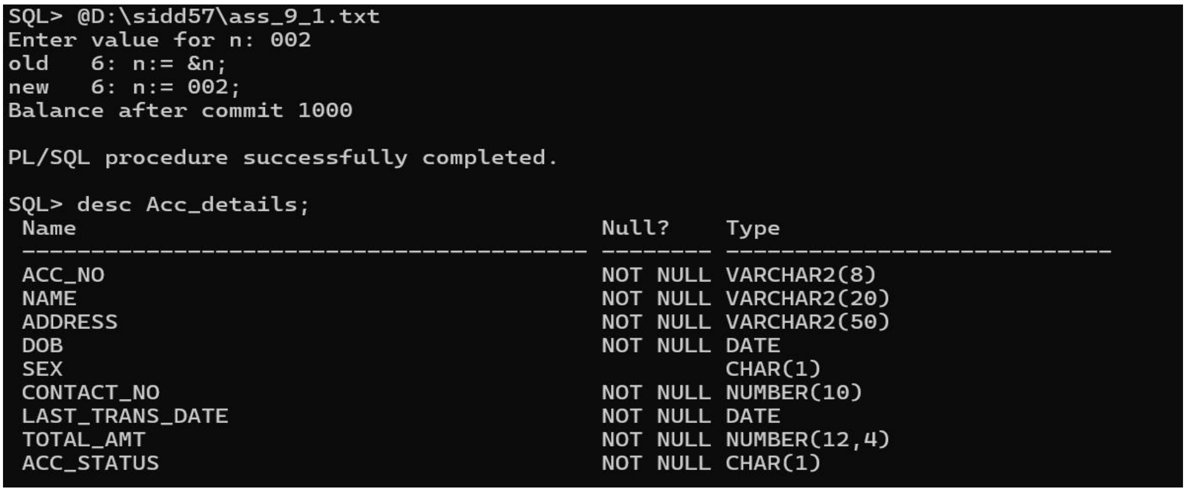
ASSIGNMENT – 9

1. **Problem Statement: 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 specificaccount number is less than Rs. 1000 then undo thelast withdraw just made.**

Query:

```
set serveroutput on
declare
n number(20);
t number(20);
amt number:=500;
begin
n:= &n;
update Acc_details set Total_amt=Total_amt-amt where Acc_No=n;
commit;
savepoint s;
select Total_amt 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 to Rollback '||t);
else
commit;
select Total_amt into t from Acc_details  where Acc_No=n;
dbms_output.put_line('Balance after commit '||t);
end if;
end;
/
```

Output:



```
SQL> @D:\sidd57\ass_9_1.txt
Enter value for n: 002
old   6: n:= &n;
new   6: n:= 002;
Balance after commit 1000

PL/SQL procedure successfully completed.

SQL> desc Acc_details;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ACC_NO                                    NOT NULL VARCHAR2(8)
 NAME                                      NOT NULL VARCHAR2(20)
 ADDRESS                                   NOT NULL VARCHAR2(50)
 DOB                                       NOT NULL DATE
 SEX                                                CHAR(1)
 CONTACT_NO                                NOT NULL NUMBER(10)
 LAST_TRANS_DATE                           NOT NULL DATE
 TOTAL_AMT                                 NOT NULL NUMBER(12,4)
 ACC_STATUS                                NOT NULL CHAR(1)
```

2. Problem Statement:

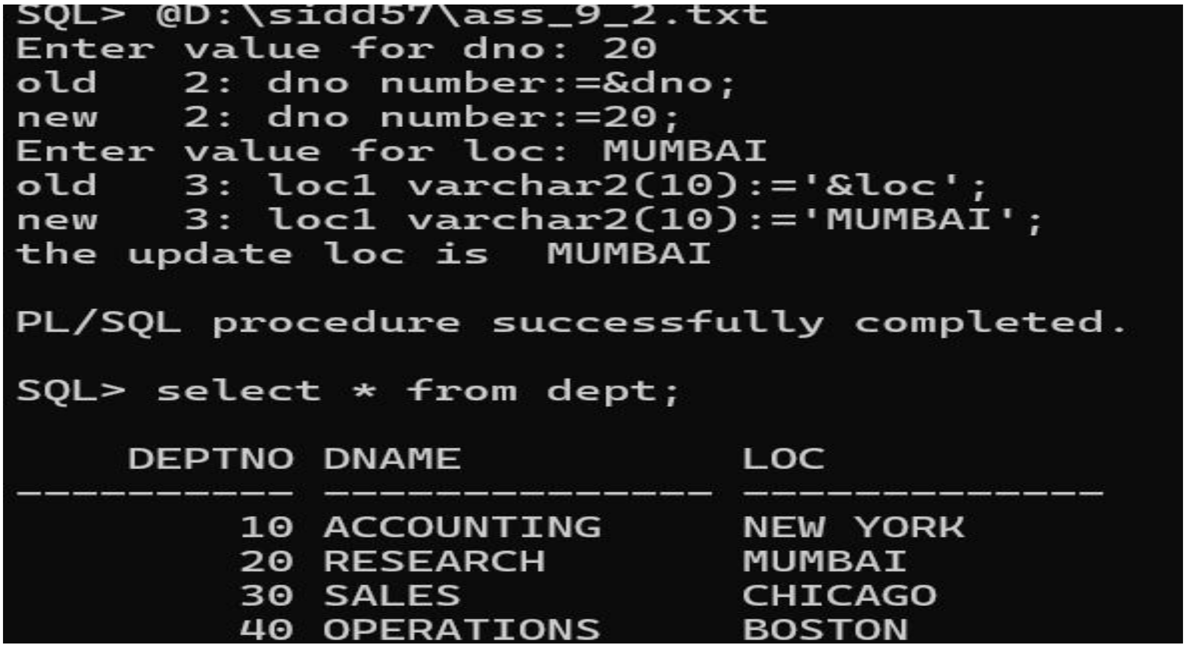
Write a PL/SQL block of code to update the location of specific department number that will be taken from user. Display an appropriate message using SQL%FOUND based on existence of the record in the Department table and display an appropriate message using SQL%NOTFOUND based on the non-existence of the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

Query:

```
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 update loc is  '|| loc1);

end if;

if sql%notfound then

dbms_output.put_line('the update loc is not found. ');

end if;
```

```
end;

/
```

Output:



```
SQL> @D:\sidd57\ass_9_2.txt
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 update loc is  MUMBAI

PL/SQL procedure successfully completed.

SQL> select * from dept;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       MUMBAI
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON
```

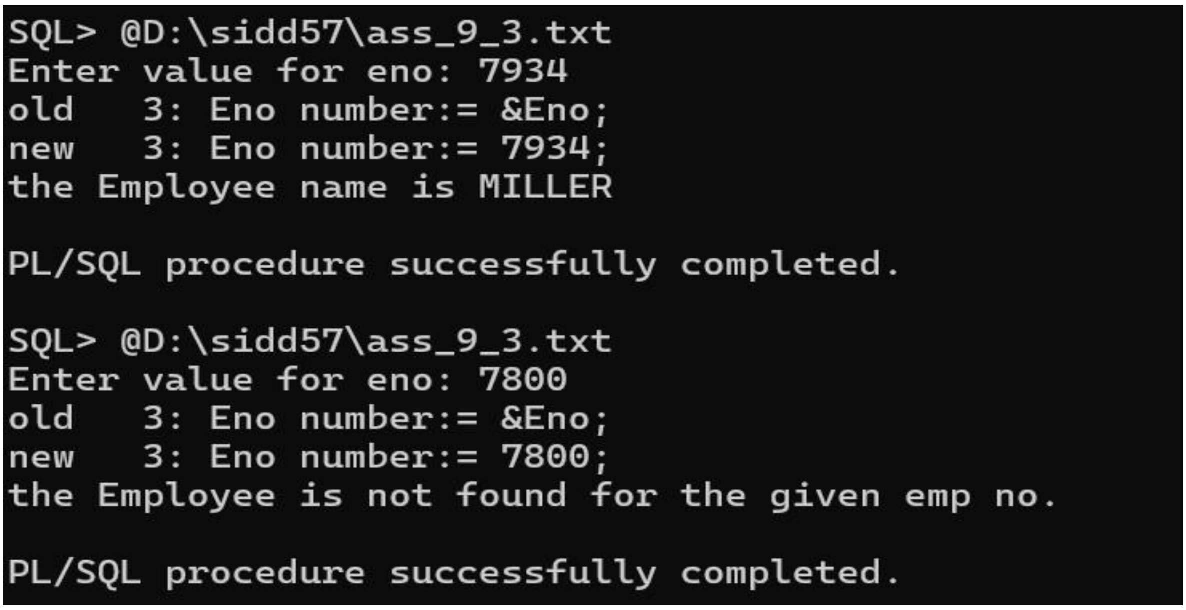
3. Problem Statement:

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.

