はじめに

■コースの概要と目的

効率よく Oracle データベースを使用するための運用管理について、管理タスクを行ううえでの考慮事項や注意点を、実習を通して習得します。

■受講対象者

データベース管理者。

■前提条件

「データベース・アーキテクチャ」コースを受講された方、もしくは Oracle システム構成とデータベース構造に関する知識をお持ちの方。

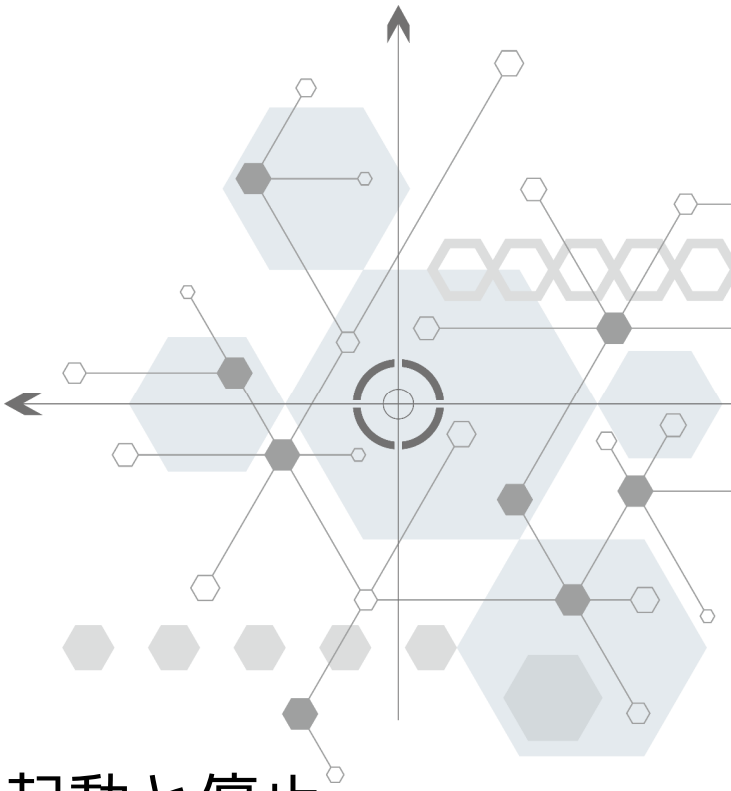
■テキスト内の記述について

▼構文

[ ]	省略可能
{ A | B }	A または B のどちらかを選択
n	数値の指定
_	デフォルト値

▼マーク

19	指定バージョンからの新機能 （左記の場合、Oracle 19c からの新機能）
EE	Enterprise Edition で使用できる機能
	注意事項
	参考情報
Technique	知っておくと便利なテクニック
	参照ページ
	データ・ディクショナリ・ビュー



第 2 章

データベースの起動と停止

この章では、メンテナンスで必要となるデータベースの起動と停止について説明します。

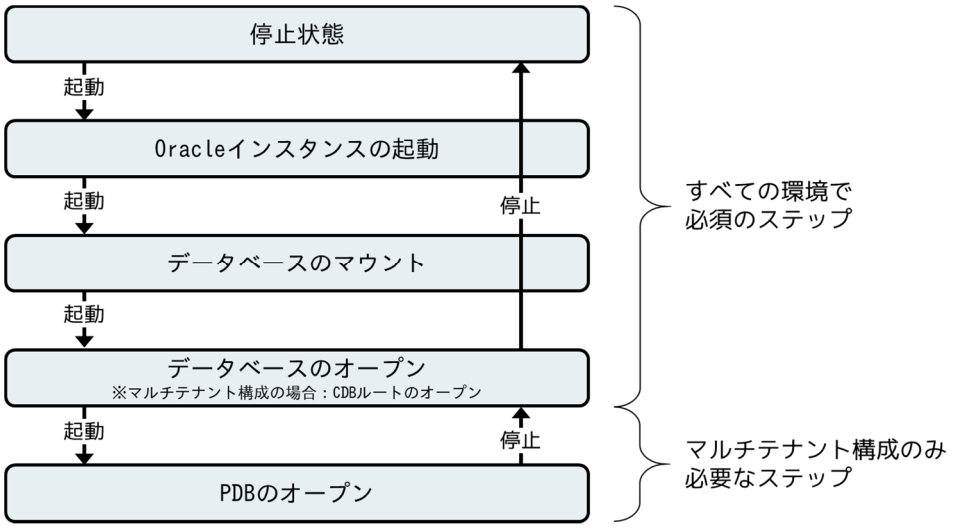
01　データベースの起動と停止の概要
02　データベースの起動
03　データベースの停止

