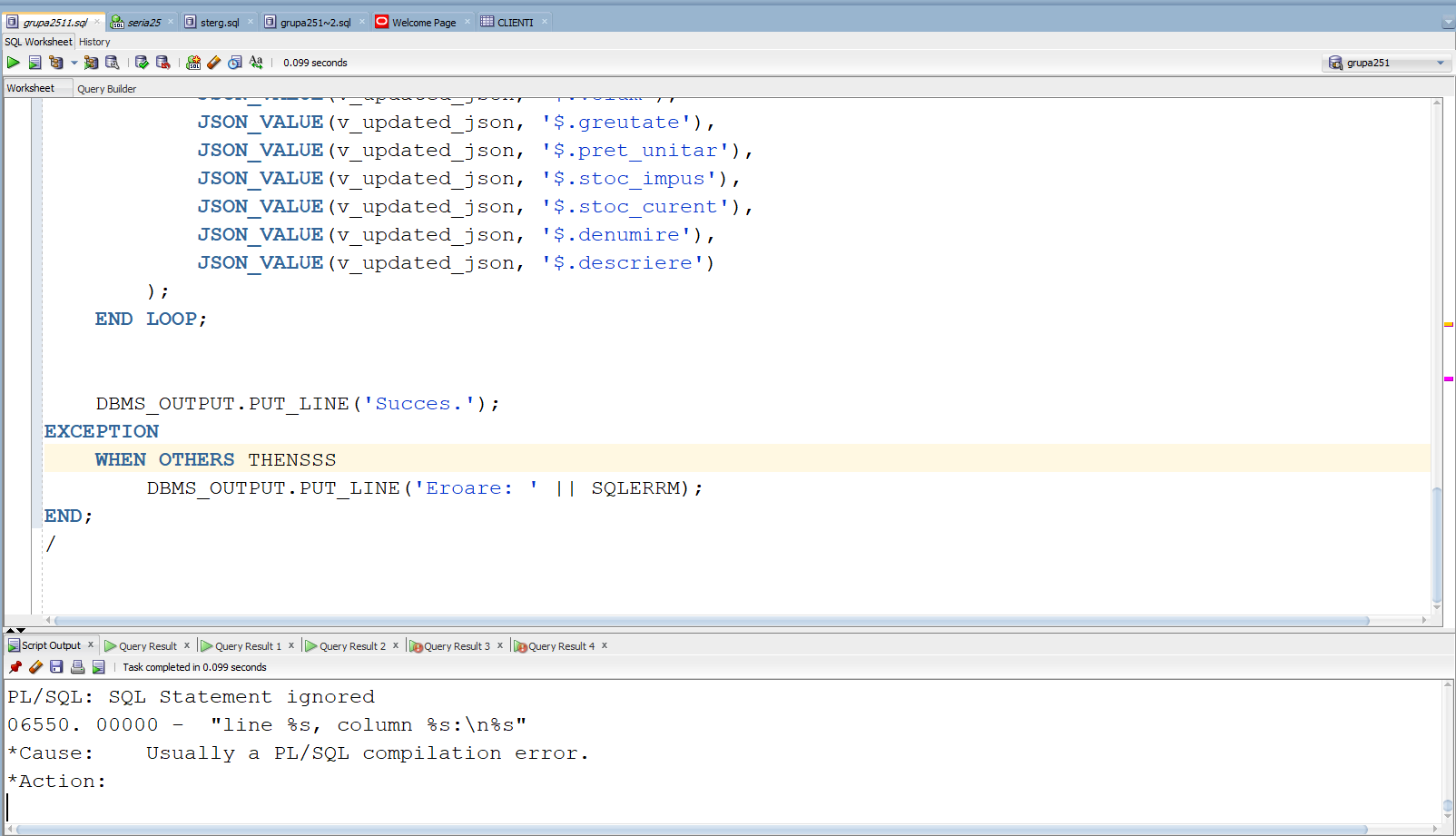
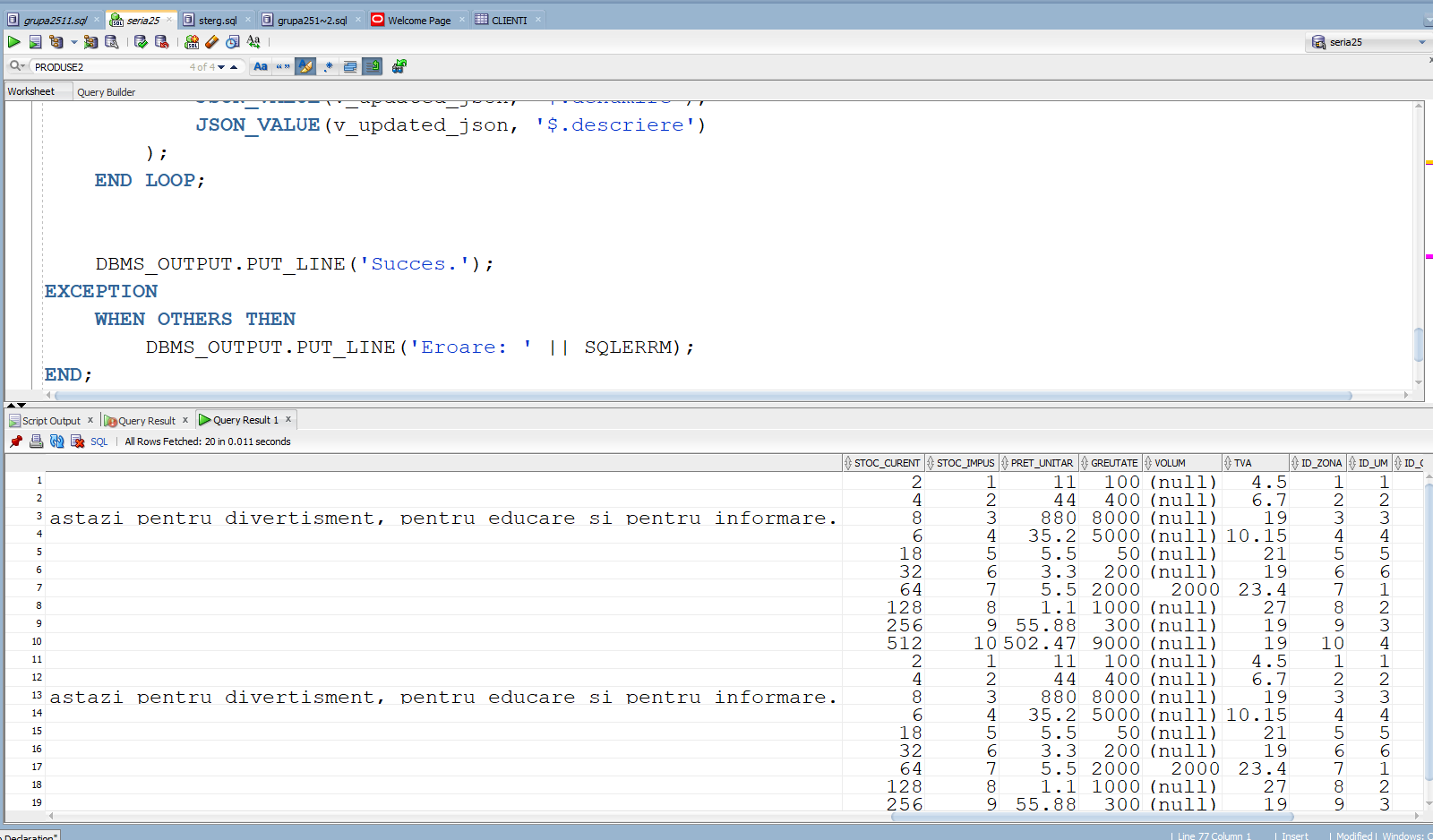
Implementați funcționalități care evidențiază diferențele dintre versiunile Oracle 19c, 21c și 23ai pentru conceptele prezentate la curs (fisier SGBD4 - PLSQL Tipuri de date.pdf), acolo unde este posibil, utilizând schema companie comercială.

