

ESYBIŲ RYŠIŲ MODELIS

Esybės

Esybė – susijusios informacijos rinkinėlis. Pati esybė yra tik informacijos aprašymas/specifikacija.

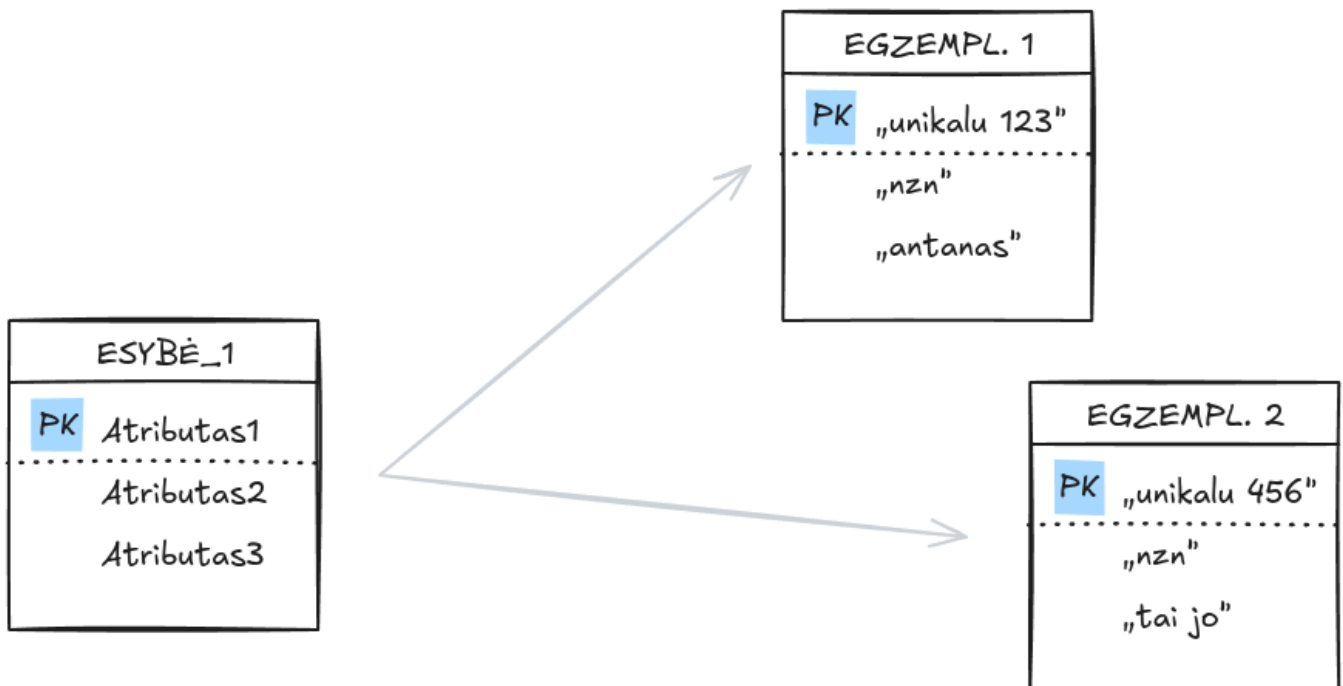
Python'e esybes atitinka klasės.

Esybės egzempliorius yra jau realūs, konkretūs duomenys sugrupuoti pagal kažkokią esybę.

Python'e egzempliorius atitinka objektai.

Atributai yra arba esybių, arba egzempliorių dalys, kurių daugiau suskaidyti neišeina: skaičiai, tekstas ir t.t.

Python'e atributus atitinka klasių kintamieji.



Taisyklės:

1. Atributai negali būti sąrašais. (tam yra vienas-su-daug ryšiai)
2. Esybių pavadinimai yra vienaskaita (nebent neleidžia lietuvių kalba)
3. Microsoft Visio, po pirminiais raktais (kažkodėl) turi būti punktyrinė linija...

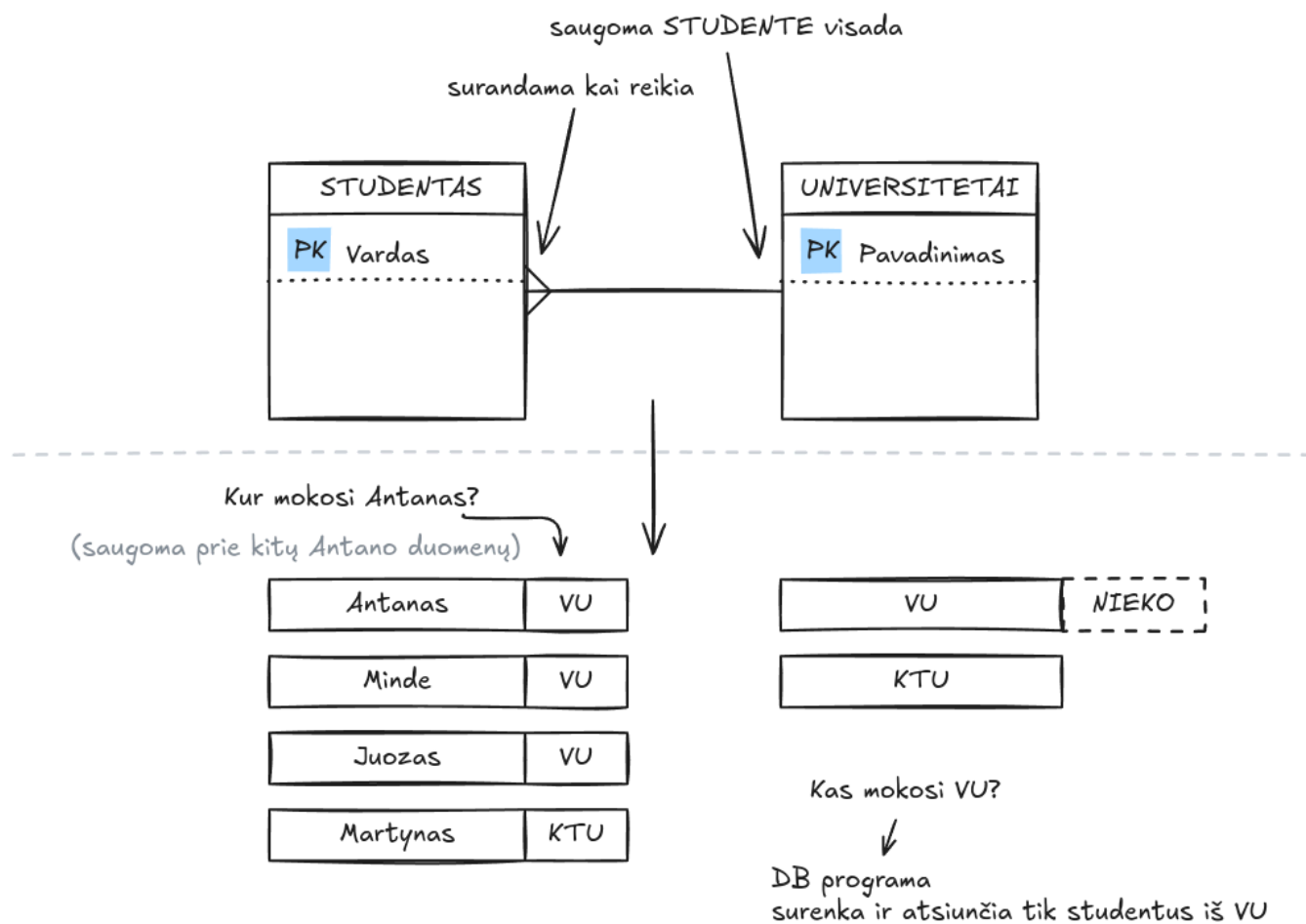
Ryšiai

Ryšys – šizofrenijos priepuolis?.. nežinau... Šiaip, ryšiai reiškia, kad pagal vienos esybės egzempliorių galima gauti kitų esybių egzempliorius.

Pirminis raktas (PK) – vienas ar daugiau atributų, pagal kuriuos galima atskirti egzempliorius.

Pirminis raktas gali būti „dirbtinis“ (tiesiog kodas/numeris), arba susidaryti iš jau esančių atributų, pvz.: žmogaus vardo, miesto ilgumos ir platumos koordinatų ir t.t.

