

表領域

- シードテンプレートを使用したデータベース作成の際、自動的に作成される表領域
 - SYSTEM
 - SYSAUX
 - TEMP
 - UNDOTBS1
 - USERS
 - EXAMPLE
- smallfile表領域のサイズ拡張
 - 表領域のサイズは、割り当てられたデータファイルの合計サイズで変わります。
 - ALTER DATABASE文（新規）で、既存のデータファイルのサイズを大きくする
 - ALTER DATABASE文（新規）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
- セグメント
 - 表領域の作成時、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
 - SEGMENT SPACE MANAGEMENT
 - 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。
- 表領域をオフラインに設定する
 - データベースの一部をアクセス不可にし、残りの部分は通常どおりアクセスできるようにする場合
 - アプリケーションのメンテナンスを行う間、アプリケーションとその表を一時的にアクセス不可にする場合
 - 表領域のバックアップをオフラインで実行する場合(表領域はオンラインでも使用中でもバックアップ可能)
 - ハードウェアまたはソフトウェア障害の後に表領域をリカバリする場合
 - 表領域のデータファイルの名前の変更または再配置をする場合
- 削除
 - 表領域
 - 表領域内のデータベース・オブジェクト
 - 表領域に割当てられているデータファイル

```
DROP TABLESPACE 表領域名 [[INCLUDING CONTENTS] AND DATAFILES] [CASCADE CONSTRAINTS];
```

- 表領域を削除すると、表領域に格納されているデータベース・オブジェクトのデータは削除され、またデータ・ディクショナリに格納されているオブジェクトの定義も削除され、オブジェクトを使用できなくなります。
 - 表領域に割り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。
 - 表を削除すると、ビューやシノニムなどのオブジェクトは無効化されますが、削除はされません。
 - なお、SYSTEM表領域は削除できません。
- 割り当て
 - 表領域の割り当て制限は、ユーザーの作成時や変更時に「QUOTA」句で設定できます
 - ユーザーが表領域内で使用できる最大サイズである割り当て制限を指定します。UNLIMITEDは無制限です。永続表領域でない一時表領域とUNDO表領域に対しては、QUOTAを設定できません。

データベーステーブル

作成

- 30バイト以下であること(12c R2以降は128バイト以下)
- 使用できる文字は、英数字(A～Z, a～z, 0～9)
- 日本語環境では漢字, ひらがな, カタカナも使用可
- 使用できる記号は、「_」,「\$」,「#」のみ
- 先頭の文字は、数字, 記号以外の文字
- Oracle SQLの予約語は使用できない(DATE, FROM, GROUPなど)
- アルファベットの太文字・小文字は区別されない

CREATE TABLE ~ AS SELECT

- CREATE TABLE ~ AS SELECT文で、既存の表の定義とデータを新しい表にコピーすることができます。制約はNOT NULL制約(NULL値を禁止する制約)のみコピーされ、その他の制約や索引(データの問合せのパフォーマンスを向上させるオブジェクト)はコピーされません。

```
CREATE TABLE 表名 [(列名 [DEFAULT デフォルト値] [, 列名...])]
AS SELECT文;
```

削除

```
``sql
DROP USER ユーザー名 [CASCADE];
``
```

制約

- データ値の制約
 - 「列制約」と「テーブル制約」という2つの基本制約
 - テーブル制約（表レベルの制約）の場合、**NOT NULL**制約は追加できない（MODIFYしろ）
 - 列レベルで外部キー制約を定義する場合は、**「FOREIGN KEY」**キーワードは指定できない

`ALTER TABLE` 表名 `MODIFY` (列名 制約タイプ); ← 列レベルの制約

`ALTER TABLE` 表名 `ADD` 制約タイプ(列名²,列名³...); ← 表レベルの制約

`ALTER TABLE` 表名 `MODIFY` (列名 [`CONSTRAINT` 制約名] 制約タイプ); ← 列レベルの制約
(制約名定義)

`ALTER TABLE` 表名 `ADD` [`CONSTRAINT` 制約名] 制約タイプ(列名 [, 列名]); ← 表レベルの
制約(制約名定義)

