**Temă Curs SGBD – S7**

**Șerba Ioana-Raluca grupa 241**

**Exercițiul 1:**

* Enunț: Pe schema video, compuneti in limbaj natural o problema a carei rezolvare implica folosirea a doua cursoare dintre care unul sa fie cursor parametrizat. Rezolvati aceeasi problema folosind doua cursoare clasice, doua ciclu cursoare si expresii cursor. (Hint: Ex 5.10)
* Cod Sursă:

*--TEMA CURS 7  
--E1  
--problema pe schema video  
--afisati cele mai inchiriate titluri  
  
--PS: fiecare subpunct o sa aiba afisare diferita  
--ca sa le pot diferntia in timp ce rulez  
  
--2 cursoare dintre care unul parametrizat*/  
DECLARE  
 CURSOR TITLE\_C IS SELECT TITLE, T.TITLE\_ID  
 FROM TITLE T;  
   
 CURSOR MAX\_RENTALS (v\_title TITLE.TITLE\_ID%TYPE) IS  
 SELECT *COUNT*(COPY\_ID)  
 FROM RENTAL  
 WHERE TITLE\_ID = V\_TITLE  
 GROUP BY TITLE\_ID;  
  
 rent\_maxim NUMBER(4);  
 TITLEC\_ID TITLE.TITLE\_ID%TYPE;  
 TITLEC\_TITLE TITLE.TITLE%TYPE;  
 TITLEC\_MAXR NUMBER(4);  
   
  
BEGIN  
 SELECT *MAX*(*COUNT*(COPY\_ID))  
 INTO RENT\_MAXIM  
 FROM RENTAL  
 GROUP BY TITLE\_ID;  
   
 OPEN TITLE\_C;  
 LOOP  
 FETCH TITLE\_C INTO TITLEC\_TITLE, TITLEC\_ID;  
 EXIT WHEN TITLE\_C%NOTFOUND;  
 OPEN MAX\_RENTALS(TITLEC\_ID);  
 FETCH MAX\_RENTALS INTO TITLEC\_MAXR;  
 EXIT WHEN MAX\_RENTALS%NOTFOUND;  
 IF TITLEC\_MAXR = RENT\_MAXIM THEN   
 DBMS\_OUTPUT.*PUT\_LINE* (TITLEC\_TITLE);  
 END IF;  
 CLOSE MAX\_RENTALS;  
 END LOOP;  
 CLOSE TITLE\_C;  
   
END;  
/  
*--yippie a dat bine*

*A screenshot of a computer

Description automatically generated  
  
--2 cursoare clasice*DECLARE  
 CURSOR TITLE\_C IS SELECT TITLE, T.TITLE\_ID  
 FROM TITLE T;  
   
 CURSOR MAX\_RENTALS IS  
 SELECT TITLE\_ID, *COUNT*(COPY\_ID)  
 FROM RENTAL  
 GROUP BY TITLE\_ID;  
  
 rent\_maxim NUMBER(4);  
 TITLEC\_ID TITLE.TITLE\_ID%TYPE;  
 TITLEC\_TITLE TITLE.TITLE%TYPE;  
 TITLEC\_MAXR NUMBER(4);  
 MR\_ID TITLE.TITLE\_ID%TYPE;  
 MR\_COUNT NUMBER(4);  
   
  
BEGIN  
 SELECT *MAX*(*COUNT*(COPY\_ID))  
 INTO RENT\_MAXIM  
 FROM RENTAL  
 GROUP BY TITLE\_ID;  
   
 OPEN TITLE\_C;  
 LOOP  
 FETCH TITLE\_C INTO TITLEC\_TITLE, TITLEC\_ID;  
 EXIT WHEN TITLE\_C%NOTFOUND;  
 *--DBMS\_OUTPUT.PUT\_LINE(TITLEC\_TITLE || TITLEC\_ID);* OPEN MAX\_RENTALS;  
 LOOP  
 FETCH MAX\_RENTALS INTO MR\_ID, MR\_COUNT;  
 EXIT WHEN MAX\_RENTALS%NOTFOUND;  
 *--IF MR\_ID = TITLEC\_ID THEN DBMS\_OUTPUT.PUT\_LINE(MR\_ID ||MR\_COUNT || RENT\_MAXIM);* IF MR\_ID = TITLEC\_ID AND MR\_COUNT = RENT\_MAXIM THEN  
 DBMS\_OUTPUT.*PUT\_LINE*('TITLUL: ' || TITLEC\_TITLE || ' ARE NR MAXIM DE INCHIRIERI: ' || MR\_COUNT);  
 END IF;  
 END LOOP;  
 CLOSE MAX\_RENTALS;  
 END LOOP;  
 CLOSE TITLE\_C;  
   
END;  
/

A screenshot of a computer

Description automatically generated

*--2 ciclu cursoare   
--asem cu cursoare clasice, insa le deschid cu for*DECLARE  
 CURSOR TITLE\_C IS SELECT TITLE, TITLE\_ID AS ID  
 FROM TITLE;  
   
 CURSOR MAX\_RENTALS IS  
 SELECT TITLE\_ID AS ID, *COUNT*(COPY\_ID) AS NR\_COPY  
 FROM RENTAL  
 GROUP BY TITLE\_ID;  
  
 rent\_maxim NUMBER(4);  
   
   
  
BEGIN  
 SELECT *MAX*(*COUNT*(COPY\_ID))  
 INTO RENT\_MAXIM  
 FROM RENTAL  
 GROUP BY TITLE\_ID;  
   
 FOR TITLEC IN TITLE\_C LOOP  
 FOR MR IN MAX\_RENTALS LOOP  
 IF MR.ID = TITLEC.ID AND MR.NR\_COPY = RENT\_MAXIM THEN  
 DBMS\_OUTPUT.*PUT\_LINE*(TITLEC.TITLE || ' ' || MR.NR\_COPY);  
 END IF;  
 END LOOP;  
 END LOOP;  
   
END;  
/  
  
  
 A screenshot of a computer

Description automatically generated   
  
/  
  
*--expresii cursor*DECLARE  
 CURSOR TITLE\_MAX IS  
 SELECT TITLE\_ID, TITLE,  
 CURSOR (   
 SELECT *COUNT*(COPY\_ID)  
 FROM RENTAL R  
 WHERE T.TITLE\_ID = R.TITLE\_ID  
 HAVING *COUNT*(COPY\_ID) = (SELECT *MAX*(*COUNT*(COPY\_ID)) FROM RENTAL GROUP BY TITLE\_ID)  
 )  
  
 FROM TITLE T;  
   
 V\_CURSOR SYS\_REFCURSOR;  
 T\_ID TITLE.TITLE\_ID%TYPE;  
 T\_T TITLE.TITLE%TYPE;  
 T\_COUNT NUMBER(4);  
   
BEGIN  
 OPEN TITLE\_MAX;  
 LOOP   
 FETCH TITLE\_MAX INTO T\_ID, T\_T, V\_CURSOR;  
 EXIT WHEN TITLE\_MAX%NOTFOUND;  
 LOOP  
 FETCH V\_CURSOR INTO T\_COUNT;  
 EXIT WHEN V\_CURSOR%NOTFOUND;  
   
 DBMS\_OUTPUT.*PUT\_LINE*('TITLUL CEL MAI INCHIRIAT: ' || T\_T);  
 END LOOP;  
 END LOOP;  
 CLOSE TITLE\_MAX;  
END;  
/   
*--NU STIU cat de corecta a fost logica pe acest exemplu insa sper ca am implementat cat de cat bine expresiile cursor*

A screenshot of a computer

Description automatically generated

**Exercițiul 2:**

* Enunț: 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).
* Cod Sursă:

*--sesiune 1* SELECT *\** FROM TITLE  
 WHERE TITLE\_ID = 94 FOR UPDATE NOWAIT;  
 *--COMMIT;  
   
  
--SESIUNE 2* SELECT *\** FROM TITLE  
 WHERE TITLE\_ID = 94  
 FOR UPDATE NOWAIT;  
 *--CAT TIMP NU DAU COMMIT, MEREU VA DA EROARE* SELECT *\** FROM TITLE  
 WHERE TITLE\_ID = 94  
 FOR UPDATE WAIT 10;   
 *--DACA NU FAC NIMIC IN SESIUNEA 1, DA EROARE  
 --DACA DAU COMMIT IN SESIUNEA 1 IN 10< SECUNDE, IMI INTOARCE LINIA* SELECT *\** FROM TITLE  
 WHERE TITLE\_ID = 94 OR TITLE\_ID = 95  
 FOR UPDATE SKIP LOCKED;   
 *--COMMIT;*

* Blocarea linii in sesiunea 1

A screenshot of a computer

Description automatically generated

-sesiunea 2:

Programul nu așteaptă iar cum liniile sunt blocate, dă eroare.

A screenshot of a computer

Description automatically generated

-programul așteaptă 10 secunde

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Apoi, pentru că nu am dat commit, dă în continuarea eroare.

A screenshot of a computer

Description automatically generated

Iar aici sare liniile blocate (id = 94) si afiseaza doar id=95