

**KTU
INFORMATIKOS FAKULTETAS**

HENRIKAS SINICKAS

**KLIENTŲ DUOMENŲ BAZĖS IR SANTYKIO SU
KLIENTAIS VALDYMO PROGRAMA**

MAGISTRO DARBAS

Darbo vadovas
doc. dr. Ojaras Purvinis

KAUNAS 2004

TURINYS

1. ĮVADAS	3
2. PROBLEMINĖS SRITIES TEORINIAI PAGRINDAI	5
2.1. MYSQL DUOMENŲ BAZĖ	5
2.2. APIE PHP.....	38
3. VADYBOS MOKYMO CENTRO VEIKLOS PROCESO MODELIAI	29
3.1. VADYBOS MOKYMO CENTRO VEIKLOS APRAŠYMAS	29
3.2. DUOMENŲ SRAUTŲ DIAGRAMOS	30
3.3. PROVISION WORKBENCH MODELIAI.....	32
4. REIKALAVIMŲ SPECIFIKAVIMAS	37
4.1. REIKALAVIMŲ SPECIFIKAVIMO DOKUMENTAS	37
4.2. BENDRIEJI REIKALAVIMAI.....	38
4.3. FUNKCINIAI REIKALAVIMAI	38
4.4. REIKALAVIMAI SAŠAJAI	40
4.5. SISTEMOS DARBO REIKALAVIMAI.....	41
4.6. PROJEKTO APRIBOJIMAI	41
4.7. KITI NEFUNKCINIAI SISTEMOS ATRIBUTAI.....	41
4.8. TAIKOMUJŲ UŽDAVINIŲ MODELIS	41
4.9. PROVISION WORKBENCH MODELIAI.....	45
5. IS ARCHITEKTŪROS MODELIAI	47
5.1. IS STRUKTŪRA	47
5.2. PROVISION WORKBENCH MODELIAI.....	48
6. DB PROPJEKTAS	52
6.1. NORMALIZAVIMAS.....	52
6.2. ER DIAGRAMOS	57
6.3. DB RYŠIAI.....	59
6.4. DB PAGRINDINĖ FORMA.....	60
7. KLIENTŲ REGISTRACIJOS INTERNETE MODULIS	61
7.1. PHP ĮDIEGIMAS	61
7.2. MY SQL ĮDIEGIMAS	61
7.3. FORMŲ KŪRIMAS	62
8. IŠVADOS	65
9. LITEATŪRA	66
10. SUMMARY	67
11. PRIEDAI	68
11.1. PRIEDAS NR.1.DB LENTELĖS, FORMOS IR ATASKAITOS	68
11.2. PRIEDAS NR.2.DB "VMC KLIENTAI" ĮDIEGIMAS	73
11.3. PRIEDAS NR.3. PHP SKRIPTAI	74
11.4. PRIEDAS NR.4. MS ACCESS'2000 DB PRIJUNGIMAS PRIE MY SQL SERVERIO	76
11.5. PRIEDAS NR.5. IS PROJEKTO TESTAVIMO REZULTATAI	80
11.6. PRIEDAS NR.6. PROJEKTO PROTOTIPO GENERAVIMO REZULTATAI	81

1. ĮVADAS

Keletą pastarųjų dešimtmečių Vakarų pasaulyje pastebima verslo procesų kaita, kuomet pagrindiniu įmonės tikslu tampa “glaudesni” ryšiai su vartotoju, o jai tai užtikrina pastovų pajamų šaltinį ir išlikimą konkurencinėje kovoje.

Tradicinio marketingo metodai, kuomet didžiausias dėmesys buvo kreipiamas ne į vartotoją, bet į produktą, neefektyvūs ir pasenę, todėl vis daugiau kompanijų atranda naujas marketingo sritis, kaip pavyzdžiui “santykių” marketingą (angl. relationship marketing) ar “duomenimis grįstą” marketingą (angl. databased marketing).

Natūralu, kad didėjant įmonių skaičiui, kurios užsiima ta pačia ar panašia veikla, aštrėja konkurencija, o konkurencinė kova visuomet susijusi su pagrindiniu rinkos dalyviu – vartotoju. Norint išlikti, būtina “geriau” pažinti savo vartotojus, idant jiems galima būtų pasiūlyti jų poreikius atitinkančias prekes ar paslaugas, taip skatinant klientus tapti lojaliais įmonei. ***Visi suprantame, kad masinio marketingo era baigėsi, kad savo veiksmus rinkoje būtina grįsti ne “nuojautomis” ar “patirtimi”, bet žiniomis, kurios įgyjamos analizuojant vartotojų duomenis.***

Reikia suprasti, kad duomenys apie vartotojus, kurie vėliau tampa žiniomis lemia pelną įmonei. Tačiau žiniomis grįsta įmonės veikla galima tik tuomet, kai apdorojus duomenis jų pagrindu priimami motyvuoti sprendimai vartotojams įgyti, išlaikyti. Suprantama, kad vien tik gerų norų neužtenka, būtina turėti specialias priemones, kurios įgalina atlikti minėtus veiksmus bei supaprastina patį sprendimo priėmimą.

Pastebima, kad šiuo metu Vakarų pasaulyje, o taip pat ir Lietuvoje, dauguma IT bei verslo konsultavimo kompanijų pabrėžia duomenų analizės svarbą ir jų taikymo būdus. ***Reikia pripažinti, kad daugelis apsiriboja tik duomenų analizės sritimi, visiškai ar iš dalies nesirūpindami, kokie veiksmai bus atliekami turint apdorotus duomenis apie vartotojus.***

Su šia problema susiduria ne tik Lietuvos, bet ir Vakarų kompanijos, kurios didžiausią dėmesį skyrė duomenų apdorojimo priemonėms įsigyti – pradedant elementariomis ataskaitų sistemomis, baigiant sudėtingomis prognozavimo, ryšių tarp skirtingų prekių / produktų nustatymo priemonėmis.

Tokių sprendimų yra be galo daug ir visi jie remiasi vartotojų duomenų analize, o tokios priemonės vadinamos “intelektualiais verslo sprendimais” (angl. business intelligence -

BI). Tačiau vien tik jų turėjimas nesudaro sąlygų įmonei keisti savo veiklą labiau priartinant prie vartotojo.

Šiuo metu dauguma pasaulio ir Lietuvos IT įmonių skelbiasi parduodančios “santykių su klientais valdymo” sistemas (angl. customer relationship management - CRM), kurios, neva, turėtų išspręsti esamas kompanijų problemas ir užtikrinti efektyvesnę įmonės veiklą.

Santykių su vartotojais valdymas (Customer Relationship Management - CRM) yra automatizuotas ir nuoseklus geresnio vartotojų poreikių, elgesio ir pelningumo supratimo ir numatymo procesas, siekiant nustatyti teisingą strategiją vartotojų atžvilgiu, optimizuoti resursus ir teikti geresnes paslaugas kiekviename kontakto su vartotoju taške.

Darbo tikslas – sukurti IS mokymo kursų paslaugas teikiančioje firmoje, kompiuterizuojant pasirengimo kursų eigai etapą. Etapas apima kursantų, dėstytojų, galimų pasirinkti kursų sąrašų sudarymus, jų koregavimus. Reikia kompiuterizuoti kursų pasirinkimo informacijos suvedimą, mokesčių nustatymą, grupių formavimą. Vartotojai kelia reikalavimus, kad sistema turėtų duomenų saugyklą, iš kurios duomenys lengvai būtų perduodami iš vieno proceso etapo į kitą vietinio tinklo pagalba.

Darbo priemonės – pasirinkau metodologijos realizavimo proceso modelį – IS gyvavimo ciklą. Gyvavimo ciklo pasirinkimas tiesiogiai susijęs su turimos programinės įrangos galimybėmis. Pasirinkau tradicinį GC (top -down) arba kitaip vadinamą – krioklio GC. Šio gyvavimo ciklo skiriami trys etapai (analizė, projektavimas, realizavimas).

Kauno Technologijos Universiteto Panevėžio filialo Vadybos technologijos ir Verslo mokymo centro (VMC) klientų registravimui, apskaitai ir planavimui skirtos informacinės sistemos „VMC klientai“ projektui kurti buvo pasitelkta Provision Workbench priemonės.

Numatyta sukurti : *Veiklos analizės modelius; Vartotojo poreikių modelį; IS projekto modelius; Atlikti IS projekto testavimą; Atlikti IS prototipo generavimą.*

Projekte numatyta sukurti klientų duomenų bazę su CRM elementais MS ACCESS programinės įrangos pagalba ir internetinį puslapį su nuorodomis į reikiamas programas ir klientų registravimu bei duomenų kaupimu MySQL duomenų bazėje ir šios duomenų bazės nuotoliniu administravimu PHP programinės įrangos pagalba sukurtais skriptais.

2. PROBLEMINĖS SRITIES TEORINIAI PAGRINDAI

2.1. MySQL duomenų bazė

MySQL - viena iš reliacinių duomenų bazių apdorojimo sistemų. *Duomenų bazių sistema* yra skirta tvarkyti, rūšiuoti ir manipuluoti informaciją. Kadangi MySQL duomenų bazė leidžia įtraukti, keisti ir ištrinti informaciją iš kelių lentelių vienu metu pagal nurodytus kriterijus, ji priskiriama reliacinėms duombazėms (angl. “relational”).

MySQL sistemoje duomenys yra talpinami lentelėse. Kiekviena lentelė yra suskirstyta į eilutes ir stulpelius. Kiekviena eilutė atitinka vieną įrašą. Įrašas gali turėti kelių rūšių informaciją, kuri priklauso nuo stulpelių tipų ir pavadinimų [16].

2.1.1. Literalai

Eilutė – simbolių seka, įrėmintą paprastomis (") arba dvigubomis kabutėmis (")::

'a string'

"another string"

Sveikieji skaičiai

Sveikieji skaičiai vaizduojami skaitmenų sekomis. Slankaus kablelio skaičiuose naudojamas dešimtainės dalies skirtukas `.`.

Sveikųjų skaičių pavyzdžiai:

1221	0	-32
------	---	-----

Slankaus kablelio skaičių pavyzdžiai

294.42	-32032.6809e+10	148.00
--------	-----------------	--------

NULL

NULL reiškia “duomenų nėra”. Tai ne tas pats, kaip 0 skaičiams ar tuščia eilutė.

NULL gali būti atvaizduotas kaip \N, jei naudojame tekstinio failo importą ar eksportą (LOAD DATA INFILE, SELECT ... INTO OUTFILE).

Duomenų bazės, lentelės, indekso, stulpelių vardai ir pseudonimai

Duomenų bazės, lentelės, indekso, stulpelių vardai ir pseudonimai sudaromi pagal vieningas taisykles:

1 lentelė. Vardų sudarymo taisyklės.

Vardas	max ilgis	Leisti simboliai
Duomenų bazės	64	Bet kokie katalogų varduose leisti simboliai, išskyrus /.
Lentelės	64	Bet kokie failų varduose leisti simboliai, išskyrus / arba .
Stulpelio	64	All characters
Pseudonimas	255	All characters

Jei vardas sutampa su rezervuotu žodžiu ar jame yra specialus simbolis, tokį vardą įrėminkite kabutėmis:

```
SELECT * from `select` where `select`.id > 100;
```

MySQL galimos tokios stulpelių nuorodos:

2 lentelė. Stulpelių nuorodos

<i>Stulpelio nuoroda</i>	<i>Reiškia</i>
col_name	Stulpelis col_name bet kurioje lentelėje, panaudotas užklausoje
tbl_name.col_name	Stulpelis col_name tbl_name šioje
db_name.tbl_name.col_name	Stulpelis col_name iš lentelės tbl_name duomenų bazėje db_name. Priimta MySQL 3.22 ar vėlesnėse versijose.
`column_name`	Stulpelis, kuris nurodytas kaip raktas ar jo pavadinime panaudoti spec. simboliai

Užrašas .tbl_name reiškia, kad lentelė tbl_name yra šioje duomenų bazėje. Tokia sintaksė priimta ODBC suderinamumui, nes kai kurios ODBC programos prieš lentelių pavadinimus rašo ` ` simbolį

Ar varduose mažosios raidės skiriamos nuo didžiųjų - priklauso nuo OS.

1. **Vartotojo kintamieji.** Kintamųjų inicializuoti nereikia. Jie prilyginami NULL pagal nutylėjimą. Gijoje panaudoti kintamieji automatiškai atlaisvinami pabaigus giją.

Kintamųjų reikšmes galima nustatyti naudojant SET:

```
SET @variable= { integer expression | real expression | string expression } [, @variable= ...].
```

Kintamųjų reikšmes galima nustatyti ir išraiškoje, panaudojus sintaksę @variable:=expr:

```
select @t1:=(@t2:=1)+@t3:=4,@t1,@t2,@t3;
```

2. **Stulpeliai.** MySQL palaiko tris stulpelių tipus: skaitinius, datų ir laiko bei eilučių tipo.

Aprašydami naudosime tokius žymėjimus:

M - nurodo rodomą ilgą (max 255)

D - nurodo slankaus kablelio skaičiaus skaitmenų skaičių po kablelio. Maksimali reikšmė - 30, tačiau ji negali viršyti M-2.

Kvadratiniai skliaustai { ir } nurodo nebūtinus elementus.

Jei stulpeliui n nurodysite ZEROFILL, MySQL automatiškai pridės atributą UNSIGNED.

TINYINT[(M)] [UNSIGNED] [ZEROFILL] - labai mažas sveikasis: su ženklų – nuo -128 iki 127; be ženklų – nuo 0 iki 255.

SMALLINT[(M)] [UNSIGNED] [ZEROFILL] - mažas sveikasis: su ženklų – nuo -32768 iki 32767; be ženklų – nuo 0 iki 65535.

MEDIUMINT[(M)] [UNSIGNED] [ZEROFILL] - vidutinio dydžio sveikasis: su ženklų – nuo -8388608 iki 8388607; be ženklų – nuo 0 iki 16777215.

INT[(M)] [UNSIGNED] [ZEROFILL] - normalaus dydžio sveikasis: su ženklų – nuo -2147483648 iki 2147483647; be ženklų – nuo 0 iki 4294967295.

INTEGER[(M)] [UNSIGNED] [ZEROFILL] - tai yra INT sinonimas.

BIGINT[(M)] [UNSIGNED] [ZEROFILL] - didelis sveikasis: su ženklų – nuo -9223372036854775808 iki 9223372036854775807; be ženklų – nuo 0 iki 18446744073709551615.

FLOAT[(M,D)] [ZEROFILL] - mažasis (normalaus tikslumo) slankaus kablelio skaičius. Būtinai turi ženklą. Galimos reikšmės nuo -3.402823466E+38 iki -1.175494351E-38, 0 ir nuo 1.175494351E-38 iki 3.402823466E+38. FLOAT be argumento arba su argumentu <= 24 reiškia normalaus tikslumo slankaus kablelio skaičių.

DOUBLE[(M,D)] [ZEROFILL] - normalaus dydžio (dvigubo tikslumo) slankaus kablelio skaičius. Būtinai turi ženklą. Galimos reikšmės nuo -1.7976931348623157E+308 iki -2.2250738585072014E-308, 0 ir nuo 2.2250738585072014E-308 iki 1.7976931348623157E+308. Čia M nurodo skilčių skaičių, o D trupmeninės dalies ilgį. DOUBLE be argumento arba FLOAT(X), kai $25 \leq X \leq 53$, reiškia dvigubo tikslumo slankaus kablelio skaičių.

DOUBLE PRECISION[(M,D)] [ZEROFILL]

REAL[(M,D)] [ZEROFILL] - tai yra DOUBLE sinonimas.

DECIMAL[(M[,D])] [ZEROFILL] - nepakuotas slankaus kablelio skaičius. Būtinai turi ženklą. Vaizduojamas kaip CHAR stulpelis: "Nepakuota" reiškia, kad skaičius saugomas kaip eilutė, vienam skaitmeniui skiriant vieną simbolį. Į M įeina dešimtainis taškas (kablelis) ir '-' ženklas (neigiamiems skaičiams). Jei D lygu 0, turime tik sveiką dalį. Galimos reikšmės apsprendžia M ir D. Jei D nenurodytas, laikoma, kad jis lygus 0. Jei M nenurodytas, laikoma, kad jis lygus 10.

NUMERIC(M,D) [ZEROFILL] - tai yra DECIMAL sinonimas.

DATE - data. Galimos reikšmės nuo '1000-01-01' iki '9999-12-31'. MySQL DATE reikšmės rodo 'YYYY-MM-DD' formatu, bet jas galima priskirti kaip eilutes ar skaičius.

DATETIME - datos ir laiko kombinacija. Galimos reikšmės nuo '1000-01-01 00:00:00' iki '9999-12-31 23:59:59'. MySQL DATETIME reikšmės rodo 'YYYY-MM-DD HH:MM:SS' formatu, bet jas galima priskirti kaip eilutes ar skaičius.

TIMESTAMP(M) - laiko žyma. Galimos reikšmės nuo '1970-01-01 00:00:00' iki 2037 metų. MySQL **TIMESTAMP** reikšmės rodo YYYYMMDDHHMMSS, YYMMDDHHMMSS, YYYYMMDD arba YYMMDD formatu, priklausomai nuo M: 14 (ar nenurodytas), 12, 8 arba 6, bet jas galima priskirti kaip eilutes ar skaičius.

TIMESTAMP stulpelis naudingas datai ir laikui įterpti **INSERT** arba **UPDATE** operacijomis, nes leidžia automatiškai fiksuoti paskutinio veiksmo datą ir laiką.

TIME - laikas. Galimos reikšmės nuo '-838:59:59' iki '838:59:59'. MySQL **TIME** reikšmės rodo 'HH:MM:SS' formatu, bet jas galima priskirti kaip eilutes ar skaičius.

YEAR[(2|4)] -metai 2- arba 4- skaitmenų formatu (pagal nutylėjimą – 4). Galimos reikšmės nuo 1901 iki 2155, ir 0000 4 skaitmenų formatu ir 1970-2069 – 2 skaitmenų formatu (70-69).

MySQL **YEAR** reikšmės rodo YYYY formatu, bet jas galima priskirti kaip eilutes ar skaičius.

CHAR(M) [BINARY] -fiksuoto ilgio eilutė, įrašant glaudžiama prie dešiniojo krašto. M = 1-255. **CHAR** rikiuojamos ir palyginamos nepaisant registro.

[NATIONAL] VARCHAR(M) [BINARY] -kintamo ilgio eilutė. M = 1-255. **VARCHAR** rikiuojamos ir palyginamos nepaisant registro (jei nenurodyta **BINARY**).

TINYBLOB

TINYTEXT

-**BLOB** (*Binary Large Object*) arba **TEXT** stulpelis, kurio maksimalus ilgis 255 ($2^8 - 1$) simboliai. **BLOB** rikiuojant ir palyginant atsižvelgiama į registrą.

BLOB

TEXT

-**BLOB** arba **TEXT** stulpelis, kurio maksimalus ilgis 65535 ($2^{16} - 1$) simboliai.

MEDIUMBLOB

MEDIUMTEXT

-**BLOB** arba **TEXT** stulpelis, kurio maksimalus ilgis 16777215 ($2^{24} - 1$) simboliai.

LOBLOB

LONGTEXT

-**BLOB** arba **TEXT** stulpelis, kurio maksimalus ilgis 4294967295 ($2^{32} - 1$) simboliai.

ENUM('value1','value2',...)

-Išvardijimas. Eilutės tipo objektas, galintis įgyti tik vieną reikšmę iš sąrašo 'value1', 'value2', ..., arba **NULL**. Maksimalus skirtingų reikšmių skaičius – 65535.

SET('value1','value2',...)

Rinkinys. Eilutės tipo objektas, galintis įgyti 0 ar kelias reikšmes iš sąrašo 'value1', 'value2', ... Maksimalus skirtingų reikšmių skaičius – 64.

2.1.2 Stulpelio tinkamo tipo parinkimas

Norint maksimaliai efektyviai naudoti atmintį, reikia kiekvieną kartą pasirinkti tinkamiausią stulpelio tipą. Pavyzdžiui, jei stulpelyje bus sveikieji skaičiai nuo 1 iki 99999, pasirinkite **MEDIUMINT UNSIGNED**.

Daugeliu atvejų svarbu pasirinkti tinkamą stulpelio tipą pinigų reikšmėms saugoti. MySQL naudokite tipą DECIMAL. Ji saugoma eilutės pavidalu, tad nebus paklaidos dėl tikslumo.

Stulpelio indeksai. Visų MySQL tipų stulpeliai gali būti indeksuoti. Tai padidina SELECT operacijų vykdymo efektyvumą.

Lentelėje gali būti indeksuota iki 16 stulpelių. Maksimalus indekso ilgis – 256 baitai.

CHAR ir VARCHAR tipo stulpeliams galima indeksuoti stulpelio prefiksą. Tai žymiai sparčiau ir reikalauja mažiau vietos diske. CREATE TABLE operatoriuje tai nurodoma taip:

```
KEY index_name (col_name(length))
```

Žemiau pateiktame pavyzdyje sukuriamas pirmųjų 10 simbolių indeksas stulpeliui *name*:

```
mysql> CREATE TABLE test (name CHAR(200) NOT NULL,  
                           KEY index_name (name(10)));
```

BLOB ir TEXT stulpeliams visada reikia taip daryti.

Kelių stulpelių indeksai. MySQL gali suformuoti kelių (iki 15) stulpelių indeksus. (CHAR ir VARCHAR stulpeliams kaip indekso dalį galima naudoti prefiksus). Kelių stulpelių indeksas suprantamas kaip jų reikšmių konkatenacijos bendras indeksas.

MySQL kelių stulpelių indeksą panaudoja greitai užklausų realizacijai.

Tarkime, sukurta lentelė tokiu būdu:

```
mysql> CREATE TABLE test (  
    id INT NOT NULL,  
    last_name CHAR(30) NOT NULL,  
    first_name CHAR(30) NOT NULL,  
    PRIMARY KEY (id),  
    INDEX name (last_name,first_name));
```

Čia indeksas *name* sukurtas stulpeliams *last_name* ir *first_name*. Jį panaudosime užklausoje, kuriose bus apibrėžtos *last_name* reikšmės, arba abiejų stulpelių (*last_name* ir *first_name*) reikšmės. Taigi, *name* indeksą panaudosime tokiose užklausoje:

```
mysql> SELECT * FROM test WHERE last_name="Widenius";
```

```
mysql> SELECT * FROM test WHERE last_name="Widenius"  
        AND first_name="Michael";
```

```
mysql> SELECT * FROM test WHERE last_name="Widenius"  
        AND (first_name="Michael" OR first_name="Monty");
```

```
mysql> SELECT * FROM test WHERE last_name="Widenius"  
        AND first_name >="M" AND first_name < "N";
```

Tačiau *name* indekso negalėsime naudoti tokiose užklausoje:

```
mysql> SELECT * FROM test WHERE first_name="Michael";
```

```
mysql> SELECT * FROM test WHERE last_name="Widenius"
      OR first_name="Michael";
```

Kitų DBVS stulpelių tipai. Kitų DBVS panaudojimui SQL užklausoje palengvinti MySQL tokiu būdu atvaizduoja tipus:

3 lentelė. Stulpelių tipai

<i>Kitose DBVS</i>	<i>MySQL tipas</i>
BINARY(NUM)	CHAR(NUM) BINARY
CHAR VARYING(NUM)	VARCHAR(NUM)
FLOAT4	FLOAT
FLOAT8	DOUBLE
INT1	TINYINT
INT2	SMALLINT
INT3	MEDIUMINT
INT4	INT
INT8	BIGINT
LONG VARBINARY	MEDIUMBLOB
LONG VARCHAR	MEDIUMTEXT
MIDDLEINT	MEDIUMINT
VARBINARY(NUM)	VARCHAR(NUM) BINARY

Atitikimas realizuojamas sukuriant lentelę.

SELECT ir WHERE operatoriuose naudojamos funkcijos

NULL turinti išraiška visada duoda reikšmę NULL, jei tik dokumentacijoje nenurodyta kitaip. Pastaba: Tarp funkcijos pavadinimo ir skliausto negali būti tarpo. Argumentai gali būti atskirti tarpais.

Grupavimo funkcijos:

(...)

Skliaustai. Reikalingi išraiškos skaičiavimo tvarkai apibrėžti.

```
mysql> select 1+2*3;
```

```
-> 7
```

```
mysql> select (1+2)*3;
```

```
-> 9
```

Įprastos aritmetinės operacijos:

+ sudėtis - atimtis

```
mysql> select 3+5;
```

```
-> 8
```

```
mysql> select 3-5;
```

```
-> -2
```

* daugyba / dalyba

```
mysql> select 3*5;
```

```
-> 15
```

```
mysql> select 3/5;
```

```
-> 0.60
```

Dalyba iš nulio duoda NULL:

Bitų funkcijos. Bitų operacijoms MySQL naudoja BIGINT (64-bit) aritmetiką, taigi, šie

operatoriai duoda maksimalų ilgį 64 bitus.

| OR (pabičiui)

```
mysql> select 29 | 15;
```

```
-> 31
```

& AND (pabičiui)

```
mysql> select 29 & 15;
```

```
-> 13
```

```

<<          postūmis į kairę          >>          postūmis į dešinę
mysql> select 1 << 2          mysql> select 4 >> 2
      -> 4                      -> 1
~          visų bitų inversija          BIT_COUNT(N)          vienetukų skaičius
mysql> select 5 & ~1          mysql> select BIT_COUNT(29);
      -> 4                      -> 4

```

Loginės operacijos

Jų rezultatas – 1 (TRUE) arba 0 (FALSE).

```

NOT arba ! loginė NE          OR arba || loginė ARBA
mysql> select NOT 1;          mysql> select 1 || 0;
      -> 0                      -> 1
mysql> select NOT NULL;          mysql> select 0 || 0;
      -> NULL                    -> 0
mysql> select !(1+1);          mysql> select 1 || NULL;
      -> 0                      -> 1
mysql> select ! 1+1;
      -> 1

```

NOT NULL duoda NULL.

Duoda 1 jei argumentas nelygus 0 .

```

AND arba && loginė IR
mysql> select 1 && NULL;
      -> 0
mysql> select 1 && 0;
      -> 0

```

Palyginimo operacijos. Jų rezultatas būna lygus 1 (TRUE), 0 (FALSE) arba NULL. Šios funkcijos atliekamos su skaičiais ir eilutėmis. Eilutės automatiškai konvertuojamos į skaičius arba atvirkščiai, kai to reikia.

Palyginimo operacijos MySQL atliekamos pagal tokias taisykles:

- * jei vienas ar abu argumentai lygūs NULL, rezultatas yra NULL, išskyrus operaciją <=>;
- * jei abu argumentai yra eilutės, jie palyginami kaip eilutės;
- * jei abu argumentai yra skaičiai, jie palyginami kaip skaičiai;
- * jei vienas argumentas yra TIMESTAMP ar DATETIME tipo, o kitas – konstanta, prieš palyginant ši konvertuojama į TIMESTAMP ;
- * visais kitais atvejais argumentai palyginami kaip slankaus kablelio skaičiai.

Pavyzdžiai:

```

mysql> SELECT 1 > '6x';
      -> 0
mysql> SELECT 7 > '6x';
      -> 1
mysql> SELECT 0 > 'x6';
      -> 0
mysql> SELECT 0 = 'x6';
      -> 1
=          lygu          <> arba !=          nelygu
mysql> select 1 = 0;          mysql> select '.01' <> '0.01';
      -> 0                      -> 1
mysql> select '0' = 0;          mysql> select .01 <> '0.01';

```

```

-> 1
mysql> select '0.0' = 0;
-> 1
mysql> select '0.01' = 0;
-> 0
mysql> select '.01' = 0.01;
-> 1
<=          mažiau arba lygu          <          mažiau
mysql> select 0.1 <= 2;
-> 1
mysql> select 2 <= 2;
-> 1
>=          daugiau arba lygu        >          daugiau
mysql> select 2 >= 2;
-> 1
mysql> select 2 > 2;
-> 0
<=>        nelygu
mysql> select 1 <=> 1, NULL <=> NULL, 1 <=> NULL;
-> 1 1 0
IS NULL
IS NOT NULL      ar lygu (nelygu) NULL
mysql> select 1 IS NULL, 0 IS NULL, NULL IS NULL;
-> 0 0 1
mysql> select 1 IS NOT NULL, 0 IS NOT NULL, NULL IS NOT NULL;
-> 1 1 0
expr BETWEEN min AND max
Jei expr >= min ir expr <= max, BETWEEN duoda 1, priešingu atveju – 0.

```

Pirmasis (expr) argumentas apsprendžia palyginimo vykdymą:

- jei tai yra TIMESTAMP, DATE arba DATETIME stulpelis, min ir max are formatuojami vienodai, jei tai – konstantos;
- jei tai yra registruojamam nejautri išraiška, prilyginamos kaip registruojamam nejautrios eilutės;
- jei tai yra registruojamam jautri išraiška, palyginimas kaip registruojamam jautrios eilutės;
- jei tai yra sveikųjų skaičių išraiška, palyginami kaip sveikieji skaičiai;
- visais kitais atvejais palyginami kaip slankaus kablelio skaičiai. •

```

mysql> select 1 BETWEEN 2 AND 3;
-> 0
mysql> select 'b' BETWEEN 'a' AND 'c';
-> 1
mysql> select 2 BETWEEN 2 AND '3';
-> 1
mysql> select 2 BETWEEN 2 AND 'x-3';
-> 0

```

expr IN (value,...)

Duoda 1, jei expr lygi vienai iš sąrašo IN list nurodytų reikšmių, kitaip duoda 0.

```

mysql> select 2 IN (0,3,5,'wefwf');
-> 0
mysql> select 'wefwf' IN (0,3,5,'wefwf');
-> 1

```

expr NOT IN (value,...)

Atitinka NOT (expr IN (value,...)).

ISNULL(expr)

Jei expr yra NULL, ISNULL() duoda 1, priešingu atveju – 0.

```
mysql> select ISNULL(1+1);
```

```
-> 0
```

```
mysql> select ISNULL(1/0);
```

```
-> 1
```

COALESCE(list)

Duoda pirmąjį ne-NULL elementą iš sąrašo.

```
mysql> select COALESCE(NULL,1);
```

```
-> 1
```

```
mysql> select COALESCE(NULL,NULL,NULL);
```

```
-> NULL
```

INTERVAL(N,N1,N2,N3,...)

Duoda 0, jei $N < N1$, ir 1, jei $N < N2$ ir t.t.. Visi argumentai laikomi sveikaisiais. Reikia, kad $N1 < N2 < N3 < \dots < Nn$.

```
mysql> select INTERVAL(23, 1, 15, 17, 30, 44, 200);
```

```
-> 3
```

```
mysql> select INTERVAL(10, 1, 10, 100, 1000);
```

```
-> 2
```

```
mysql> select INTERVAL(22, 23, 30, 44, 200);
```

```
-> 0
```

2.1.3. Eilučių palyginimo funkcijos

Paprastai, jei kuri nors eilutė (iš palyginamų) registru, palyginimas atliekamas “jautriai”.

expr LIKE pat [ESCAPE 'escape-char']

Duoda 1 (TRUE) arba 0 (FALSE). LIKE galima naudoti du universalūs simbolius:

% atitinka bet kiek (ir 0) simbolių, o _ atitinka tik vieną simbolį.

```
mysql> select 'David!' LIKE 'David_';
```

```
-> 1
```

```
mysql> select 'David!' LIKE '%D%v%';
```

```
-> 1
```

Jei reikia lyginti su tuo universaliu simboliu, prieš jį pridėkite \:

\% atitinka vieną % simbolį, _ atitinka vieną _ simbolį

```
mysql> select 'David!' LIKE 'David\_';
```

```
-> 0
```

```
mysql> select 'David_' LIKE 'David\_';
```

```
-> 1
```

LIKE galima naudoti skaitmeninėse išraiškose!

```
mysql> select 10 LIKE '1%';
```

```
-> 1
```

expr NOT LIKE pat [ESCAPE 'escape-char']

Atitinka NOT (expr LIKE pat [ESCAPE 'escape-char']).

expr REGEXP pat

expr RLIKE pat

Eilutėje expr ieško fragmento pat. Duoda 1, jei expr turi pat, priešingu atveju – 0. RLIKE yra REGEXP sinonimas (dėl mSQL).

```
mysql> select 'Monty!' REGEXP 'm%y%%';
```

```
-> 0
```

```
mysql> select 'Monty!' REGEXP '.*';
```

```
-> 1
```

```
mysql> select 'new*\n*line' REGEXP 'new\\*.*line';
```

```
-> 1
```

```
mysql> select "a" REGEXP "A", "a" REGEXP BINARY "A";
```

```
-> 1 0
```

expr NOT REGEXP pat

expr NOT RLIKE pat

Atitinka NOT (expr REGEXP pat).

STRCMP(expr1,expr2)

STRCMP() duoda 0, jei eilutės sutampa, -1, jei pirmasis argumentas mažesnis už antrąjį, kitais atvejais duoda 1.

```
mysql> select STRCMP('text', 'text2');
```

```
-> -1
```

```
mysql> select STRCMP('text2', 'text');
```

```
-> 1
```

```
mysql> select STRCMP('text', 'text');
```

```
-> 0
```

Pakeitimo operatoriai. BINARY operatorius pakeičia nurodytą eilutę dvejetainė. Tai praverčia pereiti prie jautraus registrai palyginimo, nors stulpelis aprašytas kaip nejautrus (ne BINARY ar BLOB).

```
mysql> select "a" = "A";
```

```
-> 1
```

```
mysql> select BINARY "a" = "A";
```

```
-> 0
```

BINARY įvestas MySQL 3.23.0

2.1.4. Valdymo funkcijos

IFNULL(expr1,expr2) -Jei expr1 yra ne NULL, IFNULL() duoda expr1, priešingu atveju – expr2.

IFNULL() duoda skaitmeninę arba eilutės reikšmę (priklausomai nuo konteksto).

```
mysql> select IFNULL(1,0);
```

```

-> 1
mysql> select IFNULL(0,10);
-> 0
mysql> select IFNULL(1/0,10);
-> 10
mysql> select IFNULL(1/0,'yes');
-> 'yes'

```

IF(expr1,expr2,expr3)

Jei expr1 yra TRUE (expr1 <> 0 ir expr1 <> NULL), tai IF() duoda expr2, priešingu atveju – expr3. IF() duoda skaitmeninę arba eilutės reikšmę (priklausomai nuo konteksto).

```

mysql> select IF(1>2,2,3);
-> 3
mysql> select IF(1<2,'yes','no');
-> 'yes'
mysql> select IF(strcmp('test','test1'),'yes','no');
-> 'no'

```

expr1 paskaičiuojama kaip skaitinė reikšmė. Tai reikia įvertinti, tikrinant slankaus kablelio skaičius ar eilutes.

```

mysql> select IF(0.1,1,0);
-> 0
mysql> select IF(0.1<>0,1,0);
-> 1

```

Pirmame pavyzdyje IF(0.1) duoda 0, nes 0.1 konvertuojama į sveikąjį. Antrame pavyzdyje tikrinama slankaus kablelio skaičiaus reikšmė. Palyginimo rezultatas – sveikasis skaičius.

CASE value WHEN [compare-value] THEN result [WHEN [compare-value] THEN result ...] [ELSE result] END

CASE WHEN [condition] THEN result [WHEN [condition] THEN result ...] [ELSE result] END

Pirmame variante gaunamas rezultatas, kai value=compare-value. Antrame variante gaunamas rezultatas, atitinkantis patenkinimą sąlygą. Jei nėra netenkinama, gaunamas po ELSE nurodytas rezultatas is returned. Jei nėra ELSE dalies, tai NULL.

```

mysql> SELECT CASE 1 WHEN 1 THEN "one" WHEN 2 THEN "two" ELSE "more" END;
-> "one"
mysql> SELECT CASE WHEN 1>0 THEN "true" ELSE "false" END;
-> "true"
mysql> SELECT CASE BINARY "B" when "a" then 1 when "b" then 2 END;
-> NULL

```

2.1.5. Veiksmai su eilutėmis

Šios funkcijos duoda NULL, jei rezultato ilgis viršija max_allowed_packet serverio parametą pozicijos numeruojamos nuo 1.

ASCII(str) - duoda kairiausio simbolio ASCII kodą. Duoda NULL, jei eilutė lygi NULL.

```

mysql> select ASCII('2');
-> 50

```

```
mysql> select ASCII(2);
```

```
-> 50
```

```
mysql> select ASCII('dx');
```

```
-> 100
```

BIN(N) OCT(N) HEX(N)-Duoda N dvejetainę, aštuntainę ar šešioliktainę reikšmę.

```
mysql> select BIN(12);
```

```
-> '1100'
```

```
mysql> select OCT(12);
```

```
-> '14'
```

```
mysql> select HEX(255);
```

```
-> 'FF'
```

CHAR(N,...)

2.1.6. Datos ir laiko funkcijos

Žemiau pateiktas operatorius atrenka visus įrašus, padarytus per paskutines 30

dienų:

```
mysql> SELECT something FROM table
```

```
WHERE TO_DAYS(NOW()) - TO_DAYS(date_col) <= 30;
```

DAYOFWEEK(date) -Pagal datą nurodo savaitės dieną (1 = Sunday, 2 = Monday, ... 7 = Saturday).

```
mysql> select DAYOFWEEK('1998-02-03');
```

```
-> 3
```

WEEKDAY(date) -Pagal datą nurodo savaitės dieną (0 = Monday, 1 = Tuesday, ... 6 = Sunday).

```
mysql> select WEEKDAY('1997-10-04 22:23:00');
```

```
-> 5
```

```
mysql> select WEEKDAY('1997-11-05');
```

```
-> 2
```

DAYOFMONTH(date) -Pagal datą nurodo mėnesio dieną. mysql> select DAYOFMONTH('1998-02-03');

```
-> 3
```

DAYOFYEAR(date) -Pagal datą nurodo metų dieną (1 iki 366). mysql> select DAYOFYEAR('1998-02-03');

```
-> 34
```

MONTH(date) -Pagal datą nurodo mėnesio numerį (1 iki 12). mysql> select MONTH('1998-02-03');

```
-> 2
```


`DAYNAME(date)` -Pagal datą nurodo savaitės dieną. `mysql> select DAYNAME("1998-02-05");`
-> 'Thursday'

`MONTHNAME(date)` -Pagal datą nurodo mėnesį. `mysql> select MONTHNAME("1998-02-05");`
-> 'February'

`QUARTER(date)` -Pagal datą nurodo ketvirtį (1 iki 4). `mysql> select QUARTER('98-04-01');`
-> 2

`WEEK(date)`

`WEEK(date,first)` -Pagal datą nurodo savaitę (0 iki 53) laikant, kad sekmadienis (Sunday) yra pirmoji savaitės diena. Antrasis `WEEK()` argumentas `first` nurodo, ar savaitė pradedama sekmadieniu (Sunday - 0) ar pirmadieniu (Monday - 1).

`mysql> select WEEK('1998-02-20');`
-> 7

`mysql> select WEEK('1998-02-20',0);`
-> 7

`mysql> select WEEK('1998-02-20',1);`
-> 8

`mysql> select WEEK('1998-12-31',1);`
-> 53

`YEAR(date)` -Pagal datą nurodo metus (1000 iki 9999). `mysql> select YEAR('98-02-03');`
-> 1998

`YEARWEEK(date)`

`YEARWEEK(date,first)` -Pagal datą nurodo metus ir savaitę. Antrasis `WEEK()` argumentas `first` nurodo, ar savaitė pradedama sekmadieniu (Sunday - 0) ar pirmadieniu (Monday - 1).

`mysql> select YEARWEEK('1987-01-01');`
-> 198653

`HOUR(time)` `MINUTE(time)` -Pagal laiką nurodo valandą (0 iki 23), minutę (0-59).

`mysql> select HOUR('10:05:03');`
-> 10

`mysql> select MINUTE('98-02-03 10:05:03');`
-> 5

`PERIOD_ADD(P,N)` -Prideda N mėnesių prie periodo P (formate YYMM ar YYYYMM). Reikšmę pateikia formatu YYYYMM.

`mysql> select PERIOD_ADD(9801,2);`
-> 199803

`PERIOD_DIFF(P1,P2)` -Pateikia mėnesiais išreikštą skirtumą tarp P1 ir P2. P1 ir P2 formatas YYMM ar YYYYMM.

```
mysql> select PERIOD_DIFF(9802,199703);
```

```
-> 11
```

2.1.7. Kitos funkcijos

DATABASE() -Nurodo aktyvios DB pavadinimą. mysql> select DATABASE();

```
-> 'test'
```

USER()

SYSTEM_USER()

SESSION_USER() - Nurodo MySQL vartotojo vardą. mysql> select USER();

```
-> 'davida@localhost'
```

MySQL 3.22.11 ir vėlesnėse versijose - ir kliento hosto vardą:

```
mysql> select substring_index(USER(),"@",1);
```

```
-> 'davida'
```

Užkoduoja duotą eilutę str naudodama pass_str kaip slaptažodį. Atkoduoja DECODE().

LAST_INSERT_ID([expr])

Pateikia paskutinę automatiškai sugeneruotą reikšmę, kuri buvo įterpta AUTO_INCREMENT stulpelyje.

```
mysql> select LAST_INSERT_ID();
```

```
-> 195
```

VERSION() -Pateikia MySQL serverio versijos numerį.

```
mysql> select VERSION();
```

```
-> '3.22.19b-log'
```

2.1.8. Funkcijos GROUP BY operatoriams

Jei naudosite grupavimo funkciją operatoriuje, kuriame nėra GROUP BY sakinio, tai atitiks grupavimą visoms eilutėms.

COUNT(expr) -Pateikia ne-NULL reikšmių skaičių eilutėse, kurias nurodo SELECT operatorius.

```
mysql> select student.student_name,COUNT(*)
```

```
from student,course
```

```
where student.student_id=course.student_id
```

```
GROUP BY student_name;
```

COUNT(DISTINCT expr,[expr...]) -Pateikia skirtingų reikšmių skaičių.

```
mysql> select COUNT(DISTINCT results) from student;
```

AVG(expr) -Pateikia vidutinę expr reikšmę

```
mysql> select student_name, AVG(test_score)
      from student
      GROUP BY student_name;
```

MIN(expr) MAX(expr) -Pateikia minimalią ar maksimalią expr reikšmę. MIN() ir MAX() kaip argumentą gali turėti eilutę; tuomet pateikiama eilutės reikšmė.

```
mysql> select student_name, MIN(test_score), MAX(test_score)
      from student
      GROUP BY student_name;
```

SUM(expr) -Pateikia expr reikšmių sumą.

STD(expr) STDDEV(expr) -Pateikia standartinį expr reikšmių atsilenkimą.

MySQL išplėstas GROUP BY panaudojimas. Galima naudoti stulpelius ar skaičiavimus SELECT išraiškose, kurios nepasirodo GROUP BY dalyje. Tai galima naudoti tam, kad padidinti spartą išvengiant nereikalingų elementų rikiavimo ir grupavimo. Pavyzdžiui, nereikia grupuoti customer.name tokioje užklausoje:

```
mysql> select order.custid,customer.name,max(payments)
      from order,customer
      where order.custid = customer.custid
      GROUP BY order.custid;
```

2.2. Apie PHP

PHP (PHP hypertext preprocessor) tai programavimo kalba (scripting language) kuri pačioje pradžioje buvo orientuota tik į Internetą, nors šiuo metu ją galima "drąsiai" pavadinti programavimo kalba, kadangi su ja pilnai galima programuoti ne tik Internetui [16]. Pavyzdžiui su php galima kurti pilnavertiškas platformines programas su GUI (graphical user interface), bei konsolines programėles elektroniniam paštui. PHP kodai yra interpretuojami ir įvykdomi serverio pusėje.

```
<html>
<head>
  <title>Pavyzdys</title>
</head>
<body>
  <?php echo "Sveikas aš esu
PHP kodas"; ?>
</body>
</html>
```

Po šio kodo įvykdymo rezultate (naršyklėje) turėsime:

Sveikas aš esu PHP kodas

Tai gana paprasta bet naudos iš to - jokios. Pradžioje pagrindinis PHP privalumas prieš CGI buvo tai kad PHP kodas galėjo būti lengvai įterpiamas į HTML kodą, kaip CGI reikėjo visą HTML paimti iš CGI kodo. Kuo toliau tuo rečiau HTML yra maišomas su didesniuose projektuose, tam naudojami šablonai.

PHP skirtumas nuo JavaScript (programavimo kalba) yra tai jog PHP kodai yra atliekami serverio pusėje o vartotojui yra gražinamas rezultatas (HTML formate). Rezultatą vartotojas mato savo naršyklėje. Tuo tarpu kai JavaScript yra pilnai perduodamas kliento mašinai (kompiuteriui) ir atliekamas klientinėje dalyje.

PHP sintaksės pagrindai yra paimti iš: C, Perl, Java, todėl programavusiems šiomis kalbomis yra ypatingai lengva pradėti programuoti PHP.

2.2.1. Trumpa PHP istorija

Viskas prasidėjo 1994 metų rudenį, kai Rasmus Lerdorf'as nusprendė patobulinti savo asmeninį puslapį ir parašyti asmeninį varikliuką kuris atlikinėtų paprasčiausius veiksmus. Šį varikliuką parengė 1995 metų pradžioje ir pavadino "Personal Home Page Tools". Šie kodai galėjo ne itin daug, ir galimybės jo buvo tik pačios paprasčiausios, sudarė ją vos keletą Macros komandų. (Patį pirmąjį Rasmus Lerdorfo pranešimą į www konferenciją apie savo "Personal Home Page Tools" 1995 metais.)

1995 metų viduryje atsirado antroji varikliuko versija kuri vadinosi PHP/FI version 2. FI - dėl vienos priežasties: tiesiog Rasmus padarė papildomą paketą kuris jau mokėjo interpretuoti formas (Form Interpretator). PHP/FI kompiliavosi į Apache Web serverį ir naudojo standartinį Apache API. PHP kodai pasirodė spartesni serveryje nei CGI, nes nereikėjo kurti naujo proceso. Po truputi PHP pradėjo lygintis savo galimybėmis su Perl (populiariausia kalba CGI rašymui). Su laiku buvo pridėta ir daugelių duomenų bazių palaikymas (MySQL, Oracle ir t.t.), interfeisas GD bibliotekai (parašytas Rasmuso ir iki šiol kuruojamas jo) leido dinamiškai kurti paveiksliukus. PHP/FI labai greitai pradėjo plisti.

1997 metų pabaigoje Zeev Suraski ir Andi Gutmans nusprendė perrašyti PHP varikliuką su tikslu ištaisyti daugelį klaidų bei pagreitinti PHP kodų atlikimą serveryje. Po pusės metų, 1998..06 išėjo nauja versija kuri buvo pavadinta PHP3. NetCrafto duomenimis PHP 1999 metais naudojosi jau 1 milijonas vartotojų. domenų.

1999 metų pabaigoje buvo vėlgi perrašytas PHP3 varikliukas (pavadinimu Zena Engine) ir išleista nauja versija PHP4. Skirtumai tarp šių versijų yra tikrai dideli, kartu su

pagrindiniu PHP branduoliu atsirado daugybe praplétimų ir jie vis dar nenustojami tobulinti ir rašyti (praplétimai pvz.: gd, xslt, sockets,imap, daugelių DB programos ir t.t. ir t.t.).

Šiuo metu patobulintas PHP varikliukas pavadinimu Zend Engine 2. Jis išleistas 2002 metų viduryje. Zend Engine 2 ir bus PHP5 "širdis bei smegenys". Pasak Zevo daugiausia darbo bus įdėta į naujo objektinio modulio tobulinimą. Ištaisys Objekto kopijavimo atmintyje problema (dabar sunku atkasti ar tu sukuri jau esamo objekto kopiją), ideologija Zevas paėmė iš Java. Tiesiog kuriant objektą (new Object()) kintamasis nebus pats objektas o tiesiog nuoroda į jį, taip darant priskyrimą naujam kintamajam bus priskiriamas nebe pats objektas o jo nuoroda (objektas nekopijuojamas atmintyje). Taip pat atsiras destruktoriai, exception handler (try/throw/catch) ir dauguma kitų pakeitimų (->_clone(),->_construct(), ->_destruct()...)

Apie privalumus ir trūkumus prieš kitas panašaus tipo programavimo kalbas galite rasti Internete. PHPEverywere turi puikų straipsnių rinkinį apie PHP bei cfm,perl,jsp,asp:

php yra visiškai nemokamas

- php yra visiškai universalus (veikia įvairiose operacinėse sistemose: Windows, Linux, MacOS, Solaris, HP-UX, AIX ir t.t.)
- php yra laisvas projektas todėl jį kuria didelė grupė žmonių, todėl iškilusios klaidos yra greitai ištaisomi, ir php sparčiai plečiasi
- kaip ir universalios platformos jis veikia ir ant daugelio WEB serverių: Apache, IIS, PWS, OmniHTTP, BadBlue ir t.t.
- išmokti PHP programavimo pagrindų yra labai lengva
- pasižymi dideliu greičiu serverio pusėje, bei dirbant su duomenų bazėmis
- nedideliuose projektuose PHP jūs paprastai galite įterpti į savo HTML puslapį.
- Programuotojai prirašė praplétimų kiekvienam gyvenimo atvejui
- kadangi php programuotojų yra be galo daug, daugumą jau parašytų kodų galite rasti Internete.

2.2.2. Kintamieji (variables)

Viena iš svarbiausių ir esminių programavimo kalbos struktūrų yra kintamieji. PHP kalboje taip pat apibrėžta kintamojo konstrukcija. Pagrindinė kintamojo paskirtis - saugoti duomenis, kiekvienas kintamasis skripto vykdymo momentu įgyja tam tikrą reikšmę, kuri, vykdant skriptą, gali keistis. Kintamasis apibrėžiamas nurodant jo vardą, po to į kintamąjį kreipiamasi pagal vardą. Išskirtinis PHP kintamojo požymis yra ženklas "\$". Jis visada rašomas prieš kintamąjį: PVZ.:

<? \$vardas = "jonas"; ?> (Šiame pavyzdyje panaudotas priskyrimo operatorius, kuris bus aptartas vėliau) Prasminga parinkinėti vardus, kurie atspindi tikrąją kintamojo paskirtį, pvz: \$vardas = "jonas", \$kaina = 2.00, pagal šiuos pavyzdžius lengva nustatyti kintamojo prasmę. Dėmesio! PHP kintamųjų vardai priklauso nuo raidžių registro, pvz. kintamieji \$Pirmas = 10 ir \$pirmas = 20 yra skirtingi.

2.2.3. Duomenų tipai

PHP kalboje apibrėžti tokie duomenų tipai:

4 lentelė. Duomenų tipai

1.	Sveikieji skaičiai	(integer arba int)
2.	Realieji skaičiai	(double arba real)
3.	Eilutės	(string)
4.	Masyvai	(array)
5.	Objektai	(objects)

Sveikieji skaičiai atmintyje užima 4 baitus ir gali įgyti reikšmes nuo -2147483648 iki 2147483647, realieji skaičiai naudojami realiems dydžiams sagoti ir gali įgyti paprastą bei eksponentinę išraišką (pvz.: 1.25 ir 1.25e4 = 1250). Eilutės skirtos saugoti kitiems simboliams (raidėms, ženklams ir tt), taip pat ir skaičiams (pastaba: 2 ir "2" reikšmės skiriasi 2 - tai skaičius, o - "2" - simbolis, "2" ekvivalentu "a"). Apie masyvus ir objektus paskaitysite vėliau. Daugelyje kalbų yra apibrėžtas loginis (boolean) tipas, turintis tik dvi reikšmes - teisinga arba klaidinga. PHP kalboje jo nėra, tačiau jis keičiamas kitais tipais. Reikšmę teisinga (true) atitinka bet koks sveikas skaičius, nelygus 0, realus skaičius, nelygus 0 bei netuščia eilutė, reikšmę klaidinga (false) - 0 (integer), 0.000..000 (double), "" - tuščia eilutė.

2.2.4. Konstantos (constants)

Konstantos yra nesikeičiantys dydžiai. PHP kalboje konstantos būna dviejų tipų: tikslios (literal) (pvz, skripto tekste parašytas skaičius 10 (if (\$a < 10) {echo ("laba diena, mielieji.")}), eilutė "laba diena, mielieji." Kitas konstantų tipas nustatomos pagal apibrėžtį. Konstantas apibrėžia funkcija define(). PVZ:.

```

    <?
        define( "PAVADINIMAS", "PUMPENU
SURIAI" );
define( "NE", "<BR>n" );
echo( "Firmos pavadinimas: ".PAVADINIMAS ); // "." - sujungimo operatorius
echo( "Firmos pavadinimas: PUMPENU SURIAI" ); // ši eilute ekvivalenti pirmai
?>

```

(pastaba: konstantos aprašomos be ženklo "\$")

Norėdami nustatyti, ar konstanta egzistuoja, naudokite funkciją defined(), Ji grąžina 1, jei tokia konstanta apibrėžta, priešingu atveju - 0.

```

<?
define( "PAVADINIMAS", "PUMPENU SURIAI" );
define( "NE", "<BR>n" );
echo( "Firmos pavadinimas: ".PAVADINIMAS ); // "." - sujungimo operatorius
echo( "Firmos pavadinimas: PUMPENU SURIAI" ); // ?i eilute ekvivalenti pirmai
?>

```

2.2.5. Vidinės konstantos

Vidinės konstantos yra tokios konstantos, kurios apibrėžtos kalbos vduje. PVZ, galima naudoti konstantas TRUE ir FALSE (jų reikšmes jau paaiškinome anksčiau). Taip pat yra nemažai kitų konstantų, kurios grąžina įvairios paskirties informaciją. Jas ir jų reikšmes galima sužinoti iš funkcijos phpinfo().

Kintamojo apibrėžimas ir pirminės reikšmės suteikimas.

Kintamųjų, kitaip negu konstantų, nereikia aprašyti papildomai. Kintamasis apibrėžiamas automatiškai, kai jam priskiriama reikšmė. Priskyrimo operatorius PHP kalboje yra "="; Pvz:

```

<?
    $a = 512; // nuo cia apibrečiamas kintamasis $a ir jam priskiriama reikšme 512
    $b = $a; // apibrečiamas kintamasis $b ir jam priskiriama $a reikšme
    $c = $a + $b; // $c priskiriama reikšme $a + $b (1024)
?>

```

PHP kalbo nėra griežtai tipizuota kalba ir joje nereikia apibrėžti kintamojo tipo. Bet kuriuo metu kintamasis gali įgyti bet kokio tipo reikšmę. Kintamojo tipas nustatomas pagal tai, kokia išraiška priskiriama kintamajam. Pvz.:

```

<? $a = 10; // dabar kintamasis $a yra sveikas skaičius
    $a = "mama"; // dabar $a jau eilute
?>

```

2.2.6. Perėjimas nuo eilučių prie skaičių (string conversion)

Jei Jūs su eilute atliksite skaitinę operaciją, PHP laikys eilutę skaičiumi. Pvz:

```
<?
$eil = "10 dienu ka_kas stabdo tinkla"; // $eil - string tipo kintamasis
$x = $eil + 3; // $eil bus laikomas integer tipo kintamuoju su reikšme 10
// $x bus integer tipo kintamasis su reikšme 13
// $eil reikšme nepakis
?>
```

Perėjimas nuo eilučių prie skaičių atliekamas pagal tokias taisykles:

1. PHP "moka ištraukti" sveiką skaičių iš eilutės tik tada, jei ji prasideda skaičiumi. Pvz:

```
<?
$a = "10 pinigų";
$b = $a + 10;          // $b = 20
$c = "Maai pinigų, tik 10";
$d = $c + 10;         // $d = 10
?>
```

Jei eilutė neprasideda skaičiumi, jos perkonvertuota reikšmė bus 0.

2. Eilutė bus laikoma realiu skaičiumi tik tada, jei visą eilutę atitiks double tipo išraiška. Eilutės "10.25", "-1.25", "1.5e-10" įgytų tokias reikšmes, kokias pačios atitinka. Jei eilutėje yra dar kitų simbolių, laikoma, kad eilutė atitinka integer tipą ir reikšmė apskaičiuojama pagal 1 punktą. Eilutės "4.58 litro" reikšmė 4, "-4.2 laipsniai" - -4, "Kambario temperatūra 10.5" - 0.

2.2.7. Tipo keitimas

Tipo keitimas leis tiksliai pakeisti kintamojo tipą:

```
<?
$a = 15.2;           // $a yra double tipo
$a = (int) $a;      // $a yra integer tipo (reikšme 15)
$a = (double) $a;   // $a vel yra double tipo (reikšme 15.0)
$b = (string) $a;   // $b yra eilute "15"
?>
```


2.2.8. Kintami kintamieji

PHP palaiko kintamuosius kintamuosius. Paprasto kintamojo reikšmė yra dinaminė, ją galima keisti. O tuo tarpu kintamųjų kintamųjų vardai yra dinaminiai. Ištikrųjų, kintami kintamieji įneša daugiau painianiaus negu naudos. Kintamo kintamojo pavyzdys:

```
<?
$ID = "ProductID"; // sukuriamas kintamasis, kurio reikšmė yra "ProductID"
$$ID = 100; // Pirmojo kintamojo $ID reikšmė "ProductID" naudojama kaip kintamojo
// vardas, t.y. sukuriamas kintmasis $ProductID ir jam priskiriama
// reikšmė 100 ($ProductID = 100;)
echo( $ProductID ); // Spausdins: 100
echo( $$ID ); // Spausdins: 100
echo( $ID ); // Spausdins: ProductID
?>
```

Galima net tokia situacija:

```
<?
$ID = "ProductID"; //
$$ID = "RealID"; // $ProductID = "RealID"
$$$ID = 100; // $RealID = 100;
?>
```

2.2.9. Darbai su kintamaisiais naudingos funkcijos

gettype() -Gettype() grąžina kintamojo duomenų tipą. Ji grąžina vieną iš šių reikšmių:

```
"integer" // sveikasis skaičius
"double" // realus skaičius
"string" // eilutė
"array" // masyvas
"object" // objektas
"class" // klasė
"unknown type" // nežinomas tipas
```

Apie masyvus, objektus ir klases sužinosite vėliau. `gettype()` pvz:

```
<?
if (gettype($input) == "integer")
{ $integer_value = $input;
```

```
}
    ?>
```

settype() -settype() tiksliai nustato kintmojo tipą. Tipas užrašomas eilute ir gali būti toks: "array", "double", "object", "integer", "string". Apie tipo nustatymą jau kalbėjome, tačiau ši funkcija ypatinga tuo, jog gražina true, jei tipas pakeistas sėkmingai, priešingu atveju - false.

