PL/SQL BLOCK

Advantages

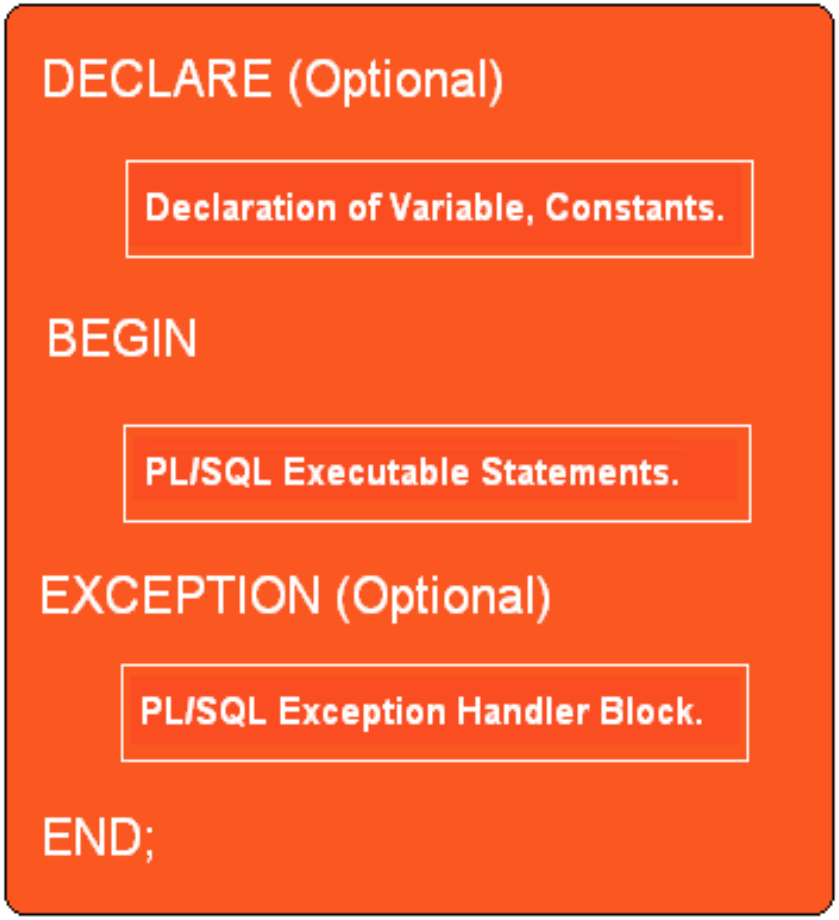
- **Procedural Language Supported:** PL/SQL is a development tools that not only supported data Manipulation but also Provide the Condition, Checking, Looping or Branching Operation.
- **Reduces Network Traffic:** PL/SQL is same entire block of SQL statement execute to the oracle engine at all at once so it's benefit to reduce the Network Traffic.
- **Error Handling:** PL/SQL also permit during with Error Handling as required facility to Display User Friendly Error Message where error are encounter.
- **Declare Variable:** PL/SQL allow to declaration and use of variable in a block of code which variable will use to store intermediate result of query for later processing.
- **Intermediate Calculation:** PL/SQL calculations done quickly and efficient without the use of oracle engines and improve the transaction.
- **Portable Application:** Application are written in PL/SQL is portable in any computer or hardware for any system means Application independence to run any computer.

Basic Structure

- **DECLARE:** Variable and constants are declared within this section and we may initialize them with value.

- **BEGIN:** It contains the PL/SQL statements which implement the actual programming logic. This section contains conditional statements (IF..ELSE), looping statements (FOR, WHILE) and Branching Statements (GOTO) etc.

- **EXCEPTION:** Exception block handling the error and show the user friendly message. Error can arise due to syntax, logical or validation rules.



PL/SQL support Advance Data Types

- %Type
  - This data type is use to store value unknown data type column in a table. column is identified by %type data type.
    Eg. emp.eno%type
    emp name is table,
    eno is a unknown data type column and
    %Type is data type to hold the value.
- %RowType
  - This data type is use to store value unknown data type all column in a table. All column is identified by %RowType datatype.
    Eg. emp%rowtype
    emp name is table,
    all column type is %rowtype.

Declaring Variables

- Variables can have any SQL datatype, such as CHAR, DATE, or NUMBER, or a PL/SQL-only datatype, such as BOOLEAN or

- ***Example Declaring Variables in PL/SQL***

```
DECLARE
  part_no    NUMBER(6);
  part_name  VARCHAR2(20);
  in_stock   BOOLEAN;
```

- **Assigning Values to a Variable**

first way uses the assignment operator (:=), a colon followed by an equal sign.

SQL> set serveroutput on

SQL>

```
DECLARE
      eno number(5) NOT NULL := 2
       ename varchar2(15) := Jay Patel;
        edept CONSTANT varchar2(15) := Web Developer;
BEGIN
    dbms_output.put_line('Declared Value:');
    dbms_output.put_line(' Eno is: ' || eno || ' Ename is : ' || ename);
    dbms_output.put_line(' Edept is: ' || edept);
END;
```

Declared Value:

Eno is : 2 Ename is : Jay Patel

EDept: Web Developer

Assigning Values to Variables by SELECTing INTO variable

```
DECLARE
    bonus  NUMBER(8,2);
    emp_id NUMBER(6) := 100;
BEGIN
    SELECT salary * 0.10 INTO bonus FROM employees
   WHERE employee_id = emp_id;
END;
```

```
declare
    salary number(10,2);
begin
     select avg(sal) into salary from emp;
     DBMS_OUtput.put_line(salary);
End;
```