

【ORACLE MASTER DBA12c DBA Bronze】書籍学習用

ノートブック: 210_×E
作成: 2016/01/06 22:10 更新: 2019/07/19 23:40
作成者: Tsutsui tomoaki
URL: <http://www.weblio.jp/content/DDL>

タイトル : ORACLE MASTER DBA12c Bronze

chapter 01 【Oracleデータベース管理の概要】

01. リレーショナルデータベースの構造およびSQLの使用法

3. データとデータを関連付けるリレーションシップ

リレーションシップ: ある列と格納した値によって示される他の行との関連のこと

4. リレーショナルデータベースの用語

外部キー: あるデータから関連する他のデータを参照するために使用する列
外部キーの参照先の列には主キー制約または一意制約が設定されている

5. SQLとは?

RDBMSに対するの命令文のこと。

タイプ	説明	例
データ検索 広くはDMLに含まれる。ORACLEでは別のものとして扱う	テーブルに格納されたデータを検索し表示する	SELECT
データ操作言語 (DML文) DML (Data Manipulation Language)	テーブルに格納されたデータを追加、更新、削除する	INSERT UPDATE DELETE MERGE
データ定義言語 (DDL文) Data Definition Language	テーブルの定義やデータベースの管理を行う	CREATE TABLE ALTER USER ほか
データ制御言語 (DCL文) Data Control Language	データベースに対する権限の設定を行う	GRANT REVOKE
トランザクション制御文 これもDCLに含まれる?	データベーストランザクションの制御を行う	COMMIT ROLLBACK SAVEPOINT

02. OracleインスタンスおよびOracleDatabaseのコンポーネント

1. Oracle Databaseサーバーの構成要素

- データベースサーバーは下記からなる
 - Oracleインスタンス
OracleDatabaseを管理するためのメモリーやプロセスの集まり
 - OracleDatabase
ハードディスク上に格納されたデータのセット
- シングルインスタンス構成
1つのOracleインスタンスと1つのOracleDatabase
- RAC構成 (Real Application Clusters構成)
1つのOracleDatabaseに対して、2つ以上のOracleインスタンス

○ Column: OracleインスタンスとOracleDatabaseの関係

- マルチテナントアーキテクチャ
12cから導入されたこの機能により、1つのインスタンスに対して複数のデータベースを持つような構成が可能になりました
(試験対象外)

2. Oracleインスタンス

- 下記2つから構成される
 - 「共有メモリー」: システムグローバル領域 (SGA) と呼ばれる
 - 共有プール
 - データベースバッファキャッシュ
 - REDOログバッファ
 - ラージプール
 - Javaプール
 - ストリームプール
 - 「プロセス」: 裏方として動作する複数のプロセスでバックグラウンドプロセス群と呼ばれる
 - データベースライタープロセス (DBWn)
 - ログライタープロセス (LGWR)
 - システムモニタープロセス (SMON)
 - プロセスモニタープロセス (PMON)

- チェックポイントプロセス (CKPT)
- アーカイバプロセス (ARCn)

3. Oracle Database

- ディスクに配置される3種類のファイルをOracleDatabaseという。
 - データファイル：表や索引などのデータが格納される
 - REDOログファイル：データに対する変更履歴が格納される
 - 制御ファイル：データファイルやREDOログファイルの物理構造に関する情報が格納される

03. OracleDatabaseの管理に使用するツールの定義

1. データベース管理者の役割

データベースを管理する人のことをDBA(Database Administrator)と呼びます

2. データベース管理ツール

- ◎Oracle Universal Installer (OUI)
- ◎Oracle Database Configuration Assistant (DBCA)
- ◎Database Upgrade Assistant
- ◎Oracle Net Manager
- ◎Oracle Enterprise Manager (EM)
- ◎SQL Developer
- ◎SQL*Plus
- ◎Recovery Manager (RMAN)
- ◎Data Pump
- ◎SQL*Loader

04. XXXXX

05. XXXXX

06. XXXXX

07. XXXXX

08. XXXXX

09. XXXXX

10. XXXXX

○ Column:

	chapter 02 【Oracleデータベースのインストールおよびデータベースの作成】
--	--

01. インストールの事前準備

1. インストール時の前提条件

Oracle Universal Installer (OUI)はOracleソフトウェアをインストールする際にインストールに必要な要件を確認します。

- 1GB以上の物理メモリが使用可能であること
- 物理メモリの1.5~2倍程度の仮想メモリ領域が用意されていること
- /tmpディレクトリに1GB以上の領域が使用可能であること
- Oracleソフトウェアのインストール先に6.1GB以上の空き領域があること
- 初期データベースを作成する場合は、データベース用の領域として、2GB以上の領域が使用可能であること
- OSには適切なサービスパックまたはパッチが当てられていること
- 適切なファイルシステムが使用されていること

2. 設定が必要な環境変数

- ORACLE_BASE

OFAにおけるOracle関連のディレクトリ構造のベースとなるディレクトリパスを設定

- ORACLE_HOME

Oracleソフトウェアのインストール先のディレクトリを設定。1つのサーバーマシン上に複数のOracleソフトウェアがインストール可能、

- ORACLE_SID

Oracleインスタンスの名前を設定

- LD_LIBRARY_PATH

共有オブジェクト・ライブラリーの場所を設定する。UNIX系OSのみで設定が必要

※OFA (Optimal Flexible Architecture)

オラクル社が推奨するディレクトリ構造のこと。

3. 事前に作成するOSユーザーおよびOSユーザーグループ

◎OSユーザーグループ

グループ名	概要
oinstall	Oracleソフトウェアの各ファイルを所有するOSユーザーグループ
dba	sysdbaとしてログインするユーザーを認証するために使用する。dbaグループに所属するOSユーザーでログインした場合、sysdbaでのログインが可能
oper	sysoperとしてログインするユーザーを認証するために使用する。使い方はdbaグループと同様

◎OSユーザー

UNIX系OSにOracleソフトウェアをインストールする場合は、rootユーザーは使用できません。
Oracle用の新規OSユーザーを作成しなければいけない。
→一般的には「oracle」を使用します

02. Oracleソフトウェアのインストール

①OUIの起動

UNIX系はマウントしたDVDドライブにあるrunInstallerを実行する

Windowsはsetup.exeを実行

②セキュリティアップデートの構成
オプションなので未入力でもOK！電子メールアドレスが未入力ですという警告が出るが、そのまま継続して問題ない

③Software Updateの設定
オプション。設定しない場合は「ソフトウェア更新のスキップを選択」

④インストールオプションの選択

インストールとともにデータベースを作成したり、アップグレードしたりすることができる。
「データベースの作成および構成」ソフトウェアのインストールとともにデータベースを作成

⑤システムクラスの選択

構築するデータベースの規模に応じて、

「デスクトップクラス」

「サーバークラス」

から適切なものを選択します。

⑥Gridインストールオプションの選択

今回は「単一インスタンスデータベースのインストール」を選択

⑦インストールタイプの選択

「標準インストール」今回はこれ！

「拡張インストール」様々なオプションを指定することができます

⑧初期設定項目の入力

入力項目	説明
Oracleベース	前述のORACLE_BASEのパスを指定する
ソフトウェアの場所	同じくソフトウェアのインストール先であるORACLE_HOMEのパスを指定する
記憶域のタイプ	データベースファイルの格納先のタイプを指定する。標準はファイル・システム
データベースエディション	インストールするOracle Databaseサーバーのエディションを指定する。正しくライセンスを受けているエディションを指定しなければならない
OSDBAグループ	データベースの起動や停止を実行するユーザーが属するOSユーザーグループ（通常はdbaグループ）を指定する
グローバルデータベース名	初期データベースの名称を指定する
管理パスワード	初期データベースで作成される主なユーザーのパスワードを指定する。ここでパスワードを指定するユーザーはSYS、SYSTEM、SYSMAN、DBSNMPの4つであり、すべて同一のパスワードが設定される
コンテナデータとして作成	12cから登場したプラグブルデータベースを構築する際にチェックを入れる

⑨インベントリディレクトリの指定

UNIX系OSのみ、最初にOracleソフトウェアをインストールする際に表示されるOracleインベントリは場所は自由に指定できますが、ディレクトリ名はoraInventoryを指定します。

⑩インストール情報の確認
サマリー画面が表示されます。

⑪スクリプトの実行

UNIX系OSにOracleソフトウェアをインストールした際は、インストール後に構成スクリプトの実行が要求されます。
rootユーザーでログインして以下の2つのスクリプトを実行します

```

・ /u01/app/oraInventory/orainstRoot.sh
インベントリポインタファイルの作成
→インストール時に作成したインベントリが格納された場所を格納したファイル
・ $ORACLE_HOME/root.sh
oratabファイルの作成を行います。
→作成したデータベースに対して自動起動の設定を行う際に使用されます

```

Windowsの場合はインベントリの場所や自動起動の設定はレジストリで行われているため、スクリプトの実行は不要。

⑫コンフィギュレーションアシスタントの呼び出し

初期データベース作成を実行すると、インストールプロセスの中で、Oracle Net Configuration Assistant (NetCA) および、Oracle Database Configuration Assistant (DBCA) が呼ばれる。
NetCA：基本的なネットワーク設定。デフォルトリスナの作成とリスナーに対する接続設定
DBCA：データベースの作成をおこなうアシスタント。初期データベースの構築を行う

⑬データベース作成の終了とパスワードの変更

⑭インストール作業の終了

03. Oracleデータベースの作成

DBCAで実行可能な作業は次の5つ

- ・データベースの作成
- ・データベースオプションの構成（範囲外）
- ・データベースの削除
- ・データベースの管理
- ・プラグフルデータベースの管理（範囲外）

1. データベースの新規作成

①作成モードの指定

「拡張モード」を指定！

②テンプレートの選択

標準で使用できるテンプレートは③種類ですが、テンプレートの管理機能により、新しくテンプレートを作成することもできる。

テンプレート	説明
汎用またはトランザクション処理	一般的な用途で使用するデータベースを構築する際に使用されるテンプレート
カスタムデータベース	データベースの設定を細かく定義するとき使用する。いわゆるオーダーメイドのデータベースが作成可能
データウェアハウス	データ・ウェアハウスなど、情報系のシステムで使用するデータベースを構築する際に使用する

今回は「汎用またはトランザクション処理」を選択!!

③データベース識別情報の設定

グローバルデータベース名：Oracleデータベースの名称・名前をつける際にはネットワークを識別するドメイン名を組み合わせた

「db_name.db_domain」の正式が一般的

SID : システム識別子のことで、インスタンス名を指します！

一般にデータベース名と同じものを使用するため、グローバルデータベース名を入力すると、動的にその最初のピリオドまでの文字列が表示されま

す。

④管理オプションの選択

データベース管理オプションとして、

- ・Enterprise Manager Database Expressを 使用する しない
- ・Enterprise Manager Cloud Controlを 使用する しない

1つのデータベースのみを管理する
複数のデータベースを集中管理できる大規模環境

⑤管理者アカウントのパスワード設定

SYSやSYSTEMなどの管理者ユーザーのパスワードを設定します。
それぞれのユーザーに対して個別にパスワードを設定することも、

一括で同一のパスワードを設定することも可能

*一括 : Calpis8946

*Oracleホームのユーザーパス : orauser

⑥ネットワークの構成

使用するリスナーの選択を行います。

既存のリスナーを選択したり、

新しいリスナーを作成したりすることができます。

Oracle Database 11gではDBCAではなくネットワーク構成用ツールでこれらの設定を行っていた。

⑦記憶域の設定

「データベースファイル」「リカバリ関連のファイル」に関し、格納する記憶領域のタイプや場所などを指定します。

・「データベースファイル」に関連して設定できるOracle Managed Filesは第6章、

・「リカバリ関連のファイル」に関しては第9章で詳細を説明

⑧データベースオプションの指定

「データベースオプション」画面では、サンプルデータの作成や、Database Vault、Label Securityに関する設定を行うことができます。

サンプルスキーマにチェックを入れて進む。

EXAMPLEという名前の表領域が作成され、ソコにサンプルのオブジェクトが作成されます。

⑨初期化パラメータの設定(メモリー)

データベースがどの用意動作するかを規定する初期化パラメータのいくつかを設定して行きます。

最初はメモリータブが選択されているが、

その他に、

「サイズ指定」

「キャラクタセット」

「接続モード」を選択し、それぞれの項目を設定することができます。

⑩メモリー画面

⑩初期化パラメータの設定(サイズ指定)

データベースで使用する

- ・標準データブロックのサイズ
- ・データベース上で起動可能なプロセスの最大数 を指定！

(今回は「汎用およびトランザクション処理」テンプレートを使用しているためブロックサイズは入力不可)

⑪初期化パラメータの設定(キャラクタ・セット)

データベースで使用する言語設定

⑫初期化パラメータの設定(接続モード)

ユーザーがデータベースに接続する際の接続モードとして、専用サーバーモードのみをサポートするか
専用サーバーモードと共有サーバーモードの2つをサポートするのを選択します。

○ Column: メモ

Oracle Databaseのデータはすべて、データ格納の基本単位である「データブロック」に格納されます。

データブロックはデータファイルが作成された時点で定義され、以後はサイズの変更ができません。

Oracle社が提供する事前定義済みテンプレートはすでに
作成済みのデータファイルを使用するため、データ・ブロック・サイズの指定はできません。。

⑬データベース作成オプションの指定

通常は「データベースの作成」にチェックを入れますが、
そのほかにも、設定した内容を元にテンプレートを作成したり、
データベース作成用のスクリプトファイルを生成したりすることも可能です。

⑭構成の確認とデータベース作成操作の完了

「サマリー」画面にて設定内容を確認します。

◆2. データベースの削除

DBCAを使用するとデータベースの削除操作を簡単に実行できます。

DBCAで削除操作をした場合

「データベース構成情報の削除」(論理情報の削除)
「データベース構成ファイルの削除」(物理ファイルの削除)
という2つの処理が実行されます。
物理的にファイルが削除されるため、これまでデータベースが専有していた領域が開放され、別のアプリケーションでも利用可能な状態になります。DBCAを使用して削除できるデータベースはDBCAによって作成されたものに限られます。SQLを使用して手動で作成したものは手動で削除しなければなりません、。

◆3. テンプレートの管理

データベースを作成するための構成情報をテンプレートとして保存し、利用することができます。

DBCAで使用可能なテンプレートの特徴は以下のとおり

- ・テンプレートの実態はXML形式のファイルである
- ・\$ORACLE_HOME/assistants/dbca/templatesディレクトリに格納される
- ・オラクル社が提供する事前定義済みテンプレートだけではなく、ユーザーが個別に作成するテンプレートも使用することができる
- ・テンプレートには、データベース構成情報に加え、データベース・ファイルそのものを保持するシードテンプレートと、データベースの構成情報しか持たない非シードテンプレートが存在する
- ・事前定義済みテンプレートのうち、「汎用またはトランザクション処理」「データ・ウェアハウス」の2つはシードテンプレートである
- ・同じく「カスタムデータベース」は非シードテンプレートである

※重要※

シードテンプレート：データベースファイルを単純にコピーするだけなので高速に処理できるが、標準のデータブロックサイズが変更できないなど柔軟性に乏しい
非シードテンプレート：データベースの作成には時間がかかるが、ブロックサイズを始めパラメータの値を自由に設定可能なので、様々なユーザーのニーズに応えることができます。

～まとめ～

- ・Oracleソフトウェアをインストールする前に、前提条件の確認や、環境変数の設定が必要である
- ・OUIによるインストール条件のチェックで警告が出た場合、インストールは継続できるが、失敗する可能性があるので注意する
- ・OUIを使用してインストールを行う場合、初期データベースの作成を実行すると、NetCAやDBCAが呼び出される
- ・DBCAではデータベースの新規作成の他に、データベースの削除、テンプレートの管理などを行うことができる
- ・DBCAではテンプレートを使用することでデータベースを簡単に作成できる
- ・テンプレートには、シードテンプレートと非シードテンプレートの2種類がある

・データベースで使用する記憶域には、「ファイルシステム」(デフォルト)と「自動ストレージ管理(ASM)」がある。!

04. DBCAを利用したテンプレート管理

- テンプレートとは
テンプレートの実態はXMLファイル
- ・データベースオプション
- ・初期化パラメータ
- ・記憶域属性

パス>

D:\app\oraiser\product\12.1.0\dbhome_1\assistants\dbca\templates

■DBCAによるテンプレートの作成・管理操作

- Oracle社によって用意されているテンプレート
- 汎用またはトランザクション処理(デフォルト)：OLTP向けのデータベースを構成する場合
- データウェアハウス：大量データを処理する場合
- カスタムデータベース：処理内容に合わせて個別に設定できる

■テンプレートのタイプ

- ・シードテンプレート
- 既存のデータベース(シードデータベース)の構造及び物理ファイルの両方が含まれているテンプレート
- 物理ファイルは作成済み

- ・非シードテンプレート
- 作成するデータベースの特性が含まれたテンプレート

05. XXXXX

06. XXXXX

07. XXXXX

08. XXXXX

09. XXXXX

10. XXXXX

○ Column:

chapter 03 【Oracle Enterprise Manager Database Express 及びSQL管理ツールの使用】

Enterprise Manager Database Express (EM Express) : Webブラウザで使用可能
SQLDeveloper : 一部データベース管理タスクやスキーマ・オブジェクトの参照、SQLの実行ができる

01. Oracle Enterprise Manager Database Expressへのアクセス

◆1. 管理フレームワーク

コンポーネント	説明
データベース・インスタンス	EnterpriseManagerの管理対象
リスナー	管理対象であるデータベース・インスタンスへの接続を橋渡しする
管理インターフェース	様々な管理機能を提供するプログラム。管理インターフェースには、Cloud Controlに接続するエージェントとDatabase Expressの2種類がある

Cloud Controlは複数のデータベース・サーバーやアプリケーション・サーバーなどを統合管理することのできる管理ツールで、Oracle Databaseサーバーとは別にインストールが必要です。
今回は単一データベースを管理するため、付属のEM Expressを使用する

◆2. Database Controlによる管理の仕組み

データベース管理者がWebブラウザからEnterprise ManagerのURLにアクセスすると、実際には、Oracle Databaseサーバーが稼働しているマシン上で起動されているEM Expressプロセスに接続します！！
EM ExpressプロセスはOracle Databaseサーバーに接続しますが、このときEM ExpressプロセスはOracle Netの機能を利用して、最初にリスナーと呼ばれるプロセスへの接続を行います。

→リスナーを経由してデータベースへの接続を要求する

◆3. EM Expressのポート構成とアクセス

DBCAを用いてデータベースを作成すると、最後にEnterprise ManagerのURLが表示されます。この時URLが表示されなかったり、表示されたURLに対してアクセス先のポート番号を変更する場合は、DBMS_XDB_CONFIG.setHTTPSPortプロシージャを用いて設定することができます。

- ①リスナーが起動されていることを確認する
- ②DISPATCHERSパラメータにPROTOCOL=TCPが含まれていることを確認する
- ③DBMS_XDB_CONFIG.setHTTPSPortプロシージャを実行し、ポート番号を設定する

```
dispatchers="(PROTOCOL=TCP)(SERVICE=<sid>XDB)"
```

```
exec DBMS_XDB_CONFIG.setHTTPSPort(5500);
```

▼EM ExpressのURL
<https://localhost:5500/em>

▼EM Expressのポート番号検索
SELECT dbms_xdb_config.gethttpsport FROM DUAL;

02. SQL*Plusの使用

◆1. SQL*Plusへのアクセス

▲ログインせずに起動！
sqlplus /nolog

▲SQL*Plusの接続コマンド
CONNECT system(ユーザ)/oracle(パス)

▲SQL*Plusの接続コマンド(SYSDBA)
CONNECT sys/oracle AS SYSDBA

sysユーザーのようにSYSDBA権限を使用してログインするユーザーはデータベースを起動するために、データベースが停止している状態で接続する必要がありユーザーの認証はデータベースの外部で行う必要がある。。。意味わかってない

- ・OS認証
 - ・パスワードファイル認証
- デフォルトではOS認証が有効になっている

▲SQL*Plusの接続コマンド(SYSDBA)
CONNECT / AS SYSDBA

なおパスワードファイル認証は、暗号化されたパスワードファイルと呼ばれるファイルに認証に必要な情報を書き込んで

ユーザー認証を行う方式です。

◆2. SQL*Plusで実行可能な操作

03. SQL Developerの使用

オラクル社が無償で提供するGUIベースのSQL実行ツール
もともとSQLやPL/SQLを用いて開発を行うためのツールとしてリリースされましたが、現在のバージョンでは
SQLやPL/SQLによるデータベースオブジェクトの操作だけではなく、幾つかの管理操作が実行できるようになりました。

◆1. SQL DeveloperによるOracle Databaseへの接続

\$ORACLE_HOME/sqldeveloperにあるsqldeveloper.shを起動する

```
接続名      : DefaultConnect
ユーザー名  : sys
パスワード  : Galpis8946
接続タイプ  : TNS
ネットワーク: TTTTWTUWI
```

◆2. DBA接続の作成

SQL Developerからデータベースに接続し、管理操作を行うためには、
前述の設定とは別にDBA接続を作成する必要があります。

DBA接続の作成手順は以下のとおりです。

- ①「表示」メニューから「DBA」を選択し、「DBA」タブページを表示する
- ②「DBA」タブページで「接続」を右クリックし、表示されたメニューの「接続の作成」を選択する
- ③事前に作成した「接続」を選択し、「OK」をクリックする

この設定により下記が可能

- ・Oracle Databaseサーバーの起動・停止
- ・データベース構成(初期化パラメータの設定、自動UNDO管理の設定など)
- ・データベースのステータスの表示
- ・RMANによるバックアップとリカバリの設定
- ・セキュリティの設定
- ・データポンプの使用
- ・パフォーマンスの監視
- ・リソース・マネージャの設定
- ・記憶域の設定

SQL Developerの使用パターンには以下の2つがある

- ・通常モードでスキーマオブジェクトを参照、作成、編集および削除するようなスキーマオブジェクトに対する作業を実行する
- ・DBAナビゲーターを使用してDBA接続することでOracleデータベースの起動・停止を含むデータベース管理タスクを実行することができる

OSコマンドを使用する場合は

Host コマンドを使用する

バッチファイルを実行する場合は

『@ファイル名』で指定する

～まとめ～

- EM Expressを使用するためには、リスナーおよびOracleインスタンスの起動が必要である
- EM Expressに接続する際には、https://ホスト名:ポート番号/emと指定する
- 接続先のポート番号を確認するには、DBMS_XDB_CONFIG.getHTTPSPortプロシージャを実行する
- EM_EXPRESS_BASICロールを付与するとEM Expressの参照権限を与えることができる
- EM_EXPRESS_ALLロールを付与するとEM Expressによる管理権限を与えることができる
- SQL*Plusを使用すると、コマンドベースでデータベースの管理操作が実行できる
- SQL*Plusでは、対話側のデータベース管理の他に、スクリプトファイルを実行してバッチ処理型の管理操作ができる
- SQL Developerを使用すると、データベースの起動などの管理操作を実行する事ができる

