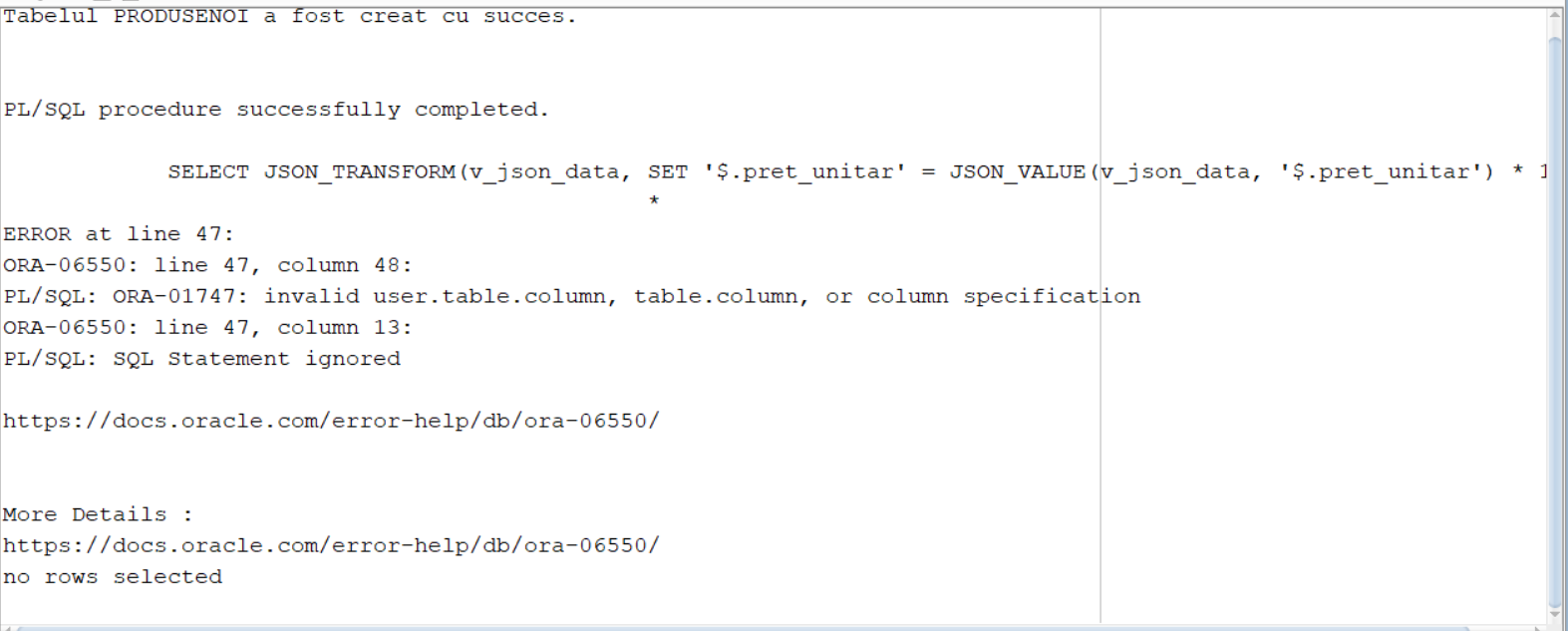
| Componența echipei |
| --- |
| Lepădatu Teodor |
| Asavei Roxana |

1. Implementați funcționalități care evidențiază diferențele dintre versiunile Oracle 19c, 21c și 23ai pentru conceptele prezentate la curs (fisier SGBD9 - PLSQL Exceptii.pdf), acolo unde este posibil, utilizând schema companie comercială.

