

Proiectare bază de date

Stabilirea entităților și a atributelor pentru acestea:

Facultati -> idFacultate, numeFacultate, adresaFacultate, webFacultate, telefonFacultate,

DomeniiStudiu -> idDomeniu, domeniuStudiu, webDomeniuStudiu

ProgrameStudiu -> idProgramStudiu, programStudiu, tipConcursPrgStudiu, nrLocuriPrgStudiu, planInvPrgStudiu, cicluPrgStudiu

Studenti -> nrMatricolStudent, numeStudent, emailStudent, adresaStudent, telStudent

Tutori -> numeTutor, emailTutor, adresaTutor, gradTutor

Cursuri -> idCurs, denumireCurs, nrCrediteCurs, tipEvaluare, anStudiuCurs

Stabilirea relațiilor inițiale:

CursuriStudenti(#idCurs, denumireCurs, nrCrediteCurs, tipEvaluare, anStudiuCurs, #nrMatricolStudent, numeStudent, emailStudent, adresaStudent, telStudent, numeTutor, emailTutor, adresaTutor, gradTutor, programStudiu)

DomeniiFacultate(idFacultate, numeFacultate, adresaFacultate, webFacultate, telefonFacultate, #idDomeniu, domeniuStudiu, webDomeniuStudiu)

ProgrameStudiuDomeniu(#idProgramStudiu,programStudiu,nrLocuriPrgStudiu, tipConcursPrgStudiu, planInvPrgStudiu, cicluPrgStudiu, idDomeniu, domeniuStudiu, webDomeniuStudiu)

Normalizarea relațiilor inițiale:

Se consideră că un student nu poate fi înscris simultan la mai multe programe de studiu!

Relația **CursuriStudenti**:

DF:

idCurs -> denumireCurs, nrCrediteCurs, tipEvaluare, anStudiuCurs

nrMatricolStudent -> numeStudent, emailStudent, adresaStudent, telStudent, programStudiu

emailTutor -> numeTutor, adresaTutor, gradTutor

idCurs, nrMatricolStudent -> emailTutor, numeTutor, adresaTutor, gradTutor

PK: (idCurs, nrMatricolStudent)

1NF este respectată deoarece avem doar valori atomice și fiecare atribut ia o singură valoare din domeniul său de definiție, un student nu poate fi înscris de mai multe ori la același curs al unui tutor

2NF nu este respectată deoarece avem cheie primară compusă și dependențe parțiale.

Prin aplicarea 2NF am obținut următoarele relații:

Cursuri(#idCurs, denumireCurs, nrCrediteCurs, tipEvaluare, anStudiuCurs)

Studenti(#nrMatricolStudent, numeStudent, emailStudent, adresaStudent, telStudent, programStudiu)

CursuriStudenti(#idCurs, #nrMatricolStudent, emailTutor, numeTutor, adresaTutor, gradTutor), idCurs(FK), nrMatricolStudent(FK)

3NF nu este respectată în relația CursuriStudenti deoarece există o dependență tranzitivă determinată de atributul emailTutor. Relațiile Cursuri și Studenti respectă 3NF.

Prin aplicarea 3NF am obținut următoarele relații:

Tutori(#emailTutor, numeTutor, adresaTutor, gradTutor)

CursuriStudenti(#idCurs, #nrMatricolStudent, emailTutor)

idCurs(FK), nrMatricolStud(FK), emailTutor(FK)

Relație DomeniiFacultate:

DF:

idFacultate -> numeFacultate, adresaFacultate, webFacultate, telefonFacultate

idDomeniu -> domeniuStudiu, webDomeniuStudiu, idFacultate

PK: idDomeniu

1NF respectată deoarece avem doar valori atomice și fiecare atribut ia o singură valoare din domeniul său de definiție, un domeniu de studiu este specific unei singure facultăți

2NF este respectată deoarece cheia primară este formată dintr-un singur atribut

3NF nu este respectată deoarece avem o dependență tranzitivă determinată de idFacultate.

Prin aplicarea 3NF am obținut următoarele relații:

Facultati(#idFacultate, numeFacultate, adresaFacultate, webFacultate, telefonFacultate)

Domenii(#idDomeniu, domeniuStudiu, webDomeniuStudiu, idFacultate) idFacultate(FK)

ProgrameStudiuDomeniu:

DF:

idProgramStudiu->programStudiu, tipConcursPrgStudiu, planInvPrgStudiu nrLocuriPrgStudiu, cicluStudiu, idDomeniu

idDomeniu -> webDomeniuStudiu, domeniuStudiu

PK: idProgramStudiu

1NF respectată deoarece avem doar valori atomice și fiecare atribut ia o singură valoare din domeniul său de definiție, un program de studiu este specific unui singur domeniu de studiu

2NF este respectată deoarece cheia primară este formată dintr-un singur atribut

3NF nu este respectată deoarece avem o dependență tranzitivă determinată de idDomeniu

Prin aplicarea 3NF am obținut următoarele relații:

ProgrameStudiu(#idProgramStudiu, programStudiu, tipConcursProgramStudiu, planInvPrgStudiu, nrLocuriPrgStudiu, cicluStudiu, idDomeniu), idDomeniu(FK)

DomeniiStudiu(#idDomeniu, domeniuStudiu, webDomeniuStudiu)

Relații obținute prin normalizare:

Tutori(#emailTutor, numeTutor, adresaTutor, gradTutor)

CursuriStudenti(#idCurs, #nrMatricolStudent, emailTutor)

Cursuri(#idCurs, denumireCurs, nrCrediteCurs, tipEvaluare, anStudiuCurs)

Studenti(#nrMatricolStudent, numeStudent, emailStudent, adresaStudent, telStudent, programStudiu)

Facultati(#idFacultate, numeFacultate, adresaFacultate, webFacultate, telefonFacultate)

ProgrameStudiu(#idProgramStudiu, programStudiu, tipConcursPrgStudiu, planInvPrgStudiu, nrLocuriPrgStudiu, cicluStudiu, idDomeniu)

Domenii(#idDomeniu, domeniuStudiu, webDomeniuStudiu, idFacultate)

DomeniiStudiu(#idDomeniu, domeniuStudiu, webDomeniuStudiu)

Relația DomeniiStudiu este deja inclusă în relația Domenii => Relația DomeniiStudiu nu se mai ia în considerare.