04. XXXXX

05. XXXXX

06. XXXXX

07. XXXXX

08. XXXXX

○ Column:

chapter 04 【Oracleネットワーク環境の構成】

クライアントとOracleDatabaseサーバーが通信を確立する仕組みについて学習します。

01. Oracle Databaseサーバーを使用する際のネットワーク構成

Oracle Net : クライアント側のアプリケーションからネットワークを介してOracle Databaseサーバーと通信する際に使用される通信用ソフトウェア
 Oracle Databaseサーバーとの通信はOracle Netによって実現されている為、Oracle Netのインストール先はOracle Databaseサーバだけでなく、クライアント側も必要
 ユーザーが意識しないところで使用されるソフトウェアなので、その存在をあまり意識する必要はありませんが、Oracle Databaseサーバーと接続する環境を構築する場合はその存在を知っておく必要があります。
 インストール形態
 ・クライアント/サーバー・アプリケーションによる接続
 ・Webアプリケーションによる接続

◆1. クライアント/サーバーアプリケーションによる接続
 企業内のネットワーク(イントラネット)においてOracleDatabaseサーバーに接続する場合などに使用されます。
 クライアントとサーバーの両方にOracle Netがインストールされている必要がある。

◆2. Webアプリケーションによる接続
 クライアントがWebブラウザからWebアプリケーション・サーバーを経由してOracleDatabaseサーバーに接続する形態です。
 クライアントとOracle Databaseサーバーの間に中間層となるWebアプリケーションサーバーが配置されます。
 この場合はクライアントにはOracle Netは不要ということになります。

●重要●
 Oracle Net : Oracle Databaseが通信をする際に使用されるネットワーク用のソフトウェアコンポーネントです。
 どのマシンにOracle Netをインストールする必要があるか、接続形態による違いを把握しておく必要があります。

02. Oracle Databaseサーバーに接続する仕組み

ユーザープロセス : Oracle Databaseに対してSQLを発行するプロセス。具体的にはSQL*Plusなどのアプリケーションが該当します。
 サーバープロセス : ユーザープロセスが発行したSQLをサーバー側で受け取り、処理を行って結果を返すプロセスです。

□メモ

サーバープロセスには

- ・「専用」
- ・「共有」

の2種類があります。それぞれの特徴は下記のとおり
 専用サーバープロセス…「1つのセッションから連続してSQLが実行される」という状況に適している
 1つのユーザープロセス専用のサーバープロセスが処理する
 ユーザープロセスからの接続要求によってサーバー側で起動し、セッションが切断されると終了

共有サーバープロセス…「多くのセッションが確立されているが、1つのセッションから実行されるSQLは多くない」という状況に適しています。
 サーバー側で事前に複数の共有サーバープロセスを起動しておき、1つの共有サーバープロセスが複数のユーザープロセスから実行されたSQLを処理します。
 アイドル状態のサーバープロセスを減らすことができ、サーバー側のリソースを効率よく使用できるメリットが得られます。

データベースが「共有サーバーモード」の設定になっている場合でも特定のクライアントのセッションのみ専用サーバープロセスで処理することは可能!!!!

◆1. 接続要求を受信するリスナー
 セッション : ユーザープロセスとOracleインスタンスとの間に保持する特定の通信経路のことを表します。
 セッションを確立する際に、ユーザープロセスはサーバー側で動作しているリスナーと呼ばれるプロセスに対して接続要求を送信。
 リスナーはクライアントからの接続要求を受信するプロセスで、クライアントからの接続要求を受けるためにサーバー側で起動しておく必要があります。

* 処理順番 *

①クライアント側のアプリケーションからリスナーに対して接続要求が送信される
 ②サーバー側で起動して常駐しているリスナーが接続要求を受信し、新しくサーバープロセスを起動する
 ③起動されたサーバープロセスとの間でセッションを確立する
 ユーザープロセスとサーバープロセスが通信することになるが、この後はリスナーはそれらの通信処理に関与することはありません!

●重要●
 サーバーがクライアントからの接続要求を受信するためには、サーバー側でリスナーを起動しておく必要があります。
 また一度接続してしまえば、リスナーを停止しても既存の接続に影響はありません。

03. リスナーの管理

サーバー側でネットワーク接続を可能にするために行う作業は下記②つ

- ・リスナーを構成する
- ・リスナーを起動する

◆1. リスナーを構成する

Oracleソフトウェアをインストールする際に、同時にデータベースを作成すると、Oracle Net Configuration Assistantと呼ばれるネットワーク管理ツールが起動して、listener.oraというリスナー校正用のファイルを作成し、LISTENERという名前のデフォルトリスナーを構成します。このようにリスナーの構成は自動的に行われるようになっています。

メモ：リスナーの構成

listener.oraには
「リスナーが動作するコンピュータのホスト名」
「リスナーが接続要求を受けるポート番号」などが定義されている
リスナーは起動時にこのファイルを読み込んで動作する為、
「リスナーを構成する」作業は「listener.oraを編集する作業」だと言える。

◆2. リスナーを起動する。

リスナーの起動や停止はリスナー制御ユーティリティ (LSNRCTL) を使用します

◆3. リスナー制御ユーティリティ (LSNRCTL)

→コマンドラインユーティリティ
リスナーの起動と停止、ステータスの確認と言った管理作業を行うことができます。
OSのプロンプト上で、lsnrctl と入力することにより起動します。

コマンド	説明
start [リスナー]	リスナーを起動
stop [リスナー]	リスナーを停止
status [リスナー]	リスナーのステータス情報(起動しているか、など)を表示
services [リスナー]	リスナーが接続をサポートしているサービス(接続先となるデータベース)を表示

コマンドの最後のリスナーは省略可能

リスナーを省略した場合、デフォルトリスナーを操作する事になります。

- ・名前はLISTENERである
- ・TCP/IPプロトコルをサポートする
- ・1521番ポートをリスニングする

04. Oracle Databaseサーバーに接続するためのクライアントの構成

1. クライアントがサーバーに接続する際に必要な情報

プロトコル名 : クライアントとサーバーが通信する際に使用する通信プロトコル
ホスト名 : リスナーが起動しているコンピュータのホスト名(接続要求を送信する宛先のホスト名)
ポート番号 : リスナーが接続要求を待ち受けているポート番号
DB・サービス名 : 接続先のOracle Databaseサーバーを特定する名前

サービス名は複数定義することができる

Orcl.edifirst.com
Sales.edifirst.com
Finance.edifirst.com

各サービス名を定義して処理ごとのパフォーマンス情報を収集したりできます。

所定のフォーマットで表した記述のことを接続記述子と言います

```
TUTTTUWI =  
(DESCRIPTION =  
  (ADDRESS = (PROTOCOL = TCP) (HOST = localhost) (PORT = 1521))  
  (CONNECT_DATA =  
    (SERVER = DEDICATED)  
    (SERVICE_NAME = tutttuwi.com)  
  )  
)
```

* 接続文字列

```
CONNECT hr[ユーザ]/passwd[パス]@orclsession[接続識別子]
```

接続識別子 : 接続先を識別する記述のこと

接続識別子の代わりに接続記述子を指定することも可能であるが、

指定する内容が長くなり不便

ネットサービス名 : orclsession

→文字列に対応付けられた接続記述子を取得することができるようになっています。

名前解決 : 「指定した接続識別子から、対応する接続記述子を取得すること」

名前解決を実現する方法のことを「ネーミングメソッド」と言います。

Oracleでは次のネーミングメソッドを採用している

Oracle Netがサポートするネーミングメソッド

- ・ローカルネーミング : ネットサービス名(接続識別子)と接続記述子の対応関係を、クライアントに存在するtnsnames.oraファイルに定義する方法
- ・簡易接続ネーミング : ホスト名、ポート番号、サービス名を用いたTCP/IP接続文字列を使用して接続する方法。事前構成は不要。
- ・ディレクトリネーミング : 名前解決にLDAP準拠のディレクトリサーバーを使用する方法
- ・外部ネーミング : 名前解決に、Oracle NetをサポートしているOracle Database以外のネーミングサービスを使用する方法

2. ローカルネーミング

クライアントコンピュータに tnsnames.ora と呼ばれるファイルを配置

このファイル内にネットサービス名(接続識別子)と接続記述子の対応関係を定義しておく、

名前解決を行う仕組みのこと

#####

```
①TUTTTUWI =  
(DESCRIPTION =  
  (ADDRESS = ((②PROTOCOL = TCP) (③HOST = localhost) (④PORT = 1521))  
  (CONNECT_DATA =  
    (SERVER = DEDICATED)  
    (⑤SERVICE_NAME = tutttuwi.com)  
  )  
)
```

#####

TUTTTUWI : ネットサービス名
ADDRESS以降が接続要求を送信する宛先(リスナー)に関する情報
CONNECT_DATA以降が接続先のデータベースを指定する情報
① ネットサービス名
② リスナーが使用するプロトコル
③ リスナーの存在しているホスト名
④ リスナーのポート番号
⑤ データベースサービス名(グローバルデータベース名)

** tnsname.ora の格納先**
\$ORACLE_HOME/network/adminに格納される

3. 簡易接続ネーミング
事前構成なしで所定のフォーマットで
接続識別子に直接指定する方法

簡易接続ネーミングの構文
CONNECT username/password@hostname [:port] [/service_name]

- ① データベースのユーザー名
② 指定したデータベースユーザーのパスワード
③ リスナーが稼働しているホストの名前
④ リスナーのポート番号(デフォルトは1521となるため、リスナーが1521番のポートを使用している場合は省略可能)
⑤ データベースサービス名(デフォルトは③で指定したホスト名)

- 重要●
・事前に構成する必要がない
・サポートするプロトコルはTCP/IPのみである

～まとめ～

- ・クライアントからOracle Databaseサーバーに接続するためには、両者にOracle Netをインストールする必要がある
- ・クライアントからの接続要求を受けるためにはサーバー側でリスナーが起動している必要がある
- ・クライアントは接続先を特定する為に接続記述子を指定する必要がある
- ・指定した接続識別子から接続記述子を取得することを名前解決という
- ・名前解決の方法をネーミングメソッドという
- ・クライアントのtnsnames.oraファイルを使用するネーミングメソッドをローカルネーミングという
- ・事前構成を必要とせず接続の際にTCP/IP接続文字列を指定する方法を簡易接続ネーミングという

07. XXXXX

08. XXXXX

09. XXXXX

10. XXXXX

○ Column:

chapter 05 【Oracleインスタンスの管理】

01. Oracleインスタンスの構造

Oracleインスタンス
- SGA(システムグローバル領域)
OSの共有メモリ内に確保される領域
複数のプロセスによって共有メモリされます
よく使用されるデータを共有領域に格納することで
複数のユーザーからのディスクアクセスを防ぎます

- バックグラウンドプロセス群
これらのメモリ構造とプロセス構造がどのように関連してSQLが処理されるかを学習する

1. Oracle Databaseのメモリー構造

●データベースバッファキャッシュ
データファイルに格納されている表のデータを処理する際に対象データを一時的に保持する領域です。
データファイルのデータはデータブロックと呼ばれる入れものに格納されている。
データベースバッファキャッシュへのデータのコピーもデータブロック単位で行われる

使用可能バッファ : データファイルと同じ内容を持つバッファのこと
使用済みバッファ : UPDATE文等を発行した際にデータファイルのデータがデータベースバッファキャッシュにコピーされ、更新される。この更新された、データファイルとは異なる内容を持つバッファのこと
使用中バッファ : サーバースタンプが処理するためにアクセスしているバッファのこと
※使用済みバッファと使用中バッファは他のデータブロックで上書きすることができない