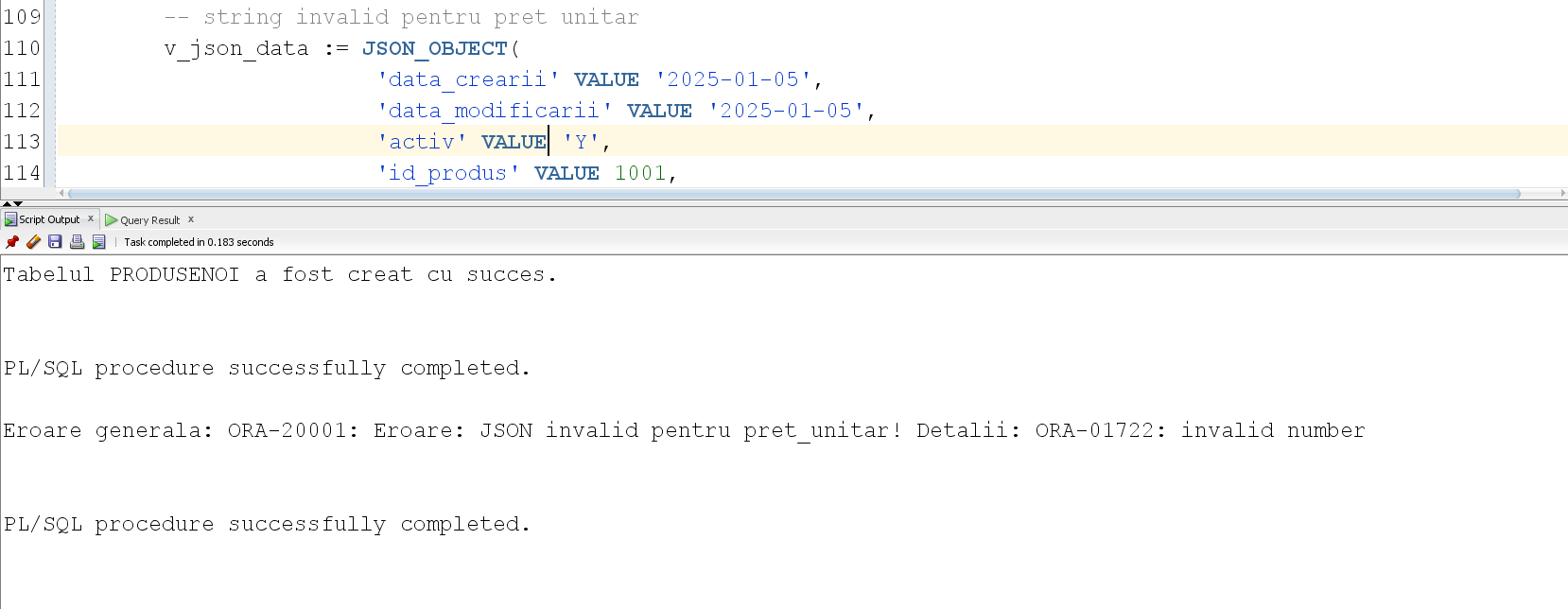
* Erori pe JSON

| SET SERVEROUTPUT ON;  BEGIN  -- Crearea tabelei daca nu exista  EXECUTE IMMEDIATE 'CREATE TABLE PRODUSENOI AS SELECT \* FROM PRODUSE WHERE 1=0';  DBMS\_OUTPUT.PUT\_LINE('Tabelul PRODUSENOI a fost creat cu succes.');  EXCEPTION  WHEN OTHERS THEN  IF SQLCODE = -955 THEN  DBMS\_OUTPUT.PUT\_LINE('Tabelul PRODUSENOI exista deja.');  ELSE  DBMS\_OUTPUT.PUT\_LINE('Eroare pentru crearea tabelului PRODUSENOI: ' || SQLERRM);  END IF;  END;  /  DECLARE  CURSOR c\_produse IS  SELECT JSON\_OBJECT(  'data\_crearii' VALUE TO\_CHAR(data\_crearii, 'YYYY-MM-DD'),  'data\_modificarii' VALUE TO\_CHAR(data\_modificarii, 'YYYY-MM-DD'),  'activ' VALUE activ,  'id\_produs' VALUE id\_produs,  'id\_categorie' VALUE id\_categorie,  'id\_um' VALUE id\_um,  'id\_zona' VALUE id\_zona,  'tva' VALUE tva,  'volum' VALUE volum,  'greutate' VALUE greutate,  'pret\_unitar' VALUE pret\_unitar,  'stoc\_impus' VALUE stoc\_impus,  'stoc\_curent' VALUE stoc\_curent,  'denumire' VALUE denumire,  'descriere' VALUE descriere  RETURNING CLOB) AS json\_data  FROM PRODUSE;  v\_json\_data CLOB;  v\_updated\_json CLOB;  BEGIN  FOR rec IN c\_produse LOOP  v\_json\_data := rec.json\_data;  -- string invalid pentru pret unitar  v\_json\_data := JSON\_OBJECT(  'data\_crearii' VALUE '2025-01-05',  'data\_modificarii' VALUE '2025-01-05',  'activ' VALUE 'Y',  'id\_produs' VALUE 1001,  'id\_categorie' VALUE 20,  'id\_um' VALUE 5,  'id\_zona' VALUE 3,  'tva' VALUE 19,  'volum' VALUE 10,  'greutate' VALUE 2,  'pret\_unitar' VALUE 'INVALID\_VALUE',  'stoc\_impus' VALUE 100,  'stoc\_curent' VALUE 50,  'denumire' VALUE 'Produs A',  'descriere' VALUE 'Descrierea produsului'  );  BEGIN  SELECT JSON\_TRANSFORM(v\_json\_data, SET '$.pret\_unitar' = JSON\_VALUE(v\_json\_data, '$.pret\_unitar') \* 1.10)  INTO v\_updated\_json  FROM dual;  EXCEPTION  WHEN OTHERS THEN  RAISE\_APPLICATION\_ERROR(-20001, 'Eroare: JSON invalid pentru pret\_unitar! Detalii: ' || SQLERRM);  END;  INSERT INTO PRODUSENOI (  data\_crearii, data\_modificarii, activ, id\_produs, id\_categorie, id\_um, id\_zona,  tva, volum, greutate, pret\_unitar, stoc\_impus, stoc\_curent, denumire, descriere  )  VALUES (  TO\_DATE(JSON\_VALUE(v\_updated\_json, '$.data\_crearii'), 'YYYY-MM-DD'),  TO\_DATE(JSON\_VALUE(v\_updated\_json, '$.data\_modificarii'), 'YYYY-MM-DD'),  JSON\_VALUE(v\_updated\_json, '$.activ'),  JSON\_VALUE(v\_updated\_json, '$.id\_produs'),  JSON\_VALUE(v\_updated\_json, '$.id\_categorie'),  JSON\_VALUE(v\_updated\_json, '$.id\_um'),  JSON\_VALUE(v\_updated\_json, '$.id\_zona'),  JSON\_VALUE(v\_updated\_json, '$.tva'),  JSON\_VALUE(v\_updated\_json, '$.volum'),  JSON\_VALUE(v\_updated\_json, '$.greutate'),  JSON\_VALUE(v\_updated\_json, '$.pret\_unitar'),  JSON\_VALUE(v\_updated\_json, '$.stoc\_impus'),  JSON\_VALUE(v\_updated\_json, '$.stoc\_curent'),  JSON\_VALUE(v\_updated\_json, '$.denumire'),  JSON\_VALUE(v\_updated\_json, '$.descriere')  );  END LOOP;  DBMS\_OUTPUT.PUT\_LINE('Succes.');  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('Eroare generala: ' || SQLERRM);  END;  /  -- Afisarea rezultatelor  SELECT \*  FROM PRODUSENOI; |
| --- |

19c



21c



23ai

