

第6章 データベース記憶域構造の管理

6-1 データベースファイル

- Oracleが読み書きするファイル群の説明
 - データベースファイル
 - データファイル、REDOログファイル、制御ファイル
 - その他
 - 初期化パラメータファイル、パスワードファイル、アーカイブログファイル（アーカイブREDOログファイル）

ファイル	説明
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 (オンラインREDOログ ファイル)	データベースに対して行われたすべての更新情報が記録される
制御ファイル	データファイル名、REDOログファイル名を含むデータベースの構成情報および管理情報が格納
初期化パラメータファイル	初期化パラメータのリストおよび各パラメータの値が格納
パスワードファイル	SYSDBAなどの管理権限を持つユーザーのパスワードを管理
アーカイブログファイル (アーカイブREDOログ ファイル)	REDOログファイルのコピー

6-1-2 データファイル

- 表や索引が格納されるファイル
- データファイルの構成（ファイル名やファイル番号）は、**制御ファイル**で管理している
- データの保管の仕組みは、データファイルを論理的にグループ化した概念である**表領域**が関わる

6-1-3 REDOログファイル

- データベースの更新履歴（**REDOデータ**）を記録するファイル
 - データベースに加えたすべての更新内容が記録されており、障害から回復するために使用
 - インスタンスが異常終了した場合の障害復旧（インスタンスリカバリ）
 - データファイルが破損した場合の障害復旧（メディアリカバリ）
 - REDOログファイルに記録された更新履歴をもとに、更新処理を再実行することを**ロールフォワード**

- システム・グローバル領域(SGA)のREDOログ・バッファに格納されているREDOエントリは、LGWRプロセスによりREDOログ・ファイルへ書き込まれます。
 - LGWRプロセスがREDOログ・ファイルへ書き込みを行うタイミングは次のとおりです。
 - トランザクションがコミットされた時
 - ログ・スイッチが発生した時
 - 前回の書き込みから3秒経過した時
 - REDOログ・バッファが1/3になった時
 - REDOログ・バッファのREDOエントリの合計が1MB以上になった時
 - DBWnが使用済みバッファをデータファイルへ書き込む必要がある時

インスタンスリカバリ

- インスタンス異常終了により、データベースファイルは一貫性のない**不整合**な状態になる
- 更新されたブロックはデータファイルに**遅延書き込み**されるため、更新がデータファイルに未反映な状態になる
- REDOログファイルからデータベースの更新履歴を読みだし、整合性の回復を行う＝**インスタンスリカバリ**

REDOログファイルの多重化とログスイッチ

- REDOログファイルの破損によって、更新履歴が失われるため、**多重化**をしている
 - 多重化構成におけるREDOログファイルは**メンバー**と呼ばれる
 - 多重化されたメンバーの組を**グループ**とよぶ（同グループ内のメンバーは同じサイズ）
- データベースに対して更新処理が実行されると、インスタンスは、**カレント**のグループ（ロググループ）内のすべてのメンバーに実行された更新に対応するREDOデータを書き込む
 - カレントのメンバーに空き容量がなくなると、別のロググループに**ログスイッチ**する
 - つまり、一つ目のロググループに上書きをすることになる
- 多重化とログスイッチによってローテーション形式での管理が実施される

REDOログファイルの構成

- Oracleデータベースには、2つ以上のグループが必要
- 1つのグループには1つ以上のメンバが必要
 - 多重化する場合には、1つのグループに2つ以上のメンバーを構成
- REDOログファイルの構成は、制御ファイルで管理される
- REDOログファイルのサイズは自動的に拡張されない

6-1-4 制御ファイル

- データベースの構成（物理構造）情報および管理情報を記録するファイル
 - データファイル名
 - REDOログファイル名などの構成情報
 - REDOログ・ファイルのログ順序番号
 - データベースの作成タイムスタンプ
 - チェックポイント情報
- 制御ファイルは多重化が推奨。DBCAでデータベースを作成するとデフォルトで多重化される
- バイナリ形式である
- 制御ファイルのファイル名は、**CONTROL_FILES**初期化パラメータに設定

- ※制御ファイルを多重化していない場合は、既存の制御ファイルをコピーし、新たに作成された制御ファイル名をCONTROL_FILESに追記することで制御ファイルを多重化できる