TEST表に対して列レベル、表レベルの制約を追加する場合

- TEST表

名前	NULL?	型
TEST_ID		NUMBER(4)
TEST_NAME		VARCHAR2(20)

- TEST_IDに主キー制約を加える（列レベル）

```
ALTER TABLE test MODIFY (test_id PRIMARY KEY);
```

- TEST_IDに主キー制約を加える（表レベル）

```
ALTER TABLE test ADD CONSTRAINT id_pk PRIMARY KEY(test_id);
```

CUSTOMERS表がある状態で、新しいテーブルを作る

- CUSTOMERS表

名前	NULL?	型
CUSTOMER_ID	NOT NULL	NUMBER(4)
CUST_FIRST_NAME	NOT NULL	VARCHAR2(20)
CUST_LAST_NAME	NOT NULL	VARCHAR2(20)
CUST_ADDRESS	NOT NULL	VARCHAR2(40)
CUST_POSTAL_CODE	NOT NULL	VARCHAR2(10)

```
CREATE TABLE order1
(order_id NUMBER(8) CONSTRAINT ord_pk PRIMARY KEY,
 prod_id NUMBER(4) CONSTRAINT prod_nn NOT NULL,
 order_date DATE,
 ---表レベル指定 (Foreign key だけ)
 customer_id NUMBER(4) CONSTRAINT cust_fk FOREIGN KEY(customer_id)
                                     REFERENCES customers,
 (customer_id),
 units NUMBER(4),
 email VARCHAR2(50) CONSTRAINT email_uk UNIQUE
);
```

```
---以下はNG
---列レベル指定
customer_id NUMBER(4) CONSTRAINT cust_fk FOREIGN KEY
                                     REFERENCES
customers(customer_id),
```

ビュー

```
CREATE [OR REPLACE] VIEW ビュー名 [(列別名 [, 列別名...])]
AS 副問合せ
```

- ビューの列別名にデフォルト値は指定できない

```
---以下hNG
CREATE OR REPLACE VIEW emp_view (emp_id, emp_name, hiredate DEFAULT SYSDATE)
AS SELECT * FROM emp;
```

- 次のSQL文でEMP表を作成しました。

```
CREATE TABLE emp
(emp_id NUMBER(4) NOT NULL,
 emp_name VARCHAR2(20) NOT NULL,
 hiredate DATE NOT NULL);
```

- 次のSQL文でEMP_VIEWビューも作成しました。このビューについて、正しい記述はどれですか。

```
CREATE OR REPLACE VIEW emp_view (id, name)
AS SELECT emp_id, emp_name FROM emp;
```

ビューの特徴として、いくつかの条件を満たせば、ビューに対して行う挿入、更新、削除が実表にも反映されます。例えば、ビューに挿入したデータが実表にも挿入されるようにするには、実表に定義されているすべてのNOT NULL制約の列がビューに含まれている必要があります。但し、実表のNOT NULL列にデフォルト値が設定されていれば、この列が

ビューに含まれていなくても、ビューを通じて表に行を挿入できます。

設問のEMP表にはすべての列にNOT NULL制約が定義されており、デフォルト値は設定されていません。EMP_VIEWビューには、実表であるEMP表のHIREDATE列(NOT NULL制約の列)が含まれていないので、ビューを通じて表に行を挿入できません。

スキーマ

- スキーマ・オブジェクト
 - 表、ビュー、索引などのデータベース・オブジェクトはいずれかのスキーマ(1人のユーザーが所有するオブジェクトの集合)に属しており、そのスキーマ内では一意の名前がつけられる
 - 表
 - 索引
 - ビュー
 - シノニム
 - 順序

索引

- 索引は、表に関連付けられ、データの問合せのパフォーマンスを向上させるスキーマ・オブジェクト
 - WHERE句の条件や結合条件としてよく使用される列
 - 列にNULL値が多く含まれており、NULL値以外の値を指定して検索する列
 - 表のデータ量が多く、多くの問合せで15%未満の行を検索する列
 - カーディナリティが高い(列の値の種類が多い)列
- 特徴
 - 1つの列、もしくは複数の列の組合せに対して作成できる
 - 1つの表に列が異なる複数の索引を作成できる
 - 主キー制約または一意キー制約が定義されている列には自動的に索引が作成される
 - DML文によってデータが挿入、更新、削除されると、索引は自動的にメンテナンスされる
 - Oracle Databaseが索引を使用するかどうかを判断する(INDEX/NO_INDEXなどヒント句の指定が無い場合)
 - 表と異なる表領域に作成できる

制御ファイル

- 制御ファイルはデータベースの物理構成と運用情報が記録されたバイナリ形式のファイル
 - データベース名
 - データファイルのファイル名と格納場所
 - REDOログ・ファイルのファイル名と格納場所
 - データベース作成のタイムスタンプ
 - チェックポイント情報
 - REDOログ・ファイルのログ順序番号
- 制御ファイルは初期化パラメータ「CONTROL_FILES」で指定された位置とファイル名で格納
- データベースのマウント時に読み込まれる
- 制御ファイルが破損すると、データベースは起動できないため、多重化して運用することが推奨。但し、**多重化した制御ファイルのうち1つでも破損してしまうとデータベースは停止する**

- 多重化の目的は、障害が発生してもシステムを停止させずに運用することではなく、制御ファイルに格納されている情報を保護すること

REDO

LGWR

- 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る
 - 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
- 多重化
 - REDOログ・ファイルは障害発生時のリカバリ処理に不可欠であるため、多重化することが推奨
 - 多重化された同一の内容のファイルをまとめて「REDOログ・グループ」と呼び、REDOログ・グループの個々のファイルを「REDOログ・メンバー」と呼びます。
 - REDOログ・グループを2つ以上のメンバーで構成し多重化していれば、あるメンバーが壊れてもインスタンスを停止することなく運用を続けることができる

UNDO

- Oracle Databaseの変更をロールバックするための情報を格納する領域がUNDO表領域
 - UNDO表領域は複数作成できます。但し、有効にできるUNDO表領域は1つだけです。
 - UNDOデータ専用のため、ユーザーが作成した表や索引などのデータベース・オブジェクトは格納できない
 - Oracle Databaseではデフォルトで自動UNDO管理が有効になっており、UNDO表領域のデータはOracle Databaseにより自動管理
- Oracle Databaseでは、実行中のトランザクションの処理を取り消すために、ロールバック用のUNDOデータを保持しています。UNDOデータとは、データの更新前の値です。
- UNDOデータはUNDO表領域のUNDOセグメントに格納されます。ユーザーがデータの更新処理などを実行しトランザクションが開始されると、そのトランザクション用にUNDO表領域のUNDOセグメントが自動的に割り当てられ、以後トランザクションがコミットされるまで、同一トランザクションで発生したUNDOデータは、割り当てられたUNDOセグメントに格納されます。1つのUNDOセグメントに複数のトランザクションが割り当てられます。

「ORA-1555: Snapshot too old(スナップショットが古すぎます)」

- 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
 - 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

権限

- オブジェクト権限は、特定のデータベース・オブジェクトへの操作を許可するための権限
 - オブジェクトの種類（表 or ビュー）によって付与できる権限が異なる
 - 付与された権限を他のユーザーに付与できる（自分のオブジェクトなら可能）
 - オブジェクトの所有者はオブジェクト権限が付与されていなくてもオブジェクトを参照できる
- CREATE ANY TABLE権限
 - 自分以外のユーザーのスキーマ(1人のユーザーが所有するオブジェクトの集合)に表を作成できる。ただし、これだけでは作成した表を参照したりデータを追加する権限はありません。
- ロール
 - ロールとは、複数の権限を1つにまとめて名前を付けたもの
 - ロールにはシステム権限、オブジェクト権限、他のロールを含めることができます。
 - また、ロールはパスワードで管理できます。パスワードで管理する場合は、ロールを付与され、且つロールのパスワードを知っているユーザーだけがロールを使用可能にできます。
 - なお、ロールの作成や、ユーザーへのロールの付与、ユーザーからのロールの取消しはデータベース管理者が行います。

ロール	説明
CONNECT	CREATE SESSIONのみ。データベースへの接続を可能にする。 OEMを利用してユーザを作成すると自動的に付与
RESOURCE	スキーマオブジェクトの作成、変更、削を可能にする。 開発者など向け
DBA	SYS/SYSTEMにデフォルトで付与。 「インスタンスの起動/停止」「バックアップ/リカバリ」の権限は含まれない
SELECT_CATALOG_ROLE	データディクショナリ内のオブジェクトに対するSELECT権限
EM_EXPRESS_BASIC	EM Expressに接続して、読み取り専用モードでページを表示する権限
EM_EXPRESS_ALL	EM Expressに接続して、すべての機能を使用できる権限

障害

- フラッシュバック・ドロップ
 - 誤って削除した表を元に戻す機能
 - 他の表を参照する参照整合性制約(外部キー制約)は復旧できない
- メディアリカバリ : 完全リカバリ」と「不完全リカバリ(Point-in-Timeリカバリ)」 = **どちらも ARCHIVELOGモードのみ**
 - 契機
 - 誤って表領域を削除した
 - 誤ってOS上のデータファイルを削除した
 - ハードディスクの故障により、制御ファイルが破損した
 - 完全リカバリ

- 障害が発生した直前の状態に戻すリカバリ方法
 - バックアップファイル
 - 全ての**アーカイブREDOログ・ファイル**
 - **REDOログ・ファイルの変更履歴**
- 不完全リカバリ（あくまで、途中まで指定してのリカバリ。全然不完全じゃない！）
 - 過去の任意の時点を指定し、その時点までのバックアップ・ファイルと変更履歴を適用して、
指定した時点の状態にリカバリする方法
- アーカイブ・モード
 - データベースのアーカイブ・モードとは、REDOログ・ファイルのコピーであるアーカイブREDOログ・ファイルを作成するかしないかの設定のことです。
 - アーカイブ・モードは、ALTER DATABASE文を使用して、データベースがMOUNT状態の時に切り替えます。アーカイブ・モードの切り替えは、SYSDBA権限を持つ管理者ユーザーで行います。変更は

```
ALTER DATABASE ARCHIVELOG;
```

- NOARCHIVELOGモード
 - メディア・リカバリできない
 - データベースのOPEN時にバックアップを取得できない(CLOSEでの一貫性バックアップなら可能)
 - データベースのデフォルトのモードである
- インスタンス障害
 - 停電や「SHUTDOWN ABORT」コマンドでの強制終了により、Oracleインスタンスが突然停止してしまった場合
 - 初めてデータベースをオープンした時にSMONプロセスによって自動的に実行
- インスタンス・リカバリ
 - SMONプロセスは、データベースのオープン時に制御ファイルとデータファイルの整合性をチェック。
 - 1. REDOログ・ファイルの更新履歴をデータファイルに適用(ロールフォワード)。この時、UNDOセグメントのUNDOデータも復元。
 - 2. ロールフォワード後のデータファイルにはまだコミットされていないデータも含まれているため、UNDOデータを使用してコミットしていないデータをロールバック。
 - 3. データファイルとREDOログ・ファイルの同期が取れた状態。

バックアップ

- AWR
 - Oracle Databaseでは、MMONプロセスがデータベースの稼働状況とパフォーマンスの統計情報を定期的に収集し、AWR(Automatic Workload Repository: 自動ワークロード・リポジトリ)と呼ばれるSYS AUX表領域内の領域に格納しています。これらの情報は特定の時点におけるシステムの統計サマリーで、「スナップショット」と呼ばれます。スナップショットは、デフォルトでは**60分ごとに収集され、8日間保存**されます。保存期間を過ぎたスナップショットは自動的に削除されます。

- 一貫性バックアップ
 - 一貫性バックアップはデータベースをクローズした状態で取得するバックアップで、オフライン・バックアップとも呼ばれます。データベースを正常に停止する際にREDOログ・ファイルのコミット済みの変更がデータファイルに書き込まれるため、データファイルはトランザクションの一貫性が保たれた状態となります。
 - データベースをクローズした状態でバックアップを取得
 - ARCHIVELOGモード/NOARCHIVELOGモードどちらの運用時でも取得できる (NOARCHIVELOGモードで運用時は一貫性バックアップしか取得できない)
 - REDOログ・ファイルのコミット済みの変更がデータファイルに適用されている
 - 一貫性バックアップのファイルをリストアすると、メディア・リカバリをしなくてもデータベースをオープンできる(バックアップ取得時点まで復旧)

アドバイザ

メモリーの設定アドバイザ

- 自動メモリー管理
 - メモリーアドバイザ : 全体のメモリーサイズ (SGA + PGA)
- 自動共有メモリー管理 (+ 自動PGAメモリー管理)
 - SGAアドバイザ + PGAアドバイザ
- 手動共有メモリー管理 (+ 自動PGAメモリー管理)
 - バッファ・キャッシュ・アドバイザ + PGAアドバイザ

パフォーマンス

- ADDM
 - データベース全体を監視し、パフォーマンス上の問題を診断する
- SQLチューニング・アドバイザ
 - SQL文を分析し、パフォーマンス改善するための推奨事項を提案する
 - SQLプロファイル(SQL文の最適な実行計画のための統計データ)の作成・変更
 - 新しい索引の作成
 - オプティマイザ(問合せ処理の最適化を行う機能)統計のリフレッシュ
 - SQLの再構築
 - SELECT文のみ!
 - トップ・アクティビティ: 過去1時間に実行された、最もリソースが集中したSQL文を分析する
 - 履歴SQL: 過去の任意のSQL文を24時間単位で分析する
 - SQLチューニング・セット: 指定した一連のSQL文を分析する
 - SQLプロファイルの自動実装が有効な場合(デフォルトでは無効です)、パフォーマンスが少なくとも3倍は向上するとみられるSQL文に対してSQLプロファイルが作成され、自動的に実装
- SQLアクセス・アドバイザ
 - SQLアクセス・アドバイザは特定のSQLのワークロード(処理量、負荷)を分析し、索引やマテリアライズド・ビュー(実体を持ったビューのこと)などの作成・削除やオブジェクトのパーティション化(細分化)といった、主にスキーマの変更に関する推奨事項を作成します。SQLチューニング・アドバイザと異なり、SELECT文のみでなくその他のDMLも分析の対象となります

- SQLワークロード: 現行および最新のSQLアクティビティや、開発環境などユーザー定義のワークロードなど
- UNDOアドバイザー
 - UNDO表領域の最小サイズと推奨サイズを提案する

ツール

- SQL*Loader
 - SQL*Loaderは外部ファイルに格納されたデータをデータベースの表にロードするためのツール
 - SQL*Loaderを使用する際は、SQL*Loaderにおける「制御ファイル」と「データファイル」が必要。
 - これらはOracle Databaseのデータベースを構成する制御ファイルとデータファイルとは全く異なる
 - 制御ファイル(*.ctl)は、データファイルの格納場所やロード先の表に関する情報が記述されたテキストファイル
 - データファイルは、ロードするデータがカンマ区切りなどで記述されたテキストファイル

用語メモ

- オーバーヘッド
 - 作業負荷
- データ・ディクショナリ
 - データベースに関する様々な管理情報が格納された読取り専用の表。データ・ディクショナリ表およびデータ・ディクショナリビューの所有者はSYSユーザー。
- DEFAULTプロファイル
 - 「パスワード・ポリシー」は、パスワードの有効期限や複雑さなどのパスワードに関するルールです。
 - ユーザーの作成時に指定する「プロファイル」と呼ばれるデータベースのリソースに制限を設定するためのオブジェクトによって、パスワード・ポリシーがユーザーに割当てられます。ユーザーの作成時に明示的にプロファイルを指定しなければ、事前構成済の「DEFAULT」プロファイルにより、デフォルト・パスワード・ポリシーが割当てられます。
- 読取り一貫性
 - あるユーザーがデータの更新中に他のユーザーがそのデータを検索した場合、更新前の最新のコミット済のデータ、つまりUNDOデータを参照させる機能です。
-

メモ

- Oracle Databaseに格納できるデータベース常駐型プログラムユニット
 - パッケージ
 - プロシージャ
 - トリガー
 - JavaソースとJavaクラス

- ファンクション