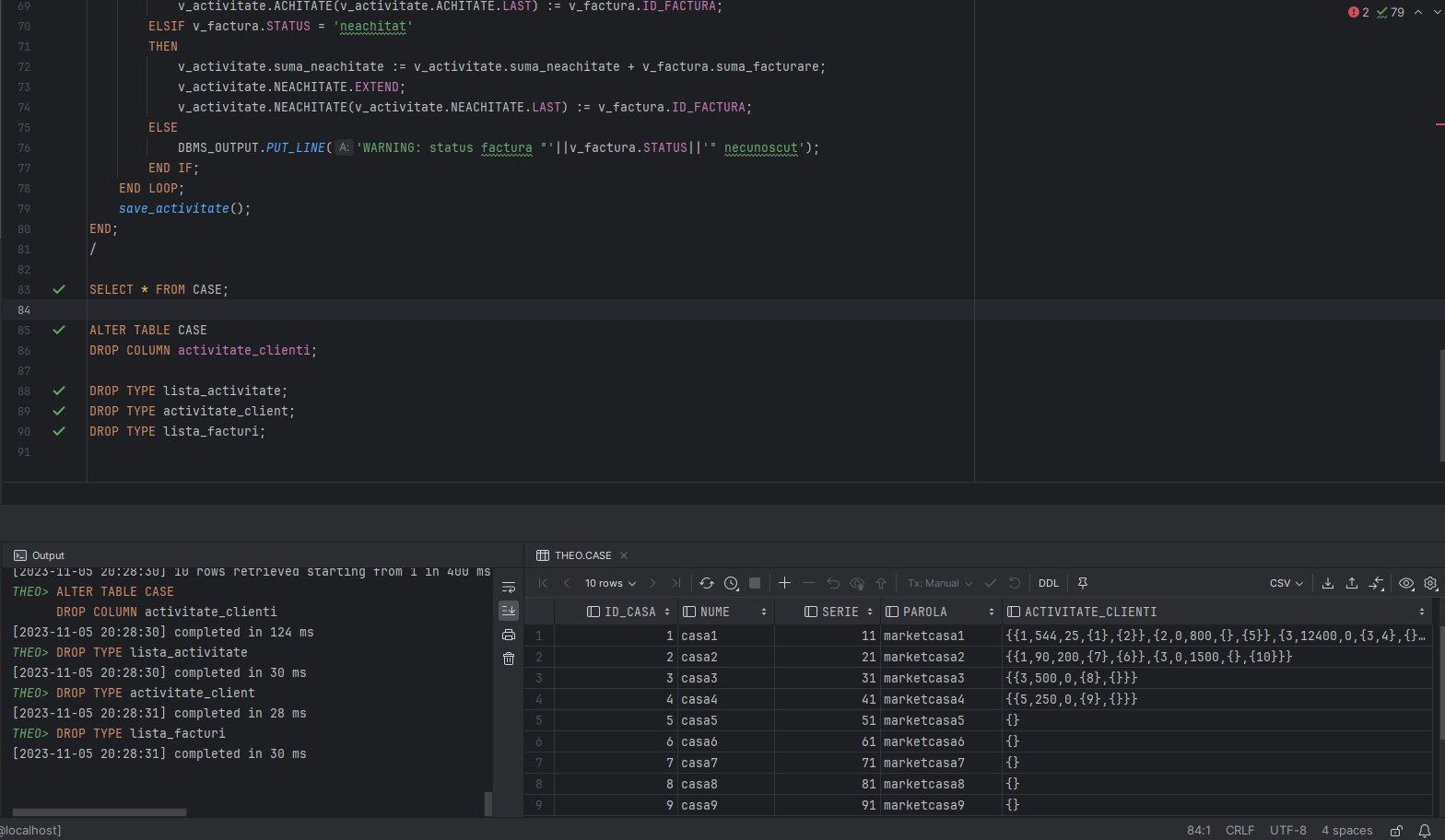
***-- Pentru fiecare casă, să se afle, pentru fiecare client care a folosit-o,  
-- totalul sumelor din facturi achitate și neachitate, și id-urile acelor facturi.  
  
-- Notă: această operație este o denormalizare a bazei de date.  
-- Acest lucru este inevitabil, deoarece cerința specifică să se actualizeze  
-- noua coloana cu informații din schema, deci este calculată  
-- din date deja existente in tabel, deci introduce redundanță.  
  
-- O listă de ID-uri de facturi. Keyword-ul %TYPE nu pare să meargă aici.*CREATE TYPE lista\_facturi IS TABLE OF NUMBER(20);  
/  
*-- Activitatea unui client la o anumită casă - facturile achitate și neachitate, si sumele.*CREATE TYPE activitate\_client IS OBJECT (id\_client NUMBER(10), suma\_achitate NUMBER(10), suma\_neachitate NUMBER(10), achitate lista\_facturi, neachitate lista\_facturi);  
/  
*-- Lista activităților tuturor clienților la o casă.*CREATE TYPE lista\_activitate IS TABLE OF activitate\_client;  
  
ALTER TABLE CASE  
ADD activitate\_clienti lista\_activitate  
DEFAULT lista\_activitate()  
NESTED TABLE activitate\_clienti STORE AS NESTED\_case\_lista\_activitate  
(NESTED TABLE achitate STORE AS NESTED\_case\_lista\_activitate\_NESTED\_achitate,  
 NESTED TABLE neachitate STORE AS NESTED\_case\_lista\_activitate\_NESTED\_neachitate);  
  
DECLARE  
 SUBTYPE st\_id\_client IS FACTURI.id\_client%TYPE;  
 SUBTYPE st\_id\_casa IS FACTURI.id\_casa%TYPE;  
 TYPE info\_factura IS RECORD ( id\_factura FACTURI.id\_factura%TYPE,  
 id\_casa st\_id\_casa,  
 id\_client st\_id\_client,  
 status FACTURI.status%TYPE,  
 suma\_facturare FACTURI\_CONTIN\_PRODUSE.pret\_facturare%TYPE);  
 v\_factura info\_factura;  
 v\_casa\_curenta st\_id\_casa := NULL;  
 v\_activitate activitate\_client := NULL;  
 PROCEDURE *init\_activitate*(casa IN st\_id\_casa, client IN st\_id\_client) IS  
 BEGIN  
 v\_casa\_curenta := casa;  
 v\_activitate := activitate\_client(client, 0, 0, lista\_facturi(), lista\_facturi());  
 END;  
 PROCEDURE *save\_activitate* IS  
 BEGIN  
 IF v\_activitate IS NULL  
 THEN  
 RETURN;  
 END IF;  
 INSERT INTO TABLE(SELECT ACTIVITATE\_CLIENTI FROM CASE WHERE ID\_CASA = v\_casa\_curenta)  
 VALUES (v\_activitate);  
 v\_activitate := NULL;  
 v\_casa\_curenta := NULL;  
 END;  
BEGIN  
 FOR v\_factura IN (SELECT id\_factura, id\_casa, id\_client, status, *SUM*(pret\_facturare) suma\_facturare  
 FROM FACTURI  
 JOIN FACTURI\_CONTIN\_PRODUSE  
 USING (id\_factura)  
 GROUP BY (id\_factura, id\_casa, id\_client, status)  
 ORDER BY id\_casa, id\_client, id\_factura)  
 LOOP  
 IF v\_activitate IS NULL OR v\_activitate.ID\_CLIENT != v\_factura.ID\_CLIENT OR v\_casa\_curenta != v\_factura.ID\_CASA  
 THEN  
 *save\_activitate*();  
 *init\_activitate*(v\_factura.ID\_CASA, v\_factura.ID\_CLIENT);  
 END IF;  
 IF v\_factura.STATUS = 'achitat'  
 THEN  
 v\_activitate.suma\_achitate := v\_activitate.suma\_achitate + v\_factura.suma\_facturare;  
 v\_activitate.ACHITATE.EXTEND;  
 v\_activitate.ACHITATE(v\_activitate.ACHITATE.LAST) := v\_factura.ID\_FACTURA;  
 ELSIF v\_factura.STATUS = 'neachitat'  
 THEN  
 v\_activitate.suma\_neachitate := v\_activitate.suma\_neachitate + v\_factura.suma\_facturare;  
 v\_activitate.NEACHITATE.EXTEND;  
 v\_activitate.NEACHITATE(v\_activitate.NEACHITATE.LAST) := v\_factura.ID\_FACTURA;  
 ELSE  
 DBMS\_OUTPUT.*PUT\_LINE*('WARNING: status factura "'||v\_factura.STATUS||'" necunoscut');  
 END IF;  
 END LOOP;  
 *save\_activitate*();  
END;  
/  
  
SELECT *\** FROM CASE;  
  
ALTER TABLE CASE  
DROP COLUMN activitate\_clienti;  
  
DROP TYPE lista\_activitate;  
DROP TYPE activitate\_client;  
DROP TYPE lista\_facturi;**

Datagrip afișeaza o eroare de sintaxă, dar asta e fiindcă nu înțelege sintaxa NESTED TABLE…STORE AS. Codul a rulat fără probleme.