

## CORRIGE TP N°: 3

### LES FPROCEDURES, LES FONCTIONS & LES EXCEPTIONS

ENSEIGNANTE : MME ONS BEN ROMDHANE

SEMESTRE : 2 - 2022/2023

MATIERE : ATELIERS SGBD

GROUPE : DSI2.2 G2

#### EXERCICE 1 : EXCEPTIONS PREDEFINIES

```
set serveroutput on;

DECLARE

    name varchar2(50);

    emp_id employees.employee_id%TYPE ;

    no_result exception;

BEGIN

    emp_id := 0 ;

    -- get the employee name by id

    SELECT first_name || ' ' || last_name INTO name

    FROM employees

    WHERE employee_id = emp_id;

    -- show the employee name

    dbms_output.put_line('Employee name is ' || name);

EXCEPTION

    when no_data_found then

        dbms_output.put_line('Aucun employé avec cet
        identifiant');

    when too_many_rows then

        dbms_output.put_line('Trop de lignes');

END;
```

## EXERCICE 2 : FONCTION STOCKEE + EXCEPTION

--1)

```
create or replace function get_fullname(empid in
employees.employee_id%type)
return varchar2 is
    nameemp varchar2(50);
begin
    select first_name || ' ' || last_name into nameemp from
employees where employee_id = empid;
    return nameemp;
end ;
```

--2)

```
begin
    DBMS_OUTPUT.PUT_LINE(get_fullname(107));
end;
```

--3)

```
create or replace function get_fullname(empid in
employees.employee_id%type)
return varchar2 is
    nameemp varchar2(50);
begin
    select first_name || ' ' || last_name into nameemp from
employees where employee_id = empid;
    --ça ne marche pas (car nameemp aura la valeur null)
    --if sql%notfound then --jamais vérifiée
    --    nameemp:= 'Not Found !';
    --end if;
    return nameemp;
exception
    when NO_DATA_FOUND then
        return 'Not Found !';
end;
```

--4)

```
create or replace function get_fullname(empid in
hr.employees.employee_id%type)
return varchar2 is
    nameemp varchar2(50);
begin
    select first_name || ' ' || last_name into nameemp
```

```

        from hr.employees where employee_id = empid;
        return nameemp;
exception
    when NO_DATA_FOUND then
        raise_application_error(-20000,'Not Found !');
end;

begin
    DBMS_OUTPUT.PUT_LINE(get_fullname(1007));
exception
    when others then
        DBMS_OUTPUT.PUT_LINE(sqlcode || sqlerrm);
end;

```

### EXERCICE 3 : PROCEDURE STOCKEE

```

CREATE OR REPLACE PROCEDURE topk(k NUMBER) IS
    CURSOR ranking IS
        SELECT last_name, salary FROM HR.EMPLOYEES
        ORDER BY salary DESC;
    p ranking%ROWTYPE;
BEGIN
    OPEN ranking;
    FETCH ranking INTO p; -- Premier produit
    WHILE ranking%FOUND AND ranking%ROWCOUNT <= k LOOP
        DBMS_OUTPUT.PUT_LINE(ranking%ROWCOUNT || ' ' ||
            rpad(p.last_name,8,' ') || ' : ' || p.salary || ' €');
        FETCH ranking INTO p; -- Produit suivant
    END LOOP;
    CLOSE ranking;
END;

-- Exécution sous SQL Developer
EXECUTE topk(5)

-- Exécution dans un bloc PL/SQL
BEGIN
    topk(150);
END;

```

## EXERCICE 4 : CURSEUR EXPLICITE ET EXCEPTION

```
DECLARE
    CURSOR commission IS
        SELECT commission_pct*salary prime FROM HR.employees
WHERE commission_pct IS NOT NULL;
    prec commission%ROWTYPE; -- n-uplet précédent
    cour commission%ROWTYPE; -- n-uplet courant
    total REAL := 0;
    n_val INTEGER; pas_assez EXCEPTION;
BEGIN
    -- Test du nombre de commissions valuées
    SELECT COUNT(*) INTO n_val FROM HR.employees WHERE
commission_pct IS NOT NULL;
    IF n_val < 2 THEN
        RAISE pas_assez;
    END IF;
    -- Accès 1er n-uplet
    OPEN commission;
    FETCH commission INTO prec; -- 1er précédent
    -- Accès aux suivants et cumul
    FETCH commission INTO cour; -- 1er courant
    WHILE commission%FOUND LOOP
        total := total + ABS(cour.prime - prec.prime);
        prec := cour; -- Le courant devient le précédent
        FETCH commission INTO cour; -- Courant suivant
    END LOOP;
    CLOSE commission;
    -- Calcul et affichage de la moyenne
    DBMS_OUTPUT.PUT_LINE('Moyenne Prime = ' ||
        to_char(total / (n_val - 1), '0999.99') || ' $');
EXCEPTION
    WHEN pas_assez THEN
        RAISE_APPLICATION_ERROR(-20501, 'Pas assez de
commissions !');
END;
```