●REDOログバッファ
INSERT文やUPDATE文などの実行によって発生するデータの変更を履歴情報として格納する領域
このような変更履歴は REDOエントリ と呼ばれ障害回復の為に保持するようになっています。

・REDOログバッファに対してREDOエントリが記録される。
・その情報がREDOログファイルに転記される
※REDOエントリとはDML文の実行などによってデータベース情報データが変更された際の変更履歴です。
REDOエントリの情報はリカバリ処理などで利用されます。
●共有プール
・実行されたSQLのテキスト (ライブラリキャッシュ)
・SQLの解析結果 (結果キャッシュ)
・データ・ディクショナリ (ディクショナリキャッシュ)
などの情報が書き込まれる領域です。

書き込まれた情報は他のユーザーと共有され次に同じSQLを実行した時に処理を高速化するために活用されます。

□MEMO: 共有プール内の領域
ライブラリキャッシュ : 共有プールの中で、SQLのテキストや解析情報が書き込まれた領域
ディクショナリキャッシュ : データディクショナリの情報が書き込まれた領域のこと
結果キャッシュ : SQLやPL/SQLファンクションの実行結果を格納する

●ラージプール (オプション領域。必要に応じて)
共有プールの補助的なメモリ領域で、RMAN(Recovery Manager)がバックアップをするときのデータや、共有サーバ構成にした時のユーザー情報などが格納されます。

●Java プール (オプション領域。必要に応じて)
Oracle JVM(Oracle Database上に用意されたJavaプログラムの実行環境)を使用してJavaプログラムを実行する際にJavaのコードやデータを格納するために利用されるメモリー領域。

●ストリームプール (オプション領域。必要に応じて)
Oracle Streamsが利用するメモリー領域です。

●PGA (プログラムグローバル領域)
PGA(プログラムグローバル領域)はバックグラウンドプロセスやサーバースタンプが処理に必要なデータを書き込むために、個別に持っているメモリー領域です。サーバースタンプが利用するPGAには、スタック情報の他に、ユーザーの処理に関連するデータを格納する領域があります。この領域のことを、UGA(ユーザーグローバル領域)と呼びます。

●重要●
PGAに格納される情報の1つに「ソートデータ」が含まれる。
ユーザーセッションでORDER BYなどのソート処理を実行した時にソート用のデータを書き込みます。
大量データのソート処理などでPGA上のソート領域が不足した場合はソートデータをディスクに書き込む処理が発生するのでパフォーマンスが低下します

2. Oracle Databaseのプロセス

●データベースライタープロセス(DBWn)
データベースバッファキャッシュ上にある、使用済みバッファをデータファイルに書き込むプロセスです。

「データベースライタープロセスが動作し、データファイルへ書き込む」処理は、ユーザーのCOMMIT文の実行などとは連動しません。ランダムアクセス方式で書き込まれる為、ある程度緩めに書き込みを行うことで効率を上げている。

書き込むタイミングは以下

- ・チェックポイントが発生した時
- ・サーバープロセスが一定のしきい値までバッファをスキャンしても使用可能バッファが見つからない時

●ログライタープロセス (LGWR)

REDOログバッファ内のREDOエントリをREDOログファイルへ書き込むプロセス
シケンシャルアクセス方式で書き込まれる為、比較的高速に処理することができます。

書き込むタイミングは以下

- ・ユーザーがCOMMIT文を実行した時
- ・3秒毎
- ・REDOログバッファに書き込んだREDOエントリが全体の3分の1まで達した時
- ・データベースライタープロセスが動作し、データファイルへの書き込みを行う直前

●システムモニタープロセス (SMON)

データベースの起動時にインスタンス障害を検知すると、クラッシュリカバリを実行します。そのほか使用されなくなった一時セグメントのクリーンアップなどを行います。

□MEMO：一時セグメントとは？

永続データ：通常ユーザーが管理する表データ
一時データ：一時的に書き込みを行い処理が終了したタイミングで自動的に削除されるデータ
一時セグメント：一時データを書き込むための領域
例)大量データに対するSELECTやORDER BYが発生した場合
PGAの領域では処理しきれずに、ディスク上の領域を使用して、ソート処理を行います。
このソート用の中間データは一時セグメントに書き込まれ、ソート処理が完了すれば自動的に削除されます。

●プロセスモニタープロセス (PMON)

ユーザープロセスがエラーになった場合に、データベース・サーバー上のリソースのクリーンアップをおこないます
エラーになったユーザーが実行していたトランザクションのロールバックを行ったり、ロックの解放を行ったりします。
また、リスナープロセスのい対して、データベースサービスに関する情報の登録や更新を行います。

●チェックポイントプロセス (CKPT)

Oracle Databaseサーバー内でチェックポイントイベントが発生した時に処理を行うプロセスです。

- ・データベースライタープロセスにシグナルを送り、データの書き込みを行わせる
- ・すべてのデータファイルのヘッダーと制御ファイルに対して、チェックポイントイベントが発生したタイミングを記録する

●アーカイバプロセス (ARCn)

ログスイッチが発生した時に、REDOログファイルのコピーを作成するプロセス
Oracle DatabaseがARCHIVELOGモードに設定されている時のみ起動されるオプションプロセス。

●サーバープロセス

ユーザープロセスからの要求に基づき、SQLの実行を行うプロセス
通常サーバープロセスは、ユーザーからの接続要求に基づき、そのユーザープロセス専属のプロセスとして起動されます。
このプロセスを専用サーバープロセスと呼びます。
一方大量ユーザーが接続した時の効率化の為、インスタンスを起動するタイミングでサーバープロセスを起動することも可能。
この時起動されたサーバープロセスを共有サーバープロセスと呼びます。

□MEMO：専用サーバー構成と共有サーバー構成

専用サーバー構成：専用サーバープロセスを使用して、ユーザーの処理を実行する構成
共有サーバー構成：共有サーバープロセスを起動し、複数のユーザーの処理を少数のサーバープロセスで実行する構成

02. SQL文を実行した時の内部処理

1. 検索処理を実行したときの処理の流れ

①ユーザーが命令したSELECT文がサーバープロセスに渡される

②サーバープロセスは共有プロセスにアクセスし、実行対象のSELECT文が格納されていないか確認

③同一のSELECT文が共有ブールに格納されていない場合サーバープロセスはSQLの解析処理を行う

- ・SELECT文に不備がないか構文のチェックを行う
- ・SELECT文で指定している表や列などが正しく存在しているか、データ・ディクショナリの情報と照らして確認
- ・同じくデータ・ディクショナリの情報からユーザーのアクセス権の情報を確認する

④サーバープロセスはオプティマイザの機能呼び出し、処理負荷が最も低い実行計画を探します。
実行計画とはSQLで支持された命令をサーバープロセスが処理する方法と順序を定義した計画(検索処理の際に索引の使用の有無などは実行計画に定義されます。)

上記の②③及び実行計画の作成を**SQLの解析処理**と読んでいます。

⑤ここまでのSELECT文の解析結果を共有ブールに格納します。

内容は、

- ・実行するSQL文
- ・実行計画
- ・解析済みコード

⑥実行計画にもとづき検索処理を実行します。
取得されるデータが識別される

⑦対象データを取りだし、ユーザープロセスに返す処理を行います。

- ・検索対象のデータがデータベースバッファキャッシュにキャッシュされているかどうかを確認
- ・見つからない場合、データファイルから該当するデータを含むデータブロックを探しデータベースバッファキャッシュにコピー
- ・データベースバッファキャッシュに用意されたデータはユーザープロセスに返されていきます。この処理をフェッチと言います

2. 更新処理を実行した時の処理の流れ

①～⑤検索時と同様なので割愛

①サーバープロセスは更新対象のデータがデータベースバッファキャッシュに存在するか確認
存在しない場合、検索処理と同様、データファイルから該当するデータを含むデータブロックをコピーしてきます。
このタイミングで、ロールバックの際に使用するUNDOデータを格納するUNDOセグメントのブロックも準備

②更新対象の行に対してロックをかける
ロックは行単位で行われる

③REDOログバッファに今回の変更内容をREDOエントリとして書き込み
REDOエントリには、DML文による変更の履歴が含まれます。

④更新前のデータをUNDOセグメントに書き込みます。
UNDOデータは主にトランザクションのロールバック処理や読み取り一貫性機能で使用されます。

⑤データベースバッファキャッシュ上の対象データの更新をおこないます。

⑥ユーザープロセスに処理の完了を通知します。

●重要●

この段階では、すべての処理がメモリー上にあるシステムグローバル領域上で行われていることに注目します。
つまりこの段階では、ディスクに何も書き込みをしていないため、電源断などによってインスタンスがダウンすると、
これらの更新情報はすべて失われます。ただしこの段階の変更内容はコミットが行われていない為失われても致命的ではありません。

3. COMMIT文を実行した時の処理の流れ

①REDOログバッファにコミットエントリを追加します

②ログライタープロセス(LGWR)がREDOログバッファ上に書き込まれているすべてのREDOエントリをREDOログ・ファイルへ書き出します。

③トランザクションによって確保されていたロックを開放します

④ユーザープロセスに処理の完了を通知します。

●重要●

COMMIT文の実行により、REDOログバッファに記録されているREDOログエントリをREDOログファイルにかきこみます。

この段階で変更内容が保証されることになる。

ただし、データベースバッファキャッシュ上で変更したデータ自体はまだデータファイルに書き込まれていないことに着目しておく

※注意※

ログライタープロセスによって書き込まれるトランザクションの情報は、

COMMIT文を実行したユーザーだけのものではありません。

ログライタープロセスはどのユーザーが実行したのかにかかわらず、REDOログバッファ上のすべての情報をREDOログファイルに書き出すこと理解しておく

4. データファイルへの書き込み処理とインスタンス障害

データベースバッファキャッシュ上で行われたデータに対する変更は、最終的にデータベースライタープロセスによってデータファイルへ書き込まれます

ただし、データベースライタープロセスはコミットとは異なるタイミングでデータファイルへの書き込みを行う為、コミットで確定した変更後のデータは即座にデータファイルに反映されるわけではありません。(必ずデータファイルへの書き込みが遅延します。)

そのため、データの更新は確定したけれどデータファイルは古いままという状態が発生します。

この状態で電源断が発生するとメモリー上にあるOracleインスタンスの情報が消えてしまい、データファイルには最新のデータが反映されないことになる。

この状態を**インスタンス障害**

といいます。

インスタンス障害はOracle Databaseサーバーの次回起動時に**システムモニタープロセス**が検知し、

クラッシュリカバリと呼ばれる、自動リカバリ処理を行います。

●重要●

Oracle Databaseサーバーでは、コミットのタイミングでデータファイルの更新を行わない為、データファイルのデータは常に最新というわけではない
その結果、インスタンス障害が発生する可能性があります。

03. Oracle インスタンスの起動と停止

1. Oracle Databaseサーバーの起動

STARTUPコマンドを実行

デフォルトで起動している際にコマンドを打つと

シャットダウンしてからにすると言われる。。。

起動時に行われる3つの処理

■1. Oracle インスタンスの起動

まずOracleインスタンスを起動します。

①パラメータファイルの読み込み

②インスタンスの起動

③診断ファイルのオープン

最初にパラメータファイルと呼ばれる設定ファイルを読み込みます。

SGAのサイズやバックグラウンドプロセスの動作を制御します。

結果、サーバーマシン上にOracleインスタンスが構築されます。

また、アラートログファイルと呼ばれる診断ファイルもオープンし書き込みを始めます。

この状態を **NOMOUNT状態** といいます

■2. データベースのマウント

④制御ファイルの読み込みとオープン対象のデータベースの認識(データベースのマウント)

