E7. 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.

Enunt creat de mine:

Definiți un subprogram prin care sa obțineți pretul unui activ al carui nume este specificat.Tratați toate excepțiile ce pot fi generate.

Adaptarea exercitiului 4:

-- Procedura

CREATE OR REPLACE PROCEDURE GET\_PRET\_ACTIV\_PROC (

V\_NUME IN ACTIV.NUME%TYPE,

V\_PRET OUT ACTIV.PRET%TYPE

) IS

BEGIN

SELECT

PRET INTO V\_PRET

FROM

ACTIV

WHERE

NUME = V\_NUME;

DBMS\_OUTPUT.PUT\_LINE('Price of '

|| V\_NUME

|| ' is '

|| V\_PRET);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20000, 'Nu exista active cu numele dat');

WHEN TOO\_MANY\_ROWS THEN

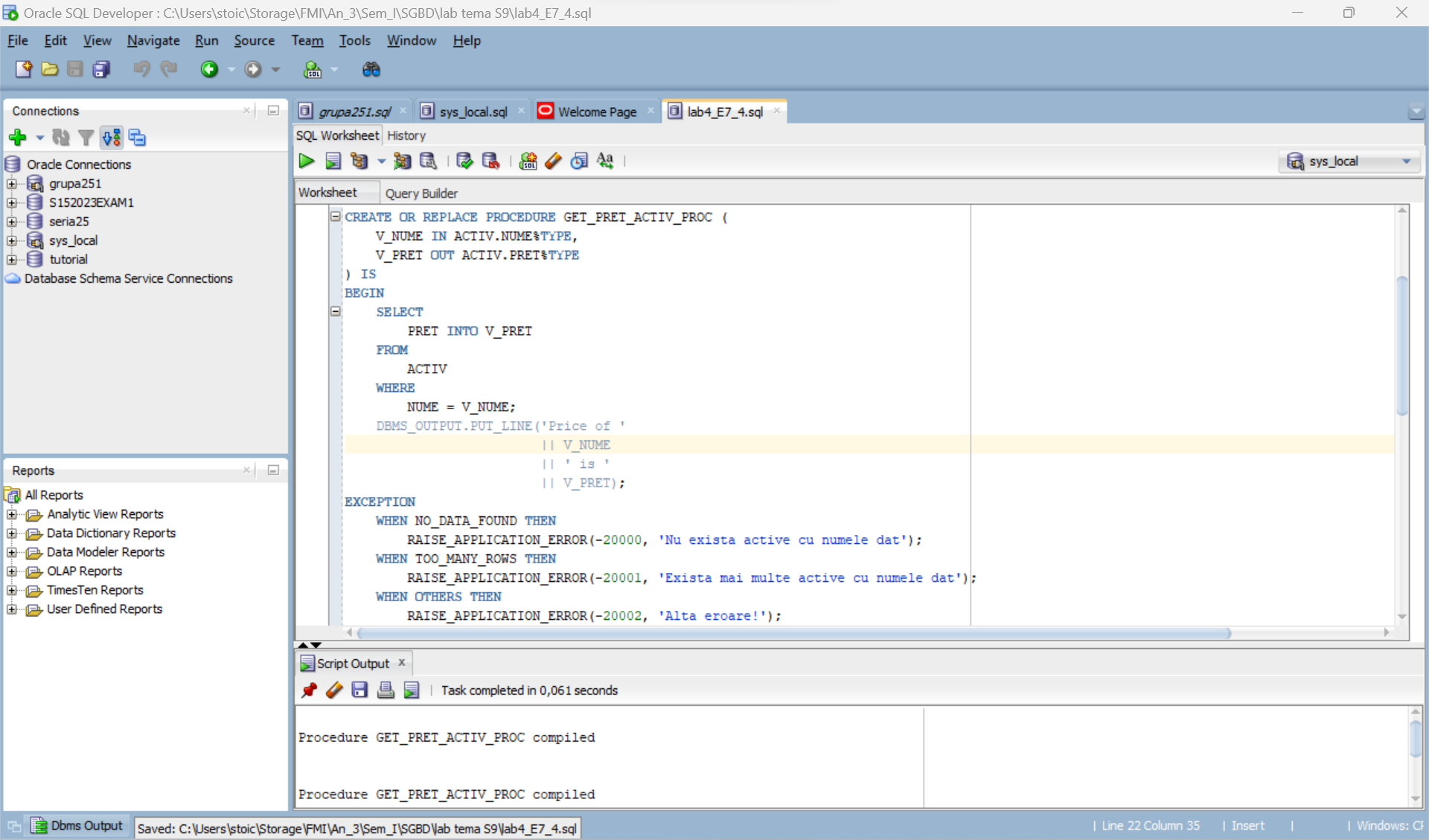
RAISE\_APPLICATION\_ERROR(-20001, 'Exista mai multe active cu numele dat');

WHEN OTHERS THEN

RAISE\_APPLICATION\_ERROR(-20002, 'Alta eroare!');

END GET\_PRET\_ACTIV\_PROC;

/



-- apelare obisnuita

DECLARE

PRET\_ACTIV NUMBER;

BEGIN

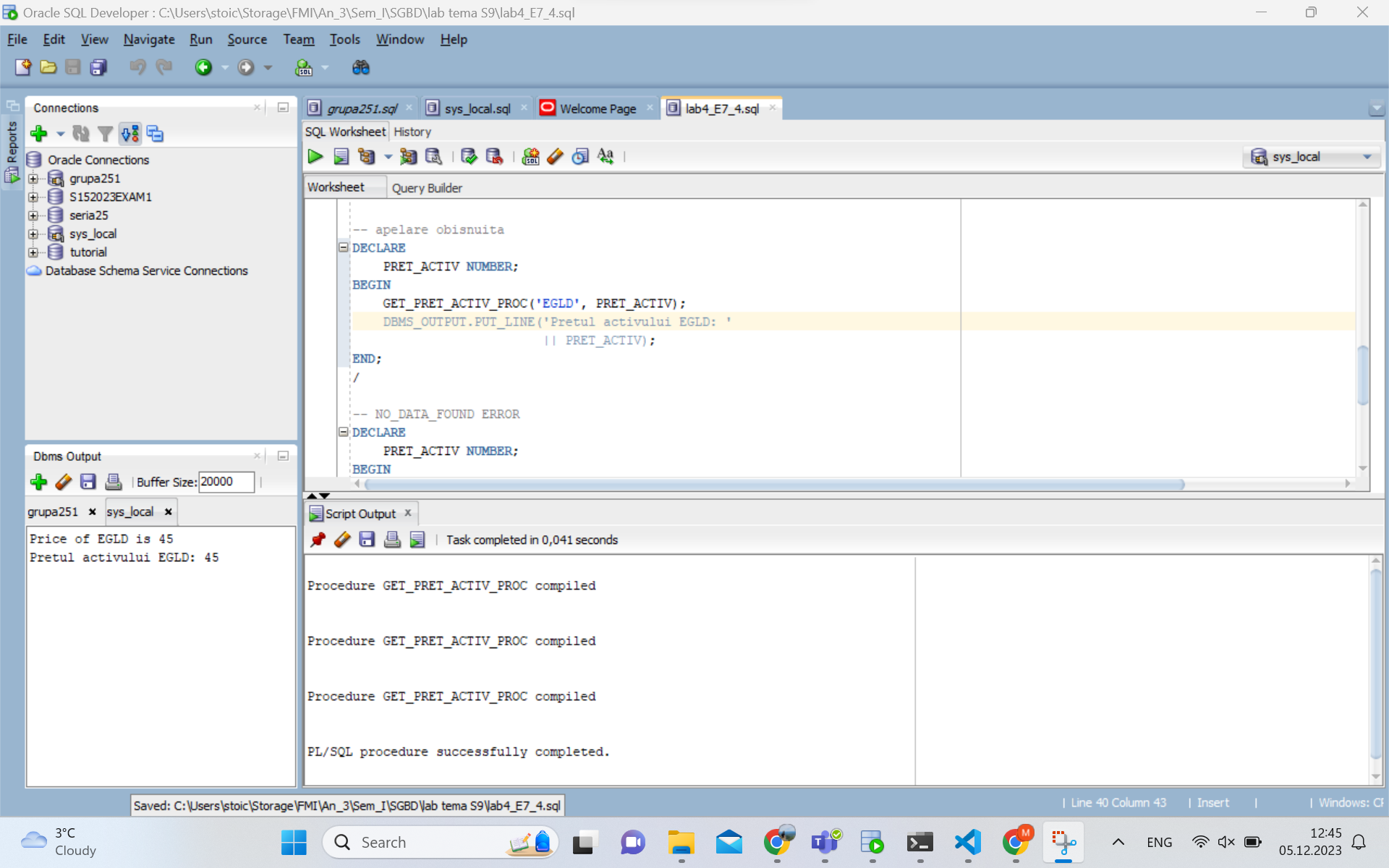
GET\_PRET\_ACTIV\_PROC('EGLD', PRET\_ACTIV);

DBMS\_OUTPUT.PUT\_LINE('Pretul activului EGLD: '

|| PRET\_ACTIV);

