## Extra Labs – Stored Procedures

My copy is uploaded ;may contain mistakes

```
drop table employees_copy
--drop table managers
create table employees_copy
as select * from pemployees

select * from employees_copy



-----------------An anonymous block example
lock for update

declare
   cursor c_emps is select * from employees_copy for update;
   v_salary_increase number:= 1.10;
   v_old_salary number;
begin
   for r_emp in c_emps loop
```

```
     v_old_salary := r_emp.salary;
     r_emp.salary := r_emp.salary*v_salary_increase + r_emp.salary *
nvl(r_emp.commission_pct,0);
     update employees_copy set row = r_emp where current of c_emps;
     dbms_output.put_line('The salary of : '|| r_emp.employee_id
                    || ' is increased from '||v_old_salary||' to '||r_emp.salary);
   end loop;
end;
```

```
---------------- Creating a procedure
-- no params   -- no declare

SET SERVEROUTPUT ON
create procedure increase_salaries as
   cursor c_emps is select * from employees_copy for update;
   v_salary_increase number := 1.10;
   v_old_salary number;
begin
   for r_emp in c_emps loop
     v_old_salary := r_emp.salary;
     r_emp.salary := r_emp.salary * v_salary_increase + r_emp.salary *
nvl(r_emp.commission_pct,0);
     update employees_copy set row = r_emp where current of c_emps;
   dbms_output.put_line('The salary of : '|| r_emp.employee_id
              || ' is increased from '||v_old_salary||' to '||r_emp.salary);
  end loop;
end;

---------------- Multiple procedure usage
begin
 dbms_output.put_line('Increasing the salaries!...');
 INCREASE_SALARIES;
 INCREASE_SALARIES;
 INCREASE_SALARIES;
 INCREASE_SALARIES;
 dbms_output.put_line('All the salaries are successfully increased!...');
end;
```

---

2nd method

```
EXECUTE INCREASE_SALARIES;
```

--

create procedure from connection

--

```
SET SERVEROUTPUT ON
create procedure new_line as
begin
  dbms_output.put_line('----------------------------------------------');
end;

----------------- Different procedures in one block
begin
  dbms_output.put_line('Increasing the salaries!...');
  INCREASE_SALARIES;
  new_line;
end;

-----------------Modifying the procedure with using the OR REPLACE command.
create or replace procedure increase_salaries as
    cursor c_emps is select * from employees_copy for update;
    v_salary_increase number := 1.10;
    v_old_salary number;
begin
    for r_emp in c_emps loop
      v_old_salary := r_emp.salary;
      r_emp.salary := r_emp.salary * v_salary_increase + r_emp.salary * nvl(r_emp.commission_pct,0);
      update employees_copy set row = r_emp where current of c_emps;
      dbms_output.put_line('The salary of : '|| r_emp.employee_id
                        || ' is increased from '||v_old_salary||' to '||r_emp.salary);
    end loop;
    dbms_output.put_line('Procedure finished executing!');
end;

----------------------------------------------------------------------------------------------------------------
```

## IN OUT Parameters

### IN Param

```
-----------------Creating a procedure with the IN parameters
SET SERVEROUTPUT ON;
create or replace procedure increase_salaries (v_salary_increase in number, v_department_id pls_integer) as
    cursor c_emps is select * from employees_copy where department_id = v_department_id for update;
    v_old_salary number;
begin
    for r_emp in c_emps loop
      v_old_salary := r_emp.salary;
      r_emp.salary := r_emp.salary * v_salary_increase + r_emp.salary * nvl(r_emp.commission_pct,0);
      update employees_copy set row = r_emp where current of c_emps;
```

```
   dbms_output.put_line('The salary of : '|| r_emp.employee_id
              || ' is increased from '||v_old_salary||' to '||r_emp.salary);
  end loop;
  dbms_output.put_line('Procedure finished executing!');
end;
```

```
call
begin
  increase_salaries;
increase_salaries(1.2,60);
increase_salaries(1.15,90);
end;
```

```
---------------- Creating a procedure with the OUT parameters

CREATE OR REPLACE PROCEDURE PRINT(TEXT IN VARCHAR2) IS
BEGIN
 DBMS_OUTPUT.PUT_LINE(TEXT);
END;
```

---

## OUT Param IN /OUT

```
SET SERVEROUTPUT ON;
create or replace procedure increase_salaries
   (v_salary_increase in out number, v_department_id pls_integer, v_affected_employee_count out number) as
   cursor c_emps is select * from employees_copy where department_id = v_department_id for update;
   v_old_salary number;
   v_sal_inc number := 0;
begin
   v_affected_employee_count := 0;
   for r_emp in c_emps loop
     v_old_salary := r_emp.salary;
     r_emp.salary := r_emp.salary * v_salary_increase + r_emp.salary * nvl(r_emp.commission_pct,0);
     update employees_copy set row = r_emp where current of c_emps;
     dbms_output.put_line('The salary of : '|| r_emp.employee_id
                       || ' is increased from '||v_old_salary||' to '||r_emp.salary);
     v_affected_employee_count := v_affected_employee_count + 1;
     v_sal_inc := v_sal_inc + v_salary_increase + nvl(r_emp.commission_pct,0);
   end loop;
   v_salary_increase := v_sal_inc / v_affected_employee_count;
   dbms_output.put_line('Procedure finished executing!');
end;
```

```
-----------------
```

```
-----------------Using the procedure that has OUT parameters
declare
  v_sal_inc number := 1.2;
  v_aff_emp_count number;
begin
 PRINT('SALARY INCREASE STARTED!..');
 INCREASE_SALARIES(v_sal_inc,80,v_aff_emp_count);
 PRINT('The affected employee count is : '|| v_aff_emp_count);
 PRINT('The average salary increase is : '|| v_sal_inc || ' percent!..');
 PRINT('SALARY INCREASE FINISHED!..');
end;
```

---

## default params

```
----------------- A standard procedure creation with a default value
create or replace PROCEDURE PRINT(TEXT IN VARCHAR2 := 'This is the print function!.') IS
BEGIN
  DBMS_OUTPUT.PUT_LINE(TEXT);
END;
```

```
----------------Executing a procedure without any parameter. It runs because it has a default value.

exec print();

----------------Running a procedure with null value will not use the default value

exec print(null);




----------------Procedure creation of a default value usage

create or replace procedure add_job(job_id varchar2, job_title varchar2,

                          min_salary number default 1000, max_salary number default null) is

begin

  insert into jobs values (job_id,job_title,min_salary,max_salary);

  print('The job : '|| job_title || ' is inserted!..');

end;

----------------A standard run of the procedure

EXECUTE ADD_JOB('IT_DIR','IT Director',5000,20000);

----------------Running a procedure with using the default values

exec ADD_JOB('IT_DIR2','IT Director',5000);


select * from jobs;

----------------Running a procedure with the named notation

exec ADD_JOB('IT_DIR5','IT Director',max_salary=>10000);

----------------Running a procedure with the named notation example 2
```

```
exec ADD_JOB(job_title=>'IT Director',job_id=>'IT_DIR7',max_salary=>10000 , min_salary=>500);
```

----------------------------------------------------------------------