マウントの処理では、制御ファイルをオープンし、制御ファイルの内容を読み取ります。

ここでオープンする制御ファイルの場所はパラメータファイルに記録されていることからNOMOUNT状態の時点で認識できるようになっています。

一方制御ファイルにはデータファイルとREDOログファイルの格納場所やステータス情報などが書き込まれています

したがって、Oracleインスタンスはこの段階で、この後オープンするデータファイルとREDOログファイルを認識する事になります。

このように制御ファイルをオープンし、OracleインスタンスがOracleデータベースを認識している状態を

MOUNT状態 といいます。

■3. データベースのオープン

最後のステップはデータベースのオープン

⑤REDOログファイルの読み込み

⑥(必要な場合)クラッシュリカバリ

⑦データファイルのオープン

データベースのマウント時に認識したデータファイルとREDOログファイルの格納場所の情報を用いて、REDOログファイルとデータファイルをオープン
します。

このREDOログファイルの内容に比べてデータファイルの情報が古い場合、インスタンス障害として検知し、システムモニタープロセスがクラッシュリ
カバリ処理を行います。

REDOログファイルとデータファイルが正常にオープンされた状態を OPEN状態 といいます。

OPEN状態は一般のユーザーが接続して様々なデータ処理を行うことができる状態(いわゆる「起動済み」の状態)です。

まとめ

◆SHUTDOWN

- ↓①パラメータファイルの読み込み
- ↓②インスタンス(SGA、バックグラウンドプロセス)の起動
- ↓③診断ファイル(アラートファイルなど)のオープン

◆NOMOUNT

- ↓④制御ファイルの読み込みとオープン対象のデータベースの認識(データベースのマウント)

◆MOUNT

- ↓⑤REDOログファイルの読み込み
- ↓⑥(必要な場合)クラッシュリカバリ
- ↓⑦データファイルのオープン

◆OPEN

□MEMO: マウント時のエラー

データベースのマウントの際は、制御ファイルを読み込み、データファイルやREDOログファイルの物理情報(格納場所)やステータス情報を取り込みますが、

データファイルやREDOログファイルへはアクセスしません。

したがってデータファイルやREDOログファイルが破損している場合は、マウント時には検知されず、次のオープン時に検知されてエラーが発生します。

2. SQL*Plusを用いたOracle Databaseサーバーの起動

書式: STARTUPコマンド

STARTUP [OPEN | MOUNT | NOMOUNT] [FORCE]

起動のどの段階まですすめるかをオプションで定義デフォルトはOPEN

稼働中のOracle Databaseサーバーに対して、FORCEオプションを付けて、STARTUPコマンドを実行すると強制的に再起動になる

※Oracle Databaseサーバーを起動するためには、SYSDBAまたはSYSOPERとして接続する必要があります!!!

3. SQL*Plusを用いたOracle Databaseサーバーの停止

書式: SHUTDOWNコマンド

SHUTDOWN [NORMAL | TRANSACTION | IMMEDIATE | ABORT]

SHUTDOWNコマンドもSYSDBAまたはSYSOPERとして接続する必要がある

4. Oracle Databaseサーバーの停止モード

4種類存在する

今すぐ停止するのか、接続中のユーザーの処理をまってから停止するのかなどの動作を選ぶことが可能

●NORMAL(標準)

デフォルトの停止モード

すべてのユーザーがログアウトするのをまってから停止操作に入ります。

停止の際には、チェックポイントイベントが発生します。

●TRANSACTION(トランザクション)

すべてのユーザーのトランザクションが終了するのを待つから停止操作に入ります。

トランザクションが終了した場合、ログアウトを待たずに強制的に切断を行います。

停止の際には、チェックポイントイベントが発生します。

●IMMEDIATE(即時)

すべてのユーザーを強制的に切断し、即座に停止操作に入ります。

実行中のトランザクションはロールバックされます。

停止の際には、チェックポイントイベントが発生します。

●ABORT(中断)

Oracleインスタンスはチェックポイントイベントを発生させず、強制的に終了します。

停止後の状態は「突然電源が切れた時」と同じであり、インスタンス障害が発生します。

したがって次回にインスタンスを起動する際はクラッシュリカバリが必要になります。

04. Oracleインスタンスの構成に使用されるパラメータの表示

1. 初期化パラメータとは

初期化パラメータは200種類以上

必要なもののみ設定。設定しないものはデフォルト値が入る

▼代表的なパラメータ

CONTROL_FILES : 制御ファイルの場所を指定する
MEMORY_TARGET : 自動メモリー管理を使用する際に、全体で使用するメモリーのサイズを指定する
DB_BLOCK_SIZE : データベース標準のデータブロックのサイズを指定する
DB_NAME : データベースの名前を指定する

2. パラメータファイル

Oracle Database 12cでは2種類のパラメータファイルを使用することができる。

パラメータファイルは\$ORACLE/dbsディレクトリに格納されている

●サーバーパラメータファイル(SPFIL)

Oracle Databaseサーバーが編集を行うバイナリ形式のファイルです。

管理者はEnterprise ManagerやSQLコマンドを使用して、Oracle Databaseサーバーに編集を命令します。

手動での編集はできません。

通常ファイル名は、 spfile<SID>.ora となります。

●テキスト初期化パラメータファイル

管理者がテキストエディタを用いて手動で編集するテキスト形式のファイル

Oracle Databaseサーバーは起動時にパラメータの読み込みだけを行い編集は行いません。

したがって、Oracleインスタンスの稼働中にパラメータを変更すると

Oracleインスタンスが認識しているパラメータは変更できません、

パラメータファイルへの反映は行いません。

このタイプのファイルは従来から使用されていましたが、特別な理由がない限り、前述のサーバーパラメータファイルの使用が推奨されています。

ファイル名は init<SID>.ora となります。

3. EM Expressによるパラメータの確認と編集

▼手順

①「構成」メニュー → 「初期化パラメータ」を選択

現行タブでは現在インスタンスに設定されているパラメータの値が表示されます

SPFILEタブでは、インスタンス起動時に読み込んだパラメータの値が表示されます。

②編集する場合は、対象パラメータを選択したまま「設定」をクリックし、編集ダイアログを表示します。

4. SQL*Plusによるパラメータの確認と編集

初期化パラメータはSQL*PlusからSHOW PARAMETERSコマンドを実行することで確認することができます。

▼書式：SHOW PARAMETERSコマンド

SHOW PARAMETERS キーワード

キーワードを指定すると、指定したキーワードを名前に含むパラメータが表示される

キーワードを省略すると、すべてのパラメータが表示される。

次にパラメータを編集するには、ALTER TABLE 文を使用します。

▼書式：ALTER SYSTEMによるパラメータの編集

ALTER SYSTEM SET パラメータ名=値 SCOPE=[SPFILE | MEMORY | BOTH]

SCOPE句には編集対象を指定する

BOTHを選択すると現在の設定値とSPFILEの両者を変更する

MEMORYは現行の値、SPFILEはSPFILEの値のみ編集する

05. Oracleインスタンスのメモリーコンポーネントの管理

これまで説明したとおりOracleサーバーが稼働するマシン上には

- ・SGA
- ・PGA

という2種類のメモリーコンポーネントが存在します。

これらのメモリーサイズはOracle Databaseサーバーの処理性能に影響を与える為、適切な値にする必要があります。

これらのメモリーを効率よく活用する為に、

- ・自動メモリー管理
- ・自動共有メモリー管理

が用意されています。管理者はこのどちらかの機能を利用してOracle Databaseサーバーを運用することができます。

1. 自動メモリー管理

自動メモリー管理 (AMM:Automatic Memory Management) では

データベース管理者はサーバーマシン上でSGAおよびPGAが使用可能なメモリーサイズの合計値を指定します。

Oracle Databaseサーバーはこれまでのメモリの使用状況などから

SGAとPGAの各コンポーネントに配分するメモリーサイズを自動調整します。

これらのメモリーサイズは随時見直されて調整されるのでメモリーの利用効率をあげることができます。

この機能はOracle Database 11gから使用できるようになりました。

自動メモリー管理では以下の2つのパラメータを用いて制御します。

▼自動メモリー管理に関連するパラメータ

MEMORY_TARGET : 自動メモリー管理で利用可能なメモリー全体のzサイズを指定する

MEMORY_MAX_TARGET : 自動メモリー管理機能で利用可能なメモリーサイズの上限を指定する

MEMORY_TARGETパラメータはこの上限値の範囲内で動的に増減させることができる。

2. 自動共有メモリー管理

自動共有メモリー管理 (ASMM:Automatic Shared Memory Management)

ユーザーは「SGAで使用するメモリーサイズ」「PGAで使用するメモリーサイズ」を個別に調整します。

この方法だとSGAを最適なサイズに保ちます。

REDOログバッファサイズは自動チューニングされないため、独自にサイズを設定する事になります。

なおこの機能はOracle Database 10g から使用可能です。

▼自動共有メモリー管理に関連するパラメータ

SGA_TARGET : 自動共有メモリー管理で利用可能なSGA全体のサイズを指定する

SGA_MAX_TARGET : 自動共有メモリー管理機能で利用可能なSGAサイズの上限を指定する。

SGA_TARGETパラメータはこの上限値の範囲内で動的に増減させることが可能

またPGAのサイズを指定する為に次のパラメータを使用する

▼自動共有メモリー管理に関連するパラメータ

PGA_AGGREGATE_TARGET : サーバマシン上で使用可能なPGAの全体サイズ。

このパラメータは手動メモリー管理でも使用する

3. 手動メモリー管理

ユーザーは個々のメモリーコンポーネントのサイズを個別に指定することもできます。

これは従来からのメモリー管理の方法です。

▼手動メモリー管理で使用するパラメータ

DB_CACHE_SIZE : データベースバッファキャッシュのサイズ

SHARED_POOL_SIZE : 共有プールのサイズ

JAVA_POOL_SIZE : Javaプールのサイズ

LARGE_POOL_SIZE : ラージプールのサイズ

STREAMS_POOL_SIZE : ストリームプールのサイズ

LOG_BUFFER : REDOログバッファのサイズ。REDOログバッファは自動共有メモリー管理の対象外であるため必ず設定する

～まとめ～

●データ更新を確定(コミット)したにもかかわらず、データファイルは古い情報を持っているため、計画外の電源断が発生するとインスタンス障害が発生する。

●SGAには、必須の領域としてデータベースバッファキャッシュ、REDOログバッファ、共有プールが存在する

- 必須のバックグラウンドプロセスには、DBWn、LGWR、SMON、PMON、CKPTがある
- NOMOUNT状態は、インスタンスが起動された直後の状態である
- MOUNT状態は制御ファイルがオープンされ、インスタンスとデータベースが関連付けられた状態である
- OPEN状態は、すべてのデータベースファイルがオープンされ一般のユーザーもアクセス可能な状態である
- Oracle Databaseサーバーの停止には、NORMAL、TRANSACTION、IMMEDIATE、ABORTという4つの停止モードがある
- 初期化パラメータを格納するファイルにはバイナリ形式のサーバーパラメータファイルとテキスト形式の初期化パラメータファイルの2種類が存在する
- サーバーパラメータファイルの編集はOracle Databaseサーバーが行いデータベース管理者は編集できない
- テキスト初期化パラメータファイルは管理者がテキストエディタを使用して編集を行う
- 自動メモリー管理機能を有効にする際は、MEMORY_TARGETパラメータを使用する

