|  |  |  |
| --- | --- | --- |
| **EX.NO:** | **7** | **Implement Cursors and its Operations** |
| **DATE:** |  |

**Programs:**

1. Create a duplicate table employee1 for employee using select statement. Then create an implicit cursor to you use %FOUND to insert a row if a delete succeeds.

DECLARE

TOTAL\_ROWS NUMBER(2);

BEGIN

DELETE FROM emp WHERE BASIC=30000;

IF SQL%NOTFOUND THEN

DBMS\_OUTPUT.PUT\_LINE('NO ROWS SELECTED');

ELSIF SQL%FOUND THEN

INSERT INTO emp(ENAME,EID,DOB,BASIC) VALUES('PRAVIN','6','01-OCT-88','35000');

DBMS\_OUTPUT.PUT\_LINE('ROW HAS BEEN INSERTED');

END IF;

END;

1. Create an implicit cursor to find the number of rows affected either by INSERT , DELETE, UPDATE command.

declare

total\_rows number(2);

begin

update emp

set basic=basic\*10;

if sql%notfound then

dbms\_output.put\_line('no Employee selected');

elsif sql%found then

total\_rows:=sql%rowcount;

dbms\_output.put\_line('employ selected'||total\_rows);

end if;

end;

/

3.Create a cursor to display multiple rows from the table

declare

employee emp%rowtype;

BEGIN

SELECT \* into employee

FROM Employee WHERE eid =1 ;

dbms\_output.put\_line('Employee ID: ' || employee.eid);

dbms\_output.put\_line('Employee Name: ' || employee.ename);

dbms\_output.put\_line('Employee da: ' || employee.da);

dbms\_output.put\_line('Employee Salary: ' || employee.basic);

END;

/

4.Create a Procedure using explicit cursor to calculate da & total.

Set DA= 25% if salary <15000, DA=35% if salary <=15,000 and >30, 000 DA=45% if Salary>=35,000

declare  
 total\_rows number(3);  
 BEGIN  
     UPDATE EMP  set da=(basic\*0.25) where basic<15000;  
         IF SQL%FOUND THEN  
             UPDATE EMp  set da=(basic\*0.35) where basic >= 15000 and basic <30000;  
              DBMS\_OUTPUT.PUT\_LINE('Row has been inserted');  
         ELSIF SQL%FOUND THEN  
               UPDATE EMP  set da=(basic\*0.45) where basic<=35000;  
     END IF;  
 END;  
/

1. Create an explicit cursor to duplicate employee table into employeenew

declare

ename employee.ename%type;

eid employee.eid%type;

dob employee.dob%type;

basic employee.basic%type;

CURSOR emp is

select ename,eid,dob,basic from employee;

begin

OPEN emp;

LOOP

FETCH emp into ename,eid,dob,basic;

insert into employee(ename,eid,dob,basic)values(ename,eid,dob,basic);

EXIT WHEN emp%notfound;

END LOOP;

CLOSE emp;

end;

/

**Additional Exercise:**

Schema of the table:

stud(name char(10),rno number(4),m1 number(10), m2 number(10),m3 umber(10),tot number(16),avg number(5,2), result char(4));

Create an explicit cursor to handle a SELECT statement that returns multiple rows using EXECUTE IMMEDIATE and BULK COLLECT INTO.

**RESULT**

Thus, the PL/SQL program using the cursors was executed successfully.

|  |  |  |
| --- | --- | --- |
| **MARK ALLOCATION** | | |
| **CONTINUOUS INTERNAL ASSESSMENT** | | |
| **Conduct of Experiment (20)** | Understanding the Problem (5) |  |
| Design the schema (5) |  |
| Writing &Execution of SQL/ Procedure/ Code (10) |  |
| **Record & Result (20)** | |  |
| **Viva (10)** | |  |
| **Total** | **50** |  |
| **Signature of the Faculty with Date** |  | |