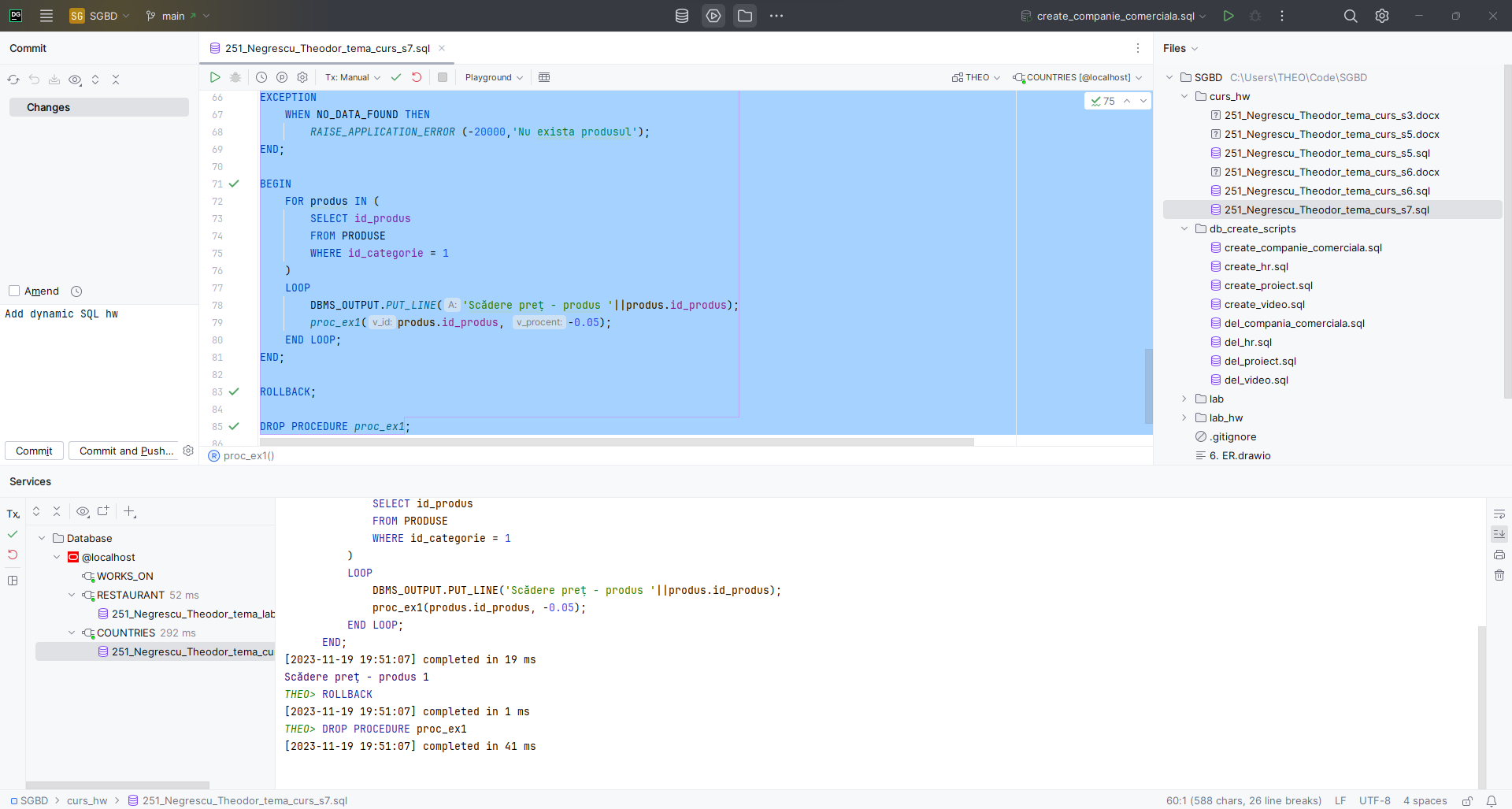
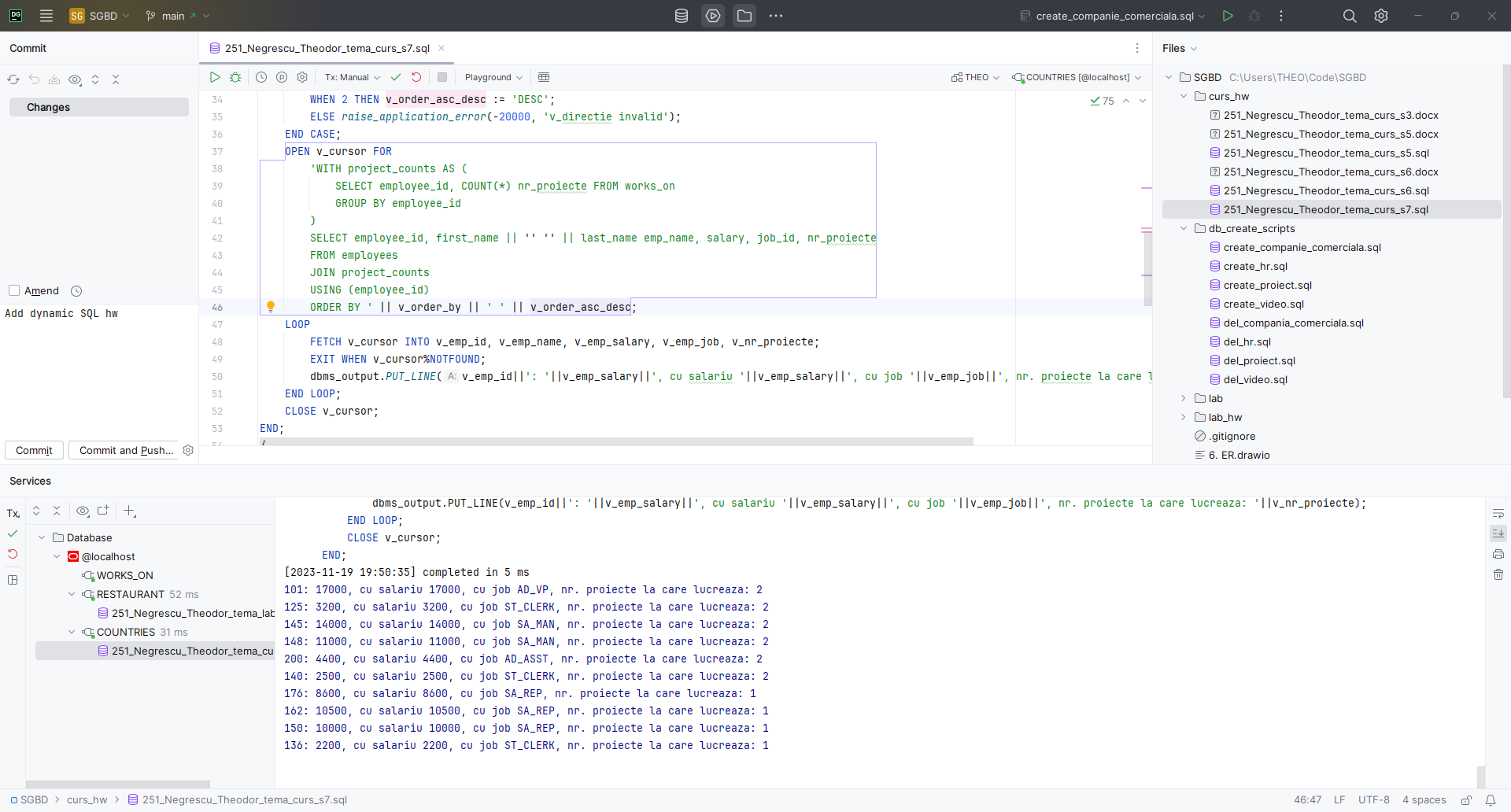
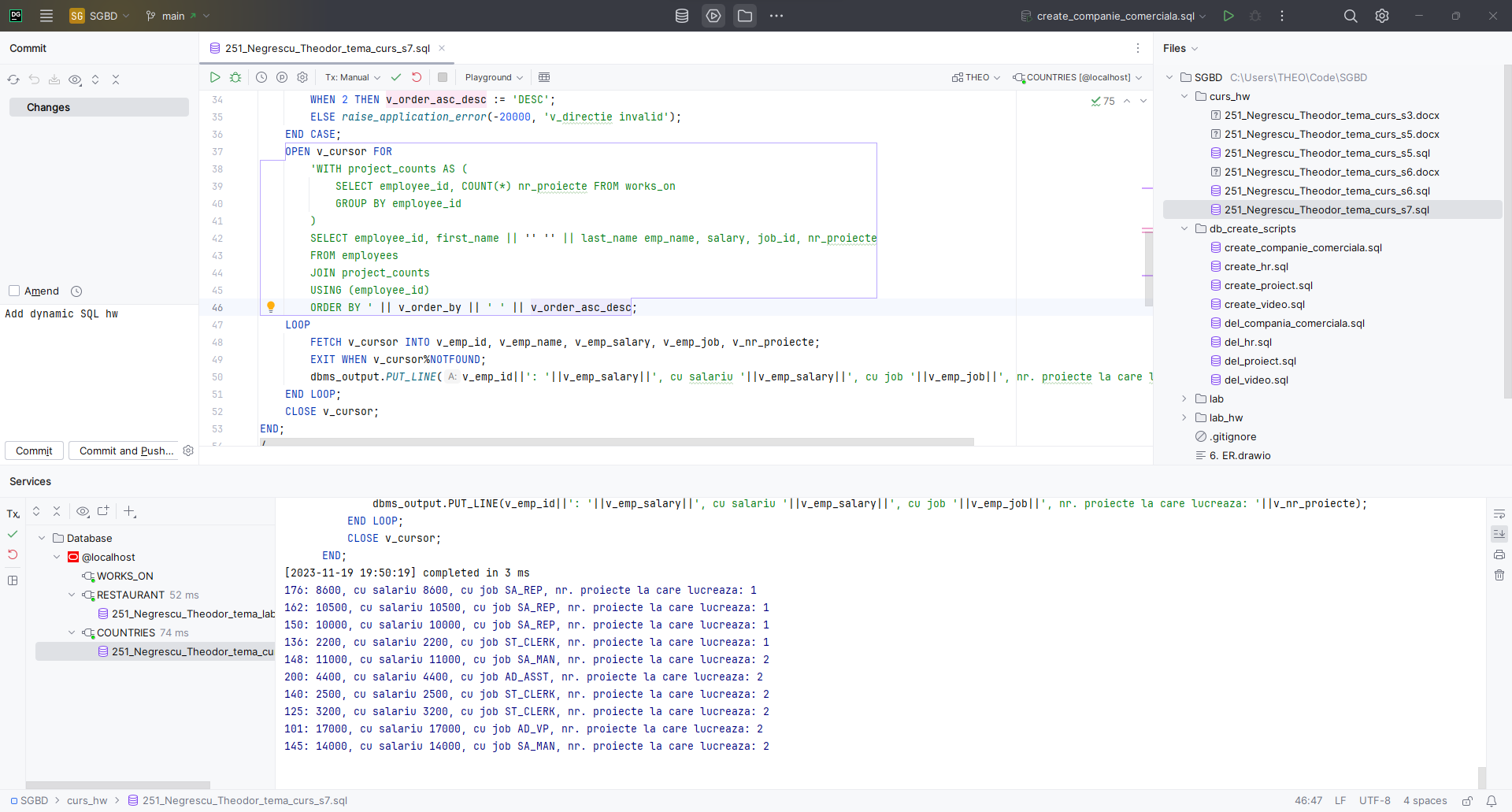
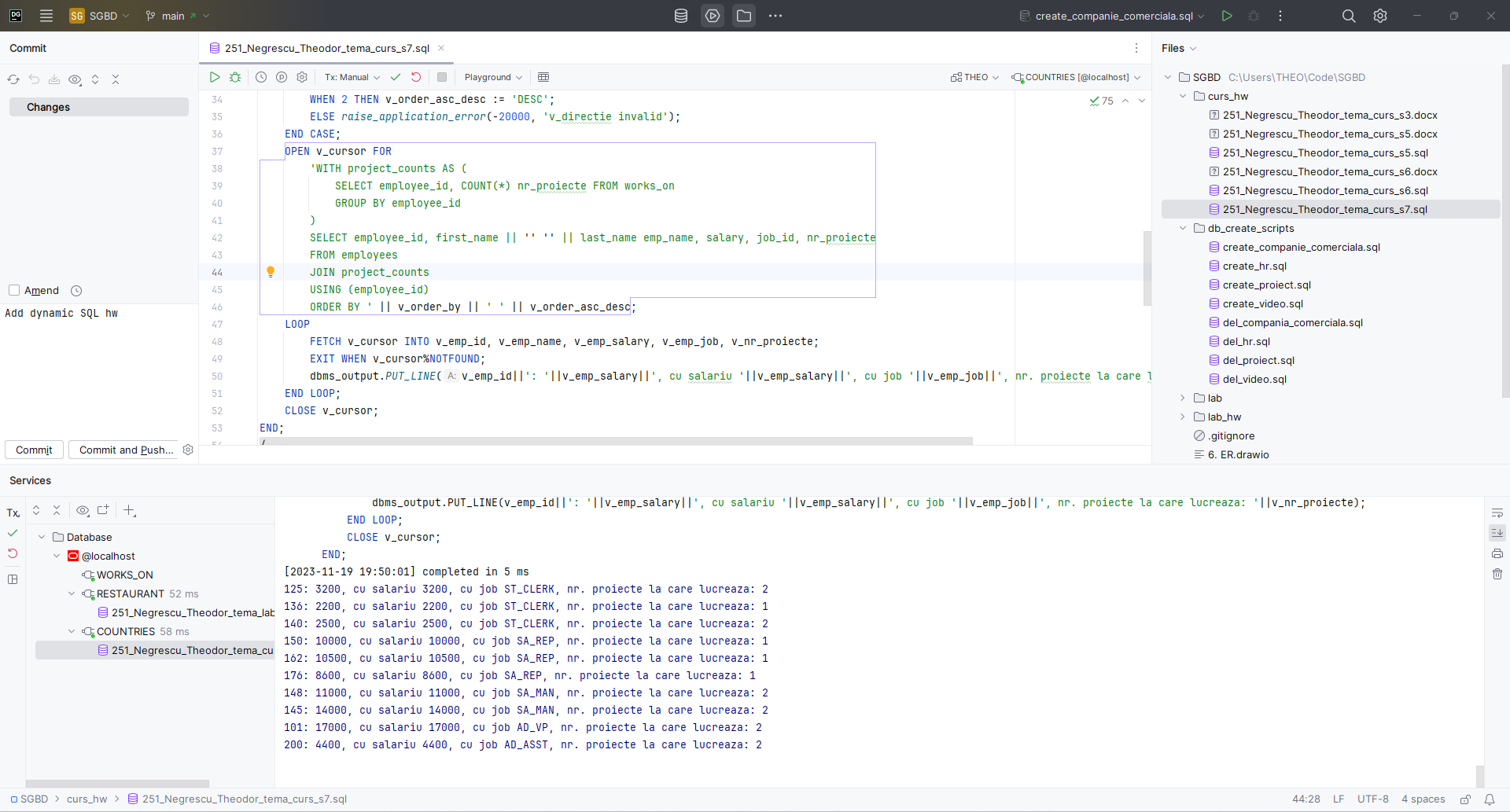
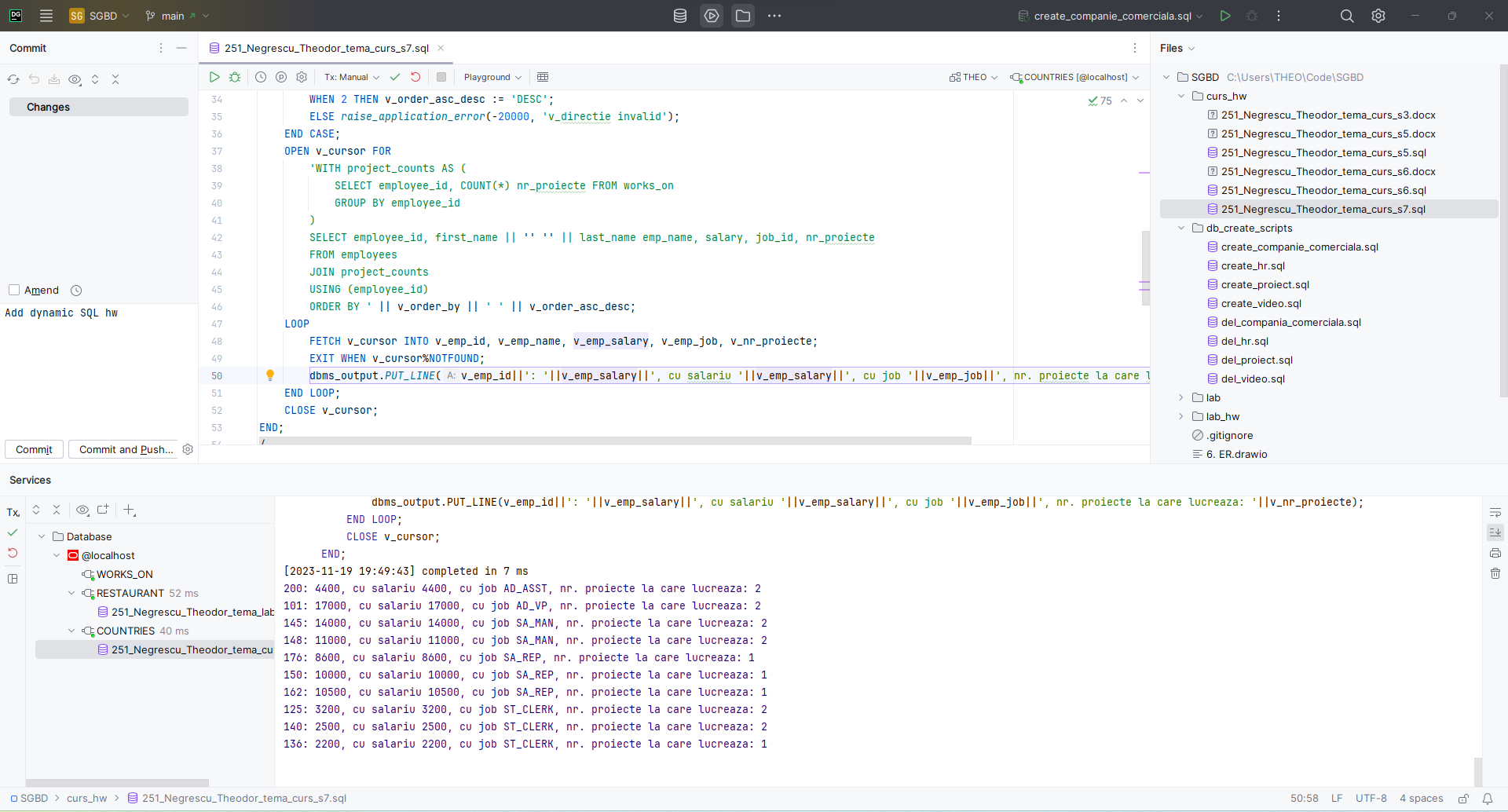
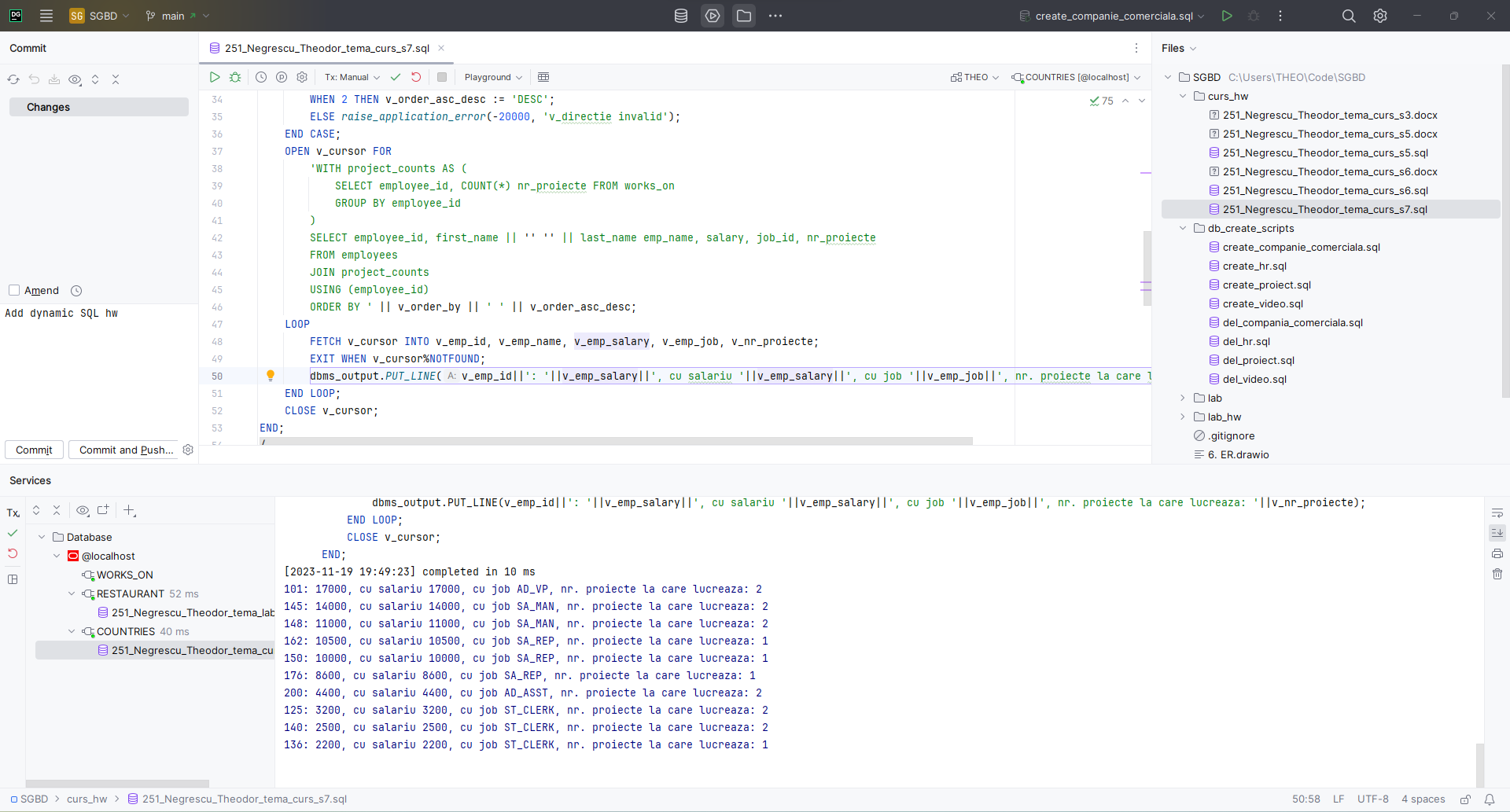
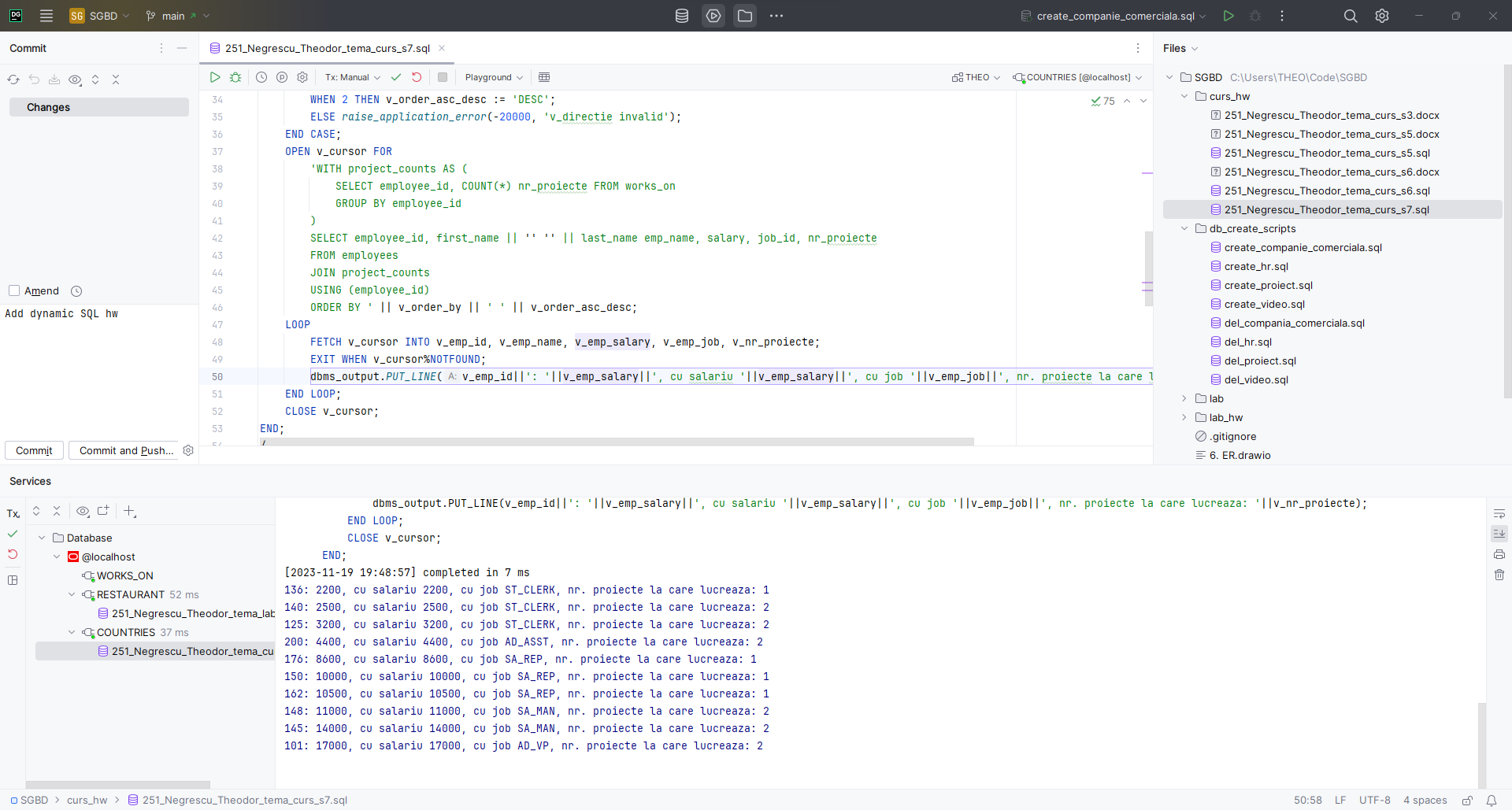
|  |
| --- |
| */\* 1. Enunțați o cerere în limbaj natural pe schema, care să implice în rezolvare utilizarea unui cursor dinamic. Scrieți un subprogram care utilizează acest cursor. Vor fi afișate informațiile din cel puțin două coloane returnate de cursor. Tratați erorile care pot să apară la apelare. Testați.  Sa se afiseze angajatii din compania, sortati fie dupa salariu, fie dupa job, fie dupa numarul de proiecte la care lucreaza. Userul decide daca sortarea e crescatoare sau descrescatore. \*/* DECLARE  v\_cursor SYS\_REFCURSOR;  v\_alegere BINARY\_INTEGER := &p\_alegere;  */\* 1 = salariu, 2 = job, 3 = proiecte \*/* v\_directie BINARY\_INTEGER := &p\_directie;  */\* 1 = ASC, 2 = DESC \*/* v\_emp\_id EMPLOYEES.employee\_id%TYPE;  v\_emp\_name VARCHAR2(100);  v\_emp\_salary EMPLOYEES.salary%TYPE;  v\_emp\_job EMPLOYEES.job\_id%TYPE;  v\_nr\_proiecte BINARY\_INTEGER;  v\_order\_by VARCHAR(20);  v\_order\_asc\_desc VARCHAR(5);  input\_invalid\_pt\_sql\_dinamic EXCEPTION;  PRAGMA EXCEPTION\_INIT (input\_invalid\_pt\_sql\_dinamic, -20000); BEGIN  CASE v\_alegere  WHEN 1 THEN v\_order\_by := 'salary';  WHEN 2 THEN v\_order\_by := 'job\_id';  WHEN 3 THEN v\_order\_by := 'nr\_proiecte';  ELSE *raise\_application\_error*(-20000, 'v\_alegere invalid');  END CASE;  CASE v\_directie  WHEN 1 THEN v\_order\_asc\_desc := 'ASC';  WHEN 2 THEN v\_order\_asc\_desc := 'DESC';  ELSE *raise\_application\_error*(-20000, 'v\_directie invalid');  END CASE;  OPEN v\_cursor FOR  'WITH project\_counts AS (  SELECT employee\_id, COUNT(\*) nr\_proiecte FROM works\_on  GROUP BY employee\_id  )  SELECT employee\_id, first\_name || '' '' || last\_name emp\_name, salary, job\_id, nr\_proiecte  FROM employees  JOIN project\_counts  USING (employee\_id)  ORDER BY ' || v\_order\_by || ' ' || v\_order\_asc\_desc;  LOOP  FETCH v\_cursor INTO v\_emp\_id, v\_emp\_name, v\_emp\_salary, v\_emp\_job, v\_nr\_proiecte;  EXIT WHEN v\_cursor%NOTFOUND;  dbms\_output.*PUT\_LINE*(v\_emp\_id||': '||v\_emp\_salary||', cu salariu '||v\_emp\_salary||', cu job '||v\_emp\_job||', nr. proiecte la care lucreaza: '||v\_nr\_proiecte);  END LOOP;  CLOSE v\_cursor; END; /  */\* 2. Definiți un bloc PL/SQL în care procedura proc\_ex2 (Exemplul 6.2 din SGBD6 - PLSQL Subprograme.pdf) este apelată pentru fiecare produs din categoria 'it' (nivel 1). Prețul acestor produse va fi micșorat cu 5%. \*/* CREATE PROCEDURE *proc\_ex1* (v\_id produse.id\_produs%TYPE, v\_procent NUMBER) AS BEGIN  UPDATE produse  SET pret\_unitar = pret\_unitar + pret\_unitar \* v\_procent  WHERE id\_produs = v\_id; EXCEPTION  WHEN NO\_DATA\_FOUND THEN  *RAISE\_APPLICATION\_ERROR* (-20000,'Nu exista produsul'); END;  BEGIN  FOR produs IN (  SELECT id\_produs  FROM PRODUSE  WHERE id\_categorie = 1  )  LOOP  DBMS\_OUTPUT.*PUT\_LINE*('Scădere preț - produs '||produs.id\_produs);  *proc\_ex1*(produs.id\_produs, -0.05);  END LOOP; END;  ROLLBACK;  DROP PROCEDURE *proc\_ex1*; |



P.S. Tema de laborator de saptamana 7 a fost trimisă înainte de a fi postate regulile de redactare, deci nu are pt. setat la 11 si nu are celula de tabel.