

Oracle Databaseの特徴

- マルチプラットフォーム
- ロック機能
 - SQL Serverなどでは、余分な箇所にロックを発生させる**ロックエスカレーション**がある
- シェアード・エブリシング型クラスタシステム
 - 複数のコンピュータがディスクを共有するため、耐障害性が強い
- オールJava化
- 管理の自動化

Oracleのインストール

- 1台のマシンにバージョンの異なる複数のOracle Databaseをインストールして管理する場合
 - バージョンごとにORACLE_HOME(Oracleソフトウェアをインストールするディレクトリ)の値を設定する
 - 12c用のORACLE_HOMEを「\$ORACLE_BASE/product/12.1.0/dbhome_1」、
 - 19c用のORACLE_HOMEを「\$ORACLE_BASE/product/19.3.0/dbhome_1」とする
- Oracle Universal Installer(OUI)は、OracleソフトウェアをインストールするためのGUIツールです。
 - [非対話型インストール]
 - 非対話形式のインストールでは、Oracle Databaseのインストールに必要な情報をレスポンス・ファイルに記述し、OUI起動時にレスポンス・ファイルを指定することで、インストールのすべての手順、または一部を自動化できます。

```
$ ./runInstaller -responseFile レスポンス・ファイル名
```

- Oracleソフトウェア所有者の環境変数に、Oracleインスタンスの識別子やインストール先ディレクトリなどのOracleの実行環境に関する情報を設定

環境変数	説明
ORACLE_BASE	Oracle関連ファイルの基準となるディレクトリのパスを設定する。 データベースファイルやログファイルなどの、 Oracleに関連するほとんどのファイルがORACLE_BASE以下に配置される
ORACLE_HOME	Oracleのソフトウェアを配置するディレクトリのパスを設定する。 ORACLE_HOMEは、ORACLE_BASEのサブディレクトリに指定する
ORACLE_SID	インスタンスの識別子であるSID(システム識別子)を設定する。 この環境変数により、接続先インスタンスが識別される。 原則的に「 インスタンスのSID=データベース名=インスタンス名 」の関係が 成り立つ
LD_LIBRARY_PATH	Oracleが使用する共有ライブラリの場所を指定する

- インベントリ・ディレクトリ

- インベントリ・ディレクトリは、1台のマシンにインストールされたOracleソフトウェアやパッチ(ソフトウェアの不具合を修正するプログラム)のメタデータ(インストール・ログ・ファイルなどの情報)の格納場所です。
 - インベントリ・ディレクトリはOracleソフトウェアを初めてインストールする時に作成され、同じマシンにインストールされるすべてのOracleソフトウェアで共有されます。
 - インベントリ・ディレクトリは、Oracleソフトウェアやパッチ(ソフトウェアの不具合を修正するプログラム)のメタデータ(インストール・ログ・ファイルなどの情報)の格納場所です。
- UNIX、Linux環境ではインベントリ・ディレクトリとして、通常「\$ORACLE_BASE/./oralInventory」が作成されます。

例) /u01/app/oralInventory

- Oracleソフトウェアのインストールの前に、OSユーザ(**Oracleソフトウェア所有者**)を作成

OSユーザ	説明
Oracleソフトウェア所有者	Oracleをインストールおよび管理するOSユーザ。後述するOracleインベントリグループ、OSDBAグループに所属する。通常、ユーザ名は" oracle "とする
OSグループ	説明
Oracleインベントリグループ	Oracleに関連するファイルを所有するOSグループ。通常、グループ名は" oinstall "とする
OSDBAグループ	Oracleの管理権限が付与されるOSグループ。通常、グループ名は" dba "とする
(OSOPERグループ)	SYSOPER管理権限(oper)、データベース管理権限の一部を持つ
(OSBACKUPDBAグループ)	バックアップおよびリカバリ(backupdba)
(OSDGDBAグループ)	Data Guard(dgdab)
(OSKMDBAグループ)	暗号化鍵関連/SYSKM管理(kmdba)
(OSRACDBAグループ)	RAC関連(racdba)

Oracleの歴史

- Oracle Database 10g
 - Oracle Enterprise Manager (OEM)
 - ブラウザベースの運用管理ツール
 - Automatic Storage Management (ASM)
 - ストレージ管理ツール
- Oracle Database 11g
 - Oracle Clusterware
 - ストレージの高可用性
- Oracle Database 12c

- マルチテナント
 - 1つのインスタンスで複数のデータベースを扱えるようにした
- インメモリ
 - バッファキャッシュとは異なる形式でデータをメモリ上保持する機能
 - レコード単位ではなく、列単位
 - バッファキャッシュとインメモリの使い分けは自動
- シャーディング
 - シェアード・ナッシング型のクラスタリングを提供
- Oracle Database 18c
 - バッチ提供体系の変更
- Oracle Database 19c
 - Automatic Indexing
 - エディション
 - Enterprise Edition : 高機能
 - Standard Edition Two (SE2)
 - Personal Edition

旧バージョンからのアップグレード

バージョンのアップグレード

- Database Upgrade Assistant (DBUA)
 - GUIベースのアップグレード

データ移行

- SQL*Loader
- データベースリンク
 - CREATE TABLE AS SELECT 文
 - INSERT SELECT 文
- Export/Import
 - Datapump
- トランスポータブル表領域 (TTS)
- レプリケーションツール
 - Oracle GoldenGate

