**Name :** Rohit Bangar **Class :** TE-C **Roll No :** TECOC306 **Batch :** C1

-----------------------------------------------------------------------------------------------------

**Assignment-6**

**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> **create table acc1(acc\_no int , name varchar(10) , amt int ,**

**status varchar(10));**

Table created.

SQL> **insert into acc1 values(101 , 'Shivani' , 30000 , 'IA');**

1 row created.

SQL> **insert into acc1 values(102 , 'Piyush' , 50000 , 'IA');**

1 row created.

SQL> **insert into acc1 values(103 , 'Snehal' , 55000 , 'IA');**

1 row created.

SQL> **select \* from acc1;**

ACC\_NO NAME AMT STATUS

---------- ---------- ---------- ---------

101 Shivani 30000 IA

102 Piyush 50000 IA

103 Snehal 55000 IA

SQL> **set serveroutput on;**

**SQL> declare**

**2 row number;**

**3 day int;**

**4 begin**

**5 update acc1 set status='A' where status='IA';**

**6 if sql%found then**

**7 row:=sql%rowcount;**

**8 dbms\_output.put\_line(row||' rows updated');**

**9 elsif sql%notfound then**

**10 row:=sql%rowcount;**

**11 dbms\_output.put\_line(row||' rows updated');**

**12 end if;**

**13 end;**

**14 /**

3 rows updated

**PL/SQL procedure successfully completed.**

SQL> **select \* from acc1;**

ACC\_NO NAME AMT STATUS

---------- ---------- ---------- ---------

101 Shivani 30000 A

102 Piyush 50000 A

103 Snehal 55000 A

**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> **create table emmp(eno int , esal int);**

Table created.

SQL> **insert into emmp values(1101 , 50000);**

1 row created.

SQL> **insert into emmp values(1102 , 65000);**

1 row created.

SQL> **insert into emmp values(1103 , 45000);**

1 row created.

SQL> **select \* from emmp;**

ENO ESAL

---------- ----------

1101 50000

1102 65000

1103 45000

SQL> **create table inc\_sal(eno int , i\_sal int);**

Table created.

SQL> **select \* from emmp;**

ENO ESAL

---------- ----------

1101 50000

1102 65000

1103 45000

SQL> **select avg(esal) from emmp;**

AVG(ESAL)

----------

**53333.3333**

**SQL> declare**

**2 cursor emmp\_cr is select eno , esal from emmp where esal<(se**

**lect avg(esal) from emmp);**

**3 no emmp.eno%type;**

**4 sal emmp.esal%type;**

**5 begin**

**6 open emmp\_cr;**

**7 loop**

**8 fetch emmp\_cr into no,sal;**

**9 exit when emmp\_cr%notfound;**

**10 update emmp set esal=esal+(esal\*0.1) where no=eno;**

**11 insert into inc\_sal values(no,(sal+(sal\*0.1)));**

**12 end loop;**

**13 end;**

**14 /**

PL/SQL procedure successfully completed.

SQL> **select \* from emmp;**

ENO ESAL

---------- ----------

1101 55000

1102 65000

1103 49500

SQL> **select \* from inc\_sal;**

ENO I\_SAL

---------- ----------

1101 55000

1103 49500

**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.**

SQL> **create table stud21(roll int , att int , status varchar(5));**

Table created.

SQL> **create table d\_stud(roll int , att int);**

Table created.

SQL> **insert into stud21 values(1 , 87 , '-');**

1 row created.

SQL> **insert into stud21 values(2 , 65 , '-');**

1 row created.

SQL> **insert into stud21 values(3 , 74 , '-');**

1 row created.

SQL> **insert into stud21 values(4 , 91 , '-');**

1 row created.

SQL> **insert into stud21 values(5 , 70 , '-');**

1 row created.

SQL> **select \* from stud21;**

ROLL ATT STATUS

---------- ---------- -----

1 87 -

2 65 -

3 74 -

4 91 -

5 70 -

SQL> **declare**

**2 cursor crsr\_att is select roll , att, status from stud21 whe**

**re att<75;**

**3 mroll stud21.roll%type;**

**4 matt stud21.att%type;**

**5 mstatus stud21.status%type;**

**6 begin**

**7 open crsr\_att;**

**8 if crsr\_att%isopen then**

**9 loop**

**10 fetch crsr\_att into mroll , matt , mstatus;**

**11 exit when crsr\_att%notfound;**

**12 if crsr\_att%found then**

**13 update stud21 set status='D' where roll=mroll;**

**14 insert into d\_stud values(mroll,matt);**

**15 end if;**

**16 end loop;**

**17 end if;**

**18 end;**

**19 /**

**PL/SQL procedure successfully completed.**

SQL> **select \* from stud21;**

ROLL ATT STATUS

---------- ---------- -----

1 87 -

2 65 D

3 74 D

4 91 -

5 70 D

SQL> **select \* from d\_stud;**

ROLL ATT

---------- ----------

2 65

3 74

5 70

**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> **create table new\_class(roll int , name varchar(10));**

Table created.

SQL> **create table old\_class(roll int , name varchar(10));**

Table created.

SQL> **insert into old\_class values(1 , 'Shivani');**

1 row created.

SQL> **insert into old\_class values(2 , 'Snehal');**

1 row created.