END;

/



-- NO\_DATA\_FOUND ERROR

DECLARE

PRET\_ACTIV NUMBER;

BEGIN

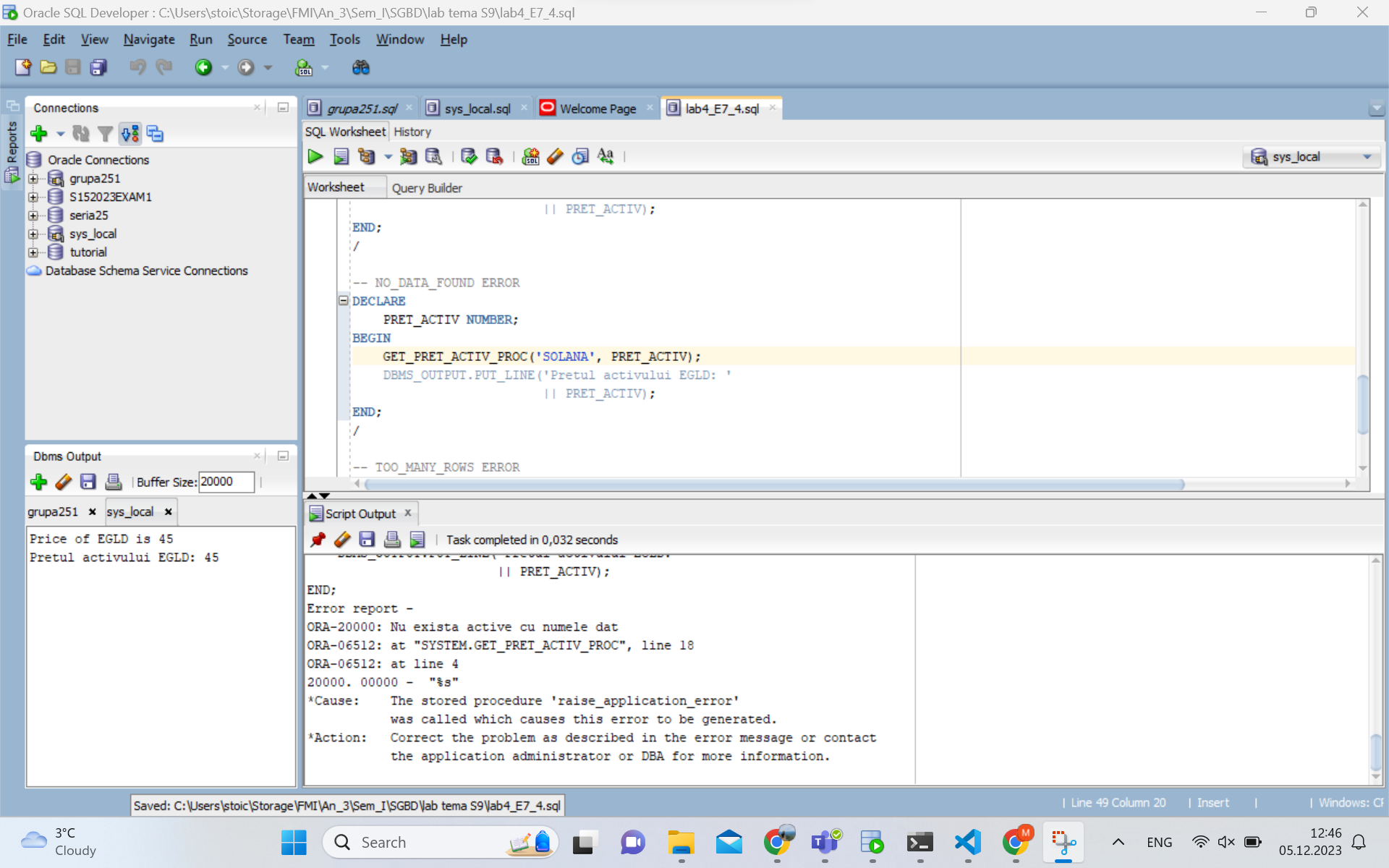
GET\_PRET\_ACTIV\_PROC('SOLANA', PRET\_ACTIV);

DBMS\_OUTPUT.PUT\_LINE('Pretul activului SOLANA: '

|| PRET\_ACTIV);

END;

/



-- TOO\_MANY\_ROWS ERROR

DECLARE

PRET\_ACTIV NUMBER;

BEGIN

GET\_PRET\_ACTIV\_PROC('ETH', PRET\_ACTIV);

DBMS\_OUTPUT.PUT\_LINE('Pretul activului ETH: '

|| PRET\_ACTIV);

END;

