Pl-sql

25 January 2023 12:34

What is PL-SQL?

PL/SQL (Procedural Language/Structured Query Language) is a high-level programming language designed specifically for the Oracle Database Management System. It is a procedural language that provides a way to manage data stored in Oracle databases and provides features like control structures, procedures, functions, and triggers to interact with this data. It supports most SQL commands and data types and offers robust error handling and exception management capabilities.

Describe PL-SQL sections

PL/SQL is divided into several sections, each serving a specific purpose:

1. Declaration Section: This section is used to declare variables, cursors, and other PL/SQL constructs that will be used in the program.
2. Execution Section: This section contains the actual PL/SQL code that performs the logic and processes the data.
3. Exception Section: This section contains exception handlers that manage errors and exceptions that may occur during the execution of the program.
4. Anonymous Block: An anonymous block is a PL/SQL program unit that is not stored in the database and is executed only once. It is often used to test PL/SQL code snippets.
5. Stored Procedure: A stored procedure is a named PL/SQL block that is stored in the database and can be executed multiple times by multiple users.
6. Function: A function is a named PL/SQL block that returns a single value. It can be called from other PL/SQL blocks or used in SQL statements.
7. Package: A package is a collection of related PL/SQL constructs, such as procedures, functions, and variables, that can be grouped together and stored in the database for reuse.

Example of PL-SQL command

Here is an example of a simple PL/SQL code that declares a variable, assigns a value to it, and displays the value:

```
DECLARE
   v_name VARCHAR2(20);
BEGIN
   v_name := 'John Doe';
   DBMS_OUTPUT.PUT_LINE('Hello, ' || v_name || '!');
END;
```

This code declares a variable called v_name with a data type of VARCHAR2 and a maximum length of 20 characters. The value 'John Doe' is assigned to the variable in the execution section. Finally, the DBMS_OUTPUT.PUT_LINE function is used to display the value of v_name, along with the string 'Hello,

' and '!'.

```
Hello, John Doe!
```

Variables in PL-SQL

In PL/SQL, variables are used to store values that can be used and manipulated within PL/SQL programs. Variables must be declared before they can be used, and they must be assigned a data type that determines what kind of values they can store. Some common PL/SQL data types include:

- VARCHAR2: variable-length character string
- NUMBER: numeric value
- DATE: date and time value
- BOOLEAN: logical value (TRUE or FALSE)

Assigning SQL query results to PL_SQL variables

In PL/SQL, you can use the SELECT INTO statement to assign the results of a SQL query to PL/SQL variables. The SELECT INTO statement retrieves one or more rows of data from a database and assigns the values to one or more PL/SQL variables.
Here's an example of using SELECT INTO to assign the result of a query to a single PL/SQL variable:

```
DECLARE
   v_name VARCHAR2(20);
BEGIN
   SELECT name INTO v_name
   FROM employees
   WHERE employee_id = 123;
   DBMS_OUTPUT.PUT_LINE('Employee Name: ' || v_name);
END;
```

And here's an example of using SELECT INTO to assign the results of a query to multiple PL/SQL variables:

```
DECLARE
   v_name VARCHAR2(20);
   v_salary NUMBER;
BEGIN
   SELECT name, salary INTO v_name, v_salary
   FROM employees
   WHERE employee_id = 123;
   DBMS_OUTPUT.PUT_LINE('Employee Name: ' || v_name);
   DBMS_OUTPUT.PUT_LINE('Employee Salary: ' || v_salary);
END;
```

Note that the SELECT INTO statement should only retrieve a single row, otherwise it will raise the

TOO_MANY_ROWS exception. You can use the IF-ELSE statement or the BULK COLLECT INTO clause to handle this exception.