CURSOR MANAGEMENT in PL/SQL

A Cursor is a private SQL work area that Oracle uses to execute SQL statements and store information. Whenever you issue a SQL statement, the Oracle server opens an area of memory in which the command is parsed and executed. **This area is called a Cursor** . 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 one row at a time. **The set of rows the cursor holds is called the Active Set**.

TYPES OF CURSOR

There are two types of Cursors in PL/SQL:-

(1) Implicit Cursor
(2) Explicit Cursor

IMPLICIT CURSOR :- It is declared by PL/SQL implicitly when DML statements like INSERT, UPDATE and DELETE statements are executed. They are also declared when a SELECT statement that returns just one row is executed.

When the executable part of a PL/SQL block issues a SQL statement, PL/SQL creates an implicit cursor, which PL/SQL manages automatically.

IMPLICIT CURSOR ATTRIBUTES :- Oracle provides few attributes for implicit cursor called as implicit cursor attributes to check the status of DML operations.

These attributes are :-

(1) %FOUND
(2) %NOTFOUND
(3) %ROWCOUNT
(4) %ISOPEN

The status of the cursor for each of these attributes is defined in the below table :-

Attribute	Return Value	Example
%FOUND	The return value is TRUE, if the DML statements like INSERT, DELETE and UPDATE affect at least one row and if SELECT...INTO statement return at least one row. The return value is FALSE, if the DML statements like	SQL%FOUND

	INSERT, DELETE and UPDATE do not affect any row and if SELECT …. INTO statement does not return a row.	
%NOTFOUND	The return value is FALSE, if the DML statements like INSERT, DELETE and UPDATE affect at least one row and if SELECT … INTO statement return at least one row. The return value is TRUE, if the DML statements like INSERT, DELETE and UPDATE do not affect any row and if SELECT ….. INTO statement does not return a row.	SQL%NOT FOUND
%ROWCOUNT	Return the number of rows affected by the DML operations INSERT, DELETE , UPDATE , or single row SELECT.	SQL%ROWCOUNT
%ISOPEN	THe return value is always FALSE because Oracle automatically closes an Implicit Cursor after executing its SQL statement.	SQL%ISOPEN

```
declare
  var_rows number(5);
BEGIN
  UPDATE BCA
  SET ID = 9;
  IF SQL%NOTFOUND THEN
     dbms_output.put_line('None of the ID were updated');
```

```
   ELsIF SQL%FOUND THEN
var_rows:=SQL%ROWCOUNT;
dbms_output.put_line (var_rows ||   'REcords are updated.');
 end if;
 end;
 /
```

EXPLICIT CURSOR

Explicit Cursors are declared explicitly by the user, along with other identifiers to be used in a PL/SQL block. These are also known as user-defined cursors, defined in the DECLARE section of the PL/SQL block. They must be declared when you are executing a SELECT statement that returns more than one row. Even though the cursor stores ,ultiple records, only one record can be processed at a time, which is called as **Current row.**

Cursor manipulation :-

It is done through the DECLARE, OPEN, FETCH and CLOSE statements. These statements are explained below :-

(1) Declaring an Explicit Cursor :- While declaring a cursor, cursor name is given and also defines the structure of the query to be performed within it. The CURSOR statement is used to declare explicit cursor.

Syntax :-

CURSOR <cursor_name> IS <select_statement>;

In the syntax :-

(a) cursor_name :- A suitable name for the cursor

(b) select_statement :- A select query which returns multiple rows.

© <select_statement> includes most of the usual clauses, but INTO clause is not allowed. Also NULL can not be defined as SELECTed item.

(2) Opening an Explicit cursor :- Once the cursor is declared in the declaration section, we can access the cursor in the execution section of the PL/SQL program. The OPEN statement is used to open an explicit cursor.

Syntax :-

OPEN <Cursor_name>;

(3) Retrieving Individual row/Fetching data from the Explicit Cursor :- After the cursor is opened, the current row is loaded into variables. The current row is the row at

which the cursor is currently pointing. The transfer of data into PL/SQL variables is done through FETCH statement.

Syntax :-

FETCH <cursor_name> INTO <variables>;

For each column value returned by the query associated with the cursor, there must be a corresponding variable in the INTO list. Also their data types must be compatible .

EXAMPLE :-

```
declare
  stu_id bca.id%type;
 stu_name bca.name%type;
 stu_gender bca.gender%type;
cursor c1 IS Select id,name,gender from bca where id=6;
BEGIN
  OPEN C1;
LOOP
  FETCH C1 INTO stu_id, stu_name,stu_gender;
EXIT WHEN c1%NOTFOUND;
dbms_output.put_line(stu_id||' '||stu_name ||' '||stu_gender);
 END LOOP;
 END;
 /
```

(4) Closing an Explicit Cursor :- Close the cursor after completing the processing of the SELECT statement. This step allows the cursor to be reopened, if required. Therefore, you can establish an active set several times. The CLOSE statement is used to close an explicit cursor.

Syntax :-

CLOSE <cursor_name>;

EXPLICIT CURSOR ATTRIBUTES :

ORACLE provides some attributes known as **Explicit Cursor Attributes** to control the data processing while using cursors. We use these attributes to avoid errors while accessing cursors through OPEN, FETCH and CLOSE statements.
These are the attributes available to check the status of an explicit cursor.

Attribute	Return Value	Example
%FOUND	TRUE, if fetch statement return at least one row. FALSE, if fetch statement does not return a row.	Cursor_name%FOUND
%NOTFOUND	TRUE, if each fetch statement doesn't return a row. FALSE, if each fetch statement returns at least one row.	Cursor_name%NOTFOUND
%ROWCOUNT	Returns the number of rows fetched by the fetch statement If no row is returned, the PL/SQL statement returns an error	Cursor_name%ROWCOUNT
%ISOPEN	TRUE if the cursor is already open in the program FALSE, if the cursor is not opened in the program.	Cursor_name%ISOPEN