06. XXXXX

07. XXXXX

08. XXXXX

09. XXXXX

10. XXXXX

○ Column:

chapter 06 【データベース記憶構造の管理】

01. Oracle Databaseの構造

■1. 制御ファイル

REDOログファイルとデータファイルの物理構造(ファイルの格納場所)やステータス情報などを格納している

制御ファイルの場所は

CONTROL_FILES初期化パラメータで指定され、データベースをマウントするときにオープンされます。

Oracle Databaseは制御ファイルに書き込んだステータス情報を参照しながらデータベース全体の整合性を維持している。そのため、制御ファイルが壊れるとデータベース全体の整合性が維持できなくなり、Oracleインスタンスは異常終了します。そこで制御ファイルは多重化することが推奨されています。

しかし、制御ファイルを多重化した場合でも、そのうちの1つが壊れるとOracleインスタンスは停止してしまいます。

多重化の目的はOracleインスタンスの停止を防ぐことではなく、すべての制御ファイルが失われること(全損)を防ぐことです。

制御ファイルは障害からリカバリ処理に必要なファイルであるため、重要な障害予防策になります。

制御ファイルを多重化するときはディスクの破損に備えてそれぞれ異なるディスクに配置することが推奨されています。

■2. REDOログファイル

ユーザーが表データを更新

↓
SGAのREDOログに格納されます。
REDOログバッファに格納される変更履歴はREDOログエントリ、または

REDOレコードと言います。
ユーザーがトランザクションをコミットすると、
ログライター(LGWR)がREDOレコード(REDOエントリ)をREDOログファイルに書き込みます。

REDOログファイルに記録されたデータベースの変更履歴は、データベースに障害が発生した際に、データベースを復元する為に使用されます。

LGWRは現在書き込んでいるファイルがいっぱいになると書き込み先を別のファイルに切り替えます。
この切替処理を**ログスイッチ**と言います。
したがってOracle Databaseは**2つ以上のREDOログファイルが必要**になります。

トランザクションを**コミットするときにLGWRがREDOログファイルにREDOレコードを書き込みますが**、
このとき、データファイルへの書き込みは行われません。
つまりトランザクションをコミットしても、
変更履歴だけがファイルに書き込まれ、データベースバッファキャッシュ上で変更された実際のデータはデータファイルに書き込まれません。
このときインスタンス障害が発生すると、データベースバッファキャッシュ上にあった変更済みのデータが失われてしまいます。
インスタンス障害が発生してもREDOログファイルにREDOレコードが書き込まれている為、
コミット前の状態のデータファイルにREDOレコードを使用して、変更を再度行うことにより、データファイルを最新の状態にすることができます。

■REDOログ・ファイルのアーカイブ
REDOログ・ファイルが上書きされる前に、コピーを作成する。
「REDOログ・ファイルをアーカイブする」といい、作成されるファイルのことを「アーカイブログ・ファイル」と言います。
ARCnプロセスによってアーカイブされるREDOログ・ファイルのコピー
ArOnプロセスはログスイッチによって書き込み先が切り替わったタイミングで、アーカイブログ・ファイルを作成

■REDOログ・ファイルの多重化
グループ内に2つ以上のメンバーがある状態が「多重化されている状態」です。

現在書き込み中のREDOログファイルが壊れて書き込みができなくなると、
インスタンスは異常終了します。
REDOログファイルに書き込みができなくなると、障害回復に使用するはずの最新のREDOログファイルを損失することになり、復旧ができなくなってしまうからです。
そこで確定済みの変更データの損失を防ぐために、データベース管理者はREDOログファイルを多重化し、REDOログファイルを損失から保護します。
Oracle DatabaseはREDOログファイルを多重化する仕組みを持っているため、簡単にREDOログファイルを多重化して運用することが可能です。
▼REDOログファイルに関連する用語
ログスイッチ : LGWRは現在書き込み中のファイルがいっぱいになると次のファイルに書き込み先を切り替える。この切替のことをログスイッチという
REDOロググループ : REDOログ・ファイルを多重化すると、同じ内容を持ったREDOログファイルをセットすることができる。
この同一の内容を持ったログファイルのセットをREDOグループと呼ぶ。ログスイッチはREDOロググループの単位で行われる。
REDOログメンバー : REDOログファイルを多重化すると、1つのREDOロググループは同一の内容を持った複数のファイルを持つことになる。
このREDOロググループの構成要素(個々のファイル)のことをREDOログメンバーという
ログ順序番号 : Oracle Databaseは書き込み先のログを識別するために「ログ順序番号」と呼ばれるものを持っている。この番号はログスイッチが発生するたびに1つずつカウントアップされる。
REDOログファイルは、循環書き込みによって内容が上書きされて再利用されるが、上書き前のログと上書き後のログはログ順序番号によって一意に識別することができる。

■3. 表領域とデータファイル
データファイルは表や索引などのデータを格納するファイルです。
一般にOracle Databaseは複数のデータファイルから構成されており、これらのファイルはデータベース上では表領域と呼ばれる論理記憶単位にまとめられ管理されます。

■4. Oracle Databaseの論理記憶構造
データファイルはOSから物理的に認識できるファイルで、これを物理記憶構造と呼んでいます。
一方Oracle Databaseはデータの管理を行うために、表領域やセグメント、エクステントと言った論理記憶構造を使用します。

●表領域
表や索引などのデータはデータファイルに作成されます。しかし、ユーザーが表を作成するときには、表の作成先として、データファイルではなく、表領域と呼ばれる論理領域を指定します。
表領域：表や索引などを格納する論理的な領域で、データベース管理者が作成します。
表領域を作成する事によって、その表領域に関連するデータファイルが作成される事になります。
1つのデータファイルは必ず1つの表領域に関連付けられる。
表領域に関連付けられていないデータファイルや複数の表領域に関連付けられているデータファイルは存在しません。

その一方で、1つの表領域は、複数のデータファイルを関連付けることも可能です。

●データブロック
表領域の作成にともなって作成されたデータファイルの中は、データブロックと呼ばれる論理領域に区切られています。
このデータブロックは1/0の最小単位となります。
例えば、
データファイルからデータベースバッファキャッシュへデータを取り出すときには、
データブロックの単位でコピーが行われます。

COMEMO : データブロックのサイズDB_BLOCK_SIZE初期化パラメータで指定します。
これがデータベースの標準ブロックサイズになりますが、表領域を作成するときブロックサイズを指定することにより、
標準ブロックサイズ以外のブロックサイズの表領域を作成することも可能!!!!

●エクステント
連続したデータブロックのこと!!
ユーザーが表を作成すると、
Oracle Databaseは表領域の中で連続した複数のデータブロックを表の為に領域として確保します。
この時確保された複数の連続したデータブロックから構成される領域をエクステントと呼びます!!!!

●セグメント
今回作成した表は、現在表領域内に2つのエクステントを保持することになりました。この2つのエクステントは同じ1つの表を構成する領域なので、論理特性は同じになります。
このように同じ論理特性を持ったエクステントの集合をセグメントと呼んでいます。

セグメントには、表のデータを格納する表セグメントの他に、
索引セグメント、UNDOセグメントなどがあります。セグメントは1つ以上のエクステントで構成されますが、エクステントは隣接している必要がありません。
ただし、1つのセグメントを構成するすべてのエクステントは1つの表領域上に存在する必要があります。

▼Oracle Databaseの論理記憶構造

構造	特徴
表領域	表などのデータを格納する領域。1つ以上のデータファイルから構成される。 一方、1つのデータファイルは必ず1つの表領域に所属し、 複数の表領域に関連付けることは出来ない。

	また、1つの表領域には複数のセグメントを作成することができる。
セグメント	同じ記憶構造を持ったエクステントの集合。 1つのセグメントには1つ以上のエクステントが含まれる。 セグメントを構成するエクステントは隣接している必要がなく、同一表領域上であれば、異なるデータファイル上に分散配置されていても構わない。
エクステント	表を作成するときなどの領域確保の基本単位。 エクステントは1つ以上の連続したデータブロックで構成される必要があり、1つのエクステントが複数のデータファイルにまたがって存在することは出来ない。
データブロック	Oracle DatabaseにおけるI/Oの最小単位。 データの検索時など、メモリーへのロードはデータブロックの単位で行われる。 データブロックには複数の行データを格納することができる。

●重要●
大小関係の確認
表領域 > セグメント > エクステント > データブロック

■5. 事前構成済みデータベースの表領域
DBCA(Database Configuration Assistant)のシードテンプレートを使用して事前構成済みデータベースを作成した場合、予め幾つかの表領域が定義されています。

SYSTEM	データ・ディクショナリ表などのOracle Databaseの管理データが格納されている表領域。すべてのOracle Databaseに必須の表領域であり、他の名前で作成することは出来ない。また、SYSTEM表領域が破損すると、Oracle Databaseサーバーはダウンする
SYSAUX	SYSTEM表領域の補助的な表領域として構成され、Enterprise Managerの管理データなどが格納される。Oracle Database10 g以降、必須の表領域として作成されるようになった
TEMP	一時表領域というタイプの表領域で、ソート処理中の中間データなどのセッションの継続中にのみ存在する一時的なデータを格納する専用の表領域。名前は別の物を指定しても構わない
UNDOTBS1	UNDO表領域というタイプの表領域で、ロールバック用のデータを格納する専用の表領域。UNDOデータの管理を自動管理に設定すると、必須となる。名前は別の物を使用しても構わない。また、自動UNDOの管理機能で使用中のUNDO表領域が破損するとOracle Databaseサーバーはダウンする
USERS	ユーザーが作成した表などのデータを格納するための表領域。通常はこれ以外にも表領域を作成してデータを管理することが多い。
EXAMPLE	データベース作成時に、「サンプルスキーマの作成」を選択した場合に作成される表領域で、サンプルのデータが格納される。位置づけとしてはUSERSと同じく、ユーザーデータ用の表領域である。

▼Oracle Databaseを構成するファイル

- ◎制御ファイル : データファイル、REDOログファイルの場所と名前を含むファイル
- ◎データファイル : 表や索引などのデータベースのデータを格納するファイル
- ◎REDOログファイル : データに対する変更情報を記録するファイル
- ▼その他のファイル
- ◎初期化パラメータファイル : Oracleインスタンスの構成を設定するファイルであり必須
- △パスワードファイル : SYSDBAおよびSYSOPER権限を持つユーザーのユーザー名とパスワードを登録するファイル
- △アーカイブREDOログファイル : REDOログ・ファイルのコピーファイルでARCHIVELOGモードで運用した場合に作成されるファイル
- ※判例
- ◎必須
- △オプション

02. 表領域の管理

■1. データベースの拡張

- >方法
 - ・新しい表領域を作成する
 - ・既存の表領域にデータファイルを追加する
 - ・既存の表領域に関連付けられたデータファイルのサイズを拡張する
 - ・既存の表領域に関連付けられたデータファイルの動的な拡張を有効にする
- 上記はすべてEnterprise Managerで行うことができる。

■2. 表領域の作成

Enterprise Managerより実施
「記憶域」>「表領域」
「作成」をクリック

●一般

- 名前 : 表領域の名前を設定する
- 表領域タイプ : 「永続」表などのデータを格納する。「一時」や「UNDO」は既存の一時表領域やUNDO表領域を切り替える場合などに設定

デフォルトとして設定 : 表領域タイプが「永続」か「一時」の時設定できる。チェックを入れるとデータベースのデフォルト表領域になる
BIGFILE : 「SMALLFILE」複数の大きくないデータファイルで構成 「BIGFILE」単一で非常に大きいデータファイルを持つことができる
ステータス : 「オンライン」を選択する。オフラインの表領域はアクセス出来ない。表領域のメンテナンスの場合などにオフラインを設定する。

