

LAB PROGRAMS - 8 PL/SQL FUNCTIONS AND PROCEDURES

* PROCEDURES:

- 1) Get the number of employees working under a given employee:

create or replace procedure nos(e1 emp.empno% type) is
cnt number;

begin

select count(*) into cnt from emp where mgr=e1;
dbms_output.put_line(cnt);

exception

when no_data_found then

dbms_output.put_line('wrong emp no');

end nos;

O/P:

Procedure created

execute nos(7698);

5

PL/SQL procedure successfully completed.

- 2) Get the no. of employees working in given department name:

~~create~~

2) Get the no. of employees working in given department name:

create or replace procedure nosd(d1 dept.dname%type) is
cnt number;

begin

select count(*) into cnt from emp,dept where
emp.deptno = dept.deptno and dname=d1;

dbms_output.put_line(cnt);

exception

when no_data_found then

dbms_output.put_line('wrong dept name');

end nosd;

O/P:

Procedure created

execute nosd('RESEARCH');

5

PL/SQL procedure successfully executed.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DBMS

Name K.S.I.SIVANI Roll No. - 052 Page No. _____

3) Create a procedure to accept an empno and a salary increase amount, if empno is not found or current salary is NULL then raise exceptions otherwise display total salary.

create or replace procedure empis(e1 emp.empno% type,
in1 emp.sal% type) as

```

s1 emp.sal% type;
nosal exception;
nsal emp.sal% type;
begin
    select sal into s1 from emp where empno=e1;
    if s1 is null then
        raise nusal;
    else
        nsal:=s1+in1;
        dbms_output.put_line(nsal);
    end if;
exception
    when nusal then
        dbms_output.put_line('Given emp.sal
                            is null');
    when no_data_found then
        dbms_output.put_line('wrong empno');
end empis;
```

Q/R:

Function created:

O/P:

Procedure created

execute empis(7698,1000);

3850

PL/SQL procedure successfully completed.

* FUNCTIONS:

1) Write a program to check whether the given number is prime or not.

Create or replace function prime(n number) return number as

```

    cnt : number;
begin
    cnt := 0;
for i in 1 .. n loop
    if mod(n, i) = 0 then
        cnt := cnt + 1
    end if;
end loop;
    return cnt;
end prime;

```

declare

```

    num   number;
    count1 number;

```

begin

```

        num := &num;
        count1 := prime(num);
if count1 > 2 then dbms_output.put_line("Not prime");
else dbms_output.put_line("Prime");
end if;
end;

```

O/P:

Function created.

Enter value for num: 7

7 is prime

PL/SQL procedure
successfully
executed.

2) Write a program to check for the existence of P# in the table parts:

create or replace function ex(pno p.p#%type) return number as

pnum p.p#%type;
cnt number;

begin

cnt:=0;

select p# into pnum from p where p#=pno;

if pno=pnum then

cnt:=1;

end if;

return cnt;

end ex;

declare

n p.p#%type;

i number;

begin n:='&n';

i:=ex(n);

if i=1 then dbms_output.put_line('Given ''n'' is
in the table');

end if;

exception

when no_data_found then

dbms_output.put_line("Given ''n'' is not in
the table");

end;

OIP:

Function created

Enter a value for n: 7698

Given 7698 is in the table.

PL/SQL procedure successfully executed.

Function created

Enter value for n: 1234

Given 1234 is not in the table.

PL/SQL procedure successfully executed.

* Procedures based on employee table:

1) Write a procedure to update salary of given employee by 10%.

create or replace procedure update_emp_sal(p_empno IN number) AS

v_salary emp.sal% type;

begin

select sal into v_salary from emp

where empno = p_empno;

v_salary := v_salary * 1.1;

UPDATE emp SET sal = v_salary where empno = p_empno;

dbms_output.put_line('Employee salary updated successfully');

exception

when no_data_found then

dbms_output.put_line('Employee not found');

when others then

dbms_output.put_line("Error occurred: "||sqlerrm);

END;

/

O/P:

procedure created

+execute update_emp_sal(7698);

Employee salary updated successfully.

+execute update_emp_sal(7568);

Employee not found.

NAME OF THE LABORATORY : DBMS

Name K.S.I.SIVANI

Roll No. -052

Page No.

81

2) Write a procedure to find the sum of salaries of employees belongs to given department.

Create or replace procedure cal-dept-salsum

(p-deptname IN varchar2,
p-sum OUT NUMBER) AS

BEGIN

p-sum := 0;

Select sum(sal) into p-sum from emp e join
dept d on d.deptno = e.deptno where d.dname
= p-deptname;

dbms_output.put_line('Sum of salaries in the department
: ' || p-sum);

exception

when no_data_found then

dbms_output.put_line ('Department not found.');

when others then

dbms_output.put_line ('An error occurred: ' || SQLERRM);

END;

/

Output:

```
* ed sum.sql;
```

DECLARE
 v_sum NUMBER;
BEGIN
 cal_dept_salsum('RESEARCH', v_sum);
END;

Output:

Sum of salaries in the department : 10875

PL/SQL procedure successfully executed.

VITVAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)
(Affiliated to Osmania University)

Hyderabad - 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DBMS

Name K.S.I.SIVANI

Roll No. -052

Page No.

82

* Functions:

- 1) Write a function to find the given number is palindrome or not:

Create or replace function is_palindrome(p_num IN NUMBER)

NUMBER

RETURN BOOLEAN AS

v_reverse := 0; v_temp NUMBER := p_num;

BEGIN

while v_temp > 0 LOOP

v_reverse := (v_reverse * 10) + v_temp mod 10;

v_temp := FLOOR(v_temp / 10);

END LOOP;

if p_num = v_reverse then return TRUE;

else return FALSE;

END IF; END;

DECLARE v_num NUMBER := 12321;

v_is_palindrome BOOLEAN;

BEGIN

v_is_palindrome := is_palindrome(v_num);

if v_is_palindrome then

dbms_output.put_line(v_num || 'is a palindrome');

else dbms_output.put_line(v_num || 'is not a palindrome.');

end if;

END;
/

Output:

Function created

12321 is a palindrome

75 is not a palindrome

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)
(Affiliated to Osmania University)

Hyderabad - 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DBMS

Name K.S.I.SIVANI

Roll No. -052

Page No. 83

2) Write a function to find average salary of the given department.

Create or replace function cal_avgsal (p_deptno IN NUMBER)

RETURN AS NUMBER AS

v_avgsal NUMBER;

BEGIN

select avg(sal) into v_avgsal from emp
where deptno = p_deptno;

return v_avgsal;

END;

/

DECLARE

v_dept_id NUMBER = 20;

v_avg_sal NUMBER;

BEGIN

v_avg_sal := cal_avgsal (v_dept_id);

dbms_output.put_line ('Avg salary in department'

|| v_dept_id ||': ' || v_avg_sal);

END;

/

Output:

Function created

Average salary in department 20 : 2175