*1. Identificați diagrama conceptuală definită în proiectul prezentat la materia Baze de Date din*

*anul I. Dacă nu aveți o astfel de diagramă, atunci definiți un model simplificat pentru acest*

*exercițiu.*

*a. adăugați ca poză diagrama conceptuală identificată;*

*b. adaptați cerințele exercițiilor 36 și 42 pentru diagrama conceptuală utilizată la*

*punctul a (formulați cerința în limbaj natural, apoi rezolvați cererea propusă în SQL).*

2. *Identificați în diagrama Entitate-Relație utilizată în proiectul prezentat la materia Baze de*

*Date din anul I, o relație din tip many-to-many. Dacă nu aveți o astfel de relație în acest*

*proiect, atunci definiți una folosind tema aleasă în acel proiect:  
  
a. descrieți entitățile și relația dintre acestea;*

*b. realizați diagrama Entitate-Relație doar pentru această parte din proiect;*

*c. descrieți modul de transformare al acestei relații în diagrama conceptuală,*

*precizând toate cheile primare, cheile externe și alte atribute esențiale;*

*d. realizați diagrama conceptuală doar pentru această parte din proiect;*

*e. pe baza diagramei conceptuale de la punctul d, definiți în SQL tabelele și toate*

*constrângerile necesare;*

*f. adaptați una dintre cerințele exercițiilor 25-34 pentru diagrama obținută la punctul d*

*(formulați cerința în limbaj natural, inserați 5-10 înregistrări în fiecare tabelă utilizată,*

*apoi rezolvați cererea propusă în SQL).*

***2***

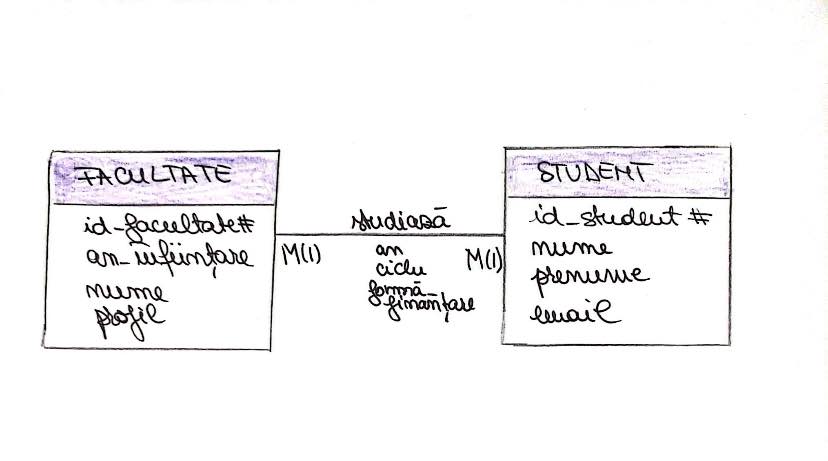
*a.*

**STUDENT** = persoană fizică înmatriculată în cadrul unei facultăți, ce frecventează cursurile acesteia; cheia primară a acestei entități este *id\_student*

**FACULTATE** = instituție de învățământ universitar în care se prestează servicii educaționale dedicate studenților. Cheia primară a acestei entități este *id\_facultate*

**STUDENT\_studiază\_FACULTATE** = relație dintre entitățile STUDENT și FACULTATE ce modelează studenții din cadrul facultăților. Cardinalitatea minimă este 1 : 1, întrucât o facultate are cel puțin un student pentru a putea exista și orice student trebuie să fie înmatriculat la cel puțin o facultate pentru a se regăsi în baza de date a universității. Cardinalitatea maximă a relației este m:m, deoarece o facultate poate avea mai mulți studenți, iar un student poate studia (fi înmatriculat) la mai multe facultăți din cadrul universității.

*b.*



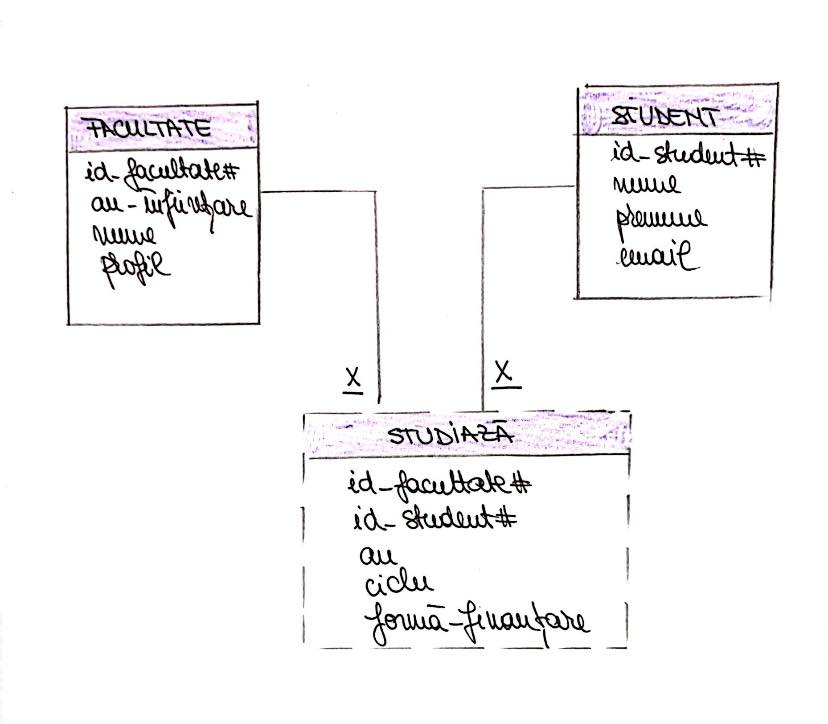
*c.*

În diagrama conceptuală, relația many to many se transformă în tabel asociativ, unde cheia primară va fi compusă din cheile celor 2 entități (id\_student#, id\_facultate#), cu semnificația că un anumit student studiază la o anumită facultate. id\_facultate# va fi foreign key in tabela STUDIAZA, deoarece acest atribut referențiază id\_facultate din tabela FACULTATE. La fel se întâmplă și în cazul lui id\_student# care este foreign key in STUDIAZA, referindu-se la id\_student din STUDENT.

Atributele relației devin atribute în tabelul asociativ, deci în final vom avea:

STUDIAZA = (id\_facultate#, id\_student#, an, ciclu, forma\_finantare)

*d.*



*e.*

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),

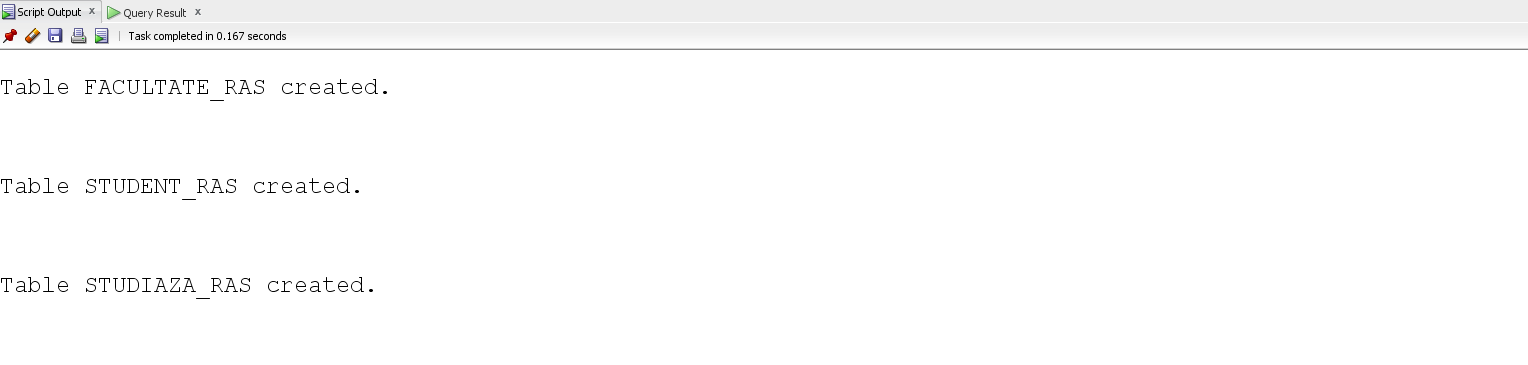
forma\_finantare VARCHAR2(30),

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

);



*f.*

INSERT INTO student\_ras

VALUES ('300', 'Popescu', 'Marian', 'popescu.m@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 ('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');

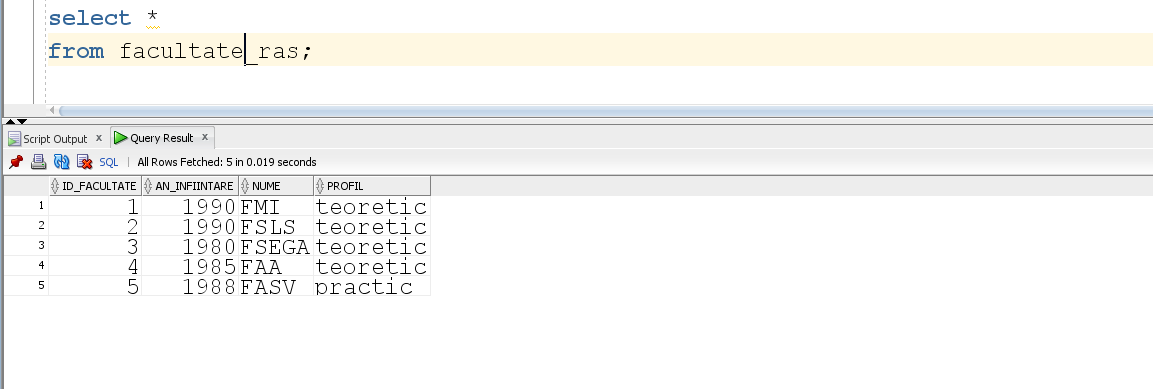
INSERT INTO studiaza\_ras

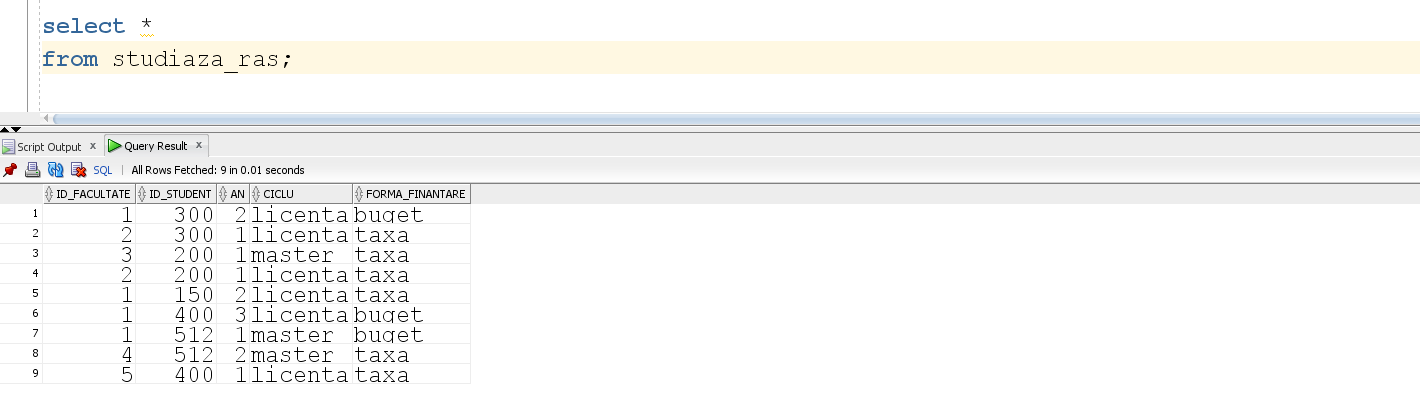
VALUES ('4', '512', '2', 'master', 'taxa');

INSERT INTO studiaza\_ras

VALUES ('5', '400', '1', 'licenta', 'taxa');







**Adaptăm cerința 26:** Să se afișeze numărul facultăților care au cel puțin 2 studenți.

WITH aux as (

select id\_facultate

from studiaza\_ras

group by id\_facultate

having count(id\_student) >= 2

)

select count(\*) nrfacultati

from aux;

