E1.

a) 2

b) ‘text 2’

c) ‘text 3 adaugat in subbloc’

d) 101

e) ‘text 1 adaugat un blocul principal’

f) 'text 3 adaugat in sub-bloc adaugat in blocul principal'

E2.

SELECT

o.zi,

COUNT(r.book\_date) AS nr\_imprumuturi

FROM

(

SELECT TO\_DATE('01-NOV-24', 'DD-MON-YY') + LEVEL - 1 AS zi

FROM dual

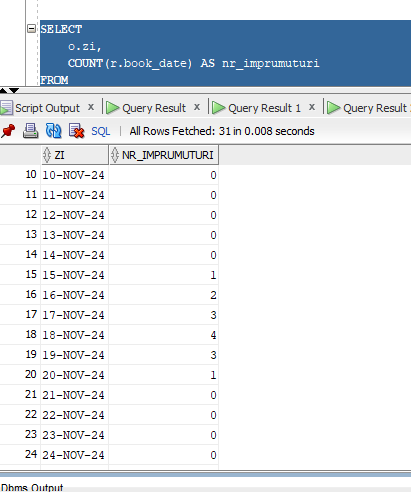
CONNECT BY LEVEL <= 31

) o

LEFT JOIN rental r ON TRUNC(r.book\_date) = o.zi

GROUP BY o.zi

ORDER BY o.zi;



CREATE TABLE noiembrie\_dpe (

id NUMBER PRIMARY KEY,

data DATE

);

DECLARE

v\_data DATE := TO\_DATE('01-NOV-24', 'DD-MON-YY');

BEGIN

FOR i IN 1..30 LOOP

INSERT INTO noiembrie\_dpe (id, data)

VALUES (i, v\_data);

v\_data := v\_data + 1;

END LOOP;

COMMIT;

END;