SQL> **insert into old\_class values(3 , 'Piyush');**

1 row created.

SQL> **insert into new\_class values(4 , 'Devesh');**

1 row created.

SQL> **insert into new\_class values(5 , 'Prerana');**

1 row created.

SQL> **insert into new\_class values(6 , 'Zuber');**

1 row created.

SQL> **select \* from old\_class;**

ROLL NAME

---------- ----------

1 Shivani

2 Snehal

3 Piyush

SQL> **select \* from new\_class;**

ROLL NAME

---------- ----------

4 Devesh

5 Prerana

6 Zuber

SQL> **declare**

**2 cursor crsr\_class is select \* from old\_class;**

**3 cursor crsr\_chk(str\_name varchar) is select roll from new\_cl**

**ass where name = str\_name;**

**4 str\_roll new\_class.roll%type;**

**5 str\_name new\_class.name%type;**

**6 v varchar(10);**

**7 begin**

**8 open crsr\_class;**

**9 loop**

**10 fetch crsr\_class into str\_roll , str\_name;**

**11 exit when crsr\_class%notfound;**

**12 open crsr\_chk(str\_name);**

**13 fetch crsr\_chk into v;**

**14 if crsr\_chk%found then**

**15 dbms\_output.put\_line(str\_name||' exist');**

**16 else**

**17 dbms\_output.put\_line(str\_name||' does not exist. Inserting i**

**n New table');**

**18 insert into new\_class values(str\_roll , str\_name);**

**19 end if;**

**20 close crsr\_chk;**

**21 end loop;**

**22 close crsr\_class;**

**23 end;**

**24 /**

**Shivani does not exist. Inserting in New table**

**Snehal does not exist. Inserting in New table**

**Piyush does not exist. Inserting in New table**

**PL/SQL procedure successfully completed.**

**SQL> select \* from new\_class;**

ROLL NAME

---------- ----------

4 Devesh

5 Prerana

6 Zuber

1 Shivani

2 Snehal

3 Piyush

6 rows selected.

**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> declare**

**2 cursor crsr\_emp is select \* from empl;**

**3 cursor crsr\_avg\_salary(var\_dno number) is select d\_no ,**

**avg(salary) from empl group by(d\_no);**

**4 var\_eno number(10);**

**5 var\_dno number(10);**

**6 var\_salary number(10);**

**7 m\_dno number(10);**

**8 m\_avsl number(10);**

**9 begin**

**10 open crsr\_emp;**

**11 loop**

**12 fetch crsr\_emp into var\_eno , var\_dno , var\_salary;**

**13 exit when crsr\_emp%notfound;**

**14 open crsr\_avg\_salary(var\_dno);**

**15 loop**

**16 fetch crsr\_avg\_salary into m\_dno , m\_avsl;**

**17 exit when crsr\_avg\_salary%notfound;**

**18 if crsr\_avg\_salary%found then**

**19 dbms\_output.put\_line('Found');**

**20 insert into dept\_salary values(m\_dno , m\_avsl);**

**21 else**

**22 dbms\_output.put\_line('Notfound');**

**23 end if;**

**24 end loop;**

**25 close crsr\_avg\_salary;**

**26 end loop;**

**27 close crsr\_emp;**

**28 end;**

**29 /**

**PL/SQL procedure successfully completed.**

**SQL> select \* from empl;**

E\_NO D\_NO SALARY

---------- ---------- ----------

1011 1 25000

1012 2 30000

1013 3 50000

1014 1 75000

1015 3 54000

1016 2 65000

6 rows selected.

**SQL> select \* from dept\_salary;**

D\_NO AVG\_SALARY

-------- ----------

1 50000

2 47500

3 52000

3 rows selected.

**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> create table studd21(roll int , att int , status int);**

Table created.

**SQL> create table d\_studd(roll int , att int);**

Table created.

**SQL> drop table studd21;**

Table dropped.

**SQL> create table studd21(roll int , att int , status varchar(2));**

Table created.

**SQL> insert into studd21 values(1 , 89 , '-');**

1 row created.

**SQL> insert into studd21 values(2 , 54 , '-');**

1 row created.

**SQL> insert into studd21 values(3 , 76 , '-');**

1 row created.

**SQL> insert into studd21 values(4 , 48 , '-');**

1 row created.

**SQL> insert into studd21 values(5 , 72 , '-');**

1 row created.

**SQL> select \* from studd21;**

ROLL ATT ST

---------- ---------- ---

1 89 -

2 54 -

3 76 -

4 48 -

5 72 -

**SQL> select \* from d\_studd;**

no rows selected

**SQL> declare**

**2 cursor crsr\_att is select roll , att , status from studd**

**21 where att<75;**

**3 begin**

**4 for m IN crsr\_att**

**5 loop**

**6 update studd21 set status='D' where roll=m.roll;**

**7 insert into d\_studd values(m.roll , m.att);**

**8 end loop;**

**9 end;**

**10 /**

**PL/SQL procedure successfully completed.**

**SQL> select \* from studd21;**

ROLL ATT ST

---------- ---------- --

1 89 -

2 54 D

3 76 -

4 48 D

5 72 D

**SQL> select \* from d\_studd;**

ROLL ATT

---------- ----------

2 54

4 48

5 72