

6. Andmestruktuuride formaliseerimine – mustriid andmemudelite loomisel

Mudelid, seal hulgas andmemudelid, on alati ebatäpsed. Seda peamiselt kahel põhjusel – konkreetsel ajahetkel huvitabki meid „terviklikust mudelist“ ainult osa ja ka selle „mudeli osa“ puhul ei huvita meid mitte tema täpne vastavus „reaalsele maailmale“ vaid vastavus meie „hetke vajadustele“. Lisaks sellele ei ole praegusel hetkel (ja veel üsna pika aja jooksul) meie kasutuses vahendeid, millega me suudaksime luua „täiesti täpseid“ mudeleid. Ja kui me enda arvates olemegi loonud „täiusliku mudeli“, siis kas on võimalik kuidagi tõestada, et see mudel on teda kirjeldava objektiga „täpselt identne“? Vaevalt. Ja kas ei muutu ehk täpne mudel modelleeritavaks asjaks eneseks, mida tuleb jälle modelleerima hakata jne.

Seega peame leppima tõsiasjaga, et meie poolt loodud andmemudelid (ja ka mistahes teised mudelid) rahuldavad vaid meie hetkevajadusi ja koos aja edenemise ja vajaduste muutumisega peavad ka mudelid muutma. Ja muutvadki. Muutuvad kogu aeg ning tekitavad suuri segadusi töötavates infosüsteemides.

Võib tunduda, et see on vältimatu, sest kui me suudame teha ainult oma hetkevajadusi rahuldavaid mudeleid, siis mis meil muud üle jääb – vajadused muutuvad ja seega peavad muutuvad ka mudelid. Nii see tõe poolest on. Siiski võib mudelite loomisel ja muutmisel rakendada meetodikaid, mis aitavad meil selle protsessi muuta kontrollitavaks ning seega vähem vaeva nõudvaks.

Enamikel juhtudel hallatakse andmemudeli muutusi „parima teadmise tasemel“ st. omatakse teatavat (ja võib olla sugugi mitte halba) kogemust andmete analüüsimisel, sünteesimisel ning normaliseerimisel ja iga vajaduste muutumise järel struktureeritakse andmemudel uuesti. Sellisel juhul ei saa me kunagi olla kindlad, milline andmemudeli osa muutub ja millises ulatuses see muutub. Tingitud on kõik muutused ainult hetke vajadustest ja üldjuhul muutub mudel meie jaoks kõige ebamugavamal viisil, mis sunnib hulgaliselt sellel andmemudelil baseeruvat rakendustarkvara ringi tegema. Kuna andmemudel on üks infosüsteemi mudeli alustalasisid, siis iga olemasoleva struktuuri muudatus seal põhjustab hulgaliselt muutusi kõigis teistes mudeli osades ja on seega väga kallis lõbu.

Selleks et sellist situatsiooni vältida peame me mõtestama ümber andmemudeli muudatuse tähenduse. Peame käsitlema andmemudeli muutust kui mingite struktuuride lisandumist mudelisse mitte aga

mudelid on alati ebatäpsed

mudelis vastavad hetke vajadusele

mudelid muutuvad kogu aeg, sest muutuvad vajadused

me ei suuda kontrollida vajaduste muutumist

modelleeritakse oma "parimate teadmiste tasemel"

mudeli muudatus peab olema mudeli täiendamine

olemasolevate struktuuride muutumist. See tähendab seda, et me peame andmemudelit üritama luua nii, et kõik meie poolt loodud struktuurid oleksid „igavesed“ ja kõik mudeli muudatused tähendaksid „uute igaveste struktuuride“ lisamist olemasolevale andmemudelile. Seega on andmemudeli muutumine selle laienemine mitte aga olemasolevate struktuuride restruktureerimine. Sellest lähtuvalt muutub lihtsamaks ka andmemudeliga seotud rakendustarkvara haldamine kuna viimases ei tule enam teha muudatusi olemasolevates osades vaid ainult lisada uut funktsionaalsust. Selline lähenemine ei väldi küll täiesti rakendustarkavara olemasolevate osade muutmist, kuid selliste muutuste põhjuseks ei ole nüüd enam mitte andmemudeli muutumine vaid reaalselt vaja mineva funktsionaalsuse muutumine, mis on tingitud äriprotsesside muutumisest.

Võib tekkida küsimus, kuidas on võimalik teha „õigesti“, mudelit mille objekt ei ole „lõpuni tunnetatav“? Oluline on mõista seda, et „lõpuni tuleb tunnetada“ need mudeli osad, mille me kirja paneme. Seejuures on oluline määrata õigesti kõik andmemudelil kirjeldatud olemite vahelised seosed. See kas olemite omadused (andmeelemendid) on lõplikult kirja pandud polegi nii oluline. Hilisem andmeelemendi lisamine olemisse tekitab juurde andmevälja vaid üksikutesse kasutajaliidese osadesse muutmata kasutajaliidese sisemist loogikat. Samas valesti kirjeldatud seose muutmine võib põhjustada (ja tavaliselt põhjustabki) kasutajaliidese ja infosüsteemi sisemise toimimise loogikat.

kuidas on võimalik teha „õigesti“, mudelit mille objekt ei ole „lõpuni tunnetatav“?

6.1. Andmemudeli üldine struktuur

Andmemudel koosneb olemitest, milledest igaüks on „reaalse maailma“ mingi üldmõiste väljendus. Konkreetne üldmõiste võib olla kas:

- a) samatüübiliste objektide või alam-objektide kogumile vastav üldmõiste (näiteks: mööbel; tööpink ja tööpingi detail; hoone, korrus ja tuba; ettevõtte, osakond ja amet)
- b) samatüübiliste subjektide kogumile vastav üldmõiste (näiteks: isik, firma, FIE, organisatsioon või siis rolli põhine - ostja, müüja, klient, vms.)
- c) samatüübiliste nähtuste või alam-nähtsute kogumile vastav üldmõiste (näiteks: poltergeist; loodusõnnetus ja kahjud; emotsioon)

andmemudel koosneb olemitest olem on reaalse maailma üldmõiste väljendus

objektide või alam-objektide üldmõiste

subjekti üldmõiste

nähtuste või alam-nähtsute üldmõiste

d) samatüübilistele sündmuste või alam-sündmuste kogumile vastav üldmõiste (näiteks: abielu; liiklusõnnetus ja toimingud; paaritamine ja sündimised)

e) samatüübiliste seoste või alam-seoste kogumile vastav üldmõiste (näiteks: omandus; sugulus; töösuhe ja osakonnas töötamine)

f) objekti, alam-objekti, subjekti, nähtuse, alam-nähtuse, sündmuse alam-sündmuse, seose või alam-seose samatüübiliste omaduste või alam-omaduste loendile vastav üldmõiste (näiteks: värv; isiku liik; tööpingi tüüp ja alam-tüüp; linn ja tänav; seadme mark ja mudel)

Kui nüüd lähemalt vaadata, siis objektid, on väga sarnased alam-objektidega ja samuti ka nähtuste ning alam-nähtustega. Samuti subjektidega, mis/kes on "toimivad objektid" (*actors*). Nähtused ja alam-nähtused on lihtsalt „kehatud“ objektid. Alam-objekt tüüpi olemite puhul tuleb lihtsalt teada, et nad on objektid, mis ei ole saa eksisteerida iseseisvalt vaid eksisteerivad alati seotuna mingi objekt-tüüpi olemiga. Oma struktuurilt on nad objekt-tüüpi olemitega täielikult identsed. Vahe on ainult selles et nad ei kirjelda mitte mingeid eraldi tähendust omavaud objekte vaid objekt-tüüpi olemi sisemist struktuuri. Seega võime üldistada kaks esimest üldmõistete gruppi (a ja b) ühe mõiste alla – andmemudeli olemid, mis kirjeldavad objekt-tüüpi üldmõisteid. Nimetame selliseid olemeid edaspidi objekt- ja alam-objekt tüüpi olemiteks.

Samasuguse üldistuse võime teha sündmuste ja seoste korral. Mõlemate puhul on tegelikult tegemist seosega, mis kirjeldatakse objekt-tüüpi olemite vahel. On ju sündmus toiming, mis tavaliselt ühendab mitmeid objekte. Alam-seos olemite korral tuleb lihtsalt teada, et nad ei saa eksisteerida iseseisvalt vaid eksisteerivad alati seotuna mingi seos-tüüpi olemiga. Oma struktuurilt ja tähenduselt on nad identsed seos-tüüpi olemitega. Vahe on ainult selles et nad kirjeldavad mingi seos-tüüpi seose jagunemist alam-seosteks, mis toimivad selle seose sees (põhiseose toimimise jooksul). Seega võime üldistada need kaks üldmõistete gruppi (c ja d) ühe mõiste alla – andmemudeli olemid, mis kirjeldavad seos-tüüpi üldmõisteid. Nimetame selliseid olemeid edaspidi seos ja alam-seos tüüpi olemiteks.

Ühe olemi tüübina võime vaadelda ka viimasest punktis kirjeldatud omadusi ja alam-omadusi. Sarnaselt alam-objekt ja alam-seos tüüpi olemitele, mis kirjeldavad vastavalt objekt- ja seos-tüüpi olemite sisemist

sündmuste või alam-sündmuste üldmõiste

seoste või alam-seoste üldmõiste

omaduse või alamomaduse üldmõiste

objektid, subjektid ja nähtused on teataval määral sarnased

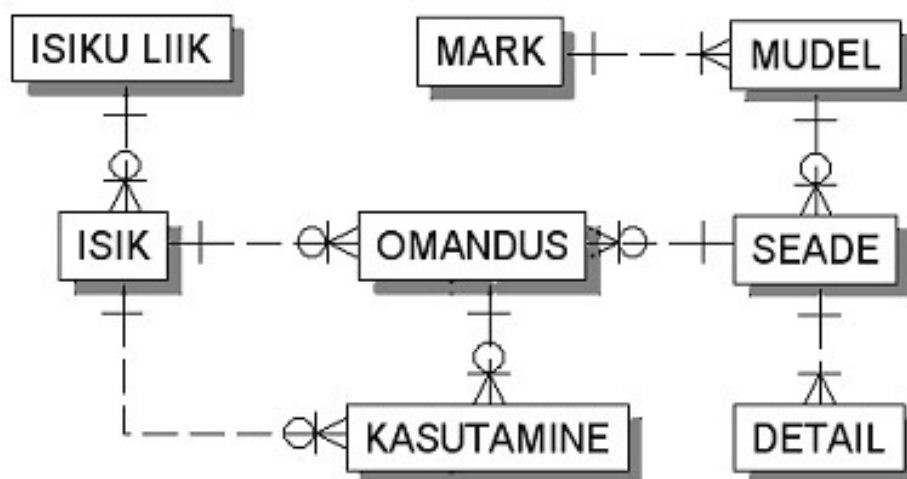
sündmused ja seosed on sarnased

omadused kui struktuuri kirjeldamise komponendid

struktuuri, kirjeldab alam-omadus tüüpi olem omadus-tüüpi olemis olevate omaduste sisemist jagunemist. Kõiges muus on nad identsed välja arvatus see, et alam-omadus tüüpi olem saab eksisteerida ainult koos omadus-tüüpi olemiga. Sisuliselt on nende näol tegemist omaduste väärtuste teatmikega. Nimetame selliseid olemeid edaspidi teatmik- ja alam-teatmik tüüpi olemiteks.

Alam-olemite tase ei ole piiratud tegelikult ühe tasemega. Kõigi tüüpide alam-olemitega võib olla seotud veel tase madalam alam-olem. Nii võib alam-objekt tüüpi olemiga olla seotud omakorda alam-objekt tüüpi olem, alam-seos tüüpi olemiga alam-seose veel detailsemaks jaotav alam-seos tüüpi olem ja alam-teatmikuga omadust veel täpsemalt detailiseeriv alam-teatmik tüüpi olem.

Et saada asjast paremat ülevaadet, siis vaatame ühte väikest näite andmemudelit, kus on kõik eelpool kirjeldatud olemitüüpi olemid olemas:



Objekt-tüüpi on siin olemitüüp „SEADE“ ja subjekt-tüüpi olem „ISIK“. Tegelikult on seda ka olem „DETAIL“, kuid täpsemalt vaadates on tema tüübiks siiski alam-objekt. Selle kohta annab tunnistust tema otseside objekt-tüüpi olemiga „SEADE“. Olemitüübid „OMANDUS“ ja „KASUTAMINE“ on seos-tüüpi. Viimane nendest on siiski alam-seos tüüpi, kuna saab eksisteerida ainult „omanduse raames“. Olemitüübid „ISIKU LIIK“, „MARK“ ja „MUDEL“ on omadus-tüüpi. Siingi on viimane nendest alam-omadus, mis ei oma tähendust ilma seoseta olemisses „MARK“.

olemid ja alam-olemid

NÄIDE

Olemis „ISIKU LIIK“ on kõigi võimalike isikuliikide loend (eraisik, FIE, AS, OÜ, ...). Olemis „ISIK“ on isikute register, millest igale määratakse omadus „isiku liik“ seosega olemisse „ISIKU LIIK“.

Olemis „MARK“ on kõigi seadmemarkide nimede loend. Olem „MUDEL“ (alam-teatmik) täpsustab igat mudelit seadmemarkide loendiga. Iga seadmemudel omab tähendust ainult läbi seadmemargi, mille alla ta kuulub. Olemis „SEADE“ on kõikide (teada olevate) seadmete register. Lisaks sellele täpsustatakse iga seadme korral tema (oluliste) detailide loend olemis „DETAIL“. Seosega olemisse „MUDEL“ määratakse seadme jaoks (oluline) omadus – milline on seadme mudeli tähis ja selle kaudu ka see millist marki on seade.

Olem „OMANDUS“ kirjeldab omandisuhte muutumise isikute ja seadmete vahel st. võimaldab registreerida seda kes ja millises ajavahemikus (konkreetset) seadet omab või omas. Olem „KASUTAMINE“ kirjeldab seadme kasutamist erinevate isikute poolt sama omanduse raames.

Andmemudelites, mille loomisel ei ole kasutatud „kavaldamist“ ja „nipitamist“, rohkem erinevaid olemitüüpe ei ole – kogu andmemudel pannakse kokku nendest. Selleks et saavutada püsivate seostega andmemudelit, tuleb teada, kuidas nende olemitüüpide olemeid võib omavahel seostada. Nagu edaspidi selgub on reeglid üllatavalt jäigad.

**on olemas reeglid
erinevate olemitüüpide
olemite seostamiseks**

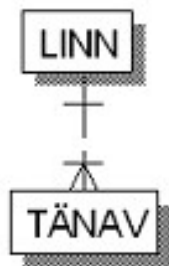
6.2. Seosed sama ja erinevat tüüpi olemite vahel

Vaatleme nüüd näidete kaudu kõigi erinevate olemitüüpide olemite vahelisi seoseid. Lisaks eel pool kirjeldatud reeglite rakendustele vaatame ka juhtumeid, kus esineb kõrvalekaldeid reeglitest.

6.2.1. Seosed teatmik-tüüpi olemite vahel

Teatmik-tüüpi olem võib alluda ainult ühele teatmik-tüüpi olemile.

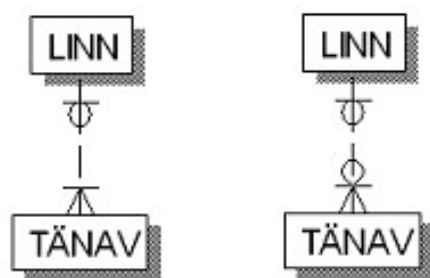
**teatmik-tüüpi olem võib
alluda ainult ühele
teatmik-tüüpi olemile**



Siin on diagrammil on kujutatud teatmik-tüüpi olemit „LINN“, mis sisaldab (vaadeldavate, meid huvitavate, infosüsteemis vajalike, vms.) linnade loendi. Linnade loendit täpsustatakse tänavate loendiga – igal linnal on oma tänavate loend ja ei ole linna, milles pole ühtegi tänavat.

Üldiselt kipubki nii olema, et kui kaks teatmiku-tüüpi olemit on omavahel seotud, siis on nende vahel seos suhteliselt jäik – igal põhi-teatmiku real peab olema vähemalt üks vaste alam-teatmikus. Mõnikord võib küll tunduda, et siiski võib olla ka nii, et põhi-teatmikku mõnda rida enam detailsemaks ei jagata (st. võib leida põhi-teatmiku rida, millel puudub vaste alam-teatmikus) kuid lähemal uurimisel selgub tavaliselt, et nii see siiski pole. Kui siiski peaks ette sattuma see „eriline juhtum“, siis tuleks situatsioon lahendada „tehniliselt“, lisades iga sellise põhi-teatmiku rea kohta, mille detailiseeringut alam-teatmikus pole, üks „tehniline“ rida alam-teatmikusse. Olgu selle sisuks kas siis „vaikimisi“ või „täpsustus puudub“ või midagi muud sarnast

Mitte kunagi ei ole võimalik see, et alam-teatmik eksisteerib vahel sõltumatult põhiteatmikust. See tähendab, et ei ole võimalik järgmised skeemid:



Kui nüüd teatmik-tüüpi olemit „TÄNAV“ alluusse lisada veel üks teatmik-tüüpi olem, näiteks „TÄNAVAOSA“, või siis olemile „LINN“ lisada ülemus-olem „RIIK“, siis kehtivad nende paaride vahel kõik täpselt samad reeglid.

NÄIDE

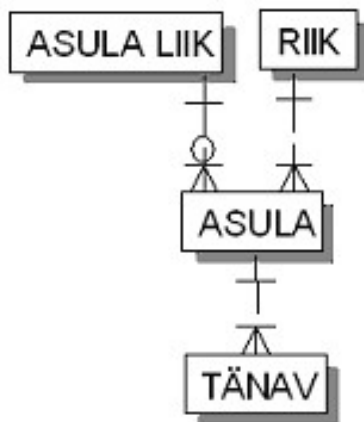
kui teatmik-tüüpi olemiga on seotud alam-teatmik, siis iga põhi-teatmiku rida on seotud vähemalt ühe alam-teatmiku reaga

alam-teatmik ei saa eksisteerita sõltumatult põhiteatmikust

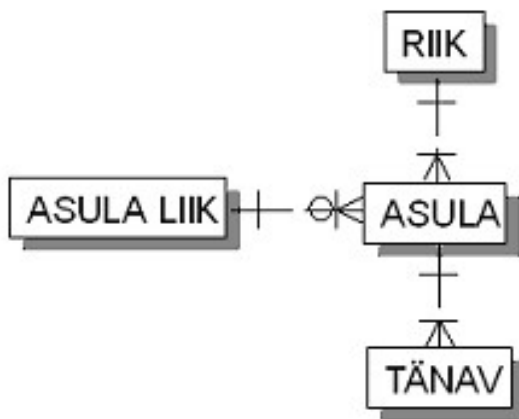
NÄIDE

6.2.2. Teatmik-tüüpi olem mitme teatmik-tüüpi olemi „alluvuses“

Ees pool väitsime, et teatmik-tüüpi olem saab olla ainult ühe teatmik-tüüpi olemi alluvuses. Vaatleme aga järgmist olemi-suhte diagrammi:



Kõik tundub loogiliselt õige olevat, kuid ikkagi on ju teatmik-tüüpi olem „ASULA“ kahe teise teatmik-tüüpi olemi alluvuses. Milles asi? Ehk muutub asi veidike selgemaks kui paigutada olemid veidike ringi:



Nüüd on näha, et alluvussuhteid on ikkagi ainult üks – olem „ASULA“ allub olemile „RIIK“ ja kirjeldab viimase sisemist jaotust. Olem „ASULA LIIK“ määrab vaid olemi „ASULA“ ühe omaduse väärtuse.

Seega võime ees pool kirjeldatud reeglit täpsustada – kui siiani rääkisime, et teatmik-tüüpi olem kirjeldab kas objekt- või seos-tüüpi olemi omaduse väärtuse, siis nüüd võime täiendada seda sellega, et teatmik-tüüpi olem võib kirjeldada ka teise teatmik-tüüpi olemi omaduse väärtust. See tähendab seda, et kahe teatmik-tüüpi olemi vaheline seos ei kirjelda tingimata ühe olemi (põhi-teatmiku) sisemist jaotust vaid võin määrata teatmiku mingi omaduse väärtuse.

NÄIDE

teatmik kahe teatmiku alluvuses?

NÄIDE jätkub

teatmik-tüüpi olem võib määrata teise teatmik-tüüpi olemi omaduse (mitte alluvuse)

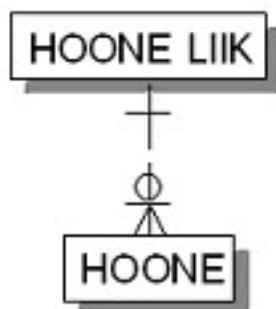
Sellise seose tunnuseks on see, et tema „mitu“ poolses otsas võib seos ka puududa – nii nagu on see näha seoses olemite „ASULA LIIK“ ja „ASULA“ vahel

6.2.3. Seosed teatmik- ja objekt- või subjekt-tüüpi olemite vahel

Seosed teatmik-tüüpi olemit ja objekt-tüüpi olemit ning teatmik-tüüpi olemit ja subjekt-tüüpi olemit vahel on täpselt ühesugused. Seepärast ei kirjeldata siin mõlemat seost vaid piirduakse ainult teatmik-tüüpi ja objekt-tüüpi olemite vahelise seose kirjeldamisega kirjeldatu on üks-ühele üle kantav ka subjekt-tüüpi ja teatmik-tüüpi olemite seose kirjeldamiseks.

Teatmik- ja objekt- tüüpi olemite vaheliste seoste anatoomia on suhteliselt triviaalne – Ühe teatmik-tüüpi olemiga võib olla seotud mitu objekt-tüüpi olemit ja ühe objekt-tüüpi olemiga või olla seotud mitu teatmik-tüüpi olemit. Samade teatmik- ja objekt-tüüpi olemite vahel võib olla ka rohkem kui üks seos. Mistahes seose valdaja on alati teatmik-tüüpi olem.

Vaatame kõigepealt kõige lihtsamat teatmik- ja objekt-tüüpi olemis seost:



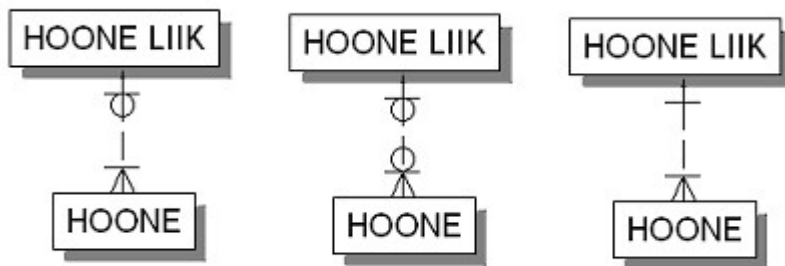
See on tüüpiline seos teatmik- ja objekt-tüüpi olemite vahel. Teistsugune seos ei ole siin lubatud. See tähendab, et on võimalikud hoone liigid, millega hooneid (veel) seotud pole, samas on aga hoone puhul hoone liik alati teada. Vahel võib tekkida mõtte luua ka mõni teistsugune seos, kus oleks võimalik luua objekt ilma teda teatmikuga seostamata või siis teatmiku rida, mis on oma loomisest peale seotud mõne (vähemalt ühe) objektiga:

teatmik-tüüpi olemit ja objekt-tüüpi olemit ning teatmik-tüüpi olemit ja subjekt-tüüpi olemit vahel toimivad täpselt samade reeglite alusel

seosed on alati teatmik-tüüpi olemite poolt objekt-tüüpi olemit poole

NÄIDE

see on ainus lubatud seos teatmik- ja objekt-tüüpi olemite vahel



Negatiivne NÄIDE

See mõte tuleb aga kohe peast heita. Esimesed kaks varianti kirjeldavad juhtumit, kus on võimalik objekt, millel seos teatmikuga puudub. Selline käsitus on tavaliselt lihtsalt vale ja realiseerumisel põhjustab suuri probleeme mudeliga seotud programmide loomisel (OUTER JOIN – vajadusi, päringute keerukust ja aeglust). Kui selline situatsioon peaks siiski ette tulema tuleb asi lahendada „tehnilise“ rea lisamisega teatmikku („määramata“, „pole teada“ vms.) ja kasutada ikkagi ülal kirjeldatud varianti.

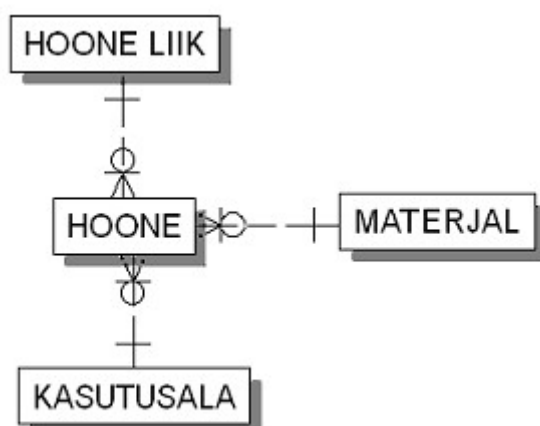
Ei!

Kolmanda variandi peale ei tasuks aga kunagi isegi mõelda – see on lihtsalt vale. Selle tõestuseks on väide, et meil ei õnnestu teatmikku eel-laadida – iga teatmiku rea saame mee baasi kirjutada alles siis kui ilmneb esimene objekt, mis konkreetse teatmiku reaga on seotud. Eel-laetavus on aga teatmik-tüüpi olemite üks põhiomadusi.

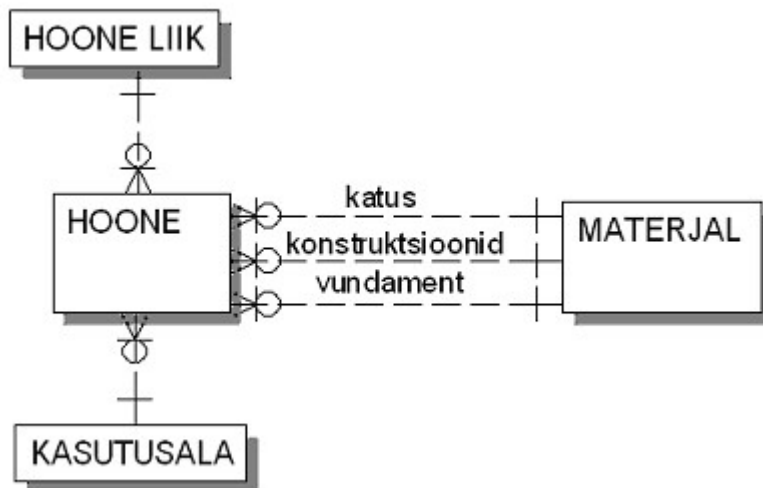
Ei!

Kui sama objekt-tüüpi olemiga on seotud mitu teatmik-tüüpi olemit, siis on suhe nende vahel alati sama tüübiline:

alati sama tüüpi seos

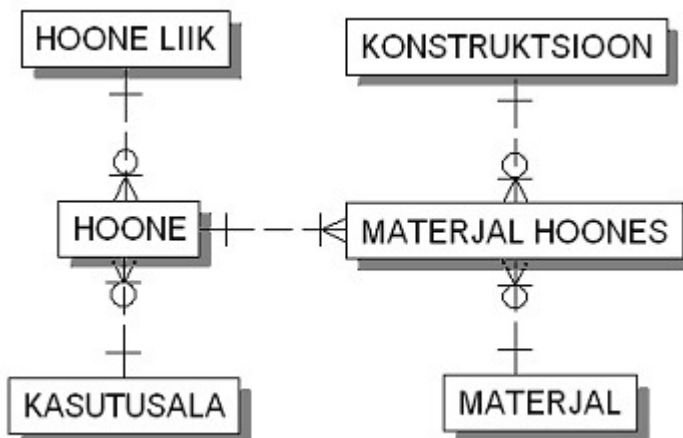


Seda isegi siis kui kahe olemit vahel on rohkem kui üks seos:



Iseasi on muidugi see, et kui kahe tabeli vahele hakkab tekkima mitmeid seoseid on mõttekas see konstruktsioon üle vaadata ja asendada millegi dünaamilisemaga. Antud juhul võiks see välja näha järgmisena:

kui kahe olemi vahele tekib seoseid rohkem kui kaks tuleb mudelit üldistada

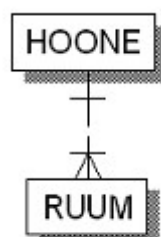


Ühtlasi tasuks mõelda, kas olem „MATERJAL“, mis seni oli teatmik-tüüpi pole ehk hoopis objekt-tüüpi ja tema käsitus hoopis nüansikam. See on aga samm järgmisele keerukuse tasemele, mida vaatleme edaspidi.

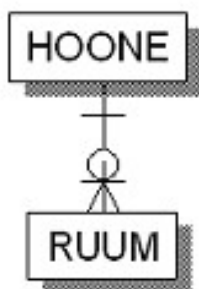
6.2.4. Seosed objekt-tüüpi olemite vahel

Nagu mitmeid kordi mainitud, kui kaks objekt-tüüpi olemit on omavahel seoses, siis üks nendest on põhi-objekt ja teine alam-objekt, mis ei ole iseseisvalt vaadeldav ja kirjeldab põhi-objekti sisemist struktuuri:

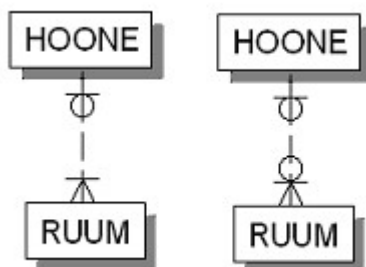
põhi-objekt ja alam-objekt



Seos on suhteliselt sarnane sellega, kui üks teatmik-tüüpi objekt oli teise alluvuses. Suhe on enamasti sama jäik - kui kaks objekt-tüüpi olemit on omavahel seotud, siis igal põhi-objektil peab olema vähemalt üks vaste alam-alamobjektide hulgas. Muidu on ta ilma sisemise struktuurita objekt. Siiski sõltuvalt protseduurilistest reeglitest, mille alusel andmebaasis andmeid registreeritakse, võib olla nii, et põhi-objekt registreeritakse enne ja ühes kohas, tema struktuur aga hiljem ja teises kohas. Siis on protseduurilistest reeglitest tingituna lubatud ka selline skeem:



Mitte kunagi ei ole võimalik see, et alam-objekt eksisteerib sõltumatult põhi-objektist - alati peab olema teada millise objekti struktuuri-komponendiga on tegemist. See tähendab, et ei ole võimalik järgmised skeemid:



Kui soovime omavahel siduda kaht objekt-tüüpi olemit, millest kumbki ei ole teise alam-objekt, siis on ainsaks võimaluseks teha seda seos-tüüpi olemit vahendusel. Seda selle pärast, et kahe „sõltumatu objekti“ sidumisel tuleb alati arvesse võtta seda, et see seos võib aja jooksul muutuda, või võib seos olla samaaegselt mitmete objektide vahel.

Vaatame kõigepealt esimest juhtumit, kus peab olema võimalik seost objektide vahel muuta ja samas ka muutuste ajalugu jälgida:



Siin on sõltumatud olemit „FIRMA“ ja „RUUM“. Kumbki ei ole teise komponent, samas on aga vaja teada, milline firma, millal ja millist ruumi rentis või rendib. Lisaks sellele on võib-olla vaja teha ka ruumide broneerimist tulevikus. Meil on firmasid, mis kunagi ei ole ühtegi ruumi rentinud ja meil võib olla ka ruume mida ükski firma kunagi pole rentinud.

igal põhi-olemis kirjeldatud ilmingul peab olema vähemalt üks seotud ilming alam-olemis

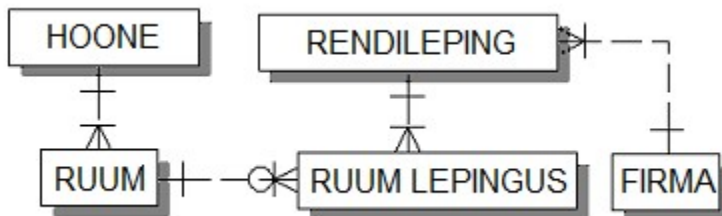
alam-objekt ei esine kunagi iseseisvalt, ilma põhi-objektita

sõltumatuid objekte saab siduda ainult läbi seos-tüüpi olemit

NÄIDE 1

Samas kui toimub rentimine on alati teada kes ja millise ruumi rentis. See, millal rentimine toimus kirjeldatakse olemis „RENTIMINE“. See on kõige tavalisem seos kahe sõltumatu olemi vahel. See kõige tavalisem seos on ühtlasi kõige nõrgem seos kahe objekt-tüüpi olemi vahel. Valdav osa sõltumatute olemite vahelisi seoseid on just sellised.

On võimalik ka natuke tugevam seos objekt-tüüpi olemite vahel. Sellisel juhul tekib üks seoses olevatest objektidest koos seosega:



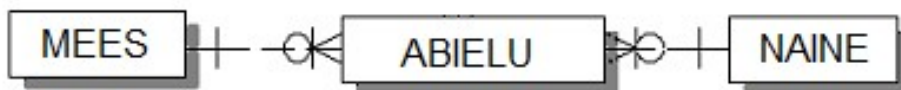
See on päris huvitav skeem. Siin on mitmeid kohustuslikke seoseid. Nii näiteks ei saa hoone eksisteerida ilma et seal oleks ühtegi ruumi (need on objekt ja alam-objekt). Samuti ei ilmu firma meie andmebaasi enne, kui tal on esimene rendileping sõlmitud. Ja lõpuks kui tekib rendileping, siis tekib sinna ka esimene ruum lepingus. Olem „RUUM LEPINGUS“ on seos-tüüpi olem, kus üks tema kaudu seotud objektidest („RENTILEPING“) tekib koos esimese rentimisega. Samas võib olla ruume, mida ei ole veel kordagi välja renditud. On vähe tõenäone, et selliseid, koos seosega tekkivaid objekte oleks samas seoses rohkem kui üks.

NÄIDE 2

6.2.5. Seosed subjekt-tüüpi olemite ja subjekt- ning objekt-tüüpi olemite vahel

Subjekt-tüüpi olem ei saa olla subjekt-tüüpi olemiga vahetult seotud. Kui selline seos eksisteeriks, siis tähendaks see seda, et üks subjekt oleks teise subjekti komponent. Elus sellist asja ei eksisteeri, seega pole ka sellist seos andmemudelites olemas.

Subjekt saab olla subjektiga seotud ainult läbi sündmus-tüüpi olemi:



Siin skeemis "MEES" ja "NAIN" on subjekt-tüüpi olemid ja "ABIELU" on sündmus-tüüpi olem, mille kaudu ON VÕIMALIK siduda subjekt-tüüpi olemeid. Nagu on näha "mullidest" seoste "mitu" otstes on tegemist

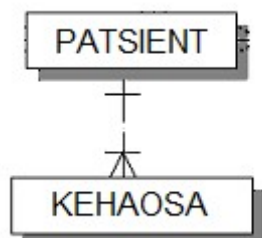
subjekt ei saa olla subjekti komponent

subjektide vahelised seosed on alati vabatahtlikud

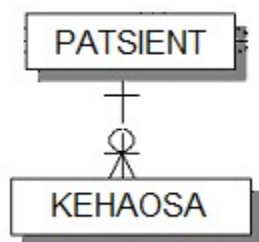
vabatahtliku ühendusega. Subjektide vaheliste seoste joonistamisel on tegemist ALATI just selliste relatsioonidega.

Subjekt-tüüpi olem ei saa kunagi olla objekt-tüüpi olemi komponent. See tähendab seda, et objekt-tüüpi olem ei saa olla kunagi seoses subjekt-tüüpi olemiha üks-mitmese suhte "üks" poolses otsas. Vastasel juhul oleks ju subjekt objekti komponendiks. Kuna elus pole nii kunagi, siis ei saa me ka andmemudelisse nii joonistada.

Küll aga saab olla objekt-tüüpi olem seotud subjekt-tüüpi olemiga nii, et objekt-tüüpi olem on suhtes alluvaks st. üks-mitmese suhte "mitu" poolses otsas. Sellisel juhul on temaga võimalik kirjeldada subjekti komponente - näiteks keha osi:



See skeem kirjeldab andmemudelit, kus registreeritakse mingil põhjusel patsientide kehaosi. Seejuures iga patsiendi kohta peab olema kirjeldatud vähemalt üks kehaosa. Üldiselt on ju nii, et igal isikul peaks olema vähemalt üks kehaosa. Subjektide kirjeldamisel ei ole aga peaaegu kunagi eesmärgiks nende "lõplik" kirjeldamine, vaid kirjeldamine ainult selles osas, mis osutub otstarbekaks. Seega on selline suhe praktiliseks kasutamiseks ilmselt liiga jäik ja tegelikult subjektide komponentsuse kirjeldamisel peaks veidi rohkem vabadust jätma:



Siin skeemis saab vajadusel jätta subjekti "komponendid" ka kirjeldamata. Ilma erilise põhjuseta ei ole subjekti ja tema komponentsust kirjeldava objekt-tüüpi olemi vahelist seost mõtet teist sugusena projekteerida.

subjekt ei saa kunagi olla objekti komponent

objekt võib kirjeldada subjekti komponentsust

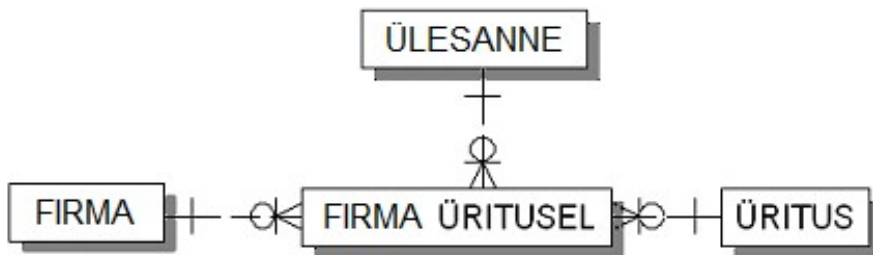
subjekt ja tema komponendid

üks - null, üks või mitu komponenti

6.2.6. Seosed teatmik- ja seos-tüüpi olemite vahel

Teatmik- ja seos- tüüpi olemite vaheliste seoste olemus on suhteliselt triviaalne – ühe teatmik-tüüpi olemiga võib olla seotud mitu seos-tüüpi olemit ja ühe seos-tüüpi olemiga või olla seotud mitu teatmik-tüüpi olemit. Samade teatmik- ja seos-tüüpi olemite vahel võib olla ka rohkem kui üks seos. Mis tahes seose valdaja (ülemus) on alati teatmik-tüüpi olem.

Vaatame ühte lihtsat teatmik- ja seos-tüüpi olemi seost:



Siin on kaks objekt-tüüpi olemit „FIRMA“ ja „ÜRITUS“, mis on omavahel seotud sündmus-tüüpi olemi „FIRMA ÜRITUSEL“ vahendusel. Teatmik-tüüpi olem „ÜLESANNE“ määrab firma ülesande(d) üritusel (toitlustamine, loterii korraldamine, lavashow, jne.) See on tüüpiline seos teatmik- ja suhe-tüüpi olemite vahel, mis on analoogiline seosele teatmik- ja objekt-tüüpi olemite vahel. Tegelikult kehtibki siin täpne analoogia teatmik- ja objekt-tüüpi olemite vaheliste seostega nii et pikemalt pole seda mõtet lahti kirjutada.

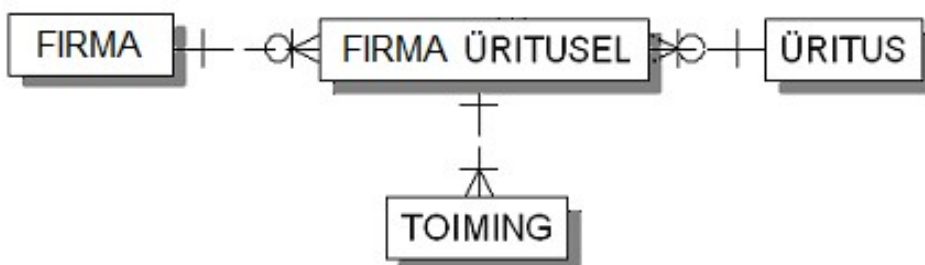
teatmik - ja seos-tüüpi olemi suhtes on teatmik- olem alati üks-mitmese suhte "üks" poolel

NÄIDE

suhe on täpselt sama mis teatmik- ja objekt-tüüpi olemite vahel

6.2.7. Seosed seos-tüüpi olemite vahel

Analoogiliselt objekt-tüüpi olemitele toimib ka seos seos-tüüpi olemite vahel. Kui kaks seos-tüüpi olemit on omavahel seoses, siis üks nendest on põhiseos ja teine alamseos, mis ei ole iseseisvalt vaadeldav ja kirjeldab põhi-seose sisemist struktuuri – selle raames toimivaid alam-seoseid:

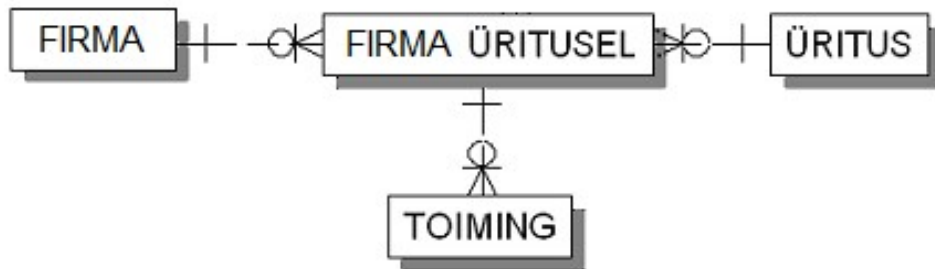


Seos on sarnane sellega, kui üks objekt-tüüpi objekt oli teise alluvuses. Suhe on enamasti sama jäik - kui kaks seos-tüüpi olemit on omavahel seotud, siis igal põhi-seosel peab olema vähemalt üks vaste alam-seoste hulgas. Muidu on ta ilma sisemise struktuurita seos. Nii näiteks olemisse „FIRMA

põhiseos ja alamseos

NÄIDE

ÜRITUSEL“ andmete tekkimisel peaks tekkima ka kohe esimene toiming näiteks kas „registreerimine“ (kui oli tegemist eelregistreerunud firmaga) või „saabumine“ kui oli tegemist firmaga, kes saabus üritusele ilma eelregistreerimiseta. Siiski sõltuvalt protseduurilistest reeglitest, mille alusel andmebaasis andmeid registreeritakse, võib olla nii, et põhiseos registreeritakse enne ja ühes kohas, tema struktuur aga hiljem ja teises kohas. Siis on protseduurilistest reeglitest tingituna lubatud ka selline skeem:



Mitte kunagi ei ole võimalik see, et alam-seos eksisteerib sõltumatult põhiseosest - alati peab olema teada millise seose struktuuri-komponendiga on tegemist (millise ürituse toiminguga on tegemist).

alam-seos ei eksisteeri kunagi iseseisvalt