/

SELECT

o.data AS zi,

COUNT(r.book\_date) AS nr\_imprumuturi

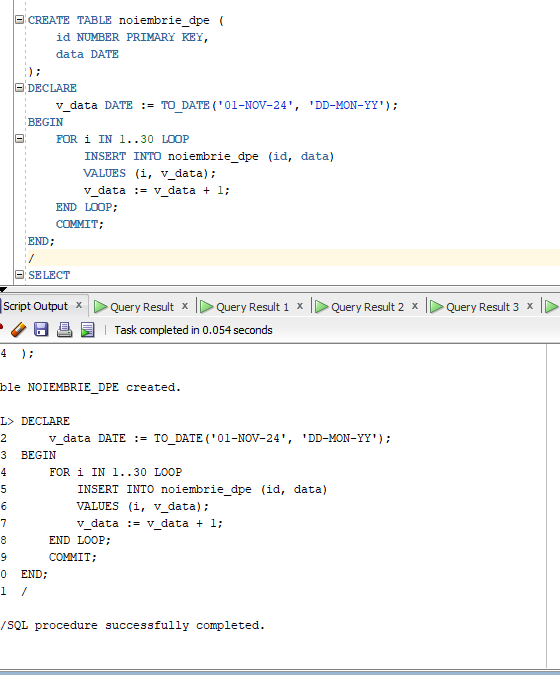
FROM

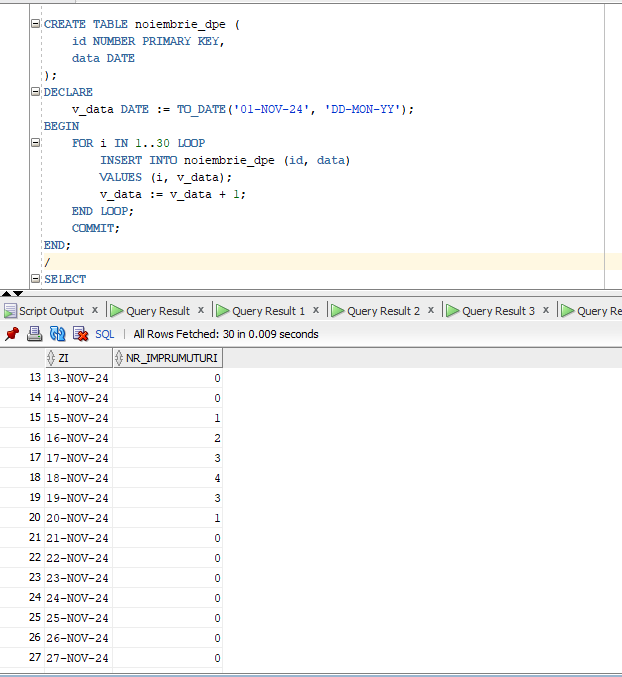
noiembrie\_dpe o

LEFT JOIN rental r ON TRUNC(r.book\_date) = o.data

GROUP BY o.data

ORDER BY o.data;





E3.

DECLARE

v\_nume VARCHAR2(50);

v\_membru\_count INTEGER;

v\_filme\_imprumutate INTEGER;

BEGIN

v\_nume := '&nume\_membru';

SELECT COUNT(\*)

INTO v\_membru\_count

FROM member

WHERE first\_name = v\_nume OR last\_name = v\_nume;

IF v\_membru\_count = 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Nu exista nici un membru cu numele ' || v\_nume);

ELSIF v\_membru\_count > 1 THEN

DBMS\_OUTPUT.PUT\_LINE('Exista mai multi membri cu numele ' || v\_nume);

ELSE

SELECT COUNT(DISTINCT r.title\_id)

INTO v\_filme\_imprumutate

FROM rental r

JOIN member m ON r.member\_id = m.member\_id

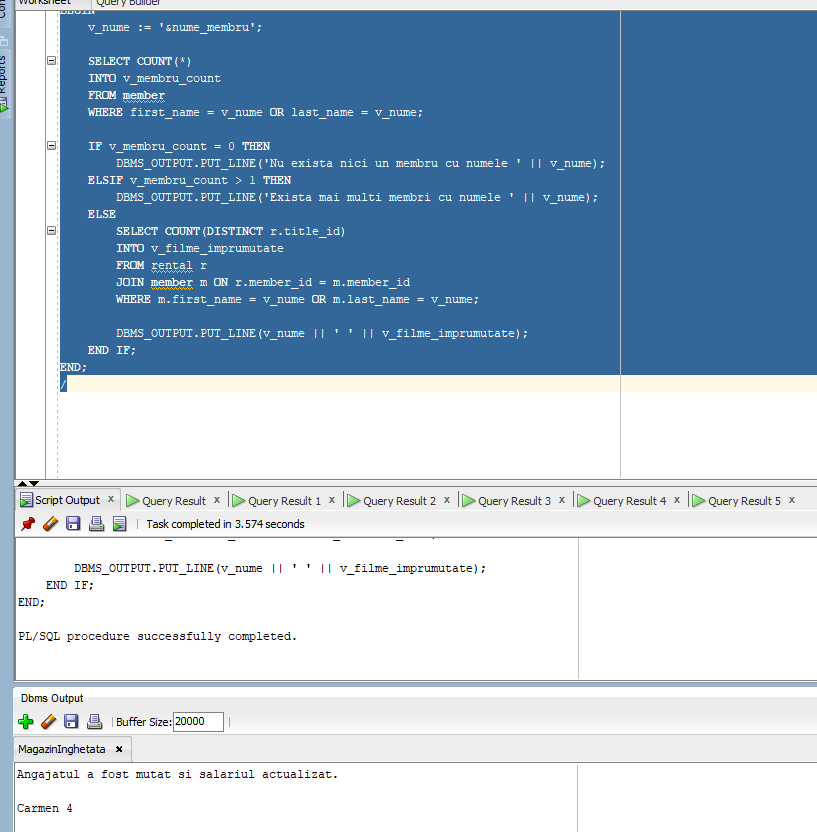
WHERE m.first\_name = v\_nume OR m.last\_name = v\_nume;

DBMS\_OUTPUT.PUT\_LINE(v\_nume || ' ' || v\_filme\_imprumutate);

END IF;

END;

/



E4.

DECLARE

v\_nume VARCHAR2(50);

v\_membru\_count INTEGER;

v\_filme\_imprumutate INTEGER;

v\_total\_filme INTEGER;

v\_procent\_imprumut INTEGER;

BEGIN

v\_nume := '&nume\_membru';

SELECT COUNT(\*)

INTO v\_membru\_count

FROM member

WHERE first\_name = v\_nume OR last\_name = v\_nume;

IF v\_membru\_count = 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Nu exista nici un membru cu numele ' || v\_nume);

ELSIF v\_membru\_count > 1 THEN

DBMS\_OUTPUT.PUT\_LINE('Exista mai multi membri cu numele ' || v\_nume);

ELSE

SELECT COUNT(DISTINCT r.title\_id)

INTO v\_filme\_imprumutate

FROM rental r

JOIN member m ON r.member\_id = m.member\_id

WHERE m.first\_name = v\_nume OR m.last\_name = v\_nume;

SELECT COUNT(DISTINCT title\_id)

INTO v\_total\_filme

FROM title;

v\_procent\_imprumut := (v\_filme\_imprumutate / v\_total\_filme) \* 100;

DBMS\_OUTPUT.PUT\_LINE(v\_nume || ' a imprumutat urmatorul numar de filme: ' || v\_filme\_imprumutate);

IF v\_procent\_imprumut > 75 THEN

DBMS\_OUTPUT.PUT\_LINE('Categoria 1: A imprumutat mai mult de 75% din titlurile existente');

ELSIF v\_procent\_imprumut > 50 THEN

DBMS\_OUTPUT.PUT\_LINE('Categoria 2: A imprumutat mai mult de 50% din titlurile existente');

ELSIF v\_procent\_imprumut > 25 THEN

DBMS\_OUTPUT.PUT\_LINE('Categoria 3: A imprumutat mai mult de 25% din titlurile existente');

ELSE

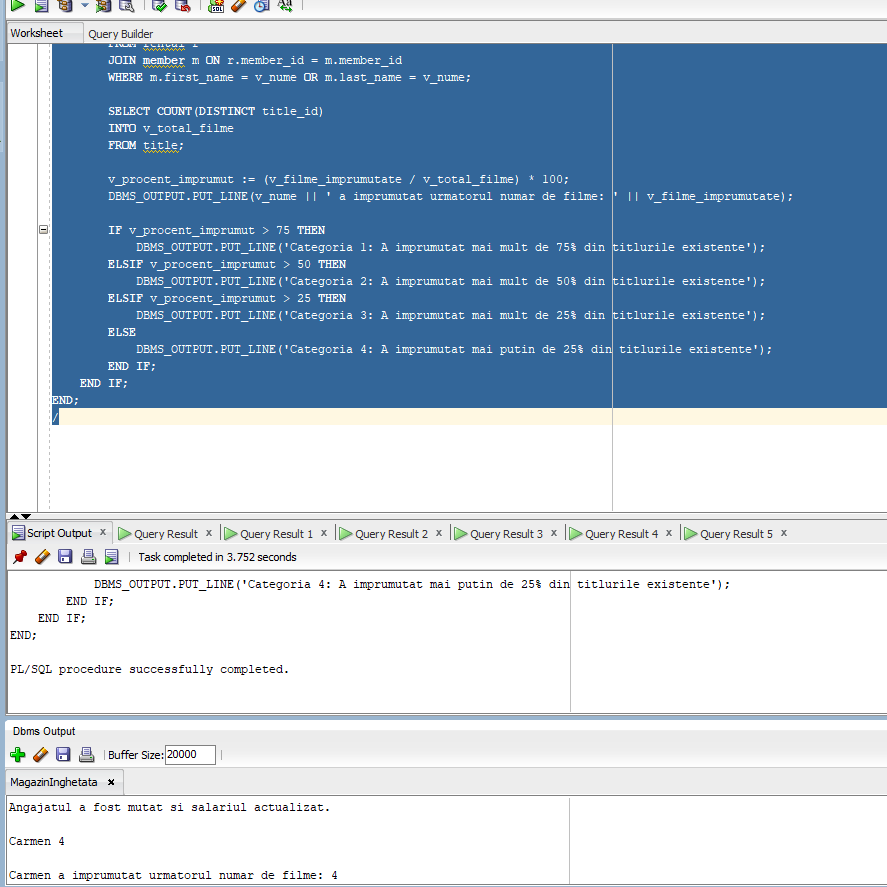
DBMS\_OUTPUT.PUT\_LINE('Categoria 4: A imprumutat mai putin de 25% din titlurile existente');

END IF;

END IF;

END;

/



E5.

CREATE TABLE member\_dpe AS

SELECT member\_id,

first\_name,

last\_name,

address,

city,

phone,

join\_date

FROM member;

-- Adaug coloana discount

ALTER TABLE member\_dpe ADD discount NUMBER(5,2);

DECLARE

v\_membru\_id NUMBER;

v\_filme\_imprumutate INTEGER;

v\_total\_filme INTEGER;

v\_procent\_imprumut INTEGER;

v\_discount NUMBER(5,2);

BEGIN

v\_membru\_id := &membru\_id;

SELECT COUNT(DISTINCT r.title\_id)

INTO v\_filme\_imprumutate

FROM rental r

JOIN member m ON r.member\_id = m.member\_id

WHERE m.member\_id = v\_membru\_id;

SELECT COUNT(DISTINCT title\_id)

INTO v\_total\_filme

FROM title;

v\_procent\_imprumut := (v\_filme\_imprumutate / v\_total\_filme) \* 100;

IF v\_procent\_imprumut > 75 THEN

v\_discount := 10;

ELSIF v\_procent\_imprumut > 50 THEN

v\_discount := 5;

ELSIF v\_procent\_imprumut > 25 THEN

v\_discount := 3;

ELSE

v\_discount := 0;

END IF;

UPDATE member\_dpe

SET discount = v\_discount

WHERE member\_id = v\_membru\_id;

IF SQL%ROWCOUNT > 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Actualizarea discount-ului s-a realizat cu succes pentru membrul cu id-ul ' || v\_membru\_id);

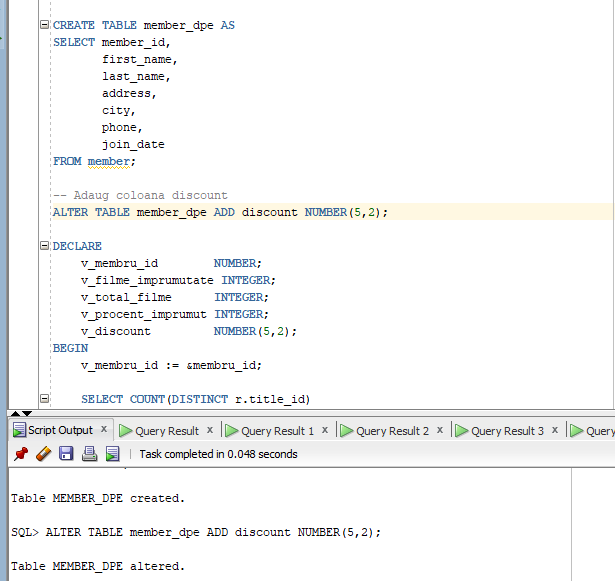
ELSE

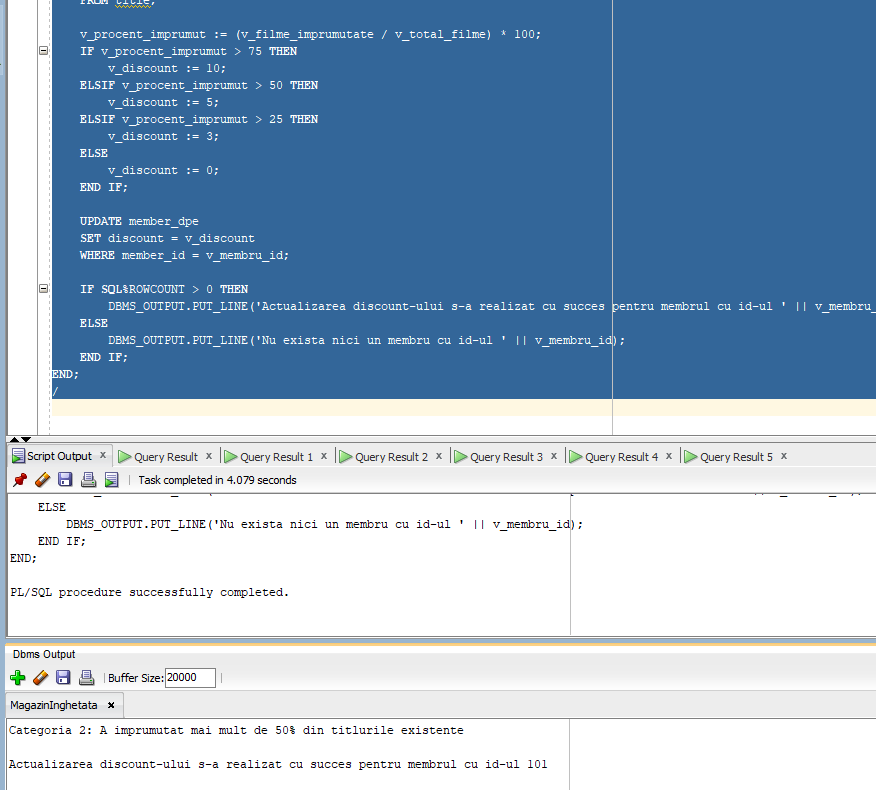
DBMS\_OUTPUT.PUT\_LINE('Nu exista nici un membru cu id-ul ' || v\_membru\_id);

END IF;

END;

/





E6.

VARIABLE rezultat VARCHAR2(100);

BEGIN

SELECT NumeMagazin

INTO :rezultat

FROM Magazine m

JOIN Angajati a ON m.IDMagazin = a.IDMagazin

GROUP BY NumeMagazin

HAVING COUNT(a.IDAngajat) = (

SELECT MAX(employee\_count)

FROM (

SELECT COUNT(\*) AS employee\_count

FROM Angajati

GROUP BY IDMagazin

)

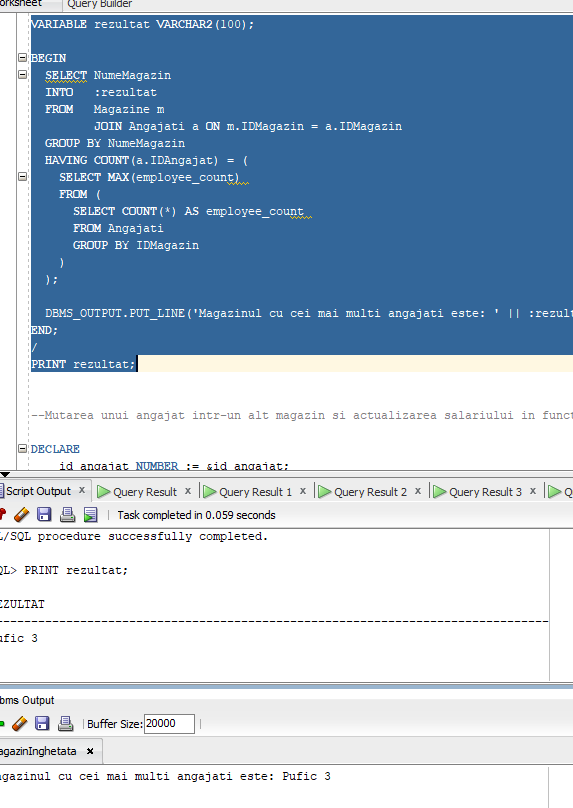
);

DBMS\_OUTPUT.PUT\_LINE('Magazinul cu cei mai multi angajati este: ' || :rezultat);

END;

/

PRINT rezultat;



SET VERIFY OFF;

DECLARE

v\_cod Angajati.IDAngajat%TYPE := &p\_cod;

v\_bonus NUMBER(8);

v\_salariu NUMBER(8);

v\_salariu\_max NUMBER(8);

BEGIN

BEGIN

SELECT Salariu INTO v\_salariu

FROM Angajati

WHERE IDAngajat = v\_cod;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nu exista niciun angajat cu codul ' || v\_cod);

RETURN;

END;

SELECT MAX(Salariu) INTO v\_salariu\_max

FROM Angajati;

-- Comparatia salariului curent cu procentele din salariul maxim

IF v\_salariu < (v\_salariu\_max \* 0.25) THEN

v\_bonus := 0;

ELSIF v\_salariu < (v\_salariu\_max \* 0.50) THEN

v\_bonus := 1000;

ELSIF v\_salariu < (v\_salariu\_max \* 0.75) THEN

v\_bonus := 2000;

ELSE

v\_bonus := 3000;

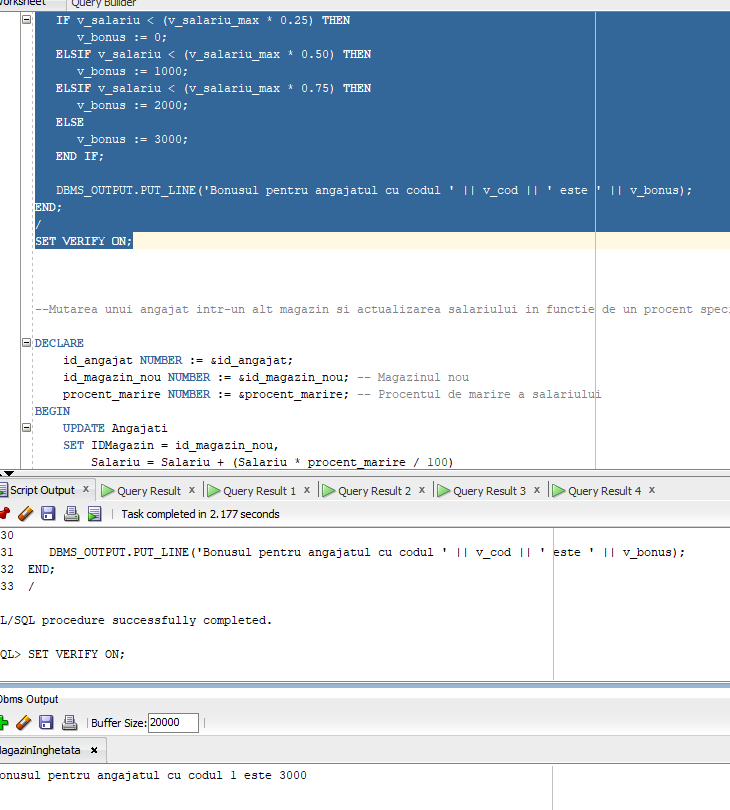
END IF;

DBMS\_OUTPUT.PUT\_LINE('Bonusul pentru angajatul cu codul ' || v\_cod || ' este ' || v\_bonus);

END;

/

SET VERIFY ON;



--Mutarea unui angajat intr-un alt magazin si actualizarea salariului in functie de un procent specific

DEFINE p\_cod\_sal = 1

DEFINE p\_cod\_mag = 3

DEFINE p\_procent = 23

DECLARE

v\_cod\_sal Angajati.IDAngajat%TYPE := &p\_cod\_sal;

v\_cod\_mag Angajati.IDMagazin%TYPE := &p\_cod\_dept;

v\_procent NUMBER(8) := &p\_procent;

BEGIN

UPDATE Angajati

SET IDMagazin = v\_cod\_mag,

Salariu = Salariu + (Salariu \* v\_procent / 100)

WHERE IDAngajat = v\_cod\_sal;

IF SQL%ROWCOUNT = 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Nu exista un angajat cu acest cod');

ELSIF SQL%ROWCOUNT > 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Angajatul a fost mutat si salariul actualizat.');

END IF;

ROLLBACK;

END;

/

