

ŠIAULIŲ UNIVERSITETAS

MATEMATIKOS IR INFORMATIKOS FAKULTETAS

INFORMATIKOS KATEDRA

Tomas Pocius

Informatikos magistro, gamybininko specialybės II kurso (neakivaizdinio skyriaus) studentas

## **Lizingo įmonės verslo kokybės valdymo sistema**

MAGISTRO DARBAS

Darbo vadovas:  
prof. L. Sakalauskas

Recenzentas:  
lekt. V. Giedrimas

Šiauliai, 2005/2006 m.m.

# Turinys

<b>1</b>	<b>ĮVADAS</b> .....	<b>3</b>
<b>2</b>	<b>TEORINĖ DALIS</b> .....	<b>5</b>
2.1	DARBO SRITIES ANALIZĖ .....	5
2.1.1	Verslo kokybės valdymo svarba .....	5
2.1.2	Lizingo samprata, pagrindinės charakteristikos .....	6
2.1.3	Periodiniai mokėjimai .....	8
2.1.4	Lizingo veiklos statistinis modelis .....	10
2.1.5	Klientų vertinimo sistema bei lizingo produktų išdavimas .....	11
2.1.5.1	Klientų vertinimo sistema .....	11
2.1.5.2	Klientų grupavimas (segmentavimas) .....	12
2.1.5.3	Pirkimo išsimokėtinai kortelė .....	13
2.1.5.4	Išdavimo tvarka .....	13
2.1.5.5	Mokesčių skaičiavimas .....	14
2.1.5.6	Klientų informavimas .....	14
2.1.5.7	Verslo valdymo įvykio (sandorio) schema .....	14
2.1.6	Duomenų bazės samprata, duomenų gavyba .....	14
<b>3</b>	<b>PROJEKTINĖ DALIS</b> .....	<b>15</b>
3.1	UŽDAVINYS IR JO ANALIZĖ .....	15
3.2	PROJEKTAS .....	15
3.3	DUOMENŲ BAZĖS PROJEKTAVIMO ETAPAI .....	16
3.4	DUOMENŲ APDOROJIMO SCHEMA .....	16
3.5	DARBO PRIEMONIŲ PARINKIMAS .....	17
3.6	DUOMENŲ APDOROJIMAS .....	18
3.7	PROGRAMAVIMO TECHNOLOGIJOS .....	20
<b>4</b>	<b>DARBO EIGOS APRAŠYMAS</b> .....	<b>23</b>
4.1	DUOMENŲ BAZĖS REALIZACIJA .....	23
4.2	PROJEKTO MODULIAI .....	23
4.2.1	Pirkimo išsimokėtinai kortelės (verslo valdymo sistemos) moduliai .....	24
4.2.2	Įmonės verslo kokybės valdymo moduliai .....	26
4.3	AUTOMATINIO DUOMENŲ APDOROJIMO SERVISAS .....	28
4.4	DARBAS SU KELIOMIS DB IR DBVS .....	29
4.5	DARBE IŠKILUSIOS PROBLEMOS BEI JŲ SPRENDIMAS .....	29
4.6	GALUTINIS PROJEKTO STOVIS .....	30
<b>5</b>	<b>IŠVADOS</b> .....	<b>31</b>
<b>6</b>	<b>LITERATŪRA</b> .....	<b>32</b>
<b>7</b>	<b>ANOTACIJA</b> .....	<b>33</b>
<b>8</b>	<b>SUMMARY</b> .....	<b>34</b>

# 1 Įvadas

Įvairiose įmonėse, įstaigose plačiai taikomas duomenų apdorojimas kompiuteriu – tai lyg tiltas tarp turimų duomenų ir priimamų sprendimų. Visų pelno siekiančių organizacijų, įmonių svarbiausias tikslas – uždirbti kuo daugiau pelno. Rinkos sąlygomis įmonių finansų valdymas nulemia veiklos sėkmę. Ilgainiui mūsų įmonių finansų tvarkytojai, nepasikliaudami savo patyrimu ir intuicija, darydami finansinius sprendimus analizuoja uždirbamą pelną, ir kitus veiklą apibūdinančius rodiklius, marketingo skyriaus darbuotojai ieško naujų klientų pritraukimo galimybių, be to labai svarbu išlaikyti jau esamus klientus. Kuo daugiau gerų klientų – tuo daugiau pelno gauna organizacija.

Kad pritraukti naujus klientus bei neprarasti senų labai svarbu išlaikyti pastovų ryšį tarp kliento ir įmonės, svarbu sekti visus veiksmus, susijusius su klientu, fiksuoti klientų poreikius ir atitinkamai reaguoti į šia informaciją. Čia į pagalbą pasitelkiamos šiuolaikinės technologijos. Kompiuterizuotas duomenų apdorojimas – labai efektyvi priemonė padedanti įmonės darbuotojams operatyviai reaguoti į iškilusius kliento poreikius, pateikti jam pasiūlymus tiems poreikiams patenkinti, atsirinkti, kada efektyviau galima išnaudoti marketingines priemones. Be to, tokių duomenų apskaitos kompiuterizavimas leidžia automatizuotai paskirstyti darbus įmonės darbuotojams bei įmonės vadovui sekti ar visi darbai atliekami sklandžiai ir laiku.

Darbo tikslai:

- Sukurti sistemą lizingo įmonės verslo ir verslo kokybės valdymui:
  - Sukurti pirkimo išsimokėtinai kortelės apskaitos modulius (verslo valdymo):
    - § kortelė – modulis, skirtas lizingo „Pirkimo išsimokėtinai“ kortelės projektų apskaitai, kuri sudaro paraiškų bei projektų valdymo, banko piniginių operacijų, mokesčių skaičiavimo bei apskaitymo bei rizikos valdymo modulius. Pirkimo išsimokėtinai kortelių periodiniai mokėjimai turi būti paskaičiuojami panaudojant *anuiteto* principą;
  - Sukurti verslo kokybės valdymui skirtus modulius:
    - § pagrindinės informacijos apie subjektus (įmones, asmenis) valdymas;
    - § informacijos apie subjektų poreikius registravimas;

- § informacijos apie atliktus veiksmus su subjektu, su juo susijusių dokumentų registravimas;
  - § Galimybė įmonės vadovui skirti darbus pavaldiniams bei stebėti operatyvų šių darbų atlikimą;
  - § Klientų informavimo automatizavimas;
  - § Marketingo priemonių bei įvykių apskaita.
- Sukurti automatinio duomenų apdorojimo servisą, skirtą duomenų apdorojimo automatizavimui;
  - Sukurti papildomus COM servigus, skirtus automatinam duomenų apdorojimui.

Darbą sudaro:

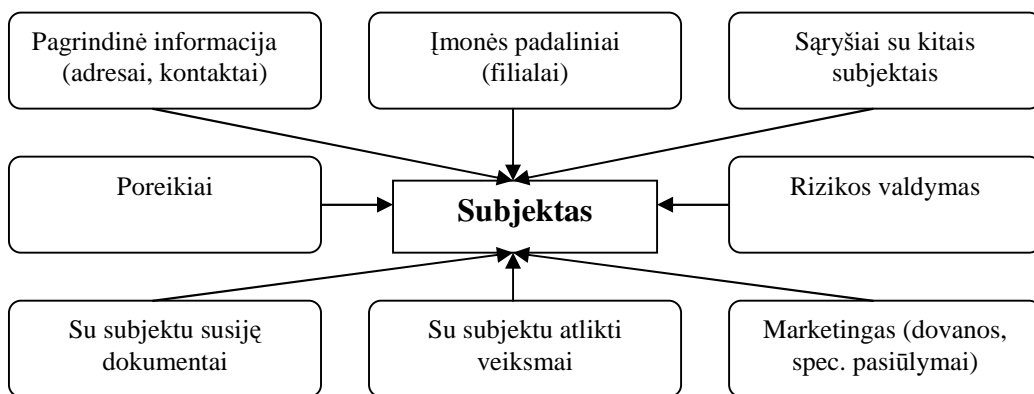
- Teorinė dalis;
- Projektinė dalis;
- Darbo eigos aprašymas;
- Darbo rezultatų analizė;
- Išvados;
- Priedai:
  - Vartotojo vadovas (1 priedas);
  - Programuotojo vadovas (2 priedas);
  - Duomenų bazių samprata (3 priedas);
  - CD laikmenos turinys ir kiti nurodymai (4 priedas).

## 2 Teorinė dalis

### 2.1 Darbo srities analizė

#### 2.1.1 Verslo kokybės valdymo svarba

Lizingo įmonės (kaip ir daugelio kitų įmonių) veikla ir pelnas priklauso nuo to, kiek įmonė pritrauks klientų, t.y. kiek sudarys naujų sandorių. Todėl įmonei svarbu ne tik pritraukti naujus klientus bet ir išlaikyti jau buvusius klientus. Ryšys su buvusiais klientais dažnai nutrūksta kai vyksta darbuotojų migracija. Atėjęs naujas darbuotojas nežino kokie pasiūlymai jau buvo pateikti klientui, kokie susitarimai pasiekti ir kokie veiksmai atlikti. Todėl šiems procesams valdyti reikalinga sistema, kuri leistų registruoti visą su klientu susijusią informaciją. Tada naujas darbuotojas bet kada gali pasižiūrėti bendravimo su klientu istoriją ir ta bendravimą toliau pratęsti. Taigi sistemos pagrindas yra subjektas (klientas, partneris, pardavėjas ir kt.) ir prie jo saugoma visa bendravimo istorija, sąryšiai.



Tai pagrindiniai verslo kokybės valdymo objektai. Jie glaudžiai siejasi su pačiu verslo valdymu, todėl ir pati verslo valdymo sistema glaudžiai siejasi su verslo kokybės valdymo sistema. Šio darbo rezultatas – sistema, atliekanti ir verslo valdymo ir verslo kokybės valdymo funkcijas. Sistemos verslo valdymo funkcijos orientuotos į lizingo įstaigos veiklą.

## 2.1.2 Lizingo samprata, pagrindinės charakteristikos

Lizingas – tai netradicinė finansavimo sistema, jungianti ilgalaikės nuomos ir kreditavimo elementus. Jo esmė yra ilgalaikė nuoma, kurią įmonei - lizingo gavėjai - suteikia bankas, lizingo objektą gaminanti įmonė ar lizingo kompanija, vadinama lizingo davėju. Sudaręs su lizingo gavėju lizingo sutartį, lizingo davėjas perka įrengimą, mašinas, kitą nekilnojamąjį turtą ir išnuomoja lizingo gavėjui už tam tikrą mokėjimų sumą, kurią šis moka per visą sutarties laiką.

Ekonominiu požiūriu lizingas – tai ilgalaikis prekinės formos kreditas, kai išnuomos priemonės savininkas už suteiktą daiktinę paskolą gauna palūkanas .

Lizingas paplitęs išsivysčiusiose Vakarų šalyse ir yra vienas iš labiausiai paplitusių mechanizmų, naudojamų šalies finansų rinkoje, nes suteikia naujų galimybių ir privalumų tiek lizingo davėjui, tiek lizingo gavėjui.

**Lizingo objektas ir subjektas.** Lizingo objektu gali būti įvairios paskirties kilnojamasis ir nekilnojamasis turtas: pastatai, žemės sklypas, pramoniniai ir kiti įrengimai, kompiuterinė ir informacinė technika, orgtechnika, parduotuvių įranga ir apipavidalinimas, žemės ūkio mašinos, automobiliai, sunkvežimiai, lėktuvai ir t.t.

Lizingo objektas išnuomojamas skirtingam laikotarpiui. Nuoma gali būti:

- trumpalaikė (iki vienerių metų);
- vidutinė (nuo vienerių iki trejų metų);
- ilgalaikė (daugiau nei treji metai).

Per visą nuomos laikotarpį nuosavybės teisę į nuomos objektą išlaiko lizingo davėjas.

Lizingo subjektas. Klasikiniam lizingui būdinga trišalė sutartis, t.y. kai sandoryje dalyvauja trys šalys: lizingo davėjas - nuomotojas, lizingo gavėjas - nuomininkas ir objekto gamintojas - tiekėjas.

Svarbiausias vaidmuo lizingo operacijoje tenka nuomotojui lizingo davėjui. Sudaręs su lizingo gavėju lizingo sutartį, lizingo davėjas nuperka įrangą, mašinas, pastatus ar kitą materialųjį turtą ir, įrašęs juos į savo balansą, išnuomoja lizingo gavėjui už tam tikrą mokestį. Nuomotojas lizingo objektą perka savo lėšomis arba šiam tikslui gauna banko paskolą. Taigi lizingo davėjas savotiškai finansuoja lizingo objekto vartotoją.

Lizingo davėju gali būti:

- komerciniai bankai ir jų filialai, kurių nuostatuose numatyta lizingo veikla;
- finansinės lizingo kompanijos, sukurtos specialiai lizingo operacijoms vykdyti. Jos gali veikti kaip banko padalinys;
- firmos, kurioms lizingo veikla nėra pagrindinis verslas, tačiau įstatuose ši veikla numatyta .

Antrasis lizingo sandorio dalyvis yra lizingo objekto nuomininkas lizingo gavėjas. Nuomininku gali būti bet kuris ūkinis subjektas, nepriklausomai nuo jo nuosavybės tipo: akcinė bendrovė ar uždara akcinė bendrovė, individuali įmonė ar valstybinė įmonė o taip pat ir fiziniai asmenys.

Trečiasis lizingo operacijos dalyvis yra lizingo objekto pardavėjas tiekėjas, turintis juridinio asmens statusą: gamybinė įmonė, tiekimo-realizavimo organizacija, prekybos firma ir t.t.

Lizingo privalumai nuomininkui:

- Lizingas suteikia iki 100% lėšų turto įsigijimui. Dažnai nereikalaujama išankstinių mokėjimų;
- Lizingas “neužšaldo” kapitalo, galima daugiau kapitalo skirti apyvartinėms lėšoms, kurios būna labai reikalingos kiekvienos įmonės veiklai;
- Nuomos mokesčiai, užfiksuoti lizingo kontrakte, padeda numatyti ir apskaičiuoti išlaidas, prognozuoti pinigų srautus;
- Lizingas yra paprasta, nesudėtinga operacija. Lizingas minimizuoja administracinius kaštus ir supaprastina mokesčių ir apskaitos procedūras, dokumentaciją;
- Leidžia atsižvelgti į įmonės veiklos ypatumus (sezoniškumas, cikliškumas ir pan.).

Lizingo privalumai nuomotojui:

- Lizingas finansinėms institucijoms yra papildomas finansinis produktas, iš kurio galima gauti pelną šalia visų kitų finansinių paslaugų;
- Lizingas sumažina riziką. Daugelyje šalių yra lengviau ir pigiau nuomotojui atgauti išnuomotą įrangą, jeigu nuomininkas nevykdo įsipareigojimų, negu užstato ar paskolos atveju. Vis dėlto kartais neįmanoma gretai atgauti nuomojamą įrangą arba ji gali turėti žymiai sumažėjusią vertę, ypač jeigu buvo blogai prižiūrima. Be to, nuomininkas kartais gali savavališkai perduoti išnuomotą įrangą kam nors kitam;
- Lizingo dokumentacija nesudėtinga tiek nuomininkui, tiek ir nuomotojui. Ji yra paprastesnė nei paskolos atveju;

- Lizingo kontraktas gali būti greitai užbaigiamas. Nuomininkas dažniausiai greitai priima sprendimus dėl lizingo;

Lizingo privalumai lizingo objekto tiekėjui (gamintojui):

- Lizingo kompanija tampa pastoviu ir patikimu gamintojo partneriu, apmokančiu visą užsakymo vertę;
- Gamintojas pastoviai bendradarbiaujantis su lizingo kompanija, gali sumažinti išlaidas reklamai, vartotojų paieškai, rinkos tyrimams, nes šias paslaugas teikia lizingo kompanija.

Lizingo trūkumai:

- Gavėjui duoda tik laikiną teisę naudotis įrengimais;
- Gali kainuoti brangiau negu banko kreditas tiems patiems įrengimams įsigyti.

### 2.1.3 Periodiniai mokėjimai

Modulis kortelė skirtas lizingo pirkimo išsimokėtinai kortelės projektų apskaitai bei valdymui.

Lizingo veikimo schema pagrįsta periodinių mokėjimų (annuity) principu. Tai yra, lizingo bendrovė klientui skolina kažkokią pinigų sumą, o klientas įsipareigoja šia sumą gražinti per nustatytą laikotarpį lygiomis dalimis bei mokėti nustatytus mokesčius – palūkanas.

**Periodiniai mokėjimai** (annuity) – tai seka mokėjimų, paprastai vienodų, atliekamų lygiais laiko intervalais. Laikas tarp dviejų nuoseklių mokėjimų vadinamas **mokėjimo intervalu** (payment interval). Laikas nuo pirmojo mokėjimo pradžios iki paskutiniojo mokėjimo pabaigos vadinamas **mokėjimų terminu** (term). Lizingo veikloje taikomi **determinuotieji periodiniai mokėjimai**, t.y. kai mokėjimų terminas yra fiksuotas.

Tarkime, kad bankas išdavė L dydžio paskolą, ją klientas turi gražinti per n periodų. Kiekvieną periodą  $t$  ( $1 \leq t \leq n$ ) klientas turi įmokėti po  $R_t$  dydžio įmoką. Kiekvienas mokėjimas  $R_t$  susideda iš paskolos dalies dengimo bei palūkanų už periodą  $t$ .

