**Temă 7 Lab SGBD 22.12.2024**

**Șerba Ioana Raluca grupa 241**

**Cerință:**

* Rezolvati ex E7 din Laborator PLSQL4.pdf. : Adaptați cerințele exercițiilor 2 și 4 (folosind ca bază cerințele exercițiilor 1, respectiv 3) pentru diagrama proiectului prezentată la materia Baze de Date din anul I. Rezolvați aceste două exerciții în PL/SQL, folosind baza de date proprie.
* Cerinta 2 adaptata:

Definiti un subprogram prin care sa obtineti numele trupei al unui album specificat. Tratati toate exceptiile ce pot fi generate. Rezolvati folosind o **functie stocata.**

* **Cod sursă:**

*--Definiti un subprogram prin care sa obtineti numele trupei al unui album specificat.  
--Tratati toate exceptiile ce pot fi generate.  
-- Rezolvati folosind o functie stocata.*CREATE OR REPLACE FUNCTION *NUME\_TRUPA\_2* (VALBUM ALBUM.NUME%TYPE DEFAULT 'Best Hits of Queen')  
RETURN TRUPA.NUME%TYPE IS  
 NUME\_TRUPA TRUPA.NUME%TYPE;  
 BEGIN  
 SELECT T.NUME INTO NUME\_TRUPA  
 FROM TRUPA T  
 JOIN ALBUM A ON A.ID\_TRUPA = T.ID\_TRUPA  
 WHERE A.NUME = VALBUM;  
 RETURN NUME\_TRUPA;  
 EXCEPTION  
 WHEN NO\_DATA\_FOUND THEN  
 *RAISE\_APPLICATION\_ERROR*(-20001, 'Nu exista albume cu numele dat');  
 *--baza de date garanteaza ca albumul are trupa, deci no data found  
 --apare doar daca albumul nu exista, nu se poate ca trupa sa nu existe* WHEN TOO\_MANY\_ROWS THEN  
 *RAISE\_APPLICATION\_ERROR*(-20002, 'Exista mai multe albume cu numele dat');  
 WHEN OTHERS THEN  
 *RAISE\_APPLICATION\_ERROR*(-20000, 'Alta eroare a avut loc');  
END *NUME\_TRUPA\_2*;  
/  
  
BEGIN   
 DBMS\_OUTPUT.*PUT\_LINE*('Numele trupei este ' || *nume\_trupa\_2*);  
END;  
/  
BEGIN  
 DBMS\_OUTPUT.*PUT\_LINE*('Numele trupei este ' || *nume\_trupa\_2*('Digital Nightmare'));  
END;  
/  
BEGIN  
 DBMS\_OUTPUT.*PUT\_LINE*('Numele trupei este ' || *nume\_trupa\_2*('Best Hits'));  
END;  
/  
BEGIN  
 DBMS\_OUTPUT.*PUT\_LINE*('Numele trupei este ' || *nume\_trupa\_2*('miau'));  
END;  
/  
*--am modificat datele ca sa exista 2 albume cu numele miau*

* Printscreen

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer screen

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated**

* Cerinta 4 adaptata:

Rezolvați acelasi exerciți folosind o **procedură stocată.**

* Cod sursa:

*--4 - procedura stocata*CREATE OR REPLACE PROCEDURE *NUME\_TRUPA\_4* (VALBUM IN ALBUM.NUME%TYPE,  
 NUME\_TRUPA OUT TRUPA.NUME%TYPE ) IS  
 BEGIN  
 SELECT T.NUME INTO NUME\_TRUPA  
 FROM TRUPA T  
 JOIN ALBUM A ON A.ID\_TRUPA = T.ID\_TRUPA  
 WHERE A.NUME = VALBUM;  
 EXCEPTION  
 WHEN NO\_DATA\_FOUND THEN  
 *RAISE\_APPLICATION\_ERROR*(-20001, 'Nu exista albume cu numele dat');  
 *--baza de date garanteaza ca albumul are trupa, deci no data found  
 --apare doar daca albumul nu exista, nu se poate ca trupa sa nu existe* WHEN TOO\_MANY\_ROWS THEN  
 *RAISE\_APPLICATION\_ERROR*(-20002, 'Exista mai multe albume cu numele dat');  
 WHEN OTHERS THEN  
 *RAISE\_APPLICATION\_ERROR*(-20000, 'Alta eroare a avut loc');  
END *NUME\_TRUPA\_4*;   
/  
DECLARE   
 NUME\_TRUPA TRUPA.NUME%TYPE;  
BEGIN  
 *nume\_trupa\_4*('',NUME\_TRUPA);  
 DBMS\_OUTPUT.*PUT\_LINE*('Numele trupei este ' || NUME\_TRUPA);  
END;  
/  
DECLARE   
 NUME\_TRUPA TRUPA.NUME%TYPE;  
BEGIN  
 *nume\_trupa\_4*('Digital Nightmare',NUME\_TRUPA);  
 DBMS\_OUTPUT.*PUT\_LINE*('Numele trupei este ' || NUME\_TRUPA);  
END;  
/  
  
DECLARE   
 NUME\_TRUPA TRUPA.NUME%TYPE;  
BEGIN  
 *nume\_trupa\_4*('miau',NUME\_TRUPA);  
 DBMS\_OUTPUT.*PUT\_LINE*('Numele trupei este ' || NUME\_TRUPA);  
END;  
/

* Printscreen:

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated