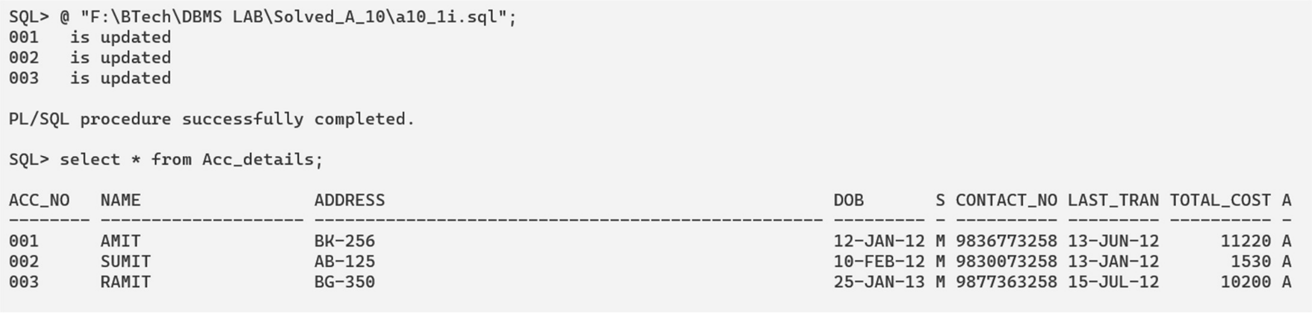
Assignment – 10

1. **Write a PL/SQL block that will add 2% interest of all customer of a bank for active account.**
   i) **For updating Acc_details updating, you have to use Cursor.**

PL/SQL Code:

```
set serveroutput on
declare
cursor add_interest
is
select Acc_no, Total_COST from Acc_details where Acc_status='A';
varaccn Acc_details.Acc_no%type;
varamt Acc_details.Total_COST%type;
begin
   open add_interest;
   if add_interest%isopen then
      loop
         fetch add_interest into varaccn, varamt;
         exit when add_interest%notfound;
         update Acc_details set Total_COST=varamt*1.02 where
         Acc_no=varaccn;
         dbms_output.put_line( varaccn || '   is updated  ');
       end loop;
    else
       dbms_output.put_line('Cursor not opened.  ');
    end if;
    close add_interest;
    commit;
end;
/
```

Output:



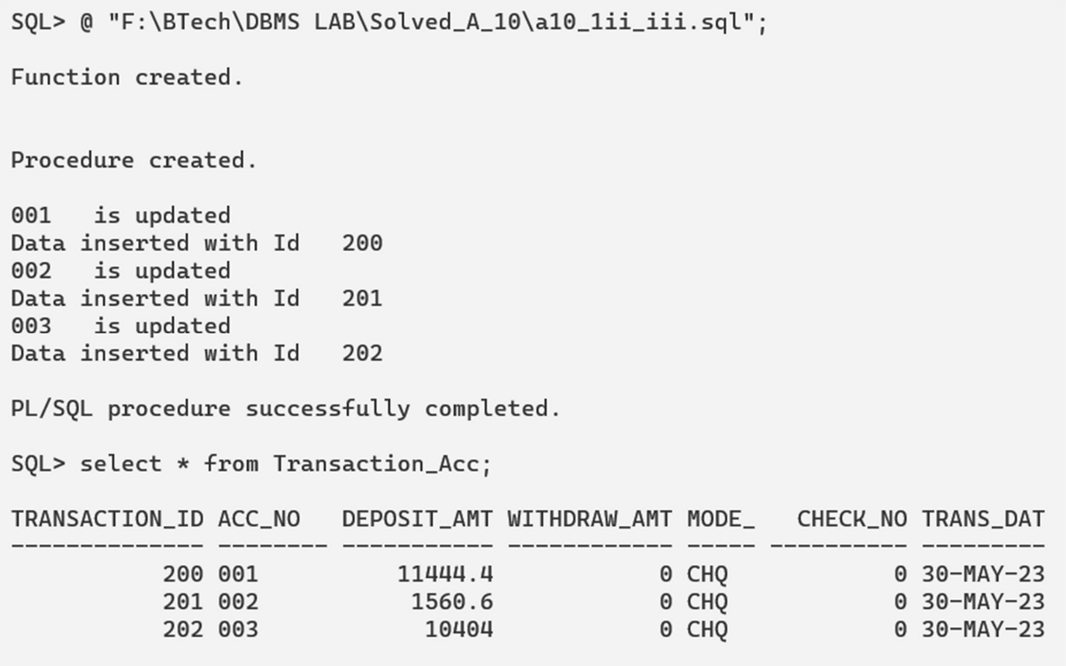
```
SQL> @ "F:\BTech\DBMS LAB\Solved_A_10\a10_1i.sql";
001   is updated
002   is updated
003   is updated

PL/SQL procedure successfully completed.

SQL> select * from Acc_details;

ACC_NO   NAME                 ADDRESS                                            DOB       S CONTACT_NO LAST_TRAN TOTAL_COST A
-------- -------------------- -------------------------------------------------- --------- - ---------- --------- ---------- -
001      AMIT                 BK-256                                             12-JAN-12 M 9836773258 13-JUN-12      11220 A
002      SUMIT                AB-125                                             10-FEB-12 M 9830073258 13-JAN-12       1530 A
003      RAMIT                BG-350                                             25-JAN-13 M 9877363258 15-JUL-12      10200 A
```

