Exercici 1: (ESQUEMA PRODUCTES)

Realitza una función anomenada “bonus_salesman_cachinero_marc”. Una funció per calcular el bonus a un vendedor en funció de les seves comandes.

Els paràmetres de la funció son 2 i no poden ser null:

- un id de treballador, exemple: 103
- un any, exemple 2015

Cal validar els paràmetres:

- que la id correspongui a un treballador i que el treballador sigui ‘Sales Manager o Sales Representative’.
- que l’any sigui no més gran que l’any actual o sigui posterior al 2000.

Si els paràmetres no son correctes la funció retorna null,

Si els paràmetres son correctes, cal cercar comandes:

- comandes on el venedor sigui l’id que pasa com a paràmetre.
- realitzades a l’any que indica el paràmetre.

El vendedor obté un bonus de 150$ per comanda, llavors el bonus total es el resultat de:

- 150 * total_comandes

Com a resultat de la funció cal retornar el bonus obtingut per venedor.

SCRIPT DE PROVES:

```
set serveroutput on
declare
result number;
id_emp number;
any_vendes number;
begin

---paràmetre null
id_emp:=null;
any_vendes:=1995;
DBMS_OUTPUT.PUT_LINE('---------- Crida amb paràmetre null');
result:=bonus_salesman(id_emp,any_vendes);
DBMS_OUTPUT.PUT_LINE('-- el bonus del treballador '||nvl(to_char(id_emp),'null')||' l''any
'||any_vendes
                                        ||' és :'||nvl(to_char(result),'null'));
---any incorrecte
id_emp:=54;
any_vendes:=2025;
DBMS_OUTPUT.PUT_LINE('---------- any incorrecte');
result:=bonus_salesman(id_emp,any_vendes);
DBMS_OUTPUT.PUT_LINE('-- el bonus del treballador '||nvl(to_char(id_emp),'null')||' l''any
'||any_vendes
                                        ||' és :'||nvl(to_char(result),'null'));
---empleat incorrecte
id_emp:=107;
```

```
any_vendes:=2017;
DBMS_OUTPUT.PUT_LINE('---------- empleat incorrecte');
result:=bonus_salesman(id_emp,any_vendes);
DBMS_OUTPUT.PUT_LINE('-- el bonus del treballador '||nvl(to_char(id_emp),'null')||' l''any
'||any_vendes
                                        ||' és :'||nvl(to_char(result),'null'));
---empleat correcte
id_emp:=54;
any_vendes:=2017;
DBMS_OUTPUT.PUT_LINE('---------- empleat correcte');
result:=bonus_salesman(id_emp,any_vendes);
DBMS_OUTPUT.PUT_LINE('-- el bonus del treballador '||nvl(to_char(id_emp),'null')||' l''any
'||any_vendes
                                        ||' és :'||nvl(to_char(result),'null'));
end;
```

```
CREATE OR REPLACE FUNCTION bonus_salesman_cachinero_marc (
    p_employee_id NUMBER,
    p_year NUMBER
) RETURN NUMBER IS
    v_bonus NUMBER;
    v_total_orders NUMBER;
    v_employee_job_title VARCHAR2(255);
    v_current_year NUMBER := EXTRACT(YEAR FROM SYSDATE);
BEGIN
    -- Validate employee_id and get the job title
    SELECT job_title
    INTO v_employee_job_title
    FROM employees
    WHERE employee_id = p_employee_id;

    -- Check if the job title is 'Sales Manager' or 'Sales Representative'
    IF v_employee_job_title NOT IN ('Sales Manager', 'Sales Representative') THEN
        RETURN NULL;
    END IF;

    -- Validate the year
    IF p_year > v_current_year OR p_year < 2000 THEN
        RETURN NULL;
    END IF;

    -- Calculate the total number of orders for the given employee and year
    SELECT COUNT(*)
    INTO v_total_orders
    FROM orders
    WHERE salesman_id = p_employee_id
      AND EXTRACT(YEAR FROM order_date) = p_year;

    -- Calculate the bonus
    v_bonus := 150 * v_total_orders;

    RETURN v_bonus;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        -- If no employee is found with the given ID
        RETURN NULL;
    WHEN OTHERS THEN
        -- Handle other exceptions
        RETURN NULL;
END;/
```