6-2 表領域

6-2-1 表領域とオブジェクトに記憶域を割り当てる仕組み

- 表や索引はデータファイルに格納されるが、表や索引を作成するとき、格納先に指定するものはデータファイルではない
- データファイルを論理的にグループ化した概念である**表領域**を指定する
 - 表領域は1つ以上のデータファイルをグループ化したもの（smallfile表領域の場合）
 - オブジェクト（表や索引など）の格納先は、表領域を指定する
 - オブジェクトの記憶域は、そのオブジェクトの格納先に指定された表領域を構成するデータファイルから割り当てられる

オブジェクトに記憶域を割り当てる仕組み

- 表や索引などのデータを保持するオブジェクトには**セグメント**と呼ばれる記憶域が1対1で対応する
 - セグメントは、表や索引などの作成時に格納先として指定した表領域内に格納
 - すべてのオブジェクトにセグメントが対応するわけではない（ビューはデータがないからセグメントなし）
 - セグメントは表領域に作成されるが、**表領域の実体はデータファイル**なので、セグメントの記憶域は実際にはデータファイルから割り当てられる
- まとめ
 - データファイルは、固定サイズ（2kb~32kb）に分割して使用=**データブロック**。あるいは**ブロック**。
 - データブロックの中にオブジェクトのデータを保管。
 - データブロックのサイズは小さいため、データブロックを1つずつセグメントに割り当てるのは非効率なため、**エクステンツ**としてまとめて、このエクステンツをセグメントに割り当てる
 - セグメントの空き容量が不足した場合は、セグメントにエクステンツを追加することで、空き容量を追加できる。

6-2-2 特殊な役割の表領域

- Oracle Database作成時に自動的に作成される、特殊な表領域
 - これらはオフライン化できない

表領域	説明
SYSTEM表領域	<div><ul style="list-style-type: none">• Oracleに必須の表領域• データベースの重要な管理情報（データディクショナリ）を格納• 表領域名は「SYSTEM」固定で変更できない• 常にオンライン必須で、破損したらインスタンスの異常終了になる</div>
SYSAUX表領域	<div><ul style="list-style-type: none">• Oracleに必須の表領域• SYSTEM表領域の補助的な役割であり、データベースの管理情報を格納• 表領域名は「SYSAUX」固定で変更できない「AWR：自動ワークロードリポジトリ」も格納</div>

表領域	説明
UNDO表領域	<ul style="list-style-type: none"> ・ UNDOセグメントという特殊なセグメントを格納するための表領域 ・ 表や索引などの通常のオブジェクトのセグメントを格納できない ・ 表領域名は任意の名前に変更可能。DBCAでデータベースを作成した場合、「UNDOTBS1」という表領域名となる
一時表領域	<ul style="list-style-type: none"> ・ 一時セグメントという特殊なセグメントを格納するための表領域 ・ 表や索引などの通常のオブジェクトのセグメントを格納できない ・ 表領域名は任意の名前に変更可能。DBCAでデータベースを作成した場合、「TEMP」という表領域名となる
	<ul style="list-style-type: none"> ・ 事前構成済みデータベースの作成で自動的に作成される表領域 <ul style="list-style-type: none"> ◦ SYSTEM ◦ SYSAUX ◦ TEMP ◦ UNDOTBS1 ◦ USERS ◦ EXAMPLE

6-3 表領域の作成・拡張・削除

6-3-1 表領域の作成

CREATE TABLESPACEコマンド

- ・ ここで作成される表領域は**永続表領域**⇔UNDO表領域

100Mのデータファイルを複数持つ表領域を作成する(自動拡張なし、制限付きあり、無制限)

```
CREATE TABLESPACE 表領域名
DATAFILE 'データファイル名1.dbf' SIZE 100M
        , 'データファイル名2.dbf' SIZE 100M AUTOEXTEND ON NEXT 500K MAXSIZE 1024M
        , 'データファイル名3.dbf' SIZE 100M AUTOEXTEND ON NEXT 500K MAXSIZE UNLIMITED
;
```

- ・ SEGMENT SPACE MANAGEMENT句で表領域のセグメント(1つのデータベース・オブジェクトを構成する複数のエクステンツの集合)領域の管理方法を指定できる
 - SEGMENT SPACE MANAGEMENT
 - セグメント領域の管理方法です。ローカル管理の永続表領域(SYSTEM表領域以外)に指定でき、デフォルトはAUTOです。
 - AUTO: 「自動セグメント領域管理」は、ビットマップを使用して表領域のセグメント内の空き領域を管理します。
 - MANUAL: 「手動セグメント領域管理」は、空きリストを使用して表領域のセグメント内の空き領域を管理します。