データベースの種類

- SAMファイル
 - シーケンシャルアクセス（先頭から順に読み込んでいく）
- ISAMファイル
 - 索引キーを用いる
- 階層型データベース
 - 親子、ツリー構造
- ネットワーク型データベース

- 親子関係だけでなく、横方向も
- **リレーショナルデータベース**
 - 2次元の表
 - 構造 - オブジェクトとリレーション
リレーショナル・データベースは、データをリレーション(表)に格納するデータベースです。「表」は、「行」と「列」から構成される2次元の構造です。表はオブジェクトとして定義されます。
 - 操作 - 一連の演算子
リレーショナル・データベースのデータを操作するには、SQL(Structured Query Language)を使用して「どの表のどの列の値を取り出す」というように表の名前と条件を指定する(論理演算)だけで、RDBMS(リレーショナル・データベース管理システム)がどのようにその操作を実行すべきかを判断して演算を実行し(物理演算)、目的のデータにアクセスできます。
 - 整合性規則
リレーショナル・データベースには、データや構造への操作を制御するためのルールである「整合性規則」を設定できます。
- オブジェクト指向データベース
 - データとメソッドをセットにしたオブジェクトで管理
- オブジェクト・リレーショナルデータベース
- 多次元データベース
 - データウェアハウス系
- XMLデータベース
 - Oracle Databaseでも、XML専門のデータ型や表を持つことができる
- NoSQLデータベース
 - Not only
 - キーバリューストア型データベース (KVS)
 - ドキュメント志向データベース

メモリ構造

- Oracle Databaseを構成する二大要素は**インスタンス**と**データベース**

インスタンスの概要

- インスタンス: データベースとユーザーの中間に位置する
 - メモリ構造 : システムグローバル領域 (SGA)
 - プロセス群 : バックグラウンドプロセス
- Oracle Databaseは、データをダイレクトに読み書きするのではなく、**システムグローバル領域 (SGA)** の共有プールなどを利用して、ワnkクッションおいてから読み書きする

複数のインスタンスの起動

- インスタンスは、それぞれ別個のメモリ領域で動作しているため、サーバー上で複数起動が可能
- インスタンスの名前
 - インスタンスを一意に識別するため、**SID (Oracle8iまで) /データベース名**が使用される

リスナープロセス

- **リスナープロセス**によってインスタンスへの通信経路が確立される
- ユーザープロセスからの接続要求があると、リスナープロセスとインスタンスの接続を行う
- 異なるデータベース間の通信経路も確立
- **サービス名（グローバルデータベース名）**により、ほかのインスタンスと差別化
- **リスナー制御ユーティリティ (lsnrctl)**
 - リスナー名を省略するとデフォルトリスナーと解釈される

コマンド指定	処理内容
lsnrctl start [リスナー名]	リスナーを起動する
lsnrctl stop [リスナー名]	リスナーを停止する
lsnrctl status [リスナー名]	リスナーの稼働状態と、リスナーが認識しているデータベースサービスを確認する
lsnrctl services [リスナー名]	リスナーの稼働状態と、リスナーが認識しているデータベースサービスの詳細情報を確認する
<ul style="list-style-type: none"> ◦ status <ul style="list-style-type: none"> ■ リスナーのプロトコル、ホスト名、ポート番号 ■ ログファイルの出力先 ■ リスナーに登録されているデータベースサービスのサマリー ■ 「lsnrctl status」コマンドは、指定したリスナーのステータス(名前や稼働時間)、リスナーの構成ファイル「listener.ora」の場所、ログ・ファイルの場所、リスニングするプロトコルやポート番号、サポートしているサービス(データベース)の概要などを表示します。 ◦ services <ul style="list-style-type: none"> ■ LISTENERという名前のリスナーがサポートしているサービス ■ サービスに割り当てられたサービス・ハンドラの詳細 ◦ 	

システムグローバル領域（SGA）

- Oracleサーバーが稼働するコンピュータのメインメモリ内に生成される**共有メモリ領域**
 - 表データの一時的な格納、SQL文の再利用、メモリ上のバッファ内のデータに加えたREDOログ情報など
 - Oracle Databaseでは、データを更新する命令が出されても、すぐに表に格納されたデータを書き換えることはせず、メモリ上で更新を行ってから、ある一定間隔で実際の物理ファイルに書き換えに行く
- SGA内のコンテンツ
 - データベースバッファキャッシュ
 - REDOログファイル
 - 共有プール

- Javaプール
- ラージプール
- Streamプール など

データベースバッファキャッシュ

- データファイルから読み込まれた**データブロック**の複製を置いておく場所
- データベースの検索や更新
 - ユーザープロセスからのSQL文により、あるデータのセットの要求がされる
 - まず、データベースバッファキャッシュの中を見る
 - なければ、その時初めてデータが格納されているディスクから必要なデータブロックをメモリに読み込む
 - メモリ内の複製データに対して、要求された処理を行う
- データベースバッファキャッシュは、**書き込みリスト**と**LRUリスト**で管理される
 - 書き込みリスト : 使用済みバッファの管理
 - LRUリスト : 使用可能バッファ/使用中バッファ/書き込みリストに未移動のバッファの管理
 - ユーザプロセスの利用頻度 (LRU/MRU) によって順番が調整される
- 初期化パラメータの「**db_cache_size**」の値によりサイズをチューニング