SCRIPT DE PRUEBAS:

```
SET SERVEROUTPUT ON;

DECLARE
   result NUMBER;
   id_emp NUMBER;
   any_vendes NUMBER;
BEGIN
   -- Test case: Parameter null
   id_emp := NULL;
   any_vendes := 1995;
   DBMS_OUTPUT.PUT_LINE('--------------- Crida amb paràmetre null');
   result := bonus_salesman_cachinero_marc(id_emp, any_vendes);
   DBMS_OUTPUT.PUT_LINE('-- el bonus del treballador ' || NVL(TO_CHAR(id_emp),
'null') || ' l''any ' || any_vendes || ' és: ' || NVL(TO_CHAR(result), 'null'));

   -- Test case: Incorrect year
   id_emp := 54;
   any_vendes := 2025;
   DBMS_OUTPUT.PUT_LINE('--------------- any incorrecte');
   result := bonus_salesman_cachinero_marc(id_emp, any_vendes);
   DBMS_OUTPUT.PUT_LINE('-- el bonus del treballador ' || NVL(TO_CHAR(id_emp),
'null') || ' l''any ' || any_vendes || ' és: ' || NVL(TO_CHAR(result), 'null'));

   -- Test case: Incorrect employee
   id_emp := 107;
   any_vendes := 2017;
   DBMS_OUTPUT.PUT_LINE('--------------- empleat incorrecte');
   result := bonus_salesman_cachinero_marc(id_emp, any_vendes);
   DBMS_OUTPUT.PUT_LINE('-- el bonus del treballador ' || NVL(TO_CHAR(id_emp),
'null') || ' l''any ' || any_vendes || ' és: ' || NVL(TO_CHAR(result), 'null'));

   -- Test case: Correct employee
   id_emp := 54;
   any_vendes := 2017;
   DBMS_OUTPUT.PUT_LINE('--------------- empleat correcte');
   result := bonus_salesman_cachinero_marc(id_emp, any_vendes);
   DBMS_OUTPUT.PUT_LINE('-- el bonus del treballador ' || NVL(TO_CHAR(id_emp),
'null') || ' l''any ' || any_vendes || ' és: ' || NVL(TO_CHAR(result), 'null'));
END;
/
```

Exercici 2: (ESQUEMA_HR)

Realitzar el procediment anomenat “emp_dep_cachinero_marc” que llisti els treballadors d’un departament.

- El parametre de entrada del procediment es:
    - id, identificador d’un departament, no pot ser null.

  Cal validar el paràmetre
    - que l’ ID no sigui null.
    - que l’ ID correspongui a un departament que existeixi.

  si el parametre no es correcte el procediment acabara sense llistar res.

  Si el parametre es correcte, es treura una llista que tindra una capçelera amb el nom del departament a continuacio la llista de treballadors. Cal tenir present el cas d’un departament sense treballadors, llavors nomes es llistara la capçelera amb el nom del departament.

  Les dades a llistar de cada empleat son les seguents:
    - FIRST_NAME, LAST_NAME, HIRE_DATE, JOB_TITLE.

  Les dades s’han de mostrar en columnes de mida fixa,excepte la capçelera.

  Exemple del que s’ha de mostrar:

  Departament <nom del departament>

  Bruce Ernst 21-05-2007 Programmer
  SCRIPT DE PROBES:

```
declare
    v_id number;
    begin
    -- prova departament null
    DBMS_OUTPUT.PUT_LINE('-- prova departament null----------');
    v_id:=null;
    emp_dept(v_id);
    DBMS_OUTPUT.PUT_LINE('');
    -- prova departament que NO existeix
    DBMS_OUTPUT.PUT_LINE('-- prova prova departament que NO existeix----------');
    v_id:=333;
    emp_dept(v_id);
    DBMS_OUTPUT.PUT_LINE('');
    -- prova departament sense treballadors
    DBMS_OUTPUT.PUT_LINE('-- prova departament sense treballadors----------');
    v_id:=180;
    emp_dept(v_id);
```

```
        DBMS_OUTPUT.PUT_LINE('');
        -- prova departament amb treballadors
        DBMS_OUTPUT.PUT_LINE('-- prova departament amb
treballadors-----------------------');
        v_id:=30;
        emp_dept(v_id);
        DBMS_OUTPUT.PUT_LINE('');
        end;
```

```
        CREATE OR REPLACE PROCEDURE emp_dep_cachinero_marc (p_id IN
departments.department_id%TYPE)
        IS
          -- Variables per emmagatzemar informació del departament
          v_department_name departments.department_name%TYPE;
          v_count          NUMBER;

          -- Cursor per llistar els empleats del departament
          CURSOR emp_cursor IS
            SELECT e.first_name, e.last_name, TO_CHAR(e.hire_date, 'DD-MM-YYYY')
AS hire_date, j.job_title
            FROM employees e
            JOIN jobs j ON e.job_id = j.job_id
            WHERE e.department_id = p_id;

        BEGIN
          -- Comprovar si el paràmetre d'entrada no és nul
          IF p_id IS NULL THEN
            DBMS_OUTPUT.PUT_LINE('El paràmetre id no pot ser null.');
            RETURN;
          END IF;

          -- Comprovar si el departament existeix
          BEGIN
            SELECT department_name INTO v_department_name
            FROM departments
            WHERE department_id = p_id;
          EXCEPTION
            WHEN NO_DATA_FOUND THEN
              DBMS_OUTPUT.PUT_LINE('El departament amb id ' || p_id || ' no existeix.');
              RETURN;
          END;

          -- Comprovar si el departament té treballadors
          SELECT COUNT(*) INTO v_count
```

```
        FROM employees
        WHERE department_id = p_id;

        -- Mostrar el nom del departament
        DBMS_OUTPUT.PUT_LINE('Departament: ' || v_department_name);
        DBMS_OUTPUT.PUT_LINE('-------------------------------');

        -- Si el departament té treballadors, llistar-los
        IF v_count > 0 THEN
          FOR emp IN emp_cursor LOOP
            DBMS_OUTPUT.PUT_LINE(RPAD(emp.first_name, 12) ||
RPAD(emp.last_name, 12) || RPAD(emp.hire_date, 12) || emp.job_title);
          END LOOP;
        END IF;

      END emp_dep_cachinero_marc;
      /




SCREIPT PRUEBAS
DECLARE
  v_id NUMBER;
BEGIN
  -- prova departament null
  DBMS_OUTPUT.PUT_LINE('-- prova departament null-----------------------------------');
  v_id := NULL;
  emp_dep_cachinero_marc(v_id);
  DBMS_OUTPUT.PUT_LINE('');

  -- prova departament que NO existeix
  DBMS_OUTPUT.PUT_LINE('-- prova departament que NO existeix-------------------------');
  v_id := 333;
  emp_dep_cachinero_marc(v_id);
  DBMS_OUTPUT.PUT_LINE('');

  -- prova departament sense treballadors
  DBMS_OUTPUT.PUT_LINE('-- prova departament sense treballadors----------------------');
  v_id := 180;
  emp_dep_cachinero_marc(v_id);
  DBMS_OUTPUT.PUT_LINE('');

  -- prova departament amb treballadors
  DBMS_OUTPUT.PUT_LINE('-- prova departament amb treballadors------------------------');
  v_id := 30;
  emp_dep_cachinero_marc(v_id);
  DBMS_OUTPUT.PUT_LINE('');
```

END;
/

Exercici 3: (esquema HR)

Crea tres paquets amb el conjunt de funcions que trobaràs en el fitxer que acompaña a l'examen:

- **funcions_employees0523.txt**

En el fitxer funcions_emloyees0523.txt hi ha functions independents que s'han de agruapar dins de tres paquets que s'han de dir:

- **pkg_employees0523_cachinero_marc**
- **pkg_jobs0523_cachinero_marc**
- **pkg_hr_utilitats0523_cachinero_marc**

Al paquet pkg_employees0523_cachinero_marc s'han de posar les funcions <<f_empleat….>>.
Al paquet pkg_jobs0523_cachinero_marc s'han de posar les funcions <<f_job…>>
Al paquet pkg_hr_utilitats0523_cachinero_marc s'ha de posar la funcio <<normalizar_string>>.

Cal crear les dues parts del paquet: especificació i cos. A part d'especificació només s'han de declarar les funcions que estan definides com a publiques,les que estan encapçalades amb el comentari <<--FUNCIÓ PÚBLICA>>.

Una vegada compilats els paquets cal fer proves, per aixo teens l'altre fitxer:

- **funcions script_proves0523.txt**

Hay que adaptar el script para llamar las funciones que creaste

scrript normal:

```
CREATE OR REPLACE PACKAGE pkg_ht_utilitats0523_Cachinero_Marc AS
function normalitzar_string
(p_string varchar2,p_convertir boolean)
return varchar2;
END pkg_ht_utilitats0523_Cachinero_Marc;
/
CREATE OR REPLACE PACKAGE BODY pkg_hr_utilitats0523_Cachinero_Marc AS
function normalitzar_string
(p_string varchar2,p_convertir boolean)
return varchar2
is
v_string varchar2(50);
v_string_out varchar2(50);
v_paraula varchar2(50);
v_paraula_aux varchar2(50);
v_pos number;
primera_vegada boolean;
-- agafa l'string d'entrada, elimina blancs a dreta i esquerra
-- i posa el primer caracter e majuscula i la resta en minúscula
begin
-- comprovar que no sigui null
if p_string is null then return null;
end if;
-- comprovar que la longitud no sigui més de 50
if length(trim(p_string))>50 then return null;
end if;

-- elimina blancs a dreta i esquerra
v_string:=trim(p_string);

-- agafar cada paraula i convertir-la
primera_vegada:=true;
while v_string is not null loop
-- busquem la posició del blanc, si no n'hi ha la posició és el final de l'string
v_pos:= instr(v_string,' ',1,1);
if v_pos=0 then v_pos:=length(v_string)+1; end if;

-- fem subtring per trobar la paraula
v_paraula:=substr(v_string,1,v_pos-1);

-- CONVERTIM la primera lletra a majuscula i la resta a minúscula
-- si l'opció convertir és true
if p_convertir then
if length(v_paraula)>1 then
v_paraula_aux:=upper(substr(v_paraula,1,1))||lower(substr(v_paraula,2));
else
v_paraula_aux:=upper(v_paraula);
```

```
end if;
else
v_paraula_aux:=v_paraula;
end if;

-- concatanem la paraula a l'string de sortida
if primera_vegada then
v_string_out:= v_paraula_aux;
primera_vegada:=false;
else
v_string_out:= v_string_out||' '||v_paraula_aux;
end if;

-- escurcem l'string per continuar buscant paraules a convertir,
-- aprofitem per eliminar blancs a l'esquerra
v_string:=ltrim(substr(v_string,v_pos+1));
end loop;

return v_string_out;
end normalitzar_string;
END pkg_hr_utilitats0523_Cachinero_Marc;
/

CREATE OR REPLACE PACKAGE pkg_employees0523_Cachinero_Marc AS

function f_job(nom jobs.job_title%type)
return jobs.job_id%type;

function f_empleat_per_cognom
(p_dept departments.department_id%type,
p_cognom employees.last_name%type)
return employees%rowtype;

function f_empleat_per_cognom_i_nom
(p_dept departments.department_id%type,
p_cognom employees.last_name%type,
p_nom employees.first_name%type)
return employees%rowtype;

END pkg_employees0523_Cachinero_Marc;
/
CREATE OR REPLACE PACKAGE BODY pkg_employees0523_Cachinero_Marc AS

function f_job_excep(nom jobs.job_title%type)
return jobs.job_id%type
is
v_id jobs.job_id%type;
begin
```

```
select job_id into v_id
from jobs where job_title=nom;
return v_id;
exception
when no_data_found then
return null;
when too_many_rows then
return null;
when others then
return -1;
end f_job_excep;

function f_job(nom jobs.job_title%type)
return jobs.job_id%type
is
begin
return
f_job_excep(pkg_hr_utilitats0523_Cachinero_Marc.normalitzar_string(nom,true));
end f_job;

function f_empleat_per_cognom
(p_dept departments.department_id%type,
p_cognom employees.last_name%type)
return employees%rowtype
is
v_emp employees%rowtype;
begin
v_emp:= f_empleat_per_cognom_i_nom(p_dept,p_cognom,null);
return v_emp;
end f_empleat_per_cognom;

function f_empleat_per_cognom_i_nom
(p_dept departments.department_id%type,
p_cognom employees.last_name%type,
p_nom employees.first_name%type)
return employees%rowtype
is
v_emp employees%rowtype;
v_convertir boolean:=true;
begin
begin
select * into v_emp
from employees where department_id=p_dept
and
last_name=pkg_hr_utilitats0523_Cachinero_Marc.normalitzar_string(p_cognom,v_con
vertir)
```

```
and
first_name=nvl(pkg_hr_utilitats0523_Cachinero_Marc.normalitzar_string(p_nom,v_co
nvertir),first_name);
exception
when no_data_found then v_emp.employee_id:=-1;
when too_many_rows then v_emp.employee_id:=-99;
when others then v_emp.employee_id:=-10;
end;
return v_emp;
end f_empleat_per_cognom_i_nom;

END pkg_employees0523_Cachinero_Marc;
```

SCRIPTPROVAS:

```
--SCRIPT DE PROVES f_empleat_per_cognom_i_nom
set SERVEROUTPUT on
declare
v_emp employees%rowtype;
begin

v_emp:=f_empleat_per_cognom_i_nom(100,'       CHEN','JOHN         ');
dbms_output.put_line('treballador: '||v_emp.employee_id);

v_emp:=f_empleat_per_cognom(100,'Chen          ');
dbms_output.put_line('treballador: '||v_emp.employee_id);

v_emp:=f_empleat_per_cognom_i_nom(80,'Cambrault','Gerald');
dbms_output.put_line('treballador: '||v_emp.employee_id);

v_emp:=f_empleat_per_cognom(80,'Cambrault');
dbms_output.put_line('treballador: '||v_emp.employee_id);
end;

/

--SCRIPT DE PROVES f_job
set SERVEROUTPUT on
declare
v_id jobs.job_id%type;
begin

v_id:=f_job('President');
dbms_output.put_line('job id és: '||v_id);

v_id:=f_job('  President ');
dbms_output.put_line('job id és: '||v_id);
```

```
v_id:=f_job(' accounting     manager');
dbms_output.put_line('job id és: '||v_id);
end;
```