

PL / SQL

PL/SQL tutorial provides basic and advanced concepts of SQL. Our PL/SQL tutorial is designed for beginners and professionals.

PL/SQL is a block structured language that can have multiple blocks in it.

Our PL/SQL tutorial includes all topics of PL/SQL language such as conditional statements, loops, arrays, string, exceptions, collections, records, triggers, functions, procedures, cursors etc. There are also given PL/SQL interview questions and quizzes to help you better understand the PL/SQL language.

SQL stands for Structured Query Language i.e. used to perform operations on the records stored in database such as inserting records, updating records, deleting records, creating, modifying and dropping tables, views etc.

What is PL/SQL

PL/SQL is a block structured language. The programs of PL/SQL are logical blocks that can contain any number of nested sub-blocks. PL/SQL stands for "Procedural Language extension of SQL" that is used in Oracle. PL/SQL is integrated with Oracle database (since version 7). The functionalities of PL/SQL usually extended after each release of Oracle database. Although PL/SQL is closely integrated with SQL language, yet it adds some programming constraints that are not available in SQL.

PL/SQL Functionalities

PL/SQL includes procedural language elements like conditions and loops. It allows declaration of constants and variables, procedures and functions, types and variable of those types and triggers. It can support Array and handle exceptions (runtime errors). After the implementation of version 8 of Oracle database have included features associated with object orientation. You can create PL/SQL units like procedures, functions, packages, types and triggers, etc. which are stored in the database for reuse by applications.

With PL/SQL, you can use SQL statements to manipulate Oracle data and flow of control statements to process the data.

The PL/SQL is known for its combination of data manipulating power of SQL with data processing power of procedural languages. It inherits the robustness, security, and portability of the Oracle Database.

PL/SQL is not case sensitive so you are free to use lower case letters or upper case letters except within string and character literals. A line of PL/SQL text contains groups of characters known as lexical units. It can be classified as follows:

- De limiters
- Identifiers
- Literals
- Comments

PL/SQL Variables

A variable is a meaningful name which facilitates a programmer to store data temporarily during the execution of code. It helps you to manipulate data in PL/SQL programs. It is nothing except a name given to a storage area. Each variable in the PL/SQL has a specific data type which defines the size and layout of the variable's memory.

A variable should not exceed 30 characters. Its letter optionally followed by more letters, dollar signs, numerals, underscore etc.

How to declare variable in PL/SQL

You must declare the PL/SQL variable in the declaration section or in a package as a global variable. After the declaration, PL/SQL allocates memory for the variable's value and the storage location is identified by the variable name.

`variable_name [CONSTANT] datatype [NOT NULL] [:= | DEFAULT initial_value]`

Declaration Restrictions:

In PL/SQL while declaring the variable some restrictions hold.

- Forward references are not allowed i.e. you must declare a constant or variable before referencing it in another statement even if it is a declarative statement.
`val number := Total - 200;`
`Total number := 1000;`
The first declaration is illegal because the TOTAL variable must be declared before using it in an assignment expression.
- Variables belonging to the same datatype cannot be declared in the same statement.
`N1, N2, N3 Number;`
It is an illegal declaration.

Naming rules for PL/SQL variables

The variable in PL/SQL must follow some naming rules like other programming languages.

- The variable_name should not exceed 30 characters.
- Variable name should not be the same as the table's column of that block.
- The name of the variable must begin with ASCII letter. The PL/SQL is not case sensitive so it could be either lowercase or uppercase. For example: `v_data` and `V_DATA` refer to the same variables.
- You should make your variable easy to read and understand, after the first character, it may be any number, underscore (`_`) or dollar sign (`$`).
- NOT NULL is an optional specification on the variable.

Initializing Variables in PL/SQL Everytime you declare a variable, PL/SQL defines a default value NULL to it. If you want to initialize a variable with

other value than NULL value, you can do so during the declaration, by using any one of the following methods.

You can also specify NOT NULL constraint to avoid NULL value. If you specify the NOT NULL constraint, you must assign an initial value for that variable.

You must have a good programming skill to initialize variable properly otherwise, sometimes program would produce unexpected result.

Variable Scope in PL/SQL:

PL/SQL allows nesting of blocks. A program block can contain another inner block. If you declare a variable within an inner block, it is not accessible to an outer block.