**Temă**

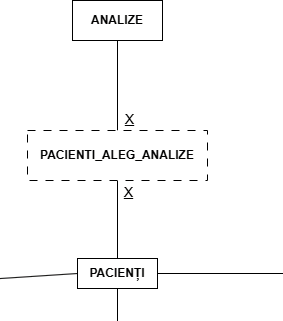
*Cerința*: E7. Adaptați cerința exercițiului 5 pentru diagrama proiectului prezentată la materia Baze de Date din anul I. Rezolvați acest exercițiu în PL/SQL, folosind baza de date proprie.

*Exercițiul 5*: Să se creeze un bloc PL/SQL prin care se afişează numărul de salariaţi care au venitul anual mai mare decât o valoare dată. Să se trateze cazul în care niciun salariat nu îndeplineşte această condiţie (excepţii externe).

*Rezolvare*: Fiind dat un număr de analize, afișați numărul de pacienți care au efectuat un număr de analize mai mare sau egal decât valoarea dată. Pentru fiecare pacient care respectă condiția, se va afișa numele, prenumele și numărul de analize.

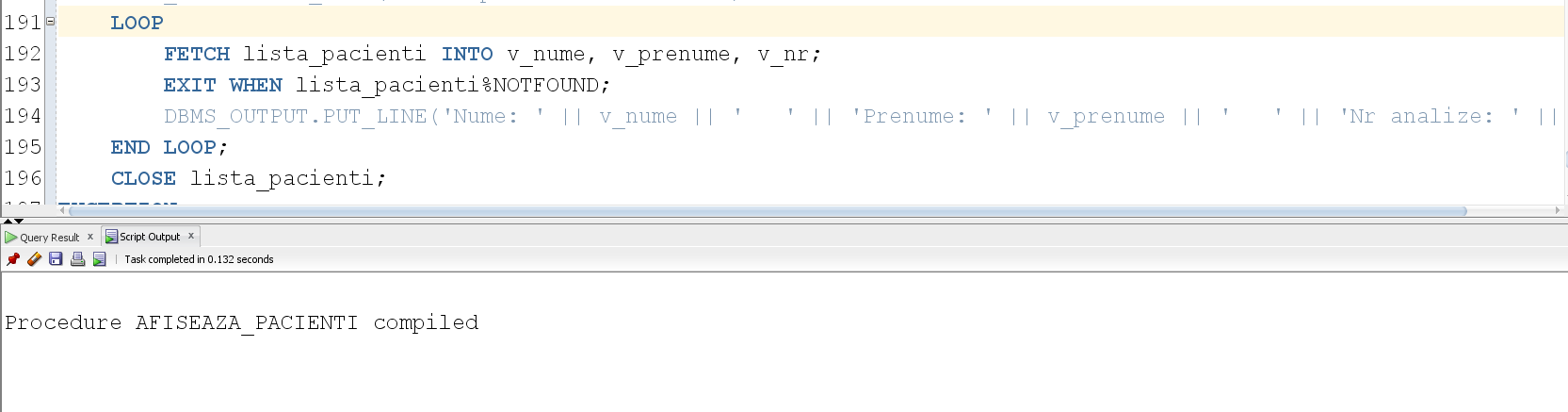
*Se va folosi schema CLINICĂ MEDICALĂ utilizată la proiect. Schema este deja implementată pe grupa251, dar mai jos se regăsește implementarea și popularea tabelelor care vor fi folosite.*

*Diagrama conceptuală*

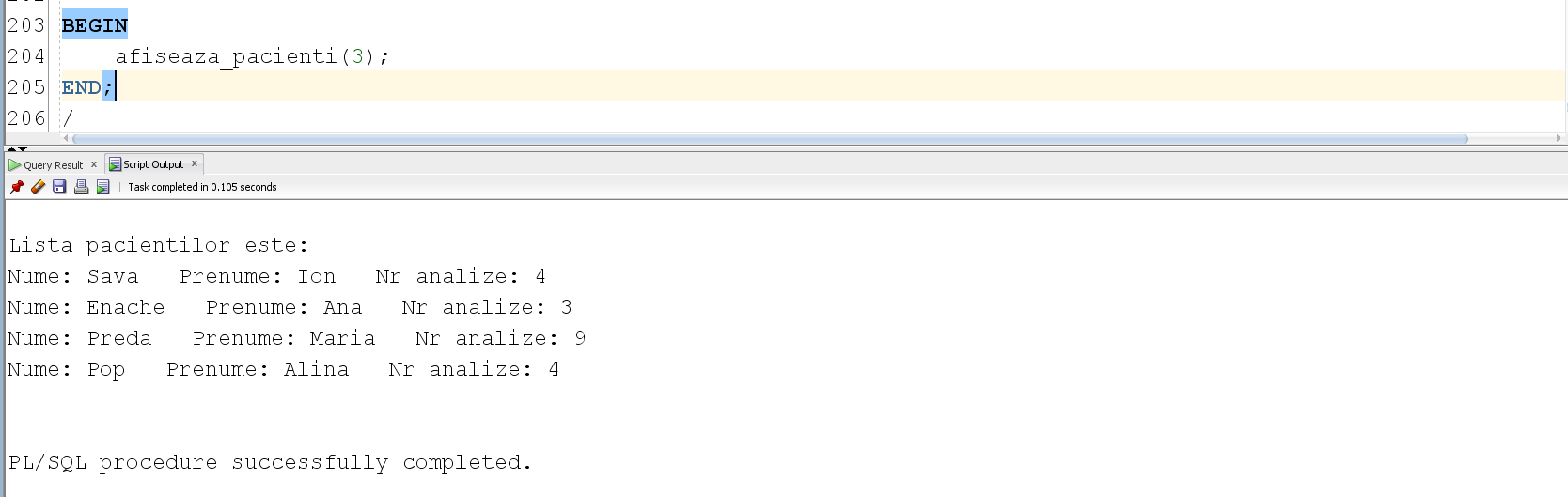


| CREATE TABLE analize\_ras  (id NUMBER PRIMARY KEY,  denumire VARCHAR2(50) NOT NULL,  pret NUMBER NOT NULL,  timp\_prelucrare NUMBER);  CREATE TABLE pacienti\_ras  (id NUMBER PRIMARY KEY,  cnp VARCHAR2(13) UNIQUE NOT NULL,  nume VARCHAR2(30),  prenume VARCHAR2(30),  data\_nasterii DATE,  telefon VARCHAR2(15));  CREATE TABLE pacienti\_aleg\_analize\_ras  (id\_pacient CONSTRAINT fk\_pacientanalize\_pacient REFERENCES pacienti\_ras(id) ON DELETE CASCADE,  id\_analiza CONSTRAINT fk\_pacientanalize\_analiza REFERENCES analize\_ras(id) ON DELETE CASCADE,  data DATE DEFAULT SYSDATE,  CONSTRAINT pk\_pacienti\_analize PRIMARY KEY (id\_pacient, id\_analiza, data)  );  INSERT INTO analize\_ras  VALUES (1, 'Vitamina B12', 73.0, 12);  INSERT INTO analize\_ras  VALUES (2, 'Homocisteina', 199.99, 14);  INSERT INTO analize\_ras  VALUES (3, 'Osteocalcin', 75.0, 17);  INSERT INTO analize\_ras  VALUES (4, 'Imunoglobulina A', 97.0, 3);  INSERT INTO analize\_ras  VALUES (5, 'Helicobacter pylori Ac', 100.0, 1);  INSERT INTO analize\_ras  VALUES (6, 'Numaratoare reticulocite', 129.99, 1);  INSERT INTO analize\_ras  VALUES (7, 'Determinare grup sangvin', 30.0, 1);  INSERT INTO analize\_ras  VALUES (8, 'Glicemie', 85, 2);  INSERT INTO pacienti\_ras  VALUES (1, '5241226155713', 'Sava', 'Ion', TO\_DATE('2000/07/02', 'yyyy/mm/dd'), '0722568199');  INSERT INTO pacienti\_ras  VALUES (2, '5241226152670', 'Preda', 'Maria', TO\_DATE('2012/04/22', 'yyyy/mm/dd'), '0750232000');  INSERT INTO pacienti\_ras  VALUES (3, '5170326156302', 'Enache', 'Ana', TO\_DATE('1987/11/24', 'yyyy/mm/dd'), '0721886095');  INSERT INTO pacienti\_ras  VALUES (4, '5170326155650', 'Pop', 'Alina', TO\_DATE('1995/04/19', 'yyyy/mm/dd'), '0723355309');  INSERT INTO pacienti\_ras  VALUES (5, '5170326157865', 'Pop', 'Alex', TO\_DATE('2000/08/28', 'yyyy/mm/dd'), '0745337969');  INSERT INTO pacienti\_ras  VALUES (6, '5170326156322', 'Pop', 'Alex', TO\_DATE('1997/12/14', 'yyyy/mm/dd'), '0753662812');  INSERT INTO pacienti\_ras  VALUES (7, '5170826156322', 'Popa', 'Miruna', TO\_DATE('1999/09/23', 'yyyy/mm/dd'), '0758962812');  INSERT INTO pacienti\_aleg\_analize\_ras  VALUES (1, 1, TO\_DATE('2024/12/23', 'yyyy/mm/dd'));  INSERT INTO pacienti\_aleg\_analize\_ras  VALUES (1, 3, TO\_DATE('2024/12/23', 'yyyy/mm/dd'));  INSERT INTO pacienti\_aleg\_analize\_ras  VALUES (1, 5, TO\_DATE('2023/08/20', 'yyyy/mm/dd'));  INSERT INTO pacienti\_aleg\_analize\_ras  VALUES (1, 7, TO\_DATE('2022/10/29', 'yyyy/mm/dd'));  INSERT INTO pacienti\_aleg\_analize\_ras  VALUES (2, 2, TO\_DATE('2024/10/14', 'yyyy/mm/dd'));  INSERT INTO pacienti\_aleg\_analize\_ras  VALUES (2, 4, TO\_DATE('2024/10/14', 'yyyy/mm/dd'));  INSERT INTO pacienti\_aleg\_analize\_ras  VALUES (2, 6, TO\_DATE('2024/10/14', 'yyyy/mm/dd'));  INSERT INTO pacienti\_aleg\_analize\_ras  VALUES (2, 2, TO\_DATE('2024/12/14', 'yyyy/mm/dd'));  INSERT INTO pacienti\_aleg\_analize\_ras  VALUES (2, 4, TO\_DATE('2024/12/14', 'yyyy/mm/dd'));  INSERT INTO pacienti\_aleg\_analize\_ras  VALUES (2, 6, TO\_DATE('2024/12/14', 'yyyy/mm/dd'));  INSERT INTO pacienti\_aleg\_analize\_ras  VALUES (2, 2, TO\_DATE('2023/03/09', 'yyyy/mm/dd'));  INSERT INTO pacienti\_aleg\_analize\_ras  VALUES (2, 4, TO\_DATE('2023/02/08', 'yyyy/mm/dd'));  INSERT INTO pacienti\_aleg\_analize\_ras  VALUES (2, 6, TO\_DATE('2023/04/12', 'yyyy/mm/dd'));  INSERT INTO pacienti\_aleg\_analize\_ras  VALUES (3, 7, TO\_DATE('2024/11/04', 'yyyy/mm/dd'));  INSERT INTO pacienti\_aleg\_analize\_ras  VALUES (3, 3, TO\_DATE('2024/11/04', 'yyyy/mm/dd'));  INSERT INTO pacienti\_aleg\_analize\_ras  VALUES (3, 6, TO\_DATE('2024/08/12', 'yyyy/mm/dd'));  INSERT INTO pacienti\_aleg\_analize\_ras  VALUES (4, 1, TO\_DATE('2022/07/10', 'yyyy/mm/dd'));  INSERT INTO pacienti\_aleg\_analize\_ras  VALUES (4, 1, TO\_DATE('2024/03/22', 'yyyy/mm/dd'));  INSERT INTO pacienti\_aleg\_analize\_ras  VALUES (4, 6, TO\_DATE('2024/12/28', 'yyyy/mm/dd'));  INSERT INTO pacienti\_aleg\_analize\_ras  VALUES (4, 4, TO\_DATE('2024/08/07', 'yyyy/mm/dd')); |
| --- |

| SET SERVEROUTPUT ON;  SET VERIFY OFF;  CREATE OR REPLACE PROCEDURE afiseaza\_pacienti (nr\_analize NUMBER)  IS  niciun\_pacient EXCEPTION;  ct\_pacienti NUMBER;  TYPE tip\_cursor IS REF CURSOR;  lista\_pacienti tip\_cursor;  v\_nume pacienti\_ras.nume%TYPE;  v\_prenume pacienti\_ras.prenume%TYPE;  v\_nr NUMBER;  BEGIN  -- vedem cati pacienti indeplinesc conditia  SELECT COUNT(\*)  INTO ct\_pacienti  FROM (  SELECT id\_pacient  FROM pacienti\_aleg\_analize\_ras  GROUP BY id\_pacient  HAVING COUNT(id\_analiza) >= nr\_analize  );    -- verif sa vedem daca exista vreun pacient  IF ct\_pacienti = 0 THEN  RAISE niciun\_pacient;  END IF;    -- selectam pacientii care indeplinesc conditia  OPEN lista\_pacienti FOR  'SELECT p.nume, p.prenume, COUNT(asoc.id\_analiza)  FROM pacienti\_ras p JOIN pacienti\_aleg\_analize\_ras asoc ON (p.id = asoc.id\_pacient)  GROUP BY p.id, p.nume, p.prenume  HAVING COUNT(asoc.id\_analiza) >= :v'  USING nr\_analize;    -- afisam pacientii  DBMS\_OUTPUT.PUT\_LINE('Lista pacientilor este: ');  LOOP  FETCH lista\_pacienti INTO v\_nume, v\_prenume, v\_nr;  EXIT WHEN lista\_pacienti%NOTFOUND;  DBMS\_OUTPUT.PUT\_LINE('Nume: ' || v\_nume || ' ' || 'Prenume: ' || v\_prenume || ' ' || 'Nr analize: ' || v\_nr);  END LOOP;  CLOSE lista\_pacienti;  EXCEPTION  WHEN niciun\_pacient THEN  RAISE\_APPLICATION\_ERROR(-20001, 'Niciun pacient nu a efectuat minim ' || nr\_analize || ' analize');  END;  /  BEGIN  afiseaza\_pacienti(3);  END;  /  BEGIN  afiseaza\_pacienti(10);  END;  / |
| --- |



*Exemplul 1*



*Exemplul 2*

