E7. Adaptați cerințele exercițiilor 2 și 4 (folosind ca bază cerințele exercițiilor 1, respectiv 3) pentru diagrama proiectului prezentată la materia Baze de Date din anul I. Rezolvați aceste două exerciții în PL/SQL, folosind baza de date proprie.

Rezolvare:

Întrucât nu am avut proiect la materia Baze de Date, am creat o structură simplificată pentru a rezolva cerințele.

| CREATE TABLE facultate\_ras  ( id\_facultate NUMBER(4) CONSTRAINT id\_facultate\_ras NOT NULL,  an\_infiintare NUMBER(4),  nume VARCHAR2(50),  profil VARCHAR2(20),  CONSTRAINT pk\_facultate\_ras PRIMARY KEY (id\_facultate)  );    CREATE TABLE student\_ras  ( id\_student NUMBER(4) CONSTRAINT id\_student\_ras NOT NULL,  nume VARCHAR2(30),  prenume VARCHAR2(30),  email VARCHAR2(60),  CONSTRAINT pk\_student\_ras PRIMARY KEY (id\_student)  );  CREATE TABLE studiaza\_ras  (  id\_facultate NUMBER(4) CONSTRAINT id\_fac\_ras NOT NULL,  id\_student NUMBER(4) CONSTRAINT id\_stud\_ras NOT NULL,  an NUMBER(1),  ciclu VARCHAR2(30),  forma\_finantare VARCHAR2(30),  CONSTRAINT pk\_studiaza\_ras PRIMARY KEY (id\_facultate, id\_student),  CONSTRAINT fk\_id\_facultate FOREIGN KEY(id\_facultate) REFERENCES facultate\_ras(id\_facultate) ON DELETE CASCADE,  CONSTRAINT fk\_id\_student FOREIGN KEY(id\_student) REFERENCES student\_ras(id\_student) ON DELETE CASCADE  );  INSERT INTO student\_ras  VALUES ('300', 'Popescu', 'Marian', '[popescu.m@gmail.com](mailto:popescu.m@gmail.com)');  INSERT INTO student\_ras  VALUES (301, 'Popescu', 'George', 'popescu.g@gmail.com');  INSERT INTO student\_ras  VALUES ('200', 'Ailei', 'Costel', 'ailenei.c@gmail.com');  INSERT INTO student\_ras  VALUES ('400', 'Barbu', 'Miruna', 'barbu.m@gmail.com');  INSERT INTO student\_ras  VALUES ('512', 'Ali', 'Bogdan', 'ali.b@gmail.com');  INSERT INTO student\_ras  VALUES ('150', 'Marin', 'Mario', 'marin.m@gmail.com');  INSERT INTO facultate\_ras  VALUES ('1', '1990', 'FMI', 'teoretic');  INSERT INTO facultate\_ras  VALUES ('2', '1990', 'FSLS', 'teoretic');  INSERT INTO facultate\_ras  VALUES ('3', '1980', 'FSEGA', 'teoretic');  INSERT INTO facultate\_ras  VALUES ('4', '1985', 'FAA', 'teoretic');  INSERT INTO facultate\_ras  VALUES ('5', '1988', 'FASV', 'practic');  INSERT INTO studiaza\_ras  VALUES ('1', '300', '2', 'licenta', 'buget');  INSERT INTO studiaza\_ras  VALUES ('3', '300', '3', 'licenta', 'buget');  INSERT INTO studiaza\_ras  VALUES ('2', '300', '1', 'licenta', 'taxa');  INSERT INTO studiaza\_ras  VALUES ('3', '200', '1', 'master', 'taxa');  INSERT INTO studiaza\_ras  VALUES ('2', '200', '1', 'licenta', 'taxa');  INSERT INTO studiaza\_ras  VALUES ('1', '150', '2', 'licenta', 'taxa');  INSERT INTO studiaza\_ras  VALUES ('1', '400', '3', 'licenta', 'buget');  INSERT INTO studiaza\_ras  VALUES ('1', '512', '1', 'master', 'buget');  INSERT INTO studiaza\_ras  VALUES ('4', '512', '2', 'master', 'taxa');  INSERT INTO studiaza\_ras  VALUES ('5', '400', '1', 'licenta', 'taxa'); |
| --- |

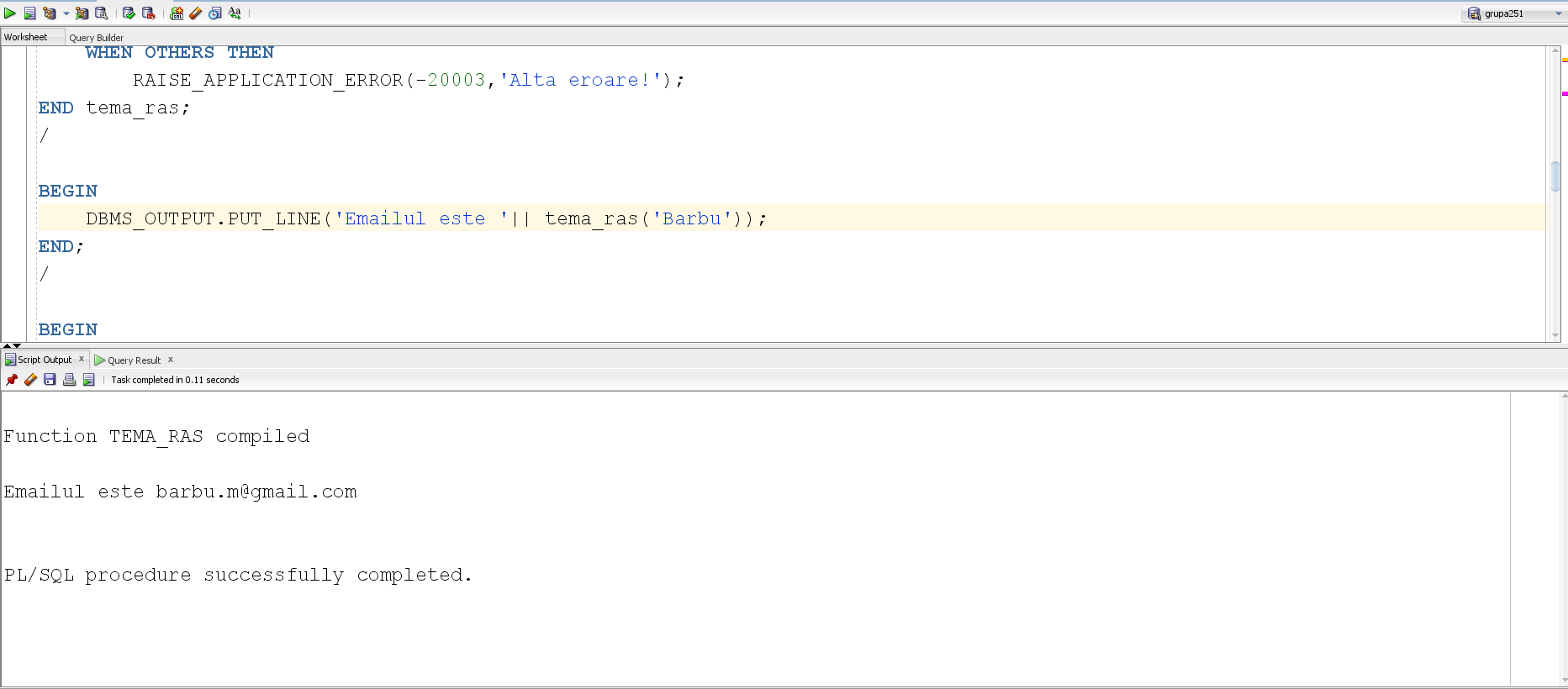
Exercițiul 2, adaptat:

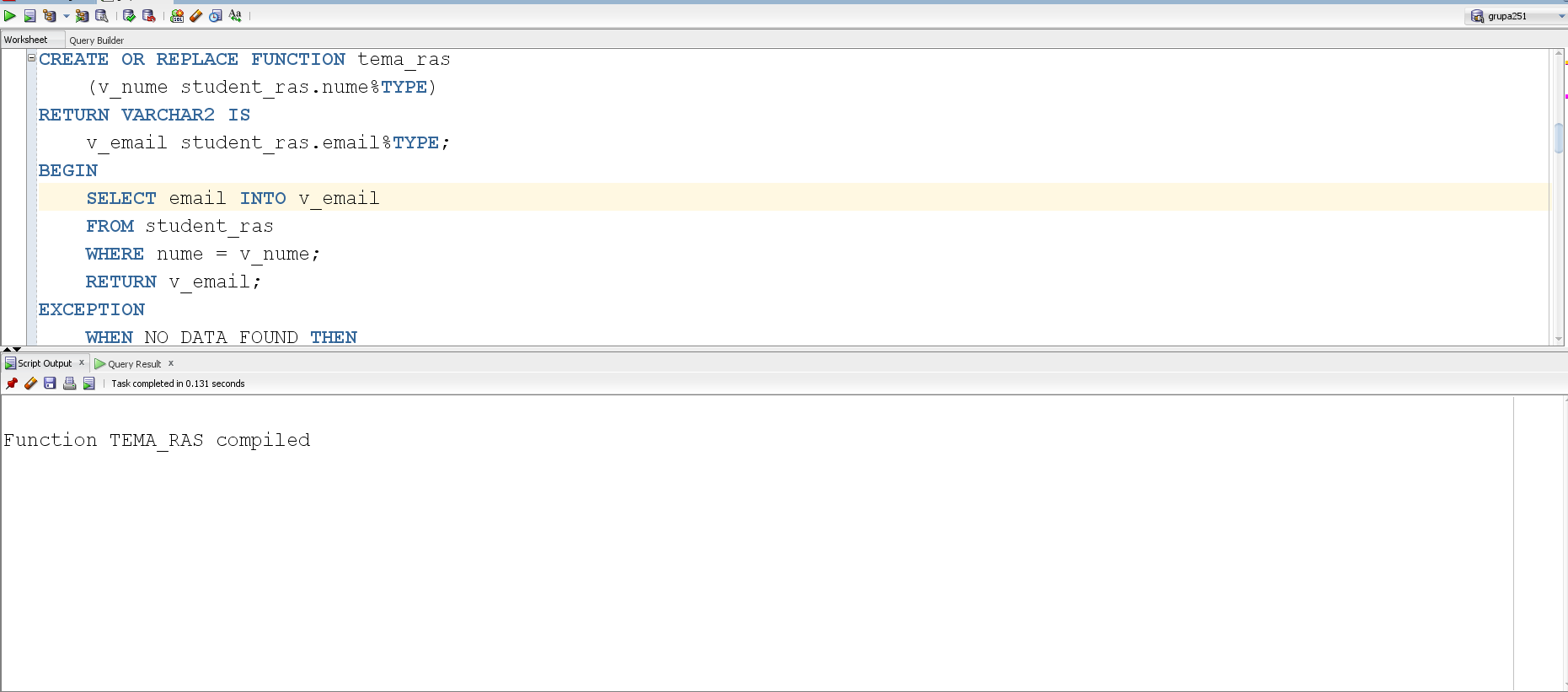
Definiți un subprogram prin care să obțineți adresa de mail a unui student al cărui

nume este specificat. Tratați toate excepțiile ce pot fi generate. Apelați subprogramul pentru următorii angajați: Barbu, Popescu, Sava.

Rezolvați problema folosind o funcție stocată.

| SET SERVEROUTPUT ON;  CREATE OR REPLACE FUNCTION tema\_ras  (v\_nume student\_ras.nume%TYPE)  RETURN VARCHAR2 IS  v\_email student\_ras.email%TYPE;  BEGIN  SELECT email INTO v\_email  FROM student\_ras  WHERE nume = v\_nume;  RETURN v\_email;  EXCEPTION  WHEN NO\_DATA\_FOUND THEN  RAISE\_APPLICATION\_ERROR(-20001,'Nu exista studenti cu numele dat');  WHEN TOO\_MANY\_ROWS THEN  RAISE\_APPLICATION\_ERROR(-20002,'Exista mai multi studenti cu numele dat');  WHEN OTHERS THEN  RAISE\_APPLICATION\_ERROR(-20003,'Alta eroare!');  END tema\_ras;  /  BEGIN  DBMS\_OUTPUT.PUT\_LINE('Emailul este '|| tema\_ras('Barbu'));  END;  /  BEGIN  DBMS\_OUTPUT.PUT\_LINE('Emailul este '|| tema\_ras('Popescu'));  END;  /  BEGIN  DBMS\_OUTPUT.PUT\_LINE('Emailul este '|| tema\_ras('Sava'));  END;  / |
| --- |







Exercițiul 4 adaptat:

Definiți un subprogram prin care să obțineți adresa de mail a unui student al cărui nume este specificat. Tratați toate excepțiile ce pot fi generate.

Apelați subprogramul pentru următorii angajați: Barbu, Popescu, Sava.

Rezolvați problema folosind o procedură stocată.



| CREATE OR REPLACE PROCEDURE ex4\_ras  (v\_nume student\_ras.nume%TYPE)  IS  v\_email student\_ras.email%TYPE;  BEGIN  SELECT email INTO v\_email  FROM student\_ras  WHERE nume = v\_nume;  DBMS\_OUTPUT.PUT\_LINE('Emailul este '|| v\_email);  EXCEPTION  WHEN NO\_DATA\_FOUND THEN  RAISE\_APPLICATION\_ERROR(-20001,'Nu exista studenti cu numele dat');  WHEN TOO\_MANY\_ROWS THEN  RAISE\_APPLICATION\_ERROR(-20002,'Exista mai multi studenti cu numele dat');  WHEN OTHERS THEN  RAISE\_APPLICATION\_ERROR(-20003,'Alta eroare!');  END ex4\_ras;  /  BEGIN  ex4\_ras('Barbu');  END;  /  BEGIN  ex4\_ras('Popescu');  END;  /  BEGIN  ex4\_ras('Sava');  END;  / |
| --- |

