

Reliacinis modeliavimas

Reliacinį modelį (arba reliacinį duomenų modelį) 1970 m. pasiūlė E. F. Kodus (*angl.* Edgar Frank Codd). Nuo tų laikų šis modelis tapo vienas iš vyraujančių DB kūrimo modelių. Pagrindines reliacinio modelio idėjas E. F. Kodus išdėstė straipsnyje „A Relational Model of Data for Large Shared Data Banks“, kuris laikomas klasikiniu darbu šioje srityje.

Prieš pateikdami reliacinio modelio apibrėžimą, trumpai apžvelgsime reliacinio modelio terminologiją:

- Sąryšis (*angl.* relation) – viena iš pagrindinių sąvokų. Kartais kaip sinonimą vietoje sąvokos „sąryšis“ naudojama žymiai dažniau vartojama sąvoka „lentelė“.
- Kortežas – sąryšio eilutė.
- Atributas – kolonėlė.
- Sąryšio laipsnis – atributų (kolonėlių) kiekis.
- Sąryšio kardinalumas – kortežų (eilučių) skaičius.

Reliacinis duomenų modelis yra loginis duomenų modelis. T.y. taikomoji duomenų bazių kūrimo teorija, kuri remiasi tokiais matematikos skyriais kaip aibių teorija ir pirmos eilės logika.

Reliacinis duomenų modelis yra reliacinių duomenų bazių konstravimo pagrindas.

Pagrindiniai reliacinio duomenų modelio aspektai:

- Struktūrinis aspektas – duomenys duomenų bazėje yra sąryšių rinkinys.
- Darnos aspektas – sąryšiai tenkina tam tikrus darnos reikalavimus. Darnos reikalavimai taikomi duomenų tipo (domeno) lygmenyje, sąryšio lygmenyje ir duomenų bazės lygmenyje.
- Manipuliavimo aspektas – reliacinis duomenų modelis palaiko sąryšių manipuliavimo operatorius (reliacinė algebra, reliaciniai skaičiavimai).

Dažnai į reliacinį modelį taip pat jungiama ir normalizacijos teorija. Vėliau šiame kurse nagrinėsime normalizaciją iki trečios normalinės formos.

Reliacinis modelis yra populiarus ir plačiai taikomas dėl šių priežasčių:

- Modelis gerai suprantamas tiek vartotojams, tiek ir kūrėjams.
- Modelis gali būti transformuotas į kitus duomenų modelius.
- Modelis nėra orientuotas į kompiuterinę terpę.
- Modelis yra kilęs iš duomenų analizės.
- Skirtingi vartotojai gali naudotis modeliu skirtingais požiūriais.

Norint geriau suprasti reliacinį modelį, reikia akcentuoti 3 svarbias modelio savybes:

- Modelis yra loginis, tai reiškia, kad sąryšiai yra loginės (abstrakčios), o ne fizinės (saugomos) struktūros.
- Reliacinės duomenų bazės turinys pateikiamas vienu ir tik vienu būdu – atributų reikšmės sąryšiuose užduodamos tiesiogiai, t.y. nenaudojamos rodyklės (adresai), kurios surištų vieną reikšmę su kita.
- Reliacinės algebros naudojimas modelyje leidžia kartu su procedūriniu programavimu realizuoti ir deklaratyvinį programavimą.

Griežtas reliacinių duomenų bazių teorijos aprašymas pateiktas K. Dž. Deito (*angl.* C. J. Date) knygoje „An Introduction to Database Systems“.

Reliacinio modelio alternatyvos:

- Hierarchinis modelis.
- Tinklinis modelis.
- Objektiškai-orientuotas modelis.

Esybių-ryšių modeliavimas

Vienas iš duomenų analizės metodų yra esybių-ryšių modeliavimas. Modelis remiasi tam tikra semantine informacija apie realų pasaulį ir skirtas loginiam duomenų atvaizdavimui. Šio modelio pagrindu gali būti formuojami visi šiuo metu naudojami duomenų modeliai (reliacinis, hierarchinis, tinklinis, objektinis).

Pagrindinės metodo sąvokos yra „esybė“ (*angl. entity*) ir „ryšys“ (*angl. relation*). Panagrinėkime šitas sąvokas plačiau.

Esybėmis vadinamos realaus pasaulio dalykų klasės, t.y. visa tai, ką galima apibūdinti bendriniais daiktavardžiais. Skiriamos fizinės (daiktai, reiškiniai, procesai, subjektai, struktūros ir pan.) ir koncepcinės (sąvokos, idėjos ir pan.) esybės.

Konkrečios esybių realizacijos, (t.y. konkretūs daiktai, reiškiniai, procesai ir pan.) vadinami esybių egzemplioriais. Pvz., ASMUO yra esybė, o konkretus asmuo, tarkime, Vardenis Pavardenis – esybės ASMUO egzempliorius.

Kad geriau suprastume esybių ir ryšių tarp jų sąvokas, panagrinėkime pavyzdį. Sakykime universitete studijuoja studentai, kurie išskirstomi į grupes. Kiekvienas studentas studijų metu gali pasirinkti įvairius modulius. Pabandykime atvaizduoti tai esybių-ryšių pagalba.

Taigi pradžioje galime išskirti 3 skirtingas esybes:

- Grupė
- Studentas
- Modulis

Susitarkime, kad grafiškai esybes vaizduosime stačiakampiais. Tuomet mūsų esybes galime pavaizduoti paveiksle apačioje.



Dabar turime nustatyti ryšius tarp išskirtų esybių.

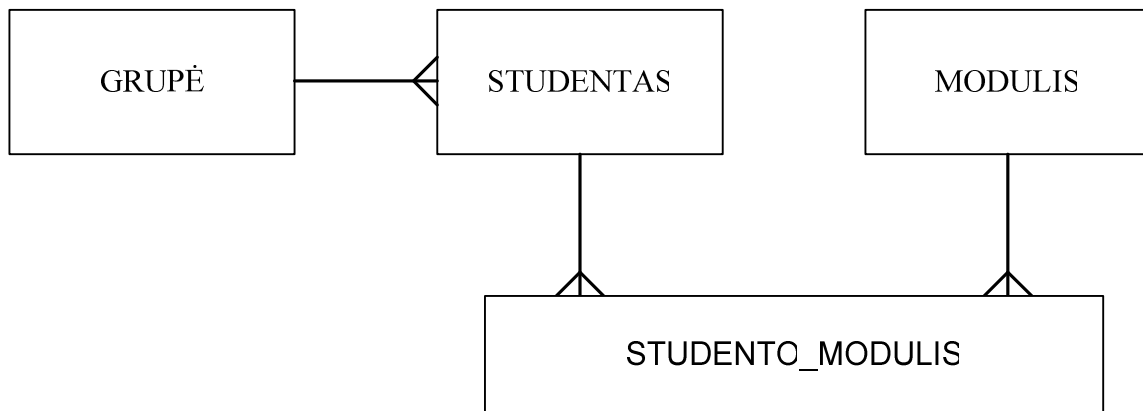
- Esybės GRUPĖ ir STUDENTAS. Vienoje grupėje gali studijuoti daug studentų, o vienas studentas gali būti priskirtas tik vienai grupei. Todėl ryšys tarp esybių GRUPĖ ir STUDENTAS yra „vienas-su-daug“.
- Esybės MODULIS ir STUDENTAS. Vieną modulį gali pasirinkti daug studentų, o vienas studentas gali studijuoti daug modulių. Todėl ryšys tarp esybių MODULIS ir STUDENTAS yra „daug-su-daug“.
- Esybės GRUPĖ ir MODULIS ryšio neturi.

Atvaizduokime nustatytus ryšius grafiškai:

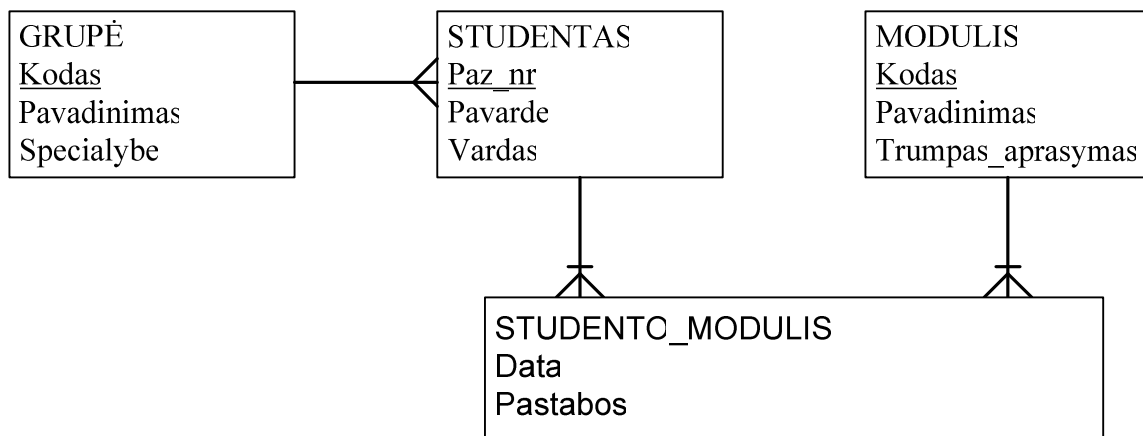


Detalizuojant esybes, ryšys „daug-su-daug“ tarp dviejų esybių turi būti pakeistas dviem ryšiais „vienas-su-daug“ ir trečia tarpine esybe.

Taigi dabar mūsų esybių-ryšių modelis atvaizduojamas taip:



Dabar pabandykime nustatyti kiekvienos esybės atributus:



Esybėje GRUPĖ išskyrėme 3 atributus:

- Kodus – raktinis esybės atributas, kurio reikšmė yra unikali kiekvienai grupei.
- Pavadinimas – grupės pavadinimas.
- Specialybė – grupės specialybė.

Esybėje STUDENTAS išskyrėme 3 atributus:

- Paz_nr – raktinis esybės atributas. Studento pažymėjimo numeris.
- Pavardė – studento pavardė.
- Vardas – studento vardas.

Esybėje MODULIS išskyrė 3 atributus:

- Kodas – raktinis esybės atributas, kurio reikšmė yra unikali kiekvienam moduliui.
- Pavadinimas – modulio pavadinimas.
- Trumpas_aprasymas – trumpas modulio aprašymas.

Esybėje STUDENTO_MODULIS išskyrė 2 atributus:

- Data – data, kai buvo pasirinktas modulis.
- Pastabos – atributas, leidžiantis pagal poreikį įrašyti pagalbinę informaciją apie įrašą.

Raktiniai atributai esybėse yra pabraukti. Kaip matome, esybėje STUDENTO_MODULIS nėra išskirta raktinių atributų. Bet tai nereiškia, kad DB lentelėje, kuri ateityje bus sukurta iš šitos esybės nebus unikalaus rakto. Unikalus raktas bus suformuotas realizuojant šiuos ryšius „daug-su-daug“ su esybėmis STUDENTAS ir MODULIS, todėl šitie ryšiai atvaizduoti su vienu papildomu brūkšniu. Plačiau apie tai bus paaiškinta vėliau.

Kalbant apie ryšius tarp esybių, nebuvo paminėtas ryšys „vienas-su-vienu“ (žymimas viena ištisine linija). Dažniausiai, jeigu gaunasi, kad tarp dviejų esybių yra ryšys „vienas-su-vienu“, esybės apjungiamos į vieną esybę, o ryšys naikinamas.

Esybių analizės etapai:

- Apibrėžti analizės sritį.
- Nustatyti esybes.
- Nustatyti ryšius tarp esybių.
- Nustatyti raktinius visų esybių atributus.
- Nustatyti visus kiekvienos esybės atributus.
- Normalizuoti visas esybes iki trečios normalinės formos.