

Duomenų bazės

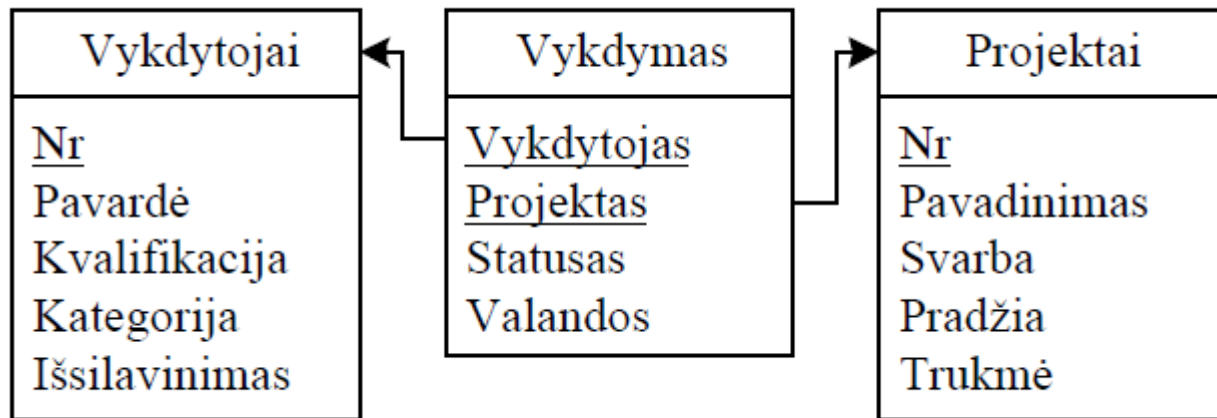
DB PROJEKTAVIMAS, IŠORINIAI RYŠIAI

DB reliacinės schemos grafinis vaizdavimas

DB reliacinę schemą galima pavaizduoti ir grafiškai. Grafiškai pavaizduota DB reliacinė schema tampa aiškesnė, ypač kai duomenų bazę sudaro daug išoriniais raktais tarpusavyje susijusių lentelių.

DB reliacinės schemos grafinis vaizdavimas

DB Darbai reliacinę schemą galima pavaizduoti taip:



Išorinių raktų apribojimai

DBVS, kuriose galimi išoriniai raktai, apibrėžimuose naudoja išlygas „ON DELETE CASCADE“ ir „ON DELETE RESTRICT“.

Išlyga „ON DELETE RESTRICT“ reiškia, kad įrašų ištrinti negalima, nebent bus ištrinti visi įrašai, susieti su išoriniu raktu. Ryšio sąskaitos eilutės-prekės atveju sąlyga ON DELETE RESTRICT sąskaitos eilučių lentelėje reiškia, kad prekė bus ištrinta tik ištrinus visas su ta preke susijusias sąskaitos eilutes. Taip išvengiama sąskaitos eilutės su nuoroda į neegzistuojančią prekę.

Sąlygos ON DELETE CASCADE rezultatas panašus, tik pasiekiamas labiau automatizuotu būdu (bet ir rizika didesnė). Jei išorinis raktas pateiktas su sąlyga ON DELETE CASCADE, ištrinus prekę bus ištrintos ir susietosios sąskaitos eilutės.

Sąlyga ON UPDATE CASCADE panaši į sąlygą ON DELETE CASCADE tuo, kad atnaujinus pirminį raktą, atnaujinamos ir visos išorinio rakto sąsajos su pirminiu raktu.

Ryšiai

Klasės susijusios tam tikrais ryšiais.

Pavyzdžiui, klientas gali priklausyti bibliotekai ir skolintis knygas. Knygą galima rasti konkrečioje bibliotekoje.

Supratę, apie ką laikysite duomenis ir kaip tie duomenys susiję, jau būsite labai priartėję prie fizinio duombazės įgyvendinimo.