ii) For entry in Transaction_Acc, you have to use procedure.
iii) For Generation Transaction_id, you have to use function.

PL/SQL Code:

```
set serveroutput on
create function Max_id return number
is
var_id number(4);
begin
    select max(Transaction_id) into var_id from Transaction_acc;
    if var_id is null then
            var_id:=200;
    else
            var_id:=var_id+1;
    end if;
    return var_id;
    exception
    when no_data_found then
    return var_id;
end;
/
create procedure Transaction_entry(varaccn in Acc_details.Acc_no%type, varamt in
Acc_details.Total_cost%type)
is
vartid Transaction_acc.Transaction_id%type;
begin
    vartid:=Max_id();
    insert into Transaction_acc values(vartid, varaccn,varamt, 0, 'CHQ',0,Sysdate);
    dbms_output.put_line('  Data inserted with Id   ' ||vartid);
end;
/
set serveroutput on
declare
cursor add_interest
is
select Acc_no, Total_cost from Acc_details where Acc_status='A';
varaccn Acc_details.Acc_no%type;
varamt Acc_details.Total_COST%type;
begin
    open add_interest;
    if add_interest%isopen then
            loop
                    fetch add_interest into varaccn, varamt;
                    exit when add_interest%notfound;
                    update Acc_details set Total_cost=varamt*1.02 where Acc_no=varaccn;
                    dbms_output.put_line( varaccn || '   is updated  ');
                    varamt:=varamt*1.02;
                    Transaction_entry(varaccn, varamt);
            end loop;
    else
            dbms_output.put_line('Cursor not opened.  ');
    end if;
    close add_interest;
    commit;
end;
/
```

Output:



```
SQL> @ "F:\BTech\DBMS LAB\Solved_A_10\a10_1ii_iii.sql";

Function created.


Procedure created.

001   is updated
Data inserted with Id   200
002   is updated
Data inserted with Id   201
003   is updated
Data inserted with Id   202

PL/SQL procedure successfully completed.

SQL> select * from Transaction_Acc;

TRANSACTION_ID ACC_NO   DEPOSIT_AMT WITHDRAW_AMT MODE_   CHECK_NO TRANS_DAT
-------------- -------- ----------- ------------ ----- ---------- ---------
           200 001          11444.4            0 CHQ            0 30-MAY-23
           201 002           1560.6            0 CHQ            0 30-MAY-23
           202 003            10404            0 CHQ            0 30-MAY-23
```