Pvz:

```
<?
$a = 78.5;
settype( $a, "integer" ); // $a = 78;
if (settype( $value, "integer"))
{
    echo( "Pakeitimas sėkmingas" );
}
else
{
    echo( "Tipa keitimas nepavyko" );
}
?>
```

isset(), unset() -Funkcija unset() skirta kintamųjų sunaikinimui. Ji išlaisvina atminties vietą, kurią buvo užėmęs kintamasis. Funkcija isset() parodo, ar kintamasis yra apibrėžtas (jam priskirta reikšmė) ir gražina true, jei taip. Pvz:

```
<?
    $ID = 548;
    if (isset($ID))
    {
        echo( "ProductID ".$ID );
    }
    unset( $ID ); // Kintmojo apibrėžtis panaikinama
    if (isset($ID)) // false
    {
        echo( "šito nespausdins" );
    }
?>
```

empty() -Funkcija empty() panaši į isset(). Ji gražina true, jei kintamasis neapibrėžtas, yra 0 (0.00) arba tuščia ilutė "". Kitu atveju gražina false.

```
<?
    empty($new); // true
```

```

$new = 1;                                empty($new); // false
$new = 0;
empty($new); // true
$new = "";
empty($new); // true
$new = "laba diena";
empty($new); // false
unset($new);
empty($new); // true

```

?>

is..() -Funkcijos `is_int()`, `is_integer()`, `is_long()` grąžina true, jei kintamasis yra integer, `is_double()`, `is_float()`, `is_real()` - jei kintamasis - double,

<?

```

if (is_string($value))
{
    echo( "Tai eilute" );
}

```

?>

..val() -Šios funkcijos dar vienu būdu leidžia tiksliai pakeisti kintamojo tipą į integer, double bei string. Pvz:

<?

```

$ProductID = "543AB";
$a = intval($ProductID); // $a = 543;

```

?>

Be to, funkcija `intval()` turi nebūtiną papildomą parametą, leidžiantį nurodyti skaičiavimo sistemą konvertuojant. Pvz:

<?

```

$a = intval( $ProductID, 16 );// $ProductID buvo interpretuotas kaip šešioliktainis skaičius

```

?>

2.2.10 Kintamieji iš HTML formos

Kaip jau sužinojote, PHP programoje galima apsibrėžti ir naudoti įvairaus tipo kintamuosius. Tačiau be to yra dar kitas būdas kintamiesiems apibrėžti. Kintamieji, gaunami iš HTML formų, yra sukuriami automatiškai programos vykdymo pradžioje. Jie atlieka labai svarbų vaidmenį, nes perduoda vartotojo įvestą informaciją. Iliustruosime pavyzdžiu.

```

<HTML>
<BODY>
  <FONT FACE="Verdana">
  <FORM ACTION="go.php" METHOD="post" NAME="goapp">
    <H3>U_pildykite šia forma:</H3>
    <TABLE BORDERCOLOR="#C0C0C0" BORDER="1" CELLPADDING="5" CELLSPACI
NG="0">
  <TR>
    <TD><STRONG>Vardas, pavarde:</STRONG></TD>
    <TD><INPUT TYPE="text" SIZE="20" MAXLENGTH="50" NAME="vardas"></TD>
  </TR>
  <TR>
    <TD><STRONG>Miestas:</STRONG></TD>
    <TD>
      <SELECT NAME="miestas" SIZE="1" >
        <OPTION VALUE="vl" SELECTED>Vilnius</OPTION>
        <OPTION VALUE="kn">Kaunas</OPTION>
        <OPTION VALUE="kl">Klaipeda</OPTION>
      </SELECT>
    </TD>
  </TR>
  <TR>
    <TD><STRONG>ADRESAS:</STRONG></TD>
    <TD><INPUT TYPE="text" SIZE="20" MAXLENGTH="50" NAME="adresas"></TD>
  </TR>
  <TR>
    <TD><STRONG>Telefonas:</strong></TD>
    <TD><INPUT TYPE="text" SIZE="20" MAXLENGTH="15" NAME="telefonas"></TD>
  </TR>
  <TR>
    <TD><STRONG>Email adresasa:</STRONG></TD>
    <TD><INPUT TYPE="text" NAME="emailas" SIZE="20" MAXLENGTH="50"></TD>
  </TR>
  <TR>
    <TD><INPUT TYPE="checkbox" NAME="moketi" CHECKED><STRONG>Moketi
    pristacius</STRONG></TD>
    <TD><INPUT TYPE="submit" NAME="go" VALUE="Siusti">
    <INPUT TYPE="Reset" VALUE="Trinti"></TD>
  </tr>
  </TABLE>
</FORM>
</FONT>

</BODY>
</HTML>
?>

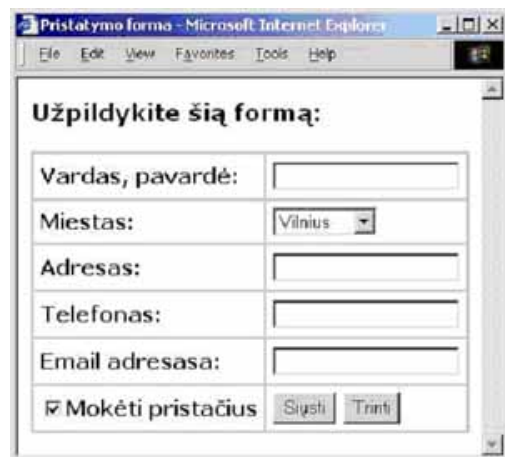
```

Kaip matome, turme formą goapp, kuri turi tokius elementus:

```

<?
<INPUT NAME="vardas"    TYPE="text" SI
ZE="20" MAXLENGTH="50">
<SELECT NAME="miestas"    SIZE="1">

```



1 pav. Forma goapp.

```

<INPUT NAME="adresas" TYPE="text" SIZE="20" MAXLENGTH="50">
<INPUT NAME="telefonas" TYPE="text" SIZE="20" MAXLENGTH="15" >
<INPUT NAME="emailas" TYPE="text" SIZE="20" MAXLENGTH="50">
<INPUT NAME="moketi" TYPE="checkbox" CHECKED>
<INPUT NAME="go" TYPE="submit" VALUE="Siusti">
<INPUT TYPE="Reset" VALUE="Trinti">
?>

```

Kaip matote, kiekvienas formos elementas turi savybę NAME, vadinasi, formos programos kode go.php automatiškai bus sukurti tokie kintamieji: \$vardas, \$miestas, \$adresas, \$telefonas, \$emailas, \$moketi, \$go. Visų jų reikšmės, išskyrus \$go, bus nustatomos vartotojo. Simple :).

3. VADYBOS MOKYMO CENTRO VEIKLOS PROCESO MODELIAI

3.1. VADYBOS MOKYMO CENTRO veiklos aprašymas.

VADYBOS MOKYMO CENTRO (VMC) misija - tenkinti verslo visuomenės poreikius, susijusius su verslo valdymo ir organizavimo tobulinimu, ekonominiu švietimu, skleisti pažangią užsienio patirtį ir naujas technologijas siekiant socialinės ir ekonominės pažangos, skatinti pilietinės visuomenės ir laisvos rinkos ekonomikos kūrimą. Kauno technologijos universiteto Panevėžio filialo Vadybos technologijos ir verslo mokymo centras (VMC) buvo įsteigtas 1995 m. įgyvendinant Jungtinių Valstijų Tarptautinės plėtros agentūros projektą "Vadybos mokymas ir ekonominis švietimas Centrinei ir Rytų Europai.

3.1.1. Organizacijos veiklos produktas yra suorganizuoti mokymo kursai įmonėms, organizacijoms bei pavienių asmenų grupėms.

3.1.2. Pagrindinės valdymo funkcijos kompanijoje VMC yra:

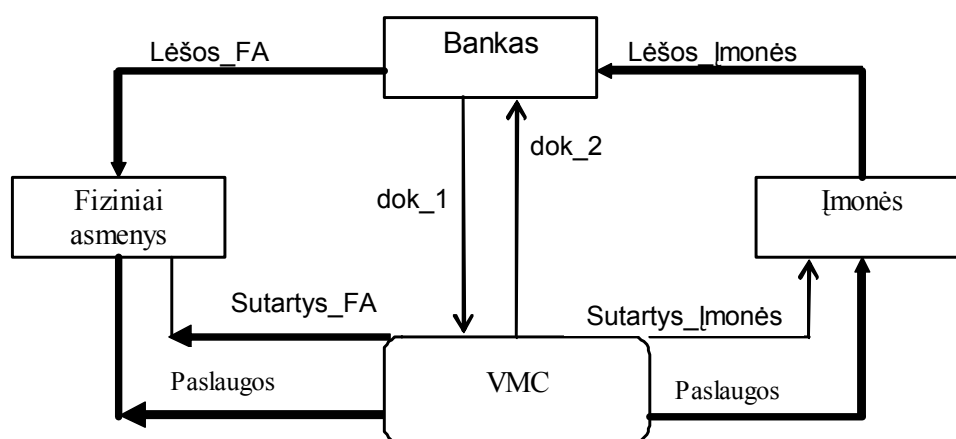
- *Bendras valdymas.* Ši valdymo funkcija:
 - Duoda nurodymus marketingo ir reklamos, programų rengimo, finansų valdymo skyriams.
 - Gauna išsamias ataskaitas ir kitą reikalingą informaciją iš visų skyrių.
 - Gauna marketingo tyrimų rezultatus iš marketingo ir reklamos skyriaus ir teikia pasiūlymus kartu su nurodymais kitiems skyriams.
- *Finansų valdymas:*
 - Kontroliuoja pinigų srautus.

- Tvarko buhalterinę apskaitą
- Teikia ataskaitas bendro valdymo skyriui.
- *Programų rengimo skyrius:*
 - Gauna nurodymus iš bendro valdymo skyriaus.
 - Siunčia ataskaitas finansų valdymo skyriui.
 - Kuria mokymo programas.
 - Rūpinasi dalyvavimu tarptautiniuose projektuose.
 - Teikia ataskaitas finansų ir bendro valdymo skyriams.
- *Marketingo ir reklamos skyrius:*
 - Tiria mokymo programų paklausą.
 - Rūpinasi organizuojamų mokymo kursų dalyviais (renka apie juos informaciją, siunčia pasiūlymus, pasirašinėja sutartis).
 - Rūpinasi tiesioginėmis ir masinėmis mokymo programų reklamomis.
 - Teikia ataskaitas finansų ir bendro valdymo skyriams.

3.2. Duomenų srautų diagramos

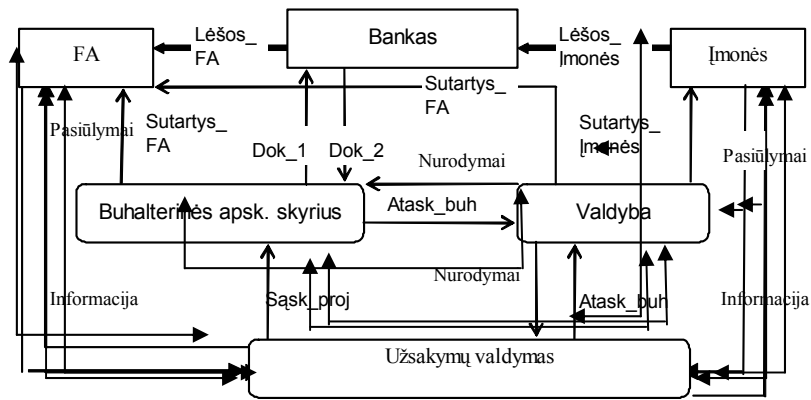
Organizacijos veiklą aprašo grafinės schemos, kuriose yra pavaizduojami materialūs ir informacijos srautai [8]. Kadangi yra kuriama informacinė sistema tai modeliuose bus naudojami tik informaciniai duomenų srautai, tai yra sudaromos duomenų srautų diagramos (DFD data flow diagrams). Modelis prasideda nuo aukščiausio lygio kontekstinės diagramos. Šioje diagramoje organizacija apibrėžiama kaip vienas procesas.

Tokia **aukščiausio lygio** “Vadybos mokymo centro” DFD yra pateikta žemiau.



2 pav. VMC aukščiausio lygio duomenų srautų diagrama

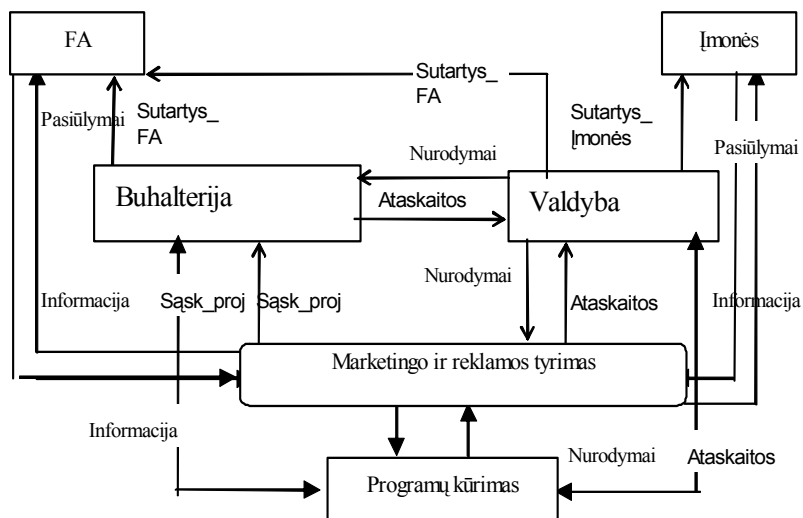
Nulinio lygio duomenų srautų diagramoje yra pateikti duomenų judėjimo srautai įmonės



3 pav. VMC nulinio lygio duomenų srautų diagrama

viduje.

Pirmo lygio:

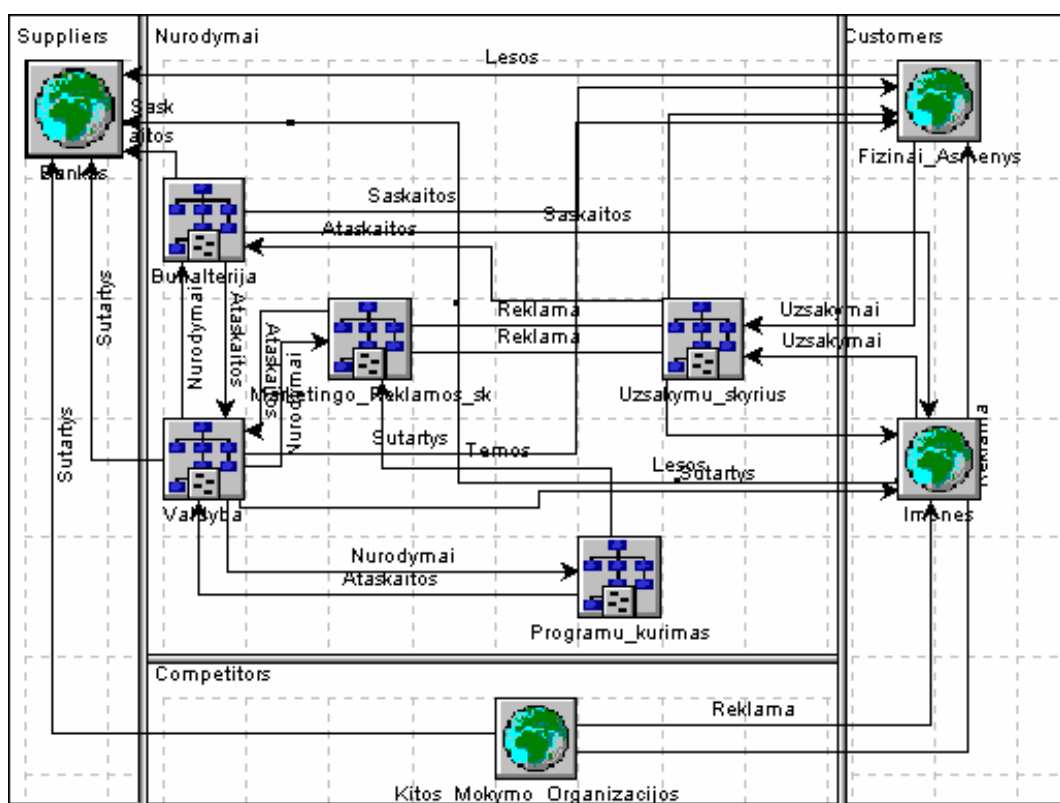


4 pav. VMC pirmo lygio duomenų srautų diagrama

3.3. Provision Workbench modeliai

3.3.1. Biznio sąveikų modelis

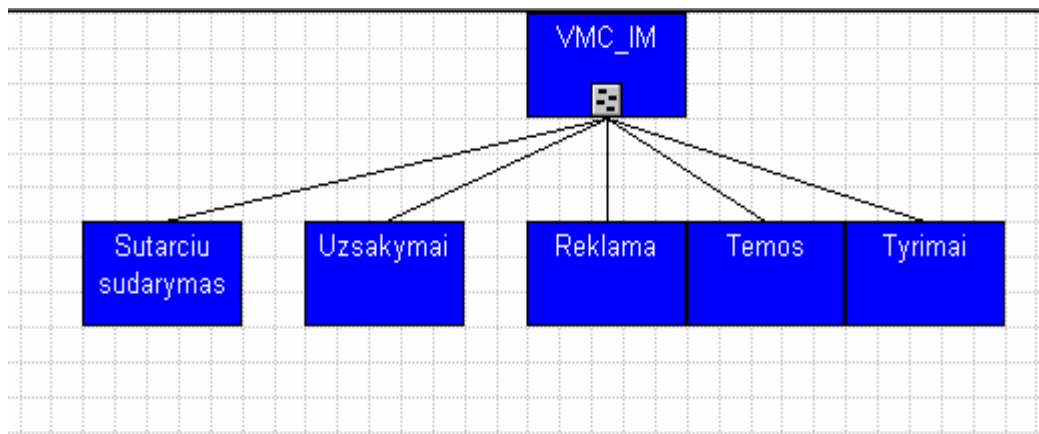
Biznio sąveikų modelis (BIM - business interaction model) atvaizduoja vartotojo biznio veiklą strateginėje perspektyvoje, parodo sąveiką tarp vidinių organizacijos objektų ir išorinių organizacijų. Šiame modelyje yra modeliuojamos ir įvertinamos organizacijos objektų sąveikos ir ryšiai su vartotojais, tiekėjais ir konkurentais. BIM nėra koncentruotas į organizacinių vienetų apibrėžimą, bet greičiau į ryšius ir informacinius bei materialinius persiuntimus tarp organizacijų. 5 Pav. Sukurtas VMC biznio sąveikų modelis.



5 pav. VMC biznio sąveikų modelis

3.3.2. Įvykių modelis

Įvykių modelis (event model) leidžia išvystyti, išdėstyti įvykių hierarchiją, kuri rodo vykstančius, atsirandančius įvykius biznio sferoje. Įvykis šiame modelyje įpareigoja biznio sferą inicijuoti tam tikrą funkciją. Įvykis gali sąveikauti ir su kitais išoriniais biznio objektais. Visi įvykiai yra surišti tėvo/vaiko jungtimi (parent/child link). 6 pav. Sukurtas VMC įvykių modelis.

