EXNO:4

DATE:

Cursors, Procedures and Functions

PROCEDURE:

Q1: Create the following tables:

'em' with empid, name, and dept.

'salary' with empid, basic, hra

Query:

SQL> create table em(empid number(3),name varchar(25),dept varchar(20));

Output:

Table created.

Query:

SQL>create table salary(empid number(3),basic number(6),hra number(6));

Output:

Table created.

Q2: Insert few records into the table em and salary.

Query:

SQL> insert into em values(&empid,'&name','&dept');

Output:

Enter value for empid: 10

Enter value for name: priya

Enter value for dept: IT

old 1: insert into em values(&empid,'&name','&dept')

new 1: insert into em values(10,'priya','IT')

1 row created.

Query:

SQL> insert into salary values(&empid,&basic,&hra);

Output:

Enter value for empid: 10

Enter value for basic: 23000

Enter value for hra: 20000

old 1: insert into salary values(&empid,&basic,&hra)

new 1: insert into salary values(10,23000,20000)

1 row created.

Q3. Display the records from the table em and salary.

Query:

SQL> select * from em;

Output:

```
EMPID          NAME                 DEPT
---------- ------------------------ --------------------
10             priya                    IT
11             reena                    ECE
12             meena                    EEE
```

Query:

SQL> select * from salary;

Output:

```
EMPID     BASIC      HRA
---------- ---------- -------------
10       23000      20000
11       33000      30000
12       43000      40000
```

Q4. Write a PL/SQL Procedure to display all the records in employee table as "The Employer <empname> has a ID <empid> working in <Dept> Department".

Query:

SQL> CREATE OR REPLACE PROCEDURE disp

IS

CURSOR emp_cur is

Select EmpId,Name,Dept from em;

emp_rec emp_cur%rowtype;

BEGIN

```
FOR emp_rec in emp_cur
LOOP
dbms_output.put_line('The Employer ' || emp_rec.name || ' has id' || emp_rec.empid || ' Working in the Department : ' || emp_rec.dept);
END LOOP;
END;
/
```

Output:

Procedure created.

Q5: Write a Query to call the above procedure to display the output.

Query:

SQL> Set serveroutput on;

SQL> exec disp;

Output:

The Employer priya has id10 Working in the Department : IT

The Employer reena has id11 Working in the Department : ECE

The Employer meena has id12 Working in the Department : EEE

PL/SQL procedure successfully completed.

Q6: Write a PL/SQL function to return the name of the employee for the employee id mention in the function.

Query:

SQL>

```
CREATE OR REPLACE FUNCTION em_dtl_func
RETURN em.name%type
IS
emp_name em.name%type;
BEGIN
SELECT name INTO emp_name FROM em WHERE empID = 12;
RETURN emp_name;
END;
/
```

Output:

Function created.

Q7: Write a Query to display the Output for the above function

Query:

SQL> select em_dtl_func from dual;

Output:

EM_DTL_FUNC

---

meena

Q8: Write PL/SQL Procedure to get the Employee Id from the input and store the Employer Name for the given ID to Out Parameter.

Query:

SQL>CREATE OR REPLACE PROCEDURE emp_name (id IN em.empid%type, ename OUT em.name%type)

IS

BEGIN

SELECT name INTO ename

FROM em WHERE empid = id;

END;

/

Output:

Procedure created.

Q9: Write a PL/SQL Block to call the above Procedure using the Cursor. The Cursor will contain the entire Employee id from the em table and give the ID to the above procedure. The PL/SQL Block code that retrieve the OUT Parameter from the above Procedure and display the Output.

Query:

SQL> DECLARE

ename em.name%type;

CURSOR id_cur is SELECT empid FROM em;

```
emp_rec id_cur%rowtype;
BEGIN
FOR emp_rec in id_cur
LOOP
emp_name(emp_rec.empid, ename);
dbms_output.put_line('The employee ' || ename || ' has id ' || emp_rec.empid);
END LOOP;
END;
/
```

Output:

```
The employee priya has id 10
The employee reena has id 11
The employee meena has id 12
PL/SQL procedure successfully completed.
```

Q10: Write a PL/SQL Procedure to get the Employee Id from the table salary as input and Basic as IN OUT Parameter and calculate the bonus based on their Basic as per the following condition.

If the Basic below 10000 then increase the Basic to 8%

If the Basic between 10000 and 20000 then increase the Basic to 12%

If the Basic between 20000 and 30000 then increase the Basic to 15%

If the Basic above 30000 then increase the Basic to 20%

Query:

SQL> CREATE OR REPLACE PROCEDURE emp_Bonus (id IN salary.empid%type , Bas IN OUT Salary.Basic%type)

```
IS
tmp_sal salary.Basic%type;
BEGIN
tmp_sal:=Bas;
IF tmp_sal < 10000 THEN
Bas := tmp_sal +(tmp_sal * .08);
ELSIF tmp_sal between 10000 and 20000 THEN
```

```
Bas := tmp_sal +(tmp_sal * .12);
ELSIF tmp_sal between 20000 and 30000 THEN
Bas := tmp_sal +(tmp_sal * .15);
ELSIF tmp_sal > 30000 THEN
Bas := tmp_sal +(tmp_sal * .20);
END IF;
END;
/
```

Output:

Procedure created.

Q11: Write PL/SQL Block for the above procedure to display the output.

Query:

```
SQL > DECLARE
CURSOR updated_sal is
SELECT empid, Basic FROM Salary;
pre_sal salary.Basic%type;
BEGIN
FOR emp_rec IN updated_sal
LOOP
pre_sal := emp_rec.Basic;
emp_Bonus(emp_rec.empID, emp_rec.Basic);
dbms_output.put_line('The Bonus of ' || emp_rec.empID || ' increased from '|| pre_sal || ' to
'||emp_rec.Basic);
END LOOP;
END;
/
```

Output:

```
The Bonus of 10 increased from 23000 to    26450
The Bonus of 11 increased from 33000 to    39600
The Bonus of 12 increased from 43000 to    51600
PL/SQL procedure successfully completed.
```

Q12: Write a PL/SQL Function to find the Net Salary for the given Employee

Query:

SQL> CREATE OR REPLACE FUNCTION NETSAL(id IN salary.empid%type)
RETURN salary.basic%type
IS
netsal salary.basic%type;
BEGIN
SELECT sum(basic) + sum(hra) INTO netsal FROM salary WHERE empid = id;
RETURN (netsal);
END;
/

Output:

Function created.

Q13: Write PL/SQL Block to display the output for the above Function.

Query:

SQL> variable sal number
SQL> execute :sal := netsal(12)

Output:

PL/SQL procedure successfully completed.

Query:

SQL> print sal

Output:

SAL

83000

RESULT:

Thus the PL/SQL stored procedures are successfully executed to perform various operations like calculation of Net Salary of the given employee through the parameter (IN and OUT) and using the cursor to display the output in formatted way.