Ecole Supérieure
d'Informatique et du Numérique
COLLEGE OF ENGINEERING & ARCHITECTURE

COMPTE RENDU

Bases de Données Avancées - Lab No. 4
3e année Cybersécurité - École Supérieure d'Informatique et du Numérique (ESIN)
Collège d'Ingénierie & d'Architecture (CIA)

Étudiant : HATHOUTI Mohammed Taha

Filière : Cybersecurité

Année : 2025/2026

Enseignants : Mme.ELHAJI & M.HJJAMI

Date : 16 octobre 2025

Table des matières

1 Partie 1 : Structures de Contrôle de Base - Boucles et Conditions

1.1 Exercice 1 : Utilisation des Boucles

Écrivez un bloc PL/SQL anonyme qui :

— Affiche tous les nombres de 1 à 10 en utilisant une boucle FOR ;

— Pour chaque nombre, affiche s'il est pair ou impair ;

Astuce : utilisez `MOD(i,2)` et `DBMS_OUTPUT.PUT_LINE`.

Code PL/SQL :

```
SET SERVEROUTPUT ON

BEGIN
    FOR compteur IN 1..10 LOOP
        IF MOD(compteur, 2) = 0 THEN
            DBMS_OUTPUT.PUT_LINE(compteur || ' est Pair');
        ELSE
            DBMS_OUTPUT.PUT_LINE(compteur || ' est Impair');
        END IF;
    END LOOP;
END;
/
```

Résultat :

```
SQL> SET SERVEROUTPUT ON
SQL> BEGIN
  2  FOR compteur IN 1..10 LOOP
  3  IF MOD(compteur, 2) = 0 THEN
  4  DBMS_OUTPUT.PUT_LINE(compteur || ' est Pair');
  5  ELSE DBMS_OUTPUT.PUT_LINE(compteur || ' est Impair');
  6  END IF;
  7  END LOOP;
  8  END;
  9* /
1 est Impair
2 est Pair
3 est Impair
4 est Pair
5 est Impair
6 est Pair
7 est Impair
8 est Pair
9 est Impair
10 est Pair


Procédure PL/SQL terminée.
```

1.2 Exercice 2 : Utilisation de IF/ELSIF

Écrivez un bloc PL/SQL qui :

- Déclare une variable v_salary initialisée à 3000 ;
- Si salaire < 2000 → affiche "Salaire faible" ;
- Si entre 2000 et 5000 → affiche "Salaire moyen" ;
- Sinon → affiche "Salaire élevé" ;

Code PL/SQL :

```
DECLARE
v_salary NUMBER := 3000;
BEGIN
    IF v_salary < 2000 THEN
        DBMS_OUTPUT.PUT_LINE('Salaire faible');
    ELSIF v_salary BETWEEN 2000 AND 5000 THEN
        DBMS_OUTPUT.PUT_LINE('Salaire moyen');
    ELSE DBMS_OUTPUT.PUT_LINE('Salaire elevé');
    END IF;
END;
/
```

Résultat :

```
SQL> DECLARE
  2  v_salary NUMBER := 3000;
  3  BEGIN
  4  IF v_salary < 2000 THEN
  5  DBMS_OUTPUT.PUT_LINE('Salaire faible');
  6  ELSIF v_salary BETWEEN 2000 AND 5000 THEN
  7  DBMS_OUTPUT.PUT_LINE('Salaire moyen');
  8  ELSE DBMS_OUTPUT.PUT_LINE('Salaire elevé');
  9  END IF;
 10  END;
 11* /
Salaire moyen


Procédure PL/SQL terminée.
```

1.3 Exercice 3 : Utilisation de CASE

Écrivez un bloc PL/SQL qui :

- Déclare une variable v_job_id VARCHAR2(10) := 'IT_PROG' ;
- Utilise une instruction CASE pour afficher :
  - 'Développeur' si job_id = 'IT_PROG' ;
  - 'Manager' si job_id = 'ST_MAN' ;
  - 'Commercial' si job_id = 'SA_REP' ;
  - 'Other' sinon ;

Code PL/SQL :

```
DECLARE
v_job_id VARCHAR2(10) := 'IT_PROG';
BEGIN
    DBMS_OUTPUT.PUT_LINE(
        CASE v_job_id
            WHEN 'IT_PROG' THEN 'DEVELOPER'
            WHEN 'ST_MAN' THEN 'Manager'
            WHEN 'SA_REP' THEN 'Commercial'
            ELSE 'Other'
        END
    );
END;
/
```

Résultat :

```
SQL> DECLARE
  2  v_job_id VARCHAR2(10) := 'IT_PROG';
  3  BEGIN
  4  DBMS_OUTPUT.PUT_LINE(
  5  CASE v_job_id
  6  WHEN 'IT_PROG' THEN 'DEVELOPER'
  7  WHEN 'ST_MAN' THEN 'Manager'
  8  WHEN 'SA_REP' THEN 'Commercial'
  9  ELSE 'Other'
 10  END
 11  );
 12  END;
 13* /
DEVELOPER


Procédure PL/SQL terminée.
```

2 Partie 2 : Travail avec les Curseurs Implicites et Explicites

2.1 Exercice 4 : Curseur Implicite

2.1.1 Affichez le nombre total d'employés travaillant dans le département RH.

Code PL/SQL :

```
DECLARE
v_count NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_count
    FROM employees
    WHERE department_id = 40; -- Département RH
    DBMS_OUTPUT.PUT_LINE('Nombre d''employés RH : ' || v_count);
END;
```

Résultat :

```
SQL> DECLARE
  2  v_count NUMBER;
  3  BEGIN
  4      SELECT COUNT(*) INTO v_count
  5      FROM employees
  6      WHERE department_id = 40; -- Département RH
  7      DBMS_OUTPUT.PUT_LINE('Nombre d''employés RH : ' || v_count);
  8  END;
  9* /
Nombre d'employés RH : 1


Procédure PL/SQL terminée.
```

2.1.2 Quel type de curseur est utilisé automatiquement ici ?

Le type de curseur utilisé automatiquement quand on utilise un *SELECT ... INTO ...* est un curseur implicite.

2.2 Exercice 5 : Curseur Explicite

Écrivez un bloc PL/SQL qui :

— Déclare un curseur pour sélectionner first_name, last_name et salary des employés avec un salaire > 10000 ;

— Récupère chaque enregistrement et affiche le nom de l'employé et son salaire ;

Astuce : Utilisez CURSOR, OPEN, FETCH et CLOSE.

Code PL/SQL :

```
DECLARE
    CURSOR pointeur_employees IS
    SELECT first_name, last_name, salary
    FROM employees
    WHERE salary > 10000;
    v_first_name employees.first_name%TYPE;
    v_last_name employees.last_name%TYPE;
    v_salary employees.salary%TYPE;
BEGIN
    OPEN pointeur_employees;
    LOOP
        FETCH pointeur_employees INTO v_first_name, v_last_name,
           v_salary;
        EXIT WHEN pointeur_employees%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE(v_first_name || ' ' || v_last_name
           || ' - Salaire :  ' || v_salary);
    END LOOP;
    CLOSE pointeur_employees;
END;
/
```

Résultat :

```
SQL> DECLARE
  2  CURSOR pointeur_employees IS
  3  SELECT first_name, last_name, salary
  4  FROM employees
  5  WHERE salary > 10000;
  6  v_first_name employees.first_name%TYPE;
  7  v_last_name employees.last_name%TYPE;
  8  v_salary employees.salary%TYPE;
  9  BEGIN
 10  OPEN pointeur_employees;
 11  LOOP
 12  FETCH pointeur_employees INTO v_first_name, v_last_name, v_salary;
 13  EXIT WHEN pointeur_employees%NOTFOUND;
 14  DBMS_OUTPUT.PUT_LINE(v_first_name || ' ' || v_last_name || ' -
    Salaire :  ' || v_salary);
 15  END LOOP;
 16  CLOSE pointeur_employees;
 17  END;
 18* /
Steven King - Salaire :  24000
Neena Kochhar - Salaire :  17000
Lex De Haan - Salaire :  17000
Nancy Greenberg - Salaire :  12008
Den Raphaely - Salaire :  11000
John Russell - Salaire :  14000
Karen Partners - Salaire :  13500
Alberto Errazuriz - Salaire :  12000
Gerald Cambrault - Salaire :  11000
Eleni Zlotkey - Salaire :  10500
Clara Vishney - Salaire :  10500
Lisa Ozer - Salaire :  11500
Ellen Abel - Salaire :  11000
Michael Hartstein - Salaire :  13000
Shelley Higgins - Salaire :  12008


Procédure PL/SQL terminée.
```

3 Partie 3 : Procédures et Fonctions

3.1 Exercice 6 : Procédures

Créez une procédure show_employee_info qui :

- Prend un paramètre d'entrée p_emp_id (employee_id) ;
- Affiche le nom, le poste et le salaire de l'employé ;
- Si l'employé n'existe pas, affiche "Employé non trouvé" ;
- Testez-la en utilisant :

```
BEGIN
    show_employee_info(101);
END;
```

Code PL/SQL :

```
CREATE OR REPLACE PROCEDURE show_employee_info(p_emp_id IN NUMBER
    )
IS
    v_first_name employees.first_name%TYPE;
    v_last_name employees.last_name%TYPE;
    v_job_id employees.job_id%TYPE;
    v_salary employees.salary%TYPE;

BEGIN

    SELECT first_name, last_name, job_id, salary
    INTO v_first_name, v_last_name, v_job_id, v_salary
    FROM employees
    WHERE employee_id = p_emp_id;

    DBMS_OUTPUT.PUT_LINE('Nom : ' || v_first_name || ' ' ||
       v_last_name);
    DBMS_OUTPUT.PUT_LINE('Poste : ' || v_job_id);
    DBMS_OUTPUT.PUT_LINE('Salaire : ' || v_salary);

    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            DBMS_OUTPUT.PUT_LINE('Employé non trouvé');
END show_employee_info;
/
```

Résultat :

```
SQL> CREATE OR REPLACE PROCEDURE show_employee_info(p_emp_id IN NUMBER)
  2  IS
  3  v_first_name employees.first_name%TYPE;
  4  v_last_name employees.last_name%TYPE;
  5  v_job_id employees.job_id%TYPE;
  6  v_salary employees.salary%TYPE;
  7
  8  BEGIN
  9
 10  SELECT first_name, last_name, job_id, salary
 11  INTO v_first_name, v_last_name, v_job_id, v_salary
 12  FROM employees
 13  WHERE employee_id = p_emp_id;
 14
 15  DBMS_OUTPUT.PUT_LINE('Nom : ' || v_first_name || ' ' || v_last_name
    );
 16  DBMS_OUTPUT.PUT_LINE('Poste : ' || v_job_id);
 17  DBMS_OUTPUT.PUT_LINE('Salaire : ' || v_salary);
 18
 19  EXCEPTION
 20  WHEN NO_DATA_FOUND THEN
 21  DBMS_OUTPUT.PUT_LINE('Employé non trouvé');
 22  END show_employee_info;
 23* /

Elément Procedure SHOW_EMPLOYEE_INFO compilé
```

```
Aucune erreur.
SQL> BEGIN
  2  show_employee_info(101);
  3  END;
  4* /
Nom : Neena Kochhar
Poste : AD_VP
Salaire : 17000


Procédure PL/SQL terminée.

SQL> BEGIN
  2  show_employee_info(9999999999);
  3  END;
  4* /
Employé non trouvé


Procédure PL/SQL terminée.
```

3.2 Exercice 7 : Fonctions

Créez une fonction get_annual_salary qui :

- Prend p_emp_id comme entrée ;
- Retourne $(salary + NVL(commission\_pct, 0) \times salary) \times 12$;
- Testez-la en utilisant :

```sql
SELECT first_name, get_annual_salary(employee_id) AS
   salaire_annuel
FROM employees
WHERE department_id = 90;
```

Code PL/SQL :

```sql
CREATE OR REPLACE FUNCTION get_annual_salary(p_emp_id IN NUMBER)
RETURN NUMBER
IS
    v_salary employees.salary%TYPE;
    v_commission employees.commission_pct%TYPE;
    v_annual_salary NUMBER;

BEGIN

    SELECT salary, commission_pct
    INTO v_salary, v_commission
    FROM employees
    WHERE employee_id = p_emp_id;

    v_annual_salary := (v_salary + NVL(v_commission, 0) *
       v_salary) * 12;

```

```
17      RETURN v_annual_salary;
18
19      EXCEPTION
20          WHEN NO_DATA_FOUND THEN
21              RETURN NULL;
22  END get_annual_salary;
23  /
```

Résultat :

```
SQL> CREATE OR REPLACE FUNCTION get_annual_salary(p_emp_id IN NUMBER)
  2  RETURN NUMBER
  3  IS
  4  v_salary employees.salary%TYPE;
  5  v_commission employees.commission_pct%TYPE;
  6  v_annual_salary NUMBER;
  7
  8  BEGIN
  9
 10  SELECT salary, commission_pct
 11  INTO v_salary, v_commission
 12  FROM employees
 13  WHERE employee_id = p_emp_id;
 14
 15  v_annual_salary := (v_salary + NVL(v_commission, 0) * v_salary) *
    12;
 16
 17  RETURN v_annual_salary;
 18
 19  EXCEPTION
 20  WHEN NO_DATA_FOUND THEN
 21  RETURN NULL;
 22  END get_annual_salary;
 23* /

Elément Function GET_ANNUAL_SALARY compilé

Aucune erreur.
SQL> SELECT first_name, get_annual_salary(employee_id) AS salaire_annuel
  2  FROM employees
  3* WHERE department_id = 90;

FIRST_NAME SALAIRE_ANNUEL
---------- --------------
Steven             288000
Neena              204000
Lex                204000
```

4 Partie 4 : Curseur avec Conditions et Boucles

4.1 Exercice 8 : Curseur + Condition IF

Ecrivez un bloc PL/SQL qui :

- Déclare un curseur pour les employés du département 60 ;
- Pour chaque employé :
  - Si salaire > 10000 → affiche "Salaire élevé" ;
  - Sinon → affiche "Salaire normal" ;

Code PL/SQL :

```
DECLARE
    CURSOR ptr_employees_dep IS
    SELECT first_name, last_name, salary
    FROM employees
    WHERE department_id = 60;
    v_first_name employees.first_name%TYPE;
    v_last_name employees.last_name%TYPE;
    v_salary employees.salary%TYPE;

BEGIN

    OPEN ptr_employees_dep;
    LOOP
    FETCH ptr_employees_dep INTO v_first_name, v_last_name,
       v_salary;
    EXIT WHEN ptr_employees_dep%NOTFOUND;
    IF v_salary > 10000 THEN
        DBMS_OUTPUT.PUT_LINE(v_first_name || ' ' || v_last_name
           || ' - Salaire :  ' || v_salary || ' Salaire élevé');
    ELSE DBMS_OUTPUT.PUT_LINE(v_first_name || ' ' || v_last_name
       || ' - Salaire :  ' || v_salary || ' Salaire normal');
    END IF;
    END LOOP;
    CLOSE ptr_employees_dep;
END;
/
```

Résultat :

```
SQL> DECLARE
  2  CURSOR ptr_employees_dep IS
  3  SELECT first_name, last_name, salary
  4  FROM employees
  5  WHERE department_id = 60;
  6  v_first_name employees.first_name%TYPE;
  7  v_last_name employees.last_name%TYPE;
  8  v_salary employees.salary%TYPE;
  9  BEGIN
 10  OPEN ptr_employees_dep;
 11  LOOP
 12  FETCH ptr_employees_dep INTO v_first_name, v_last_name, v_salary;
```

```
 13  EXIT WHEN ptr_employees_dep%NOTFOUND;
 14  IF v_salary > 10000 THEN
 15  DBMS_OUTPUT.PUT_LINE(v_first_name || ' ' || v_last_name || ' -
    Salaire :  ' || v_salary || ' Salaire élevé');
 16  ELSE DBMS_OUTPUT.PUT_LINE(v_first_name || ' ' || v_last_name || ' -
     Salaire :  ' || v_salary || ' Salaire normal');
 17  END IF;
 18  END LOOP;
 19  CLOSE ptr_employees_dep;
 20  END;
 21* /
Alexander Hunold - Salaire :  9000 Salaire normal
Bruce Ernst - Salaire :  6000 Salaire normal
David Austin - Salaire :  4800 Salaire normal
Valli Pataballa - Salaire :  4800 Salaire normal
Diana Lorentz - Salaire :  4200 Salaire normal


Procédure PL/SQL terminée.
```

5 Partie 5 : Boucle FOR sur Curseur et CASE Ensemble

5.1 Exercice 9 : Utilisation de la Boucle FOR sur Curseur avec CASE

Affichez le nom de chaque employé et un message selon son poste :

- 'SA_REP' → "Représentant Commercial" ;
- 'IT_PROG' → "Programmeur" ;
- 'ST_MAN' → "Responsable de Magasin" ;
- Sinon → "Autre poste" ;

Astuce : Utilisez le format de code suivant :

```
FOR rec IN (SELECT first_name, job_id FROM employees) LOOP
    CASE rec.job_id
        WHEN 'SA_REP' THEN ...
        WHEN 'IT_PROG' THEN ...
        ELSE ...
    END CASE;
END LOOP;
```

Code PL/SQL :

```
BEGIN
    FOR rec IN (SELECT first_name, last_name, job_id FROM
       employees)
    LOOP
    CASE rec.job_id
        WHEN 'SA_REP' THEN
            DBMS_OUTPUT.PUT_LINE(rec.first_name || ' ' || rec.
               last_name || ' : Représentant Commercial');
```

```
7           WHEN 'IT_PROG' THEN
8               DBMS_OUTPUT.PUT_LINE(rec.first_name || ' ' || rec.
                    last_name || ' : Programmeur');
9           WHEN 'ST_MAN' THEN
10              DBMS_OUTPUT.PUT_LINE(rec.first_name || ' ' || rec.
                    last_name || ' : Responsable du Magasin');
11          ELSE DBMS_OUTPUT.PUT_LINE(rec.first_name || ' ' || rec.
                last_name || ' : Autre poste');
12      END CASE;
13      END LOOP;
14  END;
15  /
```

Résultat :

```
SQL> BEGIN
  2  FOR rec IN (SELECT first_name, last_name, job_id FROM employees)
     LOOP
  3  CASE rec.job_id
  4  WHEN 'SA_REP' THEN
  5  DBMS_OUTPUT.PUT_LINE(rec.first_name || ' ' || rec.last_name || ' :
     Représentant Commercial');
  6  WHEN 'IT_PROG' THEN
  7  DBMS_OUTPUT.PUT_LINE(rec.first_name || ' ' || rec.last_name || ' :
     Programmeur');
  8  WHEN 'ST_MAN' THEN
  9  DBMS_OUTPUT.PUT_LINE(rec.first_name || ' ' || rec.last_name || ' :
     Responsable du Magasin');
 10  ELSE DBMS_OUTPUT.PUT_LINE(rec.first_name || ' ' || rec.last_name ||
     ' : Autre poste');
 11  END CASE;
 12  END LOOP;
 13  END;
 14* /
William Gietz : Autre poste
Shelley Higgins : Autre poste
Jennifer Whalen : Autre poste
Steven King : Autre poste
Neena Kochhar : Autre poste
Lex De Haan : Autre poste
Daniel Faviet : Autre poste
John Chen : Autre poste
Ismael Sciarra : Autre poste
Jose Manuel Urman : Autre poste
Luis Popp : Autre poste
Nancy Greenberg : Autre poste
Susan Mavris : Autre poste
Alexander Hunold : Programmeur
Bruce Ernst : Programmeur
David Austin : Programmeur
Valli Pataballa : Programmeur
Diana Lorentz : Programmeur
Michael Hartstein : Autre poste
Pat Fay : Autre poste
Hermann Baer : Autre poste
Alexander Khoo : Autre poste
Shelli Baida : Autre poste
```

```
Sigal Tobias : Autre poste
Guy Himuro : Autre poste
Karen Colmenares : Autre poste
Den Raphaely : Autre poste
John Russell : Autre poste
Karen Partners : Autre poste
Alberto Errazuriz : Autre poste
Gerald Cambrault : Autre poste
Eleni Zlotkey : Autre poste
Peter Tucker : Représentant Commercial
David Bernstein : Représentant Commercial
Peter Hall : Représentant Commercial
Christopher Olsen : Représentant Commercial
Nanette Cambrault : Représentant Commercial
Oliver Tuvault : Représentant Commercial
Janette King : Représentant Commercial
Patrick Sully : Représentant Commercial
Allan McEwen : Représentant Commercial
Lindsey Smith : Représentant Commercial
Louise Doran : Représentant Commercial
Sarath Sewall : Représentant Commercial
Clara Vishney : Représentant Commercial
Danielle Greene : Représentant Commercial
Mattea Marvins : Représentant Commercial
David Lee : Représentant Commercial
Sundar Ande : Représentant Commercial
Amit Banda : Représentant Commercial
Lisa Ozer : Représentant Commercial
Harrison Bloom : Représentant Commercial
Tayler Fox : Représentant Commercial
William Smith : Représentant Commercial
Elizabeth Bates : Représentant Commercial
Sundita Kumar : Représentant Commercial
Ellen Abel : Représentant Commercial
Alyssa Hutton : Représentant Commercial
Jonathon Taylor : Représentant Commercial
Jack Livingston : Représentant Commercial
Kimberely Grant : Représentant Commercial
Charles Johnson : Représentant Commercial
Winston Taylor : Autre poste
Jean Fleaur : Autre poste
Martha Sullivan : Autre poste
Girard Geoni : Autre poste
Nandita Sarchand : Autre poste
Alexis Bull : Autre poste
Julia Dellinger : Autre poste
Anthony Cabrio : Autre poste
Kelly Chung : Autre poste
Jennifer Dilly : Autre poste
Timothy Gates : Autre poste
Randall Perkins : Autre poste
Sarah Bell : Autre poste
Britney Everett : Autre poste
Samuel McCain : Autre poste
Vance Jones : Autre poste
Alana Walsh : Autre poste
Kevin Feeney : Autre poste
Donald OConnell : Autre poste
```

```
Douglas Grant : Autre poste
Julia Nayer : Autre poste
Irene Mikkilineni : Autre poste
James Landry : Autre poste
Steven Markle : Autre poste
Laura Bissot : Autre poste
Mozhe Atkinson : Autre poste
James Marlow : Autre poste
TJ Olson : Autre poste
Jason Mallin : Autre poste
Michael Rogers : Autre poste
Ki Gee : Autre poste
Hazel Philtanker : Autre poste
Renske Ladwig : Autre poste
Stephen Stiles : Autre poste
John Seo : Autre poste
Joshua Patel : Autre poste
Trenna Rajs : Autre poste
Curtis Davies : Autre poste
Randall Matos : Autre poste
Peter Vargas : Autre poste
Matthew Weiss : Responsable du Magasin
Adam Fripp : Responsable du Magasin
Payam Kaufling : Responsable du Magasin
Shanta Vollman : Responsable du Magasin
Kevin Mourgos : Responsable du Magasin


Procédure PL/SQL terminée.
```

6 Partie 6 : Défi - Procédure + Curseur

6.1 Exercice 10 : Procédure avec Curseur

Créez une procédure increase_salary qui :

- Augmente le salaire de tous les employés d'un département donné de 10% ;
- Prend p_dept_id comme paramètre ;
- Utilise un curseur explicite pour mettre à jour les salaires un par un ;
- Affiche le nombre total d'employés mis à jour ;

Testez-la en utilisant :

```
BEGIN
    increase_salary(50);
END;
```

Code PL/SQL :

```
CREATE OR REPLACE PROCEDURE increase_salary(p_dep_id IN NUMBER)
IS
    CURSOR c_employees IS
    SELECT employee_id, first_name, last_name, salary
    FROM employees
    WHERE department_id = p_dep_id
    FOR UPDATE;
    v_cmpt NUMBER := 0;

BEGIN

    FOR rec IN c_employees LOOP
    UPDATE employees
    SET salary = salary * 1.1
    WHERE employee_id = rec.employee_id;
    v_cmpt := v_cmpt + 1;
    END LOOP;
    COMMIT;
    DBMS_OUTPUT.PUT_LINE('Nombre total d''employés mis    jour :
       ' || v_cmpt);
END increase_salary;
/
```

Résultat :

```
SQL> CREATE OR REPLACE PROCEDURE increase_salary(p_dep_id IN NUMBER)
  2  IS
  3  CURSOR c_employees IS
  4  SELECT employee_id, first_name, last_name, salary
  5  FROM employees
  6  WHERE department_id = p_dep_id
  7  FOR UPDATE;
  8  v_cmpt NUMBER := 0;
  9
 10  BEGIN
 11
 12  FOR rec IN c_employees LOOP
 13
 14  UPDATE employees
 15  SET salary = salary * 1.1
 16  WHERE employee_id = rec.employee_id;
 17
 18  v_cmpt := v_cmpt + 1;
 19
 20  END LOOP;
 21  COMMIT;
 22  DBMS_OUTPUT.PUT_LINE('Nombre total d''employés mis    jour : ' ||
    v_cmpt);
 23
 24  END increase_salary;
 25* /

Elément Procedure INCREASE_SALARY compilé
```

```
Aucune erreur.
SQL> SELECT employee_id, first_name, last_name, salary
  2  FROM employees
  3  WHERE department_id = 50
  4* ORDER BY salary DESC;

EMPLOYEE_ID FIRST_NAME LAST_NAME   SALARY
----------- ---------- ----------- ------
        121 Adam       Fripp         8200
        120 Matthew    Weiss         8000
        122 Payam      Kaufling      7900
        123 Shanta     Vollman       6500
        124 Kevin      Mourgos       5800
        184 Nandita    Sarchand      4200
        185 Alexis     Bull          4100
        192 Sarah      Bell          4000
        193 Britney    Everett       3900
        188 Kelly      Chung         3800
        189 Jennifer   Dilly         3600
        137 Renske     Ladwig        3600
        141 Trenna     Rajs          3500
        186 Julia      Dellinger     3400
        133 Jason      Mallin        3300
        129 Laura      Bissot        3300
        180 Winston    Taylor        3200
        125 Julia      Nayer         3200
        138 Stephen    Stiles        3200
        194 Samuel     McCain        3200
        142 Curtis     Davies        3100
        196 Alana      Walsh         3100
        181 Jean       Fleaur        3100
        187 Anthony    Cabrio        3000
        197 Kevin      Feeney        3000
        134 Michael    Rogers        2900
        190 Timothy    Gates         2900
        130 Mozhe      Atkinson      2800
        195 Vance      Jones         2800
        183 Girard     Geoni         2800
        126 Irene      Mikkilineni   2700
        139 John       Seo           2700
        143 Randall    Matos         2600
        198 Donald     OConnell      2600
        199 Douglas    Grant         2600
        191 Randall    Perkins       2500
        131 James      Marlow        2500
        182 Martha     Sullivan      2500

EMPLOYEE_ID FIRST_NAME LAST_NAME  SALARY
----------- ---------- ---------- ------
        140 Joshua     Patel        2500
        144 Peter      Vargas       2500
        135 Ki         Gee          2400
        127 James      Landry       2400
        128 Steven     Markle       2200
        136 Hazel      Philtanker   2200
        132 TJ         Olson        2100

45 lignes sélectionnées.
```

```
SQL> BEGIN
  2    increase_salary(50);
  3  END;
  4* /
Nombre total d'employés mis    jour : 45


Procédure PL/SQL terminée.

SQL> SELECT employee_id, first_name, last_name, salary
  2  FROM employees
  3  WHERE department_id = 50
  4* ORDER BY salary DESC;

EMPLOYEE_ID FIRST_NAME LAST_NAME   SALARY
----------- ---------- ----------- ------
        121 Adam       Fripp         9020
        120 Matthew    Weiss         8800
        122 Payam      Kaufling      8690
        123 Shanta     Vollman       7150
        124 Kevin      Mourgos       6380
        184 Nandita    Sarchand      4620
        185 Alexis     Bull          4510
        192 Sarah      Bell          4400
        193 Britney    Everett       4290
        188 Kelly      Chung         4180
        189 Jennifer   Dilly         3960
        137 Renske     Ladwig        3960
        141 Trenna     Rajs          3850
        186 Julia      Dellinger     3740
        133 Jason      Mallin        3630
        129 Laura      Bissot        3630
        180 Winston    Taylor        3520
        125 Julia      Nayer         3520
        138 Stephen    Stiles        3520
        194 Samuel     McCain        3520
        142 Curtis     Davies        3410
        196 Alana      Walsh         3410
        181 Jean       Fleaur        3410
        187 Anthony    Cabrio        3300
        197 Kevin      Feeney        3300
        134 Michael    Rogers        3190
        190 Timothy    Gates         3190
        130 Mozhe      Atkinson      3080
        195 Vance      Jones         3080
        183 Girard     Geoni         3080
        126 Irene      Mikkilineni   2970
        139 John       Seo           2970
        143 Randall    Matos         2860
        198 Donald     OConnell      2860
        199 Douglas    Grant         2860
        191 Randall    Perkins       2750
        131 James      Marlow        2750
        182 Martha     Sullivan      2750

EMPLOYEE_ID FIRST_NAME LAST_NAME  SALARY
----------- ---------- ---------- ------
```

```
       140 Joshua     Patel         2750
       144 Peter      Vargas        2750
       135 Ki         Gee           2640
       127 James      Landry        2640
       128 Steven     Markle        2420
       136 Hazel      Philtanker    2420
       132 TJ         Olson         2310

45 lignes sélectionnées.
```