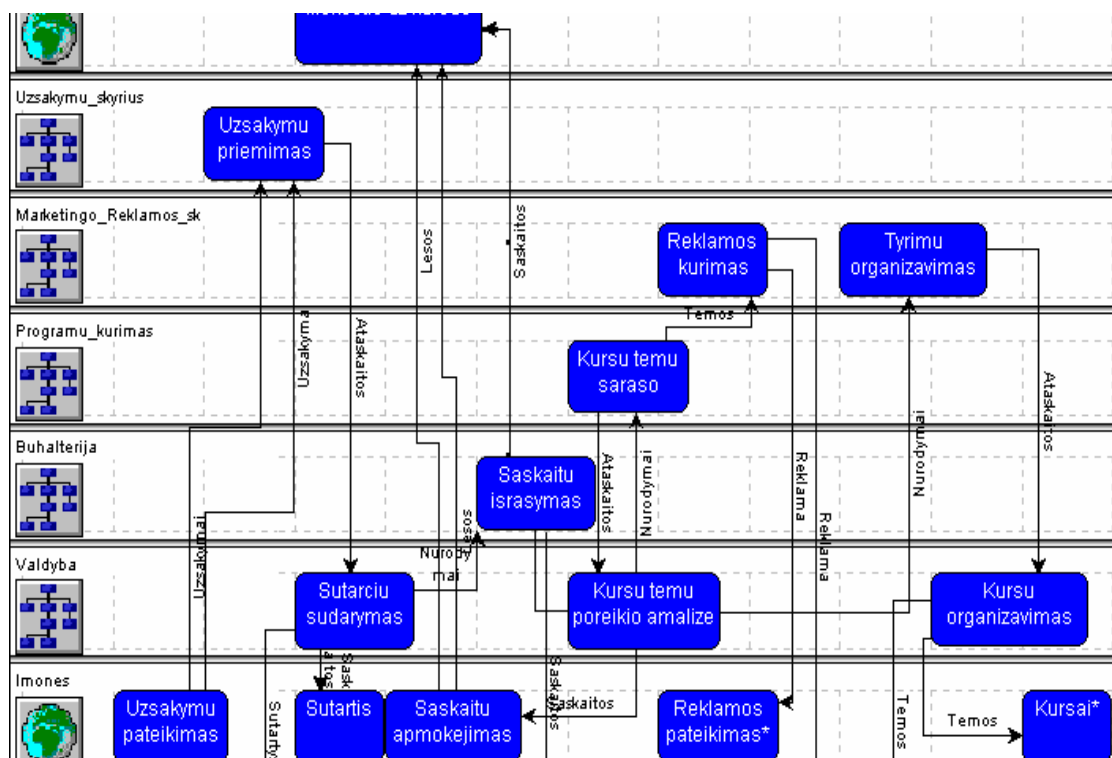


6 pav. VMC įvykių modelis

3.3.3. Darbų sekos modelis

Darbų sekos modelis (workflow model) yra kuriamas per BIM modelį, nes į šį modelį yra įtraukiamos pagrindinės organizacinės struktūros, kurios atlieka tam tikrą darbų seką. Naudojant darbų sekos modeliavimą leidžiama sukurti labiau detalesnį veiklų modelį, kuris apima vartotojo biznio procesą.

Šis modelis atvaizduoja biznio procesus išreiškiant juos veiklos komponentais ir darbų seka tarp tų veiklų. Darbų sekos modelis koncentruotas į darbų seką nuo biznio pradžios iki galo. Tai atspindi pagrindinis aukščiausio lygio modelis. Jei koks nors procesas ar darbas yra sudėtinis kitų darbų atžvilgiu, tai šiam procesui yra sukuriamas detalesnis žemesnio lygio darbų sekos modelis. 7 pav. Sukurtas VMC Darbų sekos modelis, pirmasis lygmuo.

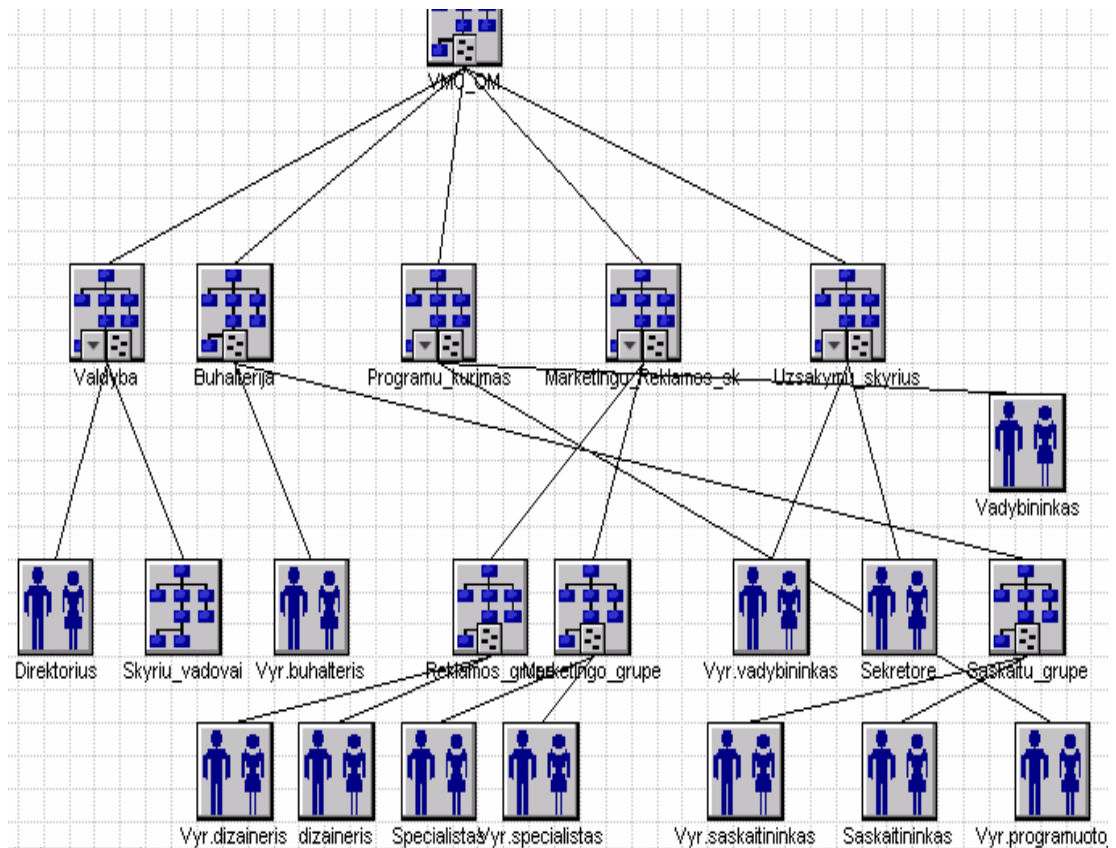


7 pav. VMCdarbų sekos modelis, pirmasis lygmuo

3.3.4. Organizacijos modelis

Organizacijos modelis (organization model) parodo organizacijos hierarchinę struktūrą. Organizacija – tai grupė žmonių, kuri organizuoja biznio veiklą arba sąveikauja su ja. Specifinės organizacijos arba padaliniai įeinantys į pagrindines organizacijas sudaro žemesnius lygius.

Šiame modelyje rinkos esybė suprantama kaip išorinė esybė, kuri sąveikauja su tiekėjais, vartotojais ar kitomis išorinėmis organizacijomis. Organizacijos esybė atitinka valstybinę struktūrą biznio veiklos viduje. Pateikto pavyzdžio modelyje (8 pav.), aukščiausio hierarchinio lygio organizacijos esybė yra „Vadybos mokymo centras“, žemesnio lygio – valdybos, užsakymų skyriaus ir kt. esybės, kurios hierarchiškai išsidėsčiusios organizacijoje. Rolės esybė atitinka darbus ar funkcijas atliekamas biznio veiklos viduje.

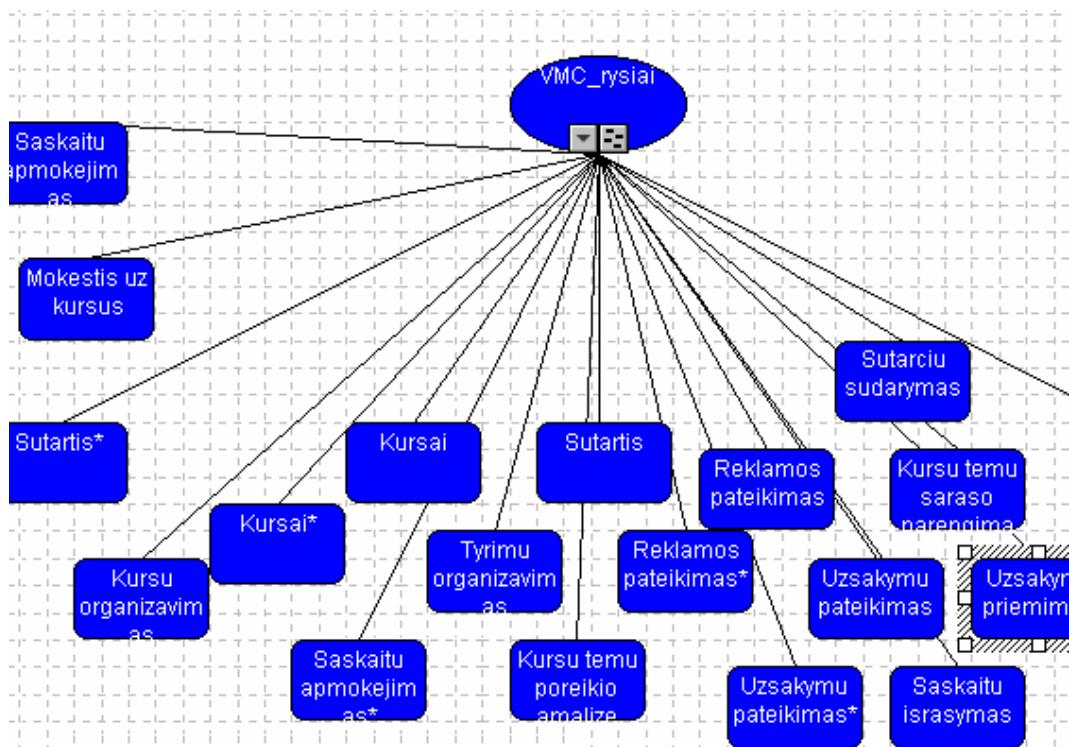


8 pav VMC organizacijos modelis

3.3.5. Procesų modelis

Biznio sfera užima aukščiausią lygį modelyje, kuri gali būti sudaryta iš specifinių žemesnių lygių biznio sferų. Biznio sferos gali turėti biznio procesus vaikus, kurie patys gali turėti vaikus komponentus. Kadangi procesų modelis yra kuriamas ir iš darbų sekos modelio, tai į modelį yra įtraukiami darbų sekos modelio darbai kaip žemesnio lygio hierarchiniai procesai. Ryšys tarp aukštesnio ir žemesnio lygio procesų ir kitų modelio elementų yra vadinamas tėvo/vaiko jungtimi (parent/child link).

Pateiktame modelio pavyzdyje (9 pav.) aukščiausio hierarchinio lygio biznio sfera yra gamybinės organizacijos ryšiai, kuri yra sudaryta iš „Vadybos mokymo centro“ veiklos proceso, kuris savo ruožtu susideda iš daugelio veiklų-darbų.



9 pav. VMC procesų modelis

4. REIKALAVIMŲ SPECIFIKAVIMAS

4.1. Reikalavimų specifikavimo dokumentas

VADYBOS MOKYMO CENTRO (VMC) misija – tenkinti verslo visuomenės poreikius, susijusius su verslo valdymo ir organizavimo tobulinimu, ekonominiu švietimu, skleisti pažangią užsienio patirtį ir naujas technologijas siekiant socialinės ir ekonominės pažangos, skatinti pilietinės visuomenės ir laisvos rinkos ekonomikos kūrimą.

4.1.1. Dokumento tikslas

Pateikti reikalavimus informacinei sistemai, diegiamai verslo mokymo centre (VMC) pagal jo poreikius.

4.1.2. Dalyvių ir tyrimo srities savybės

Su šia informacijos sistema turėtų dirbti fimos vadybininkai ir marketingo ir reklamos specialistai, programų kūrimo specialistai.. Tiriama sritis bus mokymo kursų įmonėms, organizacijoms bei pavienių asmenų grupėms organizavimas, vykdymas, reklama.

4.1.3. Kuriamo produkto savybės

Kuriama padalinio IS turi dirbti greitai (neturėtų reikėti laukti duomenų, suvestinių ar analizių), turėtų sietis su kitais organizacijos padaliniais (finansų, programų kūrimo ir marketingo – reklamos ir t.t.)

4.1.4. Kompiuterizuojamos organizacijos apibūdinimas

Kauno technologijos universiteto Panevėžio filialo Vadybos technologijos ir verslo mokymo centras (VMC) buvo įsteigtas 1995 m. įgyvendinant Jungtinių Valstijų Tarptautinės plėtros agentūros projektą „Vadybos mokymas ir ekonominis švietimas Centrinei ir Rytų Europai.

4.2. Bendri reikalavimai

4.2.1. Produkto apibrėžimas

Numatyta kompiuterizuoti *Programų rengimo* ir *Marketingo ir reklamos skyrių* veiklos sritis (valdymo funkcijas). Bus sukurta Informacinė sistema apie VMC-e rengiamas mokymo programas ir projektus su klientų duomenų baze.

4.2.2. Analogiškų sistemų apibūdinimas

Informacinė sistema, be abejo gali turėti analogų, kadangi tokie poreikiai kurie kilo šiai akcinei bendrovei, galėjo kilti anksčiau kokiai nors kitai įmonei.

4.2.3. Vartotojų apibūdinimas

Kuriamos IS vartotojai bus visi darbuotojai, kurie tiesiogiai dalyvaus kursų organizavimo procese. T.y. ja naudosis tie, kurie tiesiogiai galės perduoti kintančius duomenis (valdymo, finansų skyrių, programų kūrimo ir marketingo-reklamos skyrių darbuotojai).

4.2.4. Vartotojų problemų apibūdinimas

Duomenys apie klientus kaupiami ir saugomi „popieriniame“ variante. Todėl duomenų apibendrinimas ir jų analizė reikalauja nemažai laiko sąnaudų. Daug techninio darbo.

4.2.5. Vartotojų tikslai ir reikalavimai

Vartotojų tikslas yra informacijos operatyvumas ir visapusiškumas, laiko sąnaudų mažinimas, bei lengvai įsisavinamas IS naudojimas. Darbui su IS turi pakakti pradinių darbo su kompiuterių įgudžių ir nereikalauti specifinių žinių.

4.3. Funkciniai reikalavimai

4.3.1. Informacijos perdavimas iš vieno etapo į kitą vietinio tinklo ir duomenų bazės pagalba.

4.3.1.1. Reikalavimo apašymas:

Kiekvienas skyrius duomenų bazėje turi savo atskiras sritis kuriose laikoma jų sukurta ir gaunama informacija. Vieta gaunamai informacijai yra skirta gaunamiems nurodymams arba

kitai informacijai reikalingai darbo procese. Kiekvienas skyrius duomenų bazėje gali turėti ryšį tik su tuo skyriumi su kuriuo tiesiogiai keičiasi informacija t.y. informacijos srautai yra griežtai nustatyti ir kontroliuojami. Per vietinį tinklą gali susijungti bet kurie du ar daugiau organizacijos kompiuterių. Duomenų bazės interfeisas yra vienodas visiems skyriams, bet galimybės ir pateikiamos informacijos pobūdis yra nustatytas t.y. skyrius įsijungęs į duomenų bazę mato tik jam skirtą informaciją, o siūsti ją gali tik iš sąrašo parinktam skyriui kuris yra nustatytas konkrečiai kiekvienam skyriui.

4.3.1.2. *Reikalavimo įtaka visai sistemai:*

Šis reikalavimas yra priešastinis sistemai kurti.

4.3.1.3. *Techniniai klausimai:*

Pakankamas tinklo ir duomenų bazės greitis, duomenų bazės valdymo paprastumas

4.3.1.4. *Kaina ir darbo grafikas:*

Kaina turi būti kuo mažesnė, kad kuo greičiau įmonei atsipirktų ir toliau padėtų mažinti išlaidas.

4.3.1.5. *Rizika:*

Klaidos ar informacijos praradimas duomenų bazėje yra neleistinas.

4.3.1.6. *Sąryšis su kitais reikalavimais:*

Yra susiję su veiklos valdymo reikalavimais.

4.3.2. *Bendra firmos IS .*

4.3.2.1. Turi apjungti padalinio skyrių IS'as, bei pateikti visapusišką analizę.

4.3.2.2. Ja naudojasi padalinio vadovas, atsižvelgdamas į analizės pateikia skyriams uždavinius.

Taip pat suvestinėmis ir analizėmis (ataskaitomis) gali naudotis skyrių IS vartotojai, bet nekeisti duomenų.

4.4. Reikalavimai sąsajai

4.4.1. Vartotojo sąsajos:

Duomenų bazė turi specialią vartotojo sąsają. Vietinis tinklas turi standartinę Windows sąsają.

4.4.2. Grafinė vartotojo sąsaja (menu scenarijus, langai):

Vartotojo sąsaja realizuota mygtukų, langų, menu sąsaja. Menu punktai skirti duomenų bazės atvaizdavimui, informacijos pateikimo formatu koreguoti. Langai yra prilausomai nuo informacijos tipo: grafinės, tekstinės ar informaciniai/perspėjamieji. Mygtukai skirti duomenų srautų valdymui: priimti, siūsti, trinti, įtraukti, perduoti, koreguoti, rūšiuoti, patvirtinti, atšaukti.

4.4.3. Komandinės eilutės sąsaja (aprašomos komandos ir argumentai):

Komandinės eilutės yra nenaudojamos.

4.4.4. Taikomųjų programų sąsaja (interfeiso funkcijų, argumentų, grįžties reikšmių aprašymai):

Sąsajas turi su Microsoft Word, Excel Access programomis.

4.4.5. Diagnostika (klaidų pranešimų aprašymai):

Aptikus klaidą sistemoje ar atlikus klaidingą veiksmą stabdomi veiksmai ir išmetamas pranešimas apie klaidą priklausomai nuo jos pobūdžio.

4.4.6. Komunikacinės sąsajos (tinklo sąsajų aprašymas):

Į žiedinį tinklą yra sujungta 16 kompiuterių, 2 skeneriai, 3 spausdintuvai. Yra ryšys su Internet tinklu.

4.4.7. Programinės įrangos sąsajos (aprašomos visos kitos sąsajos, dar nepaminėtos):

- operacinės sistemos serveryje (viena iš pateiktų): Windows 2000 Server, Windows 2000 Professional arba Windows NT 4.0 Server/Workstation;
- duomenų prieigos priemonė Microsoft Data Access Components MDAC 2.5;

- Microsoft Internet Information Server (IIS) 4.0 (arba aukštesnės versijos);
- naudotojų darbo vietų operacinės sistemos: Microsoft WINDOWS XP/2000/98/95/ME/NT;
- interneto naršyklė Microsoft Internet Explorer nuo versijos 6.0;
- duomenų bazių interfeisai: OLEDB arba ODBC ;
- duomenų bazės: Microsoft SQL Server 7. 0 (2000) arba Microsoft Access⁶ 2000.

4.5 Sistemos darbo reikalavimai (reikalavimai darbo greičiui ir atminčiai).

Visi tinkle dirbantys kompiuteriai dirba su 800 Mhz taktiniu greičiu ir 128 MB RAM.

4.6. Projekto apribojimai.

4.6.1. Standartai (kurių reikia laikytis): tinklo protokolas TCP/IP, realiacinė duomenų bazė.

4.6.2. Apribojimai techninei įrangai: kompiuterio kuriame bus saugoma duomenų bazė HDD 15,3 GB, 128 MB RAM, 500Mhz.

4.6.3. Grafiniai duomenys saugomi JPG formatu.

4.7. Kiti nefunkciniai sistemos atributai .

4.7.1. Saugumas:

Būtina apsauga nuo išorinių įsilaužėlių firewall (“ugnies siena”). Būtinai duomenų kopijų kūrimas.

4.7.2. IS išplėtimo reikalavimai:

Galimybė prijunkti naujus įrenginius ir kompiuterius.

4.7.3. Taikomųjų programų suderinamumas:

Suderinamumas su Microsoft programine įranga.

4.8. Taikomųjų uždavinių modelis

Daugelis veiklos analizės ir IS projektavimo metodų naudoja grafinį modelį *Use Case Model (UCM)*, kurio autorius yra Ivar Jacobson. Šio modelio pavadinimas į lietuvių kalbą verčiamas skirtingai: “panaudojimo atvejų modelis”, “veiklos uždavinių modelis”, “vartotojo

(informacinių) poreikių modelis”, “taikomųjų uždavinių modelis”. Sąvoka šiuo atveju “*use case*” reiškia aktualų veiklos dalyviui (sistemos vartotojui) veiklos procesą (veiklos funkciją ar uždavinį).

Ivar Jacobson apibūdina *Use Case Model* taip: “Sistemos vykdomų transakcijų seka, kurios paskirtis yra pateikti veiklos “dalyviui” pageidaujamą konkretų rezultatą”. Transakcijos (sąveikos) apima informacijos srautų ir (arba) materialių srautų perdavimą veiklos procesui (funkcijai, uždaviniui) arba gavimą iš veiklos proceso (funkcijos, uždavinio).

Veiklos (biznio) uždavinių aprašo veiklos transakcijas, t.y. modeliuoja biznio aplinkos “dalyvių” sąveikas su veiklos srities (veiklos doemeno) vienetais.

Veiklos uždavinių modelis suskirsto veiklos sritį į aibę veiklos uždavinių (žymimi ovalais), nurodo materialias ir informacines sąveikas tarp dalyvių ir veiklos uždavinių. Transakcijos gali būti neįvardintos arba įvardintos.

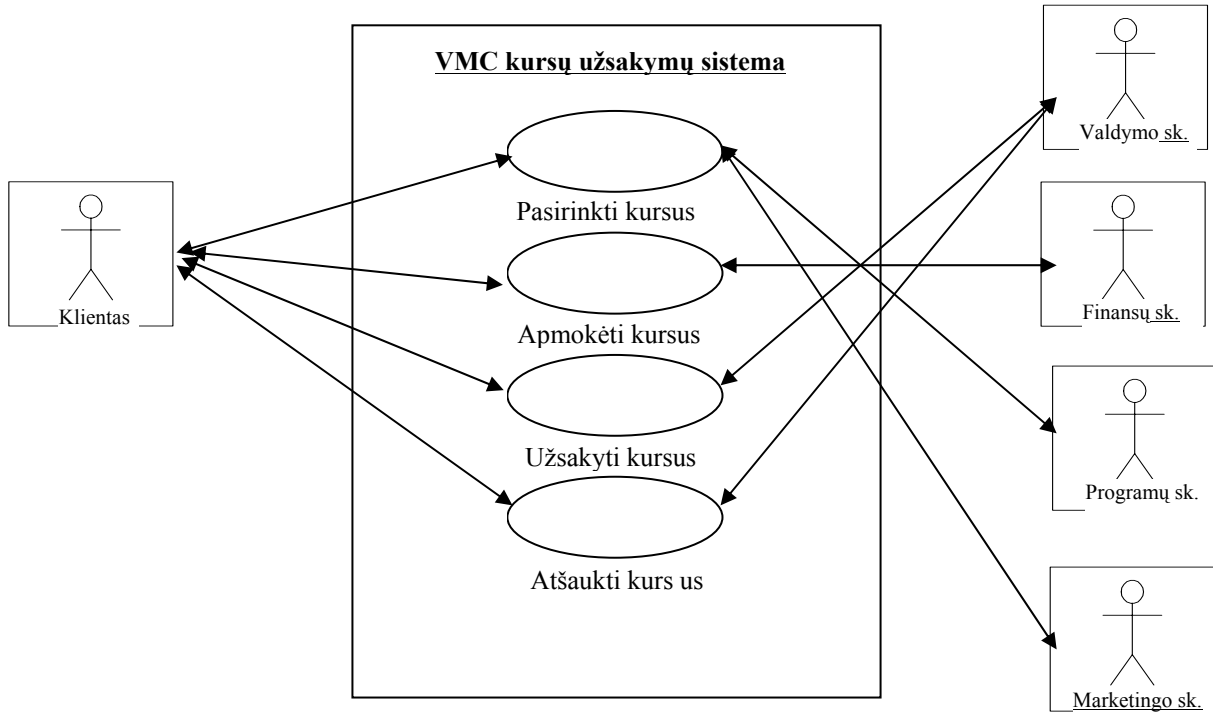
Veiklos uždavinių modelis apibrėžia veiklos srities dalyvius, veiklos procesus, veiklos procesų sąveikas tiek organizacijos viduje, tiek išorėje. Tai padeda išanalizuoti, kaip nagrinėjamos veiklos srities esminiai procesai) yra susiję su išoriniais veiklos objektais, veiklomis ir sistemomis

4.8.1. Veiklos uždavinių modelis.

Veiklos (biznio) uždavinių aprašo veiklos transakcijas, t.y. modeliuoja biznio aplinkos “dalyvių” sąveikas su veiklos srities (veiklos doemno) vienetais.

Veiklos uždavinių modelis suskirsto veiklos sritį į aibę veiklos uždavinių (žymimi ovalais), nurodo materialias ir informacines sąveikas tarp dalyvių ir veiklos uždavinių. Transakcijos gali būti neįvardintos arba įvardintos.

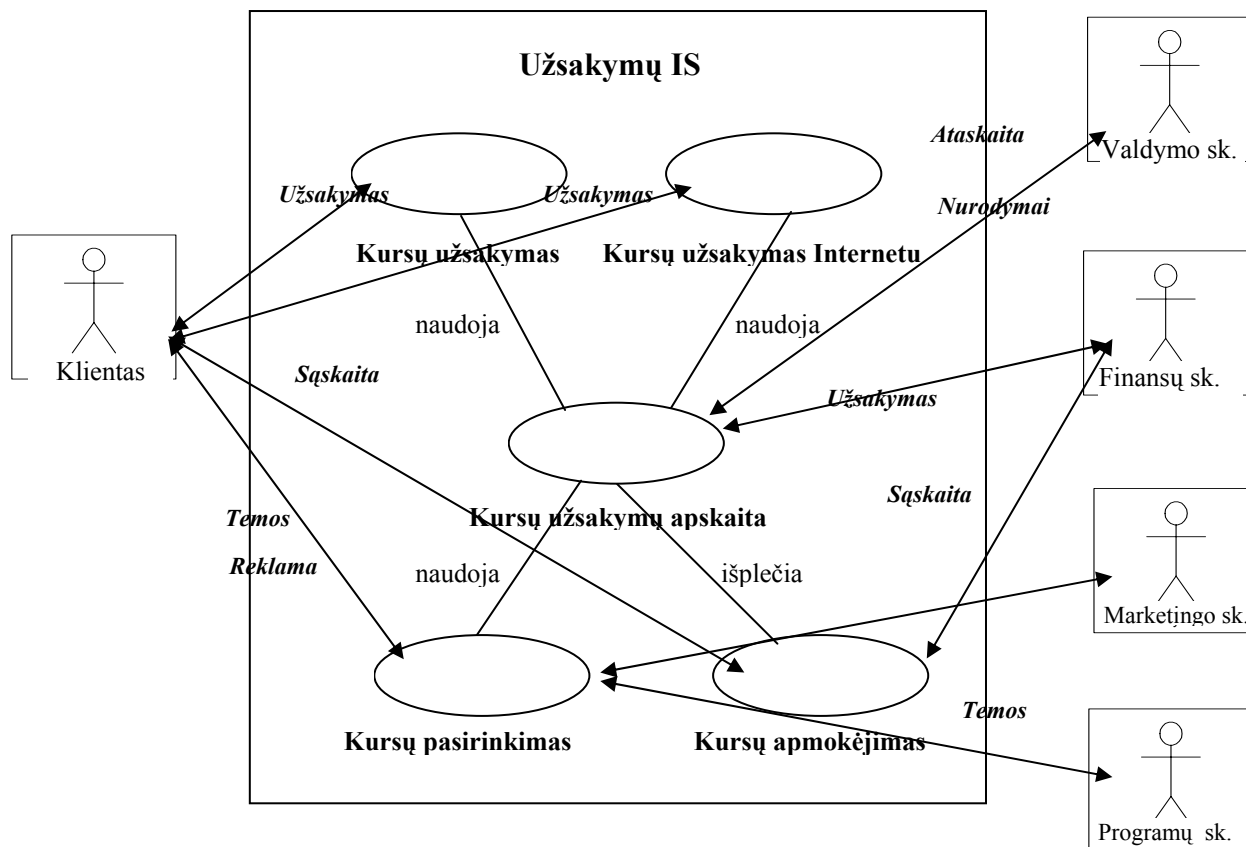
Veiklos uždavinių modelis apibrėžia veiklos srities dalyvius, veiklos procesus, veiklos procesų sąveikas tiek organizacijos viduje, tiek išorėje. Tai padeda išanalizuoti, kaip nagrinėjamos veiklos srities esminiai procesai) yra susiję su išoriniais veiklos objektais, veiklomis ir sistemomis.



10 pav. VMC veiklos uždavinių modelis

4.8.2. Vartotojo informacinių poreikių modelis.

Taikomųjų uždavinių modelis detaliai aprašo ne tik informacijos srautus, bet ir kompiuterizuojamus uždavinius. Taikomųjų uždavinių modeliai sudaromi toliau tikslinant vartotojo informacinių poreikių modelį – detaliai aprašant kompiuterizuojamus uždavinius. Taikomųjų uždavinių modeliai gali būti sudaromi veiklos modelių (pavyzdžiui, darbų sekų modelių (*workflow models*)) pagrindu arba . IS vartotojo interfeiso (sąsajos) projektavimas vyksta taip: analizuojamas kiekvienas UCM (vartotojo informacinių poreikių modelis) ir identifikuojami visi vartotojų ir UCM nurodytų (identifikuotų) taikomųjų uždavinių (IS komponentų) interfeisai.



11 pav. VMC vartotojo informacinių poreikių modelis

4.8.3. Duomenų srautų aprašymas

5 lentelė. Duomenų srautai

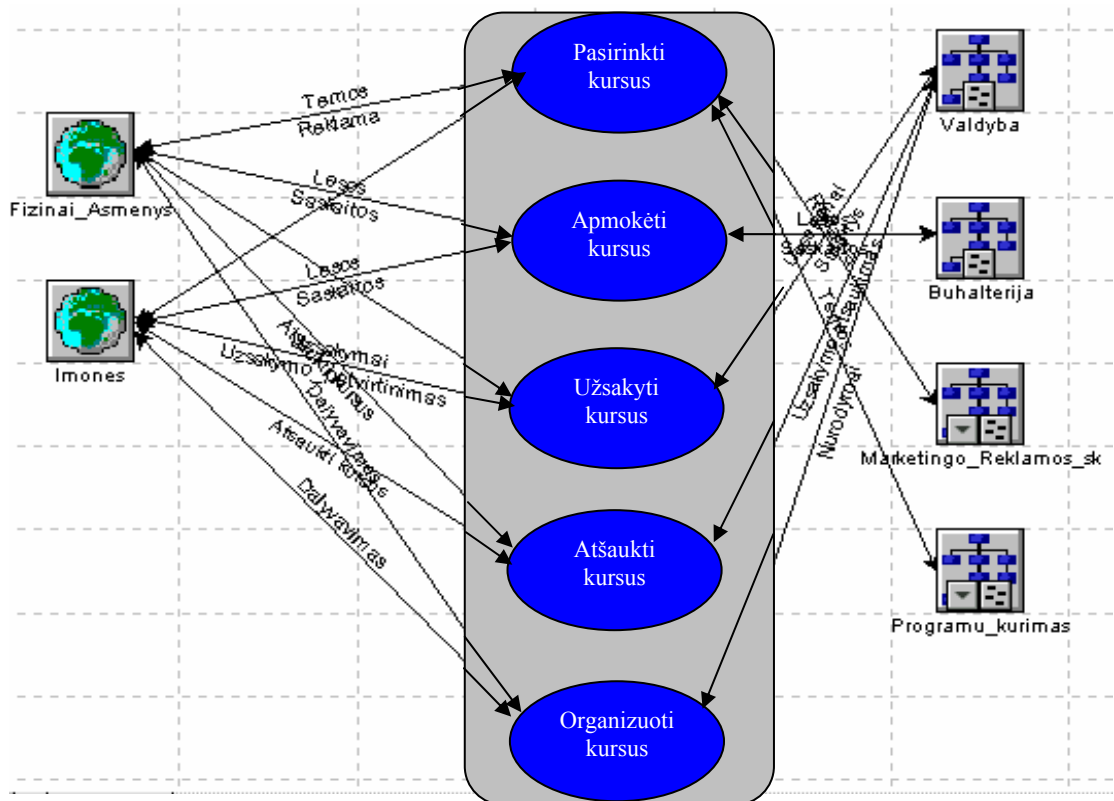
Duomenų srautas	Aprašymas
Nurodymai	Administracija įveda nurodymus, kurie susideda iš tokių atributų: skyriaus pavadinimas, vadovo pavardė, nurodymo turinys, vykdymo terminas, ataskaitos pateikimo data.
Ataskaita	IS pateikia administracijai (pasirinkimas iš sąrašo) ataskaitas apie užsakymo vykdymą. Ataskaitos sudėtis: antraštė (skyriaus pavadinimas, užsakymo gavimo data, užsakymo įvykdymo data, skyriaus vadovo pavardė), lentelė (kursų pavadinimas, kodas, kaina, kursų pravedimo data, apmokėjimo būdas ir terminas).
Temos	Kursų temų sąrašas, lektoriaus pavardė ir trumpa temų santrauka.
Reklama	Kursų temų išplėstas apibūdinimas, temos programa.
Sąskaita	Sąskaita faktūra ar apmokėjimo čekis.
Užsakymas	Užsakovas užpildo užsakymo anketą, kurioje nurodo: pageidaujamo kursų pavadinimą, kodą, kursų pravedimo datą, kursų dalyvių skaičių, atsakingo asmens vardą, pavardę, telefono numerį, adresą.
Užsakymo patvirtinimas	Užsakovas gauna pranešimą (kompiuterio ekrane) apie tai, kad užsakymas priimtas, nepriimtas ar perkeltas vėlesniam laikui.

4.9. Provision Workbench modeliai

4.9.1. Use Case modelis

Biznio panaudojimo atvejų diagrama (use case model) apibrėžia ir analizuoja svarbesnes biznio sferas, biznio procesų ar veiklos sąveikas tiek organizacijos viduje tiek ir išorėje. Tai padeda apžvelgti kaip dauguma biznio zonų yra susijusios su išorinėmis biznio esybėmis, veiklomis ir sistemomis.

Taikomųjų uždavinių modelis (use case model) aprašo kaip vartotojas sąveikauja su biznio funkcija, atlikdamas savo darbą.

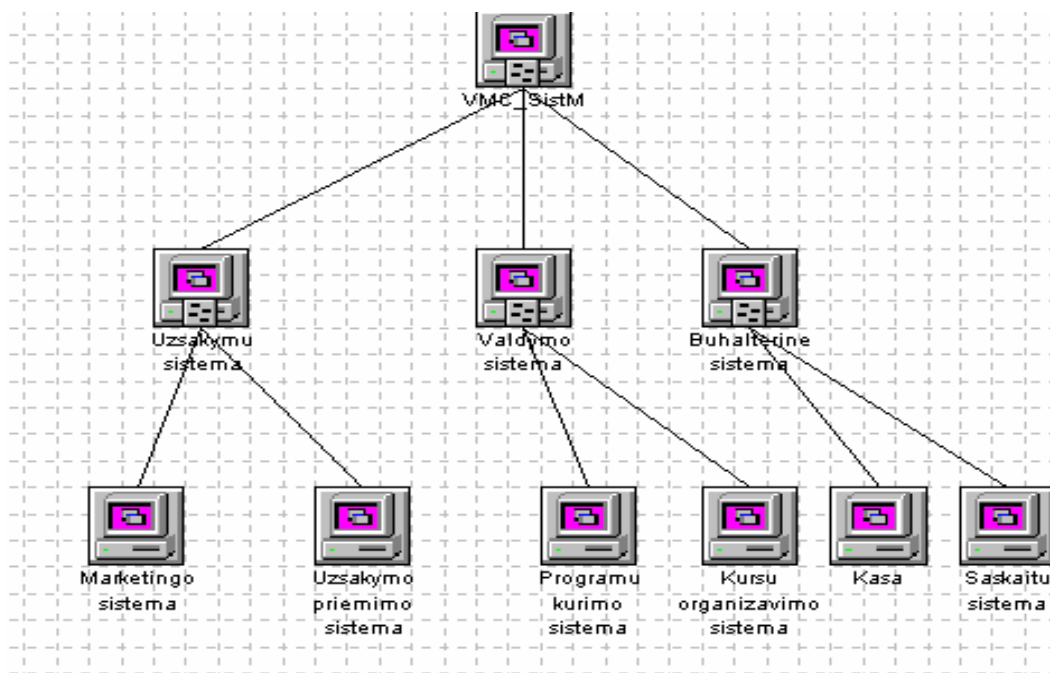


12. Pav. VMC panaudojimo atvejų modelis

4.9.2. Sistemos modelis

Sistemos modelis (system model) atvaizduoja hierarhinę sistemų, kurios yra atviros bizniui, struktūrą. Sistemų objektai gali būti naudojami Darbų sekos modelyje parodyti veikloms, kurios yra automatizuotos ir atliktos sistemos pagalba.

Pagrindinis sistemos modelio elementas yra sistema (system). Aukščiausio hierarchinio lygio sistema susideda iš kitų detalesnių žemesnio lygio sistemų, kurios gali būti toliau detalizuojamos kitomis sistemomis. Skirtingų hierarchinių lygių sistemos sujungiamos tėvo/vaiko jungtimi (parent/child link). Sistemos modelio pavyzdyje (13 pav.) aukščiausio lygio sistema – „Vadybos mokymo centras” pagrindinės sistemos. Kitos žemesnio lygio sistemos yra sudėtinės šios sistemos dalys.



13 pav. VMC sistemos modelis

5. Informacinės sistemos architektūros modeliai

5.1. IS struktūra

Kiekvienoje organizacijoje yra keletas įvairios paskirties IS, kurios kompiuterizuoja atskiras veiklos sritis. Šios atsirado skirtingu laiku, jas projektavo skirtingos projektuotojų grupės ar firmos, nepriklausomai viena nuo kitos. Nauji reikalavimai šioms IS buvo realizuoti kaip papildymai, kuire kažkokiu būdu buvo priderinti prie jau esamų IS realizacijų.

Labai aišku, kad reikalingas vieningas IS planavimo būdas, kuris išspręstų IS kūrimo ir plėtros problemas.

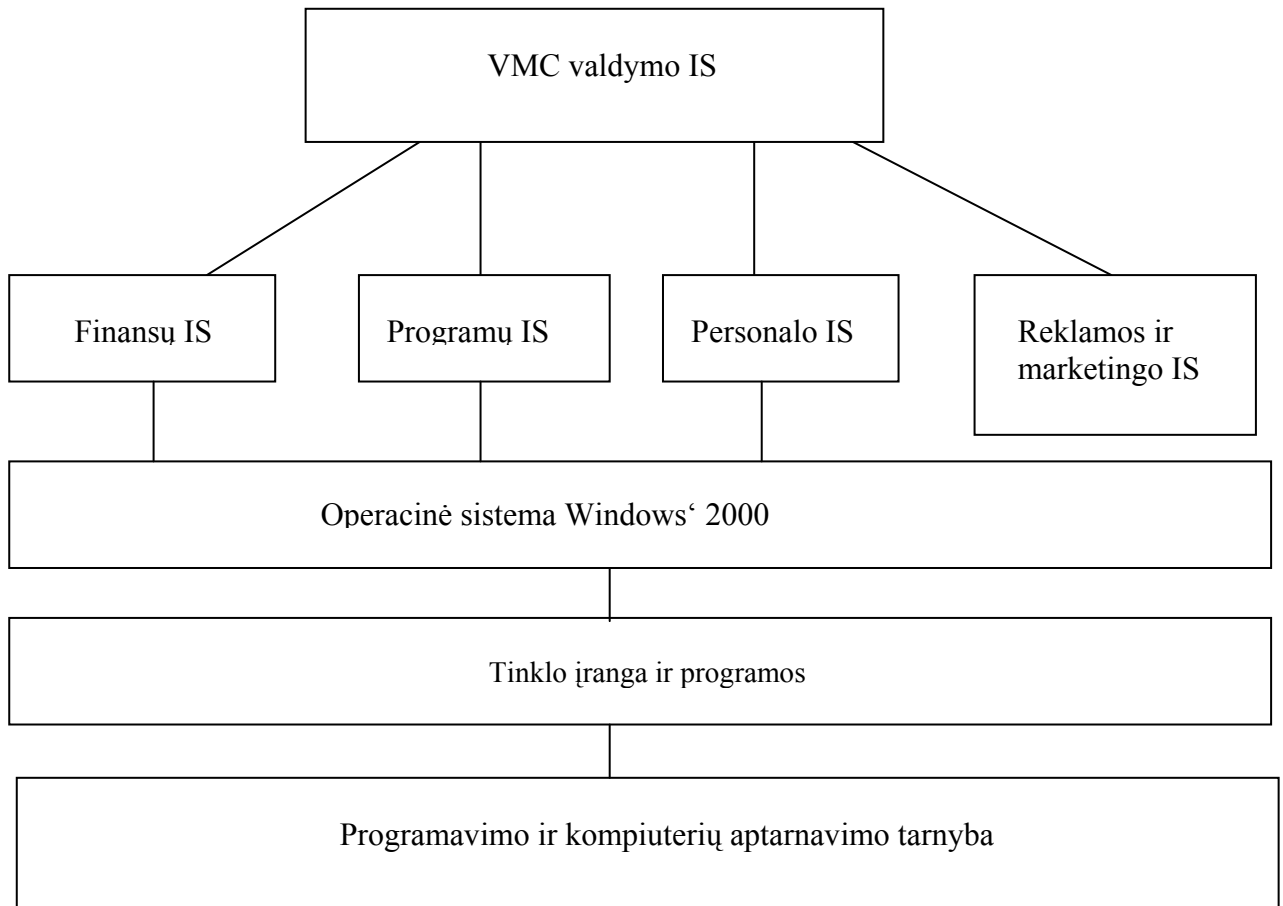
Taigi, informacinė architektūra – pagrindinis organizacijos IT planavimo įrankis. Perspektyvus požiūris yra naudoti informacinės architektūros metodus, kurie įgalina valdyti IS planavimą.

Informacinės architektūros paskirtis – kad kiekvienas veiklos dalyvis – vadybininkas, gamybininkas ar informatikas, IS skyriaus specialistas - turėtų tokį patį supratimą apie veiklą . Tai vienintelis būdas kaip pasiekti, kad veiklą kompiuterizuojanti informacinė technologija (IS duomenų bazė, taikomosios programos, kompiuterinis tinkles, komunikacijos, infrastruktūrta) atitiktų veiklos informacinius poreikius.

Informacinė infrastruktūra susideda iš:

1. Kompiuterinės techninės įrangos ir jų programinė įrangos.
2. Bendros paskirties programinės įrangos.
3. **Duomenų perdavimo ir saugojimo technologinės priemonės** - duomenų bazių valdymo sistemų (DBMS).
4. Kompiuterinių tinklų ir komunikacinių įrengimų.
5. Informacijos vadybos personalo (IS skyriaus personalas).

Kauno technologijos universiteto Panevėžio filialo Vadybos technologijos ir verslo mokymo centras (VMC)_informacinę infrastruktūrą modelį sudaro šie lygmenys:



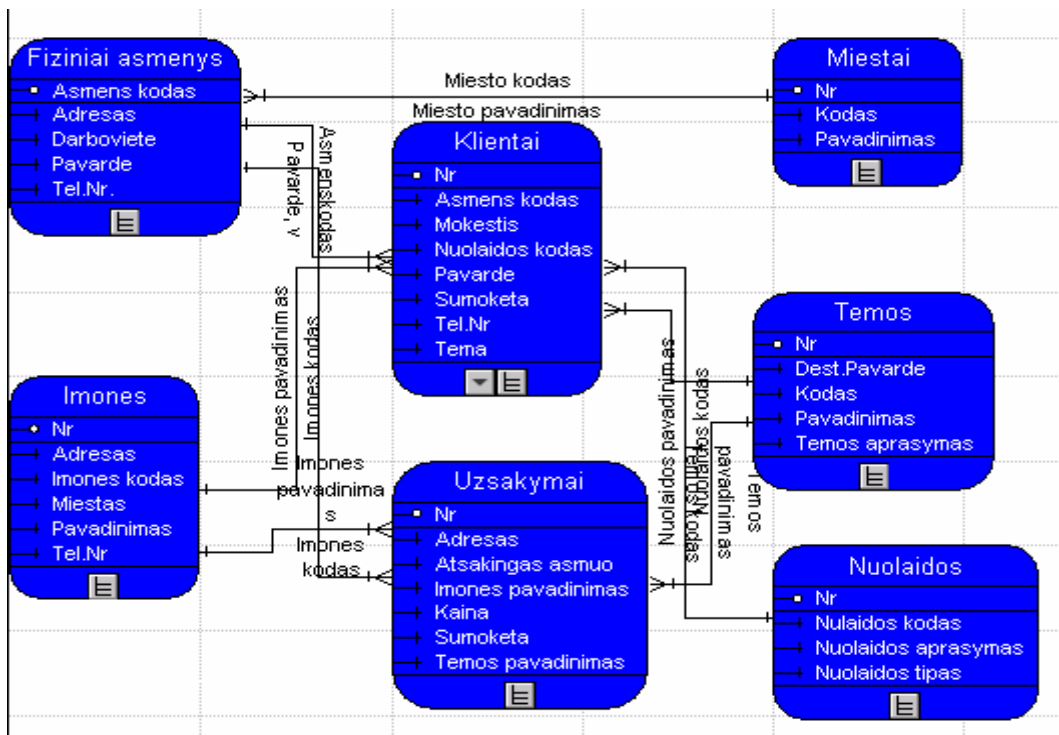
14 pav. VMC IS struktūros schema

5.2. Provision Workbench modeliai

5.2.1. Objektų modelis

Objektų tipai yra toliau apibrėžiami subtipų modelyje (subtype model) esančiuose objektuose, kurie yra specializacijomis (subtipai) arba apibendrinimais (supertipai) priklausomai nuo objekto tipo. Pagrindinis dėmesys šiame modelyje skiriamas objekto tipui. Pagrindiniai elementai naudojami modelyje:

- Objekto tipas (object type) – tai konkretaus objekto atskiras tipas, kuris gali turėti išskirtinius atributus, savybes, kurių neturi kitų tipų objektai.
- Apibendrinimas (generalization) – tai tarpusavio objekto ir jo tipų ryšys, kuris parodo iš kokių tipų susideda objektas. Šis ryšys turi kardinalumą, t.y. į objektą gali sudaryti tiksliai vienas tokio tipo objektas, gali būti arba nebūti subtipu ir gali būti neapibrėžtas ryšys.
- Apibendrinimo jungtis (generalization connector)- naudojama apjungti dviems ar daugiau objektų asociacijų.



15 pav. VMC objektų modelis

5.2.2. Būsenų modelis

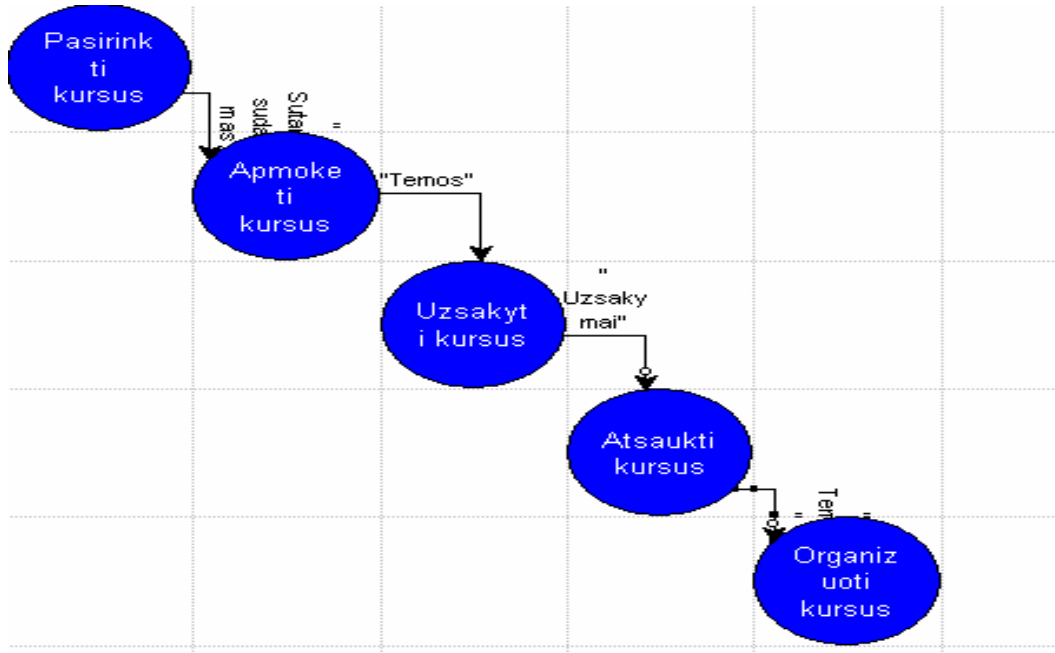
Kadangi kai kurie objektai yra sudėtingi, norint suprasti jų elgseną, sudaromi būsenų modeliai (State model). Jis iliustruoja skirtingų objektų tipų atskirų atvejų gyvavimo ciklą. Šis modelis modeliuoja objekto būsenas ir objekto perėjimus iš vienos būsenos į kitą.

Dėmesys čia skiriamas į objekto būseną, tačiau būseną gali turėti sudėtinių dalių modelį susidedantį iš jos subbūsenų. Subbūseną dar yra suskaidoma į labiau bedras būsenas. Perdavimas, kuris atliekamas tarp superbūsenų (labiau bendrų būsenų) yra sujungtas subbūsenos lygyje.

Pagrindiniai modelio elementai:

- Būseną (state) – tai objekto konkreti būseną
- Perėjimas (transition) – tai ryšys tarp būsenų.
- Perėjimų jungtis (transition connector) – tai jungtis naudojama apjungti dviem ar keliems perėjimams iš skirtingų būsenų į vieną būseną.

Žemiau esančiame 16 pav. pavaizduotas “VMC klientai” objekto būsenų modelis. Čia būsenos eina viena po kitos ir sujungtos perėjimais. Iš kelių būsenų neteisingų paskaičiavimų atvejų yra grįžtama į ankstesnę būseną.



16 pav. VMC būsenų modelis

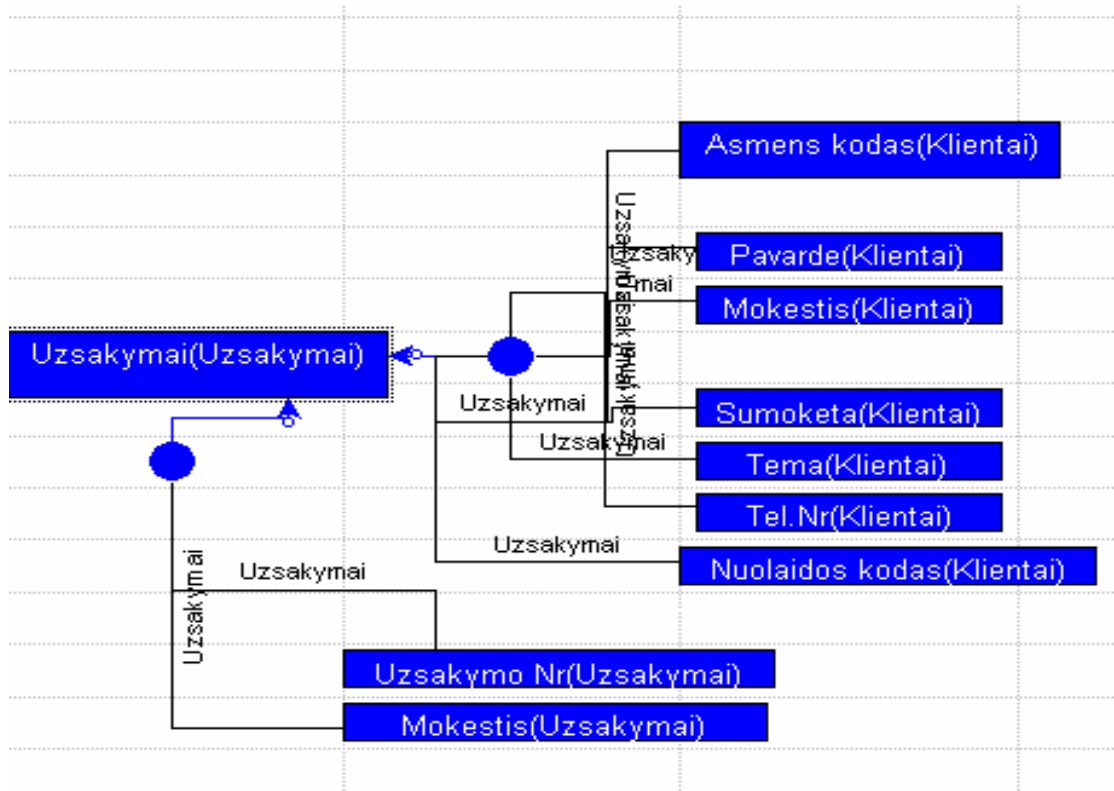
5.2.3. Metodų modelis

Metodų modelis (method model) yra tipiškai išvystytas biznio sferos elgsenos atsakant į įvykį detalizavimui. Pagrindinis dėmesys šiame modelyje yra kreipiamas į metodą, kuris reikalauja rezultato iš kelių įvykių, realizuojamų biznio sferoje. Metodo įvykdymo metu, jis gali reikalauti iš kito metodo, kad tas jam padėtų užbaigti jo funkciją. Tokie reikalavimai pateikiami pranešimų forma, kurie prašinėja metodų jų pačių objektų tipų arba kitų objektų tipų. Tokius reikalavimus metodas gali perduoti kitiems metodams. Tai atskleidžia dinamines metodų sąveikas.

Pagrindiniai modelio elementai:

- Metodas (method) – tai objekto metodas, kuris turi būti atliktas norint įvykdyti kitą metodą, arba pagrindinis metodas, kuris bus atliktas tik tada, kai bus atlikti visi jam priklausantys metodai.
- Pranešimas (message) – tai pranešimas einantis tarp metodų ir perduodantis reikalingus duomenis iš vieno metodo į kitą.

- Pranešimų jungtis (message connector) – tai jungtis naudojama apjungti dviem ar keliems pranešimams parodant, kad aukštesnio lygio metodas priklauso nuo žemesnių apjungtų metodų.



17 pav. VMC metodų modelis

5.3 Projekto testavimas

Provision Workbench priemonėmis buvo atliktas sukurto projekto testavimas. Buvo atlikta: panaudojimo ir darbų sekų modelių testavimas bei BIM modelio testavimas palyginimo metodu. Testavimas parodė, jog esminių klaidų projekte nėra. Testavimo rezultatai pateikti Priede Nr.5.

5.4 Projekto prototipo generavimas

Provision Workbench priemonėmis buvo atliktas sukurto projekto prototipo generavimas. Projekto prototipas generuotas Visual Basic aplinkoje. Gautas rezultatas patvirtino projekto teisingumą ir išryškėjo kai kurios nedidelės klaidos. Ištaisius klaidas rastas lentelėse, toliau buvo nutarta kurti duomenų bazę Access 2000 aplinkoje. Projekto prototipo generavimo rezultatai pateikti Priede Nr.6.

6. Duomenų bazės projektas

6.1. Normalizavimas

Paprasčiausiu atveju DB galėtų sudaryti tik viena lentelė. Tačiau tokiu atveju, kai yra daug lentelės laukų ir didelė duomenų apimtis, atsirastų žymus duomenų pertekliškumas (tokių pačių DE reikšmių pasikartojimas DB), o apdorojant duomenis reikėtų sugaišti nemažai papildomo laiko (pavyzdžiui, pakeičiant DE tą pačią reikšmę keliose DB įrašuose). Norint to išvengti, tenka didelę, sudėtingos struktūros lentelę suskaidyti į keletą mažesnių, paprastesnės struktūros lentelių.

Reliacinio modelio teorinis pagrindas yra reliacinė algebra, kurioje esminis objektas – matematinio santykio (angliškai “relation”) samprata [2].

Reliaciniame modelyje santykis R_i yra domenų aibės D_1, D_2, \dots, D_n dekartinės sandaugos $D_1 \times D_2 \times \dots \times D_n$ poaibis:

$$R_i \subseteq D_1 \times D_2 \times \dots \times D_n;$$

Domenas yra tiesiog reikšmių aibė, kuris siejama su vienu realaus pasaulio objektu (vadinamu esybe). Domenų aibė apibūdina nagrinėjamą probleminę sritį.

Matematinį santykį R_i patogiu vaizduoti dvimatės lentelės pavidalu, kuris yra įprastas ir suprantamas vartotojui.

Lentelę sudaro stulpeliai, vadinami laukais (domenais), ir eilutės, vadinamos įrašais (kortežais)

Lentelių skaidymas į mažesnes, racionalesnes lenteles yra vadinamas normalizavimo procesu.

DB lentelės sudėčiai keliama tokie pagrindiniai reikalavimai:

- a) tarp lentelės laukų neturi egzistuoti nepageidaujamos funkcinės priklausomybės;
- b) lentelės laukų rinkinys turi užtikrinti minimalų duomenų dubliavimą bei jų nesudėtingą apdorojimą ir atnaujinimą.

Šie reikalavimai įvykdomi atlikus DB lentelių normalizavimą.

Normalizuojama keliais žingsniais (etapais), kurių metu duomenų modelis įgyja atitinkamas pateikimo formas, vadinamas normalinėmis formomis (NF).

Išskiriamos trys pagrindinės normalinės formos: pirmoji NF, antroji NF ir trečioji NF. Normalizavimo proceso metu DB lentelės nuosekliai pervedamos iš žemesnės NF į aukštesnę NF.

Lentelė yra pirmosios NF (1NF), jeigu lentelės kiekvienas laukas yra vienareikšmis bei kiekvienas neraktinis laukas funkcionaliai priklauso nuo bet kokio tos lentelės galimo rakto.

Lentelė yra antrosios NF (2NF), jeigu ji yra 1NF ir kiekvienas iš jos neraktinių laukų yra pilnos funkcinės priklausomybės nuo bet kokio tos lentelės galimo rakto.

Lentelė yra trečiosios NF (3NF), jeigu ji yra 2NF ir joje nėra nė vienos neraktinių laukų tranzityvios funkcinės priklausomybės nuo bet kokio tos lentelės galimo rakto.

Normalizavimo išvados:

Galima padaryti išvadą, kad normalizuojant DB, kitaip tariant, pervedant DB lenteles iš vienos normalinės formos į kitą siekiama tokių esminių tikslų [2]:

- išvengti nepageidaujamų funkcinių priklausomybių tarp lentelių laukų,
- kuo labiau sumažinti duomenų perteklišumą,
- supaprastinti duomenų apdorojimą.

Tokiu būdu, norint įdiegti (realizuoti) efektyvią (duomenų apdorojimo prasme) reliacinę DB, reikia kad jos loginio modelio visos lentelės būtų 3NF.

Lentelių apdorojimui būdinga nemaža keblumų, vadinamų anomalijomis:

1. Įvedimo anomalija. Negalima įvesti tam tikrų duomenų tol, kol neįvesta pilna rakto reikšmė (pvz., 18 pav. pateiktos struktūros lentelėje negalima įvesti jokių duomenų apie klientą iki to laiko, kol jis nepristatęs užsakymo).
2. Atnaujinimo (keitimo) anomalija. Reikia didelių papildomų laiko, atminties resursų, atnaujinant (keičiant) lentelėje dažnai pasikartojančias reikšmes ar įvedant lentelėje naujas laukų reikšmes (pvz., norint įvesti papildomus duomenis apie tiekėją reikia papildomų resursų, jei tiekėjo pristatomų gaminių įvairovė didelė). Be to, jei atliktas tik dalinis atnaujinimas, t. y. ne visur DB buvo atnaujintos lauko (ų) reikšmės, tai atsiranda duomenų prieštaravimas, suardantis saugomų DB duomenų vientisumą.
3. Šalinimo anomalija. Šalinant tam tikrus duomenis, gali būti prarasta naudinga informacija, netiesiogiai susijusi su šalinamais duomenimis (pvz., šalinant konkretų klientą (žr. 18 pav.) atsiranda pavojus prarasti informaciją apie duomenis iš jo apie reklamą).

Savo darbe irgi naudosime šią grafinę priklausomybių formą, tik lentelių laukus surašysime į lenteles. Pirminiai lentelių laukai yra pažymėti pastorintu šriftu.

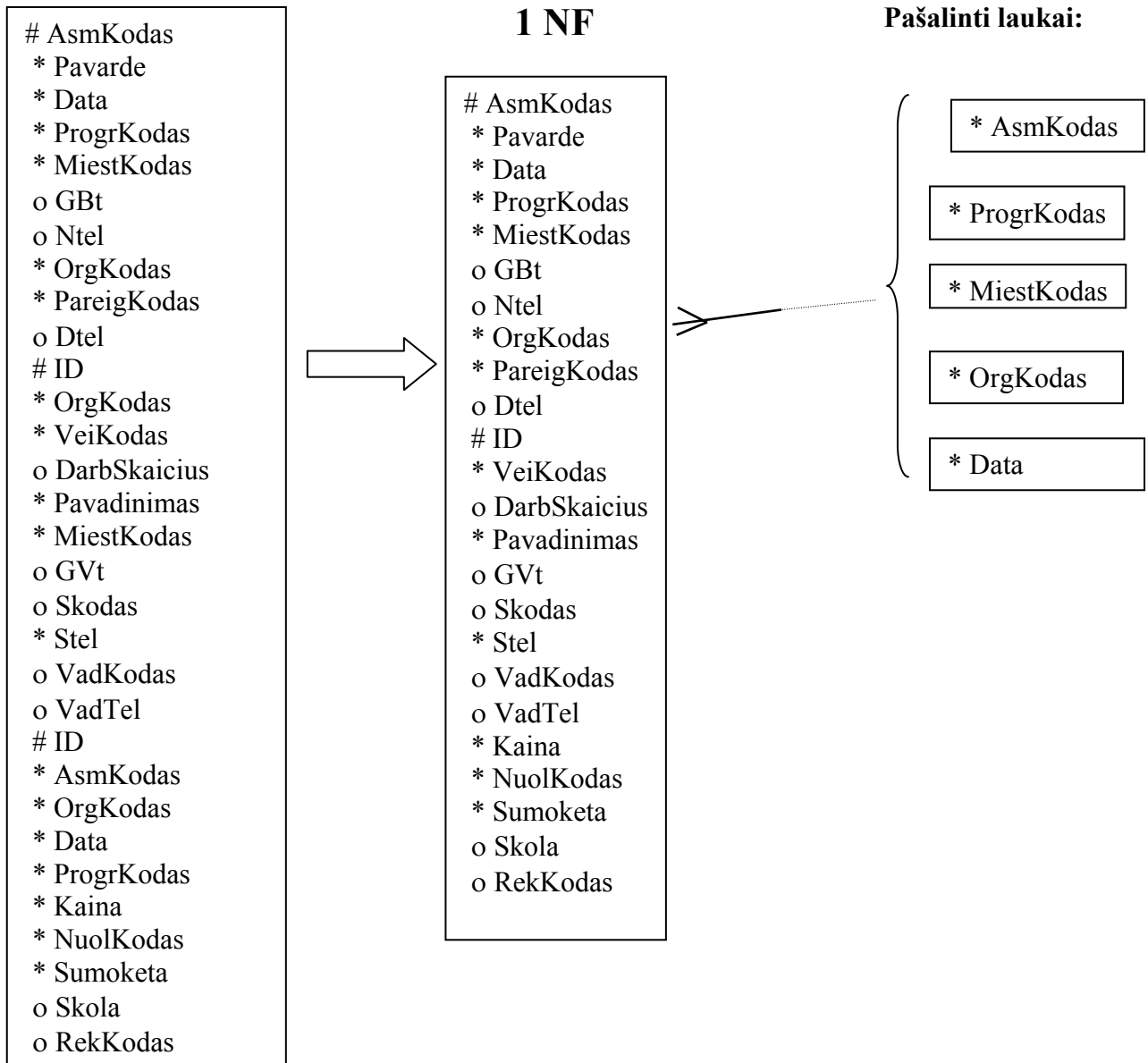
Surašysime trijų pagrindinių lentelių (Asmenys, Įmonės ir Klientai) laukus į lentelę.

Visos kitos DB bazės „VMC klientai“ lentelės yra klasifikatoriai. Pašalinę iš gautos lentelės laukus kurie pasikaroja, gausime pirmąją normalinę formą (1 NF). Naudosime ženklus:

- pažymi identifikuojantį atributą;

*** - pažymi, kad atributo reikšmę būtina žinoti (mandatory);**

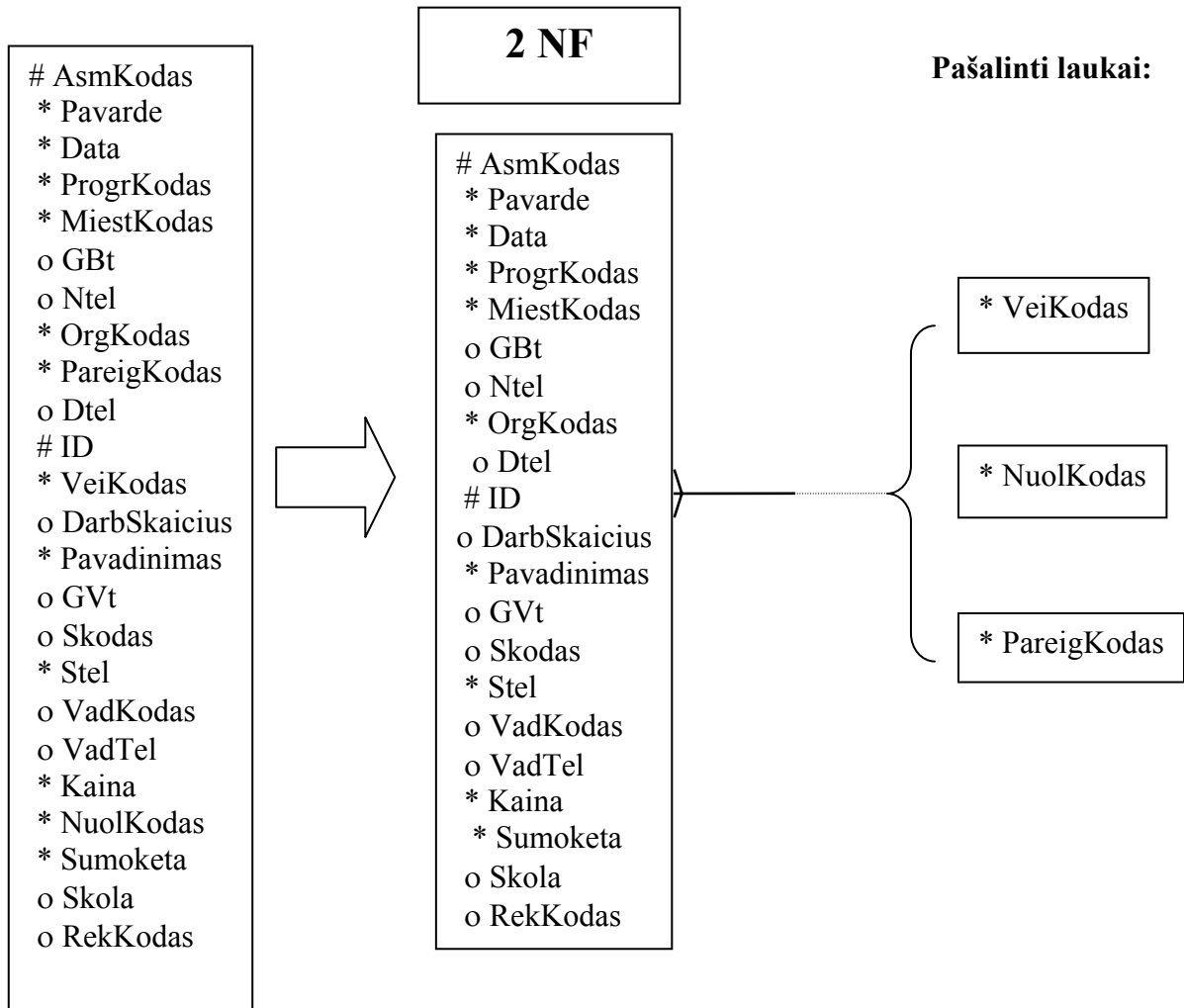
o - pažymi, kad atributo reikšmės nebūtina žinoti (optional).



18 pav. DB bazės „VMC klientai“ 1–ji normalinė forma

Toliau pašaliname atributus , kurie priklauso tik nuo dalies unikalios indentifikatoriaus.

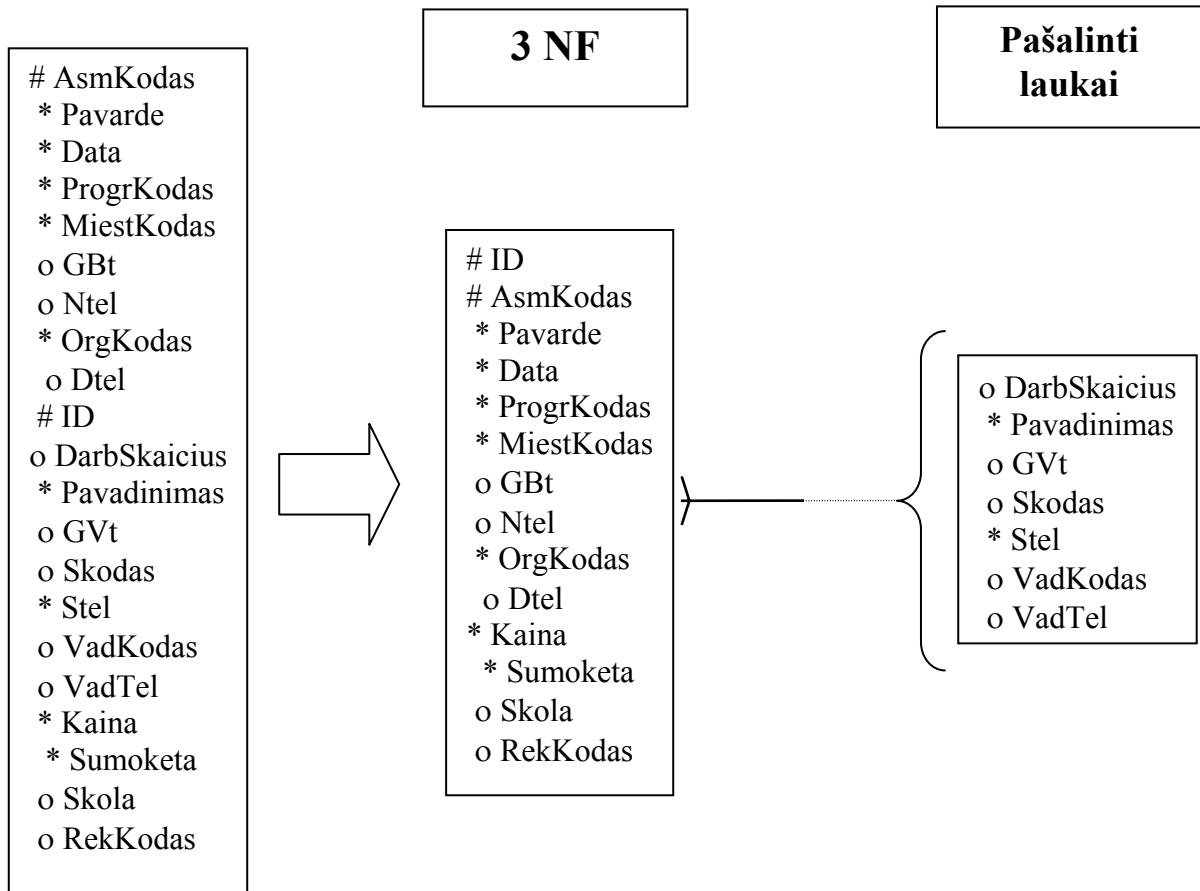
Gauname antrąją normalinę formą.



19 pav. DB bazės „VMC klientai“ 2–ji normalinė forma

Toliau pašaliname atributus , kurie nepriklauso unikalios indentifikatoriaus.

Gauname trečiąją normalinę formą.



20 pav. DB bazės „VMC klientai“ 3–ji normalinė forma

6.2. Esybių ryšių diagramos

Esybių-ryšių modeliavimas yra priemonė, leidžianti apibrėžti organizacijos informacinius poreikius [6]. Paprasčiausiu atveju, esybių-ryšių modeliavimo metu identifikuojami svarbūs organizacijos objektai (esybės), šių objektų savybės (atributai) ir kaip šie objektai siejasi tarpusavyje (ryšiai).

Esybių-ryšių modeliavimo tikslas - sudaryti organizacijos, kuriai kuriama nauja arba tobulinama jau egzistuojanti informacijos sistema, informacijos poreikių modelį.

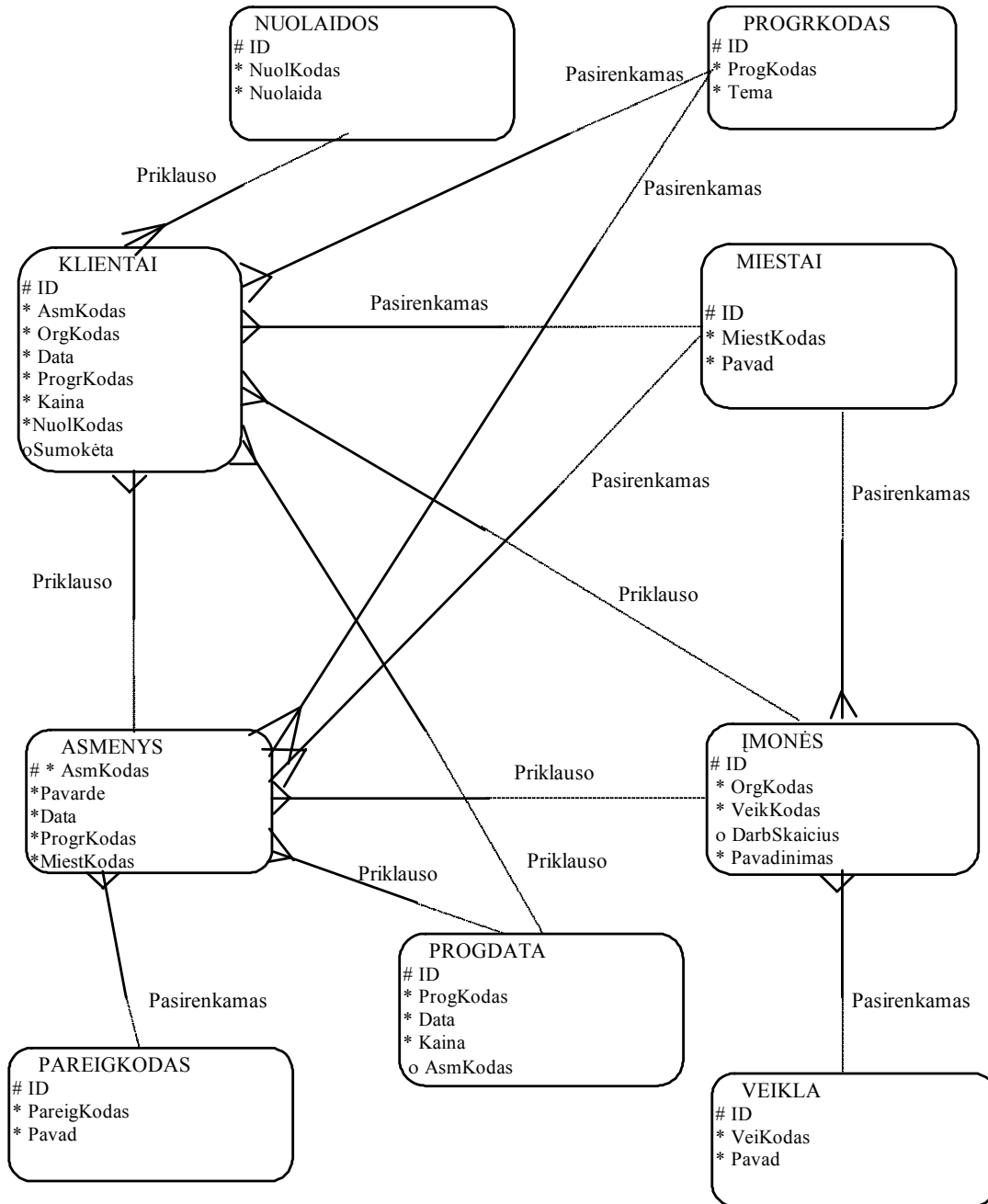
Dviejų esybių tipų A ir B sąsajos pobūdis (ryšio savybės atributų reikšmių lygmenyje) yra vadinamas **kardinalumu**:

6 lentelė. Ryšio kardinalumo reikšmės

Kardinalumo reikšmė	Kardinalumo pavadinimas (liet./anglųk.)	Asociacijos pobūdžio aprašymas
1 : 1	Vienas_su_vienu (<i>One-to-one mapping</i>)	Kiekvienai A tipo esybei yra tik viena B tipo esybė, ir kiekvienai B tipo esybei yra tik viena A tipo esybė.
1 : M	Vienas –su –daug (<i>One-to-many mapping</i>)	Kiekvienai A tipo esybei gali būti daug B tipo esybių, ir kiekvienai B tipo esybei yra tik viena A tipo esybė.
M : 1	Daug-su-vienu (<i>Many-to-one mapping</i>)	Kiekvienai A tipo esybei yra tik viena B tipo esybė, ir kiekvienai B tipo esybei gali būti daug A tipo esybių.
M : N	Daug-su-daug (<i>Many-to-many mapping</i>)	Kiekvienai A tipo esybei gali būti daug B tipo esybių, ir kiekvienai B tipo esybei gali būti daug A tipo esybių.

Ryšys yra binarinis ta prasme, kad jis visada yra asociacija tik tarp dviejų įvardintų esybių tipų arba tarp tos pačios esybės tipo egzempliorių (esyių). Kiekvienas ryšys dalinamas į dvi dalis ir kiekviena ryšio dalis iš turi tokias charakteristikas:

