**Tema Curs – S7**

**OPRAN ANDREI 242**

SET SERVEROUTPUT ON;

------------------------------- EX1 ----------------------------------

-- Folosind 2 cursoare (dintre care unul sa fie parametrizat), afisati datile in care

-- un membru a rezervat un film care nu este disponibil pentru inchiriat in acest moment

DECLARE

    CURSOR c1 IS -- cursorul c1 va memora toti membrii

        SELECT

            distinct MEMBER\_ID

        FROM MEMBER\_LBC;

    CURSOR c2 (v\_member\_id NUMBER) IS -- cursorul c2 va memora cand a inchiriat un membru un film care nu este disponibil momentan

    SELECT

        DISTINCT R.BOOK\_DATE

    FROM RENTAL\_LBC R

    JOIN TITLE\_COPY\_LBC TC ON R.TITLE\_ID = TC.TITLE\_ID

    WHERE R.MEMBER\_ID = v\_member\_id AND UPPER(TC.STATUS) != 'AVAILABLE';

    v\_member\_id MEMBER\_LBC.MEMBER\_ID%TYPE;

    v\_book\_date RENTAL\_LBC.BOOK\_DATE%TYPE;

    TYPE rezultat IS TABLE OF RENTAL\_LBC.BOOK\_DATE%TYPE;

    v\_rezultat rezultat := rezultat(); -- vector in care o sa memoram datele cerute

    v\_ok BOOLEAN; -- verificare daca mai exista deja acea data in v\_rezultat

BEGIN

    OPEN c1;

    LOOP

        FETCH c1 INTO v\_member\_id;

        EXIT WHEN c1%NOTFOUND;

        OPEN c2(v\_member\_id);

        LOOP

            FETCH c2 INTO v\_book\_date;

            EXIT WHEN c2%NOTFOUND;

            v\_ok := TRUE;

            FOR i IN 1..v\_rezultat.COUNT LOOP -- verificam daca am memorat deja data curenta

                IF v\_rezultat(i) = v\_book\_date THEN

                    v\_ok := FALSE;

                    EXIT;

                END IF;

            END LOOP;

            IF v\_ok THEN -- daca data este intalnita prima data atunci o memoram

                v\_rezultat.EXTEND;

                v\_rezultat(v\_rezultat.COUNT) := v\_book\_date;

            END IF;

        END LOOP;

        CLOSE c2;

    END LOOP;

    CLOSE c1;

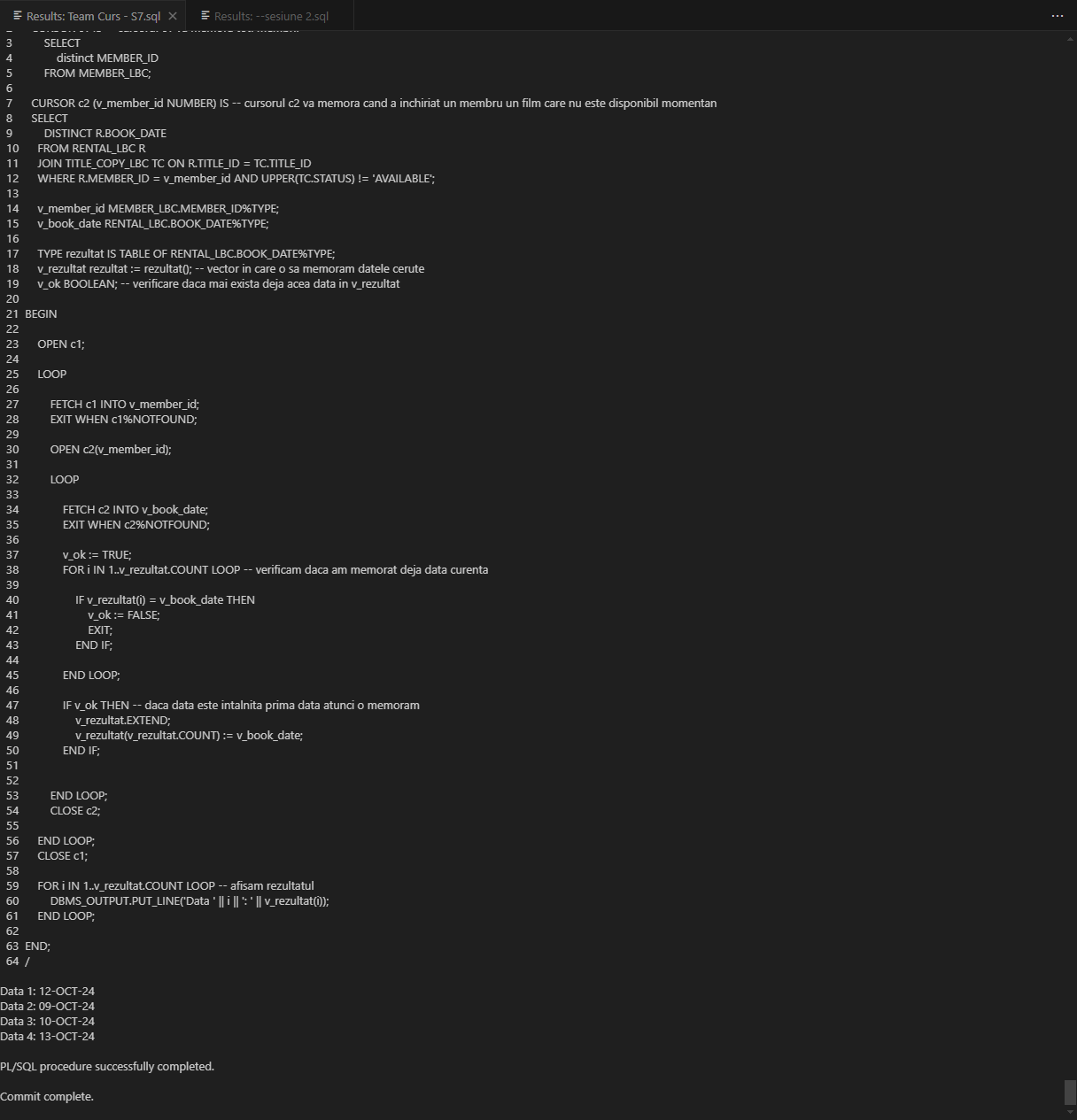
    FOR i IN 1..v\_rezultat.COUNT LOOP -- afisam rezultatul

        DBMS\_OUTPUT.PUT\_LINE('Data ' || i || ': ' || v\_rezultat(i));

    END LOOP;

END;

/

****

-- Folosind 2 cursoare clasice, afisati datile in care

-- un membru a rezervat un film care nu este disponibil pentru inchiriat in acest moment

DECLARE

    CURSOR c1 IS -- cursorul c1 va memora toti membrii

        SELECT

            DISTINCT MEMBER\_ID

        FROM MEMBER\_LBC;

    CURSOR c2 IS -- cursorul c2 va memora cand a inchiriat un membru un film care nu este disponibil momentan

        SELECT

            R.MEMBER\_ID,

            R.BOOK\_DATE

        FROM RENTAL\_LBC R

        JOIN TITLE\_COPY\_LBC TC ON R.TITLE\_ID = TC.TITLE\_ID

        WHERE UPPER(TC.STATUS) != 'AVAILABLE';

    v\_member\_id MEMBER\_LBC.MEMBER\_ID%TYPE;

    v\_member\_id\_2 RENTAL\_LBC.MEMBER\_ID%TYPE;

    v\_book\_date RENTAL\_LBC.BOOK\_DATE%TYPE;

    TYPE rezultat IS TABLE OF RENTAL\_LBC.BOOK\_DATE%TYPE;

    v\_rezultat rezultat := rezultat(); -- vector in care o sa memoram datele cerute

    v\_ok BOOLEAN; -- verificare daca mai exista deja acea data in v\_rezultat

BEGIN

    OPEN c1;

    LOOP

        FETCH c1 INTO v\_member\_id;

        EXIT WHEN c1%NOTFOUND;

        OPEN c2;

        LOOP

            FETCH c2 INTO v\_member\_id\_2, v\_book\_date;

            EXIT WHEN c2%NOTFOUND;

            IF v\_member\_id\_2 = v\_member\_id THEN

                v\_ok := TRUE;

