-- A faire une fois par session

set serveroutput on;

-- Tables

drop table employees;

create table employees as select \* from hr.employees;

drop table departments;

create table departments as select \* from hr.departments;

drop table jobs;

create table jobs as select \* from hr.jobs;

clear screen;

-- Question 1

create or replace procedure ajout\_job (p\_id Varchar2, p\_title Varchar2) is

begin

insert into jobs (job\_id, job\_title)

values (p\_id, p\_title);

commit;

end;

/

exec ajout\_job('PR', 'Professeur');

-- Question 2

create or replace procedure modif\_job (p\_id varchar2, p\_newtitle varchar2) is

e\_nojob exception;

begin

update jobs

set job\_title = p\_newtitle

where job\_id = p\_id;

if (sql%rowcount = 0) then

raise e\_nojob;

end if;

commit;

exception

when e\_nojob then

raise\_application\_error(-20002, 'Le job ' || p\_id || ' est inexistant.');

end;

/

exec modif\_job('PR', 'Professionnel');

-- Question 3

create or replace procedure liste\_emp\_mgr is

cursor cr\_emp is

select e1.first\_name as emp\_fn, e1.last\_name as emp\_ln, e2.first\_name as mgr\_fn, e2.last\_name as mgr\_ln

from employees e1, employees e2

where e2.employee\_id = e1.manager\_id;

begin

dbms\_output.put\_line('Employé - Manager');

dbms\_output.put\_line('-------------------');

for v\_emp in cr\_emp loop

dbms\_output.put\_line(v\_emp.emp\_fn || ' ' || v\_emp.emp\_ln || ' - ' || v\_emp.mgr\_fn || ' ' || v\_emp.mgr\_ln);

end loop;

dbms\_output.put\_line(' ');

end;

/

exec liste\_emp\_mgr;

-- Question 4

create or replace procedure liste\_emp\_make\_more\_than (p\_name varchar2) is

cursor cr\_emp (p\_name varchar2) is

select last\_name

from employees

where salary + decode(commission\_pct, null, 0, commission\_pct \* salary) > (

select salary + decode(commission\_pct, null, 0, commission\_pct \* salary)

from employees

where last\_name = p\_name);

begin

dbms\_output.put\_line('Nom');

dbms\_output.put\_line('----------');

for v\_emp in cr\_emp(p\_name) loop

dbms\_output.put\_line(v\_emp.last\_name);

end loop;

dbms\_output.put\_line(' ');

end;

/

exec liste\_emp\_make\_more\_than('Russell');

-- Question 5

create or replace procedure liste\_same\_job\_make\_more\_than (p\_name1 varchar2, p\_name2 varchar2) is

cursor cr\_emp (p\_name2 varchar2) is

select last\_name

from employees

where job\_id in (

select job\_id

from employees

where last\_name = p\_name1)

and salary > (

select salary

from employees

where last\_name = p\_name2);

begin

dbms\_output.put\_line('Nom');

dbms\_output.put\_line('----------');

for v\_emp in cr\_emp(p\_name2) loop

dbms\_output.put\_line(v\_emp.last\_name);

end loop;

dbms\_output.put\_line(' ');

end;

/

exec liste\_same\_job\_make\_more\_than('Kochhar', 'Fox');

-- Question 6

create or replace procedure liste\_emp\_bigger\_sal (p\_n number) is

cursor cr\_emp(p\_n number) is

select \* from (

select last\_name

from employees

order by salary desc)

where rownum <= p\_n;

begin

dbms\_output.put\_line('Nom');

dbms\_output.put\_line('----------');

for v\_emp in cr\_emp(p\_n) loop

dbms\_output.put\_line(v\_emp.last\_name);

end loop;

dbms\_output.put\_line(' ');

end;

/

exec liste\_emp\_bigger\_sal(5);

-- Question 7

create or replace procedure depts\_without\_emp is

cursor cr\_dept is

select department\_name as dname

from departments

where department\_id not in (

select distinct decode(department\_id, null, 0, department\_id)

from employees)

order by department\_name desc;

begin

dbms\_output.put\_line('Département');

dbms\_output.put\_line('-----------');

for v\_dept in cr\_dept loop

dbms\_output.put\_line(v\_dept.dname);

end loop;

dbms\_output.put\_line(' ');

end;

/

exec depts\_without\_emp;

-- Question 8

create or replace procedure emp\_ranking (p\_n number) is

cursor cr\_emp(p\_n number) is

select \* from (

select last\_name

from employees

where employee\_id >= p\_n

order by employee\_id asc);

begin

dbms\_output.put\_line('Nom');

dbms\_output.put\_line('----------');

for v\_emp in cr\_emp(p\_n) loop

dbms\_output.put\_line(v\_emp.last\_name);

end loop;

dbms\_output.put\_line(' ');

end;

/

exec emp\_ranking(200);

-- Question 9

drop view dept\_salary;

create view dept\_salary as (

select departments.department\_id, department\_name, sum(salary) as sum\_sal

from departments, employees

where departments.department\_id = employees.department\_id

group by departments.department\_id, department\_name);

create or replace procedure sum\_sal\_dept\_greater\_than (p\_n number) is

cursor cr\_dept\_sal(p\_n number) is

select \* from (

select department\_id, department\_name

from dept\_salary

where sum\_sal >= p\_n

order by department\_id asc);

begin

dbms\_output.put\_line('Identifiant - Nom');

dbms\_output.put\_line('-------------------');

for v\_dept\_sal in cr\_dept\_sal(p\_n) loop

dbms\_output.put\_line(v\_dept\_sal.department\_id || ' - ' || v\_dept\_sal.department\_name);

end loop;

dbms\_output.put\_line(' ');

end;

/

exec sum\_sal\_dept\_greater\_than (10);

-- Question 10

create or replace procedure list\_emp\_bigger\_sal\_than\_avg is

cursor cr\_emp\_avg\_sal is

select last\_name, dept\_id, salary, avg\_sal from employees, (

select nvl(department\_id,0) as dept\_id, round(avg(salary),2) as avg\_sal

from employees

group by nvl(department\_id,0))

where dept\_id = employees.department\_id

and salary >= avg\_sal

order by dept\_id;

begin

dbms\_output.put\_line('Nom');

dbms\_output.put\_line('----------');

for v\_emp in cr\_emp\_avg\_sal loop

dbms\_output.put\_line(v\_emp.last\_name);

end loop;

dbms\_output.put\_line(' ');

end;

/

exec list\_emp\_bigger\_sal\_than\_avg;

-- Question 11 - Apparemment faut le faire avec CASE.

create or replace function check\_sal(p\_empno employees.employee\_id%type)

return Boolean is

v\_dept\_id employees.department\_id%type;

v\_sal employees.salary%type;

v\_avg\_sal employees.salary%type;

begin

select salary, department\_id into v\_sal, v\_dept\_id from employees

where employee\_id = p\_empno;

select avg(salary) into v\_avg\_sal from employees

where department\_id = v\_dept\_id;

if v\_sal > v\_avg\_sal then

return true;

else

return false;

end if;

exception

when no\_data\_found then

return null;

end;

/

begin

dbms\_output.put\_line('Vérification pour l''employé avec l''ID 205.');

if (check\_sal(205) is null) then

dbms\_output.put\_line('La fonction a renvoyé NULL à cause d''une exception.');

elsif (check\_sal(205)) then

dbms\_output.put\_line('Salaire > moyenne des salaires');

else

dbms\_output.put\_line('Salaire < moyenne des salaires');

end if;

end;

/