

**GESTIUNEA UNEI UNIVERSITĂȚI**

**Broscoțeanu Daria-Mihaela**

**Grupa 143**

**Studiu de caz – Ex. 1**

Vom prezenta modelul de date, restricţiile pe care trebuie să le respecte şi vom încerca, într-o manieră didactică, să construim diagrama E/R corespunzătoare. Vom considera, în abordarea iniţială, anumite situaţii care nu sunt optime, în sensul că pot genera redundanţă, anomalii la reactualizări sau nu permit rezolvarea anumitor interogări asupra modelului. Vom încerca să arătăm care sunt deficienţele modelului, situaţiile care le-au generat şi cum pot fi corectate (parţial sau total) anomaliile respective.

Baza de date conține informații cu privire la facultățile unei universități, secțiile si grupele acestora, cât și profesorii care predau în cadrul universității, ce cursuri și căror studenți le predau aceștia, dacă sunt voluntari in cadrul unei asociaţii sau nu, cât și notele obținute de elevi în cadrul unui curs care se va desfăşura într-o anume sală.

Scopul creării acestei baze de date este de a putea ține evidența studenților din cadrul universității, a situației lor școlare, cât și cursurilor predate în cadrul fiecărei facultăți, cât și a profesorului care susține cursul.

Fiecare student este repartizat la o secție, într-o serie și grupă, participând la anumite cursuri predate de profesori care au posibilitatea de a avea contracte cu mai multe facultăți din cadrul universității. Studenții pot lua parte la acțiuni de voluntariat în cadrul unei asociații studențești.

Cursurile la care iau parte studenții sunt susținute la nivel de grupă, în diferite săli ale facultății, de către profesori diferiți.

Un profesor are contracte cu facultățile la care predă, acesta putând să predea diferite cursuri în cadrul unei facultăți.

Un examen poate lua mai multe forme: proiect, examen scris sau interviu. Forma acestuia diferă în funcție de materia la care este susținut. Un student promovează un examen în cazul în care obține o notă mai mare sau egală cu 5.

**Restricţii de funcţionare – Ex. 2**

Modelul de date respectă anumite restricţii de funcţionare.

* Un profesor este angajat la cel puțin o facultate, iar o facultate angajează cel puțin un profesor.
* O facultate are cel puțin o secție.
* O secție are cel puțin o serie.
* O serie este alcătuită din cel puțin o grupă.
* O grupă este formată din mai mulți studenți, iar un student poate să fie înscris la mai multe grupe.
* Un student poate să fie membru intr-o asociație de voluntariat.
* O asociație de voluntariat aparține doar unei facultăți.
* Un student promovează unul sau mai multe examene, un examen este promovat de unul sau mai mulți studenți.
* Un examen aparține unui curs, un curs poate avea mai multe examene.
* Un curs este predat de unul sau mai mulți profesori. Un curs este predat unei sau mai multor grupe. Un curs se poate desfășura în una sau mai multe săli.
* Un curs poate fi seminar, laborator sau curs.
* Un examen poate fi proiect, lucrare scrisă sau interviu.
* O asociație are cel puțin un membru.
* O grupă este formată din cel puțin un student.

**Entităţi – Ex. 3**

Pentru modelul de date referitor la gestiunea unei universităţi, structurile FACULTATE, SECTIE, SERIE, GRUPA, STUDENT, ASOCIATIE, SALA, PROFESOR, CURS, EXAMEN reprezintă entităţi.

Vom prezenta entităţile modelului de date, dând o descriere completă a fiecăreia. De asemenea, pentru fiecare entitate se va preciza cheia primară.

Toate entităţile care vor fi prezentate sunt independente, cu excepţia entităţilor dependente INSCRIERE, PROMOVEAZA, ORAR şi CONTRACT.

*FACULTATE* = instituţia în care se desfăşoară procesul de informare al studenţilor. Modelul de date consideră doar facultăţile semnificative din cadrul unei universităţi. Cheia primară este id*\_*facultate.

*SECTIE* = modalitate de clasificare a studenţilor în cadrul facultăţii. In funcţie de aceasta, între doi studenţi aparţinând a două serii diferite pot să difere profesorii, cursurile, grupele şi sălile. Cheia primară este id*\_*sectie.

*SERIE* ­= modalitate de clasificare a studenţilor în cadrul unei secţii. Studenţii aflaţi în aceeaşi serie au aceeaşi profesori în cadrul cursurilor. Cheia primară este id*\_*serie.

*GRUPA* = modalitate de clasificare a studenţilor în cadrul unei serii. Studenţii aflaţi în aceeaşi grupa au aceeaşi profesori atat in cadrul laboratoarelor, al cursurilor, cât şi al seminariilor. Un student poate face parte din mai multe grupe în cazul în care este înscris la mai multe domenii. Cheia primară este id*\_*grupa.

*STUDENT* = persoana fizică, aparţinând unei facultăţi, care studiază în cadrul universităţii prezentate. Stundentul poate lua parte la activităţi de voluntariat în cadrul asociaţiei de la facultatea la care studiază. Cheia primară este id*\_*student.

*ASOCIATIE* = organizaţie de voluntariat la care pot să ia parte stundenţii din cadrul unei facultăţi. Cheia primară este id*\_*asociatie.

*EXAMEN* = formă de evaluare scrisă sau prin prezentare de proiect prin care un student işi primeşte nota de la o materie. Promovarea examenului implica obţinerea notei 5. Cheia primară este id*\_*examen.

*CURS* = materia pe care o studiază un student pe parcursul unui semestru. Un student poate avea cursuri, seminarii şi laboratoare în cadrul unei materii. Cheia primară este id*\_*curs.

*SALA* = locatia de desfasurare a unui curs. Intr-o sală se pot desfăşura mai multe cursuri, la intervale orare diferite. Salile pot sa fie amfiteatre, sali de clasa si laboratoare. Cheia primară este id*\_*sala.

*PROFESOR* = persoana fizică, angajat al unei facultăţi, care predă în cadrul universităţii prezentate. Profesorul poate preda atât cursuri diferite, cât şi la mai multe facultăţi. Profesorul poate preda atât cursuri, cât şi laboratoare şi seminarii. Cheia primară este id*\_*profesor.

*INSCRIERE* =cuprinde informatii despre grupele din care face parte un student (în cazul în care este înscris la mai multe secţii sau facultăţi în acelaşi timp). Cheia primară compusă este formată din id\_grupa si id\_student.

*ORAR* = programul unei facultăţi, cuprinzând cursurile, laboratoarele, seminariile, programarea lor în săli, cât şi grupele la care participă la o acesta, alături de profesorul care predă. Cheia primară compusă este formată din id\_grupa, id\_profesor, id\_sala și id\_curs.

*CONTRACT* = act care cuprinde informaţii despre durata angajării şi salariul unui profesor în cadrul unei facultăţi. Cum un profesor poate preda la mai multe facultăţi, salariul poate să fie diferit în funcţie de facultate. Cheia primară compusă este alcătuită din id\_profesor și id\_facultate.

*PROMOVEAZA* = identifică nota la un examen si data la care această notă a fost acordată studentului în cadrul unui curs. Cheia primară compusă este alcătuită din id\_examen și id\_student.

**RELAȚII – Ex. 4**

Vom prezenta relaţiile modelului de date, dând o descriere completă a fiecăreia. De fapt, denumirile acestor legături sunt sugestive, reflectând conţinutul acestora şi entităţile pe care le leagă. Pentru fiecare relaţie se va preciza cardinalitatea minimă şi maximă.

cardinalitatea minimă şi maximă.

*PROFESOR\_este\_angajat\_la\_FACULTATE* = relaţie de tip many-to-many care leagă entităţile PROFESOR şi FACULTATE, reflectând legătura dintre acestea (la ce facultate este angajat un anumit profesor). Ea are cardinalitatea minimă 1:0 (o facultate trebuie să aibă cel puțin un profesor angajat şi un profesor poate să fie angajat la cel puțin o facultate) şi cardinalitatea maximă m:*n* (o facultate poate avea mai mulți profesori angajați – de regulă, aceasta este situația – și un profesor poate să predea la mai multe facultăți).

*FACULTATE\_are\_ASOCIATIE* = relaţie care leagă entităţile ASOCIATIE şi FACULTATE, reflectând legătura dintre acestea (pentru o facultate, există o asociație). Relaţia are cardinalitatea minimă și maximă 1:1.

*FACULTATE\_are\_SECTIE* = relaţie care leagă entităţile FACULTATE şi SECTIE, reflectând legătura dintre acestea (pentru o facultate, exista una sau mai multe secții). Relaţia are cardinalitatea minimă 1:1 şi cardinalitatea maximă 1:*n*.

*SECTIE\_are\_SERIE* = relaţie dintre SECTIE şi SERIE, reflectând legătura dintre acestea (ce serii din facultate sunt la o anumită secție). Ea are cardinalitatea minimă 1:1 şi cardinalitatea maximă 1:*n*.

*SERIE\_are\_GRUPE* = relaţie dintre entităţile SERIE şi GRUPE, reflectând legătura dintre acestea (ce grupe alcătuiesc o anumită serie în cadrul unei secții). Ea are cardinalitatea minimă 1:1 şi cardinalitatea maximă 1:*n*.

*STUDENT\_este\_membru\_in\_ASOCIATIE* = relaţie dintre entitatea STUDENT şi ASOCIATIE, reflectând legătura dintre acestea (ce studenți iau parte la activități de voluntariat în cadrul asociației de la facultate). Ea are cardinalitatea minimă 1:1 şi cardinalitatea maximă 1:*n*.

*STUDENT\_promovează\_EXAMEN* = relaţie de tip *many-to-many* dintre entităţile STUDENT şi EXAMEN (ce stundeți promovează ce examene). Ea arată și notele pe care stundeții le obțin, cât și data primirii notei. Relaţia are cardinalitatea minimă 1:1 şi cardinalitatea maximă *m*:*n*.

*PROFESOR\_preda\_CURS\_la\_GRUPA\_in\_SALA* = relaţie de tip 3 ce leagă entităţile PROFESOR, CURS, GRUPA si SALA, reflectând cine a predate un curs, in ce sala era susținut acest curs, ce grupă participa la acesta. Denumirea acestei relaţii va fi *ORAR*.

**Atribute – Ex. 5**

* Entitatea independentă *FACULTATE* are ca atribute:

FACULTATE

Cod\_facultate#

Denumire

Adresa

Telefon

Mail

Fax

Cod\_postal

*cod\_facultate* = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unei facultăți.

*denumire* = variabilă de tip caracter, de lungime maximă 100, care reprezintă denumirea facultății.

*adresă* = variabilă de tip caracter, de lungime maximă 100, care reprezintă adresa facultății.

*telefon* = variabilă de tip caracter, de lungime 10, care reprezintă numărul de telefon al facultății.

*mail* = variabilă de tip caracter, de lungime maximă 50, care reprezintă adresa de mail a unei facultăți.

*fax =*  variabilă de tip caracter, de lungime maximă 50, care reprezintă fax-ul unei facultăți.

*cod\_postal* = variabilă de tip caracter, de lungime maximă 6, care reprezintă codul postal al unei facultăți.

* Entitatea *SECTIE* are ca atribute:

*cod\_sectie* = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul sectiei.

*denumire* = variabilă de tip caracter, de lungime maximă 100, care reprezintă denumirea sectiei.

*cod\_facultate*= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul facultății din care face parte secția. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul FACULTATE.

|  |  |  |
| --- | --- | --- |
| Cod\_Sectie | Cod\_facultate | denumire |
| PK | FK | NOT NULL |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Cod\_facultate | denumire | adresa | telefon | mail | fax | Cod\_postal |
| PK | NOT NULL | NOT NULL | NOT NULL | UNIQUE | - | - |

SECTIE

Cod\_sectie#

Cod\_facultate

Denumire

* Entitatea *SERIE* are ca atribute:

*cod\_serie*= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul seriei.

*denumire* = variabilă de tip caracter, de lungime maximă 100, care reprezintă denumirea seriei.

*cod\_sectie*= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul secției din care face parte seria. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul SECTIE.

|  |  |  |
| --- | --- | --- |
| Cod\_Serie | Cod\_sectie | denumire |
| PK | FK | NOT NULL |

* Entitatea *GRUPA* are ca atribute:

*cod\_grupa*= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul seriei.

*denumire* = variabilă de tip caracter, de lungime maximă 100, care reprezintă denumirea grupei.

*cod\_serie*= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul secției din care face parte seria. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul SERIE.

|  |  |  |
| --- | --- | --- |
| Cod\_grupa | Cod\_serie | denumire |
| PK | FK | NOT NULL |

SERIE

Cod\_serie#

Cod\_sectie

Denumire

GRUPA

Cod\_grupa#

Cod\_serie

Denumire

* Entitatea *STUDENT* are ca atribute:

*cod\_student* = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unui student.

*nume* = variabilă de tip caracter, de lungime maximă 100, care reprezintă numele studentului.

*prenume* = variabilă de tip caracter, de lungime maximă 100, care reprezintă prenumele studentului.

*data\_nastere* = variabilă de tip dată calendaristică, care reprezintă data naşterii studentului respectiv.

*sex* = variabilă de tip caracter, luând valorile *masculin* sau *feminin*, de lungime 10, care reprezintă sexul studentului.

*nationalitate* = variabilă de tip caracter, de lungime maximă 30, care reprezintă naţionalitatea unui student.

*mail* = variabilă de tip caracter, de lungime maximă 50, care reprezintă emailul

*cod\_asociatie* = variabilă de tip intreg, de lungime maximă 5, care reprezintă codul asociației din care face parte studentul. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul ASOCIATIE.

STUDENT

Cod\_student#

Nume

Prenume

Data\_nasterii

Sex

Mail

Telefon

Nationalitate

Cod\_asociatie

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Cod\_student | nume | prenume | Data\_nasterii | mail | Sex | Nationalitate | Cod\_asociatie |
| PK | NOT NULL | NOT NULL | NOT NULL | UNIQUE | NOT NULL | NOT NULL | FK |

PROMOVEAZA

Cod\_student#

Cod\_examen#

Nota

|  |  |  |
| --- | --- | --- |
| Cod\_student | Cod\_facultate | Nota |
| PK | | NOT NULL |

|  |  |  |  |
| --- | --- | --- | --- |
| Cod\_asociatie | denumire | Data\_infiintarii | Cod\_facultate |
| PK | NOT NULL | NOT NULL | FK |

ASOCIATIE

Cod\_asociatie#

denumire

Data\_infiintarii

Cod\_facultate

* Entitatea *ASOCIATIE* are ca atribute:

*cod\_asociatie* = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unui asociatiei.

*denumire* = variabilă de tip caracter, de lungime maximă 100, care reprezintă numele studentului.

*data\_infiintarii* = variabilă de tip dată calendaristică, care reprezintă data naşterii studentului respectiv.

*cod\_facultate* = variabilă de tip intreg, de lungime maximă 5, care reprezintă codul facultatii de care parține asociația. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul FACULTATE.

* Relația STUDENT\_promovează\_EXAMEN (Entitatea *PROMOVEAZA)* are ca atribute:

*nota* = variabilă de tip real, de lungime maximă 2, care reprezintă nota studentului.

*cod\_student* = variabilă de tip intreg, de lungime maximă 5, care reprezintă codul studentului căruia îî aparține nota. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul STUDENT.

*cod\_examen* = variabilă de tip intreg, de lungime maximă 5, care reprezintă codul examenului la care a fost acordată nota. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul EXAMEN.

* Entitatea *EXAMEN* are ca atribute:

*cod\_examen* = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unui examen.

*forma* = variabilă de tip caracter, luând valorile *”Examen Scris”, ”Interviu” sau ”Proiect”*, care reprezintă modalitatea de evaluare.

*cod\_curs* = variabilă de tip intreg, de lungime maximă 5, care reprezintă codul cursului la care este susținut examenul. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul CURS.

* Entitatea *CURS* are ca atribute:

*cod\_curs* = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unui curs.

*denumire* = variabilă de tip caracter, de lungime maximă 50, care reprezintă denumirea cursului.

EXAMEN

Cod\_examen#

Forma

Cod\_curs

|  |  |  |  |
| --- | --- | --- | --- |
| Cod\_curs | | Denumire | |
| PK | NOT NULL | |

CURS

Cod\_curs#

Denumire

|  |  |  |  |
| --- | --- | --- | --- |
| Cod\_examen | Cod\_curs | | Forma |
| PK | | FK | NOT NULL |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cod\_profesor# | Nume | Prenume | Telefon | Mail |
| PK | NOT NULL | NOT NULL | NOT NULL | UNIQUE |

PROFESOR

Cod\_profesor#

Nume

Prenume

Telefon

Mail

Sala

Cod\_sala#

Denumire

Locatie

|  |  |  |  |
| --- | --- | --- | --- |
| Cod\_sala# | Locatie | Denumire | |
| PK | NOT NULL | NOT NULL |

* Entitatea *SALA* are ca atribute:

*cod\_sala* = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul salii.

*denumire* = variabilă de tip caracter, de lungime maximă 30, care reprezintă denumirea salii.

*locatie* = variabilă de tip caracter, de lungime maximă 50, care reprezintă locatia salii.

* Entitatea *PROFESOR* are ca atribute:

*cod\_profesor* = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unui profesor.

*nume* = variabilă de tip caracter, de lungime maximă 100, care reprezintă numele profesor.

*prenume* = variabilă de tip caracter, de lungime maximă 100, care reprezintă prenumele profesor.

*telefon* = variabilă de tip caracter, de lungime maximă 20, care reprezintă numarul de telefon.

*mail* = variabilă de tip caracter, de lungime maximă 50, care reprezintă emailul.

|  |  |  |  |
| --- | --- | --- | --- |
| Cod\_profesor# | Cod\_facultate# | Salariu | Data\_inceput |
| PK | | NOT NULL | NOT NULL |

CONTRACT

Cod\_profesor#

Cod\_facultate#

Salariu

Data\_inceput

|  |  |  |  |
| --- | --- | --- | --- |
| Cod\_profesor | Cod\_curs | Cod\_sala | Cod\_grupa |
| PK | | | |

* Relația *PROFESOR\_preda\_CURS\_la\_GRUPA\_in\_SALA (*Entitatea *ORAR)* are ca atribute:

*cod\_curs* = variabilă de tip intreg, de lungime maximă 5, care reprezintă codul cursului la care este susținut examenul. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul CURS.

*cod\_grupa* = variabilă de tip intreg, de lungime maximă 5, care reprezintă grupei care participa la curs. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul GRUPA.

*cod\_profesor* = variabilă de tip intreg, de lungime maximă 5, care reprezintă profesorului care predă cursul. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul PROFESOR.

*cod\_sala* = variabilă de tip intreg, de lungime maximă 5, care reprezintă codul sălii în care se desfășoară cursul. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul SALA. )

* Relația *PROFESOR\_este\_angajat\_la\_FACULTATE (*Entitatea *CONTRACT)* are ca atribute:

*salariu* = variabilă de tip întreg, care reprezintă salariul

*cod\_profesor* = variabilă de tip intreg, de lungime maximă 5, care reprezintă codul profesorului care preda la facultatea respectivă. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul PROFESOR.

*data\_inceput* = variabilă de tip dată calendaristică, care reprezintă data angajarii unui profesor la o facultate.

*cod\_facultate* = variabilă de tip intreg, de lungime maximă 5, care reprezintă facultății la care predă profesorul. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul FACULTATE.

ORAR

Cod\_profesor#

Cod\_curs#

Cod\_sala#

Cod\_grupa#

INSCRIERE

Cod\_grupa#

Cod\_student#

Data\_inscrierii

* Relația *GRUPA\_are\_STUDENȚI* (Entitatea *INSCRIERE)* are ca atribute:

*data\_inscriere* = variabilă de tip dată calendaristică, care reprezintă data înscrierii unui stundent în cadrul unei grupe.

*cod\_grupa*= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul grupei din care face parte studentul. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul GRUPA.

*cod\_student*= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul studentul. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul STUDENT.

|  |  |  |
| --- | --- | --- |
| Cod\_grupa# | Cod\_student# | Data\_inscrierii |
| PK | | NOT NULL |

**Diagrama Conceptuala – Ex. 7**

INSCRIERE

STUDENT

PROMOVEAZA

SALA

ORAR

PROFESOR

CONTRACT

CURS

EXAMEN

ASOCIATIE

GRUPA

SERIE

SECTIE

FACULTATE

****

****

****

****

****

****

****

****

****

****

****

****

****

**Diagrama E/R. – Ex. 6**

este\_angajat\_la

promoveaza

M(1)

M(1)

se\_desfasoara\_in

are

M(1)

are

are

Este\_membru\_in

este\_formata

este\_alcatuita

are

are

are\_loc

preda

M(1)

M(1)

M(1)

SALA

M(1)

1

1

1

M(1)

ASOCIATIE

1

M(1)

1

M(1)

M(1)

M(1)

M(1)

M(1)

STUDENT

EXAMEN

CURS

PROFESOR

GRUPA

SERIE

SECTIE

FACULTATE

**Schemele relaţionale - Ex. 8**

**Schemele relaţionale** corespunzătoare diagramei conceptuale sunt următoarele:

*ASOCIATIE(cod\_asociatie#, denumire,* *data\_infiintarii*,*cod\_facultate#)*

*CONTRACT(cod\_profesor#, cod\_facultate#,salariu)*

*CURS(cod\_curs#, denumire)*

*EXAMEN(cod\_examen#, forma,cod\_curs#)*

*FACULTATE(cod\_facultate#*, *denumire*, *adresă*, *telefon, mail*, *fax*, *cod\_postal )*

*GRUPA(cod\_grupa#, denumire, cod\_serie#)*

*INSCRIERE(data\_inscriere,* *cod\_grupa#*, *cod\_student#)*

*ORAR*(*cod\_curs#, cod\_grupa#,* *cod\_profesor#*, *cod\_sala#)*

*PROFESOR(cod\_profesor#, nume, prenume*, *data\_nastere, sex)*

*PROMOVEAZA(nota, data, cod\_student#*, *cod\_examen#)*

*SALA*( *cod\_sala#, denumire, locatie)*

*SECTIE* (*cod\_sectie#, denumire, cod\_facultate#)*

*SERIE* (*cod\_serie#,denumire, cod\_sectie#)*

*STUDENT(cod\_student#, nume, prenume, data\_nastere, sex, nationalitate,telefon,email, cod\_asociatie )*

**Normalizare și Denormalizare – Ex. 9 și Ex. 18**

* **Forma normală 1 (FN1)**

O relație se află în FN1 dacă fiecărui atribut care o compune îi corespunde o

valoare indivizabilă.

Forma normală 1 este și cea care impune și faptul că fiecare înregistrare să fie definită astfel încât să fie identificată unic prin intermediul unei chei primare.

În cadrul bazei de date pe care o construiesc, voi utiliza un exemplu pentru a prezenta aflarea acesteia in forma normal 1.

Voi lua entitățile FACULTATE și SECTIE.

În cadrul unei facultăți, denumirea secțiilor nu se repetă, dar în cadrul unei universități acest lucru nu este garantat, de multe ori chiar întâmplându-se acest lucru.

1. Mai multe valori seminifică același câmp

|  |  |
| --- | --- |
| FACULTATE | SECTIE |
| Biologie | Anatomie, Biologie Animala |
| Chimie | Anatomie, Chimie Organica |
| Medicina | Anatomie, Biologie Animala, Chimie Organica |

A selecta interegistrări pe baza câmpurilor care pot conține valori

semnificative sunt foarte dificile si greu de implementat. Spre exemplu, o interogare care ar selecta acele facultătți care au si secție de Anatomie si secție de Chimie Organică, ar trebui să parcurgem fiecare șir ”Secție”, să identificăm subșirurile Anatomie și Chimie Organică și să selectăm numai acele înregistrări în care apar ambele subșiruri.

|  |  |
| --- | --- |
| FACULTATE | SECTIE |
| Biologie | Anatomie, Biologie Animala |
| Chimie | Anatomie, Chimie Organica |
| Medicina | Anatomie, Biologie Animala, Chimie Organica |

|  |  |
| --- | --- |
| FACULTATE | SECTIE |
| Biologie | Anatomie |
| Biologie | Biologie Animala |
| Chimie | Chimie Organica |
| Chimie | Anatomie |
| Medicina | Anatomie |
| Medicina | Biologie Animala |
| Medicina | Chimie Organica |

|  |  |  |
| --- | --- | --- |
| ID | FACULTATE | SECTIE |
| 1 | Biologie | Anatomie |
| 2 | Biologie | Biologie Animala |
| 3 | Chimie | Chimie Organica |
| 4 | Chimie | Anatomie |
| 5 | Medicina | Anatomie |
| 6 | Medicina | Biologie Animala |
| 7 | Medicina | Chimie Organica |

2. Mai multe coloane reprezintă același tip de date.

|  |  |  |  |
| --- | --- | --- | --- |
| FACULTATE | SECTIE | SECTIE(1) | SECTIE(2) |
| Biologie | Anatomie | Biologie Animala |  |
| Chimie | Anatomie | Chimie Organica |  |
| Medicina | Anatomie | Biologie Animala | Chimie Organica |

|  |  |
| --- | --- |
| FACULTATE | SECTIE |
| Biologie | Anatomie |
| Chimie | Anatomie |
| Medicina | Anatomie |
| Biologie | Biologie Animala |
| Chimie | Chimie Organica |
| Medicina | Biologie Animala |
| Medicina | Chimie Organica |

|  |  |  |
| --- | --- | --- |
| ID | FACULTATE | SECTIE |
| 1 | Biologie | Anatomie |
| 2 | Biologie | Biologie Animala |
| 3 | Chimie | Chimie Organica |
| 4 | Chimie | Anatomie |
| 5 | Medicina | Anatomie |
| 6 | Medicina | Biologie Animala |
| 7 | Medicina | Chimie Organica |

Pentru a asigura unicitatea unei inregistrari, se va utiliza cheia primara. In exemplul de mai sus, prin introducerea unei coloane aditionale de tip intreg se asigura unicitatea fiecarei inregistrari.

* **Forma Normală 2 (FN2)**

O relație se află în a doua formă normal dacă și numai dacă această relație este deja in FN1 si fiecare atribut care nu este cheie primară este dependent de întreaga cheie primară.

FN2 interzice existența dependențelor funcționale parțiale în cadrul relației.

Daca unul sau mai multe elemente sunt dependente functional numai de o parte a cheii primare, atunci ele trebuie sa fie separate in tabele diferite. Daca tabela are o cheie primara formata din numai un atribut, atunci ea este automat in FN2.

Pentru baza mea de date, voi exemplifica pentru cazul diagramei CONTRACT.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cod\_profesor# | Nume Profesor | Cod\_facultate# | Data\_Inceput | Salariu |
| S1 | N1 | G1 | 12-04-2020 | 5400 |
| S1 | N1 | G2 | 01-10-2019 | 5300 |
| S2 | N2 | G2 | 03-02-2018 | 4600 |
| S2 | N2 | G3 | 05-06-2017 | 4900 |
| S2 | N2 | G4 | 01-10-2020 | 4500 |

Un profesor poate să fie predea la mai multe facultăți și o facultate are mai mulți profesori angajați.

Relația este în FN1 – avem identificator unic pentru toate intrările din table.

Fiecare atribut care nu este cheie (nu participă la cheia primară) este dependent de întreaga cheie primară – in cazul nostru atributul Nume\_profesor, nu este cheie si trebuie sa depinda direct de intreaga cheie primara cod\_profesor# si cod\_facultate# -> aceste atribute nu depind direct de intreaga cheie primara deoarece se observa dependenta directa dintre Nume\_profesor si cod\_profesor, insemnand ca Nume\_profesor depinde direct doar de o parte a cheii primare, si anume doar de cod\_profesor -> relatia nu se afla in FN2.

Astfel avem ca:

* {cod\_profesor#} -> {nume\_profesor} – cod\_profesor determina functional cod\_profesor
* {cod\_profesor #, cod\_facultate#} -> {data\_inceput, salariu}

Se aplică regula Casey Delobel și va rezulta faptul că, pentru a avea relația în FN2, numele profesorului trebuie să fie doar în entitatea profesor.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cod\_profesor# | Nume Profesor | Cod\_facultate# | Data\_Inceput | Salariu |
| S1 | N1 | G1 | 12-04-2020 | 5400 |
| S1 | N1 | G2 | 01-10-2019 | 5300 |
| S2 | N2 | G2 | 03-02-2018 | 4600 |
| S2 | N2 | G3 | 05-06-2017 | 4900 |
| S2 | N2 | G4 | 01-10-2020 | 4500 |

|  |  |  |  |
| --- | --- | --- | --- |
| Cod\_profesor# | Cod\_facultate# | Data\_Inceput | Salariu |
| S1 | G1 | 12-04-2020 | 5400 |
| S1 | G2 | 01-10-2019 | 5300 |
| S2 | G2 | 03-02-2018 | 4600 |
| S2 | G3 | 05-06-2017 | 4900 |
| S2 | G4 | 01-10-2020 | 4500 |

* **Forma Normală 3 (FN3)**

O relație este în a treia formă normală dacă și numai dacă este in FN2 și fiecare atribut care nu este cheie depinde direct de cheia primară.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Cod\_student# | nume | prenume | Data\_nasterii | sex | Nationalitate | mail | Denumire\_asociatie | Data\_infiintarii |
| P1 | N1 | Pr1 | 01-02-2000 | m | Roman | M1 | G1 | D1 |
| P2 | N2 | Pr2 | 17-01-2000 | f | Roman | M2 | G2 | D2 |
| P3 | N3 | Pr3 | 22-08-2001 | f | Roman | M3 | G3 | D3 |

Iau tabela STUDENT. Inițial, aceasta ar fi fost de forma:

Se poate observa că atributelul Data\_infiiintarii depinde de atributul Denumire\_asociatie care depinde la randul sau de cheia primară cod\_student, astfel nefiind in NF3.

Pentru a aduce in FN3, separ atributele despre asocitatie din STUDENT, apărând astfel tabela ASOCIATIE.

Astfel, inlocuiesc aceste atribute cu cheia străină cod\_asociatie (apărând astfel o relație one-to-many între asociație si student) pentru a determina mai ușor din ce asociație face un student parte sau dacă nu face voluntariat (cheia străină e nula).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Cod\_student# | nume | prenume | Data\_nasterii | sex | Nationalitate | Cod\_asociatie | mail |
| P1 | N1 | Pr1 | 01-02-2000 | m | Roman | Null | M1 |
| P2 | N2 | Pr2 | 17-01-2000 | f | Roman | A1 | M2 |
| P3 | N3 | Pr3 | 22-08-2001 | f | Roman | A2 | M3 |

|  |  |  |  |
| --- | --- | --- | --- |
| Cod\_asoctiate | Denumire\_asociatie | Data\_infiintarii | cod\_facultate |
| 1 | Asoc1 | D1 | F1 |
| 2 | Asoc2 | D2 | F2 |
| 3 | Asoc3 | D3 | F3 |

* **Forma normală Boyce-Codd (BCNF)**

Forma normală Boyce-Codd se bazează pe dependenţele funcţionale care iau

în consideraţie toate cheile candidat dintr-o relaţie.

Pentru relatiile cu o singura cheie candidat, formele FN3 si BCNF sunt echivalente (Facultate, Curs etc.)

Să luăm relația PROMOVEAZA(cod\_examen#, cod\_student#, nota)

Cod\_Student

Cod\_Examen

Nota

Regula Casey Delobel pentru PROMOVEAZA(cod\_examen#, cod\_student#, nota) având faptul că nota->cod\_student

PROMOVEAZA\_1( cod\_examen#,nota)

PROMOVEAZA\_2(nota,cod\_student)

* **Forma nomală 4 (FN4)**

FN4 elimină redundanțele datorate relațiilor m:n, adică datorate dependenței multiple.

O relație este în a patra formă nomală dacă și numai dacă este în BCNF și nu conține relații m:n independente.

Iau relația SALA(cod\_sala#, denumire, locatie) li presupun că o sală poate avea mai multe denumiri și mai multe locații – maă gândesc ca la un amfiteatru în care se poate intra prin mai multe locuri si fiecare intrare e numele salii + Intrarea (A, B, C, etc.) si locatiile sunt date de holurile din care se intra în acestea.

Cod\_sala# denumire;

Cod\_sala# locatie;

Relația Sala este in BCNF. Pentru a aduce relația în FN4 o vom descompune prin proiecție în două relații:

SALA1(cod\_sala#, denumire)

SALA2(cod\_sala#, locatie)

SALA = JOIN(SALA1, SALA2)

* **Forma normală 5 (FN5)**

O relaţie R este în FN5 (numită şi forma normală proiecţie-uniune) dacă şi numai dacă orice dependenţă de uniune a lui R este o consecinţă a unei chei candidat a lui R.

Orice relaţie care este în FN5 este şi în FN4, deoarece fiecare dependenţă multivaloare poate fi privită ca un caz particular de dependenţă de uniune. Orice relaţie poate fi descompusă fără pierderi la uniune într-o mulţime de relaţii care sunt în FN5.

Pentru a preciza dacă o relaţie este în FN5, este suficient să cunoaştem cheile candidate şi toate dependenţele de uniune din R.

Aducerea în FN5 presupune eliminarea join dependențelor.

Sa luăm relația FACULTATE și să presupunem că am avea o join dependență în aceasta – iau mulțimea (fax, adresa, cod\_postal) prespunem ca există multiple dependențe între fiecare dintre perechile din mulțimi (fax,adresa), (adresa,cod\_postal) și (fax, cod\_postal) – se pierde faptul că în acest caz cod\_postal se duce in fax - este un exemplu pur ipotetic.

* **Denormalizare**

Denormalizarea este procesul invers al procesului de normalizare. Denormalizarea funcționează adăugând date redundante sau grupând date pentru a optimiza performanța.

Denormalizare are rolul de a realiza executarea mai rapidă a interpgărilor prin introducerea redundanței, punând accentul pe rapiditatea analizei și scăzând numărul de tabele.

Motivul efectuării denormalizării este **costurile unui** produs în procesor de interogare printr-o structură supra-normalizată.

Denormalizarea poate fi definită și ca metoda de stocare a îmbinării relațiilor de formă normală superioară ca relație de bază, care se află într-o formă normală inferioară. Reduce numărul de tabele și îmbinările de tabel complicate, deoarece un număr mai mare de îmbinări poate încetini procesul.

Aici abordarea denormalizării, subliniază conceptul că, plasând toate datele într-un singur loc, ar putea elimina necesitatea căutării acelor fișiere multiple pentru a colecta aceste date.

În cadrul bazei mele de date, luând spre exemplu relația PROMOVEAZA și descompunerea ei de la FN4, este inutil și mult mai costisitor din punct de vedere al timpului de executare să parcurgem datele si din PROMOVEAZA1 și PROMOVEAZA2.

**Algebră Relațională – Ex. 17**

* Sa se afiseze numele, prenumele si salariul profesorilor care lucreaza la facultatea cu codul 1, au salariu mai mare de 5000 de lei si au fost angajati dupa 1 august 2017.

SQL:

select p.nume, p.prenume, c.salariu

from profesor p join contract c on (p.cod\_profesor = c.cod\_profesor)

where c.cod\_facultate in (select cod\_facultate

from contract t

where t.cod\_facultate = c.cod\_facultate and t.cod\_facultate = 1)

and c.salariu in (select salariu

from contract ct

where ct.salariu >= 5000 and c.cod\_profesor = ct.cod\_profesor)

and data\_inceput in (select data\_inceput

from contract cc

where cc.data\_inceput > to\_date('01-08-2017','dd-mm-yyyy') and c.cod\_profesor = cc.cod\_profesor)

;

Expresie Algebrică:

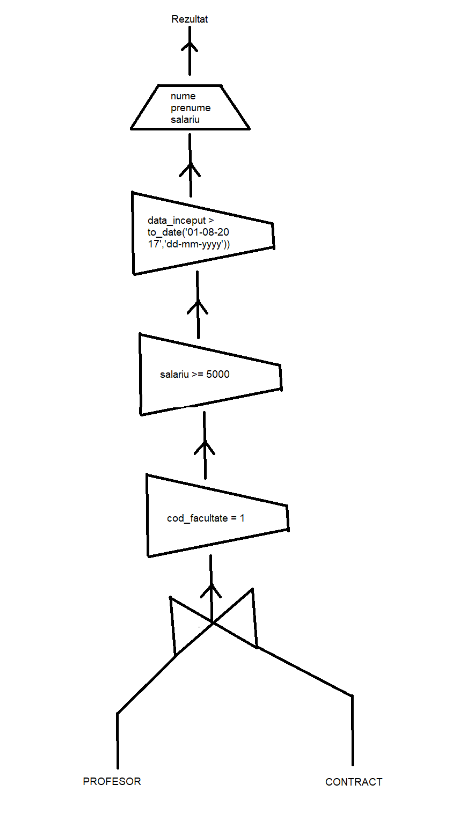
R1 = JOIN(profesor, contract)

R2 = SELECT(R1, cod\_facultate = 1)

R3 = SELECT(R2, salariu >= 5000)

R4 = SELECT(R3, data\_inceput > to\_date('01-08-2017','dd-mm-yyyy'))

REZULTAT = R5 = PROJECT(R4, nume, prenume,salariu)



Etapa Intermediara:

SQL:

select p.nume, p.prenume, c.salariu

from profesor p join contract c on (p.cod\_profesor = c.cod\_profesor)

where c.cod\_facultate = 1 and c.salariu >= 5000 and c.data\_inceput > to\_date('01-08-2017','dd-mm-yyyy');

Expresie Algebrică:

R1 = SELECT(contract, cod\_facultate = 1 and salariu >= 5000 and data\_inceput > to\_date('01-08-2017','dd-mm-yyyy'))

R2 = JOIN(R1, profesor)

R3 = PROJECT(R2, nume, prenume, salariu)

Rezultat final:

SQL:

select p.nume, p.prenume, c.salariu

from (select salariu, cod\_profesor

from contract

where cod\_facultate = 1 and salariu >= 5000 and data\_inceput > to\_date('01-08-2017','dd-mm-yyyy') ) c

join(select nume,prenume, cod\_profesor

from profesor ) p on (p.cod\_profesor = c.cod\_profesor);

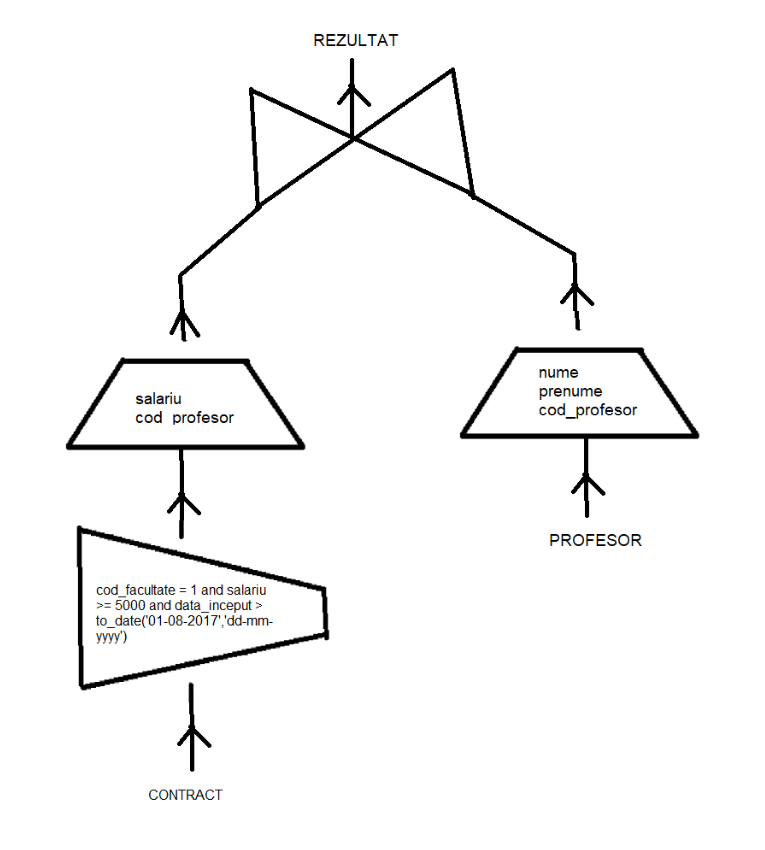
Expresie Algebrică:

R1 = SELECT(contract, cod\_facultate = 1 and salariu >= 5000 and data\_inceput > to\_date('01-08-2017','dd-mm-yyyy'))

R2 = PROJECT(R1, salariu, cod\_profesor)

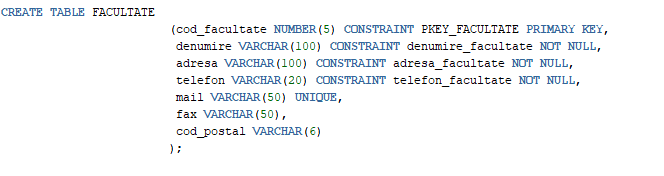
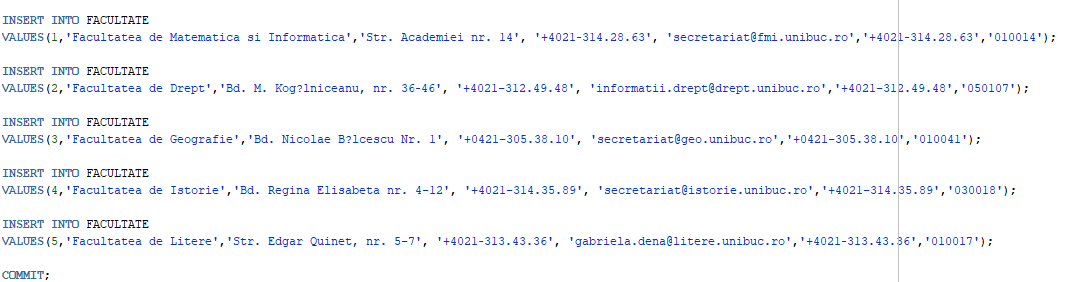
R3 = PROJECT(profesor, nume, prenume, cod\_profesor)

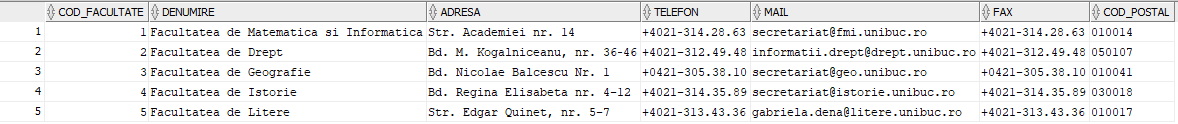
REZULTAT = R4 = JOIN(R2, R3)

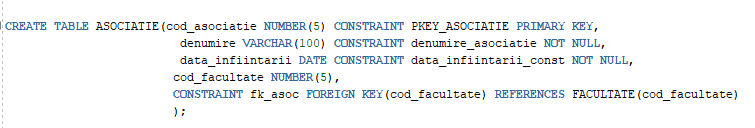


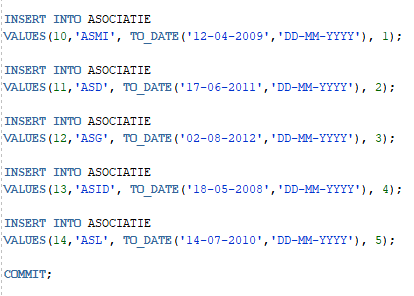
**Ex. 10**

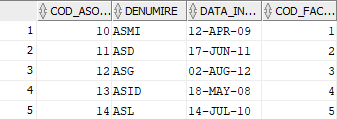
**SQL**

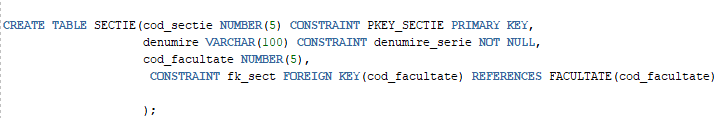


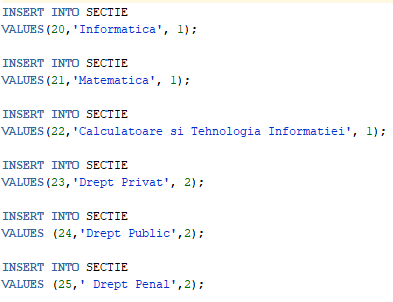




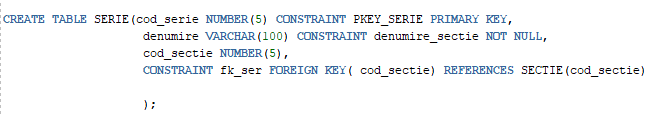


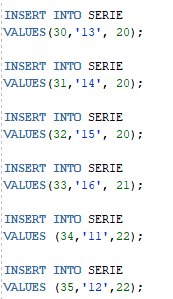


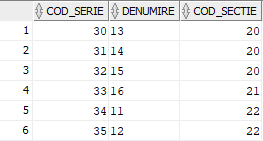


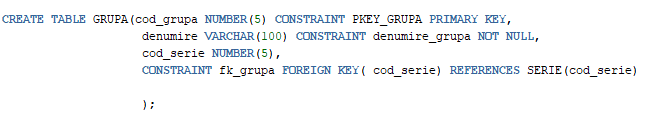


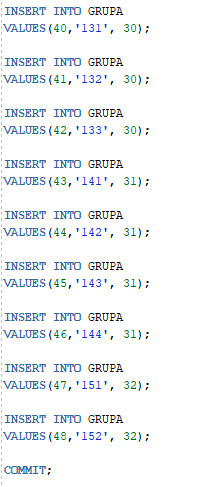


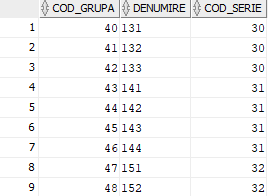


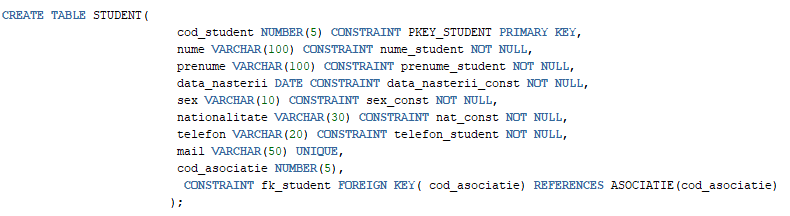




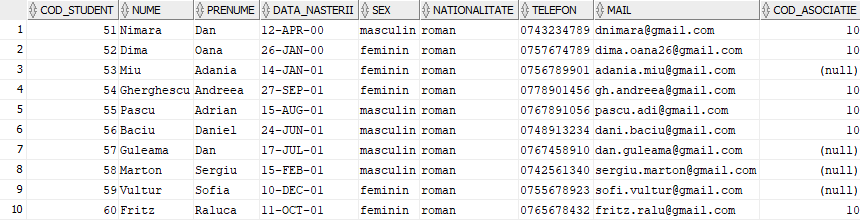


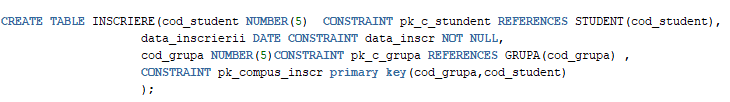


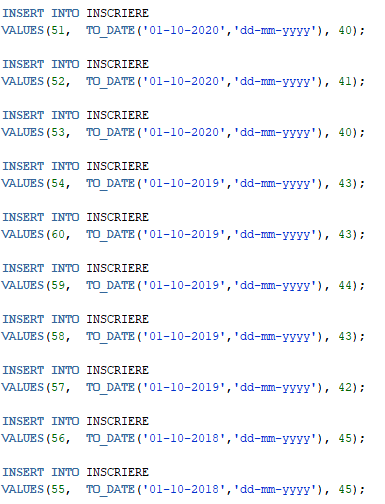


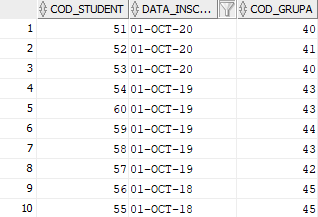


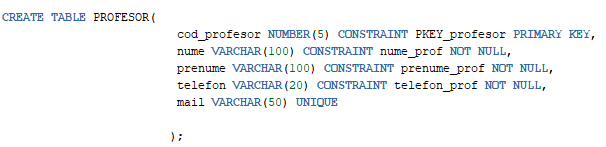






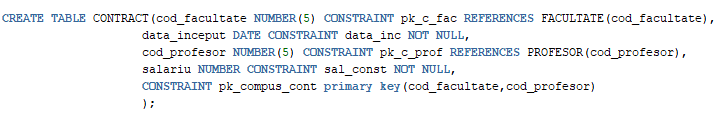


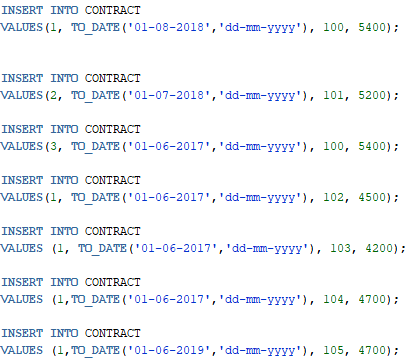


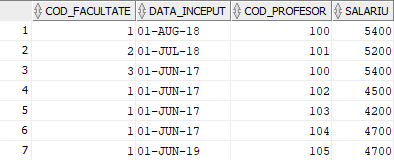


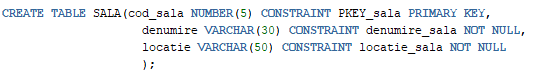




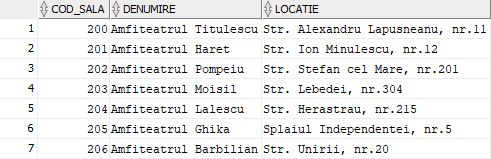




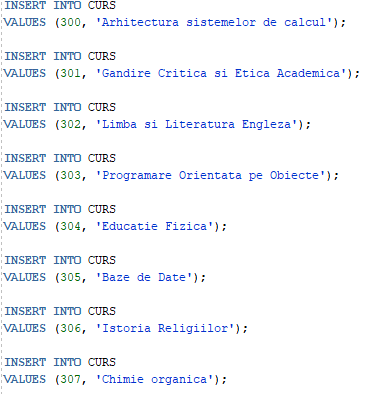


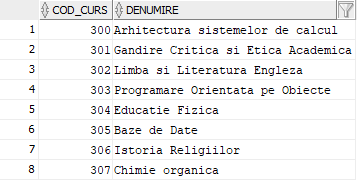


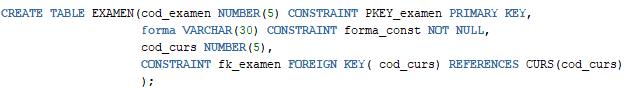


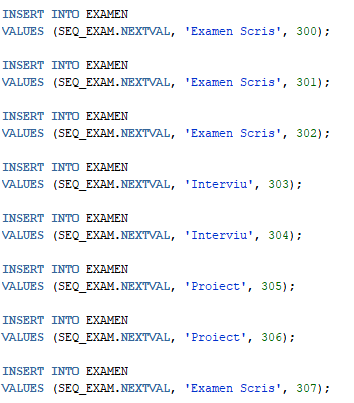


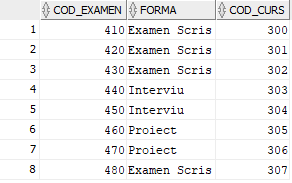




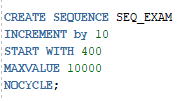


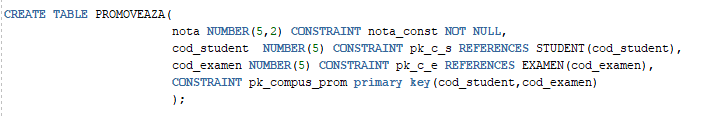


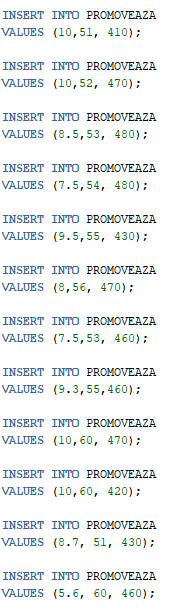


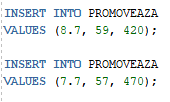


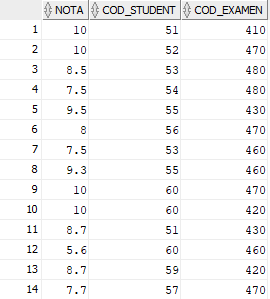
**Ex. 13**

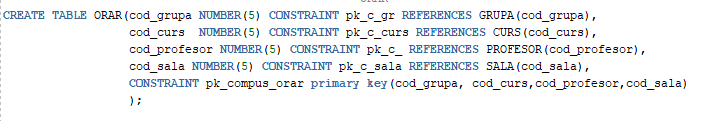


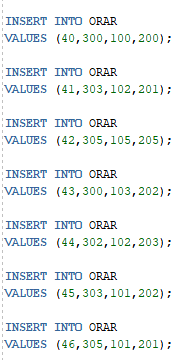


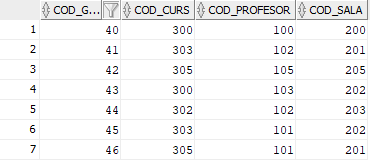




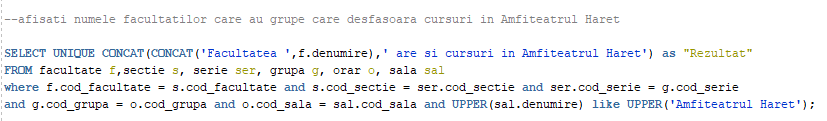




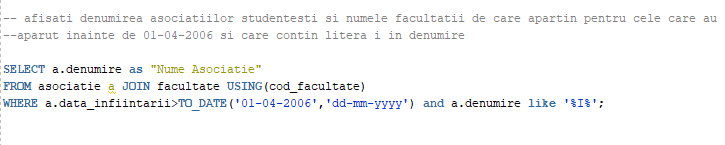


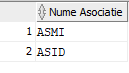


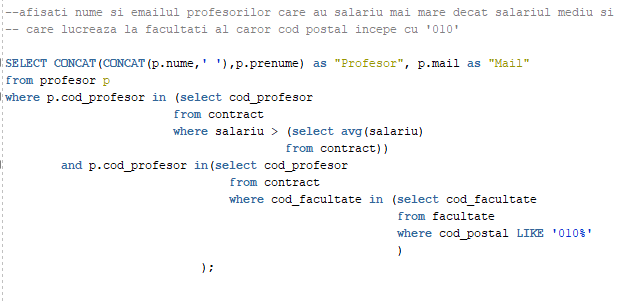
**Ex. 11**



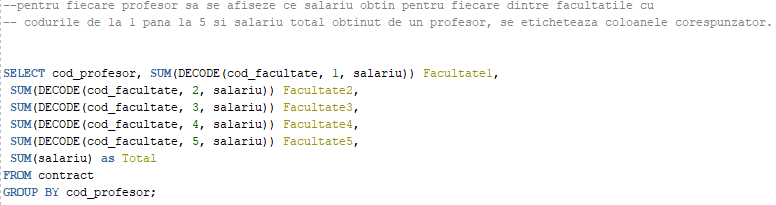


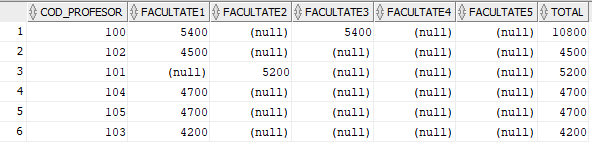


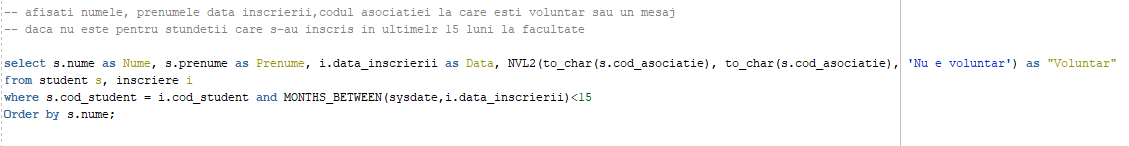


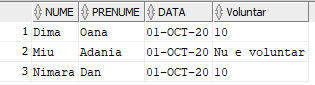


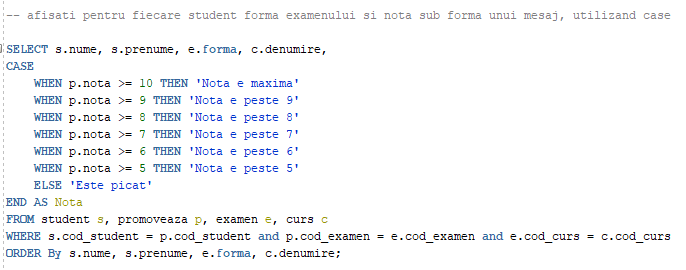


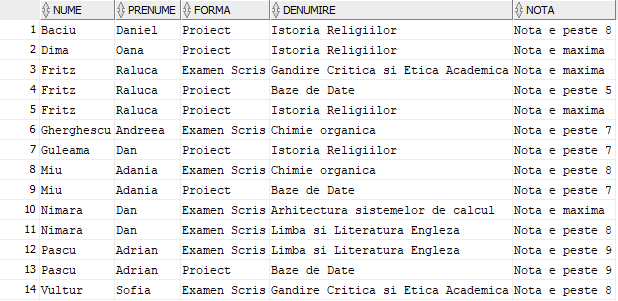




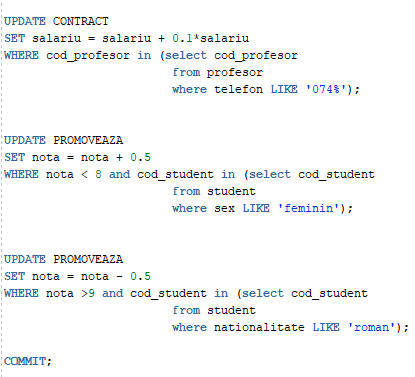








**Ex. 12**

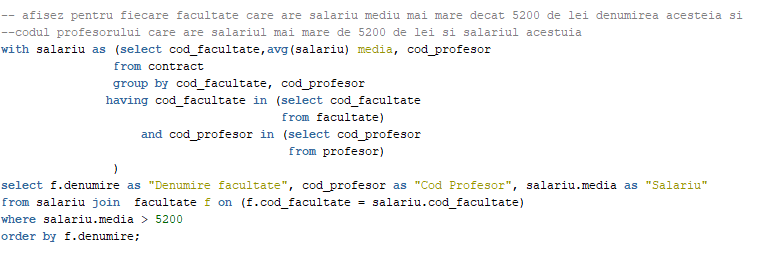


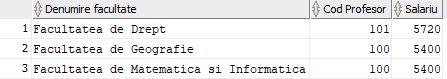
--1. Sa se mareasca salariul cu 10% pentru profesorii care au numarul de telefon incepand cu ”074”

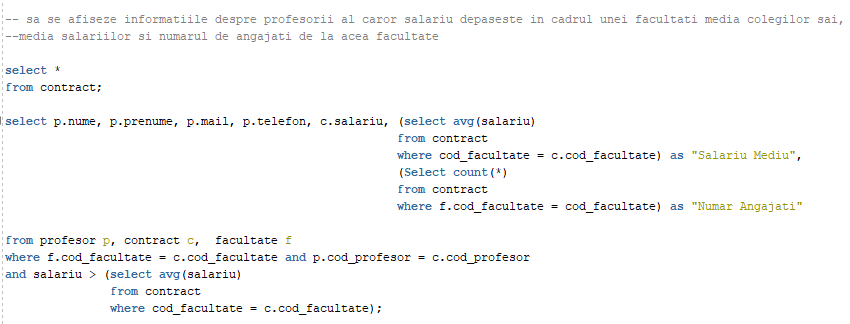
--2. Sa se mareasca notele cu 0.5 pentru studentii care au nota mai mica decat 8 si sunt fete - marire de 1 martie

--3. Sa se scada nota studentilor care au nota mai mare decat 9 si sunt romani

**Ex. 11 - continuare**

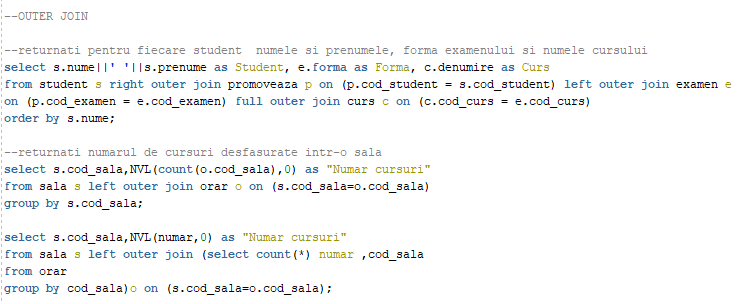


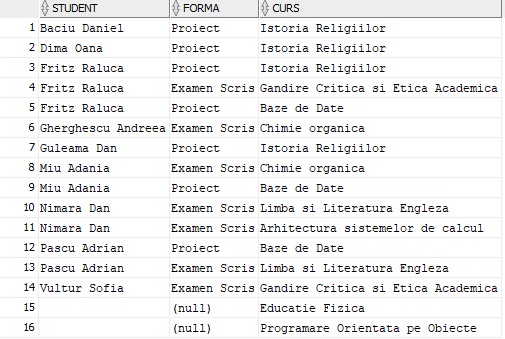


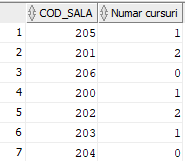


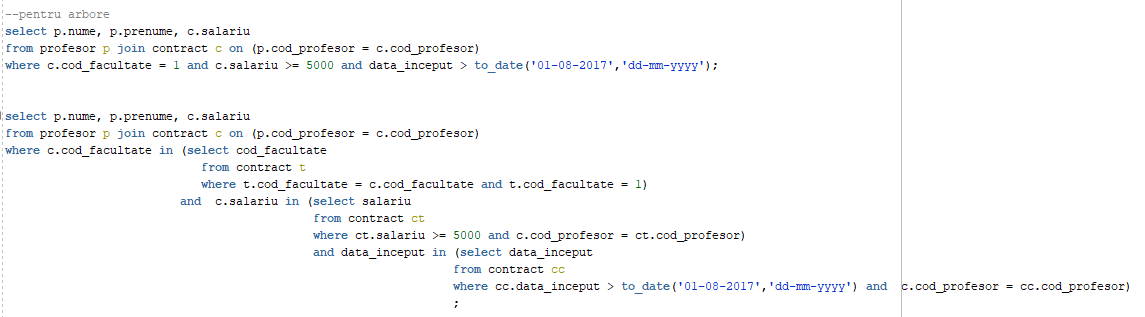


**Ex. 16**

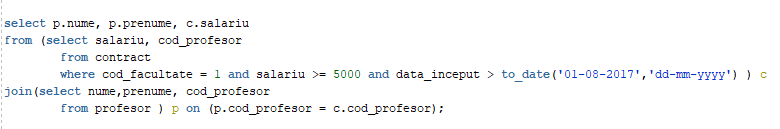








**Ex. 17 - cod**





**Ex. 16 - division**

