**Practical No 23 and 24: Implement Implicit and Explicit cursors in PL/SQL.**

1. **Practical Related Questions:**
2. Differentiate between Implicit and Explicit cursor.

|  |  |  |
| --- | --- | --- |
| **Sr.**  **No.** | **Implicit Cursors** | **Explicit Cursors** |
| 1 | Implicit cursors are automatically created when select statements are executed. | Explicit cursors needs to be defined explicitly by the user by providing a name. |
| 2 | They are capable of fetching a single row at a time. | Explicit cursors can fetch multiple rows. |
| 3 | They are more vulnerable to errors such as Data errors, etc. | They are less vulnerable to errors (Data errors etc.) |
| 4 | Provides less programmatic control to the users | User/Programmer has the entire control. |
| 5 | Implicit cursors are less efficient. | Comparative to Implicit cursors, explicit cursors are more efficient. |
| 6 | Implicit Cursors are defined as:  BEGIN  SELECT attr\_name from table\_name  where CONDITION;  END | Explicit cursors are defined as:  DECLARE  CURSOR cur\_name IS  SELECT attr\_name from table\_name  where CONDITION;  BEGIN  ... |
| 7 | Implicit cursors requires anonymous buffer memory for storage purpose. | Explicit cursors use user-defined memory space for storage purpose |
| 8 | Cursor attributes use prefix “SQL”. Structure for implicit cursors: SQL%attr\_name Few implicit cursors attributes are: SQL%FOUND, SQL%NOTFOUND, SQL%ROWCOUNT | Structure for explicit cursors: cur\_name%attr\_name Few explicit cursors are: cur\_name%FOUND, cur\_name%NOTFOUND, cur\_name%ROWCOUNT |

1. List attributes of cursor with their meaning.

* %isopen – evaluates to true if cursor is open.
* %NotFound – evaluates to true if the most recent fetch does not return a row.
* %found – evaluates to true if the most recent fetch returns a row complement of notfound.
* %rowcount – evaluates to total number of rows returned so far.

1. Can we use ISOPEN attribute with implicit cursor?

An implicit cursor has attributes that return information about the most recently run SELECT or DML statement that is not associated with a named cursor. Note: You can use cursor attributes only in procedural statements, not in SQL statements. SQL%ISOPEN always has the value FALSE.

1. State advantage and drawback of cursors.

* Cursors can be faster than a while loop but they do have more overhead.
* It is we can do RowWise validation or in other way you can perform operation on each Row. It is a Data Type which is used to define multi-value variable.
* Cursors can be faster than a while loop but at the cost of more overhead.

1. **Exercise:**
2. Execute given sample implicit cursor and note down the output.

Begin

Update Emp\_Info set Name = 'Vaishu' where Salary = 75000;

if sql%found then

dbms\_output.put\_line('Name Updated');

end if;

if sql%notfound then

dbms\_output.put\_line('Not Found');

end if;

if sql%rowcount>0 then

dbms\_output.put\_line(sql%rowcount || 'Rows Updated');

end if;

End;

1. Declare a cursor and select all students in IF course. Also display total no of rows fetched.

Create table Student (Name character(10), Roll\_No number(2), Branch character(6));

Insert all

into Student (Name, Roll\_NO, Branch) values ('Trupti', 17, 'IT')

into Student (Name, Roll\_NO, Branch) values ('Vaishnavi', 20, 'IT')

into Student (Name, Roll\_NO, Branch) values ('Chaitanya', 24, 'IT')

into Student (Name, Roll\_NO, Branch) values ('Aditya', 28, 'IT')

into Student (Name, Roll\_NO, Branch) values ('Priynka', 11, 'CE')

into Student (Name, Roll\_NO, Branch) values ('Sakshi', 12, 'CM')

into Student (Name, Roll\_NO, Branch) values ('Shravani', 25, 'ENTC')

select \* from dual;

Declare

cursor stud\_cursor is select \* from Student where Branch= 'IT';

my\_record Student%rowtype;

Begin

open stud\_cursor;

loop

fetch stud\_cursor into my\_record;

exit when stud\_cursor%notfound;

dbms\_output.put\_line('Roll\_No: ' || my\_record.Roll\_No || ' Name : ' || my\_record.Name || 'Branch: ' || my\_record.Branch);

end loop;

close stud\_cursor;

End;