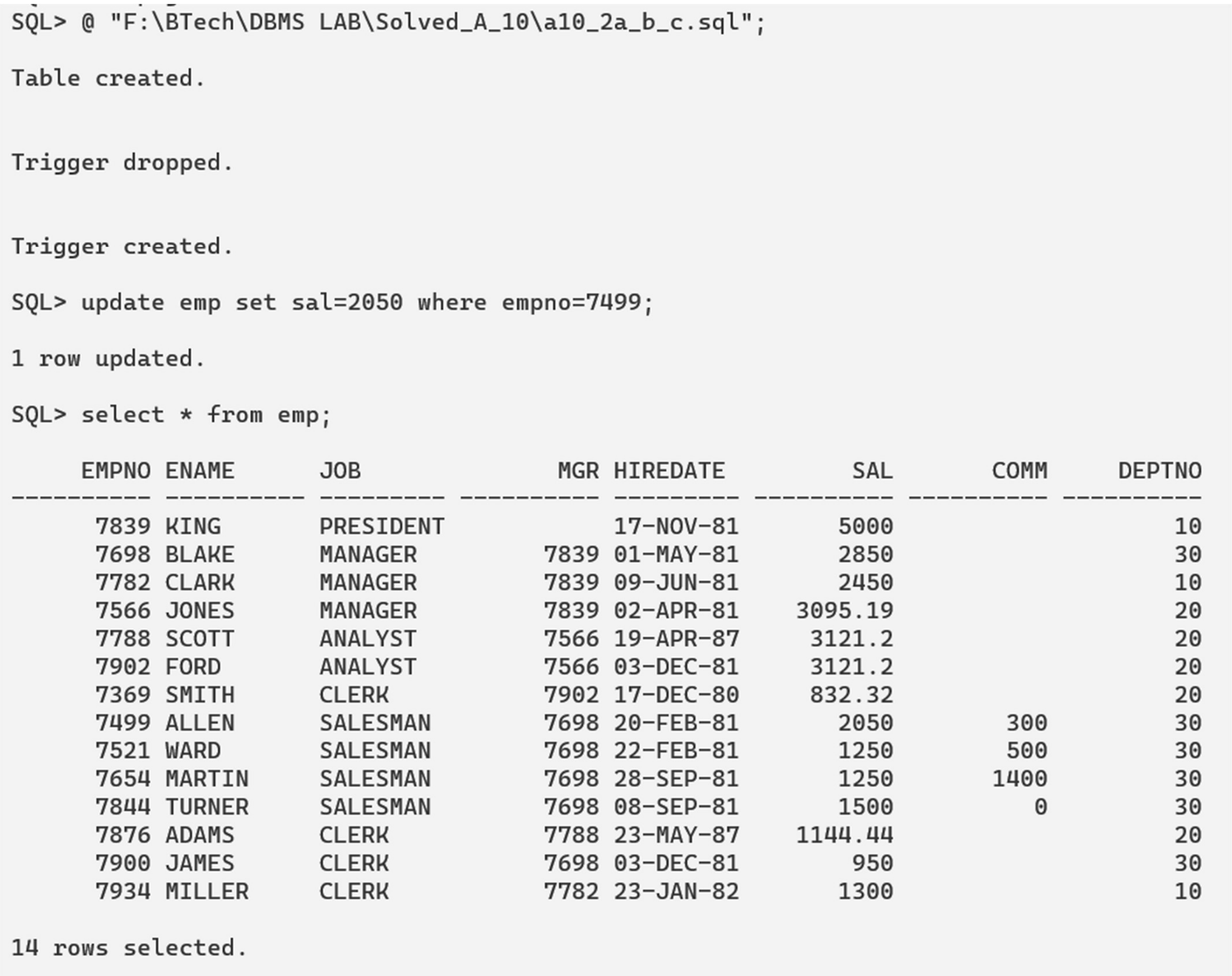
2. **a) Create the following table: (Table Name: Emp_audit)**
   b) Write a trigger that must keep track of records (in above table) that are being deleted or updated from Employee table.
   c) Write a SQL command to update the employee entry and describe the output.

   PL/SQL Code:

```
set serveroutput on
create table Emp_audit(
    Emp_no number(4) primary key,
    Dept_no number(4) not null references Dept,
    Status varchar2(8),
    Salary number(8,2) not null,
    Audit_date date not null
);

set serveroutput on
drop trigger trg_sal;
create trigger trg_sal after
update or delete on Emp for each row
declare
status varchar2(20);
begin
    if updating then
            status:='UPDATE';
    end if;
    if deleting then
            status:='DELETE';
    end if;
    insert into Emp_audit values(:Old.Empno, :Old.Deptno,status, :Old.Sal,SYSDATE);
```

```
end;
/
update emp set sal=2050 where empno=7499;
```

Output:



```
SQL> @ "F:\BTech\DBMS LAB\Solved_A_10\a10_2a_b_c.sql";

Table created.


Trigger dropped.


Trigger created.

SQL> update emp set sal=2050 where empno=7499;

1 row updated.

SQL> select * from emp;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7566 JONES      MANAGER         7839 02-APR-81    3095.19                    20
      7788 SCOTT      ANALYST         7566 19-APR-87     3121.2                    20
      7902 FORD       ANALYST         7566 03-DEC-81     3121.2                    20
      7369 SMITH      CLERK           7902 17-DEC-80     832.32                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       2050        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7876 ADAMS      CLERK           7788 23-MAY-87    1144.44                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

14 rows selected.
```