

PL/SQL– CURSOR

BY– N K SHUKLA

PL/SQL Cursor

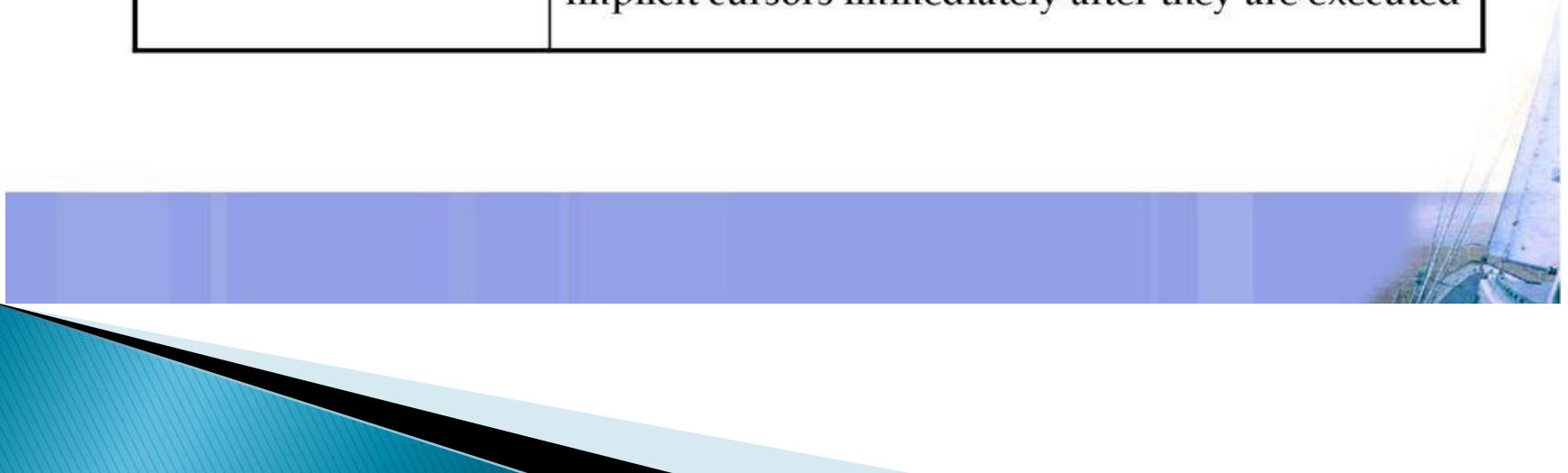
- ▶ A cursor is a pointer that points to a result of a query.
- ▶ A cursor is used to referred to a program to fetch and process the rows returned by the SQL statement, one at a time. There are two types of cursors:
 - Implicit Cursors
 - Explicit Cursors
- ▶ A cursor holds the rows (one or more) returned by a SQL statement. The set of rows the cursor holds is referred to as the **active set**.

Implicit Cursors

- ▶ Implicit cursors are automatically created by Oracle whenever an SQL statement is executed.
- ▶ Programmers cannot control the implicit cursors and the information in it.
- ▶ Oracle opens implicit cursor to process SQL statements such as SELECT, INSERT, UPDATE, DELETE.
- ▶ 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.
- ▶ Syntax of attributes:
SQL.AttributeName;

Implicit Cursor Attributes

Attribute	Description
SQL%ROWCOUNT	Number of rows affected by the most recent SQL statement (an integer value)
SQL%FOUND	Boolean attribute that evaluates to TRUE if the most recent SQL statement affects one or more rows
SQL%NOTFOUND	Boolean attribute that evaluates to TRUE if the most recent SQL statement does not affect any rows
SQL%ISOPEN	Always evaluates to FALSE because PL/SQL closes implicit cursors immediately after they are executed



Example: Get bname from user if it is in branch table then convert into uppercase else give message branch not available.

```
DECLARE
    BN BRANCH.BNAME%TYPE;
BEGIN
    BN:=:BN;
    UPDATE BRANCH SET BNAME=UPPER(BN) WHERE
    BNAME=BN;
    IF SQL%FOUND THEN
        DBMS_OUTPUT.PUT_LINE('TOTAL '||
        SQL%ROWCOUNT ||' ROWS UPDATED!!');
    ELSE
        DBMS_OUTPUT.PUT_LINE('GIVEN BRANCH NOT
        FOUND');
    END IF;
END;
```

Explicit Cursors

- ▶ **“A cursor is called explicit cursor if it is opened by user to process data through PL/SQL block.”**
- ▶ Explicit cursor will be used when there is a need to process more than one record individually.
- ▶ As it is opened by user so user has to take care of managing cursor.
- ▶ **STEPS:**
 1. Declare a Cursor
 2. Open a cursor
 3. Fetch data
 4. Process data
 5. Closing a cursor

1.Declare a cursor:

- ▶ Syntax:

CURSOR cursor_name IS SELECT...;

- ▶ It is mapped to the query given by the select statement.
- ▶ Declaration informs oracle that cursor_name named cursor will be used later.
- ▶ Eg.

