16. Cursor (Any Two) a) 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 bank\_manager(

2 id number(3) not null primary key,

3 inactive\_days number(3)

4 );

Table created.

SQL> insert into bank\_manager (id, inactive\_days) values (01,256);

1 row created.

SQL> insert into bank\_manager (id, inactive\_days) values (02,456);

1 row created.

SQL> insert into bank\_manager (id, inactive\_days) values (03,545);

1 row created.

SQL> insert into bank\_manager (id, inactive\_days) values (04,222);

1 row created.

SQL> insert into bank\_manager (id, inactive\_days) values (05,120);

1 row created.

SQL> insert into bank\_manager (id, inactive\_days) values (06,03);

1 row created.

SQL> select \* from bank\_manager;

ID INACTIVE\_DAYS

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

1 256

2 456

3 545

4 222

5 120

6 3

6 rows selected.

SQL> alter table bank\_manager add status number(2) ;

Table altered.

SQL> select \* from bank\_manager;

ID INACTIVE\_DAYS STATUS

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

1 256

2 456

3 545

4 222

5 120

6 3

6 rows selected.

SQL> edit

Wrote file afiedt.buf

1 declare

2 total\_rows number(3);

3 begin

4 update bank\_manager set status = 1 where inactive\_days>356;

5 if sql%notfound then

6 dbms\_output.put\_line('No Record Found');

7 elsifsql%found then

8 total\_rows := sql%rowcount;

9 dbms\_output.put\_line('Account Updated: '||total\_rows);

10 end if;

11\* end;

SQL> /

PL/SQL procedure successfully completed.

SQL> set serveroutput on;

SQL> /

Account Updated: 2

PL/SQL procedure successfully completed.

SQL> select \* from bank\_manager;

ID INACTIVE\_DAYS STATUS

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

1 256

2 456 1

3 545 1

4 222

5 120

6 3

6 rows selected.

SQL>

b)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.

SQL> create table employee2(

2 id number not null primary key,

3 name varchar2(20),

4 salary number(10,2) not null

5 );

Table created.

SQL> insert into employee2(id,name,salary) values (1,'Rushikesh',20000);

1 row created.

SQL> insert into employee2(id,name,salary) values (2,'Ritul',30000);

1 row created.

SQL> insert into employee2(id,name,salary) values (3,'Sanket',35000);

1 row created.

SQL> insert into employee2(id,name,salary) values (4,'Isha',40000);

1 row created.

SQL> insert into employee2(id,name,salary) values (5,'Kunal',25000);

1 row created.

SQL> insert into employee2(id,name,salary) values (6,'Ranjit',18000);

1 row created.

SQL> select \* from employee2;

ID NAME SALARY

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

1 Rushikesh 20000

2 Ritul 30000

3 Sanket 35000

4 Isha 40000

5 Kunal 25000

6 Ranjit 18000

6 rows selected.

SQL> edit

Wrote file afiedt.buf

1 declare

2 av\_salary number(10,2);

3 begin

4 av\_salary := &av\_salary;

5 update employee2 set salary = salary\*0.10 where salary <av\_salary;

6 if sql%found then

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

8 elsifsql%notfound then

9 dbms\_output.put\_line('No Record Found');

10 end if;

11\* end;

SQL> /

Enter value for av\_salary: 28000

old 4: av\_salary := &av\_salary;

new 4: av\_salary := 28000;

PL/SQL procedure successfully completed.

SQL> set serveroutput on;

SQL> /

Enter value for av\_salary: 28000

old 4: av\_salary := &av\_salary;

new 4: av\_salary := 28000;

Rows Updated: 3

PL/SQL procedure successfully completed.

c) 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));

SQL> create table stud21(

2 roll number(4) not null primary key,

3 att number(4) not null,

4 status varchar(1)

5 );

Table created.

SQL> insert into stud21 (roll,att) values (1,78);

1 row created.

SQL> insert into stud21 (roll,att) values (2,58);

1 row created.

SQL> insert into stud21 (roll,att) values (3,76);

1 row created.

SQL> insert into stud21 (roll,att) values (4,66);

1 row created.

SQL> insert into stud21 (roll,att) values (5,56);

1 row created.

SQL> insert into stud21 (roll,att) values (6,88);

1 row created.

SQL> create table d\_stud(

2 roll number(4) not null,

3 att number(4) not null,

4 status varchar(1)

5 );

Table created.

SQL> set linesize 160;

SQL> select \* from stud21;

ROLL ATT S

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

1 78

2 58

3 76

4 66

5 56

6 88

6 rows selected.

SQL> declare

2 cursor stu\_cursor is

3 select roll,att from stud21 where att<75;

4 stud\_recordstu\_cursor%rowtype;

5 begin

6 open stu\_cursor;

7 loop

8 fetch stu\_cursor into stud\_record;

9 exit when stu\_cursor%notfound;

10 insert into d\_stud (roll,att) values (stud\_record.roll,stud\_record.att);

11 update stud21 set status = 'D' where roll = stud\_record.roll;

12 end loop;

13 end;

14 /

PL/SQL procedure successfully completed.

SQL> select \* from stud21;

ROLL ATT S

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

1 78

2 58 D

3 76

4 66 D

5 56 D

6 88

6 rows selected.

SQL> select \* from d\_stud;

ROLL ATT S

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

2 58

4 66

5 56

SQL>