**SQL**: Structured query language.

**PL/SQL**: is a combination of SQL along with the procedural features of programming languages.

**Advantages of PL/SQL**

1. SQL is the standard database language and PL/SQL is strongly integrated with SQL.

PL/SQL supports both static and dynamic SQL.

a. Static SQL supports DML operations and transaction control from PL/SQL block.

b. Dynamic SQL is SQL allows embedding DDL statements in PL/SQL blocks.

2. allows sending an entire block of statements to the database at one time. This reduces network traffic and provides high performance for the applications.

3. Application written in PL/SQL are fully portable.

4. PL/SQL provides support for Object-Oriented Programming.

5. PL/SQL provides support for Developing Web Applications and Server Pages.

Declarations : This section starts with a keyword DECLARE. Defines all variables, cursors, sub programs and other elements to be used in the program.

Executable Commands : This section is enclosed between the keywords BEGIN and END and it is a mandatory section. It consists of the executable PL/SQL statements of the program. It should have at least one executable line of code, which may be just a NULL command to indicate that nothing should be executed.

Exception Handling : This section starts with the keyword EXCEPTION. This section is again optional and contains exception(s) that handle errors in the program.

Example :

DECLARE

message varchar2(20):= 'Hello, World!';

BEGIN

dbms\_output.put\_line(message);

END;

Identifiers :

+ Addition

- Subtraction

\* Multiplication

/ Division

% Attribute Indicator

' Character string delimiter

. Component selector

(,) Expression or list delimiter

: Host Variable Indicator

, Item seperator

" Quoted identifier delimiter

= Relational operator

@ Remote access indicator

; Statement terminator

:= Assignment operator

=> Association operator

|| Concatenation operator

\*\* Exponentiation operator

<<,>> Label delimiter

/\*,\*/ Multiple line comment indictor

-- Single line comment indicator

.. Range Indicator

<,> Relational operators

<>,'=,~=,^= Different versions of not equal

Data Types

PL/SQL variables, constants and parameters must have a valid data type which specifies a storage format, constraints and valid range of values

Scalar Types

1. Numeric : Numeric values on which arithematic operation are performed.

2. Character : Alphanumeric values that represent single characters or strings of characters.

3. Boolean : Logical values on which logical operations are performed.

4. Datetime : Dates and times.

5. Large object : Large object (LOB) data types refer large to data items such as text, graphic images, video clips and sound waveforms.

6. User-Defined Subtypes : A subtype is a subset of another data type, which is called its base type. A subtype has the same valid operations as its base type, but only a subset of its valid values.

Numeric data types

NUMBER(prec, scale) : Fixed-point or floating-point number with absolute value in range 1E-130 to (but not including) 1.0E126. A NUMBER variable can also represent 0.

INT : ANSI specific integer type with maximum precision of 38 decimal digits

INTEGER : ANSI and IBM specific integer type with maximum precision of 38 decimal digits

FLOAT : ANSI and IBM specific floating-point type with maximum precision of 126 binary digits (approximately 38 decimal digits)

NUMERIC(pre, scale) : Floating type with maximum precision of 38 decimal digits.

DEC(prec, scale) : ANSI specific fixed-point type with maximum precision of 38 decimal digits.

DECIMAL(prec, scale) : IBM specific fixed-point type with maximum precision of 38 decimal digits.

DOUBLE PRECISION : ANSI specific floating-point type with maximum precision of 126 binary digits (approximately 38 decimal digits)

REAL : Floating-point type with maximum precision of 63 binary digits (approximately 18 decimal digits)

PLS\_INTEGER : Signed integer in range -2,147,483,648 through 2,147,483,647,represented in 32 bits

BINARY\_INTEGER : Signed integer in range -2,147,483,648 through 2,147,483,647,represented in 32 bits

BINARY\_FLOAT : Double-precision IEEE 754-format floating-point number

Example :

DECLARE

num1 INTEGER;

num2 REAL;

num3 DOUBLE PRECISION;

BEGIN

null;

END;

Character Data Types and Subtypes :

CHAR : Fixed-length character string with maximum size of 32,767 bytes

VARCHAR2 : Variable-length character string with maximum size of 32,767 bytes

RAW : Variable-length binary or byte string with maximum size of 32,767 bytes, not interpreted by PL/SQL

NCHAR : Fixed-length national character string with maximum size of 32,767 bytes

NVARCHAR2 : Variable-length national character string with maximum size of 32,767 bytes

LONG : Variable-length character string with maximum size of 32,760 bytes

LONG RAW : Variable-length binary or byte string with maximum size of 32,760 bytes, not interpreted by PL/SQL

ROWID : Physical row identifier, the address of a row in an ordinary table

UROWID : Universal row identifier (physical, logical, or foreign row identifier)

Boolean data types

BOOLEAN : The logical values are the Boolean values TRUE and FALSE and the value NULL.

Date and Time data types :

Date : which include the time of day in seconds since midnight. the default might be 'DD-MON-YY'

Large Object (LOB) Data Types

BFILE : Used to store large binary objects in operating system files outside the database.

BLOB : Used to store large binary objects in the database.

CLOB : Used to store large blocks of character data in the database.

NCLOB : Used to store large blocks of NCHAR data in the database.

User defined data type

Syntax :

SUBTYPE User\_datatype\_name IS datatype;

Variable User\_datatype\_name;

NULLs in PL/SQL

NULL values represent missing or unknown data and they are not an integer, a character, or any other specific data type.

Variable declarations

PL/SQL variables must be declared in the declaration section or in a package as a global variable.

Syntax :

variable\_name [CONSTANT] datatype [NOT NULL] [:= | DEFAULT initial\_value];

example :

sales number(10, 2);

pi CONSTANT double precision := 3.1415;

name varchar2(25);

address varchar2(100);

sales number(10, 2);

name varchar2(25);

address varchar2(100);

Default Keyword : Has a default value stored

greetings varchar2(20) DEFAULT 'Have a Good Day';