CURSOR cAcc IS SELECT ano,balance,banme from account;



2.Open Cursor:

▶ Syntax:

OPEN cursor_name ;

- ▶ A cursor can be opened using above syntax.
- ▶ While opening cursor following operations are performed.
 - Allocate memory required to store data.
 - Execute the SELECT statement associated with the cursor.
 - Create active data set by retrieving data from tables and populating allocated memory with these data.
 - Set cursor row pointer to point first record in an active data set.
- ▶ Ex:– **OPEN cAcc;**

3.Fetching Data:

- ▶ Syntax:

FETCH curosr_name INTO var1,var2,...;

- ▶ Retrieve data from the current row in a active data set and store them in a given variables.
- ▶ Data from single row fetched at a time.
- ▶ After fetching data update pointer to point next row in a active dataset.
- ▶ Variable should be compactable with SELECT statement.
- ▶ Fetch statement always enclosed within loop.
- ▶ Ex:– **FETCH cAcc INTO no,bal, branch;**

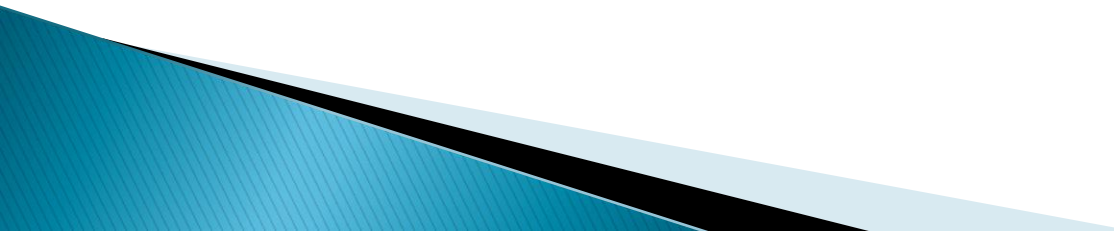
4.Processing Data:

- ▶ This step involves actual processing of table data.
- ▶ Data already fetch in variables which can be processed accordingly.

5.Closing Cursor:

- ▶ Syntax:

CLOSE curosr_name ;

- ▶ Cursor should be closed after processing data.
 - ▶ Closing cursor memory will be release.
 - ▶ Ex:– **CLOSE cAcc INTO ;**
- 

Explicit Cursor Attributes

Obtain status information about a cursor.

Attribute	Type	Description
%ISOPEN	Boolean	Evaluates to TRUE if the cursor is open
%NOTFOUND	Boolean	Evaluates to TRUE if the most recent fetch does not return a row
%FOUND	Boolean	Evaluates to TRUE if the most recent fetch returns a row; complement of %NOTFOUND
%ROWCOUNT	Number	Evaluates to the total number of rows returned so far

Example: Transfer all the account of VVN branch into another table Acc_VVN with only two column acc_no and balance

1. Cursor Declaration

DECLARE

```
CURSOR cACC IS SELECT EID,BALANCE,BNAME FROM ACCOUNT;  
NO ACCOUNT.EID%TYPE;  
BAL ACCOUNT.BALANCE%TYPE;  
BN ACCOUNT.BNAME%TYPE;
```

2. OpenCursor

BEGIN

```
OPEN cACC;  
IF cACC%ISOPEN THEN  
LOOP
```

3. Fetching

```
    FETCH cACC INTO NO,BAL,BN;  
    EXIT WHEN cACC%NOTFOUND;  
    IF BN='VVN' THEN
```

4. Processing

```
        INSERT INTO ACC_VVN VALUES(NO,BAL);  
        DELETE FROM ACCOUNT WHERE EID=NO;
```

5. Closing cursor

```
    END IF;  
END LOOP;  
COMMIT;  
CLOSE cACC;  
ELSE
```

```
    DBMS_OUTPUT.PUT_LINE('CURSOR CANT OPEN');  
END IF;
```

Parameterized Cursor:

- ▶ An explicit cursor may accept a list of parameters. Each time you open the cursor, you can pass different arguments to the cursor, which results in different result sets.
- ▶ The following shows the syntax of a declaring a cursor with parameters:

CURSOR cursor_name (parameter_list) IS SELECT;

- ▶ To open a cursor with parameters, you use the following syntax:

OPEN cursor_name (value_list);



DECLARE

CURSOR cACC(BN ACCOUNT.BNAME%TYPE) IS SELECT
EID,BALANCE,BNAME FROM ACCOUNT WHERE BNAME=BN;

NO ACCOUNT.EID%TYPE;
BAL ACCOUNT.BALANCE%TYPE;
BN ACCOUNT.BNAME%TYPE;

BEGIN

OPEN cACC('KSAD');
IF cACC%ISOPEN THEN

LOOP

FETCH cACC INTO NO,BAL,BN;
EXIT WHEN cACC%NOTFOUND;
INSERT INTO ACC_VVN VALUES(NO,BAL);
DELETE FROM ACCOUNT WHERE EID=NO;
END LOOP;

COMMIT;

ELSE

DBMS_OUTPUT.PUT_LINE('CURSOR CANT OPEN');
END IF;

END;

/