**PROIECT BAZE DE DATE**

**GESTIONAREA UNEI FACULTĂȚI**

**-Chițu Tudor 131 -**

Cuprins

[Cerința 1 2](#_Toc136036487)

[Cerința 2 2](#_Toc136036488)

[Cerința 3 2](#_Toc136036489)

[Cerința 4 3](#_Toc136036490)

[Cerința 5 4](#_Toc136036491)

[Cerința 6 6](#_Toc136036492)

[Cerința 7 7](#_Toc136036493)

[Cerința 8 8](#_Toc136036494)

[Cerința 9 9](#_Toc136036495)

[Cerința 10 9](#_Toc136036496)

[Cerința 11 10](#_Toc136036497)

[Cerința 12 13](#_Toc136036498)

[Cerința 13 16](#_Toc136036499)

[Cerința 15 18](#_Toc136036500)

[Cerința 16 19](#_Toc136036501)

[Cerința 17 22](#_Toc136036502)

[a) 22](#_Toc136036503)

[b) 22](#_Toc136036504)

[Cod SQL 23](#_Toc136036505)

# Cerința 1

**Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.**

Gestionarea facultății urmărește situația studenților cu privire la grupa din care fac parte, anul de studiu, cnp-ul, numele și prenumele. De asemenea urmărește repartizarea acestora la taxa sau buget, dar și cei care sunt bursieri. Unii dintre ei lucrează și la o lucrare de licență, având un profesor îndrumător. Cursurile la facultate sunt diverse, specifice specializărilor și ținute de un anumit profesor.. Facultatea are Decanat, format din profesori. Facultatea are parteneriate Erasmus în diferite țări.

# Cerința 2

**Prezentarea constrângerilor(restricții, reguli) impuse asupra modelului.**

* Bursele sunt cuprinse între 500 și 1500 de lei
* Un student poate fi ori la buget, ori la taxă
* Grupa în care poate fi un student este un număr între 100 și 500
* Specializările sunt:Informatică, Matematică, Matematică-Informatică, Matematică-Aplicată, CTI
* Licența trebuie să fie susținută de un fix un student, cu un singur profesor îndrumător și o singură arie de studiu. După absolvirea studentului, lucrarea sa de licență păstrează doar numele și id-ul acesteia
* Decanatul actual are minim un membru, iar baza de date reține și cronologia decanatului
* Fiecare curs are un doar profesor
* Numele lucrării de licență este unic

# Cerința 3

**Descrierea entităților, incluzând precizarea cheii primare.**

|  |  |  |
| --- | --- | --- |
| **ENTITATE** | **CHEIE PRIMARĂ** | **OBSERVAȚII** |
| Student | id\_student | Informații despre studenții facultății |
| Taxă | id\_student | Informații mai specifice pentru studenții de la taxă |
| Buget | id\_student | Informații mai specifice pentru studenții de la buget |
| Burse | id\_bursă | Informații despre bursele facultății |
| Licență | id\_licență | Informații despre lucrările de licență ale studenților |
| Profesor | id\_profesor | Informații despre profesorii facultății |
| Curs | id\_curs | Informații despre cursurile facultății |
| Specializare | id\_specializare | Informații despre specializările facultății |
| Decanat | id\_decanat | Informații despre decanatul facultății și compoziția sa |
| Erasmus | id\_erasmus | Informații despre proiectele Erasmus ale facultății |
| Tari | id\_tari | Informații despre țările partenere facultății |

# Cerința 4

**Descrierea relațiilor, incluzând precizarea cardinalității acestora.**

|  |  |  |
| --- | --- | --- |
| **RELAȚIE** | **CARDINALITATE** | **OBSERVAȚII** |
| are | Student-Bursă one-to-many  Specializare-Licență one-to-many | Un student poate să aiba mai multe burse, dar o bursă e specifică unui singur student  O licență e specifică unei singure specializări, dar o specializare cuprinde mai multe licențe |
| susține | Student-Licentă one-to-one | O licență e susținută de un student și un student susține doar o licență |
| coordonează | Profesor-Licență one-to-many | Un profesor coordonează mai multe lucrări de licență și o licență e coordonată de un profesor |
| alcătuiește | Decanat-Profesor one-to-many | Un profesor poate face parte din alcătuirea unui singur decanat, dar un decanat poate fi alcătuit din mai mulți profesori |
| predă | Profesor-Specializare many-to-many | Mai mulți profesori pot preda la o specializare, dar și un profesor poate preda la mai multe specializări |
| face | Profesor-Curs one-to-many | Un professor face mai multe cursuri, dar un curs e făcut doar de un profesor |
| participă | Student-Curs many-to-many | Un student participă la mai multe cursuri și la un curs participă mai mulți studenți |
| aparține | Specializare-Curs many-to-many | O specializare are mai multe cursuri, dar un curs poate să fie la mai multe specializări |
| ISA | Student-Taxa one-to-one  Student-Buget one-to-one | Un student poate fi ori la taxă, ori la buget |
| Studii\_Strainatate | Tari-(Student, Erasmus) one-to many  Student-(Tari, Erasmus) one-to many  Erasmus-(Tari, Student) one-to many | Un student poate merge la mai multe proiecte Erasmus, iar un proiect poate fi în mai multe țări |

# Cerința 5

Descrierea atributelor, incluzând tipul de date și eventualeleconstrângeri, valori implicite, valori posibile ale atributelor.

Entitate: Student

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ATRIBUT | TIP | DIMENSIUNE  PRECIZIE | VALORI POSIBILE ȘI DEFAULT | OBSERVAȚII, OBLIGATORIU  OPȚIONAL |
| id\_student | number | 5,0 | - | NOT NULL, PK |
| Nume | varchar2 | 30 | - | NOT NULL |
| Prenume | varchar2 | 30 | - | NOT NULL |
| CNP | varchar2 | 16 | - | NOT NULL |
| An\_studiu | number | 1,0 | - | NOT NULL |
| Grupa | number | 3,0 | - | NOT NULL |
| Tip\_student | varchar2 | 20 | Bursă/Taxă | NOT NULL |

Entitate: Taxă

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ATRIBUT | TIP | DIMENSIUNE  PRECIZIE | VALORI POSIBILE ȘI DEFAULT | OBSERVAȚII, OBLIGATORIU  OPȚIONAL |
| id\_student | number | 5,0 | - | NOT NULL, PK, FK |
| Valoare\_taxa | number | 4,0 | 3000-5000, Default 4000 | NOT NULL |

Entitate: Buget

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ATRIBUT | TIP | DIMENSIUNE  PRECIZIE | VALORI POSIBILE ȘI DEFAULT | OBSERVAȚII, OBLIGATORIU  OPȚIONAL |
| id\_student | number | 5,0 | - | NOT NULL, PK, FK |
| Medie\_admitere | float | 2 | 0-10, Default 10 | NOT NULL |

Entitate: Bursa

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ATRIBUT | TIP | DIMENSIUNE  PRECIZIE | VALORI POSIBILE ȘI DEFAULT | OBSERVAȚII, OBLIGATORIU  OPȚIONAL |
| id\_bursa | number | 5,0 | - | NOT NULL, PK |
| Nume | varchar2 | 30 | - | NOT NULL |
| Sumă | number | 4,0 | 500-1500 | NOT NULL |
| id\_student | number | 5,0 | - | NOT NULL, FK |

Entitate: Licență

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ATRIBUT | TIP | DIMENSIUNE  PRECIZIE | VALORI POSIBILE ȘI DEFAULT | OBSERVAȚII, OBLIGATORIU  OPȚIONAL |
| id\_licenta | number | 5,0 | - | NOT NULL, PK |
| Nume | varchar2 | 30 | - | NOT NULL, UNIQUE |
| id\_student | number | 5,0 | - | FK |
| id\_profesor | number | 5,0 | - | FK |
| id\_specializare | number | 5,0 | - | FK |

Entitate: Profesor

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ATRIBUT | TIP | DIMENSIUNE  PRECIZIE | VALORI POSIBILE ȘI DEFAULT | OBSERVAȚII, OBLIGATORIU  OPȚIONAL |
| id\_profesor | number | 5,0 | - | NOT NULL, PK |
| Nume | varchar2 | 30 | - | NOT NULL |
| Prenume | varchar2 | 30 | - | NOT NULL |
| CNP | varchar2 | 16 | - | - |
| Vechime | number | 2,0 | - | - |
| id\_decanat | number | 5,0 | - | FK |

Entitate: Curs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ATRIBUT | TIP | DIMENSIUNE  PRECIZIE | VALORI POSIBILE ȘI DEFAULT | OBSERVAȚII, OBLIGATORIU  OPȚIONAL |
| id\_curs | number | 5,0 | - | NOT NULL, PK |
| Nume | varchar2 | 30 | - | NOT NULL |
| id\_profesor | number | 5,0 | - | NOT NULL, FK |

