

1/ Donner une fonction qui permet de retourner le salaire maximum pour un département donné (compte SCOTT)

```
create or replace function getmaxsal(vdept dept.deptno%type)
return number is
v1 number;
begin
select max(sal) into v1 from emp where
deptno=vdept;
return v1;
exception when no_data_found then return -1;
end;
```

Appel de la fonction: select getmaxsal(20) from dual;

=====

2/ Donner une procédure qui permet de majorer le salaire des employés d'un département donné avec un taux donné (compte SCOTT) *deptno, taux*

```
create or replace procedure setsal
(vdept dept.deptno%type,taux number) is
begin
update emp set sal=sal*taux where deptno=vdept;
commit;
end;
/
```

Appel de la procédure : execute setsal (20,1.2);

```
Sql> select object_name from user_objects where
object_type='PROCEDURE';
```

```
Sql> select text from user_source where
name='SETSAL' order by line;
```

3/ Donner une procédure qui permet de créer un utilisateur donné (paramètre nom utilisateur et password)

```
create or replace procedure creuser(name varchar2,pwd
varchar2) is
begin
execute immediate 'create user ' || name || ' identified by
' || pwd;
execute immediate 'grant connect,resource to ' || name ;
end;
/
execute creuser('test','test');
```

4/ Donner un trigger qui interdit toute opération lundi sur la table DEPT (

create or replace trigger nowork before insert or update or delete on
departments

declare v number;

begin

select to_char(sysdate,'d') into v from dual;

if v=1 then

raise_application_error(-20500, 'no work');

end if;

end;

/

Déclencheur créé.

Delete from departments

ORA-20500: no work

Créer une fonction qui permet de retourner la moyenne des salaires les plus basses par département ; et qui permet d'afficher un message indiquant si la moyenne des salaires d'un employé donné passé en paramètres à cette fonction est > ou < ou = à la moyenne des salaires la plus basse

```
set serveroutput on
```

```
CREATE OR REPLACE FUNCTION F(nom emp.ename%type) RETURN emp.sal%type
```

```
IS
```

```
    moyen emp.sal%type ;
```

```
    v emp.ename%type;
```

```
    ndep emp.deptno%type;
```

```
cursor c is select avg(sal) from emp where deptno = ndep;
```

```
BEGIN
```

```
    SELECT deptno INTO ndep FROM emp WHERE ename = nom ;
```

```
    SELECT min(avg(sal)) INTO moyen FROM emp GROUP BY deptno;
```

```
open c;
```

```
fetch c into v;
```

```
    if(v <= moyen)
```

```
    then
```

```
        DBMS_output.put_line('le salaire est inferieur');
```

```
    end if;
```

```
    if(v >= moyen)
```

```
    then
```

```
        DBMS_output.put_line('le salaire est superieur');
```

```
    end if;
```

```
close c;
```

```
RETURN(moyen) ;
```

```
END ;
```