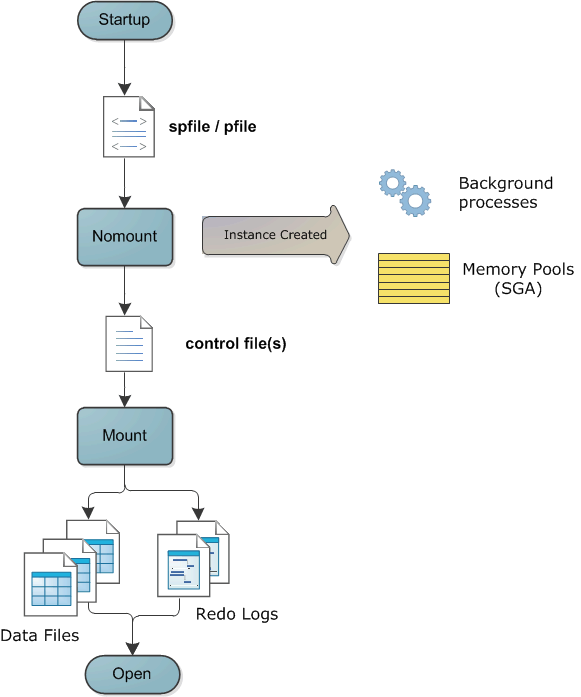
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | SQL> startup restrict  ORACLE instance started.  Total System Global Area  732352512 bytes  Fixed Size            1339036 bytes  Variable Size          440402276 bytes  Database Buffers      285212672 bytes  Redo Buffers            5398528 bytes  Database mounted.  Database opened. |

A other possibility is to open the database in READ ONLY instead of READ WRITE.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | SQL> alter database open read only;  Database altered.  SQL> select open\_mode from v$database;  OPEN\_MODE  --------------------  READ ONLY |

**Startup process summary**

This drawing summarize the Oracle database startup process.



**Shutting down a database**

Sometimes it is necessary to shutdown a database server, like for example activating the archivelog or flashback mode, changing a static parameter or simply patching it. In this case, the DBA has 4 different solutions:

1. shutdown normal (default mode with shutdown command)   => shutdown
2. shutdown transactional    => shutdown transactional
3. shutdown immediate    => shutdown immediate
4. shutdown abort   => shutdown abort

These modes are going from the safer to the most risky one, but also from the slower to the faster one.

**Shutdown Normal**

This is the default. No new user connections will be allowed, but all current connections continue normaly. Once all users have logged off, the database will finally be allowed to shutdown.

**Shutdown Transactional**

No new user connections are permitted and existing sessions that are not involved in active transactions will be terminated. However sessions currently involved in a transaction are allowed to complete the transaction and will then be terminated. Once all sessions are terminated, the database will shutdown.

**Shutdown Immediate**

No new sessions are permitted, all currently connected sessions are  
terminated an any active transactions are rolled back. Then the database will go down.

**Shutdown Abort**

As far as Oracle is concerned, this is the equivalent of a power failure. The instance terminates immediately (instance “crash”). Nothing is written to disk, no file handles are closed and there is no transactions are terminated, even not in a orderly way. A shutdown abort will not damage the database, but some operations like backups are not advisable after an abort.

**Shutdown process description**

Basically in a “clean” shutdown process (Normal, Transactionnal or Immediate), the process will be the reverse of startup. During an orderly shutdown, the database is first be closed, then dismounted, and finally the instance is stopped.

During the close phase:

* all sessions are terminated
* PMON roll back any incomplete transactions.
* A checkpoint is issued, which forces the DBWn process to write all updated data from the db buffer cache to the datafiles
* LGWR flushes any change vectors still in memory to the logfiles
* The file headers are updated, and the file handles closed.

At this point the database is in a consistent state: all datafiles and logfiles are synchronized.  
During the dismount phase the control files are closed  
Then the instance is stopped by deallocating the SGA memory and terminating the background processes.  
In case of the Abort mode, it leaves the database in an inconsistent state:

* Committed transactions have been lost, because they were only in memory and DBWn had not yet written them to the datafiles
* Uncommitted transactions in the datafiles may not yet have been rolled back.

After a shutdown abort, the SMON process will have to perform an instance recovery at next startup.  
To finish, I will open a last bracket about Windows environment

I said that in Windows there is a service OracleService to start/stop the database. Basically you can stop your database just by stopping the corresponding service. In this case Oracle runs a SHUTDOWN IMMEDIATE by default. The type of shutdown the service runs can be customized, since Oracle 11g, using the parameter -SHUTMODE normal|immediate|abort of oradim.

However there is a small issue with this principle   
Unfortunately in case of a server shutdown/reboot, the service will be stopped before the database shutdown is  performed!

**Alert.log output**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | Fatal NI connect error 12638, connecting to:   (DESCRIPTION=(LOCAL=YES)(ADDRESS=(PROTOCOL=beq)))  VERSION INFORMATION:   TNS for 64-bit Windows: Version 11.2.0.2.0 - Production   Oracle Bequeath NT Protocol Adapter for 64-bit Windows: Version 11.2.0.2.0 - Production   Time: 25-MAY-2011 17:40:47   Tracing not turned on.   Tns error struct:   ns main err code: 12638  TNS-12638: Credential retrieval failed   ns secondary err code: 0   nt main err code: 0   nt secondary err code: 0   nt OS err code: 0  This means that the database is not shutdown but crashed! Therefore a instance recovery will be performed at next startup:    Completed: alter database mount exclusive  alter database open  Beginning crash recovery of 1 threads  Started redo scan  Completed redo scan  ...  ... |

A solution is to use the registry entry ORA\_SID\_SHUTDOWN\_TIMEOUT (HKEY\_LOCAL\_MACHINE –> Software –> Oracle –> Key\_) to theoricaly define a timeout while the service waits for the database shutdown confirmation before stopping. If the timeout is reached then it performs a shutdown abort.



And here comes now the Bug 1638610!

The parameter ORA\_SID\_SHUTDOWN\_TIMEOUT isn’t taken in account if the setting SQLNET.AUTHENTICATION\_SERVICES is set in the sqlnet.ora. This bug applies from Oracle 8i to 10g and is still true in Oracle 11gR2. The bug state is set by Oracle as “not feasible to fix”…

I hope that this small article, gave you a good understanding about Oracle startup and shutdown processes