

## 第8章 スキーマオブジェクトの管理

---

### 8-1 スキーマ

---

#### 8-1-1 スキーマとは

- あるユーザが所有する表などのオブジェクトを格納する箱のようなもの
- ユーザーを作成すると、ユーザー名と同名のスキーマが自動的に作成される
- ユーザーが表などのオブジェクトを作成すると、そのユーザーがそれらのオブジェクトの所有者となる

#### 8-1-2 スキーマオブジェクトと非スキーマオブジェクト

- オブジェクトを所有するユーザーに対応するスキーマに格納されるオブジェクト=**スキーマオブジェクト**
- そうではないオブジェクト=**非スキーマオブジェクト**

#### 8-1-3 スキーマと権限

- スキーマ内のオブジェクトへのアクセス
- 他スキーマのオブジェクトへのアクセス

#### 8-1-4 名前空間としてのスキーマ

- スキーマは、オブジェクトの名前の衝突を防ぐ**名前空間**としての役割をもつ
  - HRユーザーはt1表を所有している場合
    - HRスキーマには同名の表（t1表）を作成不可
    - OEスキーマには同名の表（t1表）を作成可能

#### 他スキーマ内のオブジェクトの指定

- オブジェクト名の前に「スキーマ名.」を前置きする

```
SELECT * FROM oe.t1]
```

## 8-2 データ型・制約と表の作成

---

### 8-2-1 データ型

- それぞれの列には、データ型を指定する

データ型	格納できるデータ
NUMBER	整数及び小数

---

データ型	格納できるデータ
CHAR(n)	サイズがnバイト固定長の文字列。nは1~2000。 nバイトよりも小さいサイズの文字列を入力した場合、不足サイズ分の空白文字が自動的に末尾に追加される
VARCHAR2(n)	最大サイズがnバイトの可変長文字列。nは1~4000。空白は埋めない
BLOB	最大128テラバイトのバイナリデータ
CLOB	最大128テラバイトの文字列
DATE	小数秒を含まない日付と時刻
TIMESTAMP	小数秒を含む日付と時刻

## NUMBER型

- 3つの型の指定方法
  - NUMBER(n)
    - n桁の整数。nは1~38。
  - NUMBER(n,m)
    - 最大有効桁数n、小数点以下の最大桁数mの小数または整数。nは1~38
      - NUMBER(7,2)
        - 整数部分の桁数：5桁(=n-m) -> **7桁ではない**
        - 小数部分の桁数：2桁
  - NUMBER
    - 対応できる最大の精度を指定したことになる

## CHAR型とVARCHAR2型

- 表に記載

## BLOB型・CLOB型（ラージオブジェクト）

- BLOB型
  - 画像や音声などのバイナリデータを格納する
- CLOB型
  - CHARなどで対応できない大きなサイズの文字列

## DATE型・TIMESTAMP型

- 表に記載

## 8-2-2 制約

- 表に無効なデータが格納されることを禁止するルール

制約	意味	機能
NOT NULL	NOT NULL 制約	列にNULLが設定されることを禁止する

制約	意味	機能
UNIQUE	一意キー制約	列に重複値が設定されることを禁止する
PRIMARY KEY	主キー制約	列に重複値またはNULLが設定されることを禁止する
FOREIGN KEY	外部キー制約 参照整合性制約	列の値が、関連する表の一意キー、または主キーの値と一致することを保証する
CHECK	チェック制約	列に、指定されたチェック条件を満たさない値が設定されることを禁止する SELECT文のWHERE句へ指定できる条件と同じ

- PRIMARY KEY制約を設定すると、自動的にその列に索引が作成される

## FOREIGN KEY 制約と外部キー

- 異なる表にある行データと行データが関連する = **リレーションシップ**
  - 関連する外部のテーブルのデータに対するキー = **外部キー** = 外部キーを設定した表は**子表**
  - 外部キーが参照する列 = **参照キー** = 参照するテーブルは**親表**
- 親表の参照キーにないデータを子表の外部キーに格納するのは禁止 = **FOREIGN KEY制約**

## 8-2-3 表の作成

- CREATE TABLEコマンド

```
CREATE TABLE [スキーマ.] テーブル名
(列名1 データ型(サイズ) CONSTRAINTS 制約名 [制約]

,列名2 NUMBER CONSTRAINTS PRODUCTS_PK [PRIMARY KEY]

,列名3 VARCHAR(20) CONSTRAINTS PRODUCTS_UK [UNIQUE]
.....
);
```

- 命名規則
  - ・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)
  - ・アルファベットの大文字・小文字は区別されない
- 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文で表を作成します。表の作成時に必須の項目

- 表名
- 列名
- 列のデータ型とサイズ

## 8-3 表の削除とゴミ箱

```
DROP TABLE <表名>;
```

- 表を削除すると、表に格納されていたデータも削除される。制約や索引も削除される

### ゴミ箱とフラッシュバックドロップ

- ゴミ箱に入っている削除済みの表を復元する機能を**フラッシュバックドロップ**という
  - ゴミ箱に入っている削除済みの表の実体は、表に対して単に「削除済み」マークをつけただけ
  - 表が格納されていた表領域に格納される

```
FLASHBACK TABLE <表明> TO BEFORE DROP;
```

### フラッシュバックドロップの有効化/無効化

- 表の復元を許可するRECYCLEBIN初期化パラメータを「ON」にする必要がある
- デフォルトで「ON」

## 8-4 索引とビュー

### 8-4-1 索引

- 表の列に索引を作成すると、検索条件にマッチする行を高速に得ることができる

#### 索引列

- 1つの列、または複数の列の組み合わせに対して作成する
- 索引を設定した列を**索引列**という

#### 索引の仕組み

- 索引は、**ルートブロック->ブランチブロック->リーフブロック**という木構造
  - ある列値を持つデータを検索できる
  - 各ブロックで1-100/101-200/・・・のように範囲で区分をしている
- 索引のリーフには、列値とROWIDの組が格納される
  - ROWIDとは、その列値を含む行データの位置情報（アドレス）
- 索引を使ってROWIDを得ると、ROWIDをキーにして、表からその列値を持つ行データを直接取り出す

## 索引で処理を高速化できる条件

- 処理を高速化できるのは、検索にマッチするデータの件数が少ない場合のみ
  - 索引を使用する/しないの判断は、**オブティマイザ**によって行われる

## 索引の留意点

- 索引は1つの列に対してだけではなく、列の組み合わせに対しても作成できる
- CREATE INDEXコマンドで索引を作成できる
- 索引は検索対象のデータの件数が少ない場合に有効
- 1つの表に多数の索引を作成すると、行の更新時の負荷が増加する
- 索引を定義した行が削除されると、その索引も一緒に削除される
- PRIMARY KEY制約またはUNIQUE制約を設定すると、制約を設定した列に対して索引が自動的に作成される

## 8-4-2 ビュー

- 1つあるいは複数の表に対するSELECT文により定義されるスキーマオブジェクト
- SELECT文を指定した**CREATE VIEW**コマンドでビューを作成できる
- ビューに対して、SELECT文を発行すると、ビューの手筈に含まれるSELECT文が実行され、ビューに対する**実表**からデータを取得する
- ビューは**仮想表**とも呼ばれる（ビューは実体のない表だから）
  - 例えば、テーブルA・テーブルB・テーブルCを結合して使用する場合があります。それも使用頻度は高く、よく使う場合。この場合、使用する際に毎回結合したSQLを書くのは面倒です。なので、テーブルA・テーブルB・テーブルCを結合したビューDを作ります。ビューはテーブルと同じようにSELECTができるので、そのままビューDをSELECTするだけで使えます。長い結合SQLを書く必要がありません。

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

## ビューによるデータの保護

- ビューの実表への参照権限を与えずに、ビューへの参照権限のみを付与して、アクセス可能な範囲を限定できる

## ビューの留意点

- ビューにはデータが格納されない
- ビューに対してDMLを発行すると、ビューが参照している実表のデータが更新される
- ビューを使用してもSQL処理の高速化はできない。（めんどくさい指定を排除できるだけ）
- 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。
- ビューの列に算術式を使用できる

## 8-5 データの管理

---

### 8-5-1 SQL\*Loader

- CSVファイルのようなOS上のファイルに記載されたデータをOracleデータベースの表にロードするツール
  - 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」が必要です。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ものです。
- データロード方法
  - 従来型パス
    - 1行ずつINSERT文を実行してデータを挿入する
    - 通常のINSERT処理と同様にデータベース・バッファ・キャッシュを使用（時間かかる）
    - ロード中にほかのユーザーが表を更新できる
  - ダイレクト・パス
    - データブロックをフォーマットして、データベース・バッファ・キャッシュを経由せず書き込む
    - データベースへの負荷を大幅に軽減できる
    - ロード中にほかのユーザーが表を更新できない
  - 外部表

### 8-5-2 Data Pump

---

- Oracleデータベース間でデータを移動するツール
- expdpコマンド（エクスポート）とimpdpコマンド（インポート）がある
- 流れ
  - expdpコマンドを実行し、移動元データベースからデータをダンプファイルにエクスポートする
  - ダンプファイルを移動元データベースサーバーから、移動先データベースサーバーにコピーする
  - impdpコマンドを実行し、データを含むダンプファイルを移動先データベースにインポートする

ALTER TABLE文で、既存の列へ制約を定義できます。

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

```
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
```

- NOT NULL制約を追加するには、ALTER TABLE文のMODIFY句を使用する必要があります。

索引を作成するのに適した列

- 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

## 制約補足

- 制約とは、表に格納する値を制限するためのルールです。「整合性制約」ともいいます。
- 表に制約を定義することで、ルールに反するデータの追加や、ルールを満たさなくなるようなデータの更新、削除が行えなくなります。

制約	意味	機能	NULL 値
NOT NULL	NOT NULL 制約	列にNULLが設定されることを禁止する	×
PRIMARY KEY	主キー制約	列に重複値またはNULLが設定されることを禁止する 一意キーとNOT NULLを組み合わせたもの 1つの表に1つしか定義できない	×
UNIQUE	一意キー制 約	列に重複値が設定されることを禁止する	○
FOREIGN KEY	外部キー制 約 参照整合性 制約	列の値が、関連する表の一意キー、または主キーの値と一致する ことを保証する	○
CHECK	チェック制 約	列に、指定されたチェック条件を満たさない値が設定されること を禁止する SELECT文のWHERE句へ指定できる条件と同じ	○

- 主キー制約や一意キー制約を列に定義すると、自動的に一意索引(値の重複しない列に付与できる、問合せのパフォーマンスを向上させるオブジェクト)が作成されます。
  - NOT NULL制約は1つずつの列に対してのみ定義できます(列レベルの制約)。
  - NOT NULL制約以外は、複数の列の組合せに対しても定義できます(表レベルの制約)。
- 列レベルでの制約
  - 1つずつの列に対して制約を定義します。NOT NULL制約は列レベルでしか定義できません。
  - また、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ません。
- 表レベルの制約
  - 表の特定の列を指定して制約を定義します。複数の列の組合せに対しても定義でき、NOT NULL制約以外の制約を定義できます。

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

- PL/SQLプロシージャ
- PL/SQLファンクション

- PL/SQLパッケージ
- PL/SQLパッケージ本体
- PL/SQLトリガー
- Javaソース
- Javaクラス

## スキーマ・オブジェクト

- 表
  - データを格納するもっとも重要なオブジェクト
- 索引
  - データの検索の効率を向上させるオブジェクト
- ビュー
  - 表のデータの見せ方をカスタマイズするオブジェクト
- シノニム
  - ほかのスキーマ・オブジェクトの別名を表すオブジェクト
- 順序
  - 複数のユーザー間で連続した整数を生成するオブジェクト