

8. Relatsiooniline mudel

Relatsiooniline andmemudeli esitas 1970 aastal Edgar Frank "Ted" Codd.

Coddi andmemudeli kandvaks alustalaks on lihtne mõiste – relatsioon (*relation*). Relatsiooniks nimetas Codd vektorit, mis koosneb atribuutidest. Iga relatsioon kirjeldab ära ühe andmekoosluse - atribuutidele vastavate väärtuste jadade kogumi.

Relatsiooniline mudel kujutab endast kogumit relatsioone.

Et asja kuidagi "maisemalt" kirjeldada kaldume natuke E. F. Codd'i relatsioonilisest mudelist kõrvale ja vaatame Coddi poolt defineeritud mõisteid võrrelduna andmebaasidega seotud mõistetega - nii on kiirem ja lihtsam selgitada Coddi poolt välja pakutud mõistete süsteemi.

Coddi relatsioonilise mudeli terminoloogias on andmebaasi mõiste "tabeli" vasteks mõiste "relatsioon" (*relation*), mõiste "tabeli veerg" vasteks on mõiste "relatsiooni atribuut" (*attribute*), "tabeli rea" vasteks on "väärtuste vektor" (*tuple*) ja tabeli kirjeldust "relatsiooni skeemiks" (*relation schema*). Atribuudiga seotud andmetüübi kirjeldust nimetatakse domeeniks (*domain*).

Need vasted ei ole küll 100%-lised kuid sarnasus on silmaga märgatav. Coddi relatsioonilise mudeli mõistete ja nende abil kirjeldatud mudeli täpsem lahti mõtestus on esitatud järgnevatel lehekülgedel.

8.1. Domeen

Domeen on kogum elementaarseid (atomaarseid) väärtusi. Atomaarseid tähendab siin seda, et neid ei jagata ega defineerita enam madalama taseme mõistetena. See väärtuste kogumit kasutatakse atribuudis esineda võivata andmete piiritlemiseks. Domeen on midagi rohkemat kui on andmetüüp või andmete esitamise formaat aga võib olla ka just ainult see (või mõlemad koos).

Domeen on just nii kompleksne, kui seda vajatakse. Projekteerimise esimestel etappidel on domeeni kirjeldus hägusam ja lakoonilisem koosnedes peamiselt vabatekstilisest semantika kirjeldusest. Projekteerimise lõpus on domeenid defineeritud väga täpselt.

Edgar Frank Codd

relatsioon

tabel ~ relatsioon
veerg ~ atribuut
domeen ~ veeru tüüp
tabeli kirjeldus ~
relatsiooni skeem

domeen

domeeni täpsus

Domeeni abil defineeritakse kõik atribuudi omadused välja arvatud atribuudi detailne semantiline kirjeldus. Domeeni semantika ei ole atribuudi semantika. Domeeni semantika on atribuudi semantika kõige üldisemal tasemel ilma atribuudi spetsiifiliste täpsustusteta, mis tavaliselt kirjeldavad atribuudi käsitlemise reegleid.

Enamikel juhtudel on domeeni detailne struktuur järgmine:

- elementaarne andmetüüp (char, integer, decimal jms.)
- maksimaalne pikkus (sümbolite arv)
- kümnendkohtade arv (murdarvudel)
- formaat (N: dd.MM.yyyy ; ###.##0,00)
- NULL / NOT NULL tingimus
- miinimum ja maksimum väärtus
- vaikumisi väärtus
- võimalike väärtuste loend
- mõned keerulisemad kontrolli algoritmid
- vabatekstiline semantika

Domeen võib loogilisel tasemel olla kirjeldatud täiesti vabas vormis, näiteks:

Eestis kehtiv isikukood

lokaalne, 7-kohaline telefoninumber

mistahes riigi isikukood

Inimese eesnimi, kui inimesel on mitu eesnime, siis kõik eesnimed üle tühiku.

ERD CASE-süsteemides antakse selleks, et neid omavahel eristada, domeenidele nimed. Füüsilises mõttes tähendab domeen kasutaja poolt komplekselt kirjeldatud andmetüüpi. Andmetüübi tähiseks on domeeni nimi.

domeeni füüsiline struktuur

andmetüüp

pikkus

kümnendkohtade arv

formaat

NULL / NOT NULL

miinimum, maksimum

vaikumisi

väärtuste loend

eri-piirangud

Semantika

NÄIDE

domeen => kasutaja poolt defineeritud andmetüüp.

8.2. Relatsioon ja relatsiooni atribuudid. Relatsiooni järk.

Relatsiooni (relatsiooni skeemi) R tähistatakse:

$$R (A_1, A_2, A_3, \dots, A_n)$$

kus R on relatsiooni nimi ja A_1, \dots, A_n tähistavad selle relatsiooni järjestatud atribuute. Iga atribuut A_i on semantiline nimetus, mis tähistab mingi domeeni D ilmumist antud **kohas** ja antud **rollis**. Domeeni, mis määrab atribuudi A tähistatakse **dom(A_i)**. Relatsiooni järguks (*degree*) nimetatakse atribuutide arvu relatsioonis

Näide:

$$AUTO(\text{mark, mudel, väljalaskeaasta, mootorinumber})$$

Siin: AUTO - on relatsiooni nim

mark, mudel, väljalaskeaasta, ja mootorinumber on relatsiooni atribuudid

relatsiooni järk on 4 (relatsioonil on neli atribuuti)

dom(mark) – automarkide loend

dom(mudel) – auto mudelite loend

dom(väljalaskeaasta) – arv, mis on suurem kui 1850 ja väiksem või võrdne kui jooksev aasta

dom(mootorinumber) – vaba tekst pikkusega 10 kuni 40 sümbolit

relatsiooni tähistus

domeeni tähistus
relatsiooni järk

NÄIDE

8.3. Relatsiooni ilming, vektor (*tuple*)

Relatsiooni ilmingut relatsiooni skeemil R tähistatakse $r(R)$.

Relatsiooni ilming on kogum vektoreid:

$$\mathbf{r} = \{ \mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \dots, \mathbf{r}_m \}$$

Iga vektori järk on sama, mis relatsiooni skeemil (n). Seega on relatsiooni ilming kogum N-vektoreid. Iga vektor on järjestatud kogum N-väärtuseid:

$$\mathbf{r}_i = \{ \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \dots, \mathbf{v}_n \}$$

kus iga väärtus v_i , $1 \leq i \leq n$ on domeeni $\text{dom}(A_i)$ element või on tegemist tühja väärtusega.

relatsiooni ilming

vektor kui järjestatud väärtuste jada

vektoris olev väärtus on piiratud domeeniga või on see tühi väärtus

Relatsiooni skeem on sisuliselt relatsiooni kirjeldus, relatsiooni ilming aga relatsiooni kirjelduse alusel loodud kogum andmeid.

relatsiooni skeem vs. relatsiooni ilming

Relatsiooni ilmingut mingil konkreetsel ajahetkel nimetatakse relatsiooni jooksvaks staatuseks (*current relation state*). Relatsiooni jooksev staatus muutub siis kui, muutub mõni relatsiooni vektoritest, relatsiooni lisandub mõni vektor või kaob sealt mõni vektor.

relatsiooni jooksev staatus

Oluline on aru saada, et vektoris olevad väärtused on järjestatud alati samas järjestuses nagu on järjestatud relatsiooni skeemis atribuudid - atribuut määrab temaga samas positsioonis oleva väärtuse semantika ja füüsilised parameetrid (domeeni).

samas positsioonis olev atribuut kirjeldab samas positsioonis oleva väärtuse tähenduse

Kui me räägime relatsioonist, siis viitame me tegelikult mõlemale kontseptile – relatsiooni skeemile ja relatsiooni ilmingule so. vektorite kogumile.

relatsioon = relatsiooni skeem + relatsiooni ilming

8.4. Relatsiooni karakteristikud ja relatsioonilase mudeli piirangus

Relatsiooni olulisi karakteristikuid on kolm:

vektorite järjestus relatsioon on defineeritud kui vektorite kogum. Nendel vektoritel ei ole järjestust. Tegelikult on küll järjestus olemas, kuid see ei ole kunagi teada ja see pole ka oluline. Siiski ilmuvad vektorid nende lugemisel mingis konkreetses järjestuses. See on tingitud andmebaaside füüsilisest ehitusest. Loogilisel tasandil vektorite järjestust relatsioonis ei eksisteeri.

vektoritel ei ole järjestust

väärtuste järjestus vektoris väärtuste järjestus vektoris on väga oluline. Oluline on et relatsiooni ilmingu kõigis vektorites oleksid samatähenduslikud väärtused samas järjekorras. Oluline ei ole seega see järjekord ise vaid see, et kõikides vektorites oleks väärtused samas järjestuses – järjestuses mis on määratud relatsiooni skeemi atribuutide järjestusega.

väärtuste järjestus ei ole suvaline

väärtuste atomaarsus iga vektoris olev väärtus on atomaarse iseloomuga s.t. relatsioonilise mudeli raames jagamatu. Ei ole lubatud korduvad väärtused st. mudel peab olema viidud vähemalt esimesele normaalkujule – korduvate väärtuste grupid kirjeldatakse eraldi relatsioonis. Kõik väärtused on vektoris

väärtused on kordumatud ja atomaarsed tühja väärtust tähistatakse konstandiga NULL

alati olemas. Kui väärtus mingis positsioonis ei ole teada, siis seisab selle koha peal universaalne ilma tüübita väärtus NULL, mis tähistab tühja väärtust

Relatsioonilise mudeli olulisi piiranguid on kaks:

Domeeni piirang kõik domeeni kaudu defineeritud atribuudid on atomaarsed ja kõik nende väärtused läbi erinevate vektorite on sama tüüpi, formaati, pikkusega jne. (Mõelge tabeli peale, kus igas veerus on ainult samasugused andmed: eesnimi, perekonnanimi, sünniaeg, aadress jne.)

Võtme piirang iga vektor relatsioonis peab olema selgelt eristatav: sama relatsiooni piires ei saa olla vektoreid, mille kõik väärtused on samad. Minimaalset atribuutide kombinatsiooni, mille väärtused võimaldavad vektoreid üksteisest eristada nimetatakse supervõtmeks (super key).

Samas positsioonis asuvad vektorid on sama tüüpi

ei ole vektoreid, mille kõik väärtused langevad kokku mõne teise vektori kõigi väärtustega lühim unikaalne väärtuste kombinatsioon on Super Key

8.5. Relatsiooniline andmebaas

Relatsioonilise andmebaasi skeem on kogum relatsiooni skeeme:

$$S = \{ R_1, R_2, R_3, \dots, R_m \}$$

ja kogum terviklikkuse piiranguid (integrity constraints):

$$IC = \{ IC_1, IC_2, IC_3, \dots, IC_k \}$$

Terviklikkuse piirangud kirjeldavad relatsioonide vastavused: kui erinevates relatsioonides on atribuute, mille semantiline tähendus on sama, siis moodustavad nad seose kahe relatsiooni vahel. IC-d määravad ära sellised seosed. Kui IC on kirjeldatud hakkavad nad piirama objektides esinevate väärtuste võimalusi.

Relatsioonilise andmebaasi ilming (*relational database instance*) DB relatsioonilise andmebaasi skeemil S on kogum kogum relatsiooni ilminguid:

$$DB = \{ r_1, r_2, r_3, \dots, r_m \}$$

Seejuures ei lähe ükski relatsiooni ilming r_i vastuollu ühegi terviklikkuse piiranguga hulgast IC.

relatsiooniline andmebaas on kogum relatsiooni ilminguid

ja

kogum terviklikkuse piiranguid

terviklikkuse piirangud kirjeldavad seosed relatsioonide vahel

andmebaasi ilming on kogum relatsiooni ilminguid

Kui me räägime relatsioonilisest andmebaasist, siis viitame me tegelikult mõlemale kontseptile – andmebaasi skeemile ja andmebaasi ilmingule so. kogumile relatsiooni ilmingutele.

Andmebaasi ilmingu hetke seis nimetatakse andmebaasi jooksvaks staatuseks. Andmebaasi jooksev staatus muutub kui muutub mõne temasse kuuluva relatsiooni jooksev staatus.

relatsiooniline
andmebaas =
relatsioonilise
andmebaasi skeem +
relatsioonilise
andmebaasi ilming

andmebaasi jooksev
staatus

8.6. Relatsiooni terviklikkus ja seose terviklikkus. Primaarvõti (*Primary Key*) ja välisvõti (*Foreign Key*)

Relatsiooni terviklikkus tähendab seda, et relatsiooni super-võtme ükski väärtus ei saa olla NULL. Seda selle pärast, et nende väärtuste alusel eristatakse relatsiooni objekte.

Seose terviklikkuse piirang määratakse kahe relatsiooni vahele ja ta hoiab korras (korrastab) kahe relatsiooni ilmingu vektorite väärtusi. Sisuliselt tähendab see definitsiooni, et ühes relatsioonis olev vektor (või vektorid) saab viidata ainult teises relatsioonis olemas olevale vektorile või vektoritele.

Primaarvõti (*Primary Key*) on Super Key sünonüüm. Pikemalt ei olegi selle kohta midagi öelda

Välisvõti (*Foreign Key*) saab defineerida tema omaduste kaudu:

- 1) relatsiooni ükski välisvõti (*Foreign Key*) ei lange kunagi täielikult kokku sama relatsiooni primaarvõtmega (*Primary Key*). Vähemasti on neil erinev semantika.
- 2) kui relatsioon on seotud rekursiivselt iseendaga langevad kõik rekursiivseid seoseid kirjeldavad välisvõtmed füüsiliste parameetrite poolest kokku sama relatsiooni primaarvõtmega. Erinev on semantika. Sellisel juhul langevad kokku mõlema võtme kõik domeenid.
- 3) relatsiooni iga välisvõti (*Foreign Key*) langeb kokku mõne teise relatsiooni primaarvõtmega (*Primary Key'ga*). Sellisel juhul langevad kokku mõlema võtme kõik domeenid
- 4) vektori välisvõtme (*Foreign Key*) väärtused on kas tühjad (NULL) või langevad nad kokku seotud relatsiooni mõne vektori primaarvõtme (*Primary Key*) väärtusega

relatsiooni terviklikkus

seose terviklikkus

Primaarvõti (*Primary Key*) = Super Key

Välisvõti (*Foreign Key*)

8.7. Tegevused relatsioonilise andmebaasi ilmingul

Vektori lisamine relatsiooni – relatsiooni lisatakse uus vektor, mille väärtused vastavad relatsiooni skeemis määratud atribuutide domeenidele. Järgida tuleb seose terviklikkuse piiranguid – vektori lisamisel ei tohi rikkuda ühtegi terviklikkuse piirangut. See tähendab seda, et välisvõtme (*Foreign Key*) väljadel ei tohi esineda ühtegi väärtust, mida poleks olemas selle võtme kaudu seotud relatsiooni mõne vektori primaarvõtmes (*Primary Key's*).

vektori lisamine

Vektori uuendamine (muutmine) – relatsioonis olemasoleva vektori väärtusi muudetakse selliselt, et need ei lähe vastuollu relatsiooni skeemis määratud atribuutide domeenidega. Väärtuste muutumisel tuleb jälgida seose terviklikkuse piiranguid – vektori väärtuste muutmisel ei tohi rikkuda ühtegi terviklikkuse piirangut. See tähendab seda, et välisvõtme (*Foreign Key*) väljadel ei tohi tekkida ühtegi väärtust, mida poleks olemas selle võtme kaudu seotud relatsiooni mõne vektori primaarvõtmes (*Primary Key's*). Primaarvõtme väärtuse muutmine ei ole soovitatav. Seda saab muuta aga ainult juhul, kui relatsiooniga välisvõtme (*Foreign Key*) seotud relatsioonides ei ole selles seoses ühtegi vektorit. Vastasel korral lõhuks primaarvõtme (*Primary Key*) uuendamine need seosed ja järelikult rikutaks terviklikkuse piiranguid.

vektori uuendamine

Vektori kustutamine – relatsioonis olemasoleva vektori hävitamine.

Kustutamisel tuleb jälgida, et ei kustutataks vektorit sellise primaarvõtme (*Primary Key*), millel on seostatud vektoreid teistes relatsioonides, sest see rikuks seose terviklikkuse piiranguid. Kui selline kustutamine on siiski vajalik, tuleb eelnevalt kustutada ka kõik seotud vektorid teistes relatsioonidest.

vektori kustutamine