PL SQL basically **stands** for "Procedural Language extensions to SQL". This is the extension of Structured Query Language (SQL) that is used in Oracle. Unlike SQL, PL/SQL allows the programmer to write code in procedural format.

It combines the data manipulation power of SQL with the processing power of procedural language to create a super powerful SQL queries.

It allows the programmers to instruct the compiler 'what to do' through SQL and 'how to do' through its procedural way.

Similar to other database languages, it gives more control to the programmers by the use of loops, conditions and object oriented concepts.

In this section, we will discuss some differences between SQL and PL/SQL

|  |  |
| --- | --- |
| **SQL** | **PL/SQL** |
| * SQL is a single query that is used to perform DML and DDL operations. | * PL/SQL is a block of codes that used to write the entire program blocks/ procedure/ function, etc. |
| * It is declarative, that defines what needs to be done, rather than how things need to be done. | * PL/SQL is procedural that defines how the things needs to be done. |
| * Execute as a single statement. | * Execute as a whole block. |
| * Mainly used to manipulate data. | * Mainly used to create an application. |
| * Interaction with Database server. | * No interaction with the database server. |
| * Cannot contain PL/SQL code in it. | * It is an extension of SQL, so it can contain SQL inside it. |

**What are Cursors?**

A cursor is a temporary work area created in the system memory when a SQL statement is executed. A cursor contains information on a select statement and the rows of data accessed by it.

This temporary work area is used to store the data retrieved from the database, and manipulate this data. A cursor can hold more than one row, but can process only one row at a time. The set of rows the cursor holds is called the *active*set.

There are two types of cursors in PL/SQL:

When you execute DML statements like DELETE, INSERT, UPDATE and SELECT statements, implicit statements are created to process these statements.

For Example: Consider the PL/SQL Block that uses implicit cursor attributes as shown below:

*DECLARE var\_rows number(5);*

*BEGIN*

*UPDATE employee*

*SET salary = salary + 1000;*

*IF SQL%NOTFOUND THEN*

*dbms\_output.put\_line('None of the salaries where updated');*

*ELSIF SQL%FOUND THEN*

*var\_rows := SQL%ROWCOUNT;*

*dbms\_output.put\_line('Salaries for ' || var\_rows || 'employees are updated');*

*END IF;*

*END;*

**What is a Trigger?**

A trigger is a pl/sql block structure which is fired when a DML statements like Insert, Delete, Update is executed on a database table. A trigger is triggered automatically when an associated DML statement is executed.

**For Example:** The price of a product changes constantly. It is important to maintain the history of the prices of the products.

We can create a trigger to update the 'product\_price\_history' table when the price of the product is updated in the 'product' table.

**1)** Create the 'product' table and 'product\_price\_history' table

*CREATE TABLE product\_price\_history*

*(product\_id number(5),*

*product\_name varchar2(32),*

*supplier\_name varchar2(32),*

*unit\_price number(7,2) );*

*CREATE TABLE product*

*(product\_id number(5),*

*product\_name varchar2(32),*

*supplier\_name varchar2(32),*

*unit\_price number(7,2) );*

**2)** Create the price\_history\_trigger and execute it.

*CREATE or REPLACE TRIGGER price\_history\_trigger*

*BEFORE UPDATE OF unit\_price*

*ON product*

*FOR EACH ROW*

*BEGIN*

*INSERT INTO product\_price\_history*

*VALUES*

*(:old.product\_id,*

*:old.product\_name,*

*:old.supplier\_name,*

*:old.unit\_price);*

*END;*

*/*

**3)** Lets update the price of a product.

*UPDATE PRODUCT SET unit\_price = 800 WHERE product\_id = 100*

Once the above update query is executed, the trigger fires and updates the 'product\_price\_history' table.

Simple Loop

A Simple Loop is used when a set of statements is to be executed at least once before the loop terminates.

*LOOP*

*statements;*

*EXIT;*

*{or EXIT WHEN condition;}*

*END LOOP;*

While Loop

A WHILE LOOP is used when a set of statements has to be executed as long as a condition is true.

*WHILE <condition>*

*LOOP statements;*

*END LOOP;*

### FOR Loop

*FOR counter IN val1..val2*

*LOOP statements;*

*END LOOP;*