●データファイルの追加

Oracle ManagedFilesの使用 : OMF(Oracle Manage Files)を使用すると、Oracleがファイルを自動的に作成、命名および管理するのでチェックを入れると次のデータファイルの項目は表示されなくなる
データファイル : 作成するデータファイルのディレクトリとファイル名を指定する。OFAに準拠した名前にする場合は「表領域名」+連番2桁で拡張子を「dbf」とする
ファイルサイズ : 作成するファイルのサイズを指定する。表領域を1つのデータファイルで構成する場合は、このサイズが表領域のサイズになる。
既存ファイルの再利用 : 既存のデータファイルを再利用するときチェックを入れる
自動拡張 : チェックを入れると領域が必要になった時にデータファイルが自動的に拡張されるようにすることができる
増分 : 自動拡張設定がされたとき、一度の拡張で増加させるサイズを指定する。ここで1MBと指定すると、データファイルは1MB単位で拡張することになる
最大ファイルサイズ : 自動拡張による上限のサイズを指定する。「無制限」に設定すると、ディスクがいっぱいになるかOracle Databaseが管理可能なサイズの上限まで自動拡張を継続する

●領域

表領域のデータブロックのサイズとエクステント割当を設定します。
割り当てるエクステントのサイズをOracleに任せるか均一のサイズで割り当てるかを選択します。

～テキストでは省略～

- ロギング
- セグメント

■3. 表領域の変更

表領域のサイズは表領域に関連付けられたデータファイルのサイズの合計によって決定される為、以下の3つの方法で拡張可能
・既存の表領域にデータファイルを追加する
・既存の表領域に関連付けられたデータファイルのサイズを拡張する
・既存の表領域に関連付けられたデータファイルの動的拡張を有効にする

■4. 既存の表領域にデータファイルを追加する 「アクション」>「データファイルの追加」

■5. 既存の表領域に関連付けられたデータファイルのサイズを拡張する

表領域メニューから、表領域を展開し、
データファイルを選択、
「アクション」>「サイズの変更」
「アクション」>「自動拡張の編集」
それぞれより変更を実施する

■6. 表領域の削除

表領域とその表領域に含まれるセグメントが不要になったら、表領域をデータベースから削除することができる。

表領域削除には DROP TABLESPACE 権限が必要

表領域メニューより表領域を選択

「アクション」>「削除」

表領域削除ウィザードが表示される

- コンテンツの削除 : 表領域に格納されているオブジェクトを削除するか
- データファイルの削除 : 表領域に属するデータファイルを削除するか
- 制約の削除 : 削除する表領域の表を参照している外部キー制約を削除するか

セグメントを含む表領域は削除することができない。「コンテンツの削除」にチェックをいれると、削除する表領域に格納されているセグメントを削除し、表領域を削除します。

削除する表領域が参照している外部キー制約がある場合も、表領域を削除しようとするエラーになります。削除する表領域の表を参照する外部キー制約をあらかじめ削除しておくか、「制約の削除」にチェックを入れ表領域を削除する。

表領域に属するデータファイルも削除することができる。「データファイルの削除」にチェックを入れないと、データファイルはそのまま残ります。後からデータファイルを削除する場合はオペレーティングシステムのコマンドを実行して削除する必要があります。

Oracle Managed Files (OMF)を使用している場合は、表領域を削除すると、データファイルも削除されます。

03. UNDOデータの管理

■1. UNDOデータの概要

UNDOデータ : 実行中のトランザクションに対して処理を取り消すロールバック用のデータ

ユーザーがDML文を実行し処理が開始されると、トランザクションで使用するUNDOデータはUNDO表領域上にあるUNDOセグメントにかきこまれます。

その後同一のトランザクションで発生したUNDOデータはすべて同じUNDOセグメント上に書き込まれます。

一方1つのUNDOセグメント上には、複数のトランザクションのUNDOデータを格納することができます。

したがって大量のトランザクションが同時に実行された場合でもトランザクションの数より少ない数のUNDOセグメントで処理を実行することができます。

■2. UNDOデータの役割

主にトランザクションのロールバックを目的に生成されますが、他にも

読み取り一貫性機能やフラッシュバック機能を実行する際にも利用されます。

●トランザクションのロールバック

ロールバックを行うと、UNDOセグメント上に保持していたUNDOデータ(変更前の情報)を取り出し、元のデータブロック上のデータを上書きしてデータ

を変更前の状態に戻します。

終了したトランザクションは二度とロールバックすることができないので、終了したトランザクションで生成されたUNDOデータは、そのトランザクションにとっては不要になります。

●読み取り一貫性機能

「ユーザーが検索処理を実行した際に、確定済みのデータだけを検索対象とする」機能のこと

Oracle Databaseではこの読み取り一貫性機能を実現する為にUNDOデータを使用します。

●フラッシュバック機能

「過去のある時点に遡る」機能のこと

過去のある時点のデータを検索したい場合

```
SELECT *  
FROM PRODUCTS  
AS OF TIMESTAMP (SYSTIMESTAMP - INTERVAL '30' MINUTE);
```

フラッシュバック問い合わせ機能では、検索対象のデータが現在までに更新されていることを検知するとUNDOデータにアクセスして更新前の情報を取り出し、30分前の情報を生成してくれます。

■3. UNDOデータとREDOデータの比較

UNDOデータ : 一度実行した処理を取り消す為に使用します。トランザクションのロールバックで使用

REDOデータ : 一度変更を確定したデータが障害によって失われた際に、同じ処理をもう一度繰り返す為に使用します。障害からのリカバリで使用

▼UNDOデータとREDOデータの比較

	UNDO	REDO

レコード内容	変更を元に戻すために利用	変更を再現するために利用
使用目的	ロールバック、読み取り一貫性およびフラッシュバック	リカバリ(データベースの変更をロールフォワードするため)
格納場所	UNDOセグメント	REDOログファイル
保護の対象	マルチユーザーシステムでの一貫性のない読み取り	データ損失
アーカイブ	不要	リカバリの為に必要

■4. UNDOの管理

Oracleには自動UNDO管理と呼ばれる完全に自動化されたメカニズムが用意されており、Oracle Database 12cではこれがデフォルトモードです。

●UNDO表領域の作成と管理

UNDO表領域は、Database Configuration Assistantを使用してデータベースを作成すると、自動的に作成され、UNDOTBS1という名前の自動拡張可能なUNDO表領域になります。

1つのデータベースに複数のUNDO表領域を作成することができますが、現行のUNDO表領域として指定できるのは1つのみです。

データベースは最初に使用可能になったUNDO表領域を自動的に選択します。

特定のUNDO表領域を指定するには、UNDO_TABLESPACE初期化パラメータにUNDO表領域名を指定します。

データベース管理者は

- ・UNDO表領域のサイズ

- ・UNDO保存期間

を適切に設定することが必要。

設定によっては「領域エラー」や「スナップショットによるエラー」が発生しやすくなります。

「ORA-1555: スナップショットが古すぎます」

UNDO保存期間はUNDO_RETENTION初期化パラメータで設定します。

UNDO_RETENTIONパラメータを予想される最長のフラッシュバック操作期間と同じ値に設定します。

この保存期間を過ぎると期限切れの状態になり、上書き対象になります。

■5. UNDOの保存期間の自動チューニング

UNDO保存期間を自動チューニングしてくれる機能がある。

●UNDO表領域がAUTOEXTENDオプションで構成されている場合

UNDO_RETENTIONパラメータを予想される最長のフラッシュバック操作期間と同じ値に設定する

UNDO表領域に空き領域が少なくなると、表領域が自動的に拡張される

●UNDO表領域が固定サイズの場合

UNDO_RETENTIONパラメータは無視されます。

Oracle Databaseサーバーが表領域サイズと現行システムの負荷に対して最適な保存期間になるように、

UNDO保存期間を動的にチューニングします。

このサイズが小さいと、次の2つのエラーが発生する可能性があります。

- ・DMLが失敗 : 新しいトランザクションに対するUNDOを格納するための十分な領域が無いため

- ・スナップショットのエラーにより、長時間実行問い合わせが失敗 : UNDOデータが不足している為に、読み取り一貫性機能を維持できない。

■6. UNDO情報の表示とUNDOアドバイザの使用

Enterprise ManagerでUNDOの構成を確認できる

「記憶域」>「UNDOの管理」

「UNDO管理詳細」ページが表示されます。

UNDOサマリーの表領域でUNDO表領域の

名前

サイズ

自動拡張可能かどうかを確認できます

「UNDOアドバイザ」の機能を使用して、UNDO表領域を固定サイズにする必要がある場合は、必要な最小サイズを簡単に判定できます。

過去のトランザクション情報を元に必要なUNDO表領域のサイズをシミュレートし、推奨されるUNDO表領域のサイズをアドバイスしてくれる。

□MEMO : UNDOセグメント

- ・UNDOセグメントはUNDO表領域に作成される

- ・UNDOセグメントはロールバックするために必要なUNDOデータを格納している

- ・1つのトランザクションに対して1つのUNDOセグメントが割り当てられる

- ・1つのUNDOセグメントに対して複数のトランザクションが割り当てられる

～まとめ～

●Oracle Databaseを構成するファイルは制御ファイル、REDOログファイル、データファイル

●制御ファイルには、データベースの物理構造やステータス情報が格納され、マウント時に読み込まれる

●制御ファイルは、損失を防ぐために、多重化することが推奨される

●REDOログファイルには、リカバリのための変更履歴(REDOレコード)が格納されている

●REDOログ・ファイルは循環利用される為、2つ以上のREDOロググループが必要

●REDOログ・ファイルも損失を防ぐために多重化することが推奨される

●データファイルには、表や索引などのデータが格納される

●1つの表領域には、1つ以上のデータファイルが関連付けられる

●Oracle Databaseは、表領域、セグメント、エクステント、データブロックと言った論理記憶構造を使用して、データの管理を行っている。

●データブロックはI/Oの最小単位である

●エクステントは1つ以上の連続したデータブロックで構成され、表などの領域はエクステントの単位で割り当てられる

●セグメントは1つのオブジェクトを構成するエクステントの集まりのことである

●事前構成済みデータベースは、SYSTEM秤量記など6つの表領域で構成される

●SYSTEM表領域には、データ・ディクショナリなどのOracle Databaseの管理データが格納される

●データベースを拡張する方法としては、

- ・新規表領域の作成

- ・表領域へのデータファイルの追加

- ・データファイルサイズの拡張

がある

- UNDOデータはトランザクションのロールバック、読み取り一貫性機能、フラッシュバック機能で使用される
- デフォルトで自動UNDO管理が有効になっている
- ユーザーはUNDO表領域を適切なサイズで作成し、UNDO保存期間を設定する(細かい管理作業は不要)

04. XXXXX

05. XXXXX

06. XXXXX

07. XXXXX

08. XXXXX

09. XXXXX

10. XXXXX

○ Column:

chapter 07 【ユーザーの管理とセキュリティの管理】

01. ユーザーアカウントの作成

ユーザーアカウントを作成する際はユーザー属性をユーザーアカウントに設定します。

■ユーザー属性

- 名前 : ユーザー名
- データベース認証パスワード : パスワード。データ・ディクショナリに暗号化されて入る。大文字小文字区別される
- デフォルト表領域 : オブジェクトの格納先を指定しなかった場合に使用される表領域。
ユーザーアカウントに「オブジェクトを作成する権限」「表領域の割当制限」が設定されてないとオブジェクトに対して表領域を割当できない。
- 一時表領域 : ユーザーアカウントに対してデフォルト表領域を指定しない場合は、デフォルト永続表領域が暗黙で指定される
- 一時表領域を指定しない倍は : ユーザーが一時セグメントの作成を必要とするSQL文を実行した場合に使用される表領域。ユーザーアカウントに対して
- 表領域の割当制限 : ユーザーに対して使用を許可する表領域の容量。ユーザーが使用可能な表領域ごとに設定する。デフォルトではすべての表領域に対して0が設定されており、表領域の使用は許可されていない
- アカウントステータス : ただし一時表領域やUNDO表領域は割当制限の設定がなくても使用できる
- パスワードステータス : ログインできる状態(ロック解除)。ログインできない状態(ロック)を指定する
- パスワードステータス : パスワードを無効(期限切れ)にするか、有効にするかを指定する