Kas keisčiausia, tai kad viena ryšio pusė yra (mums slapčiom) saugoma prie kitų egzempliorio duomenų (tipo esybės „A“ egzempliorius saugo esybės „B“ egzempliorio PK)... O kita pusė, GALI būti: 1. surasta programatiškai (kai reikia) arba 2. irgi saugoma.



Identifikuojantys ryšiai leidžia naudoti kitų esybių egzempliorius, kad gautume norimą egzempliorių.

Esybės, kurios turi identifikuojančius ryšius, gali neturėti pirminio rakto.

Pavyzdžiui: jei turime

1. esybę *DYDIS* su atributais: *Ilgis*, *Plotis* ir *Aukštis*,
2. esybes *VAIZDO_PLOKŠTĖ* ir *VENTILIATORIUS*,
3. identifikuojančius ryšius tarp: *VAIZDO_PLOKŠTĖ* – *DYDIS* ir *VENTILIATORIUS* – *DYDIS*

Turėdami vaizdo plokštę arba ventiliatorių galime sužinoti jų dydžius.

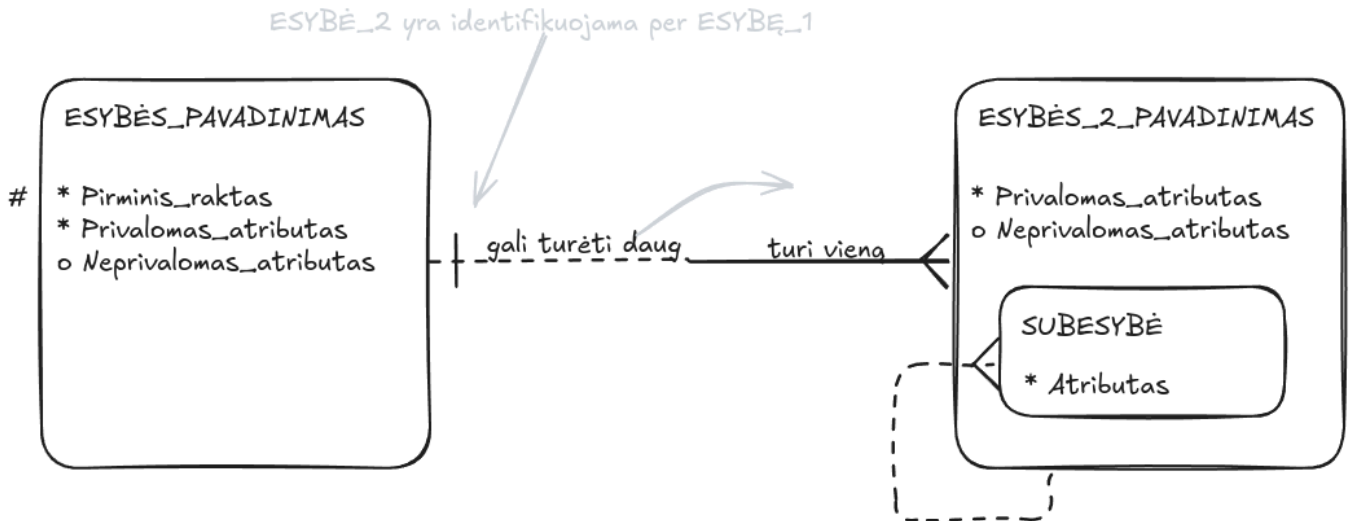
Turėdami atskirą dydį galime jį išmesti į šiukšlių dėžę...

Žymėjimas

Mes **žymime** tą pačią diagramą dviem skirtingais būdais...

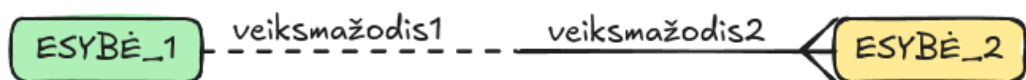
Iš pradžių maniau, kad Microsoft vėl pasiūkšlino, tačiau [čia](#) yra tas pats žymėjimas kaip Visio, iš Oracle(ERD kūrėjų) 1999 metais. Be to, aš naudoju Microsoft Visio 2010 metų ir dar net tada ten nebuvo „Crow’s foot“ diagramos, tai...

Žemiau yra dėstytojos žymėjimas, kurį, jei neklystu, naudosim per kontrolinį darbą:



Esybių ryšių „skaitymas“ – ryšio perteikimas tekstu, pagal formulę:

Kiekviena(s) (ESYBĖ_1) turi (veiksmažodis) vieng
gali vieng ar daugiau (ESYBĖ_2)



Kiekviena(s) (ESYBĖ_1) gali (veiksmažodis1) vieng ar daugiau (ESYBĖ_2)

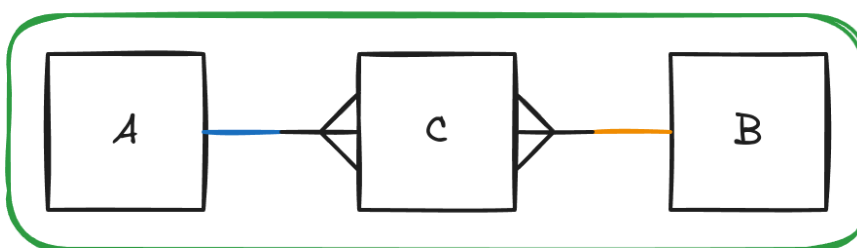
Kiekviena(s) (ESYBĖ_2) turi (veiksmažodis2) vieng (ESYBĖ_1)

„(veiksmažodis)“ gali būti ir ne vienas žodis, ir, pavyzdžiui, dalinai nueiti už „ESYBĖ_2“.

n:m ryšys

Daug-su-daug ryšio sprendimo metu reikia sukurti naują esybę su, dažniausia, dviem identifikuojančiais ryšiais.

Nauja esybė GALI turėti savo atributų, pavyzdžiui: naują kiekį ir t.t.



Dėstytojos pavyzdžiuose identifikuojančių brūkšniukų nėra. Manau, kad kažkas tiesiog užmiršo...

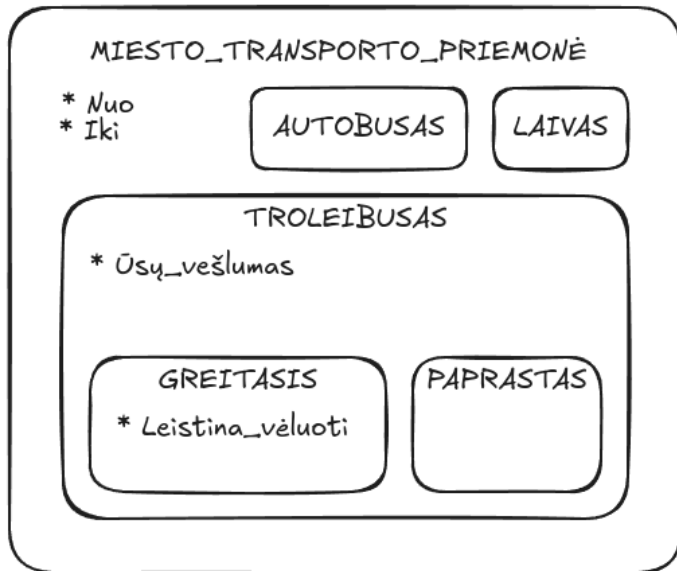
Subesybės

Subesybės yra vienos kitoms prieštaringos* esybės sukištos į „superesybę“, kurios paveldi šios „superesybės“ atributus.

*„vienos kitoms prieštaringos“ = „dvi negali egzistuoti tuo pačiu metu“ = „arba viena, arba kita“.

Subesybės dažnai atsiranda, kai mes turime kelias iš anksto žinomas kažkokios esybės kategorijas.

Pavyzdžiui:



MIESTO_TRANSP_PRIEMONĖ gali būti:

AUTOBUSAS { Nuo, Iki },
LAIVAS { Nuo, Iki }

arba

TROLEIBUSAS, kuris gali būti:

GREITASIS { Nuo, Iki, Ūsai, Vėluoti }
arba
PAPRASTASIS { Nuo, Iki, Ūsai }

Programavime, subesybų atitikmuo yra „union“ arba sumos tipai...

O čia dar gali būti ir tie lankai ant ryšių (kurių, greičiausia, nereiks)

Normalizavimas

Normalizavimas yra rinkinys taisyklių, kuris padeda nesuklysti planuojant duomenų bazes. Normalizavimas, šiaip, ir reiškia: „supaprastinti kažką į normalesnę/standartiškesnę formą (pagal tam tikras taisykles).“

Iš atributų kurie neatitinka tikrinamosios norminės reikia padaryti naujas esybės arba pan., kol galiausia viskas visur atitiks tą norminę formą.

Pirmos trys norminės formos:

1NF: „Visi atributai yra vienareikšmi.“

Tipo, atributuose negalima saugoti sąrašų, nes, durnai atrodo, kai lentelės langely kabelių prikaišyta...

2NF: „Visi esybės atributai yra priklausomi nuo viso esybės unikalaus identifikatoriaus.“

Jeigu esybė turi tik vieną raktą, tai ji iš karto atitinka 2NF.

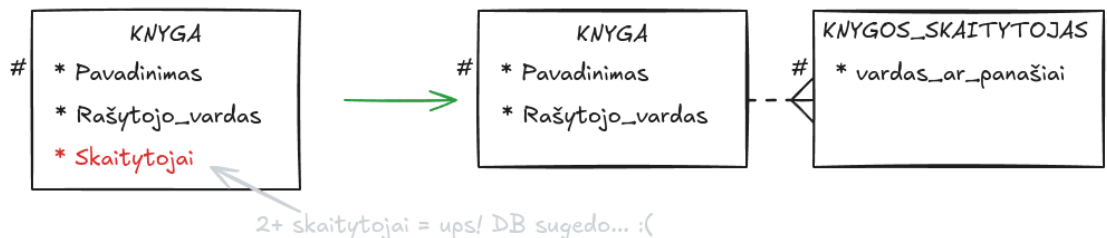
Kitaip, reikia tikrinti kiekvieną ne raktinį atributą ir žiūrėti:

ar tikrai, kad jį gautume mums REIKIA visų raktų ir identifikuojančių ryšių?

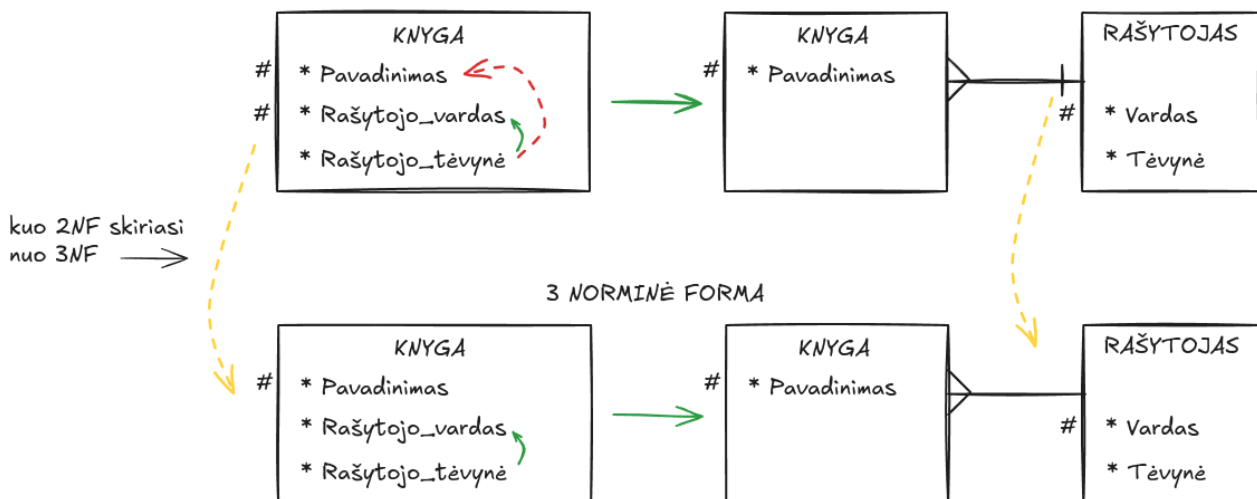
3NF: „Esybė neturi atributo nepriklausančio UID, kuris priklausytų nuo kito atributo nepriklausančio UID.“

Netinka tada, kai paprasti atributai gali priklausyti nuo kitų neidentifikuojančių/ne kandidatų į pirminius raktus.

1 NORMINĖ FORMA



2 NORMINĖ FORMA



3 NORMINĖ FORMA



Ar taip ar taip, nežinot dviejų knygų, juo labiau, su tuo pačiu pavadinimu... Nesistenkit.