

Assignment – 9

1. Write a PL/SQL block of code that first withdraws an amount of Rs. 500. Then again withdraws Rs. 500. Now if the current balance of a specific account number is less than Rs. 1000 then undo the last withdraw just made.

PL/SQL Code:

```
set serveroutput on
declare
    n number(20);
    t number(20);
    amt number:=500;
begin
    n:=&n;
    update Acc_details set Total_Cost=Total_Cost-amt where Acc_No=n;
    commit;
    savepoint s;
    update Acc_details set Total_Cost=Total_Cost-amt where Acc_No=n;
    select Total_Cost into t from Acc_details where Acc_No=n;
    if(t<1000) then
        dbms_output.put_line(' Balance after 2nd Transaction- ' || t);
        dbms_output.put_line(' Insufficient Balance. ');
        rollback to savepoint s;
        dbms_output.put_line(' Balance after Rollback ' || t);
    else
        commit;
        select Total_Cost into t from Acc_details where Acc_No=n;
        dbms_output.put_line(' Balance after COMMIT ' || t);
    end if;
end;
/
```

Output:

```
SQL> @ "F:\BTech\DBMS LAB\Solved_A_9\A9_1.sql";
Enter value for n: 001
old 6:          n:=&n;
new 6:          n:=001;
Balance after COMMIT 11000

PL/SQL procedure successfully completed.
```

2. Write a PL/SQL block of code to update the location of specific department number that will be taken from user. Display an appropriate message using SQL%FOUND based on existence of the record in the Department table and display an appropriate message using SQL%NOTFOUND based on the non-existence of the record in Department Table.

PL/SQL Code:

```
set serveroutput on
declare
    dno number:=&dno;
    loc1 varchar2(10):='&loc';
begin
    update Dept set loc=loc1 where Deptno=dno;
    if sql%found then
        dbms_output.put_line(' The updated loc is ' || loc1);
    end if;
```

```

        if sql%notfound then
            dbms_output.put_line(' The updated loc is not found. ');
        end if;
    end;
/

```

Output:

```

SQL> @ "F:\BTech\DBMS LAB\Solved_A_9\a9_2.sql";
Enter value for dno: 20
old 2:          dno number:=&dno;
new 2:          dno number:=20;
Enter value for loc: MUMBAI
old 3:          loc1 varchar2(10):='&loc';
new 3:          loc1 varchar2(10):='MUMBAI';
The updated loc is  MUMBAI

PL/SQL procedure successfully completed.

```

3. Write a PL/SQL block that will show an Employee name for a given Employee number. Here you try to enter a wrong Employee number and show an appropriate message i.e. NOT FOUND using exception handling.

PL/SQL Code:

```

set serveroutput on
declare
    ename varchar2(20);
    Eno number:=&Eno;
begin
    select ename into ename from Emp where Empno=Eno;
    dbms_output.put_line(' The Employee name is ' || ename);
    exception
    when NO_DATA_FOUND then
        dbms_output.put_line(' The Employee is not found for the given Emp No. ');
end;
/

```

Output:

```

SQL> @ "F:\BTech\DBMS LAB\Solved_A_9\a9_3.sql";
Enter value for eno: 7934
old 3:          Eno number:=&Eno;
new 3:          Eno number:=7934;
The Employee name is  MILLER

PL/SQL procedure successfully completed.

```

4. Write a PL/SQL block of code using your own exception handling that will show an error message whenever you want to insert a null value in a not null column.

PL/SQL Code:

```

set serveroutput on
declare
    IN_ERR exception;
    Pragma
    exception_init(IN_ERR, -01400);
begin
    insert into Emp values (null, 'BLAKE', 'MANAGER', 7839, to_date('1-5-1981','dd-mm-yyyy'),
    2850, null, 30);
    exception
    when IN_ERR then

```

```

        dbms_output.put_line(' Cannot insert Null values in not Null column. ');
end;
/

```

Output:

```

SQL> @ "F:\BTech\DBMS LAB\Solved_A_9\A9_4.sql";
Cannot insert Null values in not Null column.

PL/SQL procedure successfully completed.

```

5. a) Create a table Emp_sal_inc that have three column (Emp_id, Cur_sal, Inc_date).
- b) Now write a PL/SQL block of code will allow 2% salary increment of all employee of RESEARCH department. After that all records are to be inserted into the above table i.e. Emp_sal_inc.

PL/SQL Code:

```

set serveroutput on
create table Emp_sal_inc(
    Emp_id number(10),
    cur_sal number(20,4),
    inc_date date
);
declare
cursor cur is
select Empno, Sal from Emp where Deptno=(Select Deptno from Dept where
Dname='RESEARCH');
Emp_id number;
Emp_sal Emp.Sal%type;
begin
    open cur;
    if cur%isopen then
        loop
            fetch cur into Emp_id, Emp_sal;
            exit when cur%notfound;
            update Emp set Sal=Sal*1.02 where Empno=Emp_id;
            insert into Emp_sal_inc values(Emp_id, Emp_sal, SYSDATE);
        end loop;
        commit;
        dbms_output.put_line(cur%rowcount);
    else
        dbms_output.put_line(' Cursor not open....');
    end if;
close cur;
end;
/

```

Output:

```

SQL> @ "F:\BTech\DBMS LAB\Solved_A_9\A9_5.sql";

Table created.

5

PL/SQL procedure successfully completed.

```

```
SQL> select * from emp_sal_inc;
```

EMP_ID	CUR_SAL	INC_DATE
7566	3034.5	29-MAY-23
7788	3060	29-MAY-23
7902	3060	29-MAY-23
7369	816	29-MAY-23
7876	1122	29-MAY-23