Entitate: Specializare

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ATRIBUT | TIP | DIMENSIUNE  PRECIZIE | VALORI POSIBILE ȘI DEFAULT | OBSERVAȚII, OBLIGATORIU  OPȚIONAL |
| id\_specializare | number | 5,0 | - | NOT NULL, PK |
| Nume | varchar2 | 30 | - | NOT NULL |
| Nume\_diplomă | varchar2 | 60 | - | NOT NULL |

Entitate: Decanat

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ATRIBUT | TIP | DIMENSIUNE  PRECIZIE | VALORI POSIBILE ȘI DEFAULT | OBSERVAȚII, OBLIGATORIU  OPȚIONAL |
| id\_decanat | number | 5,0 | - | NOT NULL, PK |
| data | date | - | - | NOT NULL |
| durata | number | 1,0 | Default 4, Exprimat în ani | NOT NULL |

Entitate: Erasmus

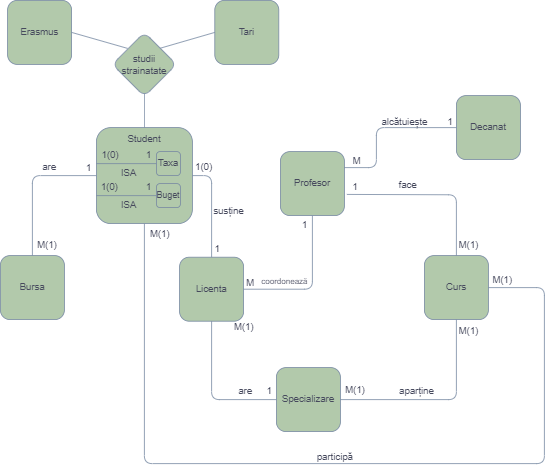
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ATRIBUT | TIP | DIMENSIUNE  PRECIZIE | VALORI POSIBILE ȘI DEFAULT | OBSERVAȚII, OBLIGATORIU  OPȚIONAL |
| id\_erasmus | number | 5,0 | - | NOT NULL, PK |
| nume | varchar2 | 30 | - | NOT NULL |
| durata | number | 5,0 | Exprimat în zile | NOT NULL |
| data\_început | date | 5,0 | - | NOT NULL |

Entitate: Tari

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ATRIBUT | TIP | DIMENSIUNE  PRECIZIE | VALORI POSIBILE ȘI DEFAULT | OBSERVAȚII, OBLIGATORIU  OPȚIONAL |
| id\_tara | number | 5,0 | - | NOT NULL, PK |
| nume | varchar2 | 30 | - | NOT NULL |
| moneda | varchar2 | 3 | - | NOT NULL |
| limba | varchar2 | 30 | - | NOT NULL |

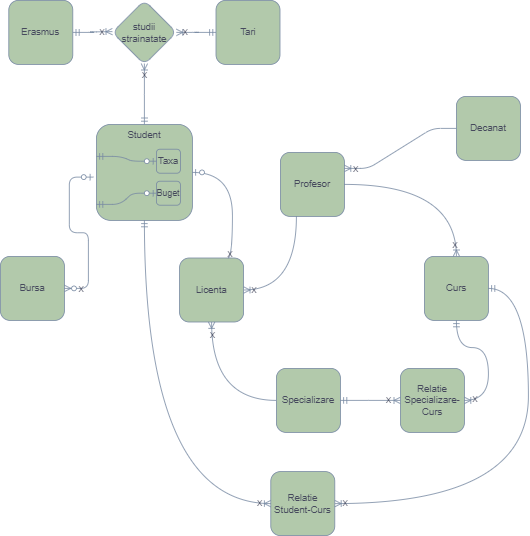
# Cerința 6

Realizarea diagrame entitate-relație corespunzătoare descrierii de la punctele 3-5.



# Cerința 7

Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6. Diagrama conceptuală obținută trebuie să conțină minimum 6 tabele (fără considerarea subentităților), dintre care cel puțin un tabel asociativ.



# Cerința 8

Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7.

Bursă(id\_bursa (PK), nume, suma, id\_student (FK))

Student(id\_ student(PK), nume, prenume, cnp, an\_studiu, grupa, tip\_student)

Taxă(id\_student (PK,FK), valoare\_taxa)

Buget(id\_student (PK,FK), medie\_admitere)

Licență(id\_licenta (PK), nume, id\_student (FK), id\_profesor (FK), arie\_subiect (FK))

Relație\_Student\_Curs(id\_relatie\_sc (PK), id\_student (FK), id\_curs (FK))

Curs(id\_curs (PK), nume, id\_specializare (FK), id\_profesor (FK))

Profesor(id\_profesor (PK), nume, prenume, cnp, id\_specializare (FK), vechime, id\_decanat (FK))

Decanat(id\_decanat (PK), data, durata)

Specializare(id\_specializare (PK), nume, nume\_diploma)

Relație\_Specializare\_Curs(id\_relatie\_spc (PK), id\_specializare (FK), id\_curs (FK))

Tari(id\_tara (PK), nume, moneda, limba)

Erasmus(id\_erasmus (PK), nume, durata, data\_inceput)

Studii\_strainatate(id\_relatie\_ss (PK), id\_tara (FK), id\_erasmus (FK), id\_student (FK))

# Cerința 9

Realizarea normalizării până la forma normală 3(FN1-FN3).

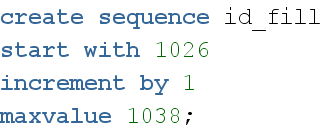
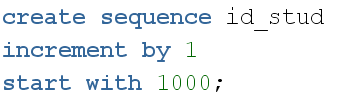
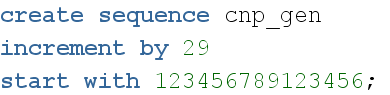
FN1: Pentru a avrea FN1 trebuie să ne asigurăm că atributele noastre sunt atomice. Exemple ce se pot întâlni pe modelul prezentat ar putea fi: student sau profesor. În ambele situații atributele pot fi descompuse în nume, prenume.

FN2: Pentru a avea o tabelă în FN2, trebuie ca aceasta să fie în FN1 și să nu aibă dependențe parțiale. Un astfel de caz ar putea fi întâlnit pe modelul ilustrat în cazul în care am fi avut tabela student-bursier, cu cheia primară compusă (id\_student, id\_bursă), iar atributele ar fi depins parțial de cheia primară. Suma ar fi depins strict de id\_bursă, iar datele personale ale studentului de id\_student. Astfel formăm cele două tabele din modelul prezentat.

FN3: Ca o tabelă să fie în FN3, aceasta trebuie să fie în FN2 și să nu aiba dependențe tranzitive. Pentru modelul prezentat o astfel de situație ar fi putut exista în cazul în care în tabela Bursă stocam și informații precum concurul pentru care se acordă bursa, aria de studiu a concursului etc. În acest caz am fi creat tabela Concurs(id\_concurs (PK), nume\_concurs, arie\_studiu), iar tabela bursă devenea Bursă(id\_bursa (PK), nume, suma, id\_student (FK), id\_concurs (FK))

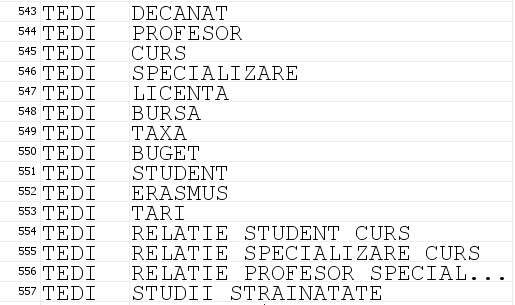
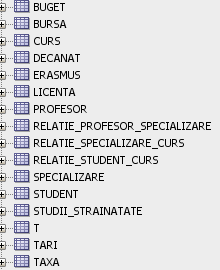
# Cerința 10

Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).

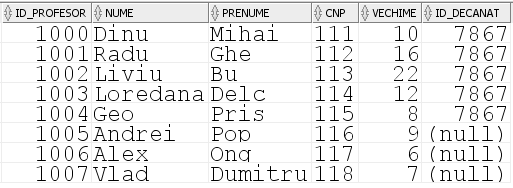


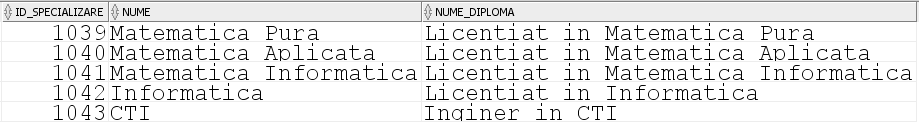
# Cerința 11

Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări întabelele asociative).Crearea tabelelor:

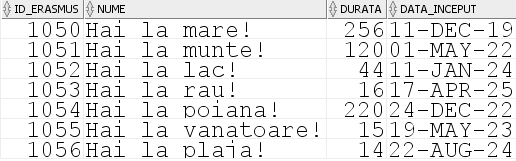


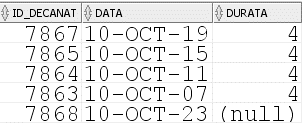
Inserarea în tabele:

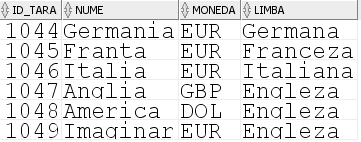


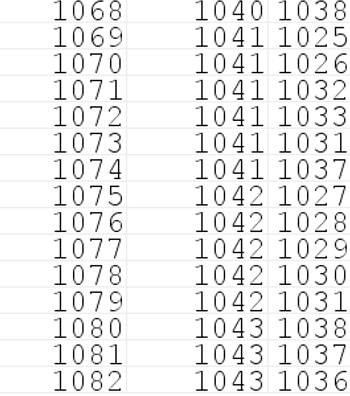
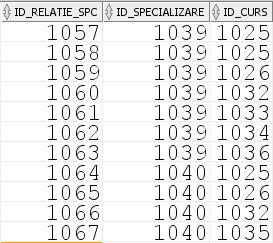


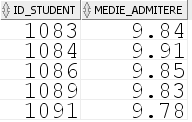
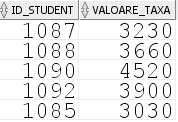


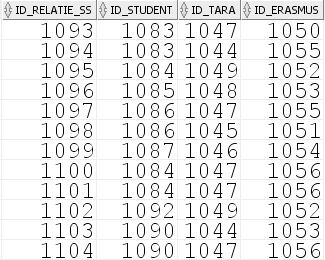
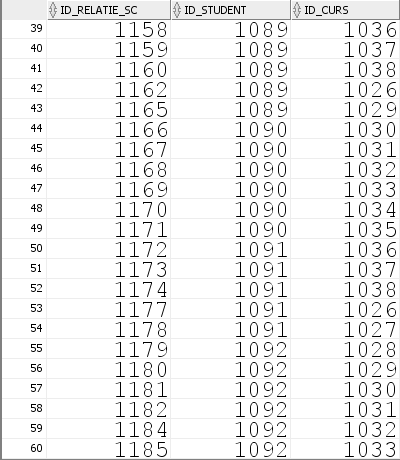


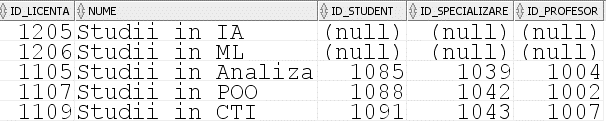


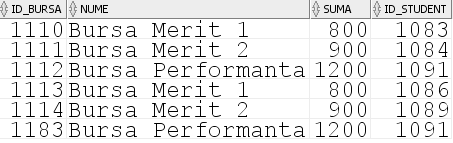


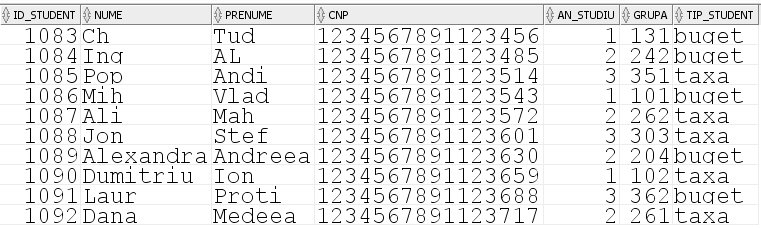








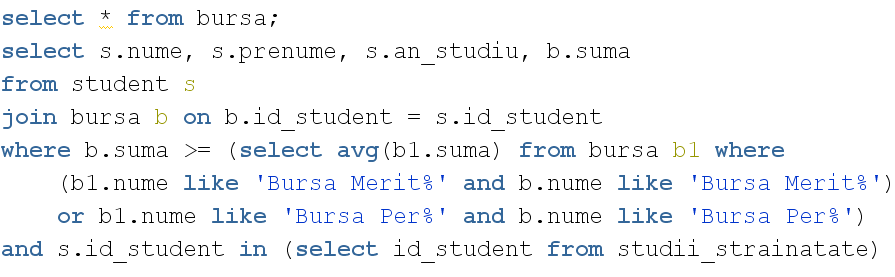




# Cerința 12

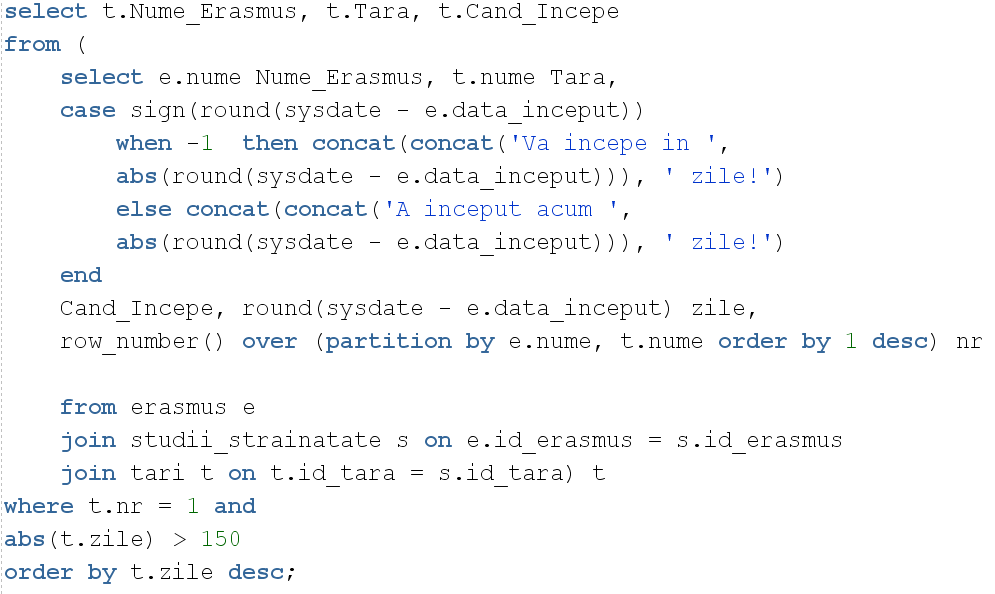
Formulați în limbaj natural și implementați 5 cereri SQL complexe

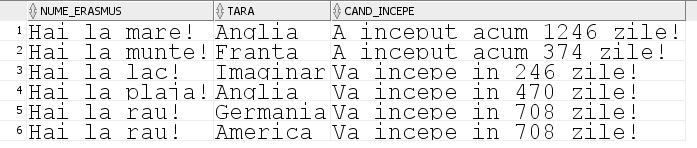
* Vrem să știm numele, prenumele și anul de studiu al studenților care au bursă cu o valoare mai mare decat media tipului respectiv de burse, precizând și această suma. Dintre acești studenți ne interesează doar cei care studiază în străinătate.



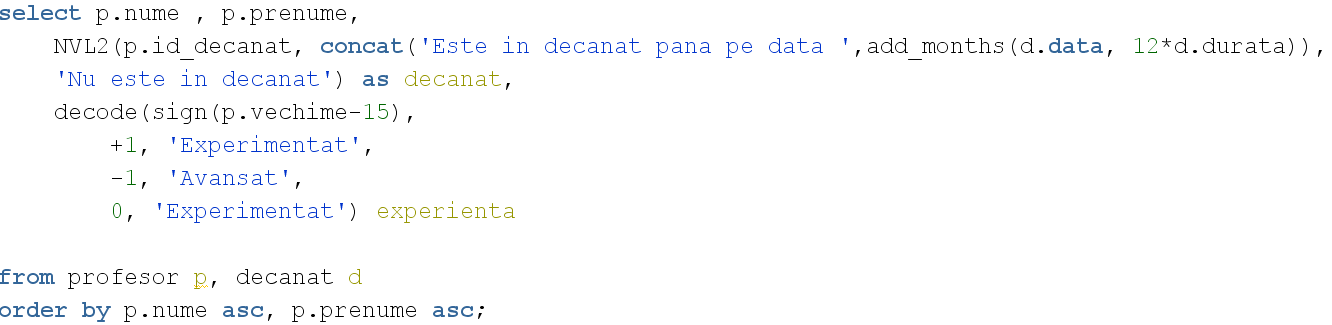


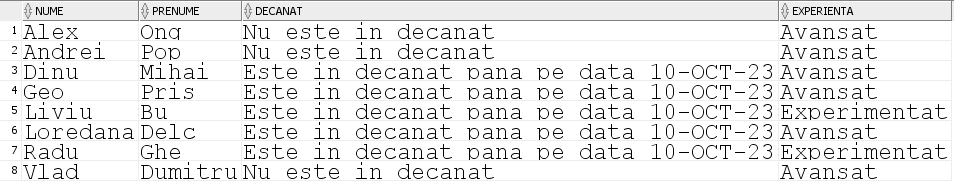
* Să se afișeze toate proiectele Erasmus unice prin nume, țara în care se desfășoară, când a început/începe, ordonat cronologic, exceptând proiectele care se alfă la 150 de zile de data curentă. Fiecare coloană să abiă un nume sugestiv, iar pentru coloana „Când\_Începe” să fie afișat un mesaj sugestiv.

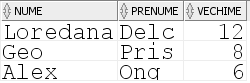


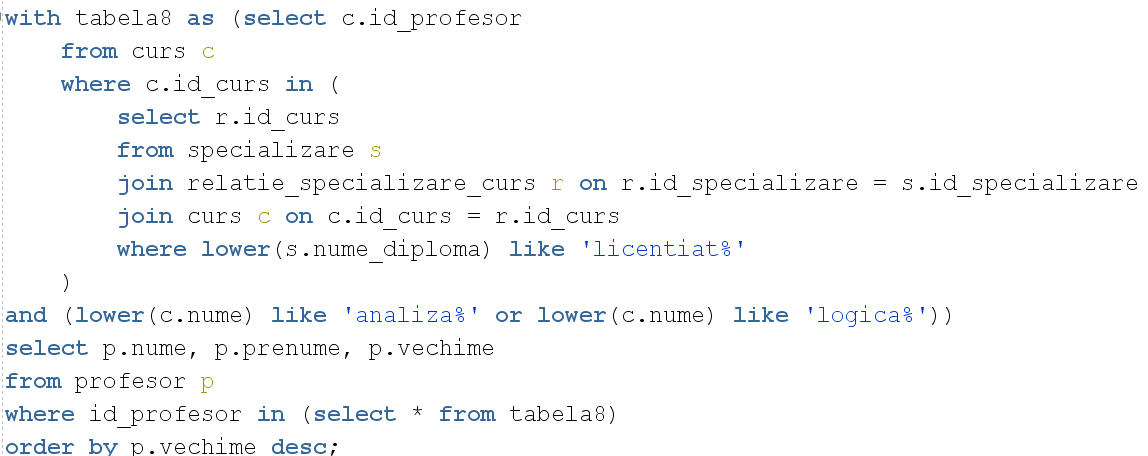


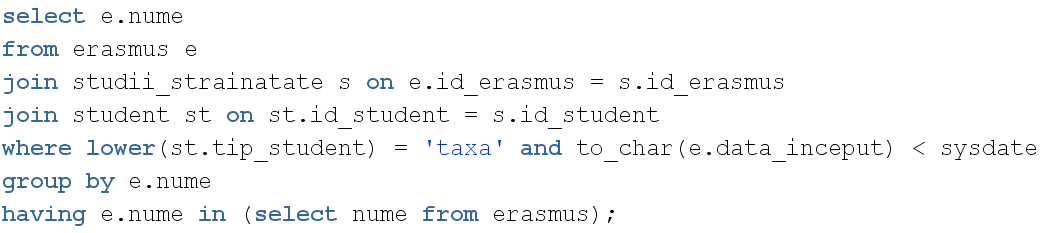
* Vrem o listă cu toți profesorii facultății. Aceasta trebuie să conțină numele și prenumele acestora. Dacă se află în decanat, să se afișeze până la ce data, dacă nu să se afișeze un mesaj sugestiv. Pentru profesorii cu vechime de minim 15 ani să se ajișeze eticheta „Experimentat”, iar pentru ceilalți o altă etichetă sugestivă. Lista să fie ordonată după numele și prenumele profesorilor.

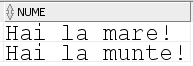




* Vrem o listă cu toți cu numele, prenumele și vechimea profesorilor care predau cursuri de Analiză și Logică la specializări care oferă diplomă de Licențiat (nu de Inginer). Lista trebuie sortata descrescător după vechimea profesorilor.

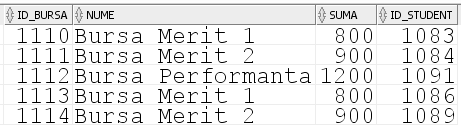
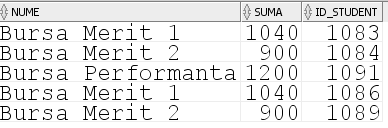


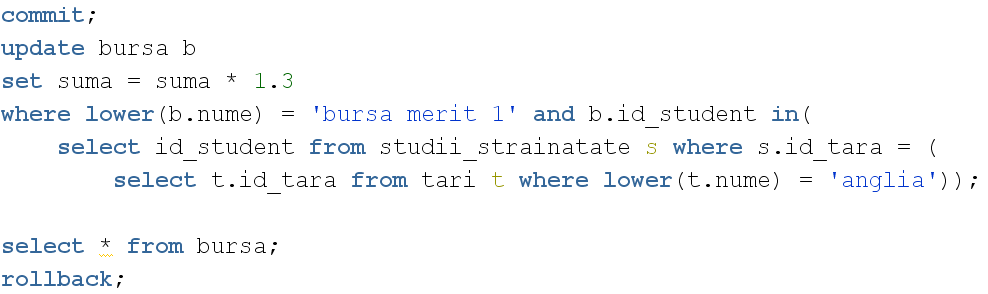
* Vrem o listă cu numele proiectelor Erasmus care au început în trecut și în care au participat cel puțin un student la buget.

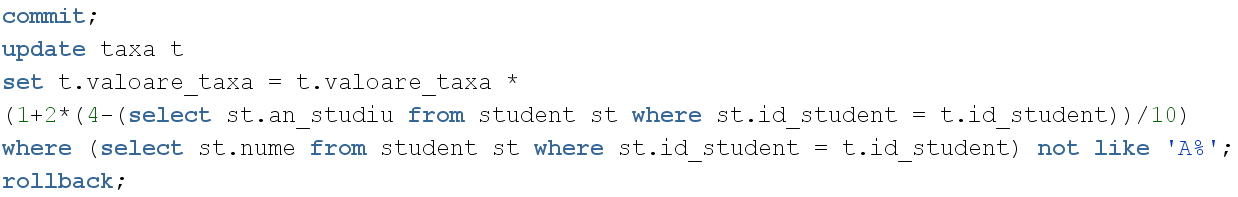


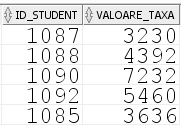
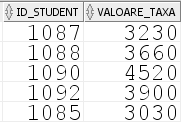
# Cerința 13

Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.

* Să se mărească bursa „Bursă Merit 1” cu 30% studenților care sunt într-un proiect Erasmus în Anglia.

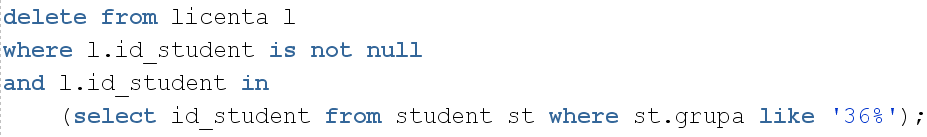


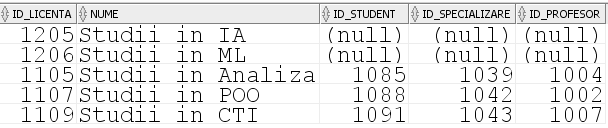
* Taxele facultății s-au mărit pentru toți studenții, în funcție de anul de studiu. Pentru cei de anul I cu 30%, pentru cei de anul II cu 20%, iar pentru cei de anul 3 cu 10%. Momentan acestea au fost mărite pentru cei cu inițiala numelui „A”. Vrem să mărim taxele și pentru ceilalți.



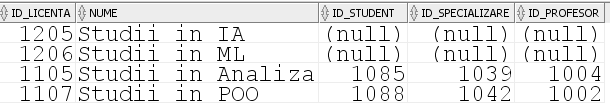


* În urma unor tentative de copiere, vrem să stergem din baza de date toate lucrările de licență, care încă nu au fost predate, susținute de studenți din grupele 360-369.



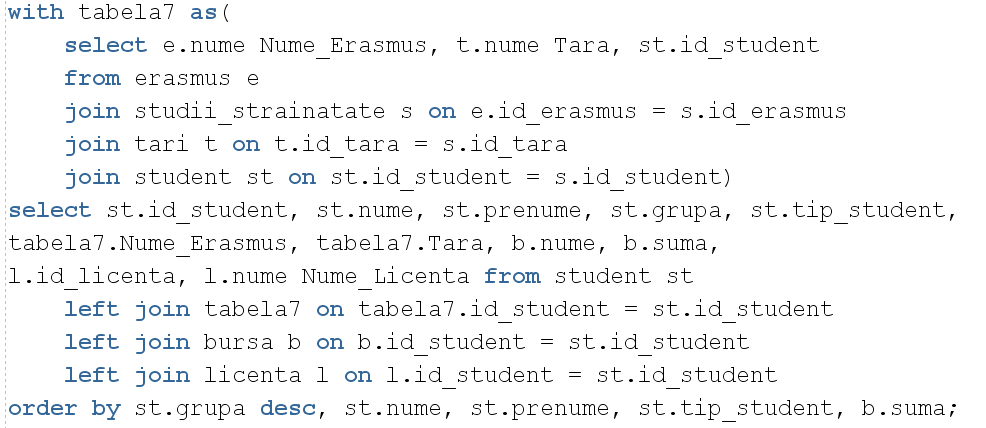
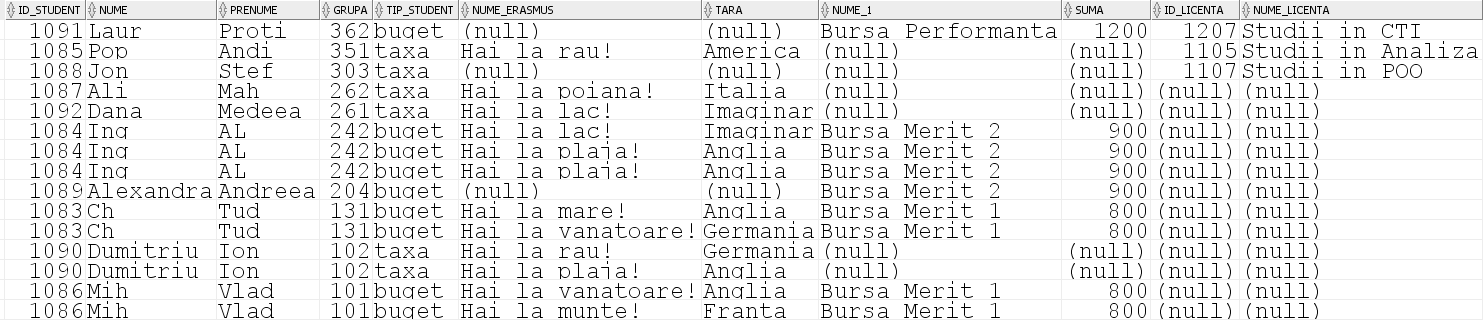


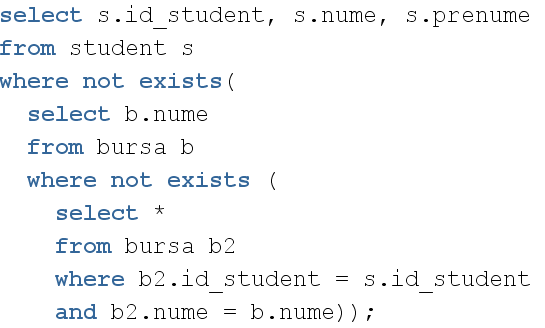




# Cerința 15

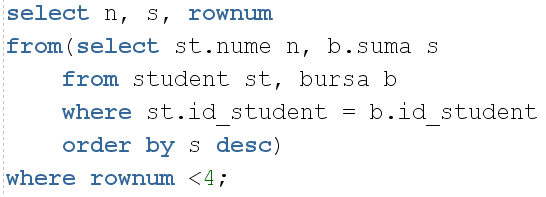
Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outerjoin pe minimum 4 tabele, o cerere ce utilizează operația division și o cerere care implementează analiza top-n.

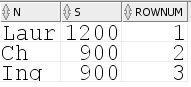
* Vrem un catalog complex cu studenții facultății în care să apară informațiile importante legate de student, bursele acestuia, proiectele Erasmus și țările respective, dar și licență dacă este cazul. Vrem duplicate în cazul în cazul în care un student merge în mai multe proiecte Erasmus.
* Vrem o listă cu toți studenții care primesc toate bursele. (bursele sunt grupate după nume)





* Vrem să facem o analiză asupra burselor ca să vedem 3 studenți cu cele mai mari burse.

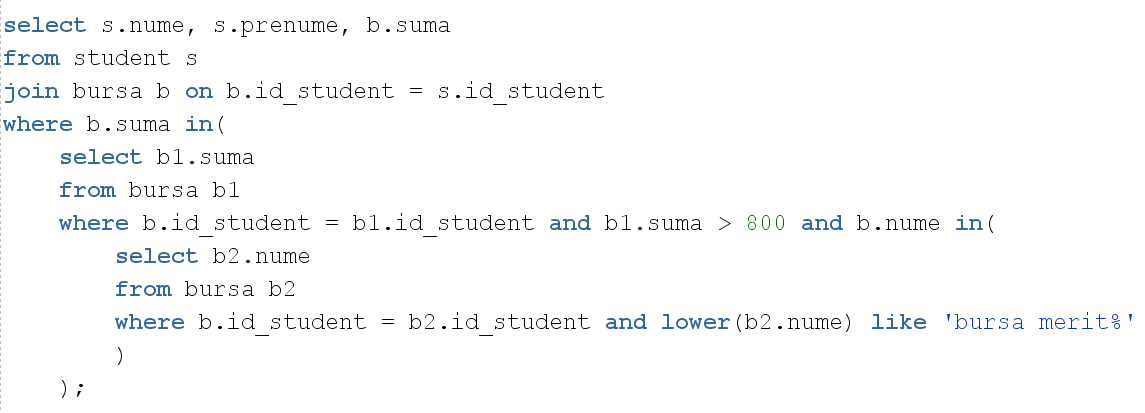




# Cerința 16

Optimizarea unei cereri, aplicând regulile de optimizare ce derivă din proprietățile operatorilor algebrei relaționale. Cererea va fi exprimată prin expresie algebrică, arbore algebric și limbaj (SQL), atât anterior cât și ulterior optimizării.

* Să se afișeze studenții la buget (nume, prenume) care au o bursă de merit în valoare de mai mult de 800 de lei.

**Inițial**

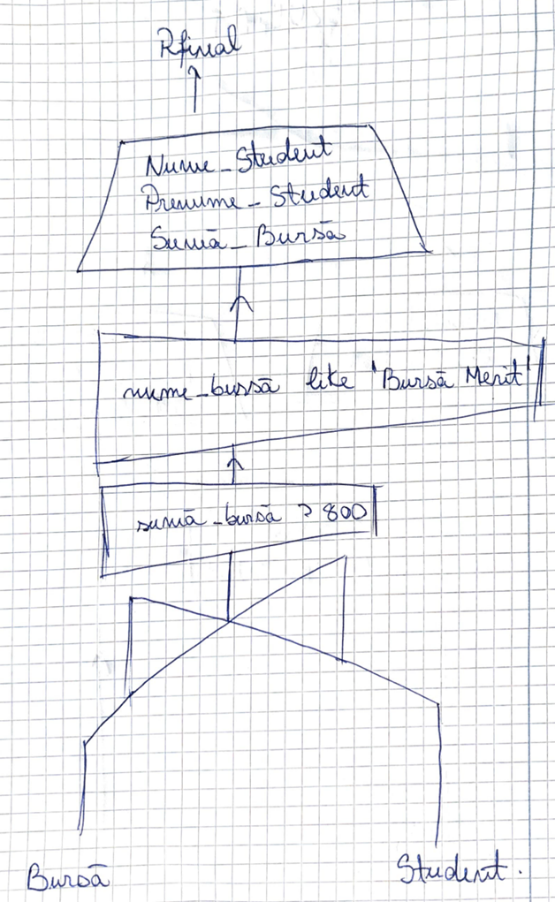
Expresie algebrică:

R1 = Join (Student, Bursă)

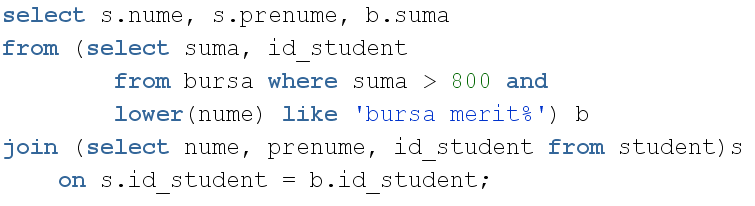
R2 = Select(R1, Suma\_Bursa > 800)

R3 = Select(R2, Nume\_Bursa like ”bursa merit%”)

Rfinal = Project(R3, Nume\_Student, Prenume\_Student, Suma\_Bursa)

Arbore algebric:

**După optimizare**



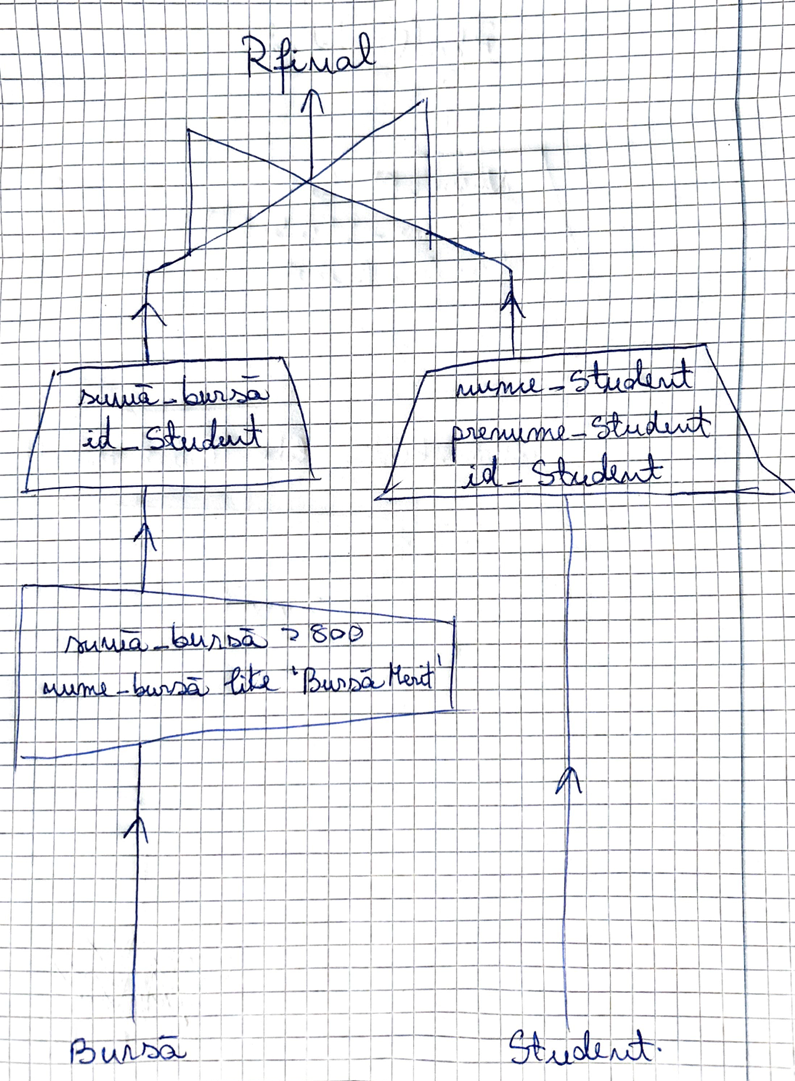
Expresie algebrică:

R1 = Select(Bursa, Suma\_Bursa > 800 and Nume\_Bursa like ”bursa merit%”)

R2 = Project(R1, Suma\_bursa, Id\_Student)

R3 = Project(Student, Nume\_Student, Prenume\_Student, Id\_Student)

Rfinal = Join(R2, R3)



Arbore algebric:

# Cerința 17

a) Realizarea normalizării BCNF, FN4, FN5.

BCNF: Pentru ca o tabelă să fie în BCNF, aceasta trebuie să fie în FN3 și pentru fiecare dependență între A și B, A trebuie să fie o super cheie. Presupunem pentru modelul prezentat că tuplul de forma student-erasmus-țară are la bază un program precis pentru fiecare proiect Erasmus. În tabela Studii-Străinătate avem dependențele (id\_student, id\_țară, id\_erasmus) -> id\_program și id\_program -> id\_erasmus, ceea ce nu respectă regulile BCNF. Pentru asta creăm tabela Program(id\_program (PK), activitate\_principala, timp\_liber, tip\_cazare), iar tabela studii\_străinătate devine (id\_relatie\_ss (PK), id\_tara (FK), id\_erasmus (FK), id\_student (FK), id\_program (FK)).

FN4: Pentru ca o tabelă să fie în FN4, aceasta trebuie să fie în BCNF și să nu aibă dependențe triviale multiple. Un exemplu este în cazul tabelei licență avem drep cheie candidat (id\_student, id\_specializare, id\_profesor) și dependențele sunt id\_student -> id\_profesor, id\_student -> id\_specializare. Tabela licență fiind în BCNF rămâne să o descompunem în două tabele: student-profesor (id\_student, id\_profesor), student-specializare (id\_student,id\_specializare).

FN5: Pentru ca o tabelă să fie în FN5, aceasta trebuie să fie în FN4 și orice dependență de uniune la R este dedusă de cheia candidat a lui R. Un caz particular al dependenței de uniune este dependența trivială multiplă, așa că orice tabelă în FN5 este și în FN4. Pe modelul propus presupunem că în tabela Erasmus avem (id\_erasmus, nume\_proiect, activitate\_principală), unde cele trei coloane alcătuiesc o cheie candidat. Dependențele sunt în fiecare din perechile (id\_erasmus, nume\_proiect), (nume\_proiect, activitate\_principală), (activitate\_principală id\_erasmus). Nu avem dependențe triviale multiple așa că tabela este în FN4. Pentru a o aduce la FN5, descompunem în trei tabele, respectiv: Erasmus(id\_erasmus, nume\_proiect), Proiect(nume\_proiect, activitate\_principală), Activitate(activitate\_principală id\_erasmus).

b) Denormalizare

Pentru a usura filtrarea datelor și a obține informațiile căutate mult mai rapid apelăm la denormalizare. Chiar dacă adăugăm infomații redundante, query-urile devin mai eficiente. Un astfel de exemplu pe modelul propus ar fi în tabela studii-străinătate, să adaugăm nume, prenume, an\_studiu, informații care se află deja în tabela student.

# Cod SQL

select \* from dual;

SELECT owner, table\_name FROM all\_tables;

create sequence id\_gen

increment by 1

start with 1025;

create sequence cnp\_gen

increment by 29

start with 123456789123456;

create table decanat(id\_decanat number Primary Key, data date, durata number);

create table profesor(id\_profesor number Primary Key,

nume varchar2(30) not null, prenume varchar2(30) not null, cnp varchar2(16),

id\_specializare number(5), vechime number(2), id\_decanat number(5));

alter table profesor drop column id\_specializare;

create table curs(id\_curs number(5) Primary Key, nume varchar2(30) not null,

id\_specializare number(5) not null, id\_profesor number(5) not null);

alter table curs drop column id\_specializare;

create table specializare(id\_specializare number(5) Primary Key,

nume varchar2(30) not null, nume\_diploma varchar2(60) not null);

alter table specializare

add constraint nume\_specializare check(nume = 'Matematica Pura' or nume = 'CTI' or nume = 'Informatica' or

nume = 'Matematica Informatica' or nume = 'Matematica Aplicata');

select \* from specializare;

create table licenta(id\_licenta number(5) Primary Key,

nume varchar2(30) not null unique, id\_student number(5) not null,

id\_specializare number(5) not null, id\_profesor number(5) not null);

alter table licenta

modify id\_student number(5) null;

alter table licenta

modify id\_profesor number(5) null;

alter table licenta

modify id\_specializare number(5) null;

select \* from licenta;

create table bursa(id\_bursa number(5) Primary Key, nume varchar2(30) not null,

suma number(4) not null, id\_student number(5) not null);

create table taxa(id\_student number(5) Primary Key, valoare\_taxa number(4) not null);

create table buget(id\_student number(5) Primary Key, medie\_admitere float(2) not null);

alter table buget

modify (medie\_admitere decimal(4,2));

create table student(id\_student number(5) Primary Key, nume varchar2(30) not null,

prenume varchar2(30) not null, cnp varchar2(16) not null, an\_studiu number(1) not null,

grupa number(3) not null, tip\_student varchar2(20) not null);

alter table student

add constraint grupa\_3 check(grupa between 100 and 500);

create table erasmus(id\_erasmus number(5) Primary Key, nume varchar2(30) not null,

durata number(5) not null, data\_inceput date not null);

create table tari(id\_tara number(5) Primary Key, nume varchar2(30) not null,

moneda varchar2(3) not null, limba varchar2(30) not null);

create table relatie\_student\_curs(id\_relatie\_sc number(5) Primary Key,

id\_student number(5), id\_curs number(5));

create table relatie\_specializare\_curs(id\_relatie\_spc number(5) Primary Key,

id\_specializare number(5), id\_curs number(5));

create table studii\_strainatate(id\_relatie\_ss number(5) Primary Key,

id\_student number(5) not null, id\_tara number(5) not null, id\_erasmus number(5) not null);

alter table taxa

add foreign key(id\_student) references student(id\_student);

alter table taxa

add foreign key(id\_student) references student(id\_student) on delete cascade;

alter table studii\_strainatate

add foreign key(id\_student) references student(id\_student);

alter table studii\_strainatate

add foreign key(id\_erasmus) references erasmus(id\_erasmus);

alter table studii\_strainatate

add foreign key(id\_tara) references tari(id\_tara);

alter table bursa

add foreign key(id\_student) references student(id\_student);

alter table bursa

add foreign key(id\_student) references student(id\_student);

alter table relatie\_student\_curs

add foreign key(id\_student) references student(id\_student);

alter table relatie\_student\_curs

add foreign key(id\_curs) references curs(id\_curs);

alter table relatie\_specializare\_curs

add foreign key(id\_curs) references curs(id\_curs);

alter table relatie\_specializare\_curs

add foreign key(id\_specializare) references specializare(id\_specializare);

alter table licenta

add foreign key(id\_student) references student(id\_student);

alter table licenta

add foreign key(id\_profesor) references profesor(id\_profesor);

alter table licenta

add foreign key(id\_specializare) references specializare(id\_specializare);

alter table curs

add foreign key(id\_profesor) references profesor(id\_profesor);

alter table profesor

add foreign key(id\_decanat) references decanat(id\_decanat);

alter table taxa

add foreign key(id\_student) references student(id\_student);

alter table buget

add foreign key(id\_student) references student(id\_student);

insert into decanat values(7867, to\_date('10.10.2019', 'DD.MM.YYYY'), 4);

insert into decanat values(7865, to\_date('10.10.2015', 'DD.MM.YYYY'), 4);

insert into decanat values(7864, to\_date('10.10.2011', 'DD.MM.YYYY'), 4);

insert into decanat values(7863, to\_date('10.10.2007', 'DD.MM.YYYY'), 4);

insert into decanat values(7868, to\_date('10.10.2023', 'DD.MM.YYYY'), null);

select \* from decanat;

delete from decanat

where id\_decanat = 7868;

select \* from profesor;

insert into profesor values(1000, 'Dinu', 'Mihai', 111, 10, 7867);

insert into profesor values(1001, 'Radu', 'Ghe', 112, 16, 7867);

insert into profesor values(1002, 'Liviu', 'Bu', 113, 22, 7867);

insert into profesor values(1003, 'Loredana', 'Delc', 114, 12, 7867);

insert into profesor values(1004, 'Geo', 'Pris', 115, 8, 7867);

insert into profesor values(1005, 'Andrei', 'Pop', 116, 9, null);

insert into profesor values(1006, 'Alex', 'Ong', 117, 6, null);

insert into profesor values(1007, 'Vlad', 'Dumitru', 118, 7, null);

insert into curs values(id\_gen.nextval, 'Analiza', 1004);

insert into curs values(id\_gen.nextval, 'Algebra', 1004);

insert into curs values(id\_gen.nextval, 'PA', 1005);

insert into curs values(id\_gen.nextval, 'SD', 1004);

insert into curs values(id\_gen.nextval, 'LFA', 1001);

insert into curs values(id\_gen.nextval, 'BD', 1007);

insert into curs values(id\_gen.nextval, 'POO', 1002);

insert into curs values(id\_gen.nextval, 'Logica', 1006);

insert into curs values(id\_gen.nextval, 'Analiza2', 1003);

insert into curs values(id\_gen.nextval, 'Algebra2', 1003);

insert into curs values(id\_gen.nextval, 'Geometrie', 1003);

insert into curs values(id\_gen.nextval, 'Mecanica Cereasca', 1003);

insert into curs values(id\_gen.nextval, 'IA', 1002);

insert into curs values(id\_gen.nextval, 'RT', 1005);

select \* from curs;

insert into specializare values(id\_gen.nextval, 'Matematica Pura', 'Licentiat in Matematica Pura');

insert into specializare values(id\_gen.nextval, 'Matematica Aplicata', 'Licentiat in Matematica Aplicata');

insert into specializare values(id\_gen.nextval, 'Matematica Informatica', 'Licentiat in Matematica Informatica');

insert into specializare values(id\_gen.nextval, 'Informatica', 'Licentiat in Informatica');

insert into specializare values(id\_gen.nextval, 'CTI', 'Inginer in CTI');

select \* from specializare;

insert into tari values(id\_gen.nextval, 'Germania', 'EUR', 'Germana');

insert into tari values(id\_gen.nextval, 'Franta', 'EUR', 'Franceza');

insert into tari values(id\_gen.nextval, 'Italia', 'EUR', 'Italiana');

insert into tari values(id\_gen.nextval, 'Anglia', 'GBP', 'Engleza');

insert into tari values(id\_gen.nextval, 'America', 'DOL', 'Engleza');

insert into tari values(id\_gen.nextval, 'Imaginar', 'EUR', 'Engleza');

select \* from tari;

insert into erasmus values(id\_gen.nextval, 'Hai la mare!', 256, to\_date('11.12.2019','DD.MM.YYYY'));

insert into erasmus values(id\_gen.nextval, 'Hai la munte!', 120, to\_date('01.05.2022','DD.MM.YYYY'));

insert into erasmus values(id\_gen.nextval, 'Hai la lac!', 44, to\_date('11.01.2024','DD.MM.YYYY'));

insert into erasmus values(id\_gen.nextval, 'Hai la rau!', 16, to\_date('17.04.2025','DD.MM.YYYY'));

insert into erasmus values(id\_gen.nextval, 'Hai la poiana!', 220, to\_date('24.12.2022','DD.MM.YYYY'));

insert into erasmus values(id\_gen.nextval, 'Hai la vanatoare!', 15, to\_date('19.05.2023','DD.MM.YYYY'));

insert into erasmus values(id\_gen.nextval, 'Hai la plaja!', 14, to\_date('22.08.2024','DD.MM.YYYY'));

select \* from erasmus;

insert into relatie\_specializare\_curs values(id\_gen.nextval, 1039, 1025);

insert into relatie\_specializare\_curs values(id\_gen.nextval, 1039, 1026);

insert into relatie\_specializare\_curs values(id\_gen.nextval, 1039, 1032);

insert into relatie\_specializare\_curs values(id\_gen.nextval, 1039, 1033);

insert into relatie\_specializare\_curs values(id\_gen.nextval, 1039, 1034);

insert into relatie\_specializare\_curs values(id\_gen.nextval, 1039, 1036);

insert into relatie\_specializare\_curs values(id\_gen.nextval, 1040, 1025);

insert into relatie\_specializare\_curs values(id\_gen.nextval, 1040, 1026);

insert into relatie\_specializare\_curs values(id\_gen.nextval, 1040, 1032);

insert into relatie\_specializare\_curs values(id\_gen.nextval, 1040, 1035);

insert into relatie\_specializare\_curs values(id\_gen.nextval, 1040, 1038);

insert into relatie\_specializare\_curs values(id\_gen.nextval, 1041, 1025);

insert into relatie\_specializare\_curs values(id\_gen.nextval, 1041, 1026);

insert into relatie\_specializare\_curs values(id\_gen.nextval, 1041, 1032);

insert into relatie\_specializare\_curs values(id\_gen.nextval, 1041, 1033);

insert into relatie\_specializare\_curs values(id\_gen.nextval, 1041, 1031);

insert into relatie\_specializare\_curs values(id\_gen.nextval, 1041, 1037);

insert into relatie\_specializare\_curs values(id\_gen.nextval, 1042, 1027);

insert into relatie\_specializare\_curs values(id\_gen.nextval, 1042, 1028);

insert into relatie\_specializare\_curs values(id\_gen.nextval, 1042, 1029);

insert into relatie\_specializare\_curs values(id\_gen.nextval, 1042, 1030);

insert into relatie\_specializare\_curs values(id\_gen.nextval, 1042, 1031);

insert into relatie\_specializare\_curs values(id\_gen.nextval, 1043, 1038);

insert into relatie\_specializare\_curs values(id\_gen.nextval, 1043, 1037);

insert into relatie\_specializare\_curs values(id\_gen.nextval, 1043, 1036);

select \* from relatie\_specializare\_curs;

insert into student values(id\_gen.nextval, 'Ch', 'Tud', cnp\_gen.nextval, 1, 131, 'buget');

insert into student values(id\_gen.nextval, 'Ing', 'AL', cnp\_gen.nextval, 2, 242, 'buget');

insert into student values(id\_gen.nextval, 'Pop', 'Andi', cnp\_gen.nextval, 3, 351, 'taxa');

insert into student values(id\_gen.nextval, 'Mih', 'Vlad', cnp\_gen.nextval, 1, 101, 'buget');

insert into student values(id\_gen.nextval, 'Ali', 'Mah', cnp\_gen.nextval, 2, 262, 'taxa');

insert into student values(id\_gen.nextval, 'Jon', 'Stef', cnp\_gen.nextval, 3, 303, 'taxa');

insert into student values(id\_gen.nextval, 'Alexandra', 'Andreea', cnp\_gen.nextval, 2, 204, 'buget');

insert into student values(id\_gen.nextval, 'Dumitriu', 'Ion', cnp\_gen.nextval, 1, 102, 'taxa');

insert into student values(id\_gen.nextval, 'Laur', 'Proti', cnp\_gen.nextval, 3, 362, 'buget');

insert into student values(id\_gen.nextval, 'Dana', 'Medeea', cnp\_gen.nextval, 2, 261, 'taxa');

select \* from student;

insert into buget values(1083, 9.84);

delete from buget where medie\_admitere = 9;

insert into buget values(1084, 9.91);

insert into buget values(1086, 9.85);

insert into buget values(1089, 9.83);

insert into buget values(1091, 9.78);

select \* from buget;

insert into taxa values(1085, 3030);

insert into taxa values(1087, 3230);

insert into taxa values(1088, 3660);

insert into taxa values(1090, 4520);

insert into taxa values(1092, 3900);

select \* from taxa;

insert into studii\_strainatate values(id\_gen.nextval, 1083, 1047, 1050);

insert into studii\_strainatate values(id\_gen.nextval, 1083, 1044, 1055);

insert into studii\_strainatate values(id\_gen.nextval, 1084, 1049, 1052);

insert into studii\_strainatate values(id\_gen.nextval, 1085, 1048, 1053);

insert into studii\_strainatate values(id\_gen.nextval, 1086, 1047, 1055);

insert into studii\_strainatate values(id\_gen.nextval, 1086, 1045, 1051);

insert into studii\_strainatate values(id\_gen.nextval, 1087, 1046, 1054);

insert into studii\_strainatate values(id\_gen.nextval, 1084, 1047, 1056);

insert into studii\_strainatate values(id\_gen.nextval, 1092, 1049, 1052);

insert into studii\_strainatate values(id\_gen.nextval, 1090, 1044, 1053);

insert into studii\_strainatate values(id\_gen.nextval, 1090, 1047, 1056);

select \* from studii\_strainatate;

create table licenta(id\_licenta number(5) Primary Key,

nume varchar2(30) not null unique, id\_student number(5) not null,

id\_specializare number(5) not null, id\_profesor number(5) not null);

insert into licenta values(id\_gen.nextval, 'Studii in Analiza', 1085, 1039, 1004);

insert into licenta values(id\_gen.nextval, 'Studii in POO', 1088, 1042, 1002);

insert into licenta values(id\_gen.nextval, 'Studii in CTI', 1091, 1043, 1007);

insert into licenta values(id\_gen.nextval, 'Studii in IA', null, null, null);

insert into licenta values(id\_gen.nextval, 'Studii in ML', null, null, null);

select \* from licenta;

insert into bursa values(id\_gen.nextval, 'Bursa Merit 1', 800, 1083);

insert into bursa values(id\_gen.nextval, 'Bursa Merit 2', 900, 1084);

insert into bursa values(id\_gen.nextval, 'Bursa Performanta', 1200, 1091);

insert into bursa values(id\_gen.nextval, 'Bursa Merit 1', 800, 1086);

insert into bursa values(id\_gen.nextval, 'Bursa Merit 2', 900, 1089);

alter table bursa

add constraint suma\_minima check(suma >= 500) ;

alter table bursa

add constraint suma\_maxima check(suma <=1500);

select \* from bursa;

create sequence id\_fill

start with 1026

increment by 1

maxvalue 1038;

insert into relatie\_student\_curs values(id\_gen.nextval, 1092, id\_fill5.nextval);

select \* from relatie\_student\_curs;

commit;

select \* from relatie\_student\_curs;

select \* from curs;

select \* from erasmus;

select \* from studii\_strainatate;

select \* from profesor;

select \* from decanat;

------------------CERINTA 12------------------

select t.Nume\_Erasmus, t.Tara, t.Cand\_Incepe

from (

select e.nume Nume\_Erasmus, t.nume Tara,

case sign(round(sysdate - e.data\_inceput))

when -1 then concat(concat('Va incepe in ',

abs(round(sysdate - e.data\_inceput))), ' zile!')

else concat(concat('A inceput acum ',

abs(round(sysdate - e.data\_inceput))), ' zile!')

end

Cand\_Incepe, round(sysdate - e.data\_inceput) zile,

row\_number() over (partition by e.nume, t.nume order by 1 desc) nr

from erasmus e

join studii\_strainatate s on e.id\_erasmus = s.id\_erasmus

join tari t on t.id\_tara = s.id\_tara) t

where t.nr = 1 and

abs(t.zile) > 150

order by t.zile desc;

select p.nume , p.prenume,

NVL2(p.id\_decanat, concat('Este in decanat pana pe data ',add\_months(d.data, 12\*d.durata)),

'Nu este in decanat') as decanat,

decode(sign(p.vechime-15),

+1, 'Experimentat',

-1, 'Avansat',

0, 'Experimentat') experienta

from profesor p, decanat d

order by p.nume asc, p.prenume asc;

with tabela8 as (select c.id\_profesor

from curs c

where c.id\_curs in (

select r.id\_curs

from specializare s

join relatie\_specializare\_curs r on r.id\_specializare = s.id\_specializare

join curs c on c.id\_curs = r.id\_curs

where lower(s.nume\_diploma) like 'licentiat%'

)

and (lower(c.nume) like 'analiza%' or lower(c.nume) like 'logica%'))

select p.nume, p.prenume, p.vechime

from profesor p

where id\_profesor in (select \* from tabela8)

order by p.vechime desc;

select \* from bursa;

select s.nume, s.prenume, s.an\_studiu, b.suma

from student s

join bursa b on b.id\_student = s.id\_student

where b.suma >= (select avg(b1.suma) from bursa b1 where

(b1.nume like 'Bursa Merit%' and b.nume like 'Bursa Merit%')

or b1.nume like 'Bursa Per%' and b.nume like 'Bursa Per%')

and s.id\_student in (select id\_student from studii\_strainatate)

;

select e.nume

from erasmus e

join studii\_strainatate s on e.id\_erasmus = s.id\_erasmus

join student st on st.id\_student = s.id\_student

where lower(st.tip\_student) = 'buget' and to\_char(e.data\_inceput) < sysdate

group by e.nume

having e.nume in (select nume from erasmus);

select \* from erasmus;

select \* from relatie\_specializare\_curs;

select \* from studii\_strainatate;

select \* from tari;

select \* from student;

select \* from bursa;

commit;

----------------------CERINTA 13----------------------

update bursa b

set suma = (select

case sign(b.suma - 1154)

when -1 then suma\*1.3

else 1500

end

from bursa b2 where b2.id\_bursa = b.id\_bursa)

where lower(b.nume) = 'bursa merit 1' and b.id\_student in(

select id\_student from studii\_strainatate s where s.id\_tara = (

select t.id\_tara from tari t where lower(t.nume) = 'anglia'));

rollback;

commit;

update taxa t

set t.valoare\_taxa = t.valoare\_taxa \*

(1+2\*(4-(select st.an\_studiu from student st where st.id\_student = t.id\_student))/10)

where (select st.nume from student st where st.id\_student = t.id\_student) not like 'A%';

rollback;

select \* from taxa;

rollback;

select \* from licenta;

delete from licenta l

where l.id\_student is not null

and l.id\_student in

(select id\_student from student st where st.grupa like '36%');

----------------------CERINTA 15----------------------

with tabela7 as(

select e.nume Nume\_Erasmus, t.nume Tara, st.id\_student

from erasmus e

join studii\_strainatate s on e.id\_erasmus = s.id\_erasmus

join tari t on t.id\_tara = s.id\_tara

join student st on st.id\_student = s.id\_student)

select st.id\_student, st.nume, st.prenume, st.grupa, st.tip\_student,

tabela7.Nume\_Erasmus, tabela7.Tara, b.nume, b.suma,

l.id\_licenta, l.nume Nume\_Licenta from student st

left join tabela7 on tabela7.id\_student = st.id\_student

left join bursa b on b.id\_student = st.id\_student

left join licenta l on l.id\_student = st.id\_student

order by st.grupa desc, st.nume, st.prenume, st.tip\_student, b.suma;

select \* from student;

select s.id\_student, s.nume, s.prenume

from student s

where not exists(

select b.nume

from bursa b

where not exists (

select \*

from bursa b2

where b2.id\_student = s.id\_student

and b2.nume = b.nume));

select n, s, rownum

from(select st.nume n, b.suma s

from student st, bursa b

where st.id\_student = b.id\_student

order by s desc)

where rownum <4;

select distinct suma from bursa;

select \* from bursa;

select \* from profesor;

select \* from student;

commit;

insert into bursa values (1115, 'Bursa Merit 2', 900, 1083);

insert into bursa values (1116, 'Bursa Performanta', 1200, 1083);

---------------------CERINTA 16---------------------

select s.nume, s.prenume, b.suma

from student s

join bursa b on b.id\_student = s.id\_student

where b.suma in(

select b1.suma

from bursa b1

where b.id\_student = b1.id\_student and b1.suma > 800 and b.nume in(

select b2.nume

from bursa b2

where b.id\_student = b2.id\_student and lower(b2.nume) like 'bursa merit%'

)

);

select s.nume, s.prenume, b.suma

from (select suma, id\_student

from bursa where suma > 800 and

lower(nume) like 'bursa merit%') b

join (select nume, prenume, id\_student from student)s

on s.id\_student = b.id\_student;

----------------------------cod extra--------------------------

with tabel1 as(

select

s.id\_student, s.nume nume\_stud, s.prenume, b.nume, b.suma

/\*row\_number() over (partition by s.id\_student, s.nume, s.prenume order by b.suma desc, b.nume) nr\*/

from student s

join bursa b on b.id\_student = s.id\_student)

/\*group by s.id\_student, s.nume, s.prenume, b.suma\*/

select

t.nume\_stud,

decode(t.nume, 'Bursa Merit 1', 'Bursa pentru media de admitere',

'Bursa Merit 2', 'Bursa pentru media anuala',

'Bursa Performanta', 'Bursa pentru rezultate exceptionale') as motiv\_bursa

from tabel1 t;

with tabel2 as(

select

s.id\_student, s.nume nume\_stud, s.prenume, c.nume

from student s

join relatie\_student\_curs r on s.id\_student = r.id\_student

join curs c on c.id\_curs = r.id\_curs)

select

t.nume\_stud, t.prenume, t.nume

from tabel2 t

order by t.nume\_stud, t.nume asc;

/\*where t.nume = 'Algebra';\*/

with tabela4 as(

select p.nume nume, p.prenume prenume, d.id\_decanat id\_d

from profesor p

join decanat d on d.id\_decanat = p.id\_decanat )

select t.nume, t.prenume, t.id\_d

from tabela4 t;

FROM dual;

select

case sign(b.suma - 1154)

when -1 then suma\*1.3

else 1500

end

from bursa b;