CURSORS

Cursors in PL/SQL are database objects used to retrieve and manipulate data from the result set of a SELECT query. Cursors are essential when you want to work with multiple rows of data in your PL/SQL program. They come in two main types: implicit and explicit.

1. Implicit Cursors:

Implicit cursors are automatically managed by the PL/SQL engine. They are used for single-row queries and are not explicitly declared. There are two common types of implicit cursors:

- **SQL%FOUND:** Returns TRUE if a DML (Data Manipulation Language) statement like INSERT, UPDATE, DELETE, or a SELECT INTO statement retrieves one or more rows.

- **SQL%NOTFOUND:** Returns TRUE if a DML statement does not retrieve any rows.

- **SQL%ROWCOUNT:** Returns the number of rows affected by a DML statement.

Example of Implicit Cursors:

```sql
DECLARE
  emp_salary NUMBER;
BEGIN
  SELECT salary INTO emp_salary FROM employees WHERE employee_id = 101;

  IF SQL%FOUND THEN
    DBMS_OUTPUT.PUT_LINE('Employee found.');
  ELSE
    DBMS_OUTPUT.PUT_LINE('Employee not found.');
  END IF;
```

```
    DBMS_OUTPUT.PUT_LINE('Rows affected: ' || SQL%ROWCOUNT);
END;
/
```

In this example, the implicit cursor is used to fetch data from the "employees" table.

2. Explicit Cursors:

Explicit cursors are user-defined cursors explicitly declared, opened, fetched, and closed by the programmer. They are used for more complex queries where you need to process multiple rows.

Syntax for Declaring and Using Explicit Cursors:

```sql
DECLARE
    CURSOR cursor_name IS
        SELECT column1, column2 FROM table_name WHERE condition;

    variable1 datatype;
    variable2 datatype;
BEGIN
    OPEN cursor_name;
    LOOP
        FETCH cursor_name INTO variable1, variable2;
        EXIT WHEN cursor_name%NOTFOUND;
        -- Process the data
    END LOOP;
    CLOSE cursor_name;
```

```
END;
/
```

- `cursor_name`: User-defined name for the cursor.
- `variable1`, `variable2`: Variables to hold the column values.
- `OPEN`: Opens the cursor.
- `FETCH`: Retrieves data from the cursor into variables.
- `EXIT`: Used to exit the loop when there are no more rows to fetch.
- `CLOSE`: Closes the cursor.

Cursor Attributes:

Cursor attributes are used to manipulate and retrieve information about the state of a cursor. Common cursor attributes include:

- `%FOUND`: Returns `TRUE` if the last `FETCH` statement retrieved a row.
- `%NOTFOUND`: Returns `TRUE` if the last `FETCH` statement did not retrieve a row.
- `%ROWCOUNT`: Returns the number of rows fetched so far.
- `%ISOPEN`: Returns `TRUE` if the cursor is open, otherwise `FALSE`.

Example of Explicit Cursor:

```sql
DECLARE
  CURSOR emp_cursor IS
    SELECT employee_id, first_name, last_name FROM employees WHERE department_id = 20;
  emp_id employees.employee_id%TYPE;
  emp_first_name employees.first_name%TYPE;
```

```
    emp_last_name employees.last_name%TYPE;
BEGIN
    OPEN emp_cursor;
    LOOP
        FETCH emp_cursor INTO emp_id, emp_first_name, emp_last_name;
        EXIT WHEN emp_cursor%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('Employee ID: ' || emp_id || ', Name: ' || emp_first_name ||
' ' || emp_last_name);
    END LOOP;
    CLOSE emp_cursor;
END;
/
```

In this example, we declare an explicit cursor (`emp_cursor`) to fetch and process employee data from the "employees" table for a specific department.

Explicit cursors give you fine-grained control over querying and processing data in your PL/SQL programs, making them a powerful tool for working with result sets.