Assignment No A7

Title: Cursors:(All types: Implicit, Explicit, Cursor FOR Loop, Parameterized Cursor)
Write a PL/SQL block of code using parameterized Cursor that will merge the data available in the newly created table N_Roll Call with the data available in the table O_RollCall. If the data in the first table already exist in the second table then that data should be skipped.

Objective: Learning the cursor and its type, how to define cursors on database to operate and manage the database.

Theory:

Definition :

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.

Explanation :

When an SQL statement is processed, Oracle creates a memory area known as context area. A cursor is a pointer to this context area. It contains all information needed for processing the statement. In PL/SQL, the context area is controlled by Cursor. A cursor contains information on a select statement and the rows of data accessed by it.

A cursor is used to referred to a program to fetch and process the rows returned by the SQL statement, one at a time.

Types of Cursor :

There are two types of cursors:

1] Implicit Cursors

2] Explicit Cursors

1. Implicit Cursors :

Implicit cursors are automatically created by Oracle whenever an SQL statement is executed, when there is no explicit cursor for the statement. Programmers cannot control the implicit cursors and the information in it.

Whenever a DML statement (INSERT, UPDATE and DELETE) is issued, an implicit cursor is associated with this statement. For INSERT operations, the cursor holds the data that needs to be inserted. For UPDATE and DELETE operations, the cursor identifies the rows that would be affected.

In PL/SQL, you can refer to the most recent implicit cursor as the SQL cursor, which always has attributes such as **%FOUND, %ISOPEN, %NOTFOUND, and %ROWCOUNT.** The SQL cursor has additional attributes, **%BULK_ROWCOUNT and %BULK_EXCEPTIONS**, designed

for use with the FORALL statement. The following table provides the description of the most used attributes −

S.No	Attribute & Description
1	**%FOUND** Returns TRUE if an INSERT, UPDATE, or DELETE statement affected one or more rows or a SELECT INTO statement returned one or more rows. Otherwise, it returns FALSE**.**
2	**%NOTFOUND** The logical opposite of %FOUND. It returns TRUE if an INSERT, UPDATE, or DELETE statement affected no rows, or a SELECT INTO statement returned no rows. Otherwise, it returns FALSE.
3	**%ISOPEN** Always returns FALSE for implicit cursors, because Oracle closes the SQL cursor automatically after executing its associated SQL statement.
4	**%ROWCOUNT** Returns the number of rows affected by an INSERT, UPDATE, or DELETE statement, or returned by a SELECT INTO statement.

NOTE: Any SQL cursor attribute will be accessed as sql%attribute_name

2] Explicit Cursors:

Explicit cursors are programmer-defined cursors for gaining more control over the context area. An explicit cursor should be defined in the declaration section of the PL/SQL Block. It is created on a SELECT Statement which returns more than one row.

The syntax for creating an explicit cursor is −

```
CURSOR cursor_name IS select_statement;
```

Working with an explicit cursor includes the following steps −

- Declaring the cursor for initializing the memory
- Opening the cursor for allocating the memory
- Fetching the cursor for retrieving the data
- Closing the cursor to release the allocated memory

- **Declaring the Cursor :**

Declaring the cursor defines the cursor with a name and the associated SELECT statement.

Syntax for explicit cursor decleration :

1. CURSOR name IS
2. SELECT statement;

- **Opening the Cursor :**

Opening the cursor allocates the memory for the cursor and makes it ready for fetching the rows returned by the SQL statement into it.

Syntax for cursor open :

1. OPEN cursor_name;

- **Fetching the Cursor :**

Fetching the cursor involves accessing one row at a time.

Syntax for cursor fetch:

FETCH cursor_name INTO variable_list;

- **Closing the Cursor :**

Closing the cursor means releasing the allocated memory.

Syntax for cursor close :

Close cursor_name;

Advantages of Cursor :

1. Using cursor to getting multiple values.
2. Where Current of Clause: this is use full when the primary key is not present.
3. Cursors can be faster than a while loop but they do have more overhead.
4. we can do RowWise validation or in other way you can perform operation on each Row.

Conclusion: Here we understood the concept of cursor and its types and how to define and use of cursors on database table to read and fetch the data.