**Introduction to PL/SQL**

PL/SQL is a combination of SQL along with the procedural features of programming languages. It was

developed by Oracle Corporation in the early 90's to enhance the capabilities of SQL. PL/SQL is one of three key programming languages embedded in the Oracle Database, along with SQL itself and Java.

* PL/SQL is a completely portable, high-performance transaction-processing language.
* PL/SQL provides a built-in, interpreted and OS independent programming environment.
* PL/SQL can also directly be called from the command-line SQL\*Plus interface.
* Direct call can also be made from external programming language calls to database.
* PL/SQL's general syntax is based on that of ADA and Pascal programming language.

**Features of PL/SQL**

PL/SQL has the following features

* PL/SQL is tightly integrated with SQL.
* It offers extensive error checking.
* It offers numerous data types.
* It offers a variety of programming structures.
* It supports structured programming through functions and procedures.
* It supports object-oriented programming.
* It supports the development of web applications and server pages.

**Advantages of PL/SQL**

PL/SQL has the following advantages −

* SQL is the standard database language and PL/SQL is strongly integrated with SQL. PL/SQL supports both static and dynamic SQL. Static SQL supports DML operations and transaction control from PL/SQL block. In Dynamic SQL, SQL allows embedding DDL statements in PL/SQL blocks.
* PL/SQL allows sending an entire block of statements to the database at one time. This reduces network traffic and provides high performance for the applications.
* PL/SQL gives high productivity to programmers as it can query, transform, and update data in a database.
* PL/SQL saves time on design and debugging by strong features, such as exception handling, encapsulation, data hiding, and object-oriented data types.
* Applications written in PL/SQL are fully portable.
* PL/SQL provides high security level.
* PL/SQL provides access to predefined SQL packages.
* PL/SQL provides support for Object-Oriented Programming.
* PL/SQL provides support for developing Web Applications and Server Pages.

**PL/SQL-Basic Syntax**

PL/SQL programs are divided and written in logical blocks of code. Each block consists of three sub-parts

**Declarations**

This section starts with the keyword DECLARE. It is an optional section and defines all variables,

cursors, subprograms, 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 optional section contains exception(s) that

handle errors in the program.

**DECLARE**

**<declarations section>**

**BEGIN**

**<executable command(s)>**

**EXCEPTION**

**<exception handling>**

**END;**

**The 'Hello World' Example**

**DECLARE**

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

**BEGIN**

**dbms\_output.put\_line(message);**

**END;**

/

**The PL/SQL Identifiers**

PL/SQL identifiers are constants, variables, exceptions, procedures, cursors, and reserved words. The identifiers consist of a letter optionally followed by more letters, numerals, dollar signs, underscores, and number signs and should not exceed 30 characters.

**The PL/SQL Comments**

The PL/SQL supports single-line and multi-line comments. All characters available inside any comment are ignored by the PL/SQL compiler. The PL/SQL single-line comments start with the delimiter -- (double hyphen) and multi-line comments are enclosed by /\* and \*/.

**DECLARE**

**-- variable declaration**

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

**BEGIN**

**/\***

**\* PL/SQL executable statement(s)**

**\*/**

**dbms\_output.put\_line(message);**

**END;**

**/**

When the above code is executed at the SQL prompt, it produces the following result − Hello World

PL/SQL procedure successfully completed.

**PL/SQL- Data Types**

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

* Scalar

Single values with no internal components, such as a NUMBER, DATE, or BOOLEAN.

* Large Object (LOB)

Pointers to large objects that are stored separately from other data items, such as text, graphic images, video clips, and sound waveforms.

* Composite

Data items that have internal components that can be accessed individually. For example, collections and records.

* Reference
* Pointers to other data items.
* PL/SQL Scalar Data Types and Subtypes
* PL/SQL Numeric Data Types and Subtypes
* PL/SQL Character Data Types and Subtypes
* PL/SQL Boolean Data Types
* PL/SQL Datetime and Interval Types