                FOR i IN 1..v\_rezultat.COUNT LOOP -- verificam daca am memorat deja data curenta

                    IF v\_rezultat(i) = v\_book\_date THEN

                        v\_ok := FALSE;

                        EXIT;

                    END IF;

                END LOOP;

                IF v\_ok THEN -- daca data este intalnita prima data atunci o memoram

                    v\_rezultat.EXTEND;

                    v\_rezultat(v\_rezultat.COUNT) := v\_book\_date;

                END IF;

            END IF;

        END LOOP;

        CLOSE c2;

    END LOOP;

    CLOSE c1;

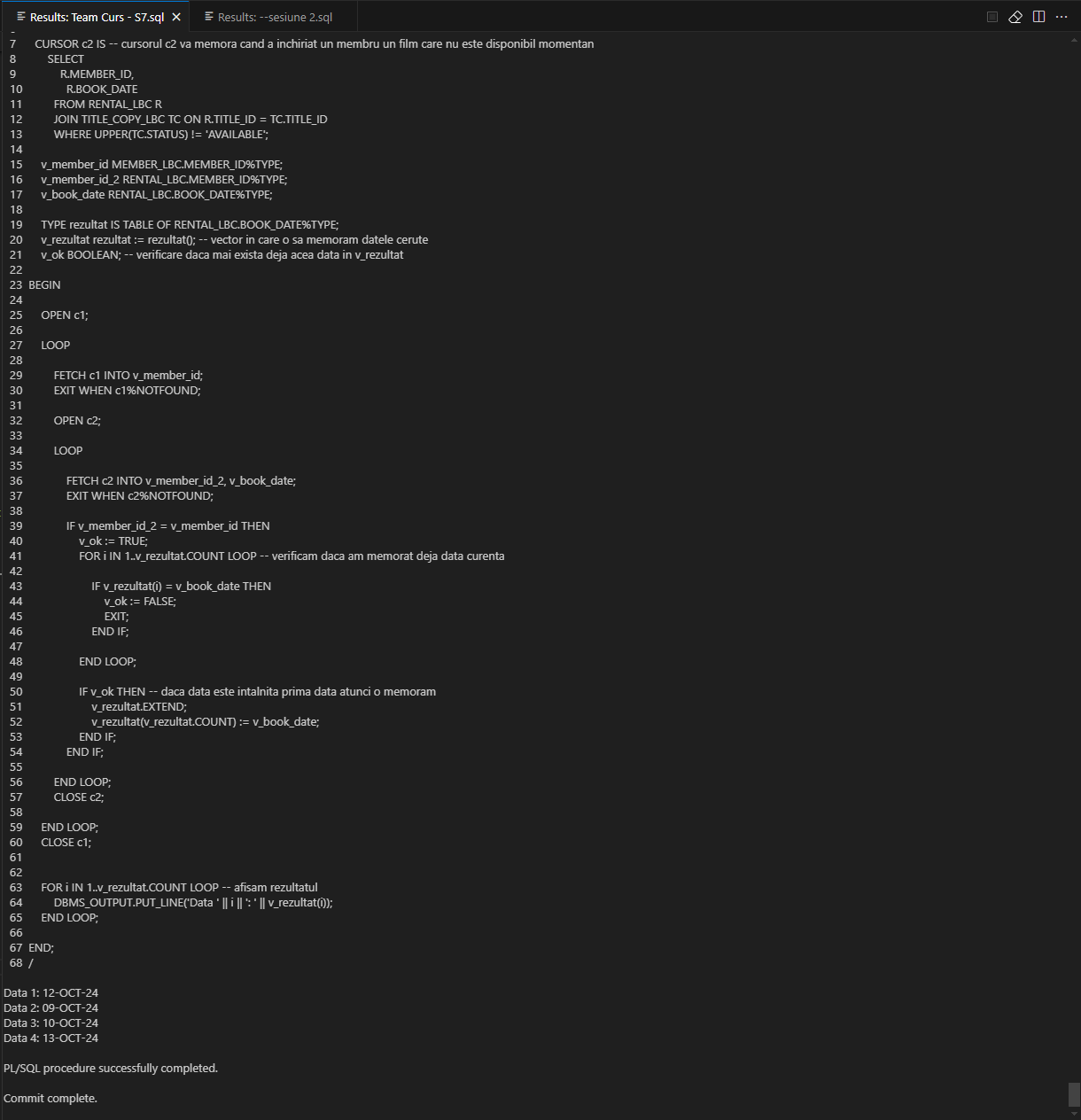
    FOR i IN 1..v\_rezultat.COUNT LOOP -- afisam rezultatul

        DBMS\_OUTPUT.PUT\_LINE('Data ' || i || ': ' || v\_rezultat(i));

    END LOOP;

END;

/

****

-- Folosind 2 ciclu cursoare, afisati datile in care

-- un membru a rezervat un film care nu este disponibil pentru inchiriat in acest moment

DECLARE

    CURSOR c1 IS -- cursorul c1 va memora toti membrii

        SELECT

            DISTINCT MEMBER\_ID

        FROM MEMBER\_LBC;

    CURSOR c2 IS -- cursorul c2 va memora cand a inchiriat un membru un film care nu este disponibil momentan

        SELECT

            R.MEMBER\_ID,

            R.BOOK\_DATE

        FROM RENTAL\_LBC R

        JOIN TITLE\_COPY\_LBC TC ON R.TITLE\_ID = TC.TITLE\_ID

        WHERE UPPER(TC.STATUS) != 'AVAILABLE';

    v\_member\_id MEMBER\_LBC.MEMBER\_ID%TYPE;

    v\_member\_id\_2 RENTAL\_LBC.MEMBER\_ID%TYPE;

    v\_book\_date RENTAL\_LBC.BOOK\_DATE%TYPE;

    TYPE rezultat IS TABLE OF RENTAL\_LBC.BOOK\_DATE%TYPE;

    v\_rezultat rezultat := rezultat(); -- vector in care o sa memoram datele cerute

    v\_ok BOOLEAN; -- verificare daca mai exista deja acea data in v\_rezultat

BEGIN

    FOR inregistrare IN c1 LOOP

        v\_member\_id := inregistrare.MEMBER\_ID; -- FETCH

        FOR inregistrare\_2 IN c2 LOOP

            v\_member\_id\_2 := inregistrare\_2.MEMBER\_ID; -- FETCH

            v\_book\_date := inregistrare\_2.BOOK\_DATE; -- FETCH

            IF v\_member\_id\_2 = v\_member\_id THEN

                v\_ok := TRUE;

                FOR i IN 1..v\_rezultat.COUNT LOOP -- verificam daca am memorat deja data curenta

                    IF v\_rezultat(i) = v\_book\_date THEN

                        v\_ok := FALSE;

                        EXIT;

                    END IF;

                END LOOP;

                IF v\_ok THEN -- daca data este intalnita prima data atunci o memoram

                    v\_rezultat.EXTEND;

                    v\_rezultat(v\_rezultat.COUNT) := v\_book\_date;

                END IF;

            END IF;

        END LOOP;

    END LOOP;

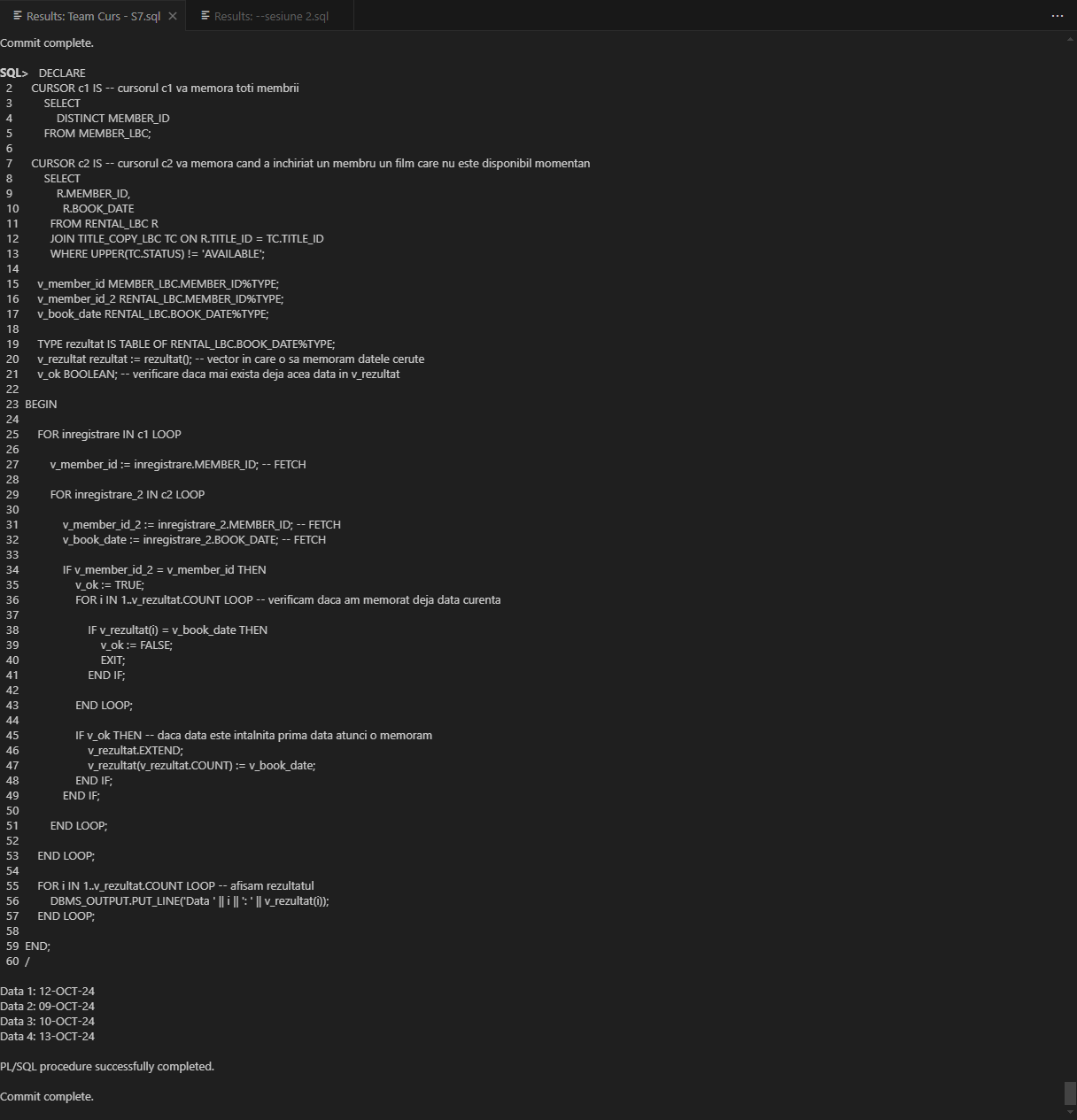
    FOR i IN 1..v\_rezultat.COUNT LOOP -- afisam rezultatul

        DBMS\_OUTPUT.PUT\_LINE('Data ' || i || ': ' || v\_rezultat(i));

    END LOOP;

END;

/

****

-- Folosind 2 expresii cursor, afisati datile in care

-- un membru a rezervat un film care nu este disponibil pentru inchiriat in acest moment

DECLARE

    CURSOR c1 IS

        SELECT

            MEMBER\_ID,

            CURSOR(

                SELECT DISTINCT R.BOOK\_DATE

                FROM RENTAL\_LBC R

                JOIN TITLE\_COPY\_LBC TC ON R.TITLE\_ID = TC.TITLE\_ID

                WHERE R.MEMBER\_ID = MEMBER\_LBC.MEMBER\_ID AND UPPER(TC.STATUS) != 'AVAILABLE'

            ) AS c2

        FROM MEMBER\_LBC;

    v\_member\_id MEMBER\_LBC.MEMBER\_ID%TYPE;

    v\_book\_date RENTAL\_LBC.BOOK\_DATE%TYPE;

    TYPE rezultat IS TABLE OF RENTAL\_LBC.BOOK\_DATE%TYPE;

    v\_rezultat rezultat := rezultat(); -- vector in care o sa memoram datele cerute

    v\_ok BOOLEAN; -- verificare daca mai exista deja acea data in v\_rezultat

    v\_cursor SYS\_REFCURSOR; -- cursor imbricat

BEGIN

    OPEN c1;

    LOOP

        FETCH c1 INTO v\_member\_id, v\_cursor;

        EXIT WHEN c1%NOTFOUND;

        LOOP

            FETCH v\_cursor INTO v\_book\_date;

            EXIT WHEN v\_cursor%NOTFOUND;

            v\_ok := TRUE;

            FOR i IN 1..v\_rezultat.COUNT LOOP -- verificam daca am memorat deja data curenta

                IF v\_rezultat(i) = v\_book\_date THEN

                    v\_ok := FALSE;

                    EXIT;

                END IF;

            END LOOP;

            IF v\_ok THEN -- daca data este intalnita prima data atunci o memoram

                v\_rezultat.EXTEND;

                v\_rezultat(v\_rezultat.COUNT) := v\_book\_date;

            END IF;

        END LOOP;

        CLOSE v\_cursor;

    END LOOP;

    CLOSE c1;

    FOR i IN 1..v\_rezultat.COUNT LOOP -- afisam rezultatul

        DBMS\_OUTPUT.PUT\_LINE('Data ' || i || ': ' || v\_rezultat(i));

    END LOOP;

END;

/

****

------------------------ EX2 ----------------------------

-- Pe schema video, exemplificati, ca la curs, ce se intampla cand avem doua sesiuni si folosim

-- select...for update in prima sesiune, iar in a doua avem select...for update nowait/ wait n / skip locked.(Hint: Ex 5.11).

--sesiune 1

SELECT \* FROM MEMBER\_LBC

WHERE MEMBER\_id=101

FOR UPDATE;

commit;

SELECT \* FROM MEMBER\_LBC

WHERE MEMBER\_id=101

FOR UPDATE NOWAIT;

SELECT \* FROM MEMBER\_LBC

WHERE MEMBER\_id=102

FOR UPDATE NOWAIT;

--sesiune 2

SELECT \* FROM MEMBER\_LBC

WHERE MEMBER\_id=101

FOR UPDATE NOWAIT;

SELECT \* FROM MEMBER\_LBC

WHERE MEMBER\_id=102

FOR UPDATE NOWAIT;

SELECT \* FROM MEMBER\_LBC

WHERE MEMBER\_id=103

FOR UPDATE NOWAIT;

-- daca suntem in sesiunea 1:

    -- putem rula toate 3 comenzile fara probleme

-- daca suntem in sesiunea 2 si am rulat prima comanda impreuna cu commit in sesiunea 1:

    -- nu vom putea rula a doua si a treia comanda deoarece sunt blocate de sesiunea 1 (de catre prima comanda cu commit) - vom primi eroare

    -- a treia cerere merge deoarece member\_id = 103 nu a fost menionat in sesiunea 1

**Sesiunea 1:**

--sesiune 1

SELECT \* FROM MEMBER\_LBC

WHERE MEMBER\_id=101

FOR UPDATE;

commit;

SELECT \* FROM MEMBER\_LBC

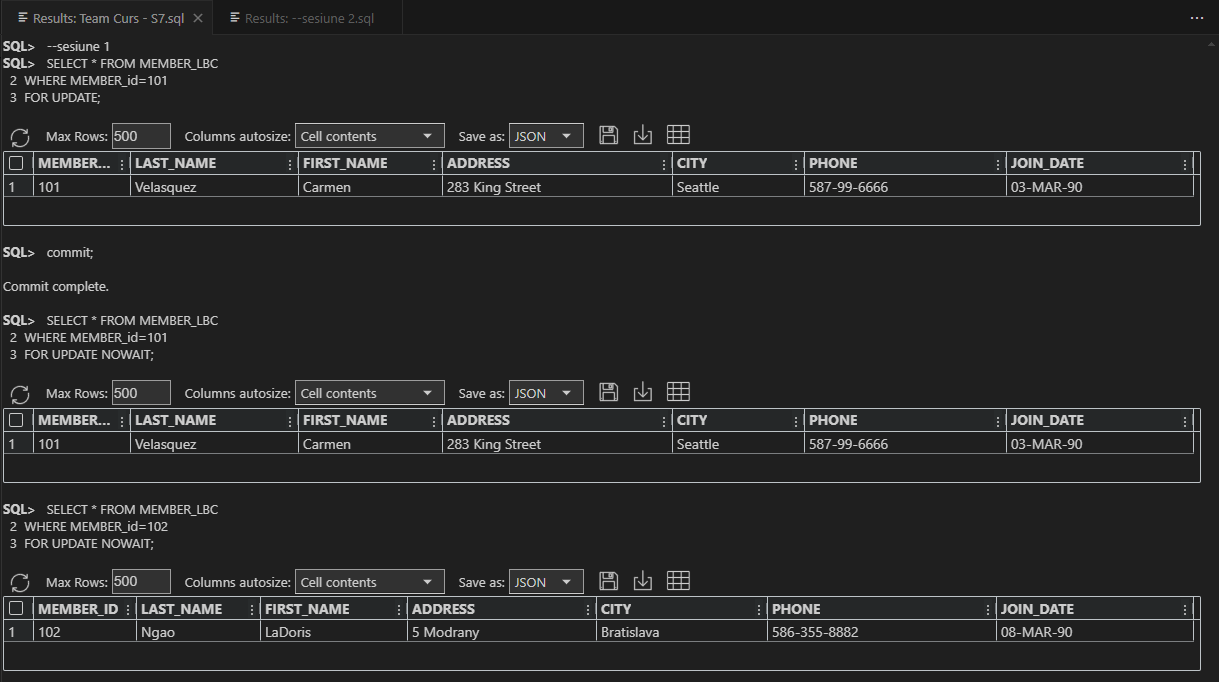
WHERE MEMBER\_id=101

FOR UPDATE NOWAIT;

SELECT \* FROM MEMBER\_LBC

WHERE MEMBER\_id=102

FOR UPDATE NOWAIT;

****

**Sesiunea 2:**

SELECT \* FROM MEMBER\_LBC

WHERE MEMBER\_id=101

FOR UPDATE NOWAIT;

SELECT \* FROM MEMBER\_LBC

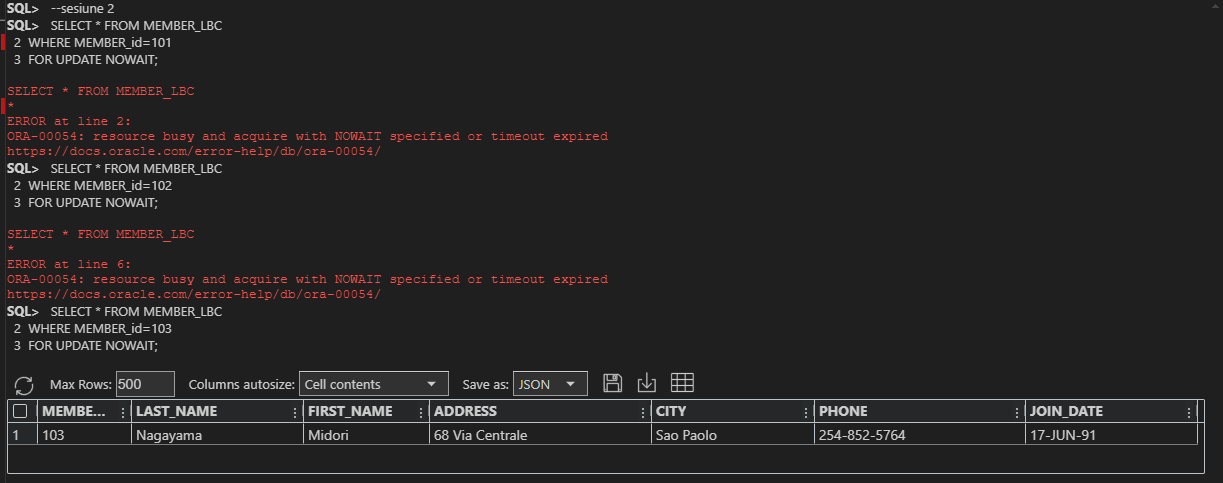
WHERE MEMBER\_id=102

FOR UPDATE NOWAIT;

SELECT \* FROM MEMBER\_LBC

WHERE MEMBER\_id=103

FOR UPDATE NOWAIT;

****