6-3-2 表領域の拡張

ALTER DATABASE DATAFILE RESIZEコマンド

表領域の拡張（データファイルのリサイズ）

```
ALTER DATABASE DATAFILE {データファイルのパス} RESIZE {サイズ};
```

ALTER TABLESPACE ADD DATAFILEコマンド

表領域の拡張（データファイルの追加）

```
ALTER TABLESPACE ADD DATAFILE {データファイルのパス} SIZE {サイズ};
```

6-3-3 表領域の削除

DROP TABLESPACEコマンド

空の表領域を削除

```
DROP TABLESPACE 表領域名;
```

表領域に格納されているオブジェクトと表領域を削除

```
DROP TABLESPACE 表領域名 INCLUDING CONTENTS;
```

表領域に格納されているオブジェクトと表領域およびデータファイルを削除

```
DROP TABLESPACE 表領域名 INCLUDING CONTENTS AND DATAFILES;
```

- 表領域に割当てられていたデータファイルは、表領域の削除時に残すか削除するかを選択できます。SQLのDROP TABLESPACE文を使用する場合は、「INCLUDING CONTENTS AND DATAFILES」句を指定しないと、データファイルは削除されません。
 - 残してしまったら、一旦、仮のテーブルスペースの再利用テーブルファイルとして作成する その後、テーブルファイルを含めてテーブルスペースを削除する

6-3-4 bigfile表領域

smallfile

- 従来の小さい表領域は「smallfile表領域」
 - 最大サイズが32GB
- smallfile表領域のサイズを拡張する方法

- ALTER DATABASE文(リサイズ)で、既存のデータファイルのサイズを大きくする
- ALTER DATABASE文（リサイズ）で、既存のデータファイルの自動拡張を有効にする
- ALTER TABLESPACE文（追加）で、新しいデータファイルを追加する

bigfile

- 大きなサイズのデータベースを想定した「bigfile表領域」
 - 最大サイズが32TB
 - 1つのデータファイルだけで構成され、あとからデータファイルの追加はできない
- bigfile表領域を大きくする方法
 - 既存のデータファイルの自動拡張を有効にする
 - 既存のデータファイルのサイズを拡張する
 - 表領域のサイズを拡張する

6-4 UNDO表領域と一時表領域

UNDO表領域

- **UNDO表領域**は、更新前の過去データ（UNDOデータ）を保管する表領域
- UNDOデータによって、実現されること
 - トランザクションのロールバック
 - 読み取り一貫性の保証
 - ほかの利用者によって更新されても、問い合わせ時点のデータを返すことができる
 - 読み取り一貫性エラー（ORA-01555）
 - 大量のデータ更新によって、UNDOデータが失われる場合。このエラーの抑止のために **UNDO_RETENTION**初期化パラメータに大きな時間を設定。あるいは、UNDO表領域の自動拡張を有効化。
 - 一部のフラッシュバック機能
 - 表のデータを過去の状態に戻す
- 古いUNDOデータは自動的に削除される
- Oracleデータベースには複数のUNDO表領域を作成できるが、アクティブにできるのは1つだけ
- 「UNDOアドバイザ」は、データベースの過去のパフォーマンスの統計情報に基づいて、UNDO表領域(データの更新前の値であるUNDOデータを格納するための表領域)を固定サイズにする際に必要な最小サイズを提案します。

一時表領域

- **一時表領域**は、メモリ上で処理ができない大量のデータを処理する場合に利用
- データサイズが大きく、PGAに展開できない場合、一時表領域に作成される**一時セグメント**に展開される
- 一時表領域は、1つ以上の**一時ファイル**から構成される

問合せ時の「ORA-01555: Snapshot too old」というエラーに対処する方法

- 問合せ結果の取得に時間がかかるSELECT文の実行時や、フラッシュバック機能(過去のある時点のデータを検索したり、過去のある時点のデータに戻すことができる)を利用した時に「ORA-1555: Snapshot too old(スナップショットが古すぎます)」というエラーが発生することがあります。

- これは、読取り一貫性やフラッシュバック機能の実現に必要なUNDOデータが新たなUNDOデータで上書きされてしまい、必要なデータがUNDO表領域にないことを意味します。
- このエラーを回避するには、UNDOデータが上書きされないように以下の対処を行います。
 - UNDO表領域のサイズを大きくする
 - UNDO保存期間(UNDO_RETENTIONパラメータ)を長く設定する
 - UNDO保存保証を有効にする

表領域の使用状況を管理するのに、アラート通知を利用しています。Oracle Databaseで定義されているアラートしきい値はどれ。

- 警告: 領域が残り少なくなり始める境界値
- クリティカル: 至急対処が必要な深刻な境界値