はじめに

■コース概要と目的

Oracle 独自の手続き型言語である PL/SQL について説明します。PL/SQL の基本構文、ストアド・サブプログラム、トリガーの作成方法、またストアド・サブプログラムの管理について習得することを目的としています。

■受講対象者

これから PL/SQL を使用してアプリケーション開発をされる方。

■前提条件

「SQL トレーニング」コースを受講された方。もしくは、同等の知識をお持ちの方。
※DML 文、トランザクションについては既に習得済みであること。

■テキスト内の記述について

▼構文

[　]	省略可能
{ A | B }	A または B のどちらかを選択
n	数値の指定
_	デフォルト値

▼マーク

	指定バージョンからの新機能 （左記の場合、Oracle 12cR1 からの新機能）
	知っておいたほうが良いテクニック、もしくは注意事項
	参照ページ
	データ・ディクショナリ・ビュー

第 2 章

PL/SQL の基本記述

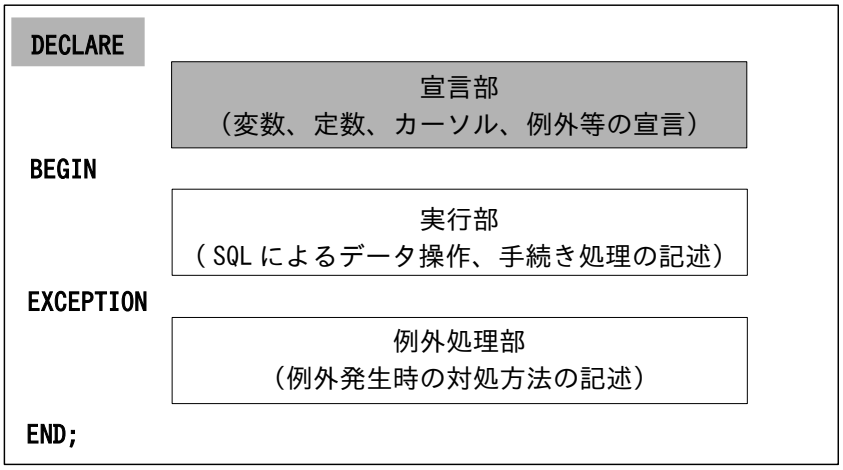
この章では、PL/SQL プログラムの基本的な記述方法について説明します。

１．宣言部
２．実行部
３．例外処理部

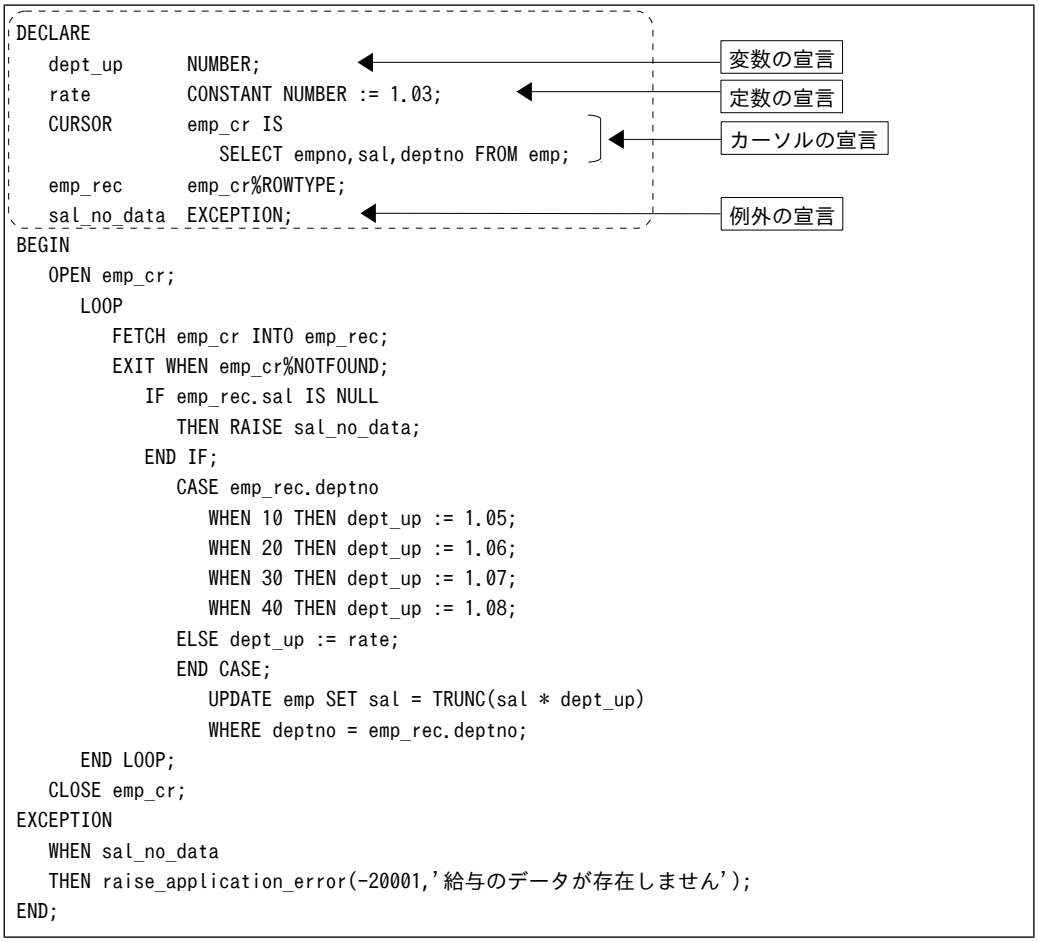
１．宣言部

宣言部では実行部で使用するオブジェクトを定義します。必要でなければ省略することも可能です。
宣言部は DECLARE キーワードで始まり、実行部の BEGIN キーワードにより暗黙的に終了します。
主に以下のオブジェクトを定義します。

・変数
・定数
・カーソル
・例外



■宣言部（点線内）



（1）変数と定数

手続き型処理では、処理で使用する値を変数や定数に格納（代入）します。変数や定数などの名前（識別子）は宣言部で指定します。

1）変数

変数とは、処理で使用する値を格納しておく場所です。処理の途中でデータベースのデータや計算結果を一時的に変数に格納（代入）できます。一度代入した変数に対して、異なる値を上書きして再利用できます。

```
変数名　データ型 [ NOT NULL ] [ { := | DEFAULT } 値 ] ;
```

・代入演算子（:=）またはDEFAULTキーワードを指定して初期値を設定できます。
・デフォルトでは変数はNULL状態です。
・NOT NULLキーワードを指定して変数にNULLが代入されないように定義することもできます。その際には変数に初期値を指定する必要があります。

2）定数

定数とは、変数と同様、処理で使用する値を格納しておく場所です。ただし、変数と異なり定数は1つの決まった値を保持するため、定数に代入した値は上書きできません。そのため、プログラム内で固定値を扱う場合に使用します。

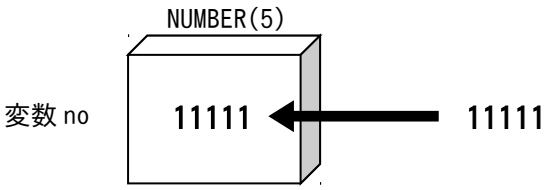
```
定数名　CONSTANT データ型 { := | DEFAULT } 値 ;
```

・定数宣言時には必ず初期値を指定します。

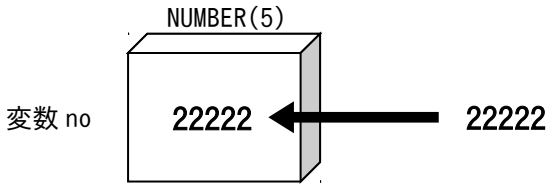
例）変数、定数の宣言を行う。

```
DECLARE
   /* 変数 */
   no      NUMBER(5);                          ← 数値型の値を代入できる変数noを宣言
   name    VARCHAR2(10) := 'SMITH';            ← 初期値を指定した変数nameを宣言
   /* 定数 */
   prod    CONSTANT VARCHAR2(3) := 'ABC';      ← 定数の宣言では必ず初期値を指定し、
                                                  それ以降は値の上書きはできない
BEGIN
…省略…
```

■変数

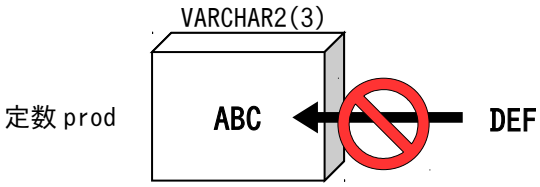


①変数noに11111という数値データを代入。



②変数noに22222という数値データを再度代入。
　→11111は上書きされる。

■定数



定数prodにDEFという文字データを代入。
→定数に対しては上書きできないためエラーとなる。

（２）%TYPE、%ROWTYPE 属性

%TYPE、%ROWTYPE 属性を使用すると、データベースの表の列や行、既に宣言された変数のデータ型を参照する変数を宣言できます。列や行を参照するため、表の定義が変更されたような場合もコードを変更する必要がありません。そのため、表のデータを取出して処理する場合に便利です。

１）%TYPE 属性

特定の表の列または既存の変数のデータ型を参照します。
※%TYPE 属性を使用した変数に初期値を指定することが可能です。

２）%ROWTYPE 属性

%ROWTYPE 属性は特定の表（またはビュー）の行構造定義を参照します。%ROWTYPE 属性は以下の特徴を持ちます。

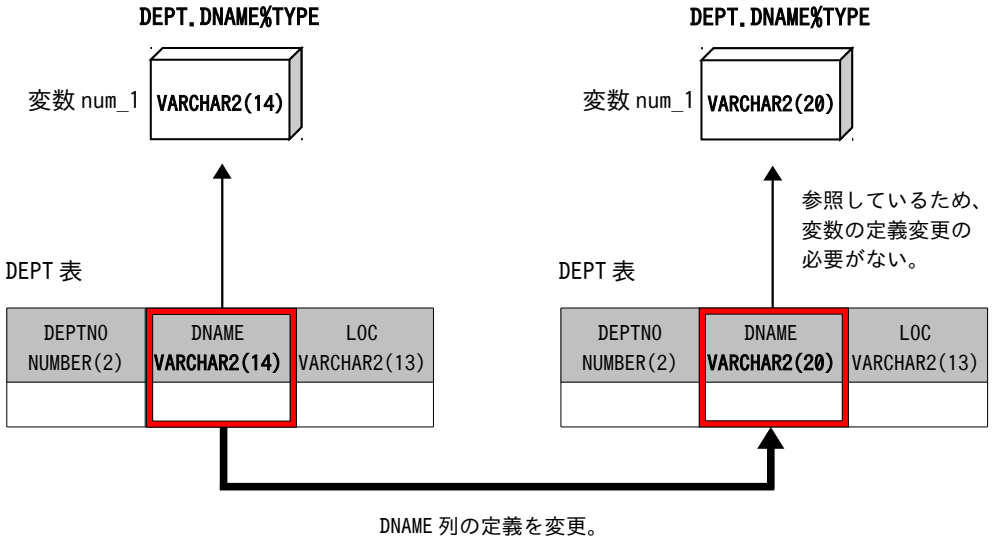
・変数は表の行構造と同じ数のフィールドを持ちます。
・各フィールドの名前とデータ型は参照表の列の名前とデータ型が対応付けられます。
・%ROWTYPE 属性を使用した変数は初期値を指定できません。
・各フィールドの参照は【変数名.フィールド名】で指定します。

例）%TYPE、%ROWTYPE 属性を使用して変数を宣言する。

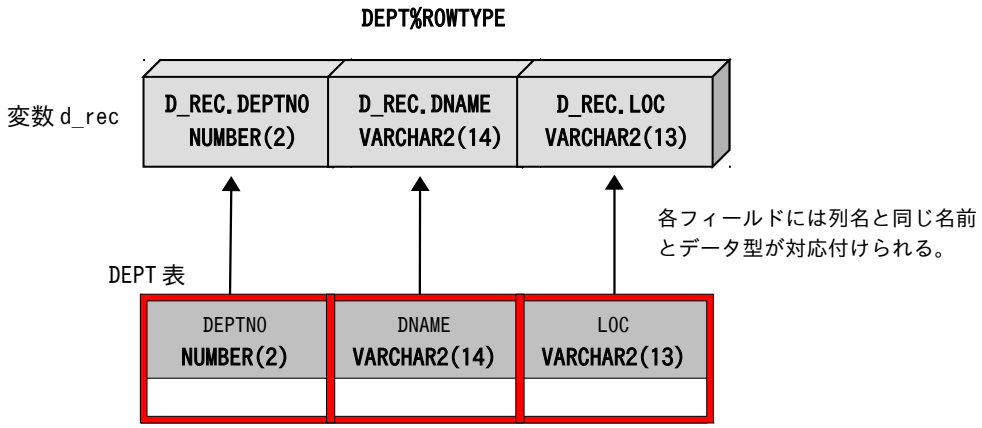
```
DECLARE
   /* %TYPE 属性 */
   num_1    dept.dname%TYPE;          ←――――――― ①
   num_2    num_1%TYPE;               ←――――――― ②
   e_job    emp.job%TYPE := 'SALESMAN';  ←――――― ③
   /* %ROWTYPE 属性 */
   d_rec    dept%ROWTYPE;             ←――――――― ④
BEGIN
…省略…
   UPDATE dept SET loc = d_rec.loc     ←――――― ⑤
   WHERE deptno = 10;
…省略…
```

①変数 num_1 は、DEPT 表の DNAME 列と同じデータ型で定義されます。
②変数 num_2 は、変数 num_1 と同じデータ型で定義されます。
③変数 e_job は、EMP 表の JOB 列と同じデータ型で定義され、初期値 SALESMAN が代入されます。
④変数 d_rec は、DEPT 表の各列と同じデータ型、名前が各フィールドで定義されます。
⑤変数 d_rec の LOC フィールドを参照するには、d_rec.loc と指定します。

■%TYPE 属性



■%ROWTYPE 属性



（3）変数、定数宣言時の考慮事項

1）識別子の命名規則

変数や定数などのPL/SQLオブジェクトに定義する名前のことを識別子と呼びます。識別子は以下の命名規則に従って定義します。

・識別子の長さは最大30バイトです。
・先頭は英字です。2文字目以降であれば、数字、ドル記号（$）、シャープ記号（#）、アンダースコア（_）を使用できます。
・ハイフン（-）、スラッシュ（/）、アンパサンド（&）、空白などの文字は使用できません。
・予約語は使用できません。

「予約語・キーワード一覧 」（付-5）

2）識別子の有効範囲

識別子は使用できる有効範囲が決められています。識別子を指定する際は以下の点に注意します。

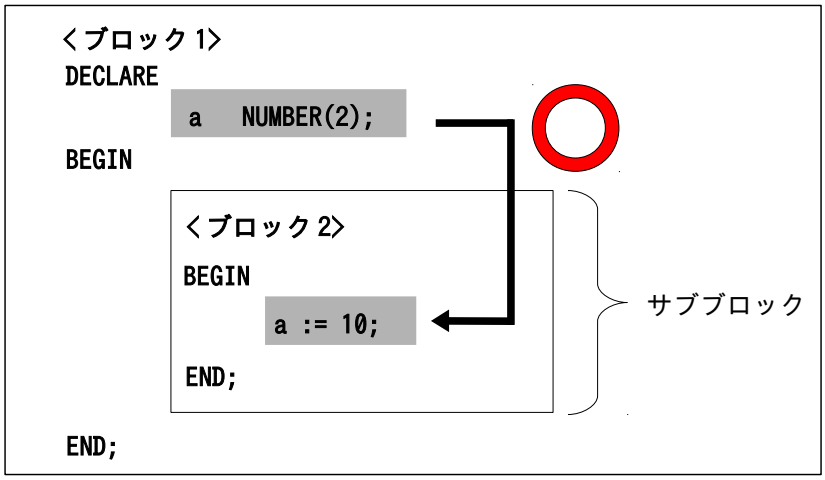
・識別子の有効範囲は、宣言されたブロックおよびサブブロックです。
・同一ブロック内で重複した識別子を宣言できません。
※異なるブロックであれば同じ識別子を宣言できますが、判別しやすくするため、異なる識別子で宣言することを推奨します。

3）注意事項

・PL/SQLでは前方参照ができません。変数または定数を参照する場合は、事前に宣言 されている必要があります。
※前方参照とは、まだ宣言していないオブジェクト名やプログラム名を参照することです。
・同一データ型の変数を、一度に複数宣言することはできません。変数は必ず1つずつ宣言してください。

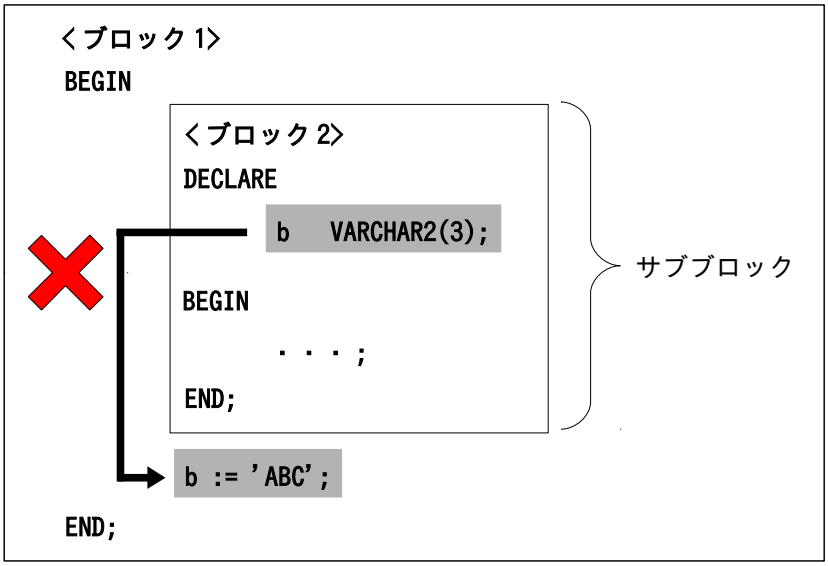
■識別子の有効範囲

＜識別子を参照できる例＞



変数 a は、ブロック 1 で宣言されているため、ブロック 1 とそのサブブロックである
ブロック 2 で参照できる。

＜識別子を参照できない例＞



変数 b は、ブロック 2 で宣言されているため、ブロック 1 では参照できない。