* JSON\_TRANSFORM

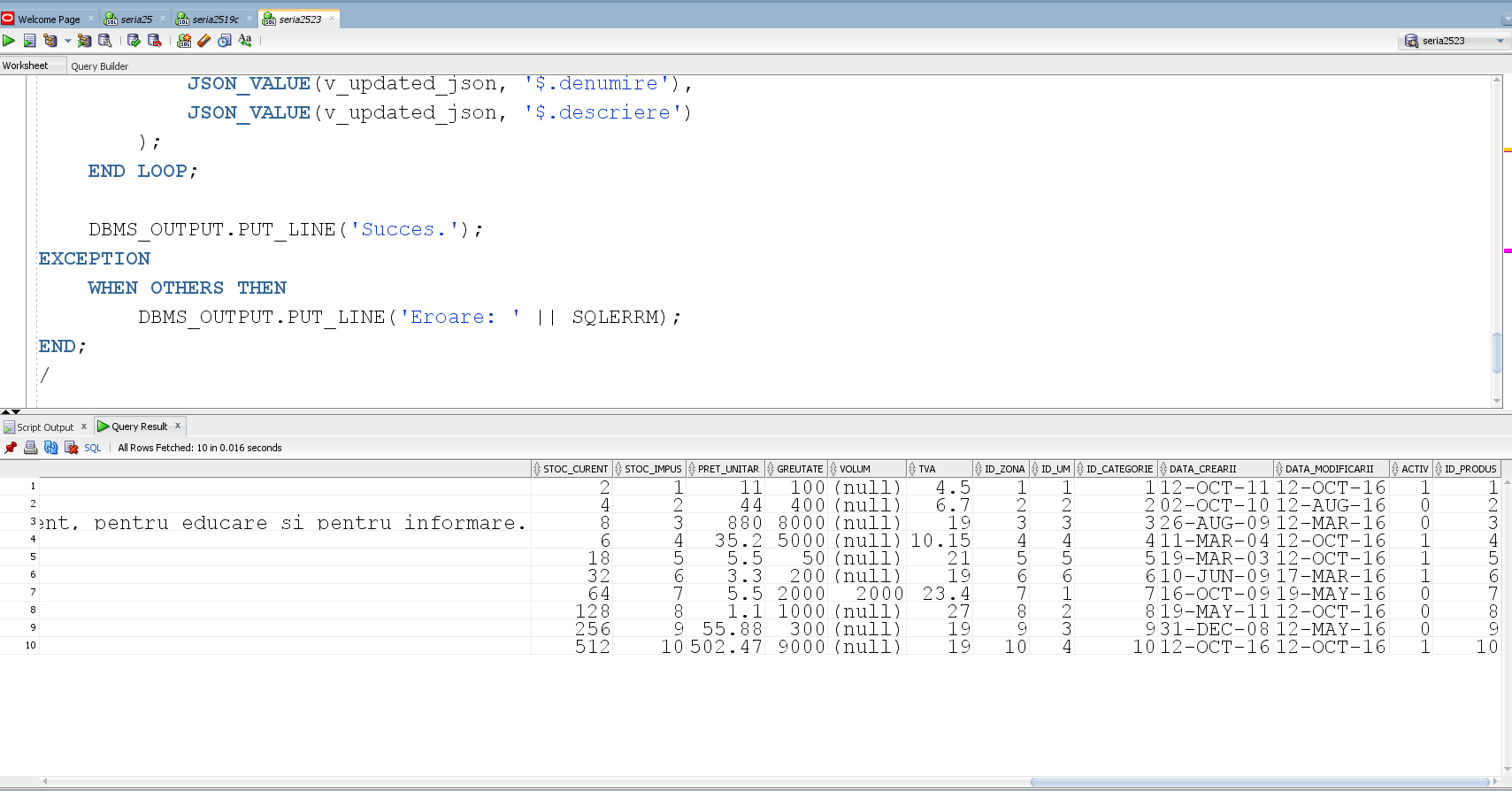
| BEGIN  EXECUTE IMMEDIATE 'CREATE TABLE PRODUSENOI AS SELECT \* FROM PRODUSE WHERE 1=0';  DBMS\_OUTPUT.PUT\_LINE('Tabelul PRODUSENOI a fost creat cu succes.');  EXCEPTION  WHEN OTHERS THEN  IF SQLCODE = -955 THEN  DBMS\_OUTPUT.PUT\_LINE('Tabelul PRODUSENOI exista deja.');  ELSE  DBMS\_OUTPUT.PUT\_LINE('Eroare 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;  SELECT JSON\_TRANSFORM(v\_json\_data, SET '$.pret\_unitar' = JSON\_VALUE(v\_json\_data, '$.pret\_unitar') \* 1.10)  INTO v\_updated\_json  FROM dual;  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: ' || SQLERRM);  END;  /  SELECT \*  FROM PRODUSENOI; |
| --- |

19c



21c

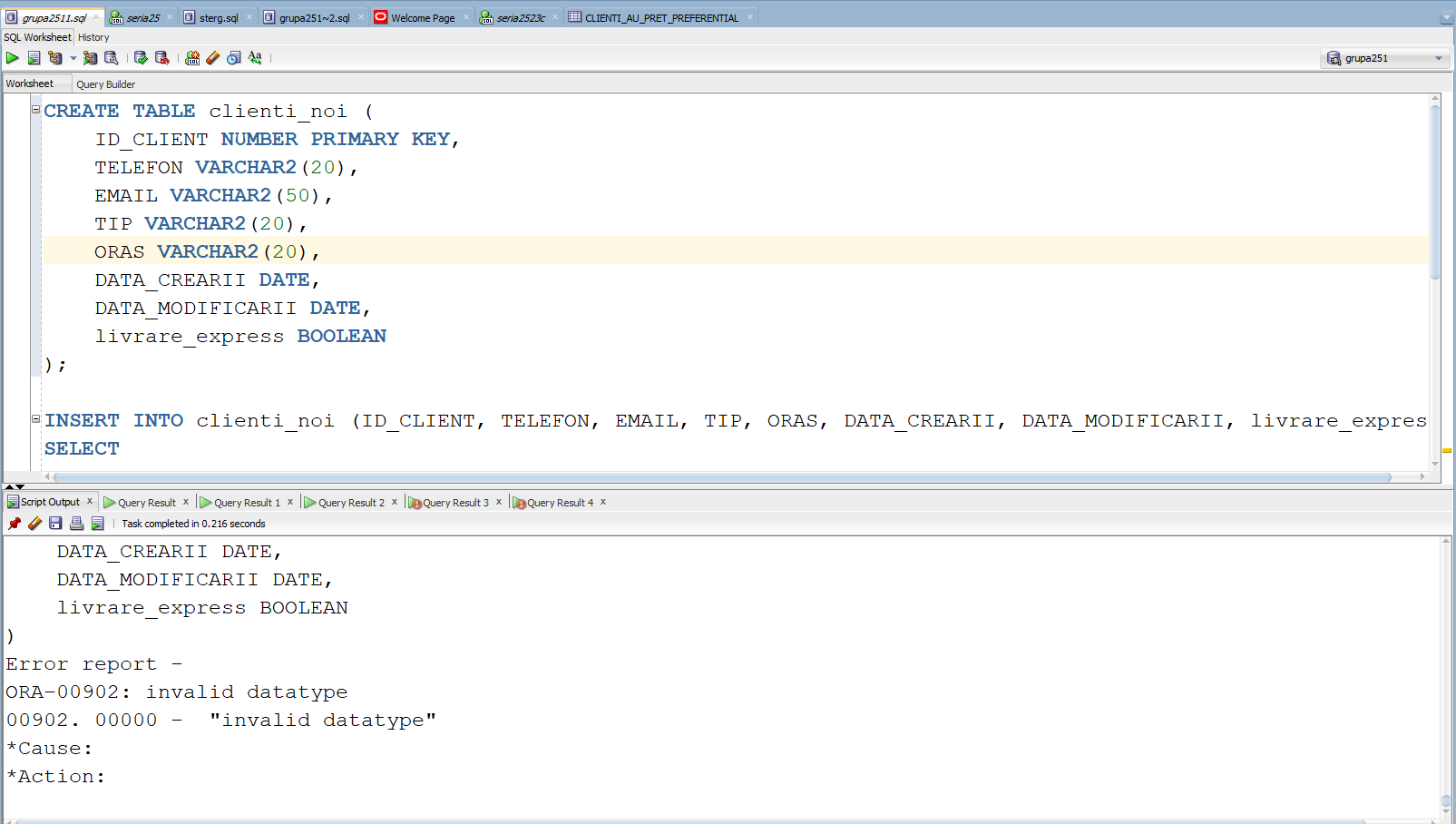
23ai



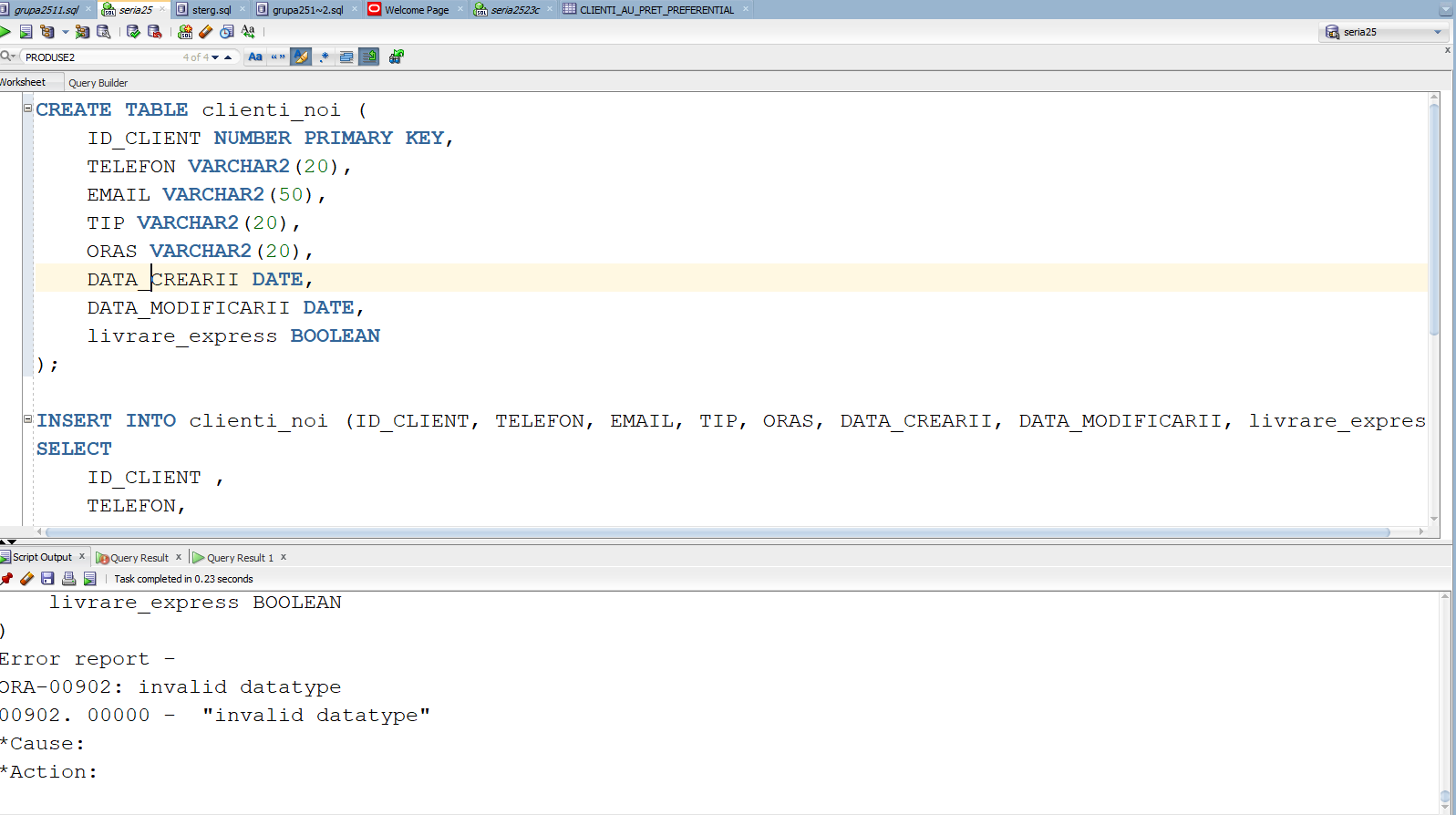
* BOOLEAN data type

| CREATE TABLE clienti\_noi (  ID\_CLIENT NUMBER PRIMARY KEY,  TELEFON VARCHAR2(20),  EMAIL VARCHAR2(50),  TIP VARCHAR2(20),  ORAS VARCHAR2(20),  DATA\_CREARII DATE,  DATA\_MODIFICARII DATE,  livrare\_express BOOLEAN  );  INSERT INTO clienti\_noi (ID\_CLIENT, TELEFON, EMAIL, TIP, ORAS, DATA\_CREARII, DATA\_MODIFICARII, livrare\_express)  SELECT  ID\_CLIENT ,  TELEFON,  EMAIL,  TIP,  ORAS,  DATA\_CREARII,  DATA\_MODIFICARII,  CASE  WHEN LOWER(ORAS) = 'bucuresti' THEN TRUE  ELSE FALSE  END AS livrare\_express  FROM clienti; |
| --- |

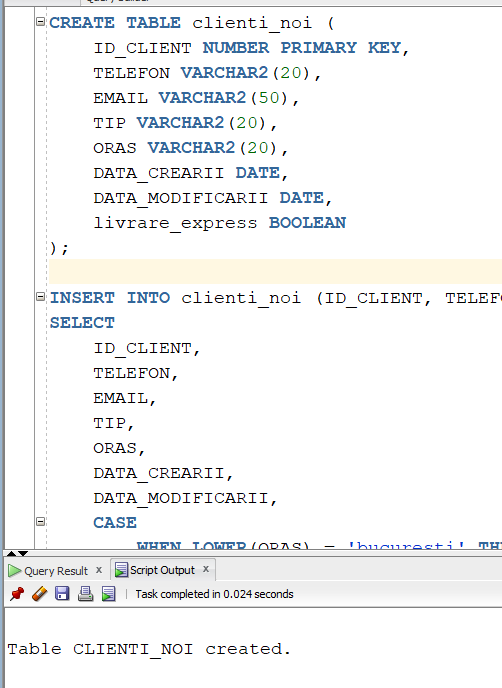
19c



21c



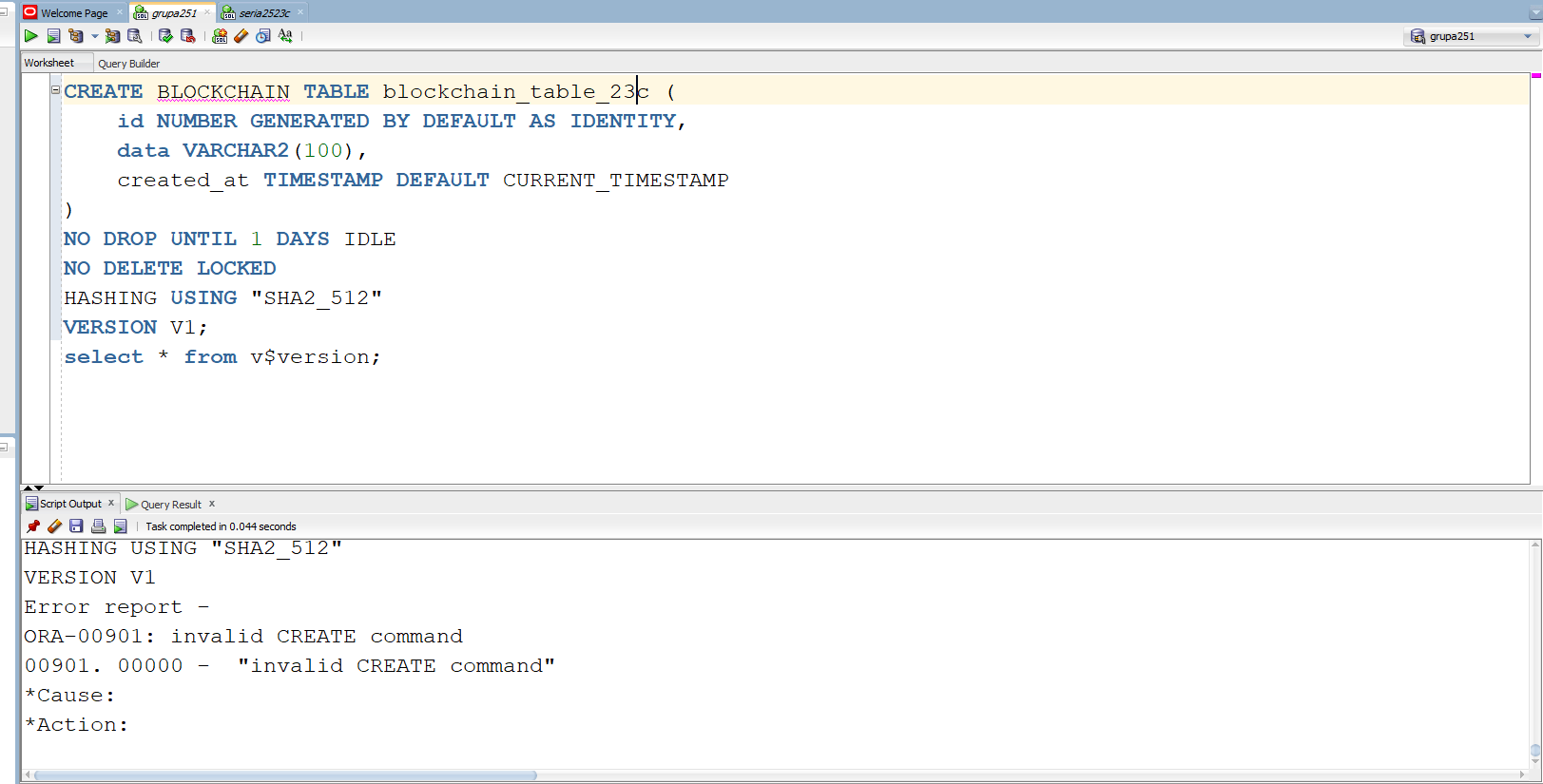
23c



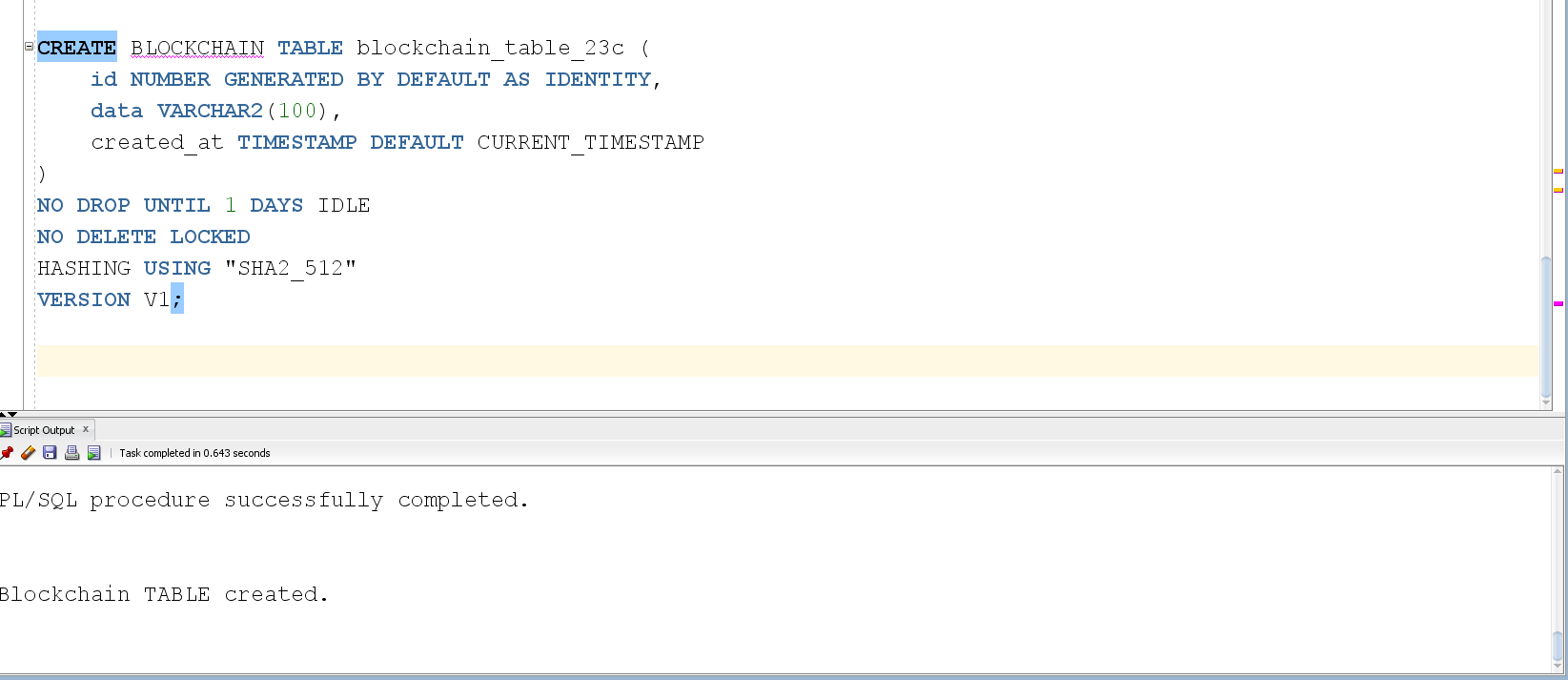
* Blockchain Table

| CREATE BLOCKCHAIN TABLE blockchain\_table\_23c (  id NUMBER GENERATED BY DEFAULT AS IDENTITY,  data VARCHAR2(100),  created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  )  NO DROP UNTIL 1 DAYS IDLE  NO DELETE LOCKED  HASHING USING "SHA2\_512"  VERSION V1; |
| --- |

19c



21c



23c

