

Name : Mayuresh Rajendraraa Deshmukh

Class : TE A

Roll No : TECOA124

Batch : A2

Assignment No. 5

Title: Database Cursor

Implicit Cursor

1) The bank manager has decided to activate all those accounts which were previously marked as inactive for performing no transaction in last 365 days. Write a PL/SQ block (using implicit cursor) to update the status of account, display an approximate message based on the no. of rows affected by the update.

(Use of %FOUND, %NOTFOUND, %ROWCOUNT)

```
SQL> select * from account;
```

ACCNO	S
1	A
2	A
3	D
4	D
5	A
6	D
7	A

7 rows selected.

```
SQL> set serveroutput on
```

```
SQL> begin
```

```
2  update account set status='A' where status='D';
3  if SQL%found then
4      dbms_output.put_line(SQL%ROWCOUNT||' rows affected');
5  elsif SQL%notfound then
6      dbms_output.put_line(SQL%ROWCOUNT||' rows affected');
7  end if;
8  end;
9  /
```

3 rows affected

PL/SQL procedure successfully completed.

```
SQL> select * from account;
```

ACCNO	S
1	A
2	A
3	A

4 A
5 A
6 A
7 A

SQL>/

0 rows affected

PL/SQL procedure successfully completed.

EXPLICIT CURSOR:

**2) Organization has decided to increase the salary of employees by 10% of existing salary, who are having salary less than average salary of organization, Whenever such salary updates takes place, a record for the same is maintained in the increment_salary table. EMP (E_no , Salary)
increment_salary(E_no , Salary)**

SQL> select * from empsal;

ENO	SALARY
1	12100
2	3025
3	121
4	49000
5	50000

SQL> declare

```
2 cursor sal_crsr is select eno,salary from empsal where salary<(select avg(salary) from empsal);
3
4 empno number(10);
5 sal number(10);
6 new_sal number(7,2);
7 begin
8
9 open sal_crsr;
10
11 if sal_crsr %isopen then
12 loop
13     fetch sal_crsr into empno,sal;
14     exit when sal_crsr %notfound;
15     if sal_crsr %found then
16         update empsal set salary=sal+(sal*10)/100 where eno=empno;
17         select salary into new_sal from empsal where eno=empno;
18         insert into inc_sal values(empno,new_sal);
19     end if;
20 end loop;
21 end if;
22 close sal_crsr;
23 end;
24 /
```

PL/SQL procedure successfully completed.

SQL> select * from empsal;

ENO	SALARY
1	13310
2	3327.5
3	133.1
4	49000
5	50000

SQL> select * from inc_sal;

ENO	SALARY
1	13310
2	3327.5
3	133.1

3) Write PL/SQL block using explicit cursor for following requirements:

College has decided to mark all those students detained (D) who are having attendance less than 75%.

Whenever such update takes place, a record for the same is maintained in the D_Stud table. create table stud21(roll number(4), att number(4), status varchar(1));
create table d_stud(roll number(4), att number(4));

SQL> select * from stud24;

ROLL	ATTEND ST
101	90
102	89.23
103	66.23
104	82.56
105	75
106	16

6 rows selected.

SQL> select * from dstud24;

no rows selected

SQL> declare

```

2
3 cursor stud_csr is select roll,attend from stud24 where attend<75;
4 mroll stud24.roll%type;
5 matt stud24.attend%type;
6
7 begin
8
9 open stud_csr;
10
11 if stud_csr %isopen then
12 loop
13   fetch stud_csr into mroll,matt;
14   exit when stud_csr %notfound;
15
```

```

16  if stud_csr % found then
17      update stud24 set status='D' where roll=mroll;
18      insert into dstud24 values(mroll,matt);
19  end if;
20 end loop;
21 end if;
22 end;
23 /

```

PL/SQL procedure successfully completed.

SQL> select * from dstud24;

ROLL	ATT
103	66
106	16

SQL> select * from stud24;

ROLL	ATTEND ST
101	90
102	89.23
103	66.23 D
104	82.56
105	75
106	16 D

6 rows selected.

Parameterized Cursor

4. Write a PL/SQL block of code using parameterized Cursor, that will merge the data available in the newly created table N_RollCall 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.

```
SQL> select * from o_rollcall;
```

ROLL NAME	

101	ashwin
103	rutuj
105	saloni

```
SQL> select * from n_rollcall;
```

ROLL NAME	

101	ashwin
102	hemangi

```
SQL> set serveroutput on
```

```
SQL> declare
```

```
2  cursor allmain is select * from o_rollcall;
3  cursor chk(ckroll number) is select roll from n_rollcall where roll=ckro
ll;
4  nm n_rollcall.name%TYPE;
5  r n_rollcall.roll%TYPE;
6  v n_rollcall.roll%TYPE;
7
8  begin
9  open allmain;
10 loop
11     fetch allmain into r,nm;
12     exit when allmain%NOTFOUND;
13     open chk(r);
14     fetch chk into v;
15     if chk%FOUND then
16         dbms_output.put_line(r||' present');
17     else
18         dbms_output.put_line(r||' not present. Inserting
in n_rollcall');
19         insert into n_rollcall values(r,nm);
20     end if;
21     close chk;
22 end loop;
23 close allmain;
24 end;
25 /
101 present
103 not present. Inserting in n_rollcall
105 not present. Inserting in n_rollcall
```

PL/SQL procedure successfully completed.

```
SQL> select * from o_rollcall;
```

ROLL NAME	

101	ashwin
103	rutuj
105	saloni

```
SQL> select * from n_rollcall;
```

ROLL NAME	

101	ashwin
102	hemangi
103	rutuj
105	saloni

Parameterized Cursor

5) Write the PL/SQL block for following requirements using parameterized Cursor:
Consider table EMP(e_no, d_no, Salary), department wise average salary should be inserted into new table dept_salary(d_no, Avg_salary)

```
SQL> select * from emp24;
```

ENO	DNO	SAL

1	101	30000
2	101	100
3	102	90000
4	102	50000
5	102	70000
6	103	100000
7	103	99000
8	103	80000

8 rows selected.

```
SQL> select * from depavgsal;
```

no rows selected

```
SQL> Declare
```

```
2  cursor csr is select * from emp24;
3  cursor csr1(deptno number) is select avg(sal) from emp24 where dno=deptno;
4  cursor csr2(deptno number) is select avg_sal from depavgsal where dno=deptno;
5  avg emp24.sal%type;
6  v emp24.sal%type;
7  Begin
8  for i in csr
9  loop
10
11      open csr1(i.dno);
12      fetch csr1 into avg;
13      open csr2(i.dno);
```

```

14          fetch csr2 into v;
15          if csr2%notfound then
16              if csr1%found then
17                  insert into depavgsal values(i.d
no,avg);
18              end if;
19          end if;
20          close csr2;
21      close csr1;
22  end loop;
23 End;
24 /

```

PL/SQL procedure successfully completed.

SQL> select * from emp24;

	ENO	DNO	SAL

1	101	30000	
2	101	100	
3	102	90000	
4	102	50000	
5	102	70000	
6	103	100000	
7	103	99000	
8	103	80000	

8 rows selected.

SQL> select * from depavgsal;

	DNO	AVG_SAL

101	15050	
102	70000	
103	93000	

[EXPLICIT CURSOR: Cursor for loop](#)

6. Write PL/SQL block using explicit cursor: Cursor FOR Loop for following requirements:

College has decided to mark all those students detained (D) who are having attendance less than 75%.

Whenever such update takes place, a record for the same is maintained in the D_Stud

table. create table stud21(roll number(4), att number(4), status varchar(1)); create

table d_stud(roll number(4), att number(4));

```
SQL> select * from stud24;
```

ROLL	ATTEND ST
101	90
102	89.23
103	66.23
104	82.56
105	75
106	16

6 rows selected.

```
SQL> select * from dstud24;
```

no rows selected

```
SQL> declare
```

```
2  cursor frstudcrsr is select * from stud24 where attend<75;
3  mroll stud24.roll%type;
4  matt stud24.attend%type;
5  rvar stud24%rowtype;
6  begin
7  for rvar in frstudcrsr
8  loop
9      update stud24 set status='D' where roll=rvar.roll;
10
11      insert into dstud24 values(rvar.roll,rvar.attend);
12  end loop;
13 end;
14 /
```

PL/SQL procedure successfully completed.

```
SQL> select * from stud24;
```

ROLL	ATTEND ST
101	90
102	89.23
103	66.23 D
104	82.56
105	75
106	16 D

6 rows selected.

```
SQL> select * from dstud24;
```


ROLL	ATT
103	66.23
106	16