REDOログバッファ

- **REDOログファイル**は、データベースに対するREDOログ情報をすべて記録する物理ファイル
 - 変更履歴の記録ファイル（これを書いた後に処理を実行する）
- **REDOログバッファ**は、このREDOログファイルに書き出される前のREDOログ情報が保存されたバッファ
- トランザクションの大きい処理を行う場合には、大きくしたほうが良い
 - 初期化パラメータの「**log_buffer**」でチューニング
-

共有プール

- 一度実行されたSQL文を再利用できるように、コンパイル結果を保存する
- 頻繁にアクセスする**データディクショナリ**をキャッシュしておく
- 構成
 - ライブラリキャッシュ
 - 共有SQL領域やPL/SQL領域を含むメモリ構造
 - まったく同一のSQL文の実行結果を保管
 - ディクショナリキャッシュ
 - データをバッファの代わりに「行」として保持する

その他の構成

- ラージプール
 - バックアップやリストア処理などのための、大量メモリ割り当てを提供するオプションのメモリ構造
- Javaプール
 - Oracle Databaseに内蔵されたJVMがjavaコードとデータを使用する際に必要なオプションのメモリ構造

- Streamプール
 - Streamメモリを割り当てるためもオプションのメモリ構造
 - 時間的に変化する大量のデータストリーム（メッセージキューイングやデータレプリケーションを提供する機能）をキャッシュ

SGAの自動管理（19c～）

- 動的にチューニングする**自動共有メモリ・チューニング機能**
- 最大メモリサイズを設定しておくだけで最適化

初期化パラメータ	説明
sga_target	SGA全体の合計サイズ指定
sga_max_size	SGAの最大サイズを指定
db_cache_size	データベースバッファキャッシュのサイズを指定
shared_pool_size	共有プールのサイズ
large_pool_size	ラージプールサイズ
java_pool_size	Javaプールサイズ
stream_pool_size	ストリームプールのサイズ
result_cache_max_size	結果キャッシュのサイズ

```
ALTER system SET sga_target= 600M;
```

プログラムグローバル領域（PGA）

- サーバプロセス起動時にOracleサーバーのメモリ内に自動的に作成されるメモリ領域
- 1つのサーバプロセスごとに割り当てられる
- **非共有メモリ領域**が使われ、変数や配列、セッション情報、ソート領域などに利用
- 構成
 - プライベートSQL領域
 - セッションメモリ

プライベートSQL領域

- バインド情報などのデータを格納する領域
 - バインド変数：SQL文に埋め込む変数で、実際の値を後から設定できるようにSQL文の一部を変数にしたもの
- ランタイム領域
 - プログラムを動かす

セッションメモリ

- ログイン情報に代表されるセッションの変数や、セッションに関係するほかの情報を保持

PGAの自動管理

- 初期化パラメータ「**workarea_size_policy**」を「AUTO」に設定で自動管理される
- 初期化パラメータ「**pga_aggregate_target**」で、インスタンス専用のPGAメモリの合計サイズを設定

自動メモリー管理

メモリー管理方法について

メモリー管理方法	説明	管理者が指定する初期化パラメータ 太=動的
自動メモリー管理 (AMM) ※SGA/PGAの合計 の管理	SGAとインスタンスPGAの合計サイズを指定する 指定したサイズ内でSGAとインスタンスPGA間のメモリーサイズが自動調節される	MEMORY_TARGET MEMORY_MAX_TARGET
自動共有メモリー 管理(ASMM) ※SGAの管理	SGAの合計サイズを指定する 指定したサイズ内でSGAの各コンポーネントのメモリーサイズが自動調整される ※PGAは自動PGAメモリー管理となる	SGA_TARGET SGA_MAX_SIZE
手動共有メモリー 管理 ※SGAの内部の管理	SGAの各コンポーネントのサイズを個別に指定する ※PGAは自動PGAメモリー管理となる	SHARED_POOL_SIZE DB_CACHE_SIZE LARGE_POOL_SIZE JAVA_POOL_SIZE STREAMS_POOL_SIZE LOG_BUFFER
自動PGAメモリー 管理	インスタンスPGAの合計サイズを指定する 指定したサイズ内で各PGAのメモリーサイズが自動調整される	PGA_AGGREGATE_TARGET

- 自動メモリー管理
 - MEMORY_TARGET(動的)：自動メモリー管理で使用する、インスタンスの合計メモリーサイズ。
MEMORY_MAX_TARGETに設定した値以下にする必要がある
 - MEMORY_MAX_TARGET(静的)：自動メモリー管理で利用できる、インスタンスの最大メモリーサイズ。
MEMORY_TARGETに設定できる最大値を表す

データベース

- 構成要素
 - REDOログ・ファイル
 - データの変更履歴が格納される
 - データファイル
 - 表や索引などのデータが格納される

- 制御ファイル
 - データベースの物理構造に関するデータが格納される

プロセス

- ユーザプロセス
 - Oracle Databaseに対して、SQLを発行するプロセス
- サーバプロセス
 - Oracleサーバが稼働するコンピュータの中に生成されるプロセス
 - サーバプロセスの開始時にPGAが作成される
 - 専用サーバ構成と共有サーバ構成
 - 専用サーバ構成 : サーバプロセスとユーザプロセスが 1 対 1
 - 共有サーバ構成 : 1つのサーバプロセスを複数のユーザプロセスで共有する
 - この構成では、ユーザプロセスとサーバプロセスの間に**ディスパッチャプロセス (Dnnn) **が介在する