パスワードが無効になった場合ユーザーは次回のログイン時にパスワードを変更する必要がある。

新しく作成したユーザーアカウントを使用してデータベースに接続するにはそのユーザーアカウントに CREATE SESSIONシステム権限 が付与されている必要があります。

■EMExpressからの登録
■SQL文での登録
CREATE USER user_name
IDENTIFIED BY password
[DEFAULT TABLESPACE space_name]
[TEMPORARY TABLESPACE space_name]
[QUOTA size ON space_name]
[PROFILE profile_name];

■事前定義されたユーザーアカウント
SYS : データベース管理者アカウント
SYSTEM : 管理情報を表示する表やビュー、Oracleデータベースのオプションやツールで使用する内部的な表やビューを所有する
SYSMAN : EnterpriseManagerによるデータベース操作を実行する為のユーザーアカウント
DBSMNP : EnterpriseManagerの管理エージェントがデータベースの監視及び管理を行う為に使用するユーザーアカウント

02. ユーザー属性の変更

■EMExpressからの変更

■SQL文を使用したユーザー属性の変更
ALTER USER username
[IDENTIFIED BY password]
[DEFAULT TABLESPACE tablespace_name]
[TEMPORARY TABLESPACE tablespace_name]
[QUOTA size ON tablespace_name]
[PROFILE profile];

03. ユーザーアカウントの削除

■EMExpressを使用した削除

※削除が失敗する例
・データベースに接続中
・削除しようとしているユーザーがオブジェクトを所有している場合
削除前に所有するすべてのオブジェクトを削除するか「カスケード」を指定する必要があります。

■SQL文を使用した削除
DROP USER test;
DROP USER test2 CASCADE;

04. ユーザーアカウントのロック・ロック解除

■EMExpressを使用したロック及びロック解除

■SQL文を使用したロック及びロック解除
ALTER USER ユーザー名 ACCOUNT LOCK;
ALTER USER ユーザー名 ACCOUNT UNLOCK;

※重要

- ・ユーザーアカウントをロックして特定のユーザーアカウントによるデータベースへのアクセスを一時的に拒否することができる
- ・ロックされているユーザーアカウントで接続したい場合は管理者がロックを解除する必要がある

05. パスワードポリシー

06. XXXXX

07. XXXXX

08. XXXXX

09. XXXXX

10. XXXXX

○ Column:

--	--

01. スキーマとスキーマオブジェクト

■命名規則

- ・長さは30バイト以下
- ・先頭の文字は数字以外の文字
- ・使用できる文字は英数字、漢字、カタカナ、ひらがな(日本語環境の場合)
- ・使用できる記号は「_」「\$」「#」の3種類のみ
- ・同一のスキーマ内で重複する名前は定義できない
- ・OracleのSQLにおける予約語は使用できない
- ・アルファベットの大文字小文字は区別されない

■ SQL文を使用した表の作成

表を作成するには「CREATE TABLE権限」が必要
他のユーザーのスキーマに表を作成するには「CREATE ANY TABLE権限」が必要

```
CREATE TABLE emp_copy AS SELECT * FROM emp1;
```

CREATE TABLE AS SELECT文(CTAS)では、
NOT NULL制約以外の制約はコピーされない

※Column: 表領域の割当制限
作成した表にデータを格納するためには
「表を作成するためのシステム権限(CREATE TABLE権限)」
「エクステンツを獲得するための表領域の割当制限」の両方が必要

■ SQL文を使用した表の変更

※重要

- ・SQL文を使用して既存の表を変更するにはALTER TABLE文を使用する
- ・列及び制約を追加するにはALTER TABLE ADD文を使用する
- ・制約は列レベルまたは表レベルで定義できる
- ・列のデータの型を変更するにはALTER TABLE MODIFY文を使用する
- ・NOT NULL制約以外の制約を追加するにはALTER TABLE ADD文を使用する

■ 表の削除

表を削除すると、表内のすべてのデータに加えて、表に定義されている制約や索引、及びデータ・ディクショナリに格納されている表の定義情報も削除されます。

一方、表を参照するビューやシノニムは削除されずに無効になります。
表を削除できるのは「DROP ANY TABLE権限」を持つユーザーのみ

※重要

- 表を削除すると次のような状態になる
 - 表の定義がデータ・ディクショナリから削除される
 - 表のすべての行が削除される
 - 表に定義されているすべての索引が削除される
 - 表を削除すると既定でゴミ箱に入る(ゴミ箱からリカバリ可能)
- SQL文を使用して既存の表を削除するには、DROP TABLE文を使用する
- 表をゴミ箱にいれないで完全に削除するにはDROP TABLE文でPURGE句を指定する

02. スキーマオブジェクトの追加作成

索引は検索時野ディスクI/Oを減らし、表データの検索を高速にするために作成するオブジェクトです。

- ・索引は、索引セグメントとして記憶域を使用するオブジェクトです
- ・索引を作成する際は表領域を指定できる
- ・索引は、表に関連付けられる
- ・索引は、表内の1つ以上の表に対して作成できる
- ・表に一意性役や主キー制約を定義すると、その列には索引が暗黙的に作成される
- ・索引は表のデータがDML文によって挿入、更新、削除されると、それにとまって自動的にメンテナンスされる

■ CREATE INDEX文

索引を作成することができる

```
CREATE INDEX 索引名 ON 表名(列名);
```

※COLUMN: 索引のタイプ

「標準-Bツリー」

「Bツリー(バランストリー)」というツリー構造になっています。

ルートブロック→ブランチブロック→リーフブロックへとめぐって行きリーフブロックから実際の表のデータにアクセスします

「ビットマップ」

主にデータウェアハウス環境で使用します。

■ DROP INDEX文

SQL文を使用して索引を削除する

```
DROP INDEX 索引名;
```

■ ビュー

「仮想の表」

■ ビューを使用するメリット

複雑なSELECT文を簡略化できる
セキュリティ的な機能を実現する

■ ストアドプログラムの管理

「ストアドプログラム」: データベースに格納されたプログラムのこと
または「データベース常駐型プログラムユニット」

PL/SQLまたはJavaで記述されたプログラムであり、
アプリケーション開発者がOracleデータベース内に作成するスキーマオブジェクトです。
これもスキーマオブジェクトらしい。。

▼作成できるプログラムコードタイプ

タイプ	説明
パッケージ仕様部	型、変数、定数、例外、カーソル、及びパッケージの外から参照できるサブプログラムを宣言する
パッケージ本体	パッケージ本体部。カーソルの問合せとパッケージに定義されたプロシージャとファンクションなどのサブプログラムのPL/SQLコードを定義する
プロシージャ	単一のタスクを実行するプログラム
ファンクション	単一のタスクを実行するプログラム。呼び出し元のプログラムに値を返す必要がある。
トリガー	表、ビュー、イベントに関連付けられたプログラム

作成できるプログラムコードのタイプには上記5種類がある！

03. 表へのデータのロード

SQL*Loaderと呼ばれるユーティリティを使用して外部ファイルに保存されている大量のデータをバッチ処理でロードすることが出来ます。

「データファイル」.csv .xlsx
「制御ファイル」.ctl

■特徴

- ・sqlldrコマンドで起動するコマンドラインインターフェースである
- ・外部ファイルからデータベースヘデータをバルクロード(大量データの一括ロード)する
- ・区切り文字、固定レコード、可変レコード及びストリームなど、多数の入力形式をサポートしている
- ・同時に複数の表をロードできる
- ・強力なデータのフィルター機能がある

■データのロード方法

- ・従来型パス : INSERT文が生成されてデータが表に挿入される
- ・ダイレクトパス : ダイレクトパスAPIを使用してデータをサーバーのロードエンジンに渡す。ロードエンジンは渡されたデータから列配列構造体を作成する
- ・外部表 :

04. XXXXX

05. XXXXX

06. XXXXX

07. XXXXX

08. XXXXX

09. XXXXX

10. XXXXX

○ Column:

chapter 09 【バックアップ及びリカバリの概要】

01. バックアップ・リストア・リカバリ

■バックアップ

「データを再構築するために使用できるデータベースのデータのコピー」です。
物理バックアップ : 物理ファイル群をコピーすること
論理バックアップ : OracleDataPumpを使用して表やストアードプロシージャなどの定義、データ等の論理データをバイナリファイルにエクスポートすること。後でそのバイナリファイルをデータベースにインポートすることでリストアする

■ Recovery Managerによる物理バックアップ
「Recovery Manager (RMAN)」と呼ばれる機能を使用して、
次に上げる「データベースを構成する物理ファイル群」をディスクまたはテープにコピーします

- ・制御ファイル
- ・アーカイブREDOログ・ファイル(オンラインREDOログファイルのコピー)
- ・データファイル
- ・(サーバーパラメータファイル)

■ データベースのリストアとは
ファイルが破損または削除された場合にバックアップ媒体にバックアップしているファイルのコピーを破損してしまったファイルの代わりに元の場所または新しい場所にコピーする作業です

■ データベースのリカバリとは
リストアしたデータファイルにREDOログ・ファイルを適用することで、データファイルに反映されていない変更情報を反映していく作業です。

□ Column: SCN(システム変更番号)
Oracleデータベースではトランザクション内で行われる変更はSCN(システム変更番号)という「内部的なタイムスタンプ」がつけられる

■ バックアップのタイプ
・一貫性バックアップ
・非一貫性バックアップ

■ Oracleデータベースのリカバリ機能
・インスタンスリカバリ
・メディアリカバリ

・フラッシュバック機能

■ バックアップ及びリカバリを自動管理するためのデータベース構成

02. XXXXX

03. リカバリの実行

■ Oracle社推奨のリカバリ
・Oracle社推奨のリカバリ方法は「データリカバリアドバイザ」を使用したリカバリ方法
・データ障害が検出されると、その情報はADR(自動診断リポトリ)に記録される
・データリカバリアドバイザはADRの情報を使用して、修復アドバイスを生成し、障害の修復を実行する
・修復アドバイスの生成と障害の修復を実行するには、RMANで次の手順を実行する
①LIST FAILURE
②ADVISE FAILURE
③REPAIR FAILURE

■ Tips

データ障害の自動検出は「状態チェック」という機能によって実行されています
状態チェックはOracle Database12c に搭載されている状態モニターによって実行される診断プロセスです。
状態チェックはエラーの発生に応じて起動しますが、手動で起動することも出来ます。

04. XXXXX

05. XXXXX

06. XXXXX

07. XXXXX

08. XXXXX

09. XXXXX

10. XXXXX

○ Column:

01. Oracle自己監視アーキテクチャ

■Oracleの自己診断エンジン

ADDM (自動データベース診断モニター)

定期的にデータベースのパフォーマンスの状態を自動診断し、その情報をデータベースに格納しています。診断は予め設定されている「メトリック」に基づいて行われ、診断結果に問題がある場合は、データベース自身がその問題箇所を特定できる仕組みになっています。

・データベースの状態及びワークロードの監視

EM Expressを使用すると、データベースの状態やワークロードを監視できます。

02. パフォーマンスの問題の診断

■AWR (自動ワークロードリポジトリ)

03. リカバリの実行

04. XXXXX

05. XXXXX

06. XXXXX

07. XXXXX

08. XXXXX

09. XXXXX

10. XXXXX

○ Column: