PL/SQL

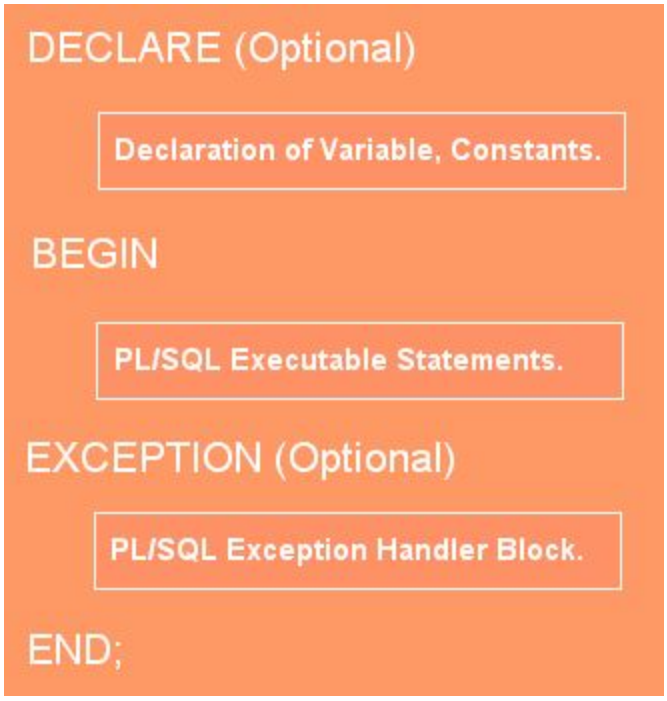
PL/SQL is a Procedural Language extension of non- procedural **Structured Query Language (SQL)**.
PL/SQL is **specially designed** for Database oriented activities. It combines the data manipulation power of SQL and procedural power of standard programming languages. It was developed by Oracle Corporation in the early 90's to enhance the capabilities of SQL.
PL/SQL is the superset of of SQL.So, it fully supports SQL data manipulation commands and SQL data types. PL/SQL also has boolean and composite data types to handle complex data.
PL/SQL is not a case sensitive language. It supports standard programming features such as control statements, procedures and functions.

PL/SQL is **capable** to **send** entire block of statements and execute it in the **Oracle engine** at once.

PL/SQL Block Structure

PL/SQL is block structured language divided into three logical blocks.
BEGIN block and END; keyword are compulsory, and other two block DECLARE and EXCEPTION are optional block. END; is not a block only keyword to end of PL/SQL program.

PL/SQL block structure follows divide-and-conquer approach to solve the problem stepwise.



DECLARE

Variables and constants are declared, initialized within this section.

Variables and Constants : In this block, declare and initialize variables (and constants). You must have to declare variables and constants in declarative block before referencing them in procedural statement.

Declare Variables and Assigning values : You can define variable name, data type of a variable and its size. Date type can be: CHAR, VARCHAR2, DATE, NUMBER, INT or any other.

Declare Constants and Assigning values : Constants are declared same as variable but you have to add the CONSTANT keyword before defining data type. Once you define a constant value you can't change the value.

BEGIN/Executable Section

It starts with the keyword BEGIN and ends with END. this is the section where the program logic is written to perform any task. The programmatic constructs like loops, conditional statements and SQL statements form the part of Executable section.

EXCEPTION

Exception section of a PL/SQL block starts with the keyword EXCEPTION. Any errors in the program can be handled in this section, so that the PL/SQL block terminates gracefully.

Note :

1. BEGIN and END; keyword are compulsory of any PL/SQL program.
2. Whereas DECLARE and EXCEPTION block are optional.

Fundamentals of PL/SQL

1. The PL/SQL Character Set

The PL/SQL language is constructed from letters, digits, symbols, and whitespace, as defined in the following table:-

Type	Characters
Letters	A-Z, a-z
Digits	0-9

Symbols	~!@#$%&*()_-+=|[ ]{ }:;"'< >?/
White Space	space, tab, carriage return

(a). Identifiers

Identifiers are names for PL/SQL objects such as constants, variables, exceptions, procedures, cursors, and reserved words. Identifiers:

- Can be up to 30 characters in length
- Cannot include whitespace (space, tab, carriage return)
- Must start with a letter
- Can include a dollar sign ($), an underscore (_), and a pound sign (#)
- Are not case-sensitive

If you enclose an identifier within double quotes, then all but the first of these rules are ignored.

(b). Literals

A literal is an explicit numeric, character, string or boolean value that is not represented by an identifier. See the following table for examples.

Literal	**Actual Value**
'That''s Entertainment!'	That's Entertainment!
'"The Raven"'	"The Raven"
'TZ="CDT6CST"'	TZ='CDT6CST'
''''	'
'''hello world'''	'hello world'

© . Delimiters

Delimiters are symbols with special meaning, such as := (assignment operator), || (concatenation operator), and ; (statement delimiter). The following table lists delimiters:-

Delimiter	Description
;	Statement Terminator
+	Addition operator
-	Subtraction Operator

*	Multiplication Operator
/	Division Operator
**	Exponentiation Operator
||	Concatenation operator
:=	Assignment Operator
=	Equality Operator
!=	Inequality operator
<	Less than operator
<=	Less than equal to operator
>	More than operator
>=	More than equal to operator
,	Item separator
“	Quoted literal delimiter
:	Host variable indicator
%	Attribute indicator
--	Single line comment indicator
/* and */	Multiline comment delimiter

(d) Comments

Comments are sections of the code that exist to aid readability. The compiler ignores them.
A single-line comment begins with a double hyphen (--) and ends with a new line. The compiler ignores all characters between the -- and the new line.
Multiline comments begin with slash asterisk (/*) and end with asterisk slash (*/). The /* */ comment delimiters can also be used on a single-line comment. The following block

```
DECLARE
  -- Two dashes comment out only the physical line.
  /* Everything is a comment until the compiler
     encounters the following symbol */
```

(e) Statements

A PL/SQL program is composed of one or more logical statements. A statement is terminated by a semicolon delimiter. The physical end-of-line marker in a PL/SQL program is ignored by the compiler, except to terminate a single-line comment (initiated by the -- symbol).

PL/SQL Variable

PL/SQL variable declaration always specifies the variable name, data type of variable and size. In PL/SQL variable declaration you can also specifies initial value of declared variables.

Variable Declaration Syntax

The general syntax to declare a variable is:
Variable_name Data type [Size] [NOT NULL: = value];

E.g. :-
DECLARE
salary number(4);
dept varchar(10) NOT NULL := “HR Dept”;

Explanation:

- variable_name is the predefined name of the variable.
- Data type is a valid PL/SQL data type.
- Size is an optional specification of data type size to hold the maximum size value.
- NOT NULL is an optional specification of the variable value can't be accept NULL.
- value is also an optional specification, where you can initialize the initial value of variable.
- Each variable declaration is terminated by a semicolon.

Variables Scope

PL/SQL variable scope is identified the region range which you can reference the variable. PL/SQL have two type scopes local scope and global scope,
Local variables - Variables declared in inner block and can't be referenced by the outside blocks.
Global variables - Whereas variables declared in a outer block and can be referencing by itself in inner blocks.

PL/SQL Constant :-

You can declare PL/SQL constants along with the value and can not change them throughout the program block. The constants declaration functionality is available in almost all programming languages.

PL/SQL Constant Declaration Syntax

The general syntax for declaring a constant variable is:
Constant_name CONSTANT Data type [Size] := Value;

e.g:-

```
DECLARE
salary_increase         CONSTANT number (3) := 10;
```

Explanation:

- Constant_name is a predefined name of the constant (similar to a variable name).
- CONSTANT is a reserved keyword.
- Data type is a valid PL/SQL data type.
- Size is an optional specification of data type. It holds maximum capacity value for the particular variable.
- Value must be assigned to a constant when it is declared. You can not assign or change it later.
- Each constant declaration is terminated by a semicolon.

Operator Precedence

The operations within an expression are done in a particular order depending on their precedence (priority).

Operator	**Operation**
1) **	exponentiation
2) +, -	identity, negation
3) *, /	multiplication, division
4) +, -, ||	addition, subtraction, concatenation
5) =, <, >, <=, >=, <>, !=, ~=, ^=, IS NULL, LIKE, BETWEEN, IN	comparison
6) NOT	logical negation
7) AND	conjunction
8) OR	inclusion.