バックグラウンドプロセス

- SGAとともにインスタンスを構成する要素
- メモリ領域に一時的に保存されているデータと物理ファイルの架け橋になる

プロセス名	説明
データベースライター (DBWn)	データベースバッファキャッシュ内の更新済みブロックをデータファイルに書き込む
ログライター (LGWR)	REDOログファイルのREDOデータ(更新履歴情報)をREOログファイルに書き出す DBWnが書き出しをする前にLGWRが完了する必要がある、DBWnによる書き出し指示がある
チェックポイント (CKPT)	DBWnに対して、データベースバッファキャッシュ内の更新済みブロックをデータファイルに書き込む指示をだす。 制御ファイルにチェックポイント情報を書き込む。 制御ファイルとデータファイルを同期。
システムモニター (SMON)	インスタンスが異常終了した場合、次回インスタンス起動時にデータベースファイルの整合性を復旧する処理を実行する
プロセスモニター (PMON)	プロセスが異常終了した場合、そのプロセスが使用していた様々なデータやリソースの後処理 (クリーンアップ) を実行する
管理モニター (MMON)	性能分析などで使用される様々な統計情報を定期収集する
アーカイバ (ARCN)	ログスイッチ(REDOログファイルが満杯になる)の発生後、REDOログファイルのREDOデータをアーカイブログファイルとしてコピーする

ネットワークの仕組み

- OracleサーバーとOracleクライアントの2つから成り立っている
 - **クライアント/サーバーアーキテクチャ = 2層接続**
- n層アーキテクチャ
 - データベースサーバーとクライアントアプリケーションの間に、橋渡しとなるアプリケーションサーバーやWebサーバーなどが介在する構成

Oracle Net Services

- 構成要素
 - Oracle Net
 - Oracle Net Listener
 - Oracle Connection Manager
 - ユーティリティ（Oracle Net Configuration Assistant、Oracle Net Manager）

Oracle Net

- Oracle Net Foundation
 - ログイン要求で渡された目的の**ネットサービス名**を**ホスト名+サービス名**に還元
 - 名前解決
 - tnsnames.ora
 - ネットサービス名のIPアドレス/ポート番号変換
 - listner.ora
- Oracle Protocol Support
 - ホスト名をIPアドレスに変換

Oracle Net Listner

- リスナー
- ネットサービス名（= 接続識別子 = 接続記述子を一つにまとめた別名）
 - プロトコル
 - ホスト名
 - ポート番号
 - グローバルデータベース名
- グローバルデータベース名とSID
 - グローバルデータベース名（= データベース名 + ドメイン名）
 - ネットワーク上でOracle Databaseを一意に識別するための名前
 - SID
 - 同一ノード上でのほかのインスタンスと区別するための一意の識別子
- ユーティリティ
 - Oracle Net Configuration Assistant (NetCA)
 - ネットサービス名を新規に作成/変更/削除
 - Oracle Net Manager
 - Oracle Net Serviceの構成状況を管理
 - tnsnames.ora
 - listner.ora
 - sqlnet.ora
 - 接続識別子(ネット・サービス名、またはネットワーク・サービス名)から接続記述子への名前解決に使用するネーミング・メソッドとその優先順

ツールズ

- データベース管理ツール

ツール名	機能	ツール起動 コマンド
Oracle Database Configuration Assistant (DBCA)	<p>データベースの作成、削除およびテンプレートの管理を行う。 データベースを削除すると、データベースのインスタンスが停止し、論理構成とすべての物理ファイルが削除され、取り消しは不可。</p> <p>なお、DBCAで削除できるデータベースはDBCAで作成したデータベースに限られる。</p> <p>CREATE DATABASE文で作成したデータベースはDBCAで削除できない。</p> <ul style="list-style-type: none"> ・データベースの作成 ・既存データベースの構成 ・データベースの削除 ・テンプレートの管理 ・プラガブル・データベース（「マルチテナント・アーキテクチャ」機能の仮想DB）管理 ・Oracle RACデータベース ・インスタンス管理 	dbca
Oracle Universal Installer (OUI)	Oracle ソフトウェアのインストールやアンインストールを行う	runInstaller
Database Upgrade Assistant (DBUA)	古いリリースのデータベースを新しいリリースにアップグレードする	dbua
Oracle Enterprise Manager Database Express (EM Express)	<p>Webブラウザの画面からGUI操作でOracleデータベースを管理可能。パフォーマンスの監視、構成管理、診断とチューニングなどを実行できる</p> <ul style="list-style-type: none"> ・メモリー・アドバイザー ・UNDOアドバイザー ・ADDM ・SQLチューニング・アドバイザー <p>※OUIでOracleソフトウェアをインストールすると自動的にインストールされる</p> <p>※DBCAでデータベースを作成する時に構成できる</p> <p>※リスナーがOracle Databaseへの接続要求を送信し、ディスパッチャというプロセスが共有サーバー・プロセス(複数のセッションの処理を行うサーバー・プロセス)に処理を割り振ります。</p> <p>★Controlで実行可能な「データベースの起動や停止」「バックアップやリカバリ」「セグメント・アドバイザー(領域の断片化を分析するアドバイザー)」は不可</p>	-