Tarkime palūkanos kiekvieną periodą yra pastovios ir lygios  $i$ . Paskolinta pinigų suma  $L$  lygi gražinamo pinigų srauto dabartinei vertei:

$$L = R_1(1+i)^{-1} + R_2(1+i)^{-1}(1+i)^{-1} + R_3(1+i)^{-1}(1+i)^{-1}(1+i)^{-1} + \dots + R_n(1+i)^{-1}(1+i)^{-1} \cdot \dots \cdot (1+i)^{-1}$$

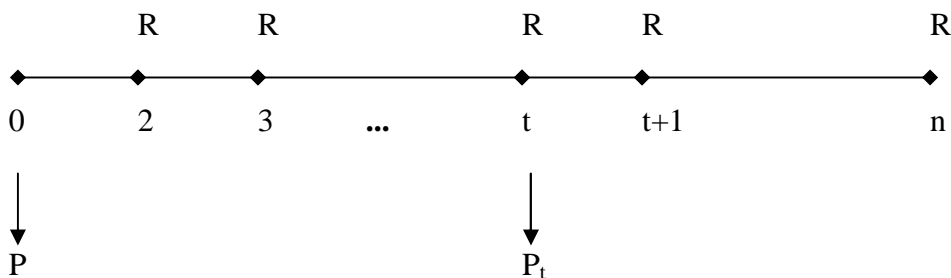
Galima sudaryti **paskolos gražinimo struktūrą** (loan schedule), kuri parodo, kurią mokėjimo



dalį sudaro palūkanos ir kurią paskolos gražinimas. Tarkime, kad nepadengta paskolos vertė kiekvieną periodą lygi  $P_t$ , o pradinė paskolos vertė lygi suteiktam paskolos dydžiui  $P_0=L$ .

Norėdami gauti likutinę paskolos vertę  $P$  periodu  $t$  turime iš buvusios vertės dydžio atimti periodo  $t$  įmoką be palūkanų  $R_t$ .  $P_t=P_{t-1}-(R_t-iP_{t-1})$  arba tai yra lygu  $P_t=(1+i)P_{t-1}-R_t$ . Periodinių

mokėjimų dydis paskaičiuojamas  $R = -\frac{P}{a(n;i)}$ , čia  $a(n;i) = \frac{1-(1+i)^{-n}}{i}$ , kuris vadinamas  $n$  mokėjimų diskontavimo daugikliu.



Galime sudaryti mokėjimų grafiką (skolos gražinimo struktūrą):

Mokėjimas	Mokėjimo palūkanų dalis	Gražintos paskolos dalis	Likusios paskolos dydis po mokėjimo
1.	$Ria(n;i) = R(1-v^n)$	$Rv^n$	$R(a(n;i) - v^n) = Ra(n-1;i)$
2.	$Ria(n-1;i) = R(1-v^{n-1})$	$Rv^{n-1}$	$R(a(n-1;i) - v^{n-1}) = Ra(n-2;i)$
...	...	...	...
t	$Ria(n-t+1;i) = R(1-v^{n-t+1})$	$Rv^{n-t+1}$	$R(a(n-t+1;i) - v^{n-t+1}) = Ra(n-t;i)$
...	...	...	...
n-1	$Ria(2;i) = R(1-v^2)$	$Rv^2$	$R(a(2;i) - v^2) = Ra(2-1;i)$
n	$Ria(1;i) = R(1-v^1)$	$Rv$	$R(a(1;i) - v) - v = 0$

Pagal šias formules klientui sudaromas **mėnesinių mokėjimų grafikas**. Pagal šį grafiką klientas kiekvieną mėnesį iki nustatytos mokėjimo dienos turi įnešti **mėnesinę įmoką**. **Mėnesinis mokėjimas R** – tai suma kurią klientas turi įnešti pagal grafiką, o **mėnesinė įmoka M** – tai suma, kurią klientas įneša. Mėnesinis mokėjimas gali nesutapti su mėnesine įmoka, jei klientas nusprendžia įnešti keletą mokėjimų iš karto (avansu) arba jei klientas įneša mažiau nei mėnesinis mokėjimas. Pastaruoju atveju susidaro kliento periodo skola  $D$  lygi mėnesinio mokėjimo ir mėnesinės įmokos skirtumui  $D=R-M$ . Klientas gali neįnešti įmokų keletą periodų tada jo periodo

$$\text{skola lygi kiekvieno praėjusio periodo mokėjimo ir įmokos skirtumų sumai } D = \sum_{i=1}^k (R_i - M_i).$$

Esant kliento nemokumo atvejui skaičiuojami papildomi mokesčiai (delspinigiai). Tarkime delspinigių tarifas  $f$  už kiekvieną pradelstą dieną, tai delspinigių suma  $F$  dieną  $d$  bus lygi  $F=D*f$ .

Lizingo bendrovės pajamas sudaro palūkanos bei kiti mokesčiai (delspinigiai), o išlaidas - suteiktos paskolos.

#### 2.1.4 Lizingo veiklos statistinis modelis

Lizingo bendrovė. Laiko momentais  $tt_j$  sudaromos sutartys. Sutarties periodas gali būti  $Tr_1$  ir  $Tr_2$  atitinkamai su tikimybėmis  $P_1$  ir  $P_2$ . Projektų srauto intensyvumas  $\lambda$ . Nagrinėjamas projektų skaičius  $N$ . Projektų vertė  $x$  gali kisti intervale  $[x_{min}; x_{max}]$ . Turto vertės dengimo periodas  $\Delta$ , palūkanos nuo kiekvienos turto vertės dengimo dalies  $\rho$ .

Išlaidos – sudarius projektą, lizingo bendrovė sumoka visą turto vertę.

Pajamos – Kiekvieną sutarties veikimo periodą klientas gražina po dalį turto vertės (išlaidų) bei nustatytą procentą palūkanų.

Tarkime, paraiškų pasirodymo srautas pasiskirstęs pagal Puasono dėsnį. Kuriant modelį paraiškų sekos imitavimui galima sugeneruoti atsitiktinę skaičių seką  $\tau_i := -\frac{\log(\text{rnd}(1))}{\lambda}$ .

Iš šios sekos gauname paraiškų pasirodymo momentus  $tt_i := \sum_{j=0}^i \tau_j$ .

Projektų trukmės turi tikimybes ir gali būti parenkamos atsitiktiniu būdu.

$$T_i := \begin{cases} Tr_1 & \text{if } \text{rnd}(1) > P_1 \\ Tr_2 & \text{otherwise} \end{cases}$$

Projektų sumos taip pat parenkamos atsitiktinių skaičių pagalba.  $x_i := x_{min} + \text{rnd}(x_{max} - x_{min})$

Iš šių duomenų galima apskaičiuoti išlaidas bei pajamas laiko momentu  $t$

$$f(t, ttt) := \begin{cases} 0 & \text{if } t < ttt \\ 1 & \text{otherwise} \end{cases}$$

$$\text{islaidos}(t) := \sum_{k=0}^{N-1} x_k \cdot f(t, tt_k)$$

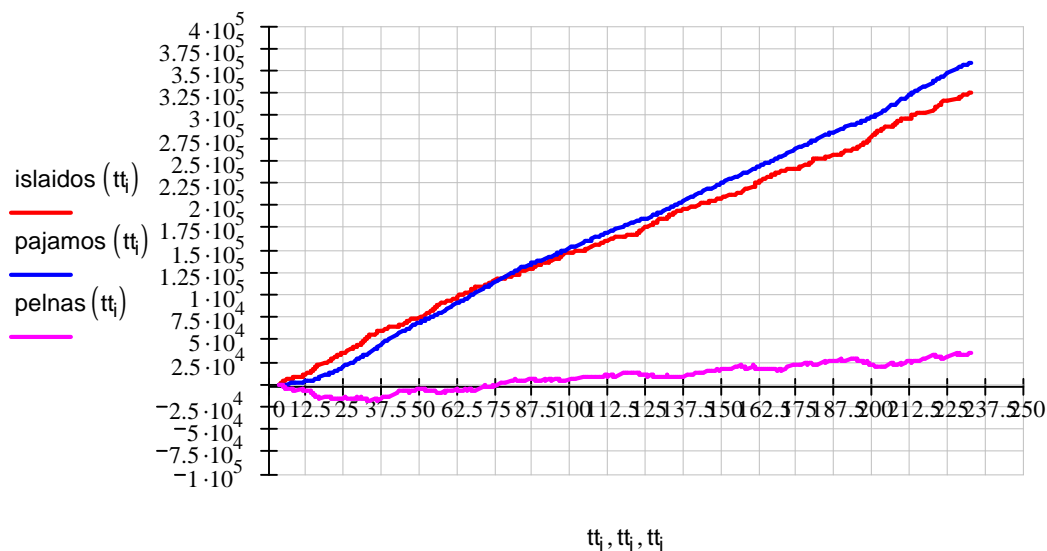
$$\rho(t, ttt, T) := \begin{cases} a & \left\{ \begin{array}{l} 0 \text{ if } t < ttt + \Delta \\ \left[ \frac{(t - ttt)}{T \cdot \Delta} \right] \text{ if } t \geq ttt \wedge t \leq ttt + T \cdot \Delta \\ 1 \text{ if } t > ttt + T \cdot \Delta \end{array} \right. \\ a \end{cases}$$

$$\text{pajamos}(t) := \sum_{k=0}^{N-1} x_k \cdot \rho(t, tt_k, T_k) \cdot (1 + \rho)$$

Pelnas laiko momentu  $t$   $\text{pelnas}(t) := \text{pajamos}(t) - \text{islaidos}(t)$

Remiantis šiuo modeliu galima modeliuoti lizingo veiklos situacijas, pelno priklausomybes nuo projektų trukmės bei lizinguojamos sumos.

Pvz.:



## 2.1.5 Klientų vertinimo sistema bei lizingo produktų išdavimas

### 2.1.5.1 Klientų vertinimo sistema

Vykdamas lizingo veiklą svarbu tinkamai įvertinti kliento mokumą, nes tai tiesiogiai susiję su lizingo bendrovės rizika. Pasitaikius kliento nemokumo atvejui lizingo bendrovė patiria nuostolius, todėl lizingo bendrovė turi nustatyti kliento vertinimo kriterijus:

Kliento mėnesio pajamos – tai vienas svarbiausių kriterijų. Kuo didesnės kliento mėnesio pajamos, tuo didesnę mėnesinę įmoką jis gali mokėti, tuo didesnę sumą jam galima lizinguoti

(suteikti kredito limitą). Lizingo bendrovė turi nustatyti minimalų darbo užmokestį – sumą  $x$ , kurią klientas turi uždirbti per mėnesį atskaičius mokesčius. Klientui galima suteikti tokį kredito limitą, kad jo mėnesio pajamų atskaičius kitus kliento įsipareigojimus santykis su mėnesiniu mokėjimu atitiktų  $x:y$ . Pvz. nustatytas santykis yra 2:1 o kliento mėnesio įmoka yra 250, tai reiškia, kad klientas turi uždirbti ne mažiau kaip  $250 \cdot 2 = 500$  Lt per mėnesį.

Kliento kredito istorija – kriterijus, taikomas jau buvusiems klientams. Jei klientas jau turėjo sutartį su lizingo bendrove ir tinkamai įvykdė savo įsipareigojimus, jam gali būti pritaikoma lojalumo programa sumažinant mokesčius (palūkanas, administracinį mokestį). Jei klientas netinkamai vykde įsipareigojimus, vėlavo mokėti įmokas, kitą sutartį pasirašyti su juo lizingo bendrovė gali atsisakyti.

Amžiaus reikalavimai – lizingo sutartį gali pasirašyti klientai sulaukę tam tikro nustatyto amžiaus (sulaukę pilnametystės). Kuo klientui daugiau metų, tuo didesnė rizika, kad jis nesugebės įvykdyti įsipareigojimų ir atvirksčiai – jauni klientai (18-21m) dažniausiai dar neturi savo gyvenamo ploto bei turto, todėl lizingo bendrovės rizika taip pat didėja.

Kliento darbo stažas – klientas turi būti bent vienoje darbovietėje išdirbęs ne mažiau kaip nustatytą periodą –  $x$  mėnesių. Dažniausiai tai būna periodas didesnis nei galimas bandomasis periodas (pvz. daugiau nei 3 mėn.).

Invalidumo pažymos galiojimo laikas – jei limitas suteikiamas klientui pagal jo pajamas iš invalidumo pensijos, yra reikalaujama, kad invalidumo pažymėjimo galiojimo data būtų didesnė už sutarties pabaigos datą, t.y. klientas gaus tas pajamas visą sutarties galiojimo laikotarpį.

Jei klientas neatitinka bent vieno iš šių kriterijų, tad paskolos suteikimo galimybę svarsto įgaliotų įmonės asmenų grupė.

#### 2.1.5.2 Klientų grupavimas (segmentavimas)

Klientai pagal nustatytus kriterijus skirstomi į tam tikras grupes. Skirstymo kriterijų gali būti keletas:

1. Patikimų klientų grupavimas – pagal įvykdytus įsipareigojimus, darbo užmokestį bei užimamas pareigas:
  - a. VIP klientai – klientai turintys dideles pajamas bei geras pareigas ir planuojantys sudaryti didesnės apimties projektus;
  - b. Standartiniai klientai – dauguma klientų, turinčių vidutines pajamas, pareigas bei planuojantys sudaryti ne didelės apimties projektus.

- c. Įmonės darbuotojai – lizingo bendrovės darbuotojai, jiems gali būti taikomos kitokios vertinimo sistemos bei mokesčiai.
2. Nepatikimų klientų grupavimas – pagal vėlavimą mokėti įmokas, skolą, priskaičiuotus papildomus mokesčius už vėlavimą:
    - a. Vėluojantys nuo 0 iki 30 dienų;
    - b. Vėluojantys nuo 31 iki 60 dienų;
    - c. Vėluojantys nuo 61 iki 90 dienų;
    - d. Vėluojantys nuo 91 iki 120 dienų;
    - e. Vėluojantys daugiau nei 120 dienų;

#### 2.1.5.3 Pirkimo išsimokėtinai kortelė

Kaip alternatyva lizingo produktui lizingo bendrovės pradėjo siūlyti įsigyti pirkimo išsimokėtinai korteles. Tokios kortelės privalumai klientui – klientas gauna grynuosius pinigus ir po to gali įsigyti pageidaujamą prekę ar paslaugą. Privalumai lizingui – paprastesnė administravimo schema, nėra atsiskaitymų su pardavėjais administravimo. Trūkumai – didesnė lizingo bendrovės rizika, nes nėra apibrėžtas įsigytas turtas, kurį būtų galima susigrąžinti kliento nemokumo atveju.

Pirkimo išsimokėtinai kortelė skiriasi nuo kreditinės kortelės nes čia klientas išnaudotą limitą turi grąžinti dalimis kiekvieną mėnesį. Jam kaip ir lizingo sutarties atveju sudaromas mokėjimų grafikas ir paskaičiuojamas mėnesinis mokėjimas. Grafikas sudaromas panaudojant *anuiteto* principą su prielaida, kad klientas išnaudos visą jam suteiktą limitą ir kiekvieną mėnesį įmokės nustatytą įmoką.

#### 2.1.5.4 Išdavimo tvarka

Pirkimo išsimokėtinai kortelė išduodama klientui pateikus paraišką. Pagal jo mėnesinį atlyginimą pateikiamas komercinis pasiūlymas: klientas gali pasirinkti kokio limitu jam reikia bei kokiam laikotarpiui jis bus suteikiamas. Jei kliento pasirinktos sąlygos atitinka įvertinimo kriterijus, tada suforminama sutartis, jei klientas pageidauja sąlygų, kurios neatitinka įvertinimo kriterijų, tada jo pageidavimą svarsto paskolų komitetas (lizingo bendrovės narių komitetas), įvertindamas sutarties rizikingumą. Po sutarties pasirašymo pagaminama kortelė ir suteikiamas limitas.

### 2.1.5.5 Mokesčių skaičiavimas

Klientas, naudodamasis lizingo paslaugomis moka palūkanas lizingo bendrovei. Pirkimo išsimokėtinai kortelėje palūkanos skaičiuojamos už kiekvieną dieną pagal tos dienos faktinį likutį ir nustatytą sutartyje metinę palūkanų normą. Pvz. Jei klientui buvo suteiktas 1000 Lt limitas o jis išnaudojo tik 500, jam palūkanos bus skaičiuojamos nuo išnaudotos sumos (500). Tokiu būdu klientas moka lizingui tik už tą sumą kurią jis realiai išnaudojo, o čia dar vienas privalumas klientui. Jei klientas vėluoja įmokėti įmoką, jam skaičiuojamas kitas vėlavimo mokestis (delspinigiai) pagal jo nesumokėtą sumą (skolą) ir sutartyje nustatytą tarifą už kiekvieną dieną. Mokesčiai nurašomi nuo pirkimo išsimokėtinai kortelės sąskaitos kiekvieną mėnesio mokėjimo dieną.

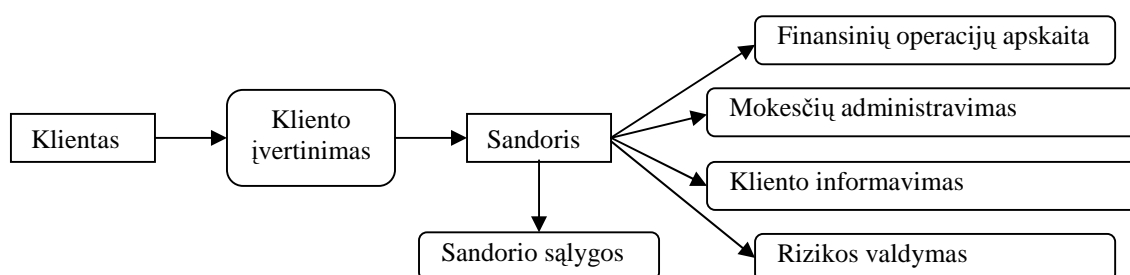
### 2.1.5.6 Klientų informavimas

Klientai kiekvieną mėnesį informuojami apie jų atliktas operacijas bei išnaudotą limitą. Klientai gali rinktis ataskaitų gavimo būdą:

1. Ataskaitų gavimas paštu;
2. Ataskaitų gavimas elektroniniu paštu;
3. Ataskaitų gavimas SMS žinutėmis;

Klientai taip pat informuojami perspėjamaisiais laiškais apie savo įsipareigojimų nevykdymą.

### 2.1.5.7 Verslo valdymo įvykio (sandorio) schema



### 2.1.6 Duomenų bazės samprata, duomenų gavyba

Sistema skirta duomenims apie subjektus kaupti. Pagrindinis skaitmeninio pavidalo duomenų kaupimo būdas – duomenų bazės. Darbui pasirinktas reliacinis duomenų bazės modelis. Plačiau apie duomenų bazes bei jų valdymo sistemas rasite 3 priede (Duomenų bazių samprata).

## 3 Projektinė dalis

### 3.1 Uždavinys ir jo analizė

**Uždavinys:** reikia sukurti programą, leidžiančią kaupti informaciją apie subjektus. Informacijos kaupimo paskirtis - išlaikyti glaudesnius santykius su klientais, optimizuoti klientų aptarnavimo kokybę. Taip pat realizuoti pirkimo išsimokėtinai kortelės apskaitos modulį, kortelė bankinių operacijų ir mokesčių apskaitos modulius bei klientų rizikos valdymo modulį.

### 3.2 Projektas

Įvertinus ir atrinkus duomenis, kurie bus saugomi, reikia juos išskaidyti į atskiras grupes pagal prasmę ir nustatant ryšius tarp grupių. Tokia duomenų dekompozicija reikalinga duomenų bazės projektavimui. Pagal reliacinį duomenų bazių modelį, duomenys saugomi lentelėse, atitinkančiose tam tikrus reikalavimus (normalizuotas formas), kurios susijusios tam tikrais ryšiais, padedančiais išlaikyti duomenų bazės vientisumą.

**Normalizacija** – tai įvairių taisyklių taikymas, siekiant supaprastinti sąryšius. Kitaip sakant, tai yra reliacinių lentelių suvedimas iki standartinio pavidalo. Ji padeda išvengti dėl neteisingo duomenų bazės projektavimo atsirandančių problemų:

Duomenų perteklius - duomenų pasikartojimas duomenų bazėje.

Duomenų vientisumas – duomenų suderinimas (tikslumas ir neprieštaringumas) duomenų bazėje.

- Atnaujinimo anomalija – duomenų tarpusavio prieštaravimas dėl duomenų pertekliaus ir dalinio atnaujinimo.
- Pašalinimo anomalija – nenumatytas duomenų praradimas dėl kitų duomenų šalinimo.
- Ivedimo anomalija – kai neįmanoma įvesti duomenis dėl kitų duomenų nebuvimo.

Pagrindiniai reikalavimai lentelėms:

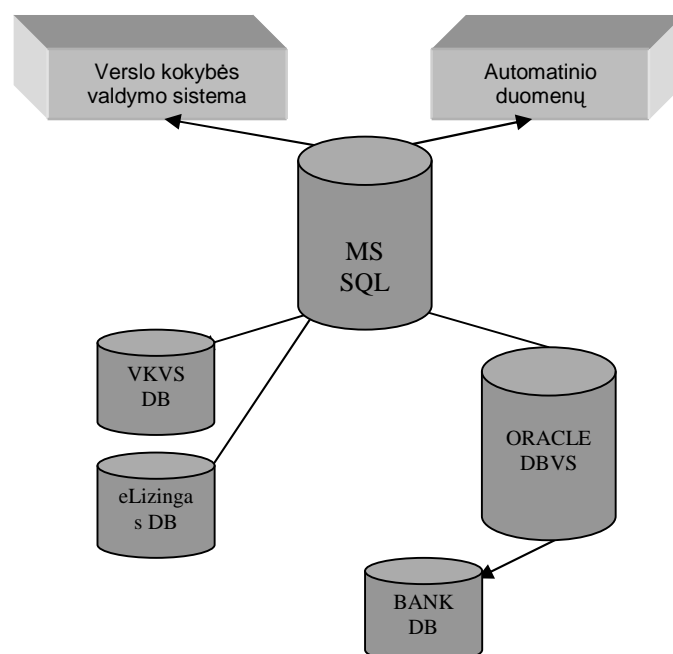
- Lentelės visose stulpeliuose (atributuose) saugomos nedalomos reikšmės ir nėra pasikartojančių stulpelių reikšmių viename įrašė. Prasmė – lentelės skaidymo procesas pašalina pasikartojančius reikšmių stulpelius ir tuščias vietas. Negali būti vienodos prasmės atributų nesvarbu, ar jų reikšmės sutaps.
- Kiekviena lentelė, turi turėti pirminį raktą – atributą, vienareikšmiškai nusakantį kiekvieną tos lentelės įrašą. Čia atliekamas lentelių skaidymas, atsiranda žinynei.
- Ryšys tarp bet kurių dviejų lentelių turi būti vienintelis, negalima sudaryti ciklinių ryšių.

### 3.3 Duomenų bazės projektavimo etapai

Sistemos duomenų bazei pasirinktas reliacinis modelis, duomenų bazė suprojektuota laikantis trijų pagrindinių normalinių formų. Pagrindiniai duomenų bazės projektavimo etapai:

- Susisteminti kiekvieno modulio duomenys;
- Atlikta duomenų dekompozicija siekiant įvykdyti reliacinei duomenų bazei keliamus normalizacijos reikalavimus.
- Pagal išskirtas duomenų grupes sukurtos duomenų bazės lentelės;
- Nustatyti duomenų bazės vientisumą užtikrinantys sąryšiai tarp lentelių.

### 3.4 Duomenų apdorojimo schema





### 3.5 Darbo priemonių parinkimas

Sparčiai tobulėjant kompiuterinei įrangai, keičiasi ir programinė kompiuterių įranga, o kartu ir programavimo sistemos bei kalbos. Programavimo kalbos ir sistemos pasirinkimą dažniausiai lemia tos kalbos ir sistemos galimybės. Šis darbas atliktas naudojant *Delphi 7* sistemą, kur realizuota *Object Pascal* programavimo kalba. Ši programavimo kalba palaiko klasių mechanizmą. Taikant objektinio programavimo metodiką, programa sudaroma iš abstraktaus tipo objektų, kurie turi tokias savybes:

- yra aprašomi parametrų rinkiniais, kurie nusako jų būvį;
- turi būvių apklausos ir valdymo priemonių (metodų) rinkinius;
- turi specialius objektų konstravimo ir naikavimo metodus;
- gali turėti vidinius (uždarus) ir išorinius (atvirus) elementus;
- iš kiekvienos abstraktaus tipo objektų klasės gali būti kuriamos išsistos išvestinių tipų šeimos, paveldinčios bazinio tipo savybes;
- tipų šeimose gali būti automatiškai keičiamos metodų savybės.

Projektuojant objektus, reikia turėti priemones, kurios leistų duomenų struktūras ir jų apdorojimo metodus sujungti į vientisą struktūrinį vienetą. Toks jų sujungimas realizuojamas klasėse. Delphi sistema turi savo klasių biblioteką VCL (Visual Component Library). Delphi sistema palaiko PME (Property – Method – Event) modelį, t.y. objektai turi savybes (Property), apibūdinančias to objekto stovį, turi metodus (Method)– funkcijas, apdorojančias savybių reikšmes ir kitus duomenis. Metodai iškviečiami įvykus kokiam nors įvykiui (Event).

Kitos programavimo kalbos turi savo bazinių klasių bibliotekas. Visual C++ sistema turi MFC (Microsoft Foundation Classes) klasių biblioteką, čia realizuotos didesnės klasių kūrimo galimybės. Be to, C kalbos sintaksė žymiai efektyvesnė lyginant su Object Pascal. Tačiau lyginant Visual C++ ir Delphi sistemas, nemažai daliai programuotojų, o ypač su mažesne programavimo patirtimi, priimtinesnė Delphi, nes čia yra tokios priemonės kaip Object Inspector, kur galima greitai nustatyti pradines objektų savybių reikšmes. Be to, čia komponentų (VCL ir kitų klasių objektų) pradiniai nustatymai gali būti valdomi vizualiai. Tai leidžia greitai ir efektyviai kurti patogią ir modernią vartotojo grafines sąsają. Nepaisant mažesnio Object Pascal kalbos efektyvumo ir mažesnių klasių mechanizmo galimybių lyginant su C kalba, Delphi sistema tinka projektuoti ir kurti tiek mažom, tiek pakankamai didelėm sistemom.

Duomenų saugojimui pasirinktas **Microsoft SQL server 2005** duomenų bazių serveris. Šis serveris skirtas didelių duomenų bazių valdymui. Šis SQL servisas turi labai daug papildomų duomenų analizės bei apdorojimui skirtų priedų bei funkcijų, užtikrinančių kokybišką ir greitą duomenų apdorojimą. MS SQL Server trūkumas – pakankamai sunkus serverio administravimas. Tačiau lyginant su kitomis duomenų bazių valdymo sistemomis MS SQL serveris turi lankstų programavimo mechanizmą (Transact SQL), be to yra daugybė papildomų priemonių, palengvinančių duomenų importą iš failų (Excel, tekstinių bylų, XML) arba kitų DB serverių (pvz. Oracle). MS SQL serveris taip pat palaiko OLE technologijas, kas palengvina tokius uždavinius kaip duomenų importą iš WEB servisų ir panašiai.

Duomenų kaupimas jų neapdorojus neturi prasmės, todėl analizė buvo visais laikais. Kadangi sistema skirta duomenų (istorijos) kaupimui, svarbu turėti galimybes tinkamai tuos duomenis apdoroti, susisteminti, sugrupuoti. Reikia priemonių, kurios leistų bet kada gauti bet kokius duomenis. Programoje sunku numatyti visus vartotojui reikalingus duomenų pateikimo variantus. Tačiau yra priemonių, kurios leidžia tai padaryti. Viena tokių priemonių – **Business Objects**. Čia taikoma viena iš pagrindinių informacijos analizės sričių — duomenų gavyba (Data Mining). Business Objeks pagalba galima tuos pačius duomenis analizuoti įvairiais „pjūviais“.

### 3.6 Duomenų apdorojimas

Vartotojo sąsajai organizuoti (užklausoms formuluoti) yra vartojama tam tikra formalizuota kalba. Daugumoje reliacinių sistemų yra vartojama vienas iš kalbos *SQL (Structured Query Language)* dialektų. Ši kalba buvo sukurta firmos IBM tyrimų centre 1970-ųjų metų pabaigoje. Praktiškai nuo pačios pirmosios firmos IBM sukurtos reliacinės sistemos System R SQL kalba tapo Amerikos nacionaliniu (ANSI) bei tarptautiniu (ISO) standartais.

Didele dalimi, SQL kalba vartojama reliacinėms operacijoms aprašyti. Šioje kalboje išskiriamos trys sakinių grupės: duomenų apibrėžimo sakiniai, kitaip dar vadinami duomenų apibrėžimo kalba (DDL, Data Definition Language); manipuliavimo duomenimis sakiniai (DML, Data Manipulation Language) ir duomenų valdymo sakiniai (DCL, Data Control Language), t.y. SQL = DDL + DML + DCL.

*DDL* sakiniiais, tarp kitų veiksmy, yra kuriamos duomenų bazės ir lentelės. Kuriant naują DB yra nurodomas jos vardas (pavadinimas), fizinė vieta ir kai kurios kitos savybės. Kuriant (apibrėžiant) lentelę, privaloma nurodyti lentelės vardą, jos stulpelių vardus ir stulpelių tipus. Stulpelio tipu nusakoma, kokios rūšies duomenys galės būti įrašomi stulpelyje. SQL leidžiamos

tokios duomenų rūšys: tekstiniai duomenys, skaičiai, dvejetainiai duomenys, datos ir laikai. Kiekviena duomenų rūšis su tam tikromis fizinėmis realizacijos savybėmis (pvz., baitų kiekis, skiriamas reikšmės vaizdavimui) nusakoma **duomenų tipu**. Duomenų tipo sąvoka yra reliacinės teorijos domeno sąvokos atitikmuo.

**DML** sakiniiais formuluojamos užklauskos (angl. query) duomenims surasti, keisti, šalinti ir įterpti.

**DCL** sakiniiais koordinuojamas bendras efektyvus vartotojų darbas su DB. SQL - tai kalba, kuria galima bendrauti su reliacinėmis duomenų bazėmis.

Dažnai realiame informacijos valdyme duomenys būna nežinomi arba nepilni. Atsižvelgiant į fakto apie duomenų nebuvimą svarbą, SQL kalboje kiekviename duomenų tipe yra speciali reikšmė **NULL**. Šia reikšme žymima, kad duomenys yra nežinomi, šiuo metu dar neįvesti, jau pašalinti ir pan. Tai lyg reikšmė, reiškianti reikšmės nebuvimą.

Formuluojant užklauskas SQL kalba, reikia išreikšti tai, kas norima gauti, nesileidžiant į detales, kaip rezultatą gauti. DBVS uždavinys - nustatyti kaip, koku būdu, kokius algoritmus panaudoti rezultatui gauti. Todėl, SQL gali būti priskirta neprocedūrinėms (deklaratyvioms) programavimo kalboms. Kita ypatybė, kuri daro SQL kalbą netradicine programavimo kalba, yra SQL sakinių vartojimas programuojant programavimo kalbomis. SQL sakiniai gali būti įterpiami į programas, sudaromas bazinė programavimo kalba. Bazinėmis programavimo kalbomis SQL kalbai gali būti daugelis plačiai vartojamų programavimo kalbų: C/C++, COBOL, FORTRAN, DELPHI ir pan. Kita vertus, SQL sakiniai gali būti vykdomi ir interaktyviai.

Pirmasis tarptautinis SQL kalbos standartas buvo priimtas 1989 m., kuris vadinamas SQL-89 arba SQL1. Dauguma šiuolaikinių RDBVS šį SQL standartą pilnai atitinka. Besivystančios informacinės technologijos įtakojo ir SQL vystymąsi. 1992 m. buvo priimtas kitas standartas, kuriam prigijo SQL-92 ir SQL2 pavadinimai. Šiuo metu, tai - reikalavimai, kurių prisilaiko dauguma komercinių RDBVS. 1999 m. pasirodė naujas standartas SQL3. SQL2 nuo SQL1 skiriasi, daugiausia, kiekybiniais parametrais. SQL3 standarto kalba išsiskiria gana dideliais kokybiniais pakitimais. Į SQL3 įvesti nauji duomenų tipai, numatyta galimybė apibrėžti naujus struktūrinius duomenų tipus. Naujų galimybių atsiradimą labai įtakojo objektinių technologijų paplitimas. Šiandien RDBVS vystosi, siekdamos įdiegti SQL3 standarto galimybes.

SQL sakinių sintaksė konkrečiose DBVS skiriasi tiek nuo standarto, tiek ir tarpusavyje. Praktiškai, SQL dialektų skaičius yra lygus komercinių RDBVS skaičiui.

Duomenų perdavimui į kitas įstaigas naudojamas XML formatas.

**XML (ang. eXtensible Markup Language)** yra bendros paskirties duomenų struktūrų bei jų turinio aprašomoji kalba. Pagrindinė XML kalbos paskirtis yra užtikrinti lengvesnį duomenų keitimąsi tarp skirtingo tipo sistemų. Pagrindinis XML kalbos vienetas yra *elementas*. Elementas visada turi vardą ir, be jo, gali turėti:

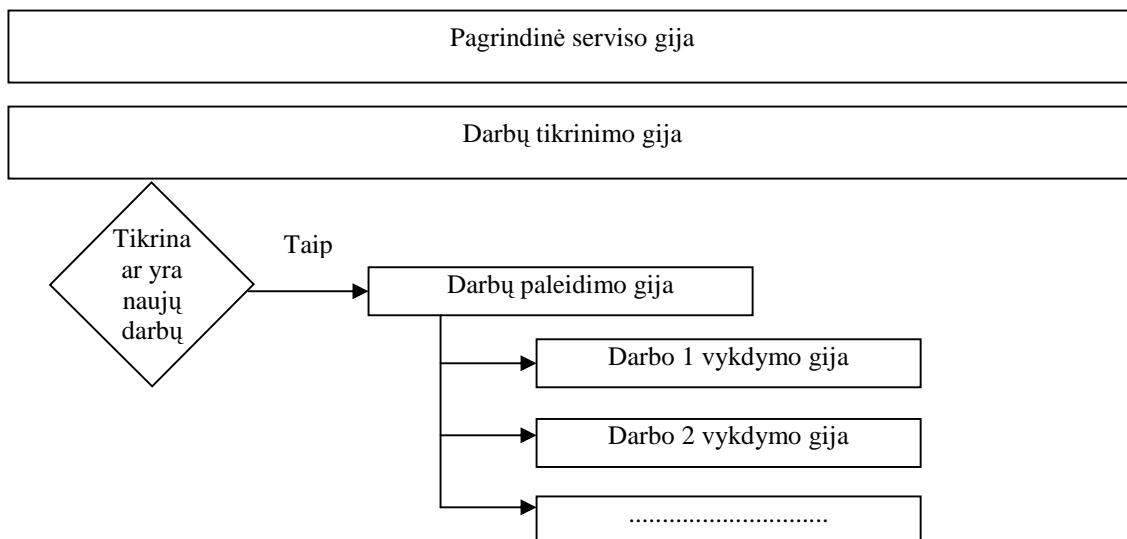
- § norimą skaičių atributų. Atributas turi savo vardą bei reikšmę.
- § kitus (dukterinius) šio elemento viduje esančius elementus.
- § su elementu susijusį tekstą.