Ryšiai

Galimos kelios sąryšių rūšys:

- **Būtinasis.** Kiekvienas klasės A pavyzdys turi būti susijęs su vienu ar keliais klasės B pavyzdžiais. Tačiau tai nereiškia, kad kiekvienas klasės B pavyzdys bus susijęs su vienu ar keliais klasės A pavyzdžiais. Klasės-sąryšiai yra būtinieji arba nebūtinieji tik viena kryptimi, todėl A ryšys su B gali būti nebūtinasis, o B ryšys su A - būtinasis.
- **Nebūtinasis.** Kiekvienas klasės A pavyzdys gali būti susijęs, o gali būti ir nesusijęs su klasės B pavyzdžiais.

Ryšiai

Vieno ryšys su vienu (1:1). Šiuo atveju kiekvienas klasės A pavyzdys susijęs su vienu klasės B pavyzdžiu ir atvirkščiai. Jei ryšys nebūtinasis, pavyzdžių gali būti vienas arba gali visai nebūti, o jei ryšys būtinasis, tuomet bus tik vienas susietosios klasės pavyzdys.

Vieno ryšys su daugeliu (1:D). Kiekvienam klasės A pavyzdžiui tenka keli klasės B pavyzdžiai, tačiau kiekvienam klasės B pavyzdžiui tenka tik vienas klasės A pavyzdys. Šis ryšys gali būti būtinasis arba nebūtinasis.

Daugelio ryšys su daugeliu (D:D). Kiekvienam klasės A pavyzdžiui tenka keli klasės B pavyzdžiai, ir atvirkščiai. Šie ryšiai gali būti būtinieji arba nebūtinieji.

Ryšiai

Šiuos ryšius galima pavaizduoti įvairiausiais būdais.

Paveiksle parodytos klasės student ir course.

Šiuo atveju, kiekvienas studentas turėjo užsiregistruoti bent į vieną kursą, tačiau bus ir tokių kursų, į kuriuos neužsiregistruos nei vienas studentas.

Todėl studento ir kurso ryšys yra būtinas, o kurso ir studento ryšys - nebūtinas.

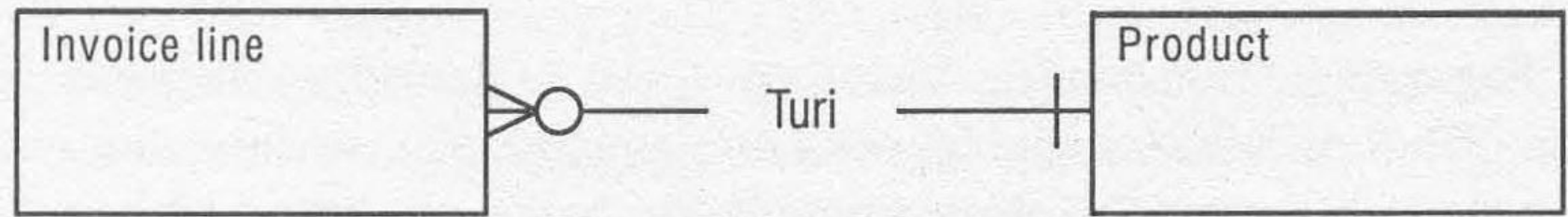


Ryšiai

Paveiksle parodytos klasės invoice line ir product.

Kiekvienoje sąskaitos eilutėje turi būti bent vienas gaminy (bet ne daugiau negu vienas), o gaminiai gali būti ir keliose sąskaitos eilutėse arba jų sąskaitoje gali visai nebūti.

Taigi sąskaitos eilutės ryšys su gaminiu yra būtinas, o gaminio ryšys su sąskaitos eilute - nebūtinas.



Ryšiai

Vyro ir žmonos klasės.

Vienas vyras gali turėti tik vieną žmoną. Taip pat ir viena žmona gali turėti tik vieną vyrą.

Abu šie ryšiai yra būtinieji.



Rekursinės klasės

Klasė gali turėti ryšį ir su savimi. Tuomet ji vadinama **rekursine klase**.

Paimkime klasę Žmogus. Jei norite surinkti duomenis apie žmones, kurie yra broliai, turėsite ryšį "yra kažkieno brolis". Tai ryšys D:D.

Pagalvokite kokios klasės gali turėti tokį ryšį jūsų kuriamoje parduotuvės Duomenų bazėje.

Klasių sąryšių diagramos kūrimas

Nubraižyta pradinė klasių-sąryšių diagrama paprastai parodoma užsakovams. Tokias diagramas lengvai supranta ne techninių specialybių žmonės.

Jos padeda aptikti ir prasiskverbusias problemas.

Modeliavimas naudingas ir tuo, kad modeliai suprantami lengviau už prirašytus puslapius, todėl didesnė tikimybė, kad užsakovai juos peržiūrės, o tai sumažina klaidų tikimybę vėlesniuose etapuose, kai jas kur kas sunkiau ištaisyti.

Klasių sąryšių diagramos kūrimas

Kai diagrama bus patvirtintos, daugelio ryšius su daugeliu reikės pakeisti dviem vieno ryšiu su daugeliu.

DBVS negali tiesiogiai įgyvendinti daugelio ryšių su daugeliu, todėl juos reikia perdaryti į dvejus mažesnius ryšius.

Sankirtos klasės

Norėdami įgyvendinti daugelio ryšių su daugeliu, turite sukurti **sankirtos** arba sudėtinę klasę.

Kadangi sankirtos klasės ne tokios praktiškos kaip paprastos klasės, kai kada jas sunku pavadinti.

Tuomet galite jas pavadinti pagal dvi susietas klases.

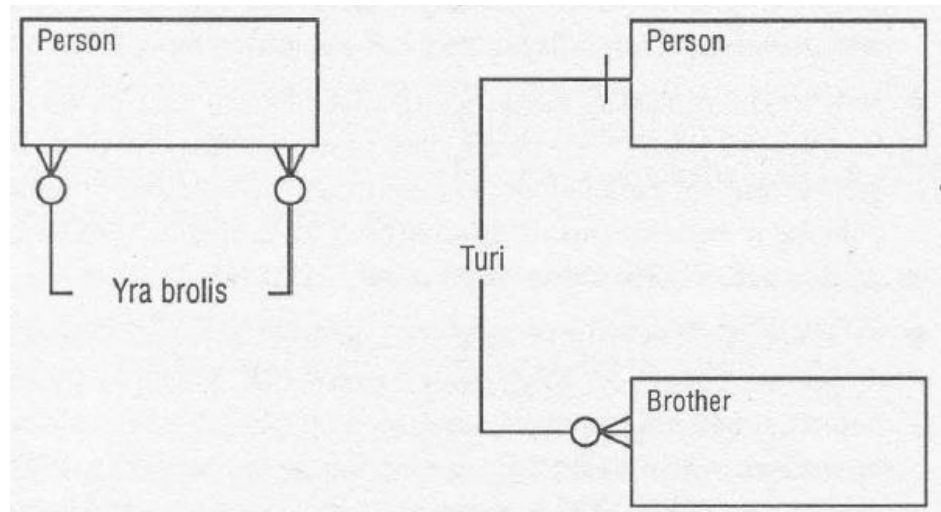
Pavyzdžiui, daugelio ryšį su daugeliu tarp student ir course galite susieti per klasę student-course

Sankirtos klasės

Sankirtos klasės tinka, net tada, jei klasė yra rekursinė.

- Asmens klasei, turinčiai D:D ryšį "yra kažkieno brolis", taip pat reikia sankirtos klasės. Šiuo atveju sankirtos klasei galite sugalvoti puikiausią vardą - Brolis.

Šioje klasėje bus du laukai, po vieną kiekvienam brolio sąryšio asmeniui, kitaip tariant, pirmojo brolio pirminis raktas ir kito brolio pirminis raktas.



Kelių lentelių jungimas

Viena iš svarbiausių SELECT sakinio galimybių yra dviejų ar daugiau lentelių jungimas.

Tarkime, mums reikia sužinoti pavardes vykdytojų, vykdančių projektą Nr. 1. Visa reikalinga informacija yra dviejose lentelėse: *Vykdytojai* ir *Vykdimas*, todėl užklausoje turi dalyvauti ne viena, o dvi lentelės. Šios lentelės turi bendrą dalį, kuri jas sieja – tai vykdytojo numeris. Pagal šį numerį lenteles galima logiškai jungti.

SQL kalba šį uždavinį galima užrašyti taip:

```
SELECT Pavardė FROM Vykdytojai, Vykdimas WHERE Vykdytojas =  
Nr AND Projektas = 1
```

Šioje užklausoje, sąlyga *Vykdytojas = Nr* yra nusakytas loginis ryšys tarp dviejų lentelių

Kelių lentelių jungimas

Jungiant lenteles, kiekviena vienos lentelės eilutė siejama su kiekviena kitos lentelės eilute. Bendru atveju, jei užklausoje dviems lentelėms nėra jokios sąlygos ir vienoje iš lentelių yra n eilučių, o kitoje - m eilučių, tai rezultatą sudaro $m \times n$ eilučių.

Kelių lentelių jungimas

Lentelė A

<i>A1</i>	<i>B1</i>
a ₁	b ₁
a ₂	b ₁
a ₂	b ₂

Lentelė B

<i>A2</i>	<i>B2</i>	<i>C2</i>
a ₁	b ₁	c ₁
a ₂	b ₂	c ₂

Tuomet užklauso

```
SELECT A1, B1, A2, B2, C2 FROM A JOIN B ON A1=A2
```

<i>A1</i>	<i>B1</i>	<i>A2</i>	<i>B2</i>	<i>C2</i>
a ₁	b ₁	a ₁	b ₁	c ₁
a ₂	b ₁	a ₁	b ₁	c ₁
a ₂	b ₂	a ₁	b ₁	c ₁
a ₁	b ₁	a ₂	b ₂	c ₂
a ₂	b ₁	a ₂	b ₂	c ₂
a ₂	b ₂	a ₂	b ₂	c ₂

Kelių lentelių jungimas

Paprastai užklausoje dviems lentelėms yra paieškos sąlyga, kuri logiškai susieja vienos lentelės vieną ar kelis stulpelius su kitos lentelės atitinkamais stulpeliais. Papildžius pastarąją užklausą paieškos sąlyga, kuri sujungtų abi lenteles, sulyginant dviejų vienos lentelės stulpelių ($A1, B1$) reikšmes su kitos lentelės dviejų stulpelių ($A2, B2$) reikšmėmis:

```
SELECT A1, B1, C2 FROM LentelėA, LentelėB WHERE A1= A2 AND  
B1 = B2
```

gausime tik dvi eilutės:

<i>A1</i>	<i>B1</i>	<i>C2</i>
a_1	b_1	c_1
a_2	b_2	c_2

Kelių lentelių jungimas

Užrašysime užklausą apie tai, kokie vykdytojai kokius projektus vykdo ir kiek kiekvienam projektui skiria valandų. Jei rezultate norime matyti vykdytojo pavardę ir projekto pavadinimą, o ne jų numerius, tai mums reikia formuluoti užklausą trimis lentelėmis:

```
SELECT Pavardė, Pavadinimas, Valandos  
FROM Vykdytojai, Projektai, Vykdymas  
WHERE Projektas = Projektai.Nr AND Vykdytojas = Vykdytojai.Nr.
```

Šioje užklausoje dvi lentelės (Vykdytojai ir Projektai) turi po vieną stulpelį tuo pačiu pavadinimu Nr, todėl šio stulpelio pavadinimą būtina patikslinti lentelės pavadinimu.

Kelių lentelių jungimas

Lentelę galima sujungti ir su ja pačia. Sudarykime poras vykdytojų, turinčių tą pačią kvalifikaciją:

```
SELECT A.Pavardė, B.Pavardė FROM Vykdytojai A, Vykdytojai B
WHERE A.Kvalifikacija = B. Kvalifikacija AND A.Nr < B.Nr
```

Jungiant lentelę su ja pačia, galime laikyti, kad turime du tos pačios lentelės egzempliorius (kopijas). Tam, kad būtų galima kreiptis ir į abudu vienos lentelės egzempliorius, būtina kiekvienam egzemplioriui suteikti pavadinimą (*A* ir *B*). Kadangi abiejų lentelių (tiksliau dviejų vienos lentelės egzempliorių) stulpelių vardai sutampa, tai užklausoje negalime apsieiti be stulpelių vardų patikslinimo lentelės egzemplioriaus pavadinimu.

Predikatas *A.Kvalifikacija = B. Kvalifikacija* atlieka dviejų lentelių loginio susiejimo funkciją.

Sąlyga *A.Nr < B.Nr* pavartota tik tam, kad išvengti vykdytojo poros su juo pačiu ir porų pasikartojimo.

Vidiniai ryšiai

Šios dvi užklausos yra tapačios:

```
SELECT first name, surname, value FROM customer,sales WHERE  
id=customer;
```

```
SELECT first_name, surname, value FROM customer INNER JOIN sales  
ON i d=customer;
```

Kairieji ryšiai (kairieji išoriniai ryšiai)

Išoriniai ryšiai „OUTER JOIN“ pateikia rezultatus, nepriklausomai nuo to, ar kitoje lentelėje yra sąlygą atitinkančių įrašų, ar ne. Net jei lentelės laukas yra NULL tipo ir neturi atitiktens kitoje lentelėje, rezultatai bus pateikti.

Kairysis išorinis ryšys LEFT OUTER JOIN yra vienas iš tų, kurie pateikia visas ryšio sąlygą atitinkančias kairiosios lentelės eilutes, nepriklausomai nuo to, ar dešinėje lentelėje yra tą sąlygą atitinkančių eilučių.

Kairieji ryšiai (kairieji išoriniai ryšiai)

Ryšio LEFT JOIN (sutrumpintas LEFT OUTER JOIN vardas) sintaksė tokia:

```
SELECT field1, field2 FROM table1  
LEFT JOIN table2 ON field1=field2
```

Ryšio LEFT JOIN sakinyje yra svarbi lentelių nurodymo tvarka. Lentelė, kurios visos eilutės turės būti pateiktos, turi būti kairioji (nurodoma prieš komandos žodžius LEFT JOIN).

Kairieji ryšiai (kairieji išoriniai ryšiai)

Galima prie ryšio prijungti ir trečiąją lentelę. Tuomet užklausa atrodys taip:

```
SELECT field1, field2, field3 FROM table1  
    LEFT JOIN table2 ON field1=field2  
    LEFT JOIN table3 ON field1=field3
```

Dešinieji ryšiai (dešinieji išoriniai ryšiai)

Dešinieji ryšiai yra tokie patys kaip ir kairieji ryšiai, išskyrus tai, kad ryšys išdėstytas atvirkštine tvarka.

```
SELECT field1, field2 FROM table1  
    RIGHT JOIN table2 ON field1=field2
```

Jeigu jums sunku nuspręsti, kokią lentelę naudoti ir kokioje ryšio pusėje - dešinėje ar kairėje, atsiminkite, kad dešinieji ryšiai pateikia visus dešinėsios lentelės įrašus, įskaitant NULL reikšmes, o kairieji ryšiai pateikia visus kairiosios lentelės įrašus, įskaitant NULL reikšmes.

Abipusiai išoriniai ryšiai

Abipusiai išoriniai ryšiai, tai tokie ryšiai, kuriuose kartu lygiagrečiai su pirmosios lentelės įrašais, net ir tais kurie neturi atitikmenų antrojoje lentelėje yra pateikiami antrosios lentelės įrašai, net ir tie, kurie neturi atitikmenų pirmojoje lentelėje.

SQL sintaksė tokios užklausos būtų:

```
SELECT field1, field2 FROM table1 FULL OUTER JOIN table2
```

MySQL kol kas nepalaiko išorinių abipusių ryšių, tačiau tai galima kompensuoti sujungiant kairinį ir dešinį išorinius ryšius.

Paprasti ryšiai (NATURAL JOIN)

Jei dvi lentelės turi laukus su vienodais vardais, galite sudaryti paprastą ryši NATURAL JOIN, kuris ieškos abeiose lentelėse įrašų pagal vienodus laukų vardus.

Šios užklausos gražintų tą patį rezultatą:

```
SELECT first_name, surname, value FROM customer  
LEFT JOIN sales ON customer.id=sales.id;
```

```
SELECT first_name, surname, value FROM customer  
NATURAL LEFT JOIN sales ;
```