

4. Andmemudelid ja ERD

4.1. Olemi-suhte skeem (ERD) ja andmebaaside loomine

Olemi-suhte diagramm (*ERD - Entity Relationship Diagram*) on kõige laiemalt levinud meetodika andmemudelite koostamiseks ja kirjelduse esitamiseks.

Tegemist on pool-formaalse mudeliga, kus on ühendatud:

1. andmemudeli graafiline esitus - olemid (*entity*), vaated (*views*) ja nende vahelised seosed (*relationships*)
2. tabelite sisemise struktuuri formaalsed kirjeldused (olemite atribuudid, atribuutide andmetüübid, piirangud, indeksid)
3. vaadete formaalsed kirjeldused (kuidas vaated olemite kaudu kirjelduvad)
4. olemite ja vaadete vaheliste seoste formaalsed kirjeldused (seoste väljade kirjeldused, seose tugevuse kirjeldused)
5. mudeli komponentide verbaalsed kirjeldused (olemite, atribuutide, seoste, protseduuride jm. semantika vabatekstiline kirjeldus)

Kogu andmebaasi loomise protsess toimub järgmiselt. Kõigepealt koostatakse andmemudel. Seda tehakse just nii täpselt, kui seda parasjagu on vaja.

Seejärel koostatakse loodud mudeli alusel SQL-laused, mille abil luuakse füüsiline andmebaas. Kui nüüd aja jooksul selgub, et andmebaasi struktuuris on vaja teha muudatusi, siis täiustatakse/muudetakse ERD mudelit ja seejärel luuakse muudetud mudeli alusel SQL-laused, mille abil muudetakse andmebaasi struktuur vastavaks loodud mudelile. jne. kuni "lõpmatuseni"

Tegelikkuses on peale andmebaasi struktuuri muutmise veel vaja teha olulisi tegevusi. Ei ole ju andmebaasid nende struktuuri muutuste ajal tühjad, vaid seal on andmed, mille säilivus tuleb tagada ka pärast muutusi. Seepärast tuleb enne andmebaasi struktuuri muutmist tõsta sellese andmebaasi osa, kus tehakse muudatusi, andmed andmebaasist välja (*backup*) ja pärast andmete struktuuri muutust tuleb olemasolevad andmed paigutada uuele struktuurile.

Tänu ERD mudeli heale formaliseeritusele on selle loomise ja haldamise tarvis võimalik päris hästi teha ERD-I põhinevaid andmete modelleerimist toetavaid CASE-süsteeme. See võimaldab kogu andmete modelleerimise ja andmebaaside loomise ning andmestruktuuride haldamise/muutmise protsessi väga hästi ära automatiseerida. Kõigepealt luuakse andmebaasi andmemudeli ERD-I põhinev kirjeldus. Loodud olemi-suhte diagrammi alusel genereeritakse

olemi-suhte diagramm

graafiline esitus

olemid ja seosed

vaated

seosed

semantika kirjeldus

andmebaasi loomise protsess

andmete ümber struktureerimine

CASE-süsteemid

andmebaasi loomise käsud. Tavaliselt on andmebaasi kirjeldamise keeleks SQL-keel. Olemite kirjeldustest genereeritakse tabelite kirjeldused. Olemi atribuutide kirjelduste alusel genereeritakse tabelite veergude kirjeldused. Vaadete ja indeksite kirjeldustest genereeritakse andmebaasi vaated ja indeksid. Olemite ja vaadete vaheliste seoste kirjeldustest genereeritakse andmebaasi seosed. Pärast loodud SQL-keelse kirjelduse käivitamist andmebaasis ongi tühi andmebaas loodud.

Kui nüüd on vaja andmebaasi struktuuri muuta, tehakse vajalikud muudatused ERD-mudelis. Seejärel võrreldakse CASE-süsteemi abil uut mudelit olemasoleva andmebaasi struktuuriga ja genereeritakse SQL-keelsed andmebaasi andmestruktuuri muutmise laused. Loodud laused täidetakse andmebaasisüsteemis ja pärast muutuste tegemist viiakse eelnevalt andmebaasis olnud andmed vastavusse uue struktuuriga

andmebaasi struktuuri muutmine

4.2. ERD komponendid. ERD-ga seotud mõisted

ERD põhilisteks komponentideks on olemid (vastavad andmebaasi tabelitele) ja olemite vahelised seosed (vastavad tabelite vahelistele seostele andmebaasides). Olemite enda sisemine struktuur koosneb atribuutidest (vastavad andmebaasi tabeli veergudele). Seoste kirjeldus koosneb seotud olemite nende atribuudi paaride kirjeldustest, mis kirjeldavad seose kahe olemi vahel. Seos saab olla kirjeldatud alati ainult kahe olemi vahel või siis olemi seosena ise endaga (sellisel juhul on seose mõlemas otsas see sama objekt - seos moodustab suletud tsükli selle ühe olemiga).

ERD põhilised komponendid

Olem on (andmekoosluse) määratlus, mis väljendab reaalse maailma mingile objektile, nähtusele, sündmusele või reaalse maailma objektide seosele vastavat loogilist üldmõistet (kirjeldust). Olem on kompleksne mõiste mis hõlmab endas kogu antud temaatikaga seotud kirjeldust. Kui olemile on kirjeldatud ka sisemine struktuur, siis "kirjeldab olem maailm" ainult nendes piirides, mida olemi sisemine struktuur võimaldab.

reaalse maailma üldmõiste väljendus

Näiteks väljendab olem "Auto" reaalse elu objektide kogumit "autod" ilma lahti mõtestamata, millist konkreetset autot me mõtleme (mingi auto, kõik teadaolevad autod, kõik autod, kõik "meie" autod, kõik antud protsessis osalevad autod)

NÄIDE - olem "Auto"

Olemi-suhte diagrammi semantika, milles olem asub määrab olemiga kirjelduvatele objektidele mingid täpsemad piirid – autoregistri mudelis olev

olemi semantika mudeli semantika skoobis

olem "Auto" tähistab kõiki registris olevaid autosid fikseerituna nende omadustega, mis vastava registri seaduses on ette nähtud.

Näiteks väljendab olem "Isik" reaalse elu objektide kogumit "isikud" ilma lahti mõtestamata, milliseid konkreetseid isikuid me silmas peame (kõik maailma inimesed, kõik eesti kodanikud, kõik Eesti pinnal olevad füüsilised ja juriidilised isikud ning FIE'd vms.) Kui me nüüd aga ütleme, et tegemist on Eesti kodanike registri olemi-suhte diagrammiga, siis kirjeldub olem "Isik" kohe hulga täpsemalt, sest Elanikkonna registri seaduses on väga täpselt kirjas, kes tohivad sellesse registrisse kuuluda (ainul kodanikud ja pikaajalise elamisloa alusel Eestis viibivad isikud) ja millised andmed iga registrisse kuuluva isiku kohta tuleb talletada.

Olem "Abielu" väljendab seost kahe inimese vahel, millel on alguskuupäev, lõpukuupäev, seost reglementeeriv seadusandlus ja võib-olla ka abieluleping (või vara lahususe leping). Selline olem omab tähendust ainult kas riikliku abieluregistri või mõne kiriku infosüsteemi olemi-suhte diagrammis. See olem eeldab muidugi ka seda, et kusagil üsna lähedal asub olem "Isik".

Igal olemil on semantika, mis kirjeldab olemi olemust ja toimimist. Olemi semantika on olemi verbaalne ("jutustav") kirjeldus, milles kirjeldatakse olemi seos reaalse maailmaga ning olemi ilmingute andmebaasi tekkimise, nende muutumise ja hävitamise reeglid.

Näiteks võib olemi "Auto" semantika olla järgmine – autoregistrisse kantavate autode teatmik, kuhu auto tekkib siis, kui ta esimest korda arvele võetakse. Muutused auto andmetes tehakse siis, kui muutub tema kere number (kere remondi käigus), värv (värvitakse üle), mootori number (mootor vahetatakse) või vahetatakse auto registrinumbrit. Auto mark, mudel ja väljalaskeaasta ei saa muutuda. Autot ei kustutata registrist füüsiliselt kunagi – kui auto võetakse registrist arvelt, kas maha kandmiseks või välismaale müügiks, siis tehakse autole sellekohane märge.

Olemit võib vaadelda kontseptuaalsel tasemel st. ilma tema sisemist struktuuri täpsustamata rahuldudes lihtsalt olemi semantika tundmisega. Andmemudelite projekteerimise alguses vaadeldaksegi kõiki, või vähemalt enamusi olemeid, kontseptuaalsel tasemel. Sellist ERD mudelit, kus olemit sisemine struktuur ei ole veel täpsustunud nimetatakse infooloogilisteks mudeliteks. Infooloogilistes mudelites koosneb ERD omavahel seostatud olemitest, mida tähistavad vaid risküliku sisse kirjutatud olemitest semantilised nimed:

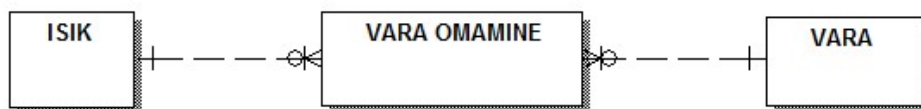
NÄIDE - olem "Isik"

NÄIDE - olem "Abielu"

**olemi semantika
täendus**

**NÄIDE - olemi "Auto"
semantika**

**Kontseptuaalne ERD -
Infooloogiline mudel**



See skeem kirjeldab vara ja isikute vahelisi seoseid. See kas me teame skeemil kirjeldatud olemite sisemist struktuuri ei aita meil paremini mõista antud skeemi kontseptuaalset tähendust. Olulisem on teada, millised on skeemil toodud olemite semantika. Oluline on näiteks teada, milliseid isikuid seob olem ISIK. On seal ainult füüsilised isikud või kuuluvad sinna ka firmad, mittetulundusühingud, parteid ja FIE'd. Oluline on teada, millised objektid kirjelduvad olemis OBJEKT. On seal ainult kinnisvara või ka vallasvara. Olemi VARA OMAMINE puhul on oluline teada, kas selle olemiga kirjeldatakse ainult kehtivad omandussuhted või hoitakse seal ka omanduse ajalugu. Andmemudeli kontseptuaalse mudeli loomisel on vaja vastata just sellistele küsimustele. See kõik on täiesti piisav selleks, et projekteerida vajalikud olemid ja nende vahelised seosed. Atribuudid (olemite sisemise struktuuri) võib kirjeldada alles pärast seda, kui infooloogiline mudel on valmis.

Pärast seda kui olemite semantika on kirjeldatud "tekkib" semantika kirjeldatud seostele. Olemite ISIK ja VARA OMAMINE vahel oleva seose semantikaks on "isiku poolt omatav või kunagi omatud kinnisvara" ja tabelite VARA ning VARA_OMAMINE vahelise seose semantikaks on "isikud, kes omavad seda kinnisvara praegu või on kunagi omanud seda". Kui kahe olemi vahele on kirjeldatud seos, kuid selle semantikat pole võimalik verbaalselt kirjeldada, siis ilmselt on tegemist valesse kohta joonistatud seosega.

Oletame, et me tahame luua kinnisvara omanike registrit. Sellisel juhul kirjeldame me olemi VARA, kui kõigi kinnisvaraobjektide kogumi. Olem ISIK kirjeldab kõiki selliseid inimesi, kellel on praegu kinnisvara, või kelle seda varem on olnud. Seejuures tuleb lisapiiranguna määrata kindlaks, kas me tahame iga olemis registreeritud kinnisvara objekti kohta registreerida selle kõik varasemad omanikud või mitte. Esimesel juhul on isikute ring oluliselt laiem kui teisel juhul. Kinnisvara registri korral peame me lisaks kehtivatele omandussuhetele säilitama ka omandussuhete ajaloo minevikus.

Vaatame nüüd ülal toodud ERD mudelit (kontseptuaalset skeemi) kirjeldatud piirangute valguses. Pärast mudeli, olemite ja olemite vaheliste seoste semantika kirjeldamist on joonisel toodud ERD skeem tunduvalt konkreetsem

NÄIDE

**mudeli semantika
mudeli olemi
semantika**

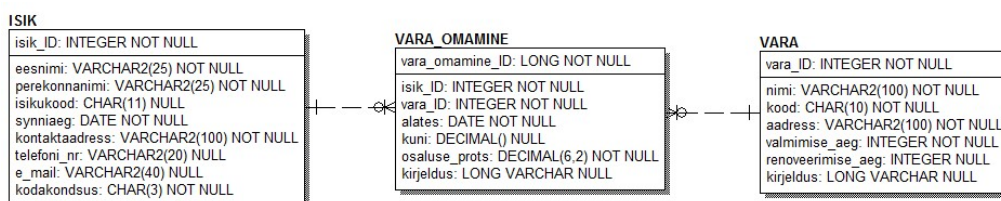
seose semantika

NÄITE täpsustus

ja täpsemini määratletud. Sellise taseme kirjelduse alusel võime me juba üpris kindlalt hinnata skeemi vastavust vajadustele.

Pea mitte kunagi ei joonistata ERD skeemi lõpuni ilma olemite sisemist struktuuri kirjeldamata. Ikka on olemitel mingeid selliseis omadusi, mille kirjeldamine juba mudeli kontseptuaalsel tasemel annab mudelile olulist informatsiooni juurde. See ei muuda aga kontseptuaalse (infooloogilise) mudeli olemust. Atribuudid, mis kirjeldatakse olemitele olemite oluliste omaduste kirjeldamiseks täidavad sellisel juhul teist eesmärki. - nad ei kirjelda mitte niivõrd olemit sisemist struktuuri kui võrd on olemite semantilise kirjelduse komponentideks. Olemi struktuuri komponentideks muutuvad nad pärast olemit täieliku struktuuri projekteerimist.

Pärast andmemudeli semantiliste mudeli (infooloogilise mudeli) koostamist tuleb see viia andmeloogilisele tasemele. Andmeloogilise modelleerimise käigus projekteeritakse olemite sisemine struktuur, mis koosneb olemi omadustest ja mille väljenduseks ERD-mudelis on olemit atribuudid:



Siin joonisel on projekteeritud olemite atribuudid. Atribuut on olemit kirjelduse madalaim tase, millede kogum kirjeldab olemit struktuuri. Iga atribuudi juures tuleb kirjeldada ka atribuudi semantika - kirjeldus, milles määratakse, milliseid väärtusi kirjeldatav atribuut võib omandada, millal ta neid väärtusi võib omada ja mille tulemusena need väärtused muutuvad (kui nad muutuda võivad).

Sellega projekteerimine veel ei lõppe. Selleks, et andmeloogilise mudeli alusel luua realselt eksisteeriv andmebaas, tuleb andmeloogilise kirjelduse alusel viia läbi füüsiline modelleerimine. Kuni selle hetkeni läbi viidud modelleerimine on olnud üsna sõltumatu reaalsest andmebaasisüsteemidest, mida tulevikus rakendatakse loodud mudeli alusel reaalse andmebaasi loomiseks. Füüsilise modelleerimise käigus saab andmemudelist andmebaasi skeem so. skemaatiline kirjeldus, mille alusel saab luua andmebaasi.

Füüsilise modelleerimise käigus lisatakse andmeloogilise andmemudelile (minimaalselt) järgmised kirjeldused:

**olemit atribuut kui
olemit semantika
täpsustaja**

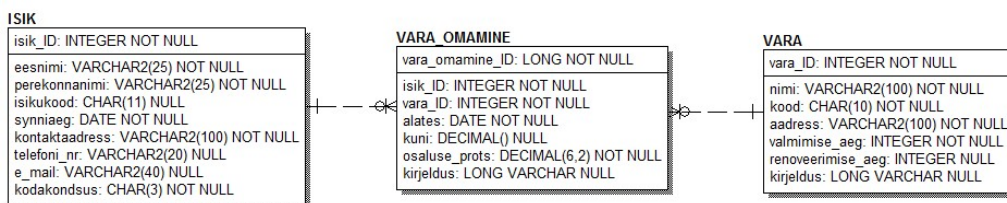
**andmeloogiline
modelleerimine
olemit omadused ja
atribuudid**

NÄIDE

atribuudi semantika

**füüsiline
modelleerimine**

- olemitele tabelite nimed, mille all olemite kirjelduse järgi luuakse andmebaasi tabelid
- olemitele seoste võtmete kirjeldused (primaarvõtmed ja välisvõtmed)
- olemitele indeksite kirjeldused
- olemite atribuutidele veergude nimed, mille all olemite kirjelduse järgi luuakse andmebaasi tabelite veerud
- olemite atribuutidele andmetüübid, pikkused ja NULL NOT NULL piirangud

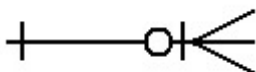


NÄIDE

Siin joonisel on näha, et kõikidele atribuutidele on lisatud andmetüübid, pikkused ja NULL/NOT NULL piirang, mis määrab, kas antud atribuut võib olla mõnes kirjes tühi (NULL) või mitte (NOT NULL). Tabelitele on kirjeldatud ka indeksid, kuid need kirjeldused ei paista skeemi peal välja.

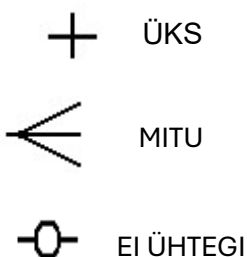
4.3. Seosed olemite vahel. Üks-mitmene seos

Valdav osa (üle 99%) seostest olemite vahel on nn. just sellised "üks-mitmene" seosed (üks - null, üks või mitu):



Joonisel toodud sümbol on võetud metoodikast mida tuntakse IE (*Information Engereering*) all . On veel teisi tähistusi, kuid nendest räägime hiljem. Siin jaotuses toodud notatsioon on enim kasutatud ja ilmselt ka parim seni välja töötatud notatsioonidest.

Igal komponendil selle suhte kujutise peal on oma kindel tähendus:



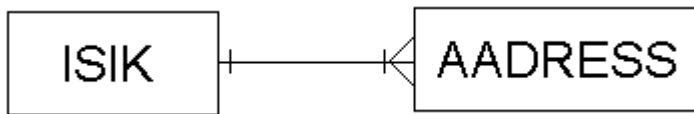
James Martin Group

IE - Information Engereering

üks, mitu või ei ühtegi

Nendest komponentidest saab kokku panna mitmeid erinevaid kombinatsioone, milledest igal on oma sügav tähendus. . Vaatleme neid kombinatsioone üks haaval. Et oleks lihtsam esile tuua iga konkreetse variandi iseärasusi kasutame lihtsat näidet isikute sidumisest aadressidega.

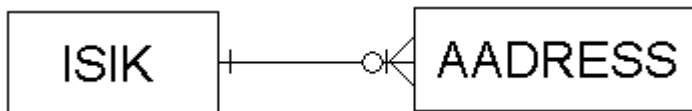
1.



seos: üks - üks või mitu

Skeemil on kirjeldatud inimene (inimesed), kellele kohta on teada igal juhul vähemalt üks aadress, aga tema kohta võib neid olla teada ka mitu. Ei ole võimalik, et aadress eksisteerib iseseisvalt – ilma et ta oleks seostatud isikuga. Ei ole võimalik, et eksisteerib isik, kelle kohta pole teada ühtegi aadressi.

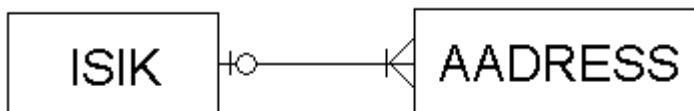
2.



seos: üks - null, üks või mitu

Skeemil on kirjeldatud inimene (inimesed), kes võib eksisteerida nii, et tema kohta pole teada ühtegi aadressi. Me võime tema kohta siiski teada ühte või enam aadressi. Ei ole võimalik, et aadress eksisteerib iseseisvalt – ilma et ta oleks seostatud isikuga.

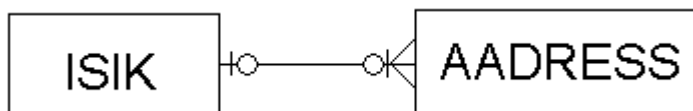
3.



seos: null või üks - üks või mitu

Skeemil on kirjeldatud inimene (inimesed), kellele kohta on teada igal juhul vähemalt üks aadress, aga tema kohta võib neid olla teada ka mitu. On võimalik, et aadress eksisteerib iseseisvalt – ilma et ta oleks seostatud ühegi isikuga.

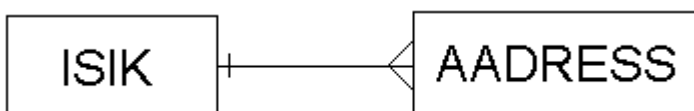
4.



seos: null või üks - null, üks või mitu

Skeemil on kirjeldatud inimene (inimesed), kes võib eksisteerida nii, et tema kohta pole teada ühtegi aadressi. Me võime tema kohta siiski teada ühte või enam aadressi. Aadress võib eksisteerida ka iseseisvalt – ilma et ta oleks seostatud isikuga.

5.

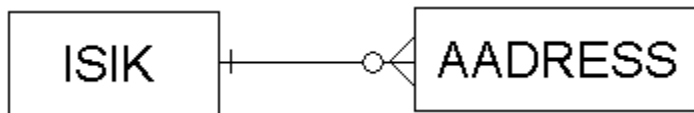


seos: üks - mitu

VÄGA HARVA ESINEV !!!

Skeemil on kirjeldatud inimene (inimesed), kellele kohta on teada igal juhul vähemalt kaks (mitu) aadressi, aga tema kohta võib neid olla teada ka rohkem. Ei ole võimalik, et aadress eksisteerib iseseisvalt – ilma et ta oleks seostatud isikuga. Ei ole võimalik, et eksisteerib isik, kelle kohta pole teada vähemalt kahte aadressi.

6.

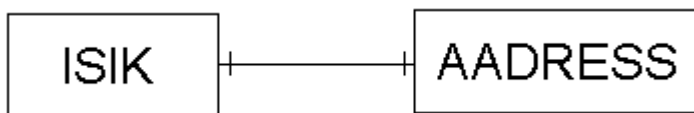


seos: üks - null või mitu

ERITI HARVA ESINEV !!!

Skeemil on kirjeldatud inimene (inimesed), kellele kohta pole teada ühtegi aadressi või kui neid mõni teada on, siis on neid vähemalt kaks (mitu), aga tema kohta võib neid olla teada ka rohkem. Ei ole võimalik, et isiku kohta on teada ainult üks aadress. Ei ole võimalik, et aadress eksisteerib iseseisvalt – ilma et ta oleks seostatud isikuga.

7.

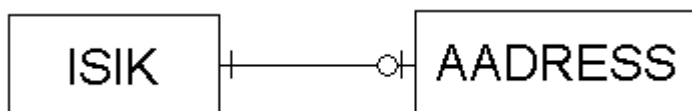


seos: üks - üks
ESINEB AINULT TEHNILISTEL KAALUTLUSTEL !!!

Skeemil on kirjeldatud inimene (inimesed), kellele kohta on teada alati ainult üks aadress. Ei ole võimalik, et isiku kohta on teada rohkem kui üks aadress või et tema kohta pole teada ühtegi aadressi. Ei ole võimalik, et aadress eksisteerib iseseisvalt – ilma et ta oleks seostatud isikuga. Nendes kahes olemis on alati võrdne arv kirjeid.

Sisuliselt on tegemist olukorraga kus loogiliselt ühe olemi atribuudid on jaotatud kahe olemi vahel. Üldjuhul tähendab see seda, et tegemist on kahe olemiga, mis asuvad erinevates andmebaasides. On veel ka võimalus, et olem on jagatud kaheks turva kaalutlustel - et oleks võimalik lubada/keelata erinevatel kasutajatel juurdepääsu erinevatele andmekooslustele ja tagada eriti kõrge turvalisuse tase füüsilise andmelahususega.

8.



seos: üks - null või üks
ESINEB AINULT TEHNILISTEL KAALUTLUSTEL !!!

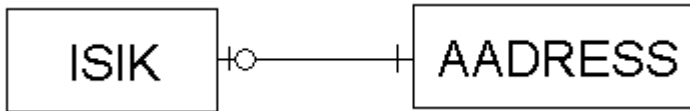
Skeemil on kirjeldatud inimene (inimesed), kellele kohta on teada kuni üks aadress. Ei ole võimalik, et isiku kohta on teada rohkem kui üks aadress. Ei ole võimalik, et aadress eksisteerib iseseisvalt – ilma et ta oleks seostatud isikuga. Olemis ISIK on alati kas sama arv kirjeid kui olemis AADRESS või on neid seal rohkem kui olemis AADRESS.

Nagu eelmise seose variandi korral, kasutatakse ka sellist seost ainult tehnilistel kaalutlustel. Põhjus on aga eelmisest seose variandist erinev. Sellist

seost kasutati palju siis, kui kõvaketaste maht oli väike ja seda tuli iga hinna eest kokku hoida. Seda õnnestus teha sellistel juhtudel, kui olemis oli atribuute, millele väärtused ilmnesisid harva. Sellisel juhul koondati harva esinevate väärtustega atribuudid eraldi olemisse aj seoti see põhi olemiga selliselt, et kirje seotud olemisse tekkis ainult siis, kui oli sinna mingeid väärtusi kirjutada.

Tühjade väärtustega kirjeid ei tekitatud ja seega hoiti mälu kokku.

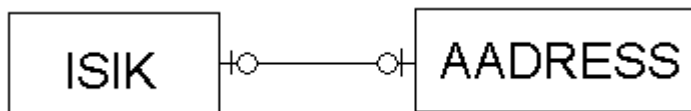
9.



Skeemil on kirjeldatud inimene (inimesed), kellele kohta on teada alati ainult üks aadress. Ei ole võimalik, et isiku kohta on teada rohkem kui üks aadress või et tema kohta pole teada ühtegi aadressi. Aadress võib eksisteerida iseseisvalt – ilma et ta oleks seostatud isikuga. Olemis AADRESS on alati kas sama arv kirjeid kui olemis ISIK või on neid seal rohkem kui olemis ISIK.

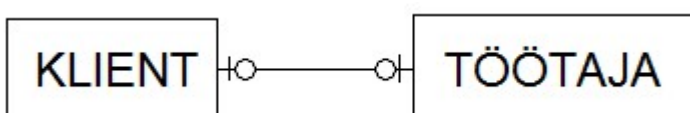
Tegemist on täpselt sama seosega, mida on kirjeldatud punktis 8 ainsa vahega, et see on olemite vahel teist pidi. Ka kasutamise põhjendus on sama

10.



Skeemil on kirjeldatud inimene (inimesed), kellele kohta on teada kuni üks aadress. Ei ole võimalik, et isiku kohta on teada rohkem kui üks aadress. Aadress võib eksisteerida iseseisvalt – ilma et ta oleks seostatud isikuga.

Seda seost kasutatakse tehnilisel kaalutlusel samadel põhjustel kui punktides 8. ja 9. kirjeldatud suhteidki. Samas leidub siin hoopis sisukam kasutamise põhjus. Seda seost kasutatakse siis, kui mingi olem on rolli või mingi muu tunnuse alusel jagatud osadeks. Sellisel juhul tekkib vajadus sama reaalse maailma objekti sisestamiseks kõikidesse olemitesse. Kui nüüd osutub vajaduseks näidata, millised objektid erinevates olemites on samad, siis antud seosega saabki seda teha. Selle illustreerimiseks sobib väga hästi olukord, kus andmemudelil on jagatud kaheks osaks olem ISIK - KLIENDID ja TÖÖTAJAD:



Kuna ettevõtte töötajad võivad olla ka sama ettevõtte kliendid, siis sellisel juhul on sama isiku andmed registreeritud mõlemas olemis. Kui nüüd soovitakse näidata, millised on erinevates olemis olevate samade inimeste andmed, siis seostatakse sama inimese andmed mõlemas olemis.

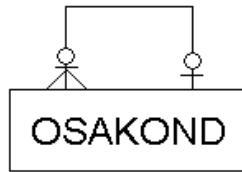
**seos: null või üks - üks
ESINEB AINULT
TEHNILISTEL
KAALUTLUSTEL !!!**

**seos: null või üks - null
või üks**

**olemi jaotamine
kaheks olemiks mingi
tunnuse (näiteks rolli)
alusel**

Skeem kirjeldab olukorda, kus kõik firma töötajad pole sama firma kliendid ja kõik firma kliendid pole sama firma töötajad. Samas vastupidi asja vaadates on osad firma töötajad ka sama firma kliendid ja osad firma kliendid sama firma töötajad.

11.



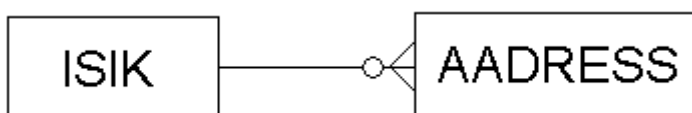
rekursiivne seos

Iga osakond võib olla teise osakonna alguses (aga ei pruugi). Igal osakonnal võib olla suvaline hulk alluvaid osakondi, aga ei pruugi olla. Tühja seose võimalikkust tähistavad "mullid" PEAVAD siin olema mõlemas suhte otsas sest – kõige alumisel tasemel on osakonnad, millele ei allu enam osakondi ja kõige ülemisel tasemel on osakonnad, mis ei allu ühelegi teisele osakonnale. Kui tegemist on juhtimispiramiidiga, siis on ülemus-osakondi, mis ei allu ühelegi teisele osakonnale, ainult üks.

Siin tuleb tähelepanu juhtida sellele, et rekursiivne seos saab olla ainult "null või üks - null, üks või mitu" ja MITTE KUNAGI teistsugune. Antud näites on alati olemas vähemalt üks osakond, millel pole ülemust (näiteks "Juhatus") ja alati on teatud hulk kõige madalamal tasemel olevaid osakondi, millele ei allu enam ühtegi teist osakonda.

Nüüd mõned näited seostest, mida ei ole olemas:

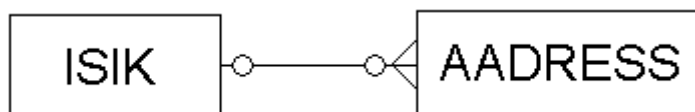
1.



määratlemata seos

Seose otsas "üks" puudub "kriips", mis tähistab "ühte" ja seega on seos määratlemata (olenemata sellest, mis on seose "mitmeses" otsas). Vahel kasutatakse seda siiski ka lihtsalt määratlemata seose kirjeldamiseks. Seda skeemi joonistamise väga varajases staadiumis tähistamaks seda, et kõik ei ole veel lõpuni läbi mõeldud

2.



olematu seos

Seose otsas "üks" ei saa olla ainult seose puudumise võimalikkust tähistav "mull" (olenemata, mis on seose "mitmeses" otsas). Praegusel kujul on

tegemist seosega, mida kunagi ei eksisteeri, kuna tema otsas "üks" eksisteerib ainult variant, kus seos puudub.

Sellest notatsiooniost on ka „vabameelsemaid“ tõlgendusi. Näiteks harjutustundides kasutatav QSEE ei pea „nulli“ näitava seose otsa „mitte ükski“ puhul oluliseks „ühte“ näitavat püstist kriipsu vaid eeldab, et null tähendabki ka ühte st 0 või 1. Sama on ka mitmese otsa korral. Kui mitmese otsa pole valida „mitte ühtegi“, siis ühte näitav kriips kaob ja tekkib justkui seos „null või mitu“. Tegelikult tähendab see „null kuni mitu“, mille vahele jääb ka ju üks. Just seda „mitte lubatavat“ seost QSEE notatsioonis me näemegi.

4.4. Mitu mitmene seos

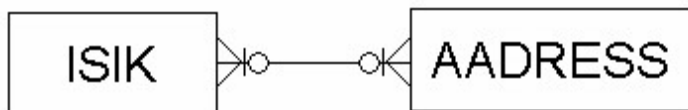
Kui vaadata eelmise jaotise alguses kirjeldatud sümbolikat olemite vaheliste seoste kirjeldamiseks, siis tekkib kindlasti küsimus, kas pole võimalik ehk joonistada mõnda sellist seost



Sellele küsimusele võib vastata nii "ei" kui "jah". "Jah" tähendab seda, et väga kõrge üldistuse tasemega infooloogilistes mudelites võime me seda kasutada.

"Ei" peame ütlema aga siis, kui üritatakse kasutada sellist seost andmebaasi füüsilise skeemi kirjeldamisel - andmebaasides sellist seost vähemasti sellisel kujul realiseerida ei õnnestu.

Vaatame seda asja nüüd lähemalt. Võtame kasutusele jälle kas olemit ISIK ja ADDRESS:



On ju täiesti loogiline, et ühel isikul võib olla üks või mitu aadressi ja samas ei pruugi tal neid olla ühtegi. Teiselt poolt võib samal aadressil elada üks või mitu isikut aga samas on kindlasti aadresse kus ei ela kedagi. Kuna infooloogilise mudeli kirjeldamisel on olulisim semantika, siis võime me loomulikult selliseid seoseid kasutada.

Andmebaasi mudelite kirjeldamisel on aga asi sootuks teistsugune - siin peab lisaks toimivale semantikale olema olemas veel toimiv füüsiline struktuur.

Sellisel kujul joonistatud mitu-mitmest suhet ei ole võimalik andmebaasis aga

QSEE notatsiooni erinevused

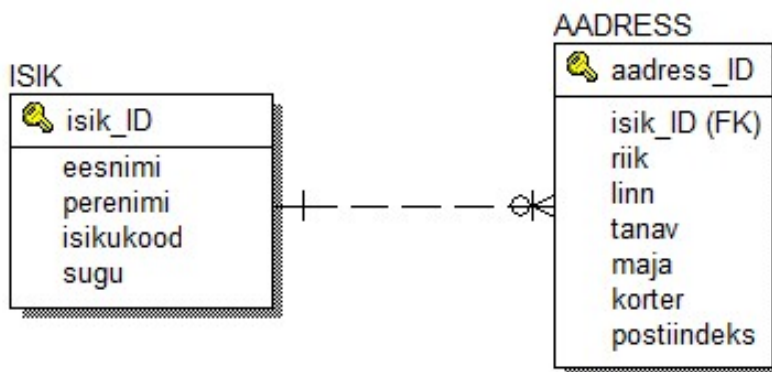
selliseid seoseid ei ole andmebaasides tegelikult olemas !

mitu-mitmene seos infooloogilises mudelis

erinevus füüsilises andmebaasi mudelis

kirjeldada. Miks? Vaatame asja lähemalt.

Kui kaks tabelit on andmebaasi omavahel seotud, siis viiakse suhte "üks" otsas oleva tabeli primaarvõti üle suhte "mitu" poolses olevasse tabelisse:



Nagu näha on tabeli ISIK primaarvõti olemas ka tabelis address aga siin on ta välisvõtmena (FK - foreign key). Kui tabelis ISIK on isik_ID unikaalne, siis tabelis AADDRESS pole ta seda mitte. Tabelis ISIK olev iga kirje saabki olla tabeli AADDRESS mitme kirjega olla seotud seeläbi, et igas temaga seotud kirjes on veerus isik_ID selle tabeli ISIK kirje isik_ID väärtus, millega tabeli AADDRESS kirje seotud on. Siit võime tuletada reegli üldistuse selle jaoks, kuidas seos tekitatakse - seose tekitamisel kahe tabeli vahel toimub selliselt, et seose "üks" poolses otsas oleva olemi primaarvõtme väli kantakse üle suhte "mitu" poolsesse tabelisse.

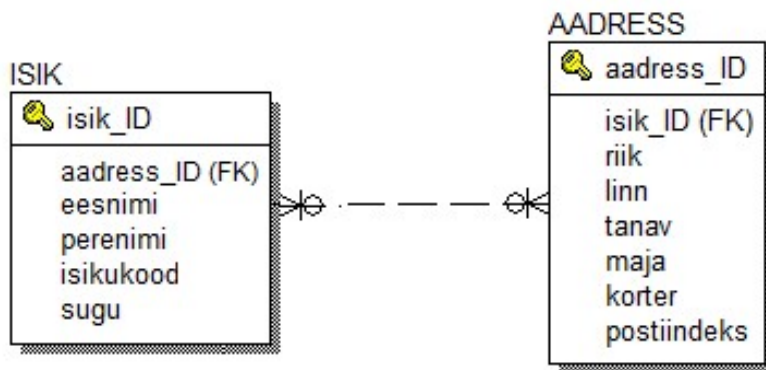
Mõtleme, mis nüüd juhtub, kui me teeme suhte, mille mõlemad otsad on tüüpi "mitu". Kuna suhte "üks" poolne ots kadus ära, siis ei saa me rakendada just äsja välja mõeldud reeglit - pole lihtsalt suhte "üks" poolset otsa ja seega ei saa enam selle reegli järgi valida suhte liiget mille primaarvõti kantakse üle teise tabelisse. Sedasi ka ei saa, et ühe tabeli primaarvõtit teise tabelisse üldse üle ei kannu, sest siis ei tekki mingit seost kohe kindlasti. Kuna mõlemad suhte otsad on võrdväärsed ja kuna ka suhte "mitmeses" otsas realiseerub ka vahel seos "üks", siis ainule loogiline järeldus võiks olla see, et tuleb mõlema tabeli primaarvõtmed viia üle teise tabelisse. Vaatame, mis sellest saab:

üks-mitmese seose olemus

seos: PK = FK

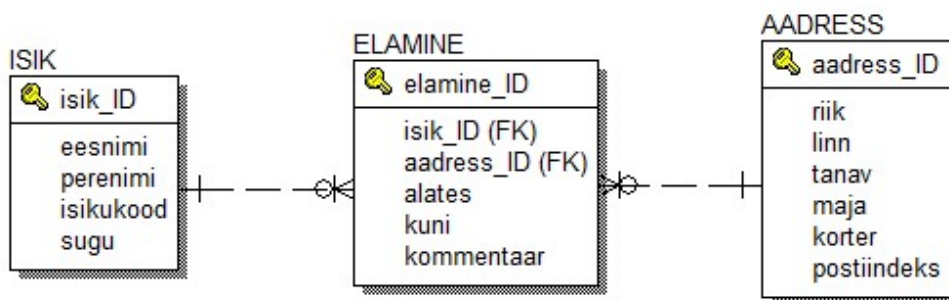
PK - primary key (primaarvõti)

FK - foreign key (välisvõti)



Kui nüüd üritada sellise skeemi abil teha mitu-mitmest seost, siis märkame, et saame mõlemale suhte poolele kirjutada ainult ühe teise poole ID väärtuse ja oleme seega sisuliselt loonud üks-ühese suhte. Kuidas siis asja lahendada? Mitu-mitmese seose kirjeldamiseks andmebaasis tuleb meil seos tabelite ISIK ja aadress vahel asendada uue tabeliga, mis on seotud mõlema tabeliga. See tabel kirjeldab mitu-mitmest seost. Selle tabeli semantikaks on selle seose semantika, mis tabeliga asendati. Uued seosed saavad aga veidi muudetud semantika:

mitu-mitmese seose asendamine tabeli ja kahe uue seosega



Uus tabel on seotud olemasolevate seostega teist pidi kui seda vanad tabelid oma vahel. Kui enne osutasid suhted "mitmese" otsaga tabeleid ISIK ja AADRESS, siis nüüd on tabelite ISIK ja AADRESS poole seoste "ühesed" otsad. Vaatame milline on uute seoste semantika. Tabelite ISIK ja ELAMINE vahel oleva seose semantika on "üks isik võib elada ühes või paljudes korterites või ei ela ta kusagil". Tabelite ELAMINE ja AADRESS vahel oleva seose semantika on "samal aadressil võib elada üks või palju inimesi või ei ela seal kedagi". Kui me nüüd võrdleme neid semantikaid enne suhte asendamist olemas olnud suhte semantikaga, siis märkame, et see semantika on just kui pooleks tehtud - suhte erinevate otste juures kirjelduv semantika on üle antud sellele seosele, mis asendab vana suhte vastavat otsa.

jaotatud semantika

Mida kirjeldab nüüd siis uus tabel? Uus tabel kirjeldab seda, milline isik on millise aadressiga seotud ja millises ajavahemikus (alates - kuni). Siin ongi tegelikult selle asja mõte - mitu-mitmeste seoste korral on tähtis seoses olemise aeg ja vähemasti seda ei saanud me ilma uut tabelit lisamata kuidagi registreerida.

Iga kord, kui isiku ja aadressi vahele tekib seos kirjeldatakse see tabelis ELAMINE selliselt, et näidatakse ära seotud isiku ja aadressi ID-d, ning kirjutatakse juurde ka see, mis ajast mis ajani seos kehtib - millal inimene selle aadressil elas või elab. Kui algus-kuupäev on minevikus ja lõpu kuupäev on määramata, siis on tegemist kehtiva seosega. Kui mõlemad kuupäevad on minevikus, siis on tegemist minevikus kehtinud seosega. Kui alguskuupäev on tulevikus on tegemist tulevikus jõustuva seosega.

mitu-mitmese seose tähendus

seose kehtivus

4.5. ERD skeemide erinevad notatsioonid

Eelnevates jaotistes toodud näidetes kasutasime ERD skeemide joonistamiseks IE (Information Engineering) sümboolikat (notatsiooni). See pole aga sugugi ainuke ERD notatsioon maailmas. Mõningase aimu andmiseks teistest notatsioonidest toome siin kohal lühikeste näidetena ära mõned tähtsamad:

IDEF1 (*Integration DEFINition for Information Modeling*)



IDEF1

See notatsioon on tähtis selle poolest, et tegemist on ISO standardiga.

Täpsemalt saab selle notatsiooni kohta lugeda aadressilt <http://www.edef.com>

Peter Chen'i notatsioon



Peter Chen'i notatsioon

Peter Cheni ERD notatsioonist saab täpsemalt lugeda aadressilt

<https://www.conceptdraw.com/examples/what-is-chen-notation>

Oma kursuses kasutame andmemudelite joonistamiseks metoodilistel kaalutlustel Crow's Foot notatsioone, milledest üks on ka IE. Crow's Foot notatsioonid visualiseerivad üks-mitmeseid seoseid õpetamise jaoks kõige paremini.

Õppetöös kasutame
Crow's Foot
notatsioone

4.6. ERD Case-süsteemid

Nagu juba ees pool mainitud on ERD Case süsteemid ette nähtud andmemudelite (ERD) joonistamiseks ja formaalseks kirjeldamiseks. ERD Case-süsteemide peamine funktsionaalsus on järgmine:

ERD Case-süsteemide
funktsionaalsus

1. andmebaaside skeeme mugav loomine ja muutmine (olemid, olemite atribuudid, olemite vahelised seosed)
2. Kirjeldada andmemudeli füüsilisi parameetreid selleks, et nende alusel hiljem genereerida korraldused andmebaasi struktuuri loomiseks ja muutmiseks (tabelite nimed, veergude nimed, andmetüüp, pikkus, kohustuslikkus, andmekontrolli tingimused, esituskirjeldused, indeksite kirjeldused, vaadete kirjeldused jne.)
3. andmemudeli erinevate komponentide semantika kirjeldamine
4. erinevate raportite genereerimine loodud skeemide dokumenteerimiseks
5. suurte skeemide osadeks jagamine ja nende osade seostamine
6. genereerida loodud mudeli alusel andmebaaside loomise korralduste jadasid – samast skeemist saab genereerida korralduste jada erinevate andmebaasisüsteemide jaoks
7. sünkroniseerida mudeli muudatused andmebaasidesse (muuta graafilise kirjelduse alusel andmebaaside struktuure) ja vastupidi (korrigeerida skeemi baasis tehtud muudatuste alusel)
8. andmebaasidest kirjelduste välja lugemine ja esitamine skeemina (olemi-suhte diagrammina)
9. andmemudelite ühe andmebaasisüsteemi alt teise alla konverteerimine
10. Töögruppide töö sünkroniseerimine
11. Esitlusmaterjalide genereerimine

Mõned ERD Case-süsteemid (süsteemi nimi / firma):

ErWin / Quest Software

ER/Studio / Embarcadero

SAP PowerDesigner / Novalys

Enterprise Architect (EA) / Sparx

**Eestis kasutatavad ERD Case-
süsteemid:**

erwin.com

erstudio.com

powerdesigner.biz

sparxsystems.com/products/ea/