NAME: GOURAV KUMAR SHAW

ENROLLMENT NO: 2020CSB010

SECTION: GX

SUBJECT: DBMS Lab

ASSIGNMENT NO. – 9

Creation of Tables:

Creating DEPT table:

Query:

```
create table DEPT (
DEPTNO int primary key,
DNAME varchar(15) ,
LOC varchar(10)
);
```

TABLE DEPT

Column	Null?	Type
DEPTNO	NOT NULL	NUMBER
DNAME	-	VARCHAR2(15)
LOC	-	VARCHAR2(10)

Download CSV

3 rows selected.

DEPT table after insertion of values:

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Download CSV

4 rows selected.

Creating EMP table :

Query:

```
create table EMP (
EMP_NO int primary key,
ENAME varchar(15),
JOB char(16),
MGR int references EMP(EMP_NO),
HIREDATE date,
SAL int,
COMM int,
DEPTNO int references DEPT(DEPTNO)
);
```

TABLE EMP

Column	Null?	Type
EMP_NO	NOT NULL	NUMBER
ENAME	-	VARCHAR2(15)
JOB	-	CHAR(16)
MGR	-	NUMBER
HIREDATE	-	DATE
SAL	-	NUMBER
COMM	-	NUMBER
DEPTNO	-	NUMBER

Download CSV

8 rows selected.

EMP table after insertion of values:

EMP_NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7934	MILLER	CLERK	7782	23-JAN-82	1300	-	10
7698	BLAKE	MANAGER	7839	01-MAY-81	2850	-	30
7782	CLARK	MANAGER	7839	09-JUN-81	2450	-	10
7788	SCOTT	ANALYST	7566	19-APR-87	3000	-	20
7902	FORD	ANALYST	7566	03-DEC-81	3000	-	20
7369	SMITH	CLERK	7902	17-DEC-80	800	-	20
7900	JAMES	CLERK	7698	03-DEC-81	950	-	30
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	23-MAY-87	1100	-	20
7566	JONES	MANAGER	7839	02-APR-81	2975	-	20

Download CSV

14 rows selected.

Creating x table :

Query:

```
CREATE TABLE x (
A NUMBER(2) PRIMARY KEY,
B VARCHAR2(25))
```

TABLE X

Column	Null?	Type
A	NOT NULL	NUMBER(2,0)
B	-	VARCHAR2(25)

Download CSV

2 rows selected.

A. ASSIGNMENTS ON CURSORS

1. Take the help of implicit cursor to print the total number of rows deleted after a deletion operation performed on a table.

```
declare
affected char(4);
begin
delete from x;
affected:=to_char(SQL%ROWCOUNT);
dbms_output.put_line(affected);
end;
```

Output:

```
Statement processed.
4
```

2. Create a table X that contains two attributes A(number(2)) and B(varchar2(25)). Now, create an explicit cursor that will help you to populate the table X with the value from DEPT table (Ref. Assignment No. - 2) where A is same as deptno and b is same as dname.

```
declare
cursor cur2 is
select deptno,dname
from dept;
```

```
a_m dept.deptno%type;
b_m dept.dname%type;
begin
open cur2;
if cur2%isopen then
delete from x;
loop
fetch cur2 into a_m,b_m;
exit when cur2%notfound;
if cur2%found then
insert into x values(a_m,b_m);
end if;
end loop;
commit;
else
dbms_output.put_line('Unable to open cursor');
end if;
close cur2;
end;
```

Output:

```
Statement processed.
```

3. Print the second highest salary of the EMP table (Ref. Assignment No. - 2) using an explicit cursor.

```
declare
cursor cur3 is
select max(sal)
```

```
from emp
where sal<(select max(sal) from emp);
sal_m emp.sal%type;
begin
open cur3;
if cur3%isopen then
fetch cur3 into sal_m;
dbms_output.put_line(to_char(sal_m));
else
dbms_output.put_line('Unable to open Cursor');
end if;
close cur3;
end;
```

Output:

```
Statement processed.
3000
```

B. ASSIGNMENTS ON TRIGGERS

1. Create a row trigger that will display the second highest salary of EMP table (or perform any other valid action) after any updation performed on the DEPT table.

```
create trigger tr1
after update on DEPT
for each row
declare
cursor cur3 is
select max(sal)
from emp
where sal<(select max(sal) from emp);
sal_m emp.sal%type;
begin
open cur3;
if cur3%isopen then
fetch cur3 into sal_m;
dbms_output.put_line('AFTER
TRIGGER'||to_char(sal_m));
else
dbms_output.put_line('Unable to open Cursor');
end if;
close cur3;
end;
```

Output:

```
Trigger created.
```

2. Repeat problem no. 1 for before trigger.

```
create trigger tr2
before update on DEPT
for each row
declare
cursor cur3 is
select max(sal)
from emp
where sal<(select max(sal) from emp);
sal_m emp.sal%type;
begin
open cur3;
if cur3%isopen then
fetch cur3 into sal_m;
dbms_output.put_line('BEFORE
TRIGGER'||to_char(sal_m));

else
dbms_output.put_line('Unable to open Cursor');
end if;
close cur3;
end;
```

Output:

```
Trigger created.
```

3. Modify the triggers of problem nos. 1 and 2 to statement triggers and observe the changes in the outputs.

```
create trigger tr3
before update on DEPT

declare
cursor cur3 is
select max(sal)
from emp
where sal<(select max(sal) from emp);
sal_m emp.sal%type;
begin
open cur3;
if cur3%isopen then
fetch cur3 into sal_m;
dbms_output.put_line('BEFORE
TRIGGER'||to_char(sal_m));

else
dbms_output.put_line('Unable to open Cursor');
end if;
close cur3;
end;
```

Output:

```
Trigger created.
```