ツール名	機能	ツール起動 コマンド
Oracle Enterprise Manager Cloud Control	Webブラウザの画面からGUI操作で複数のOracleデータベースを一元的に管理可能。非常に多くの管理作業を実行できる	-
SQL*Plus	コマンドラインインターフェイスでSQLを実行してOracleデータベースを管理可能 SQL*Plusでは、SQL文、PL/SQL、SQL*Plusコマンドを実行できる。SQL文ではデータの検索や追加などのデータ操作のほか、インスタンスの起動・停止、バックアップやリカバリなどの管理作業も行える	sqlplus
Oracle Net Manager	Oracleをネットワーク環境で使用するための設定を行う。Oracle Net Configuration Assistant よりも広範な設定をサポートする	netmgr
Oracle Net Configuration Assistant (NetCA)	Oracleをネットワーク環境で使用するための主要な設定(リスナーやリモート接続の構成など)を行う	
リスナー制御ユーティリティ (LSNRCTL)	リスナーの起動/停止を行う	lsnrctl
SQL*Loader	CSVファイルなどのデータベース外部にあるファイルのデータを、データベースの表にロードする	sqlldr
Oracle Data Pump	データベース内のデータを別のデータベースに転送する(スキーマ単位でデータをエクスポート/インポート)	8
Recovery Manager (RMAN)	データベースのバックアップおよび障害発生時の復旧作業(メディアリカバリ)を行う	10
Oracle Secure Backup	テープバックアップ管理を提供するツール	-

- Oracle Database Configuration Assistant (ODCA) について
 - ・[メモリー]: メモリーに関する初期化パラメータの値を設定
 - ・[サイズ設定]: ブロック・サイズと、同時起動プロセス数の指定。
 - ・[キャラクタ・セット]: キャラクタ・セット(画面文字表示コード体系)を指定する。
 - ・[接続モード]: 専用サーバー・モードまたは共有サーバー・モードのどちらか
 - ・[サンプル・スキーマ]: サンプルのHuman Resourcesスキーマをインストールするか
 - 選択可能な記憶域
 - ファイルシステム
 - OSのファイルシステム上にデータベースを構築する
 - 自動ストレージ管理(ASM)
 - Oracle Automatic Storage Management
データベースファイルのボリューム・マネージャおよびファイルシステム。ASMにより自動管理されるディスク・グループ上にデータベースを構築するストレージ管理ソリューション。
 - Oracle Managed Filesの使用

- このオプションにチェックを付けると、データベースを構成するOSファイルをOracle Databaseが管理する。データベース管理者はファイル名やサイズを指定する必要はない
- データベース・テンプレート
 - テンプレートはデータベースの作成に必要な情報が含まれたデータベースのひな型で、XML形式のファイルです。
Oracle社が提供する事前定義済テンプレートの他、既存のテンプレートや既存のデータベースを基にして、新しいテンプレートを作成して利用することもできます。
 - あらかじめ用意されているテンプレート(事前定義済みテンプレート)
 - データ・ウェアハウス（シード）
 - 汎用またはトランザクション処理（シード）
 - カスタム・データベース（非シード）
 - マクロな視点でのテンプレート種類
 - シード
 - テンプレートにデータファイルが含まれる。
シードデータベースの複製として、データベースにを作成する。
作成するデータベースの構成はシードデータベースに準ずるものになる。
 - 非シード
 - テンプレートにデータファイルが含まれない。作成するデータベースは柔軟に構成できる

テンプレートに含まれる主な情報は以下のとおりです。

- ・データベース・オプション
- ・初期化パラメータ
- ・記憶域属性(データファイル、表領域、制御ファイルおよびオンラインREDOログ・ファイルの属性)

※シードテンプレートをもとにしたデータベースの作成で変更できるもの

- データベースの名前
- データファイルの格納先
- 制御ファイルの数
- REDOロググループの数
- 初期化パラメータ

- 作成するときの補足
 - 専用サーバー・モードまたは共有サーバー・モードに構成できる
 - 管理者ユーザーのパスワードを設定できる
 - Enterprise Manager Cloud Controlで管理できるように構成できる
- Oracle Universal Installer (OUI)
 - Oracle Universal Installer(OUI)での前提条件チェックでインストール要件を満たしていない場合、[修正および再チェック]ボタンをクリックすると、システム構成を修正する「修正スクリプト」と呼ばれるシェル・スクリプトが生成されます。
 - 生成された修正スクリプト「runfixup.sh」をrootユーザーでコマンドラインから実行すると、OSのカーネル・パラメータやディレクトリの権限などがインストール要件を満たすように修正

されます。

- EM Express
 - [構成]
 - メモリー : メモリー割り当ての参照、メモリー・アドバイザ(メモリーのサイズを提案するアドバイザ)の使用
 - [記憶域]
 - UNDO管理 : UNDO(実行した操作を取り消す機能)の管理、UNDOアドバイザ(UNDO表領域のサイズを提案するアドバイザ)の使用
 - [パフォーマンス]
 - パフォーマンス・ハブ : リアルタイムでのパフォーマンス監視、ADDM(データベース全体を監視、診断するアドバイザ)の使用
 - SQLチューニング・アドバイザ: SQLチューニング・アドバイザ(パフォーマンス向上のためにSQL文をチューニングするアドバイザ)の使用
- ソフトウェア
 - Oracle Automatic Storage Management
 - データベース・ファイルのボリューム・マネージャ及びファイルシステム。**RAC構成において、共有ストレージ用のクラスタファイルシステムとして機能します。**
 - Oracle Data Guard
 - 障害発生時のフェイルオーバー、データ保護を実現する機能。
 - Oracle Data Guardは障害発生時のフェイルオーバー、データ保護を実現する機能です。
 - 本番用のプライマリ・データベースとは別に、遠隔地に予備用のスタンバイ・データベースを用意しておき、災害時などプライマリ・データベースが使用できなくなった場合にフェイルオーバーでスタンバイ・データベースに切り替えます。
 - Oracle Data Guardは、プライマリ・データベースのREDOデータ(データの変更履歴)をスタンバイ・データベースに転送して適用することによってデータをコピーします。
 - また、通常はスタンバイ・データベースを読み取り専用で稼働させレポート生成などの処理を行わせることで、負荷分散も実現できます。
 - Oracle Universal Installer
 - Oracleソフトウェアのインストールを行うツール。
 - マルチテナント・アーキテクチャ
 - 複数のOracleデータベースが1つのOracleインスタンスを共有する機能。
 - Oracle Clusterware
 - RAC構成(1つのOracleデータベースに対して2つ以上のOracleインスタンスを構成する)において、複数のデータベースサーバーを一群にまとめてクラスタ(一群、塊)として動作させる
 - 通常、Oracle Databaseサーバーは1つのOracleデータベースと1つ以上のOracleインスタンスから構成されます。1つのOracleデータベースと1つのOracleインスタンスからなる構成を「シングル・インスタンス構成」といいます。
 - 1つのOracleデータベースに対して2つ以上のOracleインスタンスを構成することを「Real Application Clusters構成(RAC構成)」といいます。RACでは異なるサーバー上のインスタンスから共有ストレージのデータベース・ファイルにアクセスします。

クライアント・サーバー間の流れ

SQL文

- SQLを使用するときに、データの物理的な格納場所やアクセス順序を意識する必要はありません。「どの表のどの列の値を取り出す」というように、表の名前と条件を指定するだけで目的のデータにアクセスできます。
- Oracle DatabaseではSQLの実行時、オプティマイザ機能(SQLを実行する際に、最も効率的な実行方法を決定する機能)により、目的のデータに最も高速にアクセスするための方法が自動的に決定されます。

SELECT

- [1] ユーザー・プロセスがサーバー・プロセスへSELECT文を送信します。
- [2] サーバー・プロセスは、システム・グローバル領域の共有プールに[1]のSELECT文が格納されているか確認します。格納されている場合は、手順[6]に進みます。
- [3] 格納されていない場合は、SQLの解析処理を行います。
- [4] 実行計画を作成します。
- [5] 解析済みのSQLと実行計画を共有プールに格納します。
- [6] 実行計画に従ってSELECT文を実行し、取得するデータを識別します。この時、SELECT文の検索対象となるデータがデータベース・バッファ・キャッシュにあるか確認し、無い場合はデータファイルからデータをコピーします。データがある場合は「キャッシュヒット」、無い場合は「キャッシュミス」と呼びます。
- [7] サーバー・プロセスは、データベース・バッファ・キャッシュの対象データを取り出し、SELECT文の実行結果としてユーザー・プロセスへ返します。

UPDATE

- [1] ユーザー・プロセスがサーバー・プロセスへUPDATE文を送信します。
- [2] サーバー・プロセスは、システム・グローバル領域の共有プールに[1]のUPDATE文が格納されているか確認します。格納されている場合は、手順[6]に進みます。
- [3] 格納されていない場合は、SQLの解析処理を行います。
- [4] 実行計画を作成します。
- [5] 解析済みのSQLと実行計画を共有プールに格納します。
- [6] 更新の対象となるデータがデータベース・バッファ・キャッシュにあるか確認し、無い場合はデータファイルから対象のデータ・ブロックをコピーします。
- [7] 更新の対象となるデータの行をロックします。
- [8] **サーバープロセスが、REDOログ・バッファに更新内容を書き込みます。**
- [9] **ロールバック時に必要な更新前のデータを、UNDOデータとしてUNDO表領域のUNDOセグメントにコピーします。**
- [10] データベース・バッファ・キャッシュのデータを更新します。
- [11] サーバー・プロセスはユーザー・プロセスに更新完了の通知を行います。
- **この時、データベース・バッファ・キャッシュの更新データは、すぐにデータファイルへ書き込まれるわけではありません。パフォーマンスを考慮し、遅延書き込みが行われます。**

COMMIT

- [1] ユーザー・プロセスがサーバー・プロセスへCOMMIT文を送信します。
- [2] REDOログ・バッファにCOMMITの情報を書き込みます。
- [3] LGWRプロセスがREDOログ・バッファの全てのREDOエントリをREDOログ・ファイルに書き込みます。
- [4] データベース・バッファ・キャッシュのデータ更新前にロックされていた行を解放します。

- 他のユーザーが変更後のデータを参照できる
- [5] サーバー・プロセスはユーザー・プロセスにCOMMIT完了の通知を行います。

トランザクション

- 1つ以上のSQLによるデータベースへの一連の操作のまとまりを「トランザクション」と呼び、トランザクションごとに処理を確定(コミット)または取消し(ロールバック)できます。
- トランザクションには、完全に実行されるか、全く実行されないかのどちらかとなる「Atomicity(原子性)」という性質があります。これより、トランザクションは「1つ以上のSQL文を含むアトミックな論理作業単位」とも呼ばれます。
- SQL*Plusの終了などでセッションを終了するとトランザクションは終了しますが、その逆にトランザクションを終了すると必ずセッションが終了するわけではありません。よって誤りです。トランザクションは、Oracle Database接続後、または直前のトランザクションの終了後、最初のデータを操作するSQL文(SELECT文は除く)実行時に開始され、下記のいずれかによって終了します。
 - COMMIT文またはROLLBACK文の実行
 - DDL文の実行
 - DCL文の実行
 - SQL*Plusの終了
 - システム障害の発生
- Oracle Databaseに接続する場合に、クライアント側で必要な情報
 - 通信プロトコル
 - ホスト名
 - リスナーが使用しているポート番号
 - サービス名

Oracle Database 起動について

起動

[インスタンスの起動(SHUTDOWN → NOMOUNT)]

- 1. サーバー・パラメータ・ファイルまたはテキスト初期化パラメータ・ファイルを読み込み、初期化パラメータの値を取得する
- 2. 初期化パラメータの値に従って、システム・グローバル領域(SGA)を割当て
- 3. バックグラウンド・プロセスを起動する
- 4. アラート・ログ・ファイルをオープンし、ログの出力を開始する

[データベースのマウント(NOMOUNT → MOUNT)]

- 5. 初期化パラメータ「CONTROL_FILES」で指定された格納場所から制御ファイルを読み込む

[データベースのオープン(MOUNT → OPEN)]

- 6. 制御ファイルで指定された格納場所にあるデータファイルとREDOログ・ファイルをオープンする

停止

[データベースのクローズ(OPEN → CLOSE)]

- 1. チェックポイントが発生し、SGA内のデータをデータファイルとREDOログ・ファイルに書き込む
- 2. データファイルとREDOログ・ファイルをクローズする

[データベースのアンマウント(CLOSE → NOMOUNT)]

- 3. データベースをアンマウントしてインスタンスから切り離す
- 4. 制御ファイルをクローズする

[インスタンスの停止(NOMOUNT → SHUTDOWN)]

- 5. SGAをメモリーから削除する
- 6. バックグラウンド・プロセスを停止する

初期化パラメータファイル

- サーバーパラメータファイル（SPFILE）
 - 現在、主に使用される
 - バイナリ形式。ユーザーが直接編集できない
 - 設定値をSQL（ALTER SYSTEMコマンド）で変更可能
 - SQL文またはEM Expressを使用して読み書き
 - 動的パラメータを変更した場合、SPFILEに変更が書き込まれ、再起動後も変更が保持される
- テキスト形式のパラメータファイル（PFILE）
 - テキスト形式。ユーザーがテキストエディタなどで直接編集可能
 - 設定値はSQLで変更不可
- 初期化パラメータファイルは、インスタンスが**NOMOUNT状態**になるときに読まれる
 - SPFILEとPFILEが存在するとき、**SPFILEが優先される**

パラメータ	説明	起動中 の変更
CONTROL_FILES	制御ファイルのファイル名。複数指定可能	×変更不可
DISPATCHERS	ディスパッチャプロセスの構成	○変更可
MEMORY_MAX_TARGET	MEMORY_TARGET初期化パラメータの上限値	×変更不可

パラメータ	説明	起動中 の変更
MEMORY_TARGET	すべてのSGAコンポーネントと全プロセスのPGAの合計サイズ。 自動メモリー管理使用時に設定。	○変更 可
PGA_AGGREGATE_TARGET	全プロセスのPGAの合計サイズの目標値。自動PGAメモリー管理使用時に設定。	○変更 可
SHARED_POOL_SIZE	共有プールのサイズ	○変更 可
SGA_MAX_SIZE	すべてのSGAコンポーネントの合計サイズの上限值	×変更 不可
SGA_TARGET	すべてのSGAコンポーネントの合計サイズ。自動共有メモリー管理使用時に設定する	○変更 可
UNDO_RETENTION	UNDO保持の下限値（秒）	○変更 可

テーブル構造

- リレーショナル・データベースは、データをリレーション(表)に格納するデータベースです。
- 「表」は、「行」と「列」から構成される2次元の構造です。
- 行は関連のあるデータのまとまりで、1行で1件のデータとなります。列はデータの種類を表す項目です。
- 行と列の交わる箇所をフィールドといい、1つのフィールドには1つの値が格納されます。
- また、値が格納されていないフィールドの状態をNULL値といいます。
- リレーショナル・データベースでは、関連のある表同士を、フィールドの値で関連付けることができます。
- 関連付けには主キー、一意キーと外部キーと呼ばれる列を使用します。
- 主キーとなる列には必ず値を格納する必要がありますが、一意キー、外部キーとなる列はNULL値でも構いません。
- 外部キーを持つ表と、外部キーの値を持つ主キーまたは一意キーを持つ表が関連付けられます。
- NULL値を使用して算術演算を行うと、結果はNULL値となります。