Query:

```
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:



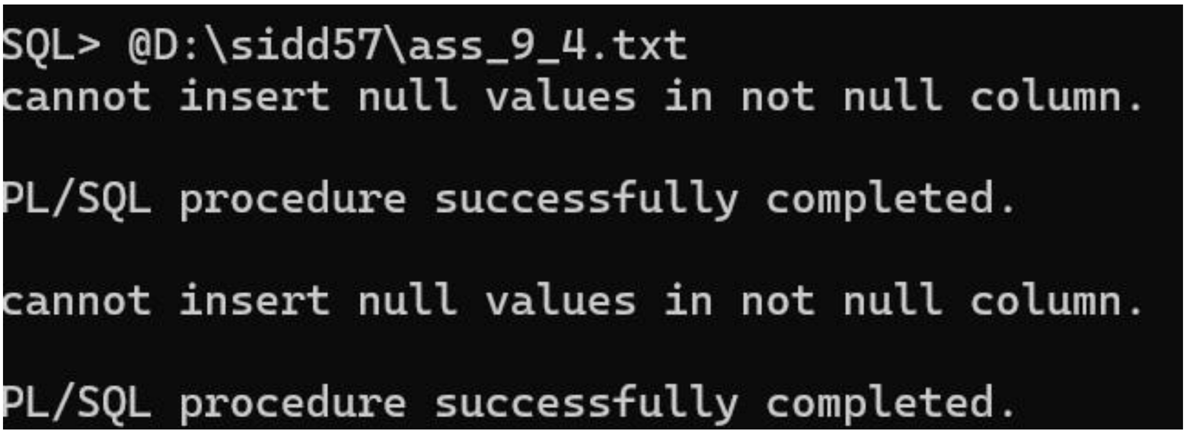
```
SQL> @D:\sidd57\ass_9_3.txt
Enter value for eno: 7934
old   3: Eno number:= &Eno;
new   3: Eno number:= 7934;
the Employee name is MILLER

PL/SQL procedure successfully completed.

SQL> @D:\sidd57\ass_9_3.txt
Enter value for eno: 7800
old   3: Eno number:= &Eno;
new   3: Eno number:= 7800;
the Employee is not found for the given emp no.

PL/SQL procedure successfully completed.
```

4. **_Problem Statement: 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 valuein a not null column.**

Query:

```
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> @D:\sidd57\ass_9_4.txt
cannot insert null values in not null column.

PL/SQL procedure successfully completed.

cannot insert null values in not null column.