Pagrindinis XML privalumas - galimybė pridėti naujo tipo elementus nepažeidžiant dokumento struktūros ir nesutrikdant tik seną formatą suprantančių programų darbo.

### 3.7 Programavimo technologijos

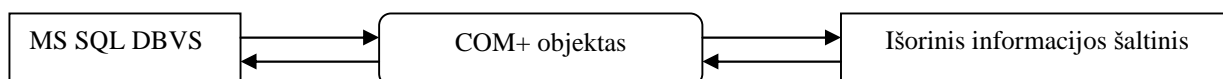
Sistemos kūrimui naudojamos kelios programavimo technologijos.

Viena iš jų – **lygiagretus programavimas**. Ši technologija leidžia vienu metu atlikti keletą skirtingų veiksmų. Ji panaudota automatinio duomenų apdorojimo servisui kurti. Servisui registruojami darbai automatiniam apdorojimui nurodant konkretų kiekvieno darbo paleidimo laiką bei periodiškumą. Serviso veikimo principas – kas tam tikrą laiko intervalą tikrinti ar yra darbų, kuriuos reikia vykdyti tuo laiko momentu. Jei jų yra – tie darbai pradedami vykdyti. Standartinėmis objektinio ar procedūrinio programavimo priemonėmis vienu metu programa gali atlikti tik vieną darbą. Tai reiškia, kad kol vykdomas vienas darbas, kiti darbai laukia. O kol pats servisas vykdo kokį nors darbą, jis negali atlikti kitų darbų patikrinimo. Todėl toks servisas būtų visiškai netinkamas nes darbai būtų vykdomi ne laiku, o kai kurie net iš vis neįvykdomi. Lygiagretaus programavimo atveju kiekvienas serviso atliekamas veiksmas turi savo atskirą giją (thread), kurios veikia nepriklausomai viena nuo kitos. Viena gija atlieka naujų darbų patikrinimo procesą ir jei randa tokių darbų kuriuos reikia vykdyti, paleidžia kitą giją, kuri atlieka darbų vykdymą. Ši technologija itin efektyviai išnaudojama kai serveris (techninė įranga) turi keletą procesorių.

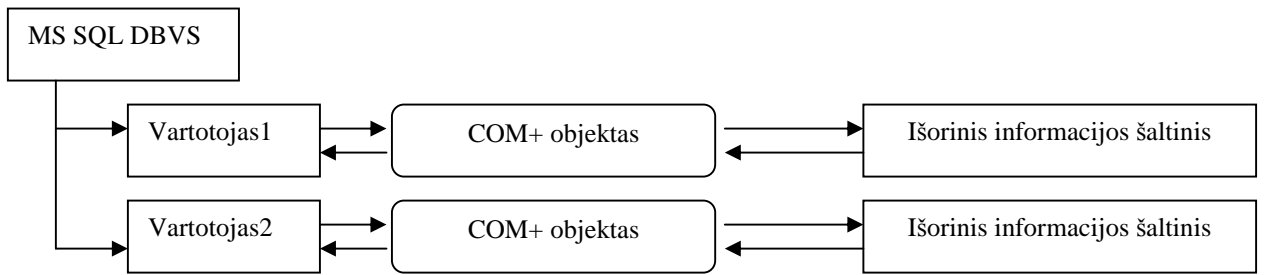


Kita automatinio duomenų apdorojimo servise panaudota programavimo technologija – **Windows servisų** kūrimo. Automatinio duomenų apdorojimo servisas gali veikti ir kaip atskira programa ir gali būti įdiegta kaip Windows servisas. Pastarasis būdas užtikrina serviso veikimo stabilumą bei apsaugo nuo netyčinio serviso sustabdymo (programos uždarymo), nes tada servisą paleisti ir stabdyti galima tik per Windows Services servisų valdymo modulį arba komandinės eilutės servisų valdymo komandomis.

Dar viena šiame darbe naudojama technologija – **COM+ servaisi**. COM+ servisas tai tarsi atskira programa, paleidžiama sukūrus OLE objektą. Darbe šis servisas naudojamas atlikti automatizuotiems veiksams, kurių negali atlikti duomenų bazių serveris. Kitas COM+ serviso privalumas – kad jį galima iškviešti iš MS SQL serverio. COM+ servaisi gelbsti ir tada kai reikia apdoroti ir importuoti išorinio šaltinio informaciją, pvz. duomenis iš SODROS duomenų bazės, arba duomenis apie skolas iš bendro subjektų skolų registro.



COM+ objektas gali paimti informaciją iš išorinio duomenų šaltinio, ją apdoroti ir įrašyti į duomenų bazę. O duomenų bazė prireikus atlikti duomenų importą iškviečia COM+ objektą. Kitas COM+ privalumas tas, kad kiekvienai duomenų bazės vartotojo sesijai galima iškviešti atskirus COM+ objektus, tai reiškia, kad skirtingi vartotojai gali vykdyti duomenų importą tuo pačiu metu. Jei šitoks elgesys neleistinas (pvz. COM+ atlieka importą, tačiau vienu metu galima vykdyti tik vieną ar kelis importo veiksmus), tada galima atlikti COM+ objektų sinchronizaciją ir vienu metu leisti veikti tik vienam ar keliems objektam kitus statant į eilę.



Dar viena programavimo technologija – **automatizuotas dokumentų ruošimas** taip pat besiremiantis OLE objektų kūrimu. Čia naudojamos MS Word tekstų redaktoriaus galimybės. Veikimo principas remiasi Word dokumente sudėliotomis žymelėmis (Bookmarks). Sukuriamas Word programos šablonas (dot byla) ir vietoje teksto, kurį reikia užpildyti duomenimis iš programos, įstatomos žymelės su tam tikru vardu. Programa perskaito dokumente esančių žymelių pavadinimus, pagal iš anksto paruoštą užklausų sąrašą ištraukia reikiamus duomenis ir juos įrašo tų žymelių vietoje. Toks dokumentų ruošimo būdas labai patogus, nes pats dokumento maketas gali būti lengvai koreguojamas.

## 4 Darbo eigos aprašymas

### 4.1 Duomenų bazės realizacija

Atlikus susistemintų duomenų normalizavimo procesą buvo suprojektuota reliacinė duomenų bazė, tenkinanti pirmas tris reliacinės duomenų bazės normalines formas. Sudėtingesni duomenų bazių serveriai (MS SQL serveris, ORACLE, DB2) palaiko ne tik SQL standarte numatytas funkcijas, tačiau čia yra galimybė ir kurti vartotojo funkcijas bei procedūras, taip pat vartotojo duomenų tipus, trigerius. Todėl kuriant ir šios sistemos duomenų bazę buvo stengiamasi padaryti viską taip, kad kiek įmanoma daugiau sistemos funkcionalumo būtų realizuota pačioje duomenų bazėje o ne aplikacijoje. Buvo sukurta daug papildomų skaičiavimams atlikti naudojamų funkcijų bei duomenų apdorojimui skirtų procedūrų. Duomenų bazės pagrindinių lentelių sąryšių schemą rasite 2 priede (Programuotojo vadovas).

### 4.2 Projekto moduliai

Sistemoje realizuota keletas modulių:

<b>Verslo ir verslo kokybės valdymo sistema VKVS Everestas</b>	
Verslo valdymo sistemos moduliai	
	Produktai
	Mokesčių skaičiavimo schemas
	Klientų rizikos valdymo schemas
	Pasiūlymų pateikimo bei sutarčių formavimo
	Mokesčių apskaitos
	Bankinių operacijų (transakcijų) apskaitos
	Klientų rizikos valdymas (skolų sudengimas)
Verslo kokybės valdymo sistemos moduliai	
	Administravimas
	Subjektų sąrašas
	Kontaktų centras
	Dokumentų šablonai
	Klientų informavimo servisas
	Klientų duomenų tvarkymo sutikimai
	Marketingas

	Inventorius
	Vidinių naujienų ir darbų skirstymo modulis
	Registrai
<b>Automatinis duomenų apdorojimas</b>	
	Automatinio duomenų apdorojimo servisas (Windows serviso technologija)
	Informacinių pranešimų siuntimo servisas
	Ataskaitų El. paštu generavimo servisas (COM+ technologija)
	Ataskaitų paštu generavimo servisas (COM+ technologija)
	SMS žinutes siuntimo servisas (COM+ technologija)
	XML duomenų bylos generavimo servisas (COM+ technologija)

#### 4.2.1 Pirkimo išsimokėtinai kortelės (verslo valdymo sistemos) moduliai

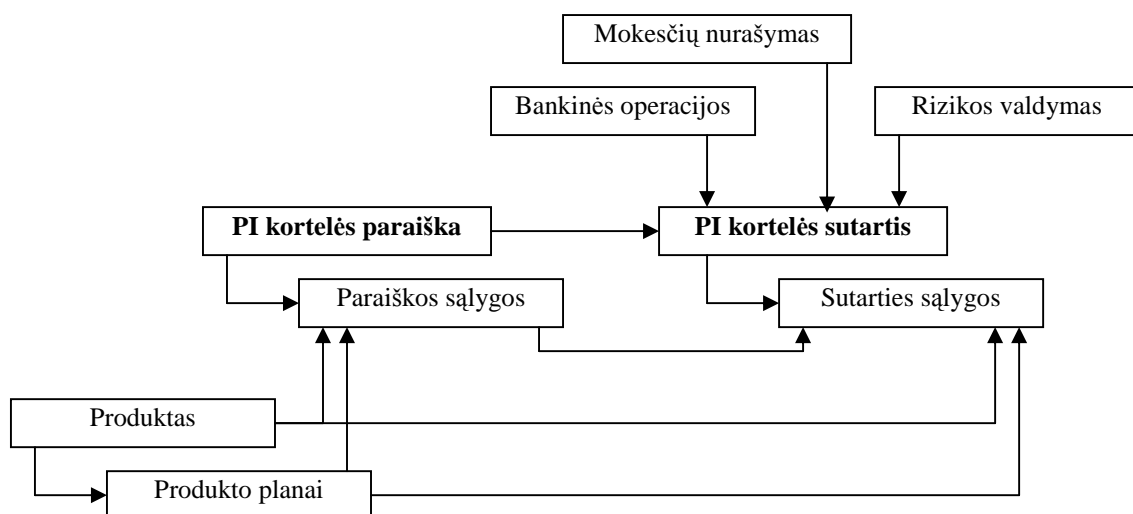
Pirkimo išsimokėtinai kortelės verslo valdymo sistemai reikalingi keli moduliai:

- **Mokesčių skaičiavimo schemas.** Aprašomi mokesčių skaičiavimo šablonai pagal kuriuos bus skaičiuojami sutarčių mokesčiai, nurodomi mokesčių tarifai bei kokie mokesčiai bus skaičiuojami, kokie ne (pvz. palūkanos, viršyto kredito palūkanos, delspinigiai).
- **Klientų rizikos valdymo schemas.** Aprašomi veiksmai, kurie turi būti atliekami klientui tapus nemokiu bei esant netinkamam sutarties sąlygų vykdymui. Galimi veiksmai – automatinis perspėjamojo laiško siuntimas klientui, vindikacijos tarnybos perspėjimas apie nemokų klientą (pasiūlymas nutraukti sutartį).
- **Produktai.** Tai modulis, skirtas kortelės produktams aprašyti. Produktas – sutarties šablonas, kuriame aprašomi standartiniai projekto nustatymai (mokesčių skaičiavimo schemas, rizikos valdymo schemas, klientų grupės kuriems produktas taikomas ir kt.)
- **Pasiūlymų pateikimo bei sutarčių forminimo** modulis skirtas automatiškai (be darbuotojo įsikišimo) įvertinti kliento mokumo galimybes bei jam pateikti komercinį pasiūlymą pasirenkant kliento grupę atitinkantį produktą. Jei šis pasiūlymas klientui priimtinas, pereinama prie sutarties sudarymo. Esant išskirtiniams atvejams kliento įvertinimą atlieka lizingo bendrovės darbuotojai. Sutarčių forminimo modulis saugo informaciją apie kiekvieną sutartį:
  - sutarties klientas;



- sutarties informacija (numeris, data, statusas);
  - sutarties sąlygos (kredito limitas, terminas, mokėjimo diena, mokesčių tarifai, sutarties būseną);
  - mokėjimų grafikas pagal kurį turi būti gražinamas kreditas;
  - sutarties sąlygų keitimo faktas (kas kada ir kokias sąlygas keitė);
  - papildomos garantijos, laidavimai.
- **Mokesčių apskaitos** modulis skirtas palūkanų, delspinigių bei kitų mokesčių apskaitai bei nurašymui. Kiekvienam aktyviam projektui (sutarčiai) kiekvieną dieną skaičiuojami mokesčiai. Kiekvieną mėnesį mokėjimo dieną priskaičiuoti mokesčiai susumuojami pagal jų tipus (palūkanos, delspinigiai ir kt.) bei nurašomi nuo kliento sąskaitos į lizingo bendrovės sąskaitą.
  - **Bankinių operacijų apskaitos** modulis skirtas peržiūrėti klientų atliktas pinigines operacijas kortele atsiskaitant už prekes arba išgryninant pinigus. Atliktos operacijos importuojamos iš bankinės sistemos, čia jos išskaidomos pagal projektus bei įtraukiamos į projekto apskaitą.
  - **Klientų rizikos valdymas.** Kiekvieną dieną atliekamas visų aktyvių projektų klientų mokumo patikrinimas bei vėlavimo skaičiavimas. Klientui vėluojant mokėti įmoka nustatoma jo rizikos grupė bei pagal projektui prisikirtą rizikos valdymo schemą atliekami atitinkami veiksmai (perspėjamojo laiško išsiuntimas, vindikacijos tarnybos įspėjimas apie nemokų klientą).

Verslo valdymo modulių tarpusavio sąryšio schema:



#### 4.2.2 Įmonės verslo kokybės valdymo moduliai

**Administravimas** – modulis, skirtas duomenų bazės administravimui automatizuoti. Čia registruojami ir iš šio modulio tiesiogiai vykdomi duomenų koregavimai. Duomenų keitimo funkcijas šiame modulyje galima bet kada pakeisti ir papildyti, todėl jo funkcionalumas labai įvairus.

**Administravimo** modulyje taip pat realizuotas įmonės vidinių problemų registras. t.y. įmonės darbuotojai registruoja jiems iškilusias problemas, kurių jie savarankiškai pašalinti neturi galimybės. Atsakingi darbuotojai seka problemų registrą, nustato jų sprendimo galimybes bei terminus. Darbuotojai taip pat išreikšti savo pageidavimus bei pasiūlymus.

**Subjektų sąrašas** – modulis, skirtas informacijai apie klientus ir verslo partnerius kaupti. Šis modulis ko gero svarbiausias sistemoje. Pateikiamas subjektų (klientų, pardavėjų, verslo partnerių) sąrašas. Galima atsidaryti norimo subjekto kortelę su detalia, į grupes suskirstyta informacija apie jį:

- Subjekto kortelė – pagrindinė informacija apie subjektą: vardas, pavardė arba pavadinimas, asmens tapatybės dokumento informacija, prioritetas, grupė, pagrindinė darbovietė bei pareigos, adresų bei kontaktinių duomenų sąrašai, nuotrauka, rolės.
- Darbovietės – galima saugoti subjekto darbovietes, kontaktinius subjekto duomenis toje darbovietėje, pareigas, įsidarbinimo bei išėjimo iš darbo datas.
- Pastabos – laisvai renkamas tekstas, kur galima įvesti bet kokią nesistemintą informaciją.
- Sąskaitos ir kortelės – galima registruoti kokias ir kokiame banke klientas turi sąskaitas bei korteles prie jų.
- Pajamos ir įsipareigojimai – galima registruoti klientų finansinę būklę apibūdinančius dokumentus – pajamų deklaracijas, bei įsiskolinimus įvairioms institucijoms.
- Poreikiai – registruojami subjekto poreikiai, nurodant statusą, datą bei ar jis dar aktyvus.
- Dokumentai – saugomi su tuo subjektu susiję dokumentai: pavadinimas, grupė, data, dokumento turinys (galima įkelti bet kokią bylą [Word, Excel, video, muziką, paveiksliukus ir kt.]).

- Veiksmai – veiksmų, atliekamų su subjektu registras. Galima registruoti veiksmų planą, kiekvieną veiksmą skirti atsakingam darbuotojui. Veiksmą taip pat galima skaidyti į atskiras užduotis ir jas išskirstyti darbuotojams. Galima suformuoti darbų atlikimo medį, t.y. įmonės vadovas gali registruoti veiksmą bei jį išskaidyti į užduotis, skirdamas jas atitinkamai skyrių vadovams. Skyrių vadovai gavę paskirtą užduotį, ją gali dar išskirstyti savo pavaldiniams ir t.t. Grandinės ilgis neribojamas. Galima nustatyti nuo kada iki kada užduotis turi būti įvykdyta bei užstatyti priminimus norimą dieną bei laiką. Veiksmai skaidomi į tris grupes: pradelsti veiksmai, šios dienos veiksmai bei ateities veiksmai.
- Marketingas – galima registruoti su subjektu susijusią marketinginę informaciją (nuolaidos suteikimas, dovana ir kt.)
- Sutartys – sutartys (pirkimo išsimokėtinai) su subjektu.
- Rizikos grupės – subjektai skirstomi į rizikos grupes. Čia pateikiama dabartinė subjekto rizikos grupė bei jų keitimo istorija.
- Filialai – juridinių asmenų informacija, registruojami subjekto filialai su adresu bei kontaktiniais duomenimis.
- Kontaktiniai duomenys – registruojami subjekto kontaktiniai asmenys (pagrindinė informacija).

**Kontaktų centras** – modulis, labai susijęs su subjektų sąrašu. Čia kiekvienas vartotojas individualiai gali susiformuoti subjektų sąrašą, su kuriuo jam dažniausiai tenka dirbti.

**Dokumentų šablonai** – sukaupti visų įmonėje naudojamų dokumentų šablonai. Vartotojas gali pasinaudoti bet kuriuo šablonu jam prireikus. Šablonų informacija gali būti pildoma automatiškai t.y. galima automatiškai užpildyti darbuotojo duomenis (pvz. Prašymas, tarnybinis raštas)

**Klientų informavimo servisas** – modulis, skirtas automatizuotam marketinginės bei ataskaitų informacijos siuntimui. Modulyje registruojami informavimo įvykiai, kiekvienas įvykis turi nurodytus gavėjus bei prisegtus dokumentus nurodant kurie gavėjai turi gauti kiekvieną dokumentą. Laiškų išsiuntimą atlieka pagalbinis laiškų siuntimo servisas, kiekvieną dieną nurodytu laiku išsiunčiantis neišsiųstus laiškus. Automatiniam įvykių registravimui bei įvykio gavėjų ir siunčiamos informacijos registravimui (pvz. kiekvieno mėnesio ataskaitų siuntimo įvykiai) naudojami papildomi servisirai. Pavyzdžiui, kiekvieną mėnesį papildomas COM+ technologijos servisirai registruoja klientų informavimo įvyki, registruoja visus ataskaitų gavėjus bei kiekvienam gavėjui sugeneruoja ataskaitą HTML formatu, kurią išsaugo kaip siunčiamą dokumentą.

**Klientų duomenų tvarkymo sutikimai** - prieš pradėdant registruoti fizinio asmens duomenis įmonės sistemoje, įmonės darbuotojai turi gauti iš kliento sutikimą jo duomenų tvarkymui. Todėl realizuotas klientų sutikimų kontrolės registras.

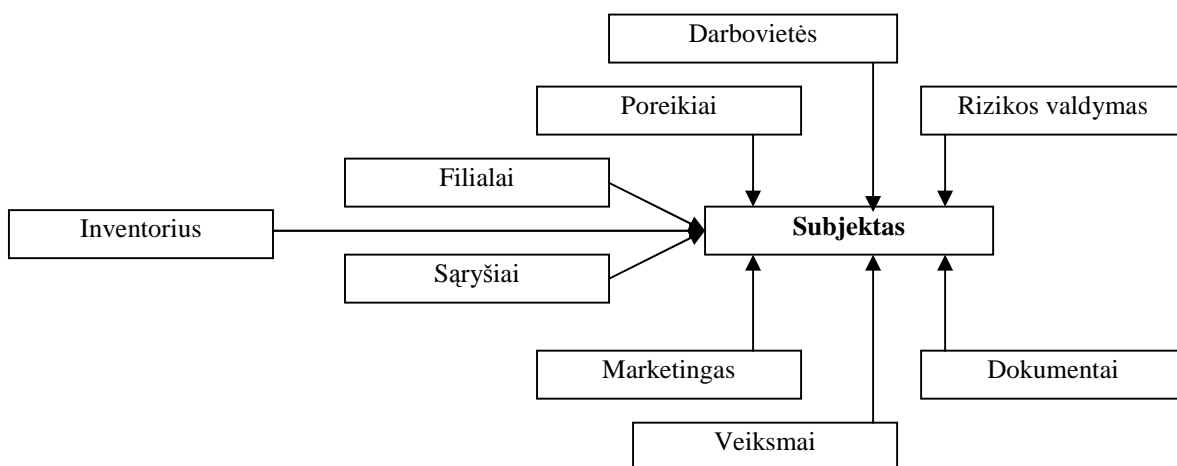
**Marketingas** – modulis, skirtas registruoti marketingo veiksams bei įvykiams. Pvz. Klientui išsiųsta dovana, rengta akcija, akcijos laimėtojų paskirtas prizas ir pan.

**Inventorius** – modulis, skirtas įmonės turimam inventoriui registruoti.

**Vidinių naujienų bei darbų skirstymo** – modulis, skirtas vidinių naujienų paskelbimui bei vartotojo dienvakės valdymui. Darbą gali registruoti pats vartotojas sau arba jei vartotojas turi pavaldinių, jis gali registruoti ir paskirti darbą pavaldiniui. Darbą galima suskaidyti į atskiras užduotis ir kiekvieną užduotį paskirti skirtingam pavaldiniui. Pavaldinys gauna žinutę apie jam paskirtą užduotį ir ją gali dar skaidyti į smulkesnes užduotis.

**Registrai** – įvairūs klasifikatoriai ir sąrašai, pvz. miestų, kaimų, rajonų sąrašai, subjektų rolės, valiutos ir kt.

Verslo kokybės valdymo modulių schema:



Kaip naudotis kiekvienu modulių rasite 1 priede (vartotojo vadove).

#### 4.3 Automatinio duomenų apdorojimo servisas

Kadangi programos funkcionalumas reikalauja daug automatizuoto duomenų apdorojimo, vien MS SQL serverio galimybių nepakanka. Reikalinga papildoma priemonė, galinti atlikti automatizuotą duomenų apdorojimą be žmogaus įsikišimo. Tam tikslui sukurtas *automatinio duomenų apdorojimo servisas*, atliekantis numatytas duomenų apdorojimo procedūras nustatytu laiku. Šis servisas atlieka tokius duomenų apdorojimus kaip bankinių transakcijų importą iš

bankinės sistemos, sutarties mokesčių paskaičiavimą ir grafikų užskaitymą, klientų rizikos įvertinimą, automatinį persėjamųjų laiškų bei sutarties ataskaitų siuntimą, automatinį duomenų bazės kopijų (backup) darymą. Visi veiksmai gali būti paleidžiami nurodytu laiku, galima iš karto vykdyti automatinio duomenų apdorojimo procedūrų seką.

Automatinam duomenų išsiuntimui klientams realizuota dar viena pagalbinė sistema – Pranešimų siuntimo servisas. Ši sistema patikrina ar yra registruotų naujų įvykių klientų informavimo modulyje. Jei tokių įvykių yra, juos išsiunčia atitinkamai pagal tipą Elektroniniu paštu arba SMS žinute. Kad pranešimus sistema siuntinėtų efektyviau naudojama lygiagretaus programavimo technologija.

#### 4.4 Darbas su keliomis DB ir DBVS

Šią verslo kokybes valdymo sistemą galima suderinti su kitomis verslo valdymo sistemomis. Šio suderinimo pavyzdys – UAB „Šiaulių banko lizingas“, kur sistema suderinta su lizingo verslo valdymo sistema eLizingas (UAB „Duomenų bazės ir technologijos“) bei su daugumoje bankų Lietuvoje naudojamose banko verslo valdymo sistemų Forpost (UAB „Forbis“).

#### 4.5 Darbe iškilusios problemos bei jų sprendimas

Viena iš sudėtingesnių problemų buvo dokumentų (pvz. sutarčių, paraiškų) generavimas. Sukurtame Word sistemos dokumento šablone tam tikrose vietose reikia įterpti duomenis iš duomenų bazės. Šios problemos sprendimui buvo panaudotos OLE technologijos, kurių pagalba programa gali valdyti Word sistemą, perduoti jai duomenis bei nurodyti kurioje vietoje dokumente juos išvesti. Delphi programavimo sistema turi Word programos valdymui skirtus komponentus, kurie suteikia galimybę realizuoti šį funkcionalumą. Tačiau naudojant šią technologiją iškilo kita problema - Word versijų nesuderinamumas. Minėti Word valdymo komponentai skirti vienai tam tikrai Word versijai (2000, XP). O skirtingų vartotojų kompiuteriuose buvo įdiegtos įvairios Word versijos. Šios problemos sprendimui teko atsisakyti standartinių Delphi komponentų ir panaudoti dinaminį OLE objektų kūrimą bei valdymą. Tam Delphi sistemoje skirtas atskira klasė ***WordApplication: Olevariant; WordApplication := CreateOLEObject('Word.Application');*** Dinamiškai sukurtas Word OLE objektas veikia su bet

kuria Word (97, 2000, XP, 2003) versija. Šios problemos sprendimui buvo įsisavinta nauja programavimo technologija.

Kita problema iškilo realizuojant automatinio duomenų apdorojimo servisą. Realizavus sistemą standartinėmis technologijomis iškilo problema, kad kol vykdomas vienas duomenų apdorojimo veiksmas, sistema „pakimba“ kol jis nebus baigtas. Taip atsitinka todėl kad pagrindinė programos gija gali vienu metu vykdyti tik vieną veiksmą. Kol būdavo vykdomas vienas veiksmas, sistema negalėdavo laiku paleisti kitų veiksmų. Šios problemos sprendimui buvo panaudota lygiagretaus programavimo technologija, kai atskiriems veiksams vykdyti sukuriama atskira gija (thread). Tokios gijos gali kelias vykdyti veiksmus vienu metu. Kadangi šis servisas veikia serverinėje techninėje platformoje su keliais procesoriais, tokia programos architektūra ypač efektyviai išnaudojama. Kad užtikrinti automatinio duomenų apdorojimo stabilumą, sistema buvo realizuota kaip Windows sistemos servisas. Šios problemos sprendimui buvo įsisavintos dvi naujos programavimo technologijos.

Dar viena darbe iškilusi problema – duomenų importas iš išorinių duomenų šaltinių (SODROS duomenų bazės, Skolininkų bazės SAIS ). Šie išoriniai šaltiniai duomenis teikia Web servisų pagalba, t.y. internetiniu protokolu pateikiama užklausa išorinio šaltinio serveriams ir gaunamas atsakymas XML pavidalu. MS SQL serveris negali tiesiogiai daryti užklauskos iš Web serviso, todėl čia reikalinga programa – tarpininkas, atliekanti užklauską bei išsauganti rezultatą duomenų bazėje. Be to, reikia, kad vienu metu būtų galima vykdyti keletą užklauskų iš to paties Web serviso. Šią problemą išspręsti padeda COM+ servisų panaudojimas. Šie servais gali atlikti tokį tarpininko vaidmenį, be to vienu metu galima iškviešti keletą tokių objektų (pvz. kiekvienai vartotojo sesijai).

#### 4.6 Galutinis projekto stovis

Sistemoje buvo realizuotas lizingo įmonėje naudojamas verslo ir verslo kokybės valdymo funkcionalumas, sukurti programos moduliai, leidžiantys kaupti įvairia su klientais susijusia informaciją, atliekantys automatizuotus veiksmus, palengvinančius įmonės darbuotojų darbą bei užtikrinančius geresnę įmonės veiklos kokybę.

Tačiau tokio pobūdžio sistemas galima pavadinti „neturinčiomis pabaigos“. Visada galima sugalvoti papildomų modulių įmonės veiklai gerinti. Kadangi sistemoje realizuotas ne tik verslo kokybės bet ir verslo valdymo pirkimo išsimokėtinai kortelių modulis, sistemą galima papildyti ir pačiu lizingo projektų apskaitos moduliu.

## 5 Išvados

- Išanalizuotas matematinis periodinių mokėjimų modelis, šis modelis (anuiteto principas) realizuotas pirkimo išsimokėtinai kortelės grafiko sudarymui MS SQL duomenų bazės lygyje;
- Sukurtas statistinis lizingo veiklos modelis;
- Realizuota sistema lizingo įmonės verslo bei verslo kokybės valdymui, susidedanti iš keletos modulių bei servisų;
- Sukurtas automatizuoto dokumentų ruošimo principas;
- Papildomų servisų pagalba sistemą galima sujungti su išoriniais duomenų šaltiniais (SODRA duomenų baze, SAIS skolų baze, Registrų centro duomenų baze ir kt.);
- Lygiagretaus programavimo ir COM+ objektų technologijos sistemai suteikia daugiau funkcionalumo bei efektyvumo.
- Įvaldyta Microsoft SQL Server 2005 duomenų bazių valdymo sistema, pagilintos žinios apie reliacines duomenų bases, normalizavimo procesą bei reikalavimus normalizuotai duomenų bazei;
- Pagilintos SQL (Transact SQL) žinios;
- Pagilintos programavimo Delphi sistema žinios;
- Įvaldyta lygiagretaus programavimo Delphi aplinkoje technologija;
- Įvaldyta Windows sistemos servisų kūrimo technologija;
- Įvaldyta COM+ objektų kūrimo technologija;
- Verslo bei verslo kokybės valdymui šią programą naudoja UAB „Šiaulių banko lizingas“, ją taip pat galima pritaikyti ir kitoms įvairaus pobūdžio veikla užsiimančioms įmonėms.

## 6 Literatūra

1. Borland Software Corporation Delphi Language Guide. 2002
2. Vidžiūnas A., Blonskis J. Bukšnaitis B., Brazdaitis V. Delphi 5 . K., 2000
3. Blonskis J. Bukšnaitis B., Dagienė V. ir kt. Programavimas Delphi. V., 2003
4. Baronas R. Duomenų bazių sistemos V., 2002
5. "Naujoji komunikacija" Nr.6 (127), 2003 m. Balandžio 16 -30 d.
6. <http://www.swissdelphicenter.ch/torry/>
7. <http://www.delphi.lt>
8. <http://www.programmersheaven.com/>
9. <http://www.delphi3000.com>
10. <http://delphi.icm.edu.pl/ftp/d30share/INDEX>
11. <Http://bcb-tools.com>
12. <Http://visualcomponentlibrary.com/bcb>
13. <http://perkunas.vtu.lt/kompiuteriai/dbvs/>
14. <http://www.nkm.lt/052/index.html>
15. <http://www.alfitas.lt/start.html>
16. <http://gama.vtu.lt/orc/>
17. <http://www.it.lt/itweb/>
18. <http://www.geocities.com/brianaltnann/olaptopic-intro.html>
19. <http://www.olapreport.com/about.htm>
20. <http://www.sblizingas.lt>



## 7 Anotacija

Pocius, Tomas. Informatikos magistro baigiamasis darbas. Lizingo įmonės verslo kokybės valdymo sistema. Darbo vadovas doc. L. Sakalauskas. Šiaulių universitetas. – Šiauliai, 2006. – 34 lapai.

Šiame darbe nagrinėjama lizingo įmonės verslo bei verslo kokybės valdymo svarba jos veiklai, įtaka darbo rezultatams (pelniui). Lizingo įmonės verslo valdymas apima lizingo produktų (išperkamosios nuomos, pirkimo išsimokėtinai kortelių) pardavimą klientams. Įmonės kokybės valdymas apima priemones, palengvinančios pačios įmonės viduje vykstančių procesų kontrolę (naujų perdavimą darbuotojams, darbų paskirstymą, dokumentų administravimą, vidinių įmonės problemų sprendimą) bei priemones, leidžiančias glaudžiau bendrauti su klientais, greičiau reaguoti į naujus klientų poreikius.

Darbe iškelti pagrindiniai tikslai – sukurti sistemą lizingo įmonės verslo ir verslo kokybės valdymui. Realizuotos dvi modulių grupės: verslo valdymo (pirkimo išsimokėtinai kortelės moduliai) bei verslo kokybės valdymo moduliai. Pirkimo išsimokėtinai kortelių periodiniams mokėjimams apskaičiuoti naudojamas *anuiteto* principas. Sistemos stabiliam darbui užtikrinti pasirinktas reliacinis duomenų bazės modelis bei MS SQL 2005 duomenų bazių serveris.

Pagrindinės darbe iškilusios problemos buvo MS Word dokumentų automatinis užpildymas duomenimis bei įvairių MS Word versijų palaikymas, automatizuoto duomenų apdorojimo serviso kelių užduočių vykdymas vienu metu, duomenų importas iš išorinių duomenų šaltinių (failų, Web servisų). Problemų sprendimui panaudotos lygiagretaus programavimo bei servisų programavimo technologijos.

Buvo sukurta lizingo įmonės verslo ir verslo kokybės valdymo sistema bei papildomi automatinio duomenų apdorojimo servisi, atliekantys automatizuotus sistemos veiksmus (automatinį mokesčių skaičiavimą, ataskaitų klientams generavimą, informacinių bei reklaminių pranešimų išsiuntimą). Sistemą galima pritaikyti įvairaus pobūdžio veikla užsiimančioms įmonėms.

## 8 Summary

Pocius, Tomas. Informatics Master's Final Thesis. Business Quality Management System of a Leasing Company. Work leader ass. prof. L. Sakalauskas. Siauliai University. Siauliai, 2006. 34 pages

In the work I analyse management importance of leasing company business and its quality as well as its influence upon the work results (profit). Business management of a leasing company includes leasing production (leasing, hire purchase cards) sales to customers. Quality management of a company includes the measures, making easier internal control of the processes (sharing news with employees, work allocation, document administration, solving the company's internal problems) as well as measures, enabling closer co-operation with customers and quicker response to new customers' needs.

Basic aims of the work are to create a system for leasing company's business and business quality management. Two groups of modules are realized: business management (hire purchase cards management) and business quality management modules. Annuity principle is employed to calculate periodic payments for hire purchase cards. In order to assure stable work of the system, relation pattern of database and MS SQI 2005 database server are selected.

Main problems, encountered in the work, were automated filling of MS Word documents with data and support of different MS word versions, simultaneous execution of a number of tasks in automated data processing, data import from external data sources (files, Web services). Parallel programming and service programming technologies were used to meet the problems.

Leasing company business and business quality management system as well as extra automated data processing services, performing automated system actions (automated calculation of payments, report generation for customers, sending information and promotional messages) were compiled. The system is suitable to be applied in the companies engaged in various activities.

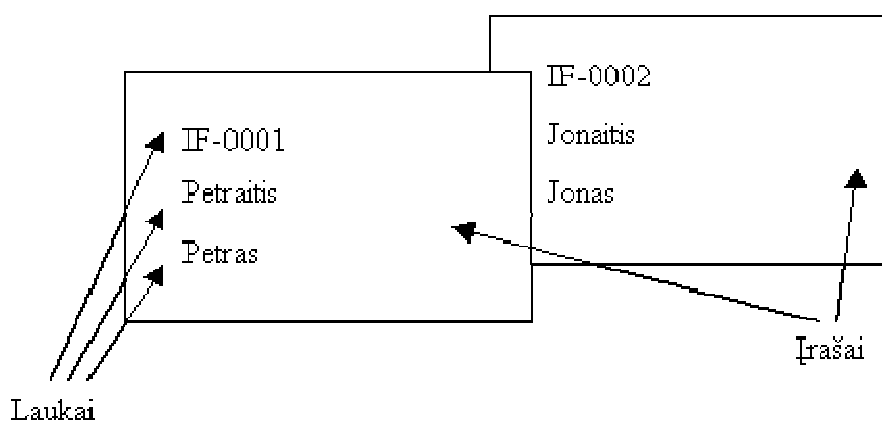
Duomenų bazių samprata  
3 priedas

# Turinys

<b>1</b>	<b>DUOMENŲ BAZĖS SAMPRATA, TIPAI.....</b>	<b>3</b>
1.1	HIERARCHINIS MODELIS.....	4
1.2	TINKLO MODELIS .....	4
1.3	OBJEKTINIS MODELIS .....	5
1.4	RELIACINIS MODELIS .....	5
<b>2</b>	<b>PAGRINDINĖS RELIACINIO MODELIO SĄVOKOS.....</b>	<b>7</b>
2.1	DUOMENŲ VIENTISUMO SĄLYGOS.....	8
2.2	DUOMENŲ ANOMALIJOS.....	9
2.3	PIRMOJI NORMALINĖ FORMA.....	10
2.4	FUNKCINIAI SĄRYŠIAI .....	11
2.5	ANTROJI NORMALINĖ FORMA.....	12
2.6	TREČIOJI NORMALINĖ FORMA .....	13
2.7	BOISO-KODO NORMALINĖ FORMA .....	14
<b>3</b>	<b>DUOMENŲ BAZIŲ VALDYMO SISTEMOS, JŲ FUNKCIJOS IR SUDĖTIS .....</b>	<b>15</b>
<b>4</b>	<b>DUOMENŲ GAVYBA (DATA MINING) .....</b>	<b>17</b>
4.1	SUDĖTINGESNĖ DUOMENŲ ANALIZĖ.....	18
4.2	PJŪVIAI IR SKAIČIUOJAMOSIOS REIKŠMĖS .....	19

## 9 Duomenų bazės samprata, tipai

Daugelį žmonijos gyvenimo ir veiklos aspektų atspindintys duomenys turi tam tikrą bendrumą ir tam tikrą natūralią tvarką bei sąsajas. Tokie duomenys yra grupuojami, susiejami ir saugomi duomenų bazėse. Todėl galima apibrėžti, kad **duomenų bazė** - tai tarpusavyje susietų struktūrizuotų duomenų visuma, skirta tam tikrai tematikai ar tikslams. Duomenų bazės pavyzdžiu gali būti universitete kaupiama informacija apie studentus, kur kiekvieno studento anketoje yra bilieta nr., pavardė, vardas, gimimo metai, fakultetas ir t.t. Kiekvienas anketos elementas duomenų bazėje yra vadinamas **lauku**, o anketa apie vieną studentą vadinama **įrašu**.



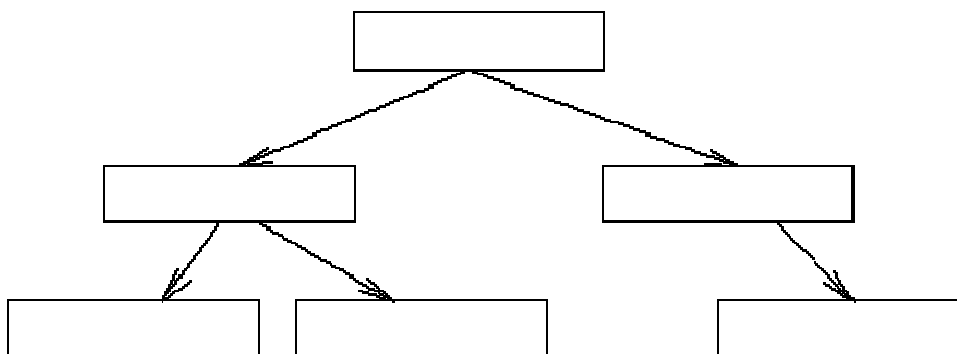
Su kaupiamais duomenimis gali būti atliekami įvairūs veiksmai, tokie kaip papildymas, atnaujinimas, pašalinimas, atrinkimas, rūšiavimas, apibendrinimas; pagal juos gali būti kuriami įvairūs dokumentai, atliekami skaičiavimai.

Dažnai realiuose procesuose duomenys yra susiję vieni su kitais. Šie ryšiai turi atsispindėti ir duomenų bazėje. Tai atliekama nustatant ryšius tarp atskirų įrašų. Ryšiai tarp duomenų nustatomi ir tuo atveju, kai kaupiami dideli informacijos kiekiai skaidomi į poaibius, atspindinčius skirtingus objektų ar reiškinių aspektus.

Didelė duomenų apimtis, sudėtingi ryšiai bei keliami uždaviniai nulemia sudėtingą duomenų bazės loginę struktūrą. Projektuojant duomenų bazes naudojami keturi pagrindiniai duomenų bazės loginės struktūros modeliai:

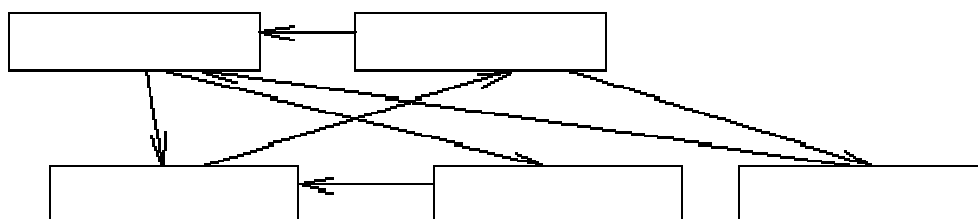
## 9.1 Hierarchinis modelis

**Hierarchinis modelis** susieja griežtai struktūrizuotus duomenis, suformuojant "apversto medžio" struktūrą. Kiekvienas įrašas sudarytas iš dviejų elementų: vieno pagrindinio lauko, vadinamo raktu, kuris vienareikšmiškai identifikuoja įrašą, ir aibės jam pavaldžių laukų, kurie apibūdina likusius įrašo duomenis. Hierarchine struktūra yra charakterizuojama **vienas-su-daugeliu** ryšių tarp duomenų, t.y. kiekvienas įrašas gali turėti daug ryšių su įrašais, esančiais žemesniame hierarchiniame lygyje, ir tik vieną ryšį su aukštesniame lygyje esančiu įrašu. Toks modelis lengvai pritaikomas, nes hierarchiniai ryšiai būdingi tiek organizacijoms, tiek įvairiems procesams.



## 9.2 Tinklo modelis

**Tinklo modelis** sukuria ryšius tarp nepriklausomų duomenų. Kiekvienas įrašas gali turėti daug griežtai nustatytų ryšių su kitais bazės įrašais. Tinklo bazės modelyje realizuojamas ryšys **daugelis-su-daugeliu**. Toks modelis yra artimas realiems biznio ryšiams, kur pardavėjai turi daug pirkėjų, o pirkėjai - daug pardavėjų.



### 9.3 Objektinis modelis

- **Objektinis modelis** - tai naujausias duomenų bazių modelis, kuriame pritaikoma objektinio programavimo samprata ir sąvokos. Čia:
  - objektas - tai asmens, vietos, daikto ar reiškinių apibūdinimas (įrašo ekvivalentas)
  - atributai - objekto savybės duotu laiko momentu (lauko ekvivalentas)
  - metodas - veiksmai, kuriuos turi atlikti objektas
  - pranešimai - informacija, perduodama tarp objektų

Kiekvienas objektas priklauso kuriai nors klasei, kuri apibrėžia objekto struktūrą, savybes ir veiksmus. Klasės sudaro medžio tipo struktūrą, kur žemesniame hierarchijos lygyje esančios klasės paveldi aukštesniame lygyje esančios klasės apibrėžtas savybes ir atributus, bei turi savus, tik joms būdingus parametrus. Šis duomenų bazių modelis taikomas multimedijos sistemose, kadangi leidžia apjungti į vieną objektą vaizdą, garsą, duomenis ir valdančią informaciją. Be to, jis taikytinas greitai besikeičiančiose sistemose, nes nauja objekto versija gali būti lengvai įdiegta objektinėje duomenų bazėje.

### 9.4 Reliacinis modelis

**Reliacinis modelis** paremtas paprasta lentelės koncepcija, kur kiekviena eilutė yra ekvivalentiška įrašui, o kiekvienas stulpelis - laukui. Paprastai duomenų bazė yra realizuojama kaip susietų lentelių aibė. Kiekviena lentelė turi turėti unikalų identifikatorių - sąsajos pavadinimą, o kiekvienas įrašas turi turėti lauką arba laukų grupę, kuri yra pirminiu raktu (primary key), vienareikšmiškai nusakančiu įrašą. Reliacinėje bazėje galima paprastai susieti įrašus ar lenteles iš anksto griežtai neapibrėžtu būdu. Toks duomenų modelis gali būti naudojamas daugelio taikomųjų programų. Dėl jo lankstumo ir taikymo paprastumo tai dažniausiai šiuo metu naudojamas modelis.

**Reliacinis duomenų modelis** apdoroja ir pateikia duomenis lentelių (arba reliacijų) pavidale.

**Reliacija** – tai terminas, atėjęs iš matematikos ir reiškiantis dvimatę lentelę, susidedančią iš duomenų eilučių ir stulpelių.

Kiekvienas reliacijos stulpelis – tai reliacijos **atributas**. Stulpelio pavadinimas – **atributo vardas**. Toliau naudosime terminus atributas ir atributo vardas vietoj terminų stulpelis ir stulpelio

pavadinimas. Reliacijos atributų skaičius vadinamas *reliacijos laipsniu*. Reliacijos eilutės vadinamos *kortežais*. Laikoma, kad nėra nustatytos eilučių (kortežų) išdėstymo tvarkos, todėl jokie du kortežai negali turėti vienodų reikšmių rinkinių.

Rinkinys visų galimų reikšmių, kurias gali turėti atributai, vadinamas atributo sritimi. Dvi atributų sritys sutampa tik tuo atveju, jeigu jos turi tas pačias reikšmes.



## 10 Pagrindinės reliacinio modelio sąvokos

**Reliaciniame modelyje** duomenys pateikiami dvimatėmis lentelėmis, susidedančiomis iš eilučių ir stulpelių. Lentelės stulpelis reliaciniame modelyje dažnai vadinamas **atributu (lauku)**. Stulpelis turi pavadinimą - tai atributo vardas. Reliacinėje teorijoje laikomasi nuostatos, kad tiek stulpelių, tiek ir eilučių tvarka lentelėje yra neapibrėžta. Visų leistinų atributo reikšmių aibė vadinama **domenu**. **Tuščioji reikšmė** (žymima *NULL*) - tai reikšmė, kuri priskiriama atributui eilutėje, kai atributo reikšmė nežinoma arba atributas yra neprasmingas.

Minimalus atributų rinkinys (aibė), vienareikšmiškai apibrėžiantis (nusakantis) kiekvieną lentelės eilutę, vadinamas **raktu**. Raktą taip pat galima apibrėžti kaip minimalią atributų aibę, vienareikšmiškai apibrėžiančią (kitaip dar sakoma funkciškai apibrėžiančią) kiekvieno atributo reikšmę eilutėje. Kitaip tariant, **lentelės L raktu** vadinamas toks jos atributų aibės poaibis K, kad:

rakto atributų reikšmės vienareikšmiškai apibrėžia visų lentelės L atributų reikšmes, t.y. dviejose skirtingose lentelės eilutėse negali sutapti visos aibės K atributų reikšmės (vienareikšmiška identifikacija);

joks aibės K poaibis neturi vienareikšmiškos identifikacijos savybės (pertekliaus nebuvimas), t.y. joks aibės K poaibis negali vienareikšmiškai apibrėžti kiekvienos lentelės eilutės.

Lentelėje gali būti keli atributų rinkiniai, turintys rakto savybę. Visi jie vadinami **kandidatiniais raktais**, arba tiesiog raktais. Raktas, kurį sudaro du ar daugiau atributų vadinamas **sudėtiniu raktu**.

Paprastai vienas iš raktų paskelbiamas **pirminiu raktu**. Jei lentelėje yra keli raktai, tai pirminiu parenkamas tas, kuriuo paprasčiausia naudotis, pavyzdžiui, trumpiausias. Dažniausiai, raktu vadinsime pirminį raktą, kuris yra ir vienintelis galimas lentelės pirminis raktas.

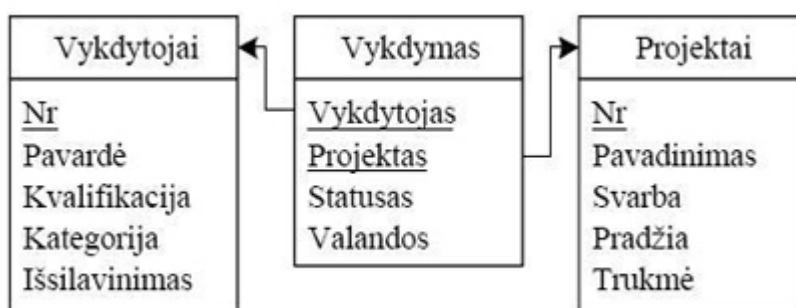
**Lentelės reliacinė schema** (struktūra, aprašas) - tai lentelės ir visų jos stulpelių pavadinimai, su pažymėtu (išskirtu, pvz., pabrauktu) pirminiu raktu.

**Išoriniu raktu** vadinamas vienos lentelės atributų rinkinys, kuris kitoje lentelėje (ar net toje pačioje) yra pirminis raktas. Išorinio rakto atributų pavadinimai nebūtinai turi sutapti su pirminio rakto atributų pavadinimais. Kadangi išorinis raktas paprastai siejamas su lentelės pirminiu raktu, tai jam apibrėžti visiškai pakanka nurodyti išorinį raktą sudarančius atributus ir pavadinimą lentelės, į kurios pirminį raktą išorinis raktas rodo. Išorinis raktas vartojamas loginiams ryšiams tarp lentelių apibrėžti. Informacija apie išorinius raktus yra svarbi DB loginei struktūrai. Todėl **duomenų bazės**

**reliacine schema** vadinsime visų jos lentelių reliacinių schemų rinkinį kartu su lentelių išoriniais raktais.

DB reliacinę schema galima pavaizduoti ir grafiškai. Grafiškai pavaizduota DB reliacine schema tampa aiškesnė, ypač kai duomenų bazę sudaro daug išoriniais raktais tarpusavyje susijusių lentelių. Kiekvienos lentelės schema vaizduojama stačiakampiu, kurio viršuje užrašomas lentelės pavadinimas. Žemiau išvardijami visi lentelės atributai (stulpelių pavadinimai). Pirminį raktą sudarantys atributai pabraukiami. Pagrindinis grafinio vaizdavimo privalumas - galimybė grafiškai pavaizduoti ryšius tarp lentelių. Lentelės išoriniai raktai vaizduojami rodykle iš išorinį raktą sudarančių atributų į lentelės, į kurią išorinis raktas nurodo, pavadinimą.

DB Darbai reliacine schema galima pavaizduoti taip:



### 10.1 Duomenų vientisumo sąlygos

Reliacinėje teorijoje yra keletas reikalavimų, kuriuos turi atitikti duomenų bazės duomenys. Viena reikalavimų rūšis vadinama **duomenų vientisumo sąlygomis**:

- § kategorijų vientisumas;
- § nuorodų vientisumas;
- § funkciniai sąryšiai.

Reliacinės lentelės eilutės yra konkrečių realaus pasaulio objektų atvaizdai duomenų bazėje. Realaus pasaulio objektai, kurie vaizduojami lentelės eilutėmis, reliacinėje teorijoje vadinami **kategorijomis**. Lentelės raktas vienareikšmiškai nusako lentelės eilutę, tuo pačiu ir kategoriją. Norėdamas rasti duomenis apie konkrečią kategoriją (objektą), vartotojas privalo žinoti rakto reikšmes. Tai reiškia, kad kategorija neturi prasmės duomenų bazėje, jei bent vieno rakto atributo reikšmė yra nežinoma. Todėl pagal kategorijų vientisumo reikalavimą joks lentelės rakto atributas nė vienoje eilutėje negali turėti NULL reikšmės.

Jau minėjome, kad reliacinėje duomenų bazėje vienos lentelės eilučių ryšiui su kitos lentelės eilutėmis vartojami išoriniai raktai. Duomenų bazė, kurioje visi ne tušti išoriniai raktai nurodo į egzistuojančią pirminio rakto reikšmę, tenkina nuorodų vientisumo reikalavimą. Pastebėsime, kad lentelėje išorinis raktas gali būti tuščias (jo atributų reikšmės yra NULL), jei tik jo atributai neįeina į pirminį raktą. **Nuorodų vientisumo reikalavimas** gali būti suformuluotas taip: kiekvieno išorinio rakto reikšmė duomenų bazėje turi būti arba tuščia, arba sutapti su viena pirminio rakto reikšme lentelėje, į kurią išorinis raktas nurodo.

## 10.2 Duomenų anomalijos

Tarkime, turime lentelę Projektai, išorinis raktas: Vykdytojas nurodo į Vykdytojai:

### Projektai

Projektas	Pavadinimas	Svarba	Trukmė	Vykdytojas
1	Studentų apskaita	Maža	12	1
1	Studentų apskaita	Maža	12	2
1	Studentų apskaita	Maža	12	3
1	Studentų apskaita	Maža	12	4
2	Buhalterinė apskaita	Vidutinė	10	1
2	Buhalterinė apskaita	Vidutinė	10	2
2	Buhalterinė apskaita	Vidutinė	10	4
3	WWW svetainė	Didelė	6	1
3	WWW svetainė	Didelė	6	2
3	WWW svetainė	Didelė	6	3

Nesunku pastebėti, kad lentelė Projektai sudaryta nesėkmingai. Pavyzdžiui, keturiose eilutėse, atitinkančiose projektą Nr. 1, kartojasi tas pats projekto pavadinimas, jo svarba ir trukmė. Tokie **pertekliniai duomenys** ne tik užima papildomą vietą kompiuterio atmintyje - dėl jų gali būti prarastas duomenų vientisumas, arba, kitaip tariant, atsirasti duomenų prieštaravimas. Mat tada vieną projektą gali vykdyti keli vykdytojai. Tarkime, projekto Nr. 1 trukmė pasikeitė, ir naujoji trukmė buvo pakeista tik vienoje eilutėje. Tuomet tarp eilučių, kuriose yra informacija apie tą patį projektą Nr. 1, atsiranda neatitikimas, kuris vadinamas atnaujinimo anomalija. **Atnaujinimo anomalija** - tai duomenų prieštaravimas, atsirandantis dėl duomenų pertekliaus, atnaujinus tik dalį jų.

Tarkime, vykdytojai, kurių numeriai yra 1, 2 ir 3 išeina iš darbo. Tada iš lentelės pašalinamos visos eilutės, atitinkančios šiuos vykdytojus. Todėl iš lentelės Projektai, o kartu ir iš duomenų bazės, pašalinami visi duomenys apie projektą Nr. 3, nes jį vykdė tik tie iš darbo išėję darbuotojai. Tai nėra gerai, nes projektą vis tiek reikės tęsti, paskyrus dirbti kitus darbuotojus. Tai **pašalinimo anomalija** - nenumatytas duomenų praradimas, susijęs su kitų duomenų pašalinimu.

Kita vertus, firma gali gauti užsakymą naujam projektui vykdyti, kai dar nėra paskirtas nė vienas projekto vykdytojas. Kadangi vykdytojo numeris, tiksliau, atributas Vykdytojas, yra lentelės Projektai raktas, tai, kol nebus paskirtas bent vienas vykdytojas, mes negalėsime įvesti į duomenų bazę duomenų apie naują projektą. Tai *įvedimo anomalija* - nebuvimas galimybės įvesti duomenis dėl to, kad trūksta kitų duomenų.

Duomenų bazė turi būti sudaryta taip, kad atnaujinimo, pašalinimo ir įvedimo anomalijų nebūtų. Todėl duomenų bazėje Darbai vietoj lentelės Projektai\_Vykdymas turi būti dvi lentelės Projektai ir Vykdymas, kaip tai buvo padaryta anksčiau. Tačiau tai tik intuityvus problemos sprendimas, kuris, esant duomenų bazėje daug lentelių ir atributų, gali nebūti toks akivaizdus. Tolimesniuose skyreliuose mes apibrėšime formalesnį metodą, vadinamą lentelių skaidymu, kuris duos tokį patį rezultatą. *Lentelių skaidymu* vadinamas lentelės padalijimas į kelias lenteles, siekiant išvengti anomalijų ir neprarasti duomenų vientisumo. Tam naudojamos normalinės formos ir lentelių struktūrizavimo taisyklės.

### 10.3 Pirmoji normalinė forma

*Normalinė forma* (sutrumpintai, NF) vadinami apribojimai duomenų bazės reliacinei schemai, kuriuos taikant DB išvengia tam tikrų nepageidaujamų savybių.

Imkime anksčiau įvestą lentelę Projektai. Nustatėme, kad šiai lentelei būdingos duomenų anomalijos. Anomalijų priežastis yra ta, kad vieną projektą gali vykdyti keli vykdytojai, ir dėl to lentelėje Projektai atsiranda duomenų perteklius. Šią lentelę pertvarkykime:

#### *Projektai*

Projektas	Pavadinimas	Svarba	Trukmė	Vykdytojas
1	Studentų apskaita	Maža	12	{1,2,3,4}
2	Buhalterinė apskaita	Vidutinė	10	{1,2,4}
3	WWW svetainė	Didelė	6	{1,2,3}

Pastarojoje lentelėje atributo Vykdytojai reikšmės yra visų projekto vykdytojų numerių aibė.

Iš pirmo žvilgsnio gali pasirodyti, kad ši lentelė sudaryta gerai, nes joje nėra duomenų pertekliaus. Visų pirma, šiai lentelei negalima apibrėžti išorinio rakto, nes lentelės atributas Vykdytojai negali būti susietas su lentelės Vykdytojai pirminiu raktu, kaip buvo anksčiau, nors ryšys tarp jų yra. Nesant išorinio rakto, stulpelyje Vykdytojai gali atsirasti numeriai, neatitinkantys nė vieno vykdytojo, užregistruoto lentelėje Vykdytojai.

Lentelė yra *pirmos normalinės formos (INF)*, jei visų jos atributų reikšmės yra nedalomos (nėra nei aibė, nei sąrašas).

Lentelė Projektai nėra pirmos normalinės formos, nes jos stulpelyje Vykdytojai yra aibės, kurių elementai kitoje lentelėje yra atributo reikšmės. Ankstesnė lentelė Projektai atitinka 1NF, tačiau jau anksčiau padarėme išvadą, kad ji nėra gerai sudaryta, nors ji ir yra INF. Taigi reikalavimas lentelei atitikti 1NF yra per silpnas, kad lentelės schema būtų galima vadinti pakankamai gera. Todėl nagrinėjamos ir aukštesnės eilės normalinės formos. Kuo aukštesnė NF, tuo aukštesni reikalavimai lentelės schemai. Taigi kuo aukštesnės NF reikalavimus tenkina lentelės schema, tuo mažiau blogų savybių ji turi. Duomenų bazė yra tam tikros NF, jei visos jos lentelės yra tos NF.

#### 10.4 Funkciniai sąryšiai

Jau aptarėme kategorijų ir nuorodų vientisumo sąlygas. Dar griežtesnius reikalavimus reliacinei schemai apibrėžia funkciniai sąryšiai. Tai bene svarbiausias apribojimas. Jo esmė -vienų atributų reikšmės eilutėje gali vienareikšmiškai apibrėžti kitų atributų reikšmes. Pavyzdžiui, kiekvienoje anksčiau nagrinėtos lentelės Projektai eilutėje atributas Projektas vienareikšmiškai apibrėžia atributų Pavadinimas, Svarba, Trukmė ir Pradžia reikšmes.

Tokiu būdu, jei atributų aibės A reikšmės eilutėje vienareikšmiškai apibrėžia atributų aibės B reikšmes eilutėje, tai sąryšį tarp A ir B vadiname funkciniu. Formaliau, **funkcinis sąryšis** (f-sąryšis) gali būti apibrėžtas taip: jei A ir B yra lentelės L atributų aibės, tai užrašas  $A \rightarrow B$  reiškia, jog jei dvi lentelės L eilutės turi vienodas atributų A reikšmes, tai atributų B reikšmės taip pat sutampa. Užrašą  $A \rightarrow B$  galima skaityti dvejopai: B **funkciškai priklauso** nuo A, taip pat A **funkciškai apibrėžia** B. F-sąryšio kairiosios dalies atributai vadinami **determinantu**. **Determinantas** - tai atributai, kurių reikšmės apibrėžia kitų atributų reikšmes.

Jau minėti lentelės Projektai \_Vykdymas atributų tarpusavio sąryšiai yra f-sąryšių pavyzdžiai, kuriuos galima užrašyti taip:

*Projektas*  $\rightarrow$  *Pavadinimas*,

*Projektas*  $\rightarrow$  *Svarba*,

*Projektas*  $\rightarrow$  *Trukmė*,

*Projektas*  $\rightarrow$  *Pradžia*.

Šių keturių funkcinių sąryšių užrašą galima pakeisti vienu trumpesniu:

*Projektas*  $\rightarrow$  {*Pavadinimas*, *Svarba*, *Trukmė*, *Pradžia*}.

Be šių f-sąryšių, lentelėje Projektai\_Vykdymas galioja dar ir toks:

{*Projektas*, *Vykdytojas*}  $\rightarrow$  {*Statusas*, *Valandos*}.

Atkreipsime dėmesį į tai, kad f-sąryšis, kaip ir kiti apribojimai duomenims, galioja reliacinei schemai, o ne apie konkrečius lentelės duomenims. Pagal einamąjį lentelės turinį neįmanoma nustatyti, ar koks nors f-sąryšis teisingas (būdingas) schemai. Pavyzdžiui, iš turimų duomenų lentelėje Projektai\_Vykdymas būtų galima daryti išvadą, kad šios lentelės schemai būdingas f-sąryšis: Svarba  $\rightarrow$  Trukmė, kadangi jis tinka visoms lentelės eilutėms. Tačiau taip tvirtinti negalima, nes kitu metu gali būti vykdomi du (ar daugiau) projektai, kurių svarba vienoda, tačiau trukmė skirtinga. Taigi kai sakoma, kad lentelei būdingas konkretus f-sąryšis, tai turima omenyje, kad tas sąryšis galioja bet kuriuo momentu.

### 10.5 Antroji normalinė forma

Atributų aibė B yra *visiškai priklausoma* nuo atributų aibės A sąryšių aibės F atžvilgiu, jei B - funkciškai priklausoma nuo visos aibės A, bet nėra funkciškai priklausoma nei nuo jokio A poaibio.

Lentelės L(R) atributas A vadinamas *pirminiu* lentelės L atributu sąryšių aibės F atžvilgiu, jei a priklauso kuriam nors lentelės L raktui, priešingu atveju a vadinamas *nepirminiu atributu*.

Lentelė L(R) yra *antros normalinės formos (2NF)* sąryšių aibės F atžvilgiu, jei ji yra 1NF ir kiekvienas jos nepirminis atributas visiškai priklauso nuo kiekvieno lentelės L rakto. Duomenų bazė yra 2NF sąryšių aibės F atžvilgiu tada ir tik tada, kai visos jos lentelės yra 2NF atžvilgiu F.

Jau anksčiau pastebėjome, kad šiai lentelei būdinga pertekliniai duomenys bei duomenų įvedimo, atnaujinimo ir šalinimo anomalijos. Šių nepageidautinų savybių nelieka išskaidžius lentelę į dvi lenteles Projektai ir Vykdymas. Abi lentelės Projektai ir Vykdymas yra 2NF. Tokiu būdu, 2NF sumažina duomenų perteklių ir pašalina duomenų anomalijas. Lentelės, neatitinkančios 2NF, skaidymo procesas į kelias lenteles, atitinkančias 2NF, susideda iš kelių nesudėtingų žingsnių:

- § sukurama nauja lentelė, kurios atributai yra pradinės lentelės atributai, įeinantys į f-sąryšį tarp nepirminių atributų (atributo) ir rakto dalies. Šio f-sąryšio determinantas tampa naujos lentelės raktu;
- § atributai, esantys dešinėje f-sąryšio pusėje, pašalinami iš pradinės lentelės;
- § jei pradinė lentelė nėra 2NF dėl daugiau nei vieno f-sąryšio, tai žingsniai 1 ir 2 kartojami kiekvienam tokiam sąryšiui.

Gali pasirodyti, kad naujose lentelėse galiojantys sąryšiai skiriasi nuo pradinėje lentelėje galiojusių. Tačiau taip nėra. Toks lentelių skaidymas, kai informacija neprarandama, vadinamas *dekompozicija be praradimo*. Suskaidžius pradinę lentelę L į dvi lenteles L1 ir L2, lentelės L2 raktas A gali būti pirminiu, o lentelėje L1 galima apibrėžti išorinį raktą, susiejant jos atributai su L2 pirminiu raktu.

Lentelė, kurią sudaro tik kai kurie kitos lentelės stulpeliai, vadinama *projekcija*. Taigi ir lentelė L1 ir lentelė L2 yra lentelės L projekcijos.

### Projektai

Projektas	Pavadinimas	Svarba	Trukmė
1	Studentų apskaita	Maža	12
2	Buhalterinė apskaita	Vidutinė	10
3	WWW svetainė	Didelė	6

### Vykdymas

Projektas	Vykdytojas
1	1
1	2
1	3
1	4
2	1
2	2
2	4
3	1
3	2
3	3

## 10.6 Trečioji normalinė forma

Lentelės L(R) atributų aibės R poaibis C ( $C < R$ ) *tranzityviai priklauso* nuo atributų aibės R sąryšių aibės F atžvilgiu, jei egzistuoja toks atributų aibės R poaibis B, kad A funkciškai apibrėžia B, bet ne atvirkščiai, bei B funkciškai apibrėžia C atžvilgiu F.

Lentelė L(R) yra *trečios normalinės formos (3NF)* sąryšių aibės F atžvilgiu, jei ji yra 2NF ir nėra nepirminių atributų, tranzityviai priklausančių nuo rakto. DB yra 3NF sąryšių aibės F atžvilgiu tada ir tik tada, kai visos jos lentelės yra 3NF F atžvilgiu.

Tarkime, lentelėje Vykdytojai reikia saugoti pabaigtos aukštosios mokyklos adresą.

### Vykdytojai

Nr	Pavardė	Išsilavinimas	UAdresas
1	Jonaitis	VU	Universiteto 3, Vilnius
2	Petraitis	VU	Universiteto 3, Vilnius
3	Antanaitis	VDU	Donelaičio 58, Kaunas
4	Onaitytė	VDU	Donelaičio 58, Kaunas

Stulpelis Nr yra lentelės vienintelis raktas, taigi turime priklausomybes:

$$Nr \rightarrow \{Pavardė, Išsilavinimas, UAdresas\}$$

Tačiau atriutas UAdresas funkciškai priklauso ir nuo atributo Išsilavinimas.

$$Išsilavinimas \rightarrow \{UAdresas\}$$

Ši lentelė atitinka 2NF, nes raktą sudaro tik vienas atributas, todėl nėra atributų, priklausančių nuo rakto dalies bei nėra atributų reikšmių, sudarytų iš sąrašų ar aibių. Tačiau šiai lentelei būdingas duomenų perteklius bei anomalijos.

Kadangi lentelėje Vykdytojai galioja sąryšis  $Nr \rightarrow Išsilavinimas$  ir negalioja sąryšis  $Išsilavinimas \rightarrow Nr$ , be to Išsilavinimas funkciškai apibrėžia atributą UAdresas, todėl gauname, kad UAdresas tranzityviai priklauso nuo lentelės rakto Nr. Todėl ši lentelė nėra 3NF. Kad reliacinė schema atitiktų 3NF, šią lentelę reikia skaidyti į dvi lenteles Vykdytojai ir LavavimoIstaigos.

### 10.7 Boiso-Kodo normalinė forma

3NF, kaip ir 1NF bei 2NF, pirmasis apibrėžė Kodas (E.F.Codd). Kurį laiką buvo skaitoma, kad 3NF yra praktiškai pakankama. Vėliau paaiškėjo, kad 3NF neviseškai tinka lentelėms, kurios turi du ar daugiau raktų, bent du iš jų yra sudėtiniai (sudaryti iš keleto atributų) raktai ir jie tarpusavyje kertasi (turi bent vieną bendrą atributą). Todėl tas 3NF apibrėžimas buvo pakeistas griežtesniu Boiso-Kodo (Boyce-Codd) apibrėžimu. Lentelė yra **Boiso-Kodo normalinės formos (BKNF)**, jei kiekvieno netrivialaus f-sąryšio determinantas yra raktas. Gana nesunku įsitikinti, kad lentelė, esanti BKNF, yra ir 3NF. BKNF įgavo tokį pavadinimą dėl to, kad jos suformulavimo metu jau buvo suformuluota 4NF. DB yra BKNF sąryšių aibės F atžvilgiu tada ir tik tada, kai visos jos lentelės yra BKNF F atžvilgiu.

Realiuose uždaviniuose, jei duomenų bazė yra 3NF ar BKNF, sakoma, kad to pakanka. Praktiškai dar taikoma 4NF. Reliacinėje teorijoje nagrinėjamos ir aukštesnės normalinės formos. Tačiau aukštesnės normalinės formos svarbios daugiau teoriniu, negu praktiniu požiūriu.



## 11 Duomenų bazių valdymo sistemos, jų funkcijos ir sudėtis

Duomenų bazės valdymo sistema (DBVS) vadinama programine įranga, skirta DB kurti, jas saugoti ir įvairiais būdais apdoroti. Svarbiausios DBVS funkcijos yra šios:

- DB struktūros projektavimas;
- DB pildymas, kaupimas, redagavimas;
- navigacija DB;
- duomenų peržiūra, paieška, rikiavimas ir kitas tvarkymas;
- taikomųjų vartotojo programų kūrimas;
- ataskaitų kūrimas.

Bet kurią DBVS sudaro priemonės DB struktūrai projektuoti, duomenims įvesti, papildyti ir modifikuoti. Šiuolaikinė DBVS suteikia jos vartotojui lanksčias galimybes lengvai surasti ir atrinkti reikalingus duomenis pagal vieną ar kelis kriterijus, tuos duomenis rikiuoti, grupuoti, vaizduoti juos pageidaujama forma. Tam tikrais laiko tarpais reikia sukauptą informaciją apibendrinti, atlikti analizę, vykdyti matematinius, statistinius skaičiavimus ir kurti įvairių formatų ataskaitas. Vartotojai, esant reikalui, rašo bei derina savas taikomas programas, naudodamiesi tam skirtomis DBVS priemonėmis. DBVS turi atlikti ir daug kitų svarbių funkcijų: duomenų apsaugą, jų korektiškumo, neprieštaringumo ir išsamumo kontrolę, DB kopijų išsaugojimą ir kt.

Daugelį šiuolaikinių DBVS sudaro tokios dalys:

- duomenų tvarkymo dialoginė aplinka;
- duomenų aprašymo ir manipuliavimo jais kalba;
- programų generatoriai.

Dialoginė aplinka yra viena svarbiausių šiuolaikinių DBVS dalių. Ji skirta interaktyviam darbui su DBVS atliekant veiksmus su visa DB ar atskirais įrašais. Šiuolaikinės dialoginės aplinkos suteikia vartotojui galimybę pasirinkti jam priimtinausią darbo su DB stilių. Yra du pagrindiniai darbo su DB būdai:

- grafinių, vizualiai orientuotų instrumentų panaudojimas;
- specialių instrukcijų rašymas.

Grafiniai instrumentai pagal savo veikimo principą skirstomi į dizainerius, meistrus (žynius-suflerius) ir meniu sistemas. Dizaineriai suteikia galimybę vartotojui pačiam laisvai pasirinkti, ką daryti ir kaip daryti. Meistrai ir meniu sistemos leidžia tik išsirinkti, ką daryti, t. y. pateikiamas galimų veiksmų sąrašas (menu), o vartotojui telieka patvirtinti savo pasirinkimą iš daugelio

siūlomų veiksmų sekų ar maršrutų. Be minėtų instrumentų dažnai naudojami konstruktoriai, padedantys rašyti sudėtingas išraiškas, formules ir pan.

Kita svarbi DBVS dalis yra duomenų aprašymo ir manipuliavimo jais kalba. Ji skirta kurti taikomas programas, t. y. tokias programas, kurias vartotojas ar jų grupė naudoja konkrečioms, specifiniams uždaviniams spręsti. Į tokios kalbos sudėtį įeina duomenų aprašymo ir manipuliavimo jais instrukcijos (komandos). Aprašant duomenis nurodoma DB lentelių struktūra, raktiniai laukai, ryšiai tarp atskirų lentelių. Manipuliavimo duomenimis instrukcijos - tai daugybė įvairios paskirties komandų, skirtų tiesiogiai apdoroti duomenų elementus (laukų reikšmes), lentelių įrašus, lenteles ar visą DB.

Iš daugybės manipuliavimo duomenimis instrukcijų į savarankišką grupę išskiriamos vadinamosios užklausos. Jos aprašomos ir vykdomos atitinkama kalba. Viena plačiausiai naudojamų užklausų kalbų yra SQL (Structured Query Language - struktūrizuota užklausų kalba). Tai gana paprasta kalba su aiškiais standartizuotomis instrukcijomis. Ši kalba įtraukta į daugumą DBVS. Plačiau SQL nagrinėjama tolesniuose skyriuose.

Modernios DBVS turi programų generatorius. Tai specialūs instrumentai, skirti automatizuoti programų rašymą, kartu palengvinti vartotojų, ypač programuotojų, darbą kuriant taikomas programas. Generatoriai įgalina žymiai sumažinti darbo sąnaudas, kai reikia programuoti daug laiko jų rašymui reikalaujančias ir dažnai pasikartojančias operacijas, pvz., ekrano apiforminimo, veiksmų (menu) juostų, ataskaitų sudarymo ir kt. operacijas.

Yra daugybė DBVS. Jas kuria įvairios firmos, kurios šias sistemas platina tarp vartotojų kaip atskirus, savarankiškus programinės įrangos vienetus. Tokie vienetai vadinami paketais. DB valdymo paketai skiriasi vieni nuo kitų kokybinėmis ir kiekybinėmis (techninėmis) charakteristikomis. Kokybinės DBVS charakteristikos - tai apimtis (sudėtingumo laipsnis), taikymo sritis, funkcionavimo bazė, darbo patogumas. Kiekybinės charakteristikos - tai, pvz., leistina apdorojamos DB apimtis, DB lentelių skaičius, lentelės apimtis ir pan. Vienas iš pagrindinių sistemų klasifikavimo kriterijų yra sistemų apimtis. Pagal savo apimtį DBVS galima suskirstyti į dideles (labai sudėtingas) sistemas, vidutines (mažiau sudėtingas) sistemas ir mažas sistemas. Didelės DBVS yra šios: Oracle, Sybase, Informix, DB2, SQL Server, IMS, Ingres. Vidutinių sistemų yra daugiau. Pagrindinės yra šios: Foxpro, Access, Paradox, Clipper, Clarion, dBase ir kt. Dar daugiau yra mažų DBVS - jų šiuo metu suskaičiuojama daugiau kaip 50.

Daugumos šiuolaikinių DBVS funkcionavimo bazė yra personaliniai kompiuteriai. Visos didelės sistemos, be to, dar gali funkcionuoti ir minikompiuteriuose bei super-kompiuteriuose. Paprastai

visi DB valdymo paketai yra orientuojami į darbą Windows tipo operacinėse sistemose, būtent, Windows NT, Windows 95 ir pan. Dideli paketai gali veikti ir operacinėse sistemose, skirtose superkompiuteriams, pvz., Unix, VAX VMS, OS/2.

Pagrindinės kiekybinės DBVS charakteristikos yra šios:

- maksimalus leistinas lentelių kiekis DB;
- maksimalus lentelės dydis;
- maksimalus įrašų kiekis lentelėje;
- maksimalus simbolių kiekis įrašė (lentelės plotis);
- maksimalus laukų kiekis lentelėje;
- maksimalus lauko plotis;
- maksimalus lauko vardo ilgis ir kt.

## 12 Duomenų gavyba (Data Mining)

Duomenų kaupimas jų neapdorojus neturi prasmės, todėl analizė buvo visais laikais. Visų rūšių duomenų tvarkymas (kartotekos, rūšiavimas ir t.t.) yra pirmas žingsnis norint palengvinti duomenų analizę, apdoroti didelį informacijos srautą. Viena iš informacijos analizės sričių — duomenų gavyba (Data Mining).

*Duomenų gavyba yra prasmingų dėsningumų, modelių ir tendencijų radimo procesas dideliuose informacijos kiekiuose, naudojant modelių atpažinimo, statistinius bei matematinius metodus.*

Jei įprasti duomenų analizės metodai parodo norimų kintamųjų priklausomumą, tai duomenų gavyba unikali tuo, kad analizės rezultatas yra naujų priklausomybių, apie kurių egzistavimą buvo ar net nebuvo įtariama, radimas.

Duomenų gavyba turi vieną savybę, išskiriančią ją iš kitų procesų ar operacijų, susijusių su duomenų bazėmis. Jei įprastos ataskaitos bei įvairios analitinės priemonės atsako į klausimus "kas yra?" ar "kas bus, jeigu?", tai duomenų gavybos rezultatas yra atsakymas į nepateiktus klausimus, t.y. naujų priklausomybių radimas. Tyrimai rodo, kad dažniausiai duomenų gavyba pasiteisina neblogą finansinę padėtį turinčiose kompanijose. Tuomet sudaromos galimybės pagerinti veiklą arba nustatyti nedidelius nesklandumus.

Duomenų gavyba yra labai plati sritis, todėl yra ir daug jai skirtų metodų, algoritmų bei taikomųjų sistemų:

- **Asociacijų paieška** — tai dėsningumų analizė tam tikrose reiškinių ar daiktų grupėse. Pavyzdys — pirkimo analizė. Nagrinėjama, kokios prekės perkamos kartu, kokia tikimybė, kad bus būtent toks derinys ir t.t. Tokio tipo uždaviniai gali būti sprendžiami rengiant reklamos kampanijas, kuriant nuolaidų sistemas.
- **Eiliškumo analizė** — tai dėsningumų paieška atsižvelgiant į laiką. Šiuo atveju svarbu ne tik tai, kokiomis paslaugomis naudojasi klientas, bet ir kokia eilės tvarka. Šis metodas padeda efektyviau teikti paslaugas.
- **Grupavimas (klasterizavimas)** dažnai būna vienas pirmųjų duomenų gavybos žingsnių. Tai visos duomenų aibės suskaidymas į poaibius pagal skiriamuosius bruožus. Tai ir rinkos ar klientų segmentavimas, ir nekilnojamojo turto grupavimas pagal būdingus duomenis, ir daugelis kitų uždavinių.
- **Klasifikavimas** dažnai atliekamas po grupavimo. Kai nagrinėjama aibė jau padalyta į pogrupius, dažnai kyla klausimas, kam priskirti naujus elementus. Grupavimu anksčiau neįvardyti poaibiai išskiriami iš duomenų visumos, o klasifikuojant sprendžiama, kaip sudėti elementus į žinomas grupes.
- Po klasifikavimo atliekamas **įvertinimas**. Pavyzdžiui, finansinė institucija ne tik nori žinoti savo išduotų paskolų apibūdinimą "gera — bloga", bet ir jų įvertinimą.
- **Prognozavimas** taip pat labai svarbus duomenų gavybai. Atsižvelgiant į turimus duomenis bei pastebėtas tendencijas, bandomi prognozuoti ateities įvykiai.

Dažnai uždaviniui išspręsti taikomi keli metodai iš eilės ar net sudėtingi jų deriniai. Uždavinių bei metodų įvairovę papildo grupė duomenų gavybos algoritmų. Nė vienas jų nėra universalus ar nepriekaištingas. Vienų trūkumas — sudėtingumas, kitų — didelė modelių apimtis, daug sugaištama laiko.

### 12.1 Sudėtingesnė duomenų analizė

Matyt jau seniai praėjo laikai, kai kiekvienos kaimo parduotuvėlės savininkas atmintinai žinojo kiekvieno savo kliento vardą, numatė, ką tą dieną šis klientas pirs ir t. t. Šiais laikais didelės bendrovės turi tūkstančius ar net dešimtis tūkstančių klientų.

Duomenų saugyklose saugoma istorinė informacija leis paklausti, pavyzdžiui, kiek tam tikras pardavėjas tame rajone pardavė tam tikrų prekių 2003 metų vasario ir kovo mėnesiais. Į analitiko klausimą: "Kas yra pagrindiniai mūsų klientai tam tikram produktui?" padės atsakyti tik sudėtingesnė duomenų analizė (data mining).

Kadangi duomenų kubuose duomenys jau yra reikiamu būdu struktūrizuoti, OLAP duomenų bazės dažnai leidžia ne tik apibendrinti duomenis pagal norimus parametrus, bet ir gana nesudėtingai atlikti gilesnę jų analizę. Dauguma OLAP produktų siūlo įvairius duomenų analizės įrankius. Pavyzdžiui, "MS SQL Server 2000 Analysis Services" leidžia panaudoti du automatizuotus analizės metodus - sprendimų medžius (Decision Tree) ir klasifikavimą - klasterinę analizę (Clustering), kurie leidžia surasti įdomią ir neakivaizdžią informaciją.

Sprendimų medis vizualiai pateikia informaciją kaip medžio šakų pavidalo struktūrą. Klasifikavimo (klasterizavimo) algoritmas leidžia išskirti duomenų įrašų sutankėjimo grupes įvairiose dimensijose. Pažymėtina, kad grupės nebūtinai identifikuojamos pagal visas dimensijas - gali paaiškėti, kad grupei kai kurių dimensijų reikšmės gali būti visiškai nesvarbios.

## 12.2 Pjūviai ir skaičiuojamosios reikšmės

Duomenų apdorojime naudojama pjūvių sąvoka (dimensions). Duomenų bazėje esantys įvykiai (eilutė lentelėje) gali būti analizuojami pagal įvairius parametrus - įvairiais pjūviais (dimensijomis).

Pamėginsime paaiškinti pjūvių sąvoką naudodami pardavimų modelio pavyzdį. Užregistruotas prekių pardavimo įvykis apibūdinamas keliais parametrais (kada tai įvyko, koks produktas parduotas, kokioje parduotuvėje), todėl ir įmonės mastu pardavimus galima analizuoti įvairiais pjūviais:

- Pagal laiką. Galima sužinoti, kaip prekyba sekėsi šį mėnesį, palyginti su praėjusiu, su praėjusiais metais ir pan.
- Pagal produktą. Parduodamos įvairios produktų grupės, jų tipai ir pagaliau - konkretūs produktai. Galima sužinoti, kokią pajamų dalį sugeneravo ne tik konkretus produktas, bet ir viena ar kita produktų grupė, koks perkamiausias produktas ir pan.
- Pagal vietovę. Didelės organizacijos pardavimus vykdo didelėje teritorijoje, todėl konkrečios parduotuvės gali būti priskirtos prie apskričių, rajonų, rajonuose gali būti numatyti mažesni struktūriniai vienetai, pavyzdžiui, miestai ir pan. Tokiu būdu galima atsakyti, kaip sekėsi parduoti tą ar kitą prekę apskrityje, rajone ar mieste.

Kaip matome iš pavyzdžio, pjūviai gali turėti tam tikrą hierarchinę struktūrą, pavyzdžiui, vietovės ir laiko pjūviai turi net po tris hierarchinius lygius (apskritis > rajonas > miestas ir metai > ketvirtis > mėnuo). Tokia hierarchinė struktūra gali būti ir labai sudėtinga. Šiuo atveju sukūrėme trimatį duomenų kubą, kurio matmenis apibrėžia laiko, produkto ir vietovės dimensijos. Tačiau loginė duomenų kubo struktūra gali būti ne tik trimatė, bet ir daugiamatė (keturmatė, penkiamatė ir t. t.).

Kita pagrindinė duomenų bazių sąvoka yra skaičiuojamoji vertė (measure). Galima pasirinkti, kokią su konkrečiu įvykiu susijusią skaitinę reikšmę analizuosime. Pateikiamame pavyzdyje galėtume analizuoti ne tik pajamas iš šiame mieste per mėnesį parduoto konkretaus produkto, bet ir jo pardavimo pelną ar parduotų prekių vienetų skaičių.

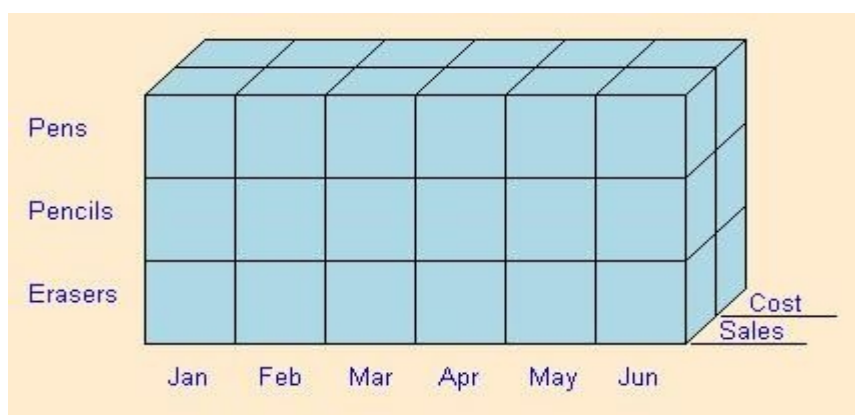
Duomenų bazėje lengvai atlikti įvairius veiksmus:

- Perėjimą nuo bendresnių reikšmių iki individualių ir atvirkščiai, t. y. judėti hierarchine pjūvių (dimensijų) struktūra. Žinodami kurio nors produkto pardavimų pokytį per mėnesį visoje apskrityje, esant reikalui galime pažiūrėti išsamiau - kaip produkto pardavimai kito skirtinguose šios apskrities rajonuose ar net atskirose parduotuvėse.
- Duomenų apibendrinimą pagal įvairius parametrus, t. y. analizę įvairiais pjūviais. Pavyzdžiui, atskaitas apie pelną pardavus tam tikrą produktą galime analizuoti ne laike, bet pagal vietovę.
- Analizuojamojo matmens pakeitimą, t. y. analizę pagal skirtingas skaičiuojamąsias reikšmes.

Pavyzdys. Tarkime turime lentelę:

Product	Sales	Cost
Pens	250	191
Pencils	322	217
Erasers	108	83

Galima sudaryti duomenų kubą:



Duomenų kubą galima analizuoti įvairiais pjūviais:

