**SGBD TEMA 5**

**OPRAN ANDREI 242**

**Exercitiul 2 pagina 2**

-- Rezolvați exercițiul 1 menținând informațiile din cursor în colecții. Comentați.

-- Procesați toate liniile din cursor, încărcând la fiecare pas câte 5 linii.

DECLARE

    TYPE TIP\_TABEL\_DEPARTMENT\_NAME IS TABLE OF DEPARTMENTS.DEPARTMENT\_NAME%TYPE INDEX BY PLS\_INTEGER;

    TYPE TIP\_TABEL\_NR IS TABLE OF NUMBER(4) INDEX BY PLS\_INTEGER;

    TABEL\_DEPARTMENT\_NAME TIP\_TABEL\_DEPARTMENT\_NAME;

    TABEL\_NR TIP\_TABEL\_NR;

    CURSOR C IS

    SELECT

        DEPARTMENT\_NAME AS NUME,

        COUNT(EMPLOYEE\_ID) AS NR

    FROM DEPARTMENTS D, EMPLOYEES   E

    WHERE D.DEPARTMENT\_ID=E.DEPARTMENT\_ID(+)

    GROUP BY DEPARTMENT\_NAME;

    V\_CONT NUMBER(6) := 0;

BEGIN

    OPEN C;

    LOOP

        FETCH C BULK COLLECT INTO TABEL\_DEPARTMENT\_NAME, TABEL\_NR LIMIT 5;

        EXIT WHEN TABEL\_DEPARTMENT\_NAME.COUNT=0; -- CA SA AFISEZE SI 1-4 LINII CARE SUNT DUPA UN MULTIPLU DE 5

        FOR I IN 1..TABEL\_DEPARTMENT\_NAME.COUNT LOOP

            V\_CONT := V\_CONT + 1;

            IF TABEL\_NR(I) = 0 THEN

                DBMS\_OUTPUT.PUT\_LINE(V\_CONT || ': ' || 'In departamentul '|| TABEL\_DEPARTMENT\_NAME(I)|| ' nu lucreaza angajati');

            ELSIF TABEL\_NR(I) = 1 THEN

                DBMS\_OUTPUT.PUT\_LINE(V\_CONT || ': ' || 'In departamentul '|| TABEL\_DEPARTMENT\_NAME(I)|| ' lucreaza un angajat');

            ELSE

                DBMS\_OUTPUT.PUT\_LINE(V\_CONT || ': ' || 'In departamentul '|| TABEL\_DEPARTMENT\_NAME(I) || ' lucreaza ' || TABEL\_NR(I) || ' angajati');

            END IF;

        END LOOP;

    END LOOP;

    CLOSE C;

END;

/

****

--  Rezolvați problema folosind cursorul și o singură colecție.

DECLARE

    TYPE PAIR IS RECORD(

        V\_NUME DEPARTMENTS.DEPARTMENT\_NAME%TYPE,

        V\_NR NUMBER(4)

    );

    TYPE TIP\_TABEL\_PAIR IS TABLE OF PAIR INDEX BY PLS\_INTEGER;

    TABEL\_PAIR TIP\_TABEL\_PAIR;

    V\_NR NUMBER(4);

    CURSOR C IS

    SELECT

        DEPARTMENT\_NAME AS NUME,

        COUNT(EMPLOYEE\_ID) AS NR

    FROM DEPARTMENTS D, EMPLOYEES   E

    WHERE D.DEPARTMENT\_ID=E.DEPARTMENT\_ID(+)

    GROUP BY DEPARTMENT\_NAME;

BEGIN

    OPEN C;

    LOOP

        FETCH C BULK COLLECT INTO TABEL\_PAIR;

        EXIT WHEN TABEL\_PAIR.COUNT = 0;

        FOR I IN 1..TABEL\_PAIR.COUNT LOOP

            IF TABEL\_PAIR(I).V\_NR = 0 THEN

                DBMS\_OUTPUT.PUT\_LINE(I || ': ' || 'In departamentul '|| TABEL\_PAIR(I).V\_NUME  || ' nu lucreaza angajati');

            ELSIF TABEL\_PAIR(I).V\_NR = 1 THEN

                DBMS\_OUTPUT.PUT\_LINE(I || ': ' || 'In departamentul '|| TABEL\_PAIR(I).V\_NUME || ' lucreaza un angajat');

            ELSE

                DBMS\_OUTPUT.PUT\_LINE(I || ': ' || 'In departamentul '|| TABEL\_PAIR(I).V\_NUME || ' lucreaza ' || TABEL\_PAIR(I).V\_NR || ' angajati');

            END IF;

        END LOOP;

    END LOOP;

    CLOSE C;

END;

/

****

--  Rezolvați problema folosind doar colecții.

DECLARE

    TYPE TIP\_TABEL\_DEPARTMENT\_NAME IS TABLE OF DEPARTMENTS.DEPARTMENT\_NAME%TYPE INDEX BY PLS\_INTEGER;

    TYPE TIP\_TABEL\_NR IS TABLE OF NUMBER(4) INDEX BY PLS\_INTEGER;

    TABEL\_DEPARTMENT\_NAME TIP\_TABEL\_DEPARTMENT\_NAME;

    TABEL\_NR TIP\_TABEL\_NR;

BEGIN

    SELECT

        DEPARTMENT\_NAME AS NUME,

        COUNT(EMPLOYEE\_ID) AS NR

    BULK COLLECT INTO TABEL\_DEPARTMENT\_NAME, TABEL\_NR

    FROM DEPARTMENTS D, EMPLOYEES   E

    WHERE D.DEPARTMENT\_ID=E.DEPARTMENT\_ID(+)

    GROUP BY DEPARTMENT\_NAME;

    FOR I IN 1..TABEL\_DEPARTMENT\_NAME.COUNT LOOP

        IF TABEL\_NR(I) = 0 THEN

            DBMS\_OUTPUT.PUT\_LINE(I || ': ' || 'In departamentul '|| TABEL\_DEPARTMENT\_NAME(I)|| ' nu lucreaza angajati');

        ELSIF TABEL\_NR(I) = 1 THEN

            DBMS\_OUTPUT.PUT\_LINE(I || ': ' || 'In departamentul '|| TABEL\_DEPARTMENT\_NAME(I)|| ' lucreaza un angajat');

        ELSE

            DBMS\_OUTPUT.PUT\_LINE(I || ': ' || 'In departamentul '|| TABEL\_DEPARTMENT\_NAME(I) || ' lucreaza ' || TABEL\_NR(I) || ' angajati');

        END IF;

    END LOOP;

END;

/

****

**E1 pagina 10**

-- a. cursoare clasice

DECLARE

    titluJob JOBS.JOB\_TITLE%TYPE;

    auxNume EMPLOYEES.LAST\_NAME%TYPE;

    auxSalariu EMPLOYEES.SALARY%TYPE;

    v\_cont NUMBER(6) := 0;

    CURSOR C1 IS

    SELECT

        DISTINCT JOB\_TITLE

    FROM JOBS;

    CURSOR C2(titluJobParametru JOBS.JOB\_TITLE%TYPE) IS

    SELECT

        E.LAST\_NAME AS NUME,

        E.SALARY AS SALARIU

    FROM EMPLOYEES E, JOBS J

    WHERE E.JOB\_ID = J.JOB\_ID AND J.JOB\_TITLE = titluJobParametru;

BEGIN

    OPEN C1;

    LOOP

        FETCH C1 INTO titluJob;

        EXIT WHEN C1%NOTFOUND;

        DBMS\_OUTPUT.PUT\_LINE('---------------------------------------------------------');

        DBMS\_OUTPUT.PUT\_LINE('TITLU JOB: ' || titluJob);

        DBMS\_OUTPUT.PUT\_LINE(' ');

        OPEN C2(titluJob);

        v\_cont := 0;

        LOOP

            FETCH C2 INTO auxNume, auxSalariu;

            EXIT WHEN C2%NOTFOUND;

            DBMS\_OUTPUT.PUT('Nume: ' || auxNume);

            IF(auxSalariu IS NOT NULL) THEN

                DBMS\_OUTPUT.PUT\_LINE(' - salariu: ' || auxSalariu);

            ELSE

                DBMS\_OUTPUT.PUT\_LINE(' - salariu: N/A');

            END IF;

            v\_cont := v\_cont + 1;

        END LOOP;

        IF v\_cont = 0 THEN

            DBMS\_OUTPUT.PUT\_LINE('Nu exista angajati cu jobul ' || titluJob);

        END IF;

        CLOSE C2;

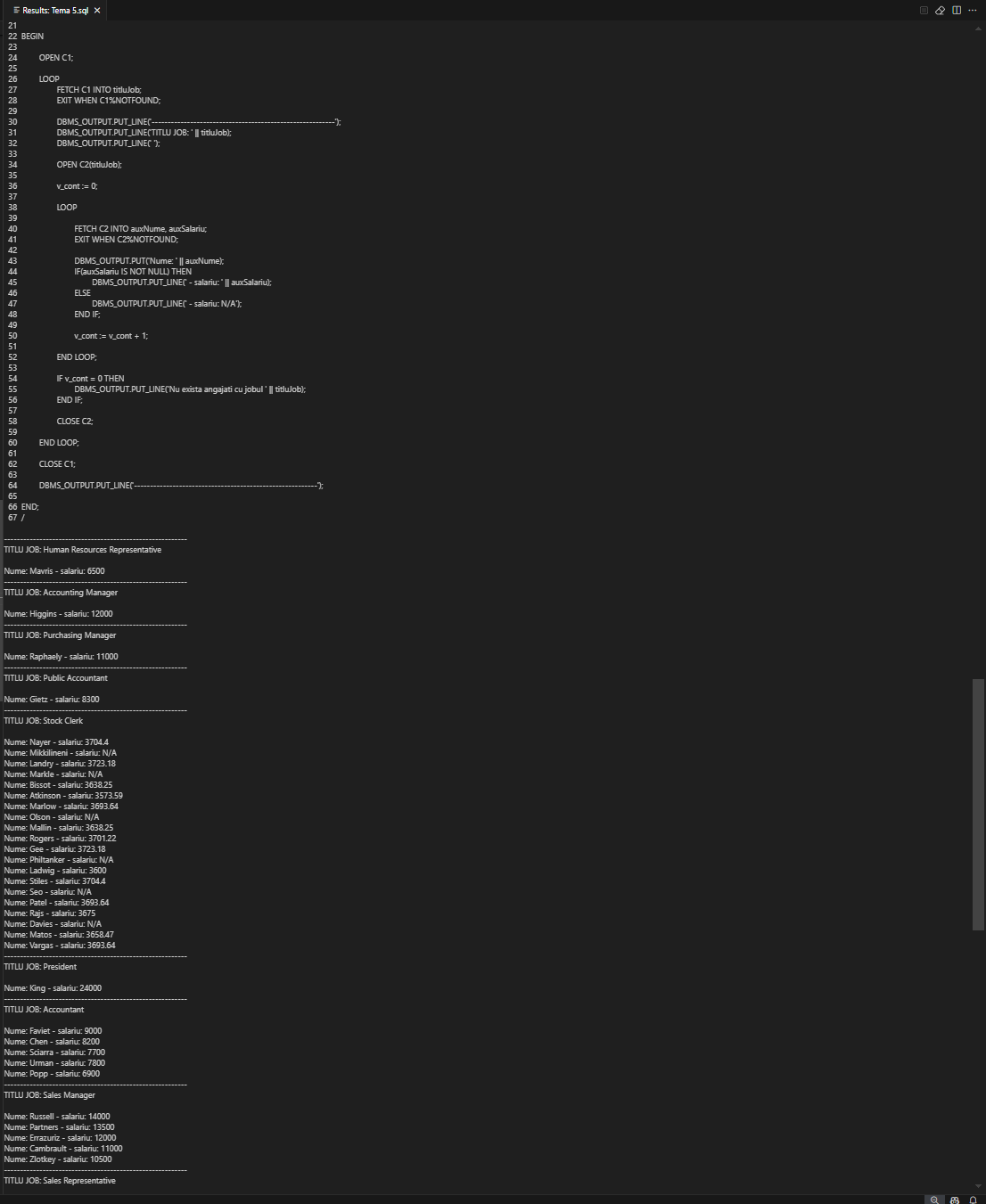
    END LOOP;

    CLOSE C1;

    DBMS\_OUTPUT.PUT\_LINE('---------------------------------------------------------');

END;

/

****

****

-- b. ciclu cursoare

DECLARE

    titluJob JOBS.JOB\_TITLE%TYPE;

    auxNume EMPLOYEES.LAST\_NAME%TYPE;

    auxSalariu EMPLOYEES.SALARY%TYPE;

    v\_cont NUMBER(6) := 0;

    CURSOR C1 IS

    SELECT

        DISTINCT JOB\_TITLE

    FROM JOBS;

    CURSOR C2(titluJobParametru JOBS.JOB\_TITLE%TYPE) IS

    SELECT

        E.LAST\_NAME AS NUME,

        E.SALARY AS SALARIU

    FROM EMPLOYEES E, JOBS J

    WHERE E.JOB\_ID = J.JOB\_ID AND J.JOB\_TITLE = titluJobParametru;

BEGIN

    FOR inregistrareOuter in C1 LOOP

        titluJob := inregistrareOuter.JOB\_TITLE;

        DBMS\_OUTPUT.PUT\_LINE('---------------------------------------------------------');

        DBMS\_OUTPUT.PUT\_LINE('TITLU JOB: ' || titluJob);

        DBMS\_OUTPUT.PUT\_LINE(' ');

        v\_cont := 0;

        FOR inregistrare IN C2(titluJob) LOOP

            auxNume := inregistrare.NUME;

            auxSalariu := inregistrare.SALARIU;

            DBMS\_OUTPUT.PUT('Nume: ' || auxNume);

            IF(auxSalariu IS NOT NULL) THEN

                DBMS\_OUTPUT.PUT\_LINE(' - salariu: ' || auxSalariu);

            ELSE

                DBMS\_OUTPUT.PUT\_LINE(' - salariu: N/A');

            END IF;

            v\_cont := v\_cont + 1;

        END LOOP;

        IF v\_cont = 0 THEN

            DBMS\_OUTPUT.PUT\_LINE('Nu exista angajati cu jobul ' || titluJob);

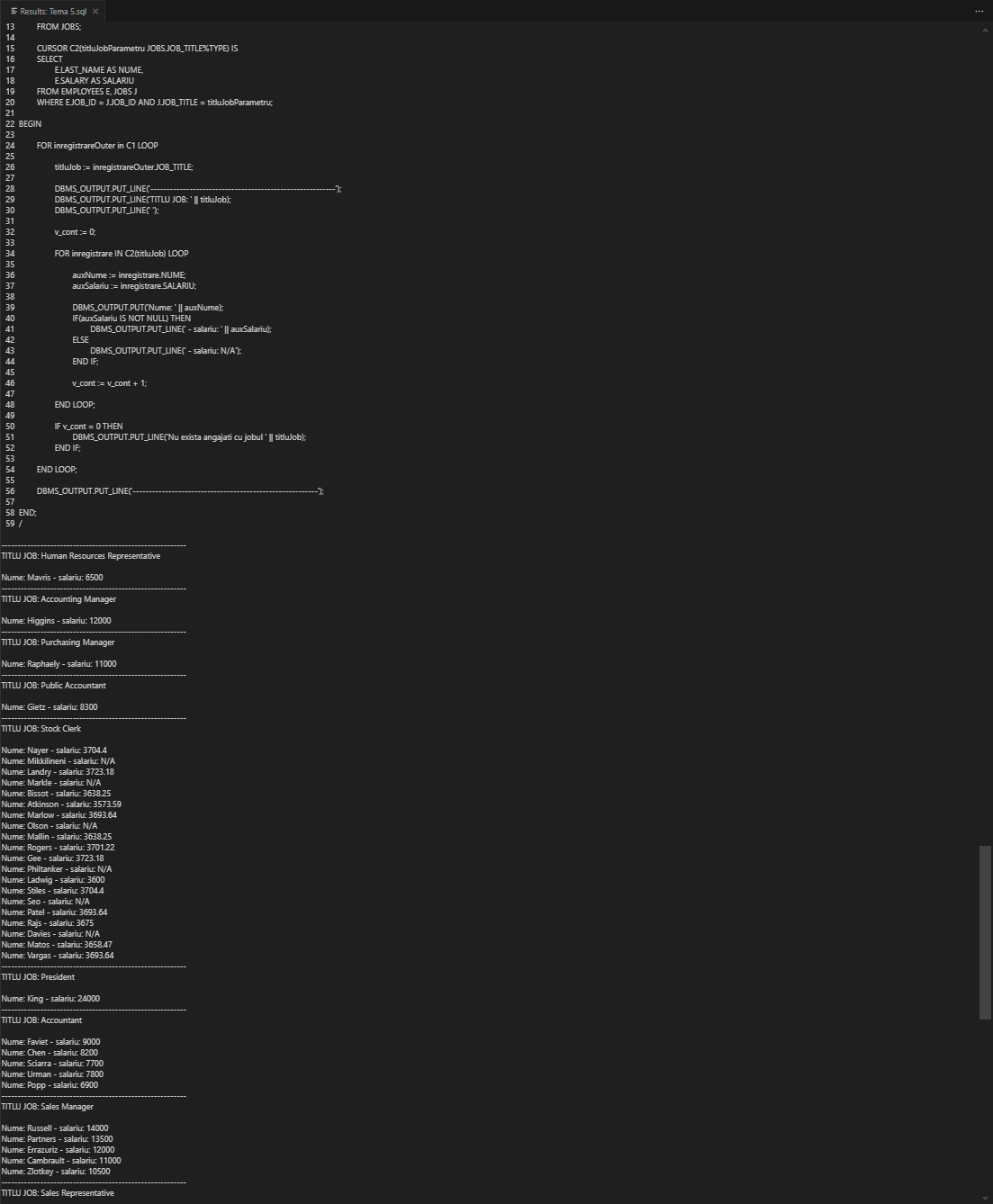
        END IF;

    END LOOP;

    DBMS\_OUTPUT.PUT\_LINE('---------------------------------------------------------');

END;

/

****

****

-- c. ciclu cursoare cu subcereri

DECLARE

    titluJob JOBS.JOB\_TITLE%TYPE;

    auxNume EMPLOYEES.LAST\_NAME%TYPE;

    auxSalariu EMPLOYEES.SALARY%TYPE;

    v\_cont NUMBER(6) := 0;

BEGIN

    FOR titluFor in (

        SELECT

            DISTINCT JOB\_TITLE

        FROM JOBS

    ) LOOP

        titluJob := titluFor.JOB\_TITLE;

        DBMS\_OUTPUT.PUT\_LINE('---------------------------------------------------------');

        DBMS\_OUTPUT.PUT\_LINE('TITLU JOB: ' || titluJob);

        DBMS\_OUTPUT.PUT\_LINE(' ');

        v\_cont := 0;

        FOR inregistrare IN (

            SELECT

                E.LAST\_NAME AS NUME,

                E.SALARY AS SALARIU

            FROM EMPLOYEES E, JOBS J

            WHERE E.JOB\_ID = J.JOB\_ID AND J.JOB\_TITLE = titluJob

        ) LOOP

            auxNume := inregistrare.NUME;

            auxSalariu := inregistrare.SALARIU;

            DBMS\_OUTPUT.PUT('Nume: ' || auxNume);

            IF(auxSalariu IS NOT NULL) THEN

                DBMS\_OUTPUT.PUT\_LINE(' - salariu: ' || auxSalariu);

            ELSE

                DBMS\_OUTPUT.PUT\_LINE(' - salariu: N/A');

            END IF;

            v\_cont := v\_cont + 1;

        END LOOP;

        IF v\_cont = 0 THEN

            DBMS\_OUTPUT.PUT\_LINE('Nu exista angajati cu jobul ' || titluJob);

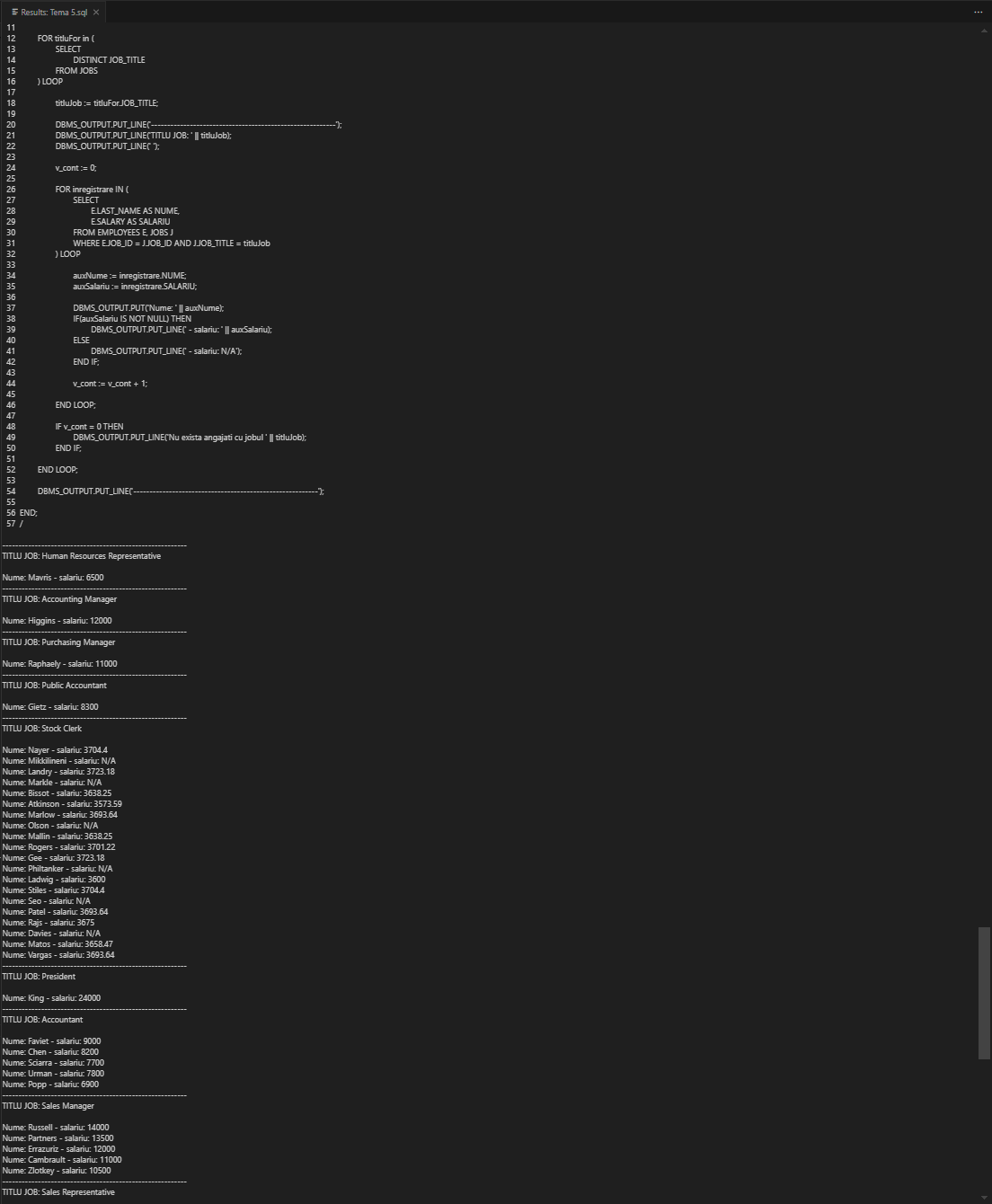
        END IF;

    END LOOP;

    DBMS\_OUTPUT.PUT\_LINE('---------------------------------------------------------');

END;

/

****

****

-- d. expresii cursor

DECLARE

    titluJob JOBS.JOB\_TITLE%TYPE;

    auxNume EMPLOYEES.LAST\_NAME%TYPE;

    auxSalariu EMPLOYEES.SALARY%TYPE;

    v\_cont NUMBER(6) := 0;

    CURSOR C is

    SELECT

        DISTINCT JOB\_TITLE,

        CURSOR(

            SELECT

                E.LAST\_NAME AS NUME,

                E.SALARY AS SALARIU

            FROM EMPLOYEES E, JOBS J1

            WHERE E.JOB\_ID = J1.JOB\_ID AND J1.JOB\_TITLE = J2.JOB\_TITLE

        ) AS CURSOR\_IMBRICAT

    FROM JOBS J2;

    v\_cursor SYS\_REFCURSOR;

BEGIN

    OPEN C;

    LOOP

        FETCH C INTO titluJob, v\_cursor;

        EXIT WHEN C%NOTFOUND;

        DBMS\_OUTPUT.PUT\_LINE('---------------------------------------------------------');

        DBMS\_OUTPUT.PUT\_LINE('TITLU JOB: ' || titluJob);

        DBMS\_OUTPUT.PUT\_LINE(' ');

        v\_cont := 0;

        LOOP

            FETCH v\_cursor INTO auxNume, auxSalariu;

            EXIT WHEN v\_cursor%NOTFOUND;

            DBMS\_OUTPUT.PUT('Nume: ' || auxNume);

            IF(auxSalariu IS NOT NULL) THEN

                DBMS\_OUTPUT.PUT\_LINE(' - salariu: ' || auxSalariu);

            ELSE

                DBMS\_OUTPUT.PUT\_LINE(' - salariu: N/A');

            END IF;

            v\_cont := v\_cont + 1;

        END LOOP;

        IF v\_cont = 0 THEN

            DBMS\_OUTPUT.PUT\_LINE('Nu exista angajati cu jobul ' || titluJob);

        END IF;

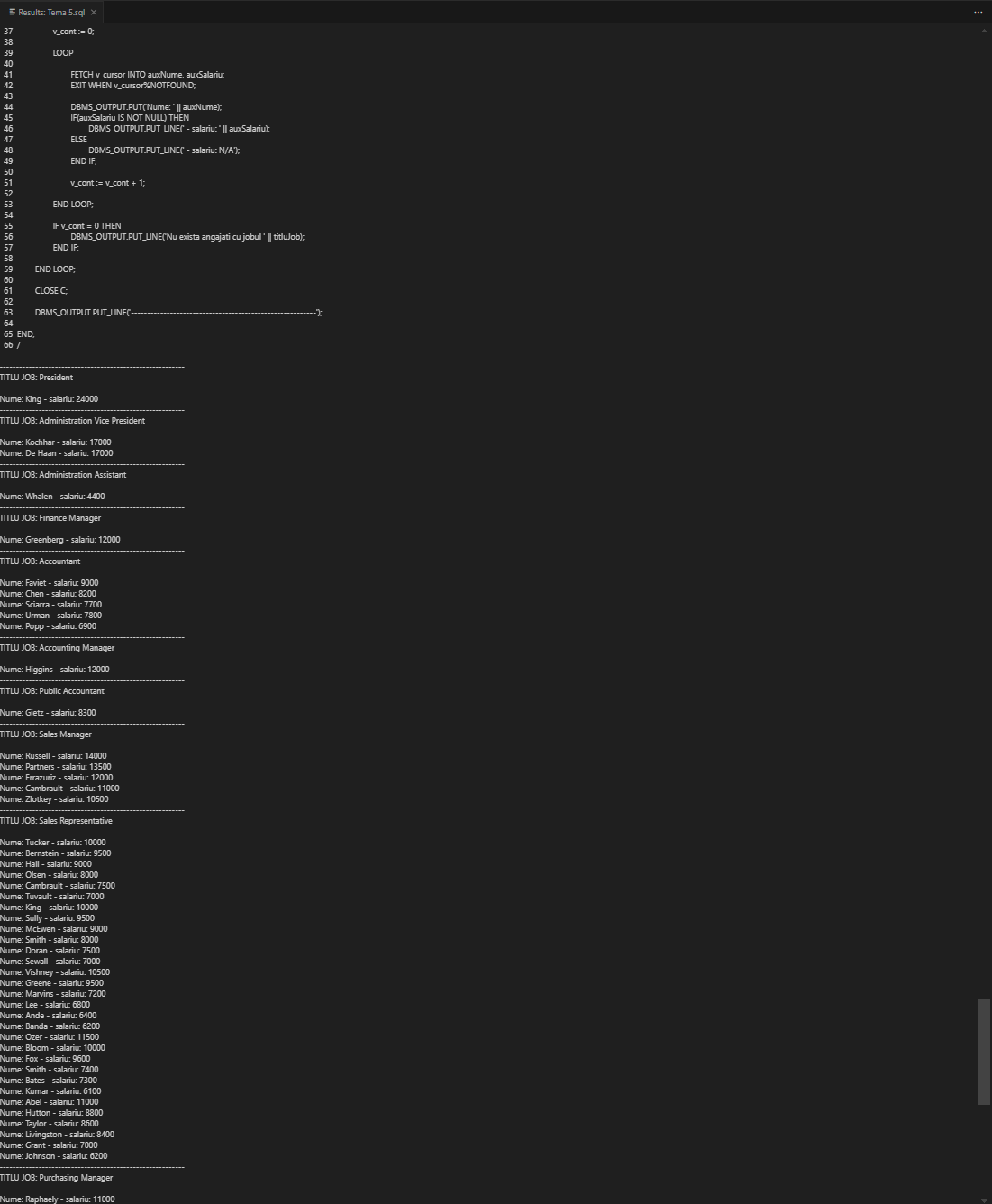
    END LOOP;

    CLOSE C;

    DBMS\_OUTPUT.PUT\_LINE('---------------------------------------------------------');

END;

/

****

****

**OBS: ordinea la d. este diferita fata de a, b, si c.**