用語	説明
主キー（PRIMARY KEY）	表内の行を一意に識別するための列・列の組合せ。 重複した値やNULL値を含めることができない。 1つの表に1つだけ設定できる
一意キー（UNIQUE KEY）	列内の値が一意である列。 主キーと異なり、NULL値を含めることができる
外部キー（FOREIGN KEY）	同じ表や別表の主キーまたは一意キーを参照する列。 外部キーに設定された列には、参照する主キーまたは一意キーに格納されている値、NULL値を格納できる

プロセス 2

- DBWn(データベース・ライター)プロセスがデータベース・バッファキャッシュ内の使用済みバッファ(変更されたデータが格納されているバッファ)をデータファイルへ書き込むタイミングは、データの更新、削除やトランザクションのコミットと連動していません。次の場合に、DBWnはデータベース・バッファキャッシュ内の使用済みバッファをデータファイルへ書き込みます。
 - チェックポイント・イベントの発生時
 - データベース・バッファキャッシュ内の再利用可能バッファが見つからない時
- REDOログ・バッファはデータの変更履歴であるREDOエントリが格納される領域です。データの追加や更新時に、サーバー・プロセスによって、REDOエントリがREDOログ・バッファに書き込まれます。
 - UPDATE文の実行時に書き込まれる
 - データの変更履歴が格納される
 - サーバー・プロセスによって書き込まれる
- CKPT(チェックポイント)プロセスはOracleインスタンスを構成するバックグラウンド・プロセスの1つで、チェックポイント・イベント発生時に、使用済みバッファをデータファイルへ書き込むようDBWnへ指示を出します。同時にチェックポイント・イベントが発生したことを制御ファイルとデータファイルへ書き込みます。

ログ

- アラート・ログは、Oracle Databaseでの管理作業やエラーの情報が記録されるXML形式のログ・ファイルです。アラート・ログに記録される主な項目は、次のとおりです。
 - 発生したすべての内部エラー(ORA-00600)、ブロック破損エラー(ORA-01578)、デッドロック・エラー(ORA-00060)
 - CREATE/ALTER/DROP文(DDL)、STARTUP/SHUTDOWN文、ARCHIVELOG文などの管理操作
 - インスタンスの起動時点でデフォルト値から変更のあった初期化パラメータの値

コマンドの類

共有プールのメモリーを最低300Mは確保しなければなりません。直ちに現行のインスタンスに設定を適用して、再起動後もその値を保持するにはどうしますか。なお、インスタンス起動時にはSPFILEを読み込んだものとしします。

```
ALTER SYSTEM SET shared_pool_size = 300M;
```

db_cache_sizeを150Mにセットする。インスタンス起動時にPFILEを読み込んでいる。

```
ALTER SYSTEM SET db_cache_size = 150M;  #デフォルト
or
ALTER SYSTEM SET db_cache_size = 150M SCOPE = MEMORY;  #現行を指定
```

- PFILEは手動編集（コマンドで変えられるのは現行のインスタンスのみ）

オプション 説明

SPFILE	SPFileのみに変更が適用される
MEMORY	稼働中のインスタンスのみに変更が適用される
BOTH	SPFileと稼働中のインスタンスの両方に変更が適用される

Oracleソフトウェアのインストール時、Linux環境において次のスクリプトを実行しますが、どのユーザーで実行しますか。

- root
 - orainstRoot.sh
 - インベントリ(インストールされたOracleソフトウェアの情報)の場所を記録する「oralnst.loc」というインベントリ・ポインタ・ファイルを作成するスクリプト。インベントリ・ポインタ・ファイルが存在しない場合にのみ実行を促される
 - root.sh
 - oratabファイル(ORACLE_HOMEやORACLE_SIDが記録される)の作成と、Oracle用の環境変数(OSに設定する、システムの属性を記録している変数)を変更するスクリプトのコピーを実行する

Enterprise Manager Database ExpressのHTTPSポートを構成する方法

```
EXEC DBMS_XDB_CONFIG.setHTTPSPort(ポート番号)
```

- アクセスURL
 - https://ping-t.com:5500/em

SQL*Plusで作業中にOSの「ls」コマンドを実行する方法

```
host ls
```

Enterprise Manager Database Expressが使用するポート番号を確認する方法

- コマンドラインで「lsnrctl status リスナー名」を実行する
- SQL文「SELECT dbms_xdb_config.gethttpsport() FROM dual;」を実行する

SYSユーザーでSQL*Plusからデータベースに接続する際の接続方法

- SQL*Plus起動時に指定


```
$ sqlplus sys/パスワード as sysdba
```
- SQL*PlusのCONNECTコマンド実行時に指定


```
SQL> sqlplus /nolog
SQL> connect sys/パスワード as sysdba
```

SQL*Plusでスクリプトを実行する方法として、正しいものはどれですか

```
SQL> @スクリプト名
SQL> START スクリプト名
```

初期データベースの作成時に構成されるデフォルトのリスナーのポート番号

- 1521

listener.ora

```
LISTENER1 =
(DESCRIPTION_LIST =
(DESCRIPTION =
(ADDRESS = (PROTOCOL = TCP)(HOST = ping-t.com)(PORT = 1522))
)
)
```

クライアントからSQL*Plusを起動する際に、次のコマンド

```
# ユーザー名/パスワード@ホスト名[:ポート番号]/[サービス名]

sqlplus pingt/oracle@db.pingt.com/orcl.pingt.com
```