01 データベースの起動と停止の概要

データベースは運用中オープン状態であり、一般ユーザーからのアクセスが可能な状態です。
しかし、データベースの管理作業によってはオープン状態のデータベースを停止したり、停止後に起動の各ステップでメンテナンスを行ったりする必要があります。

（1）起動と停止のステップ概要

データベースの起動と停止は、以下のようなステップを経て行われます。



・起動時の特徴
途中のステップまで起動することも、データベースのオープンまで一気に起動することも可能です。
※マルチテナント構成の場合は、CDB ルートをオープンした後で、PDB を個別にオープンします。

・停止時の特徴
起動時とは逆のステップを経て停止します。また、途中のステップで停止することはできません。
※マルチテナント構成の場合は、特定の PDB のみを停止し、CDB や他の PDB を起動させておくことが可能です。

Copyright (C) 2020 K.K.Ashisuto All Rights Reserved.
株式会社 アシスト 無断転載・複製禁止

（2）データベースの起動と停止を必要とする作業

データベースの起動と停止は、主に以下の作業にて必要になります。

■データベースの各状態で実行できる主な作業

データベースの状態	実行可能な作業	
	非マルチテナント構成	マルチテナント構成
停止	・オフライン・バックアップ	
Oracle インスタンスの起動	・データベースの作成 ・制御ファイルの再作成	
データベースのマウント	・アーカイブ・モードの切り替え ・データベースのリカバリ ・ファイルの改名、移動	・アーカイブ・モードの切り替え ・CDB ルートのリカバリ ・CDB ルートのファイルの改名
データベースのオープン（CDB ルートのオープン）	・通常運用（一般ユーザーのアクセス） ・オンライン・バックアップ	・CDB ルートのオンライン・バックアップ ・PDB のリカバリ ・PDB のファイルの改名、移動
PDB のオープン		・通常運用（一般ユーザーのアクセス） ・PDB のオンライン・バックアップ

データベース起動時のステップを理解することは、データベースの起動に失敗した際の原因究明にも役立ちます。

株式会社 アシスト 無断転載・複製禁止 Copyright (C) 2020 K.K.Ashisuto All Rights Reserved.

02 データベースの起動

Oracle を起動するには Oracle インスタンスの起動、データベースのマウント、データベースのオープンの各ステップを実行します。1 ステップごとに起動することも、データベースのオープンまで自動的に起動することも可能です。

（1）起動ステップ

データベースの起動は、以下のステップで行われます。

1）Oracle インスタンスの起動

Oracle インスタンスを起動するためにパラメータ設定ファイルから初期化パラメータの値を読み込みます。次にメモリー領域（SGA）を割り当て、バックグラウンド・プロセスを作成します。

■Oracle インスタンスの起動に失敗した場合に考えられる原因
・無効な初期化パラメータの値を設定している
・パラメータ設定ファイルが破損している

2）データベースのマウント

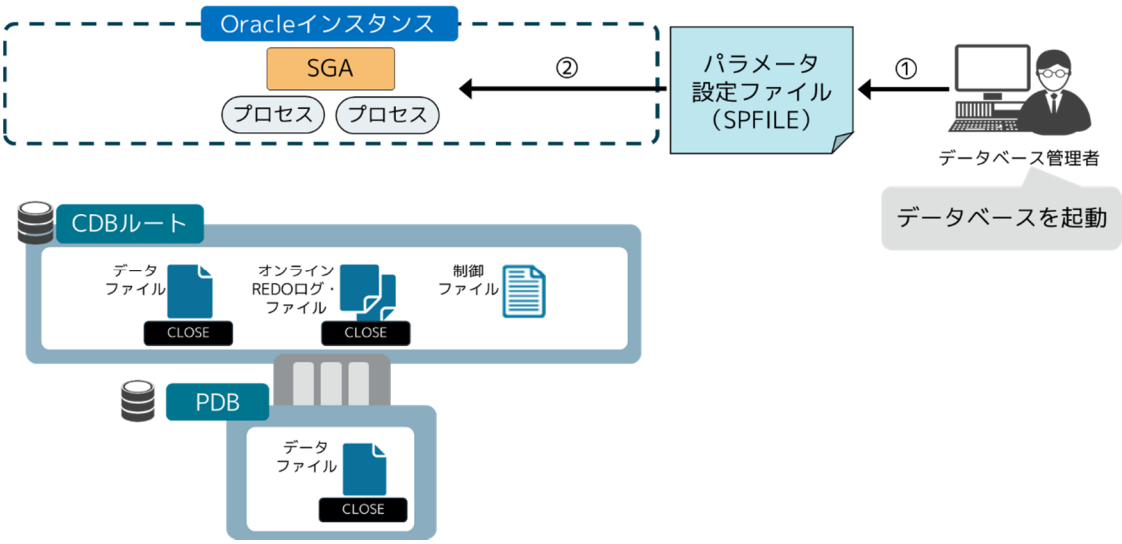
CONTROL_FILES パラメータの値にもとづいて制御ファイルを読み込み、データファイル、オンライン REDO ログ・ファイルのパスを検出します（Oracle インスタンスとデータファイル、オンライン REDO ログ・ファイルの対応づけ）。

■データベースのマウントに失敗した場合に考えられる原因
・制御ファイルが破損している
・制御ファイルが存在しない
・パラメータ設定ファイルに記載されている制御ファイルのパスが間違っている

Copyright (C) 2020 K.K.Ashisuto All Rights Reserved.　株式会社 アシスト 無断転載・複製禁止

＜データベースの起動ステップ＞

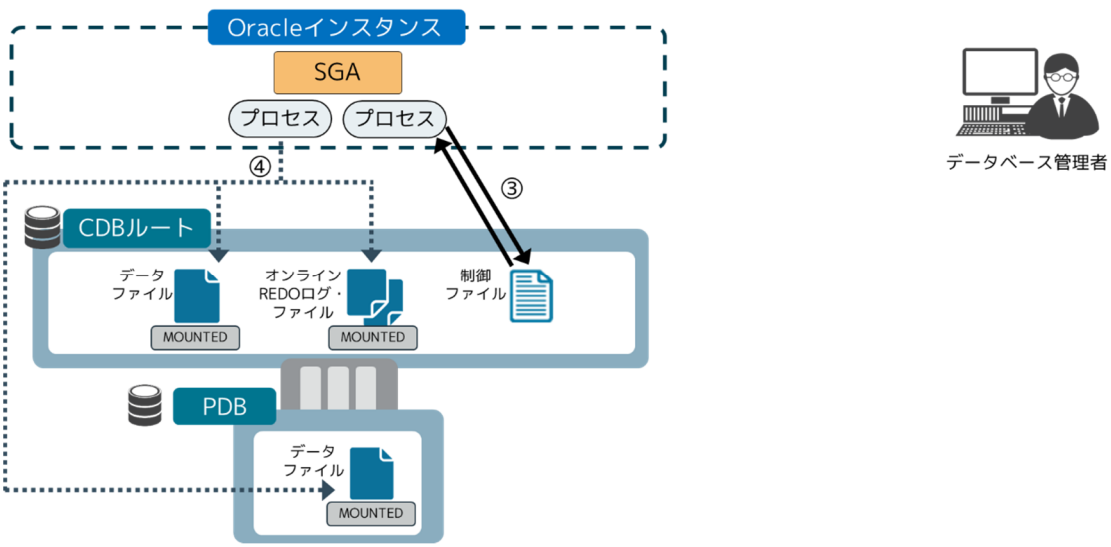
1.Oracle インスタンスの起動



①パラメータ設定ファイル（SPFILE）を読み込む。

②Oracle インスタンスを起動する。

2.データベースのマウント



③制御ファイルが読み込まれる。

④データファイルとオンライン REDO ログ・ファイルのパスが検出される

（Oracle インスタンスとデータファイル、オンライン REDO ログ・ファイルの対応づけ）。

株式会社 アシスト 無断転載・複製禁止 Copyright (C) 2020 K.K.Ashisuto All Rights Reserved.

3）データベースのオープン（CDB ルートのオープン）

CDB ルートを構成するデータファイルとオンライン REDO ログ・ファイルをオープンし、ユーザーがアクセス可能な状態にします。

■データベースのオープンに失敗した場合に考えられる原因
・（CDB ルートの）データファイルやオンライン REDO ログ・ファイルが破損している
・（CDB ルートの）データファイルやオンライン REDO ログ・ファイルが存在しない

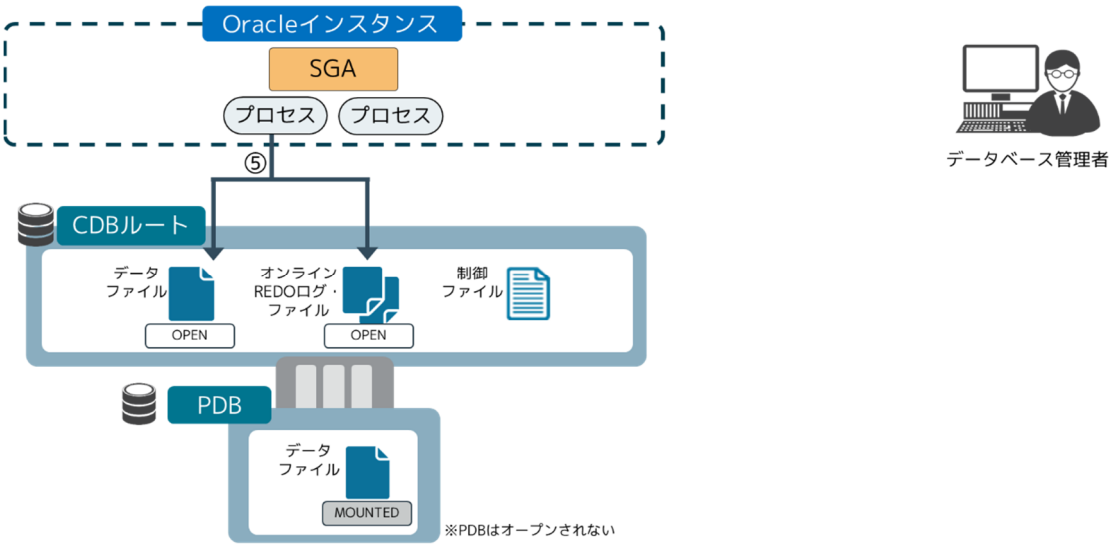
4）PDB のオープン

PDB を構成するデータファイルをオープンし、ユーザーがアクセス可能な状態にします。
※マルチテナント構成でのみ実施するステップです。

■PDB のオープンに失敗した場合に考えられる原因
・PDB のデータファイルが破損している
・PDB のデータファイルが存在しない

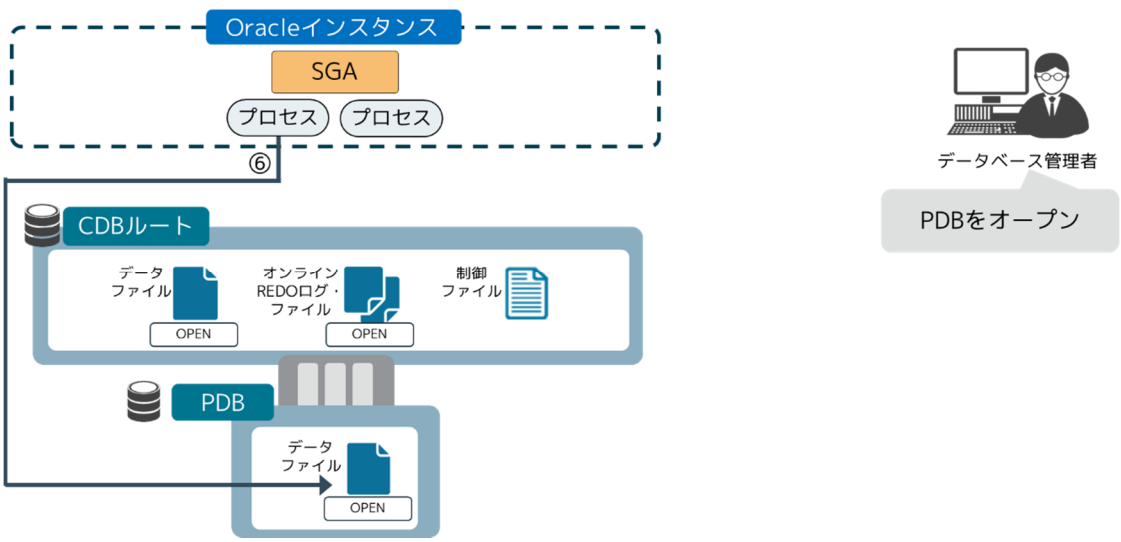
Copyright (C) 2020 K.K.Ashisuto All Rights Reserved.　株式会社 アシスト 無断転載・複製禁止

3.データベースのオープン（CDB ルートのオープン）



⑤CDB ルートのデータファイルとオンライン REDO ログ・ファイルがオープンされ、CDB ルートにアクセス可能になる（PDB はオープンされず、アクセスできない）。

4.PDB のオープン



⑥PDB がオープンされ、PDB にアクセス可能になる。

Copyright (C) 2020 K.K.Ashisuto All Rights Reserved.

（2）STARTUP コマンド

SQL*Plus の STARTUP コマンドで、Oracle インスタンスの起動、データベースのマウント、データベースのオープンを実行できます。ステップごとに起動することも、データベースのオープンまで自動的に起動することも可能です。
※データベースの起動・停止は特別な管理権限でログインしたユーザーのみ実行できます。

「特別な管理権限」（1-18）

構文

```
STARTUP { NOMOUNT | MOUNT | OPEN }
[PFILE = ファイル名 ]
```

NOMOUNT	Oracle インスタンスを起動します。
MOUNT	Oracle インスタンスを起動し、データベースをマウントします。
OPEN	Oracle インスタンスを起動し、データベースをマウント、オープンします。 ※PDB はオープンされません。
PFILE	指定した初期化パラメータ・ファイル（init<SID 名>.ora）を読み込んで起動します。 「初期化パラメータ・ファイル」（3-2）

STARTUP コマンドでは、CDB ルートのオープンまでしか制御できません。そのため、PDB は別途オープンする必要があります。

Copyright (C) 2020 K.K.Ashisuto All Rights Reserved.　株式会社 アシスト 無断転載・複製禁止

例）データベースを起動する。

```
/* STARTUP コマンドを実行 */
SQL> STARTUP
ORACLE インスタンスが起動しました。

Total System Global Area 1610610688 bytes
Fixed Size                  9267200 bytes
Variable Size             402653184 bytes
Database Buffers         1191182336 bytes
Redo Buffers                7507968 bytes
データベースがマウントされました。
データベースがオープンされました。

/* Oracle インスタンスの状態を確認 */
SQL> SELECT instance_name,status,TO_CHAR(startup_time,'YYYY-MM-DD HH24:MI:SS') AS startup_time
  2  FROM   v$instance;

INSTANCE_NAME    STATUS       STARTUP_TIME
---------------- ------------ -------------------
manage19         OPEN         2020-04-01 14:36:03

/* PDB の状態を確認 */
SQL> SHOW PDBS

    CON_ID CON_NAME                        OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
         2 PDB$SEED                       READ ONLY  NO
         3 PDB1                           MOUNTED
```

※SQL*Plus の SHOW PDBS コマンドで、PDB の起動状態を確認できます。

V$INSTANCE ビュー

現行 Oracle インスタンスの状態を示します。

■主な列

・INSTANCE_NAME　Oracle インスタンス名。

・STATUS　Oracle インスタンスの状態。

- STARTED　：Oracle インスタンス起動状態
- MOUNTED　：マウント状態
- OPEN　：オープン状態

・STARTUP_TIME　Oracle インスタンスの起動時刻。

Copyright (C) 2020 K.K.Ashisuto All Rights Reserved.

（3）データベースの状態変更コマンド

Oracle インスタンスの起動後、データベースをマウント、オープンする場合は、以下の ALTER DATABASE 文を実行し、1 ステップずつ状態を変更します。

構文

```
ALTER DATABASE { MOUNT | OPEN }
```

MOUNT	Oracle インスタンスの起動状態からマウントします。
OPEN	マウント状態からオープンします。

例）データベースの状態変更を行う。

```
SQL> STARTUP NOMOUNT
ORACLE インスタンスが起動しました。
…（省略）…

SQL> ALTER DATABASE MOUNT;

データベースが変更されました。

SQL> ALTER DATABASE OPEN;

データベースが変更されました。

/* Oracle インスタンスの状態を確認 */
SQL> SELECT instance_name,status,TO_CHAR(startup_time,'YYYY-MM-DD HH24:MI:SS') AS startup_time
  2  FROM   v$instance;

INSTANCE_NAME    STATUS       STARTUP_TIME
---------------- ------------ -------------------
manage19         OPEN         2020-03-17 14:27:01

/* PDB の状態を確認 */
SQL> SHOW PDBS

    CON_ID CON_NAME                       OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
         2 PDB$SEED                       READ ONLY  NO
         3 PDB1                           MOUNTED
```

Copyright (C) 2020 K.K.Ashisuto All Rights Reserved.
株式会社 アシスト 無断転載・複製禁止

※PDB をオープンする場合は、ALTER SESSION SET CONTAINER 文で PDB に接続し、ALTER DATABASE OPEN 文を実行します。

構文

```
ALTER SESSION SET CONTAINER = PDB 名
```

PDB 名	接続する PDB 名を指定します。

例）PDB に接続してオープンする。

```
/* PDB の状態を確認 */
SQL> SHOW PDBS

    CON_ID CON_NAME                        OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
         2 PDB$SEED                       READ ONLY  NO
         3 PDB1                           MOUNTED

SQL> ALTER SESSION SET CONTAINER = pdb1;

セッションが変更されました。

SQL> ALTER DATABASE OPEN;

データベースが変更されました。

SQL> SELECT name,open_mode,TO_CHAR(open_time,'YYYY-MM-DD HH24:MI:SS') AS open_time
  2  FROM   v$pdbs;

NAME                 OPEN_MODE  OPEN_TIME
-------------------- ---------- ----------------------------------------
PDB1                 READ WRITE 2020-03-17 16:04:01
```

V$PDBS ビュー

現行インスタンスに関連する PDB の情報を示します。

■主な列

・NAME　　PDB の名前。

・OPEN_MODE　　PDB の状態。

　MOUNTED　　：マウント状態

　READ WRITE　　：オープン状態（読み書き可能）または

　READ ONLY　　：オープン状態（読み取り専用）

・OPEN_TIME　　PDB のオープン日時。

Copyright (C) 2020 K.K.Ashisuto All Rights Reserved.

■補足：ALTER PLUGGABLE DATABASE 文

ALTER PLUGGABLE DATABASE 文を使用すると、CDB ルートに接続した状態で 1 つ以上の PDB をまとめてオープンできます。また、CDB ルートのオープンに合わせて、PDB が自動的にオープンするように設定することもできます。

構文

```
ALTER PLUGGABLE DATABASE
{ PDB 名 [ , PDB 名 … ] | ALL [ EXCEPT PDB 名 [ , PDB 名 … ] ] }
{ OPEN | { SAVE | DISCARD } STATE }
```

PDB 名	操作対象となる PDB 名を指定します。
ALL	すべての PDB を操作対象とします。
EXCEPT PDB 名	ALL 句を使用するときに、操作対象外とする PDB を指定します。
OPEN	PDB をオープンします。
SAVE STATE	指定した PDB の現在の状態を保存します。
DISCARD STATE	指定した PDB がデータベース起動時にマウント状態になるようにします。

例）すべての PDB をまとめてオープンする。

```
SQL> SHOW PDBS

    CON_ID CON_NAME                        OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
         2 PDB$SEED                       READ ONLY  NO
         3 PDB1                           MOUNTED
         4 PDB2                           MOUNTED

/* すべての PDB をまとめてオープン */
SQL> ALTER PLUGGABLE DATABASE ALL OPEN;

プラガブル・データベースが変更されました。

/* PDB の状態を確認 */
SQL> SELECT name,open_mode,TO_CHAR(open_time,'YYYY-MM-DD HH24:MI:SS') AS open_time
  2  FROM   v$pdbs;

NAME                 OPEN_MODE  OPEN_TIME
-------------------- ---------- -------------------
PDB$SEED             READ ONLY  2020-04-01 16:12:06
PDB1                 READ WRITE 2020-04-01 16:12:36
PDB2                 READ WRITE 2020-04-01 16:12:36
```

Copyright (C) 2020 K.K.Ashisuto All Rights Reserved.　株式会社 アシスト 無断転載・複製禁止

例）CDB ルートのオープンに合わせて、すべての PDB が自動的にオープンするように設定する。

```
SQL> SELECT name,open_mode,TO_CHAR(open_time,'YYYY-MM-DD HH24:MI:SS') AS open_time
  2  FROM   v$pdbs;

NAME                 OPEN_MODE  OPEN_TIME
-------------------- ---------- -------------------
PDB$SEED             READ ONLY  2020-04-01 16:12:06
PDB1                 READ WRITE 2020-04-01 16:12:36
PDB2                 READ WRITE 2020-04-01 16:12:36

/* PDB の現在の状態（READ WRITE でのオープン）を保存 */
SQL> ALTER PLUGGABLE DATABASE ALL SAVE STATE;

プラガブル・データベースが変更されました。

/* データベースを再起動 */
SQL> SHUTDOWN IMMEDIATE
データベースがクローズされました。
データベースがディスマウントされました。
ORACLE インスタンスがシャットダウンされました。

SQL> STARTUP
ORACLE インスタンスが起動しました。
…（省略）…
データベースがマウントされました。
データベースがオープンされました。

/* PDB が自動的にオープンされることを確認 */
SQL> SELECT name,open_mode,TO_CHAR(open_time,'YYYY-MM-DD HH24:MI:SS') AS open_time
  2  FROM   v$pdbs;

NAME                 OPEN_MODE  OPEN_TIME
-------------------- ---------- -------------------
PDB$SEED             READ ONLY  2020-04-01 16:15:53
PDB1                 READ WRITE 2020-04-01 16:15:54
PDB2                 READ WRITE 2020-04-01 16:15:54
```

※SAVE STATE 句は指定した PDB の現在の起動状態を保存するコマンドです。そのため、自動オープンの設定を行うためには、コマンドを実行する前に対象の PDB をオープン状態にしておく必要があります。

株式会社 アシスト 無断転載・複製禁止 Copyright (C) 2020 K.K.Ashisuto All Rights Reserved.