PL/SQL procedure successfully completed.
```

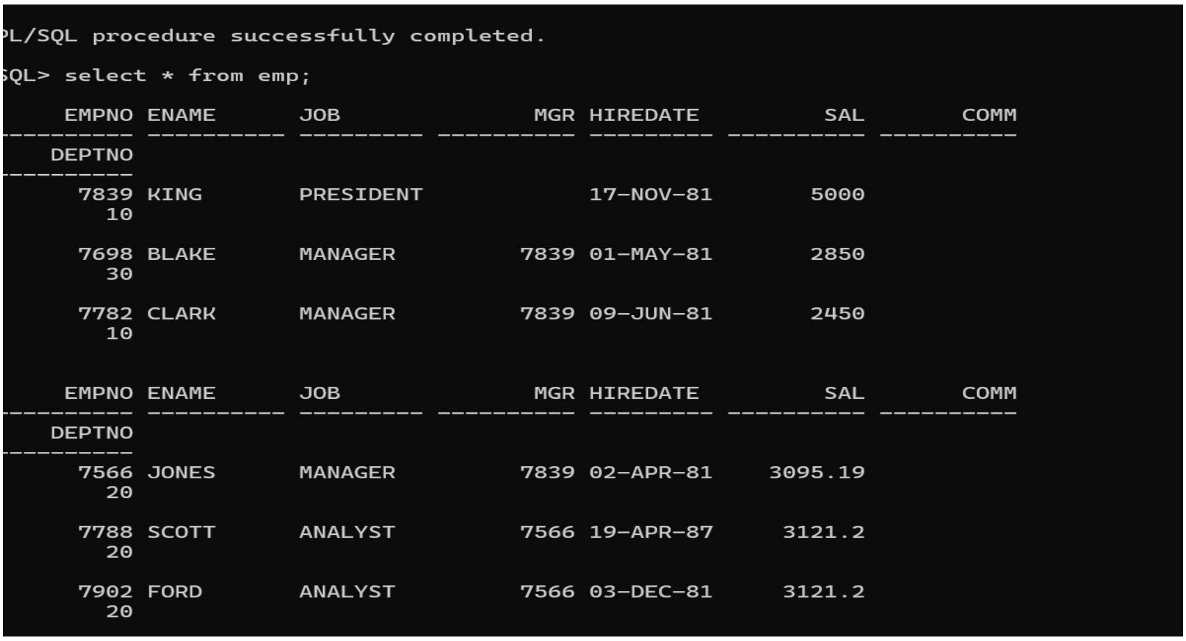
5. Problem Statement:

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.

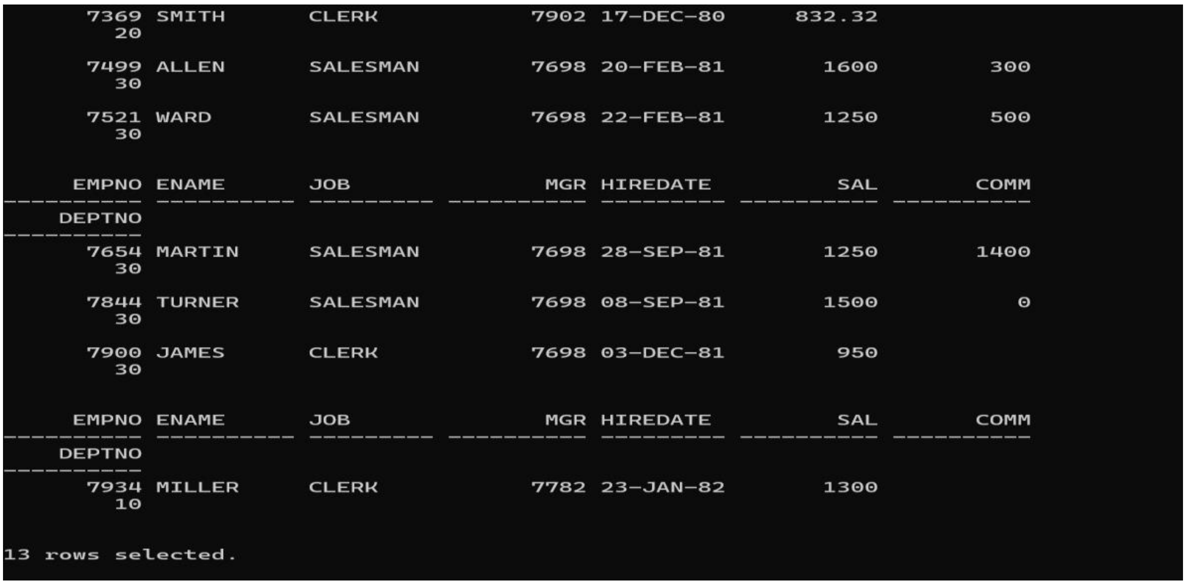
Query:

```
set serveroutput on
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;
/
```

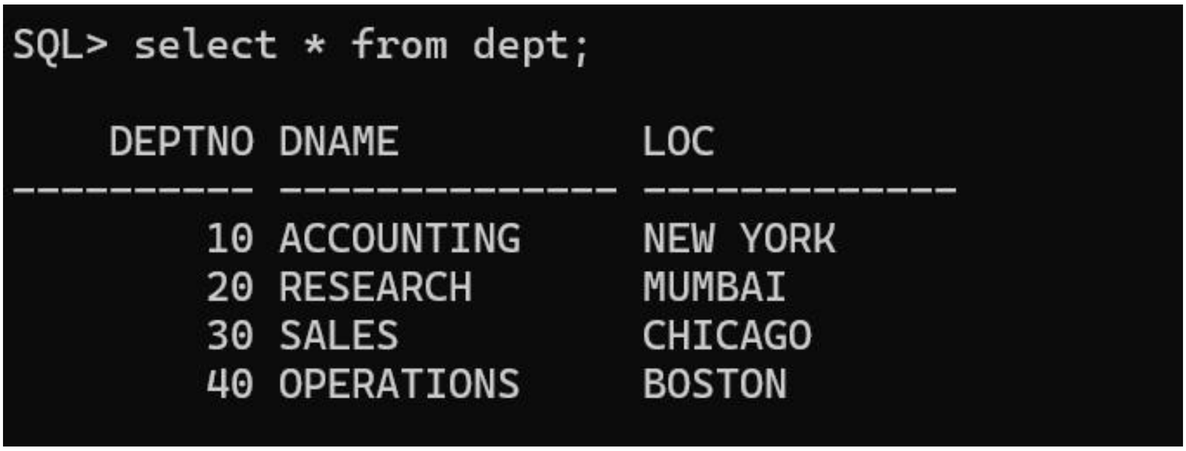
Output



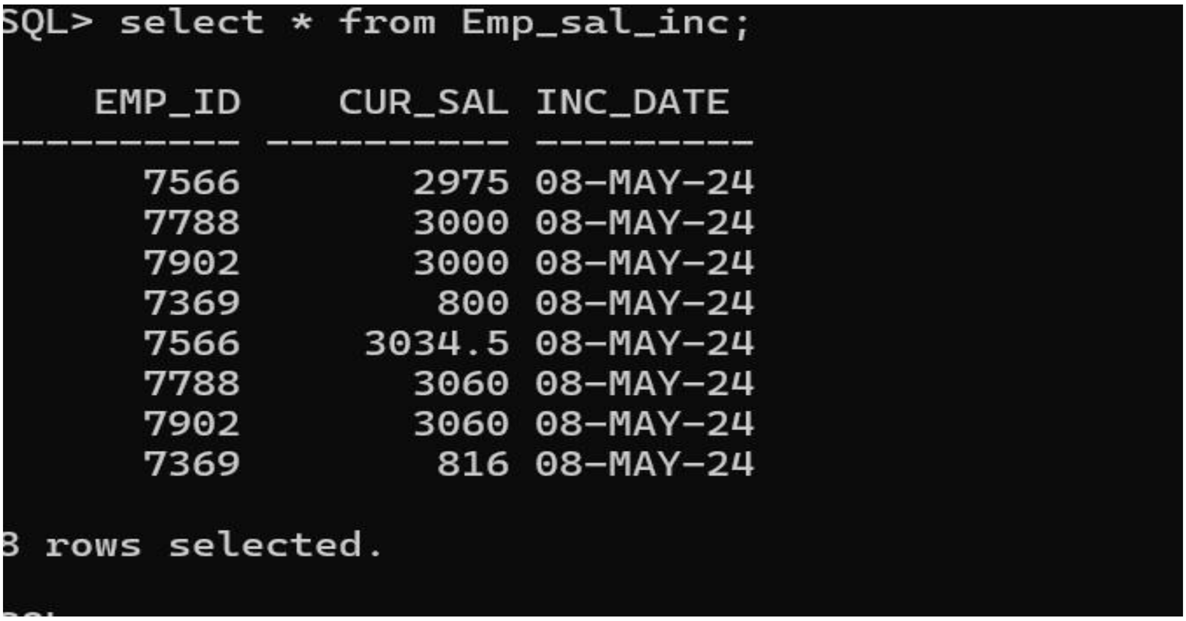
```
PL/SQL procedure successfully completed.

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


     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM
---------- ---------- --------- ---------- --------- ---------- ----------
    DEPTNO
----------
      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       1600        300
        30

      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500
        30

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM
---------- ---------- --------- ---------- --------- ---------- ----------
    DEPTNO
----------
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400
        30

      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0
        30

      7900 JAMES      CLERK           7698 03-DEC-81        950
        30

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM
---------- ---------- --------- ---------- --------- ---------- ----------
    DEPTNO
----------
      7934 MILLER     CLERK           7782 23-JAN-82       1300
        10


13 rows selected.
```



```
SQL> select * from dept;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       MUMBAI
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON
```



```
SQL> select * from Emp_sal_inc;

    EMP_ID    CUR_SAL INC_DATE
---------- ---------- ---------
      7566       2975 08-MAY-24
      7788       3000 08-MAY-24
      7902       3000 08-MAY-24
      7369        800 08-MAY-24
      7566     3034.5 08-MAY-24
      7788       3060 08-MAY-24
      7902       3060 08-MAY-24
      7369        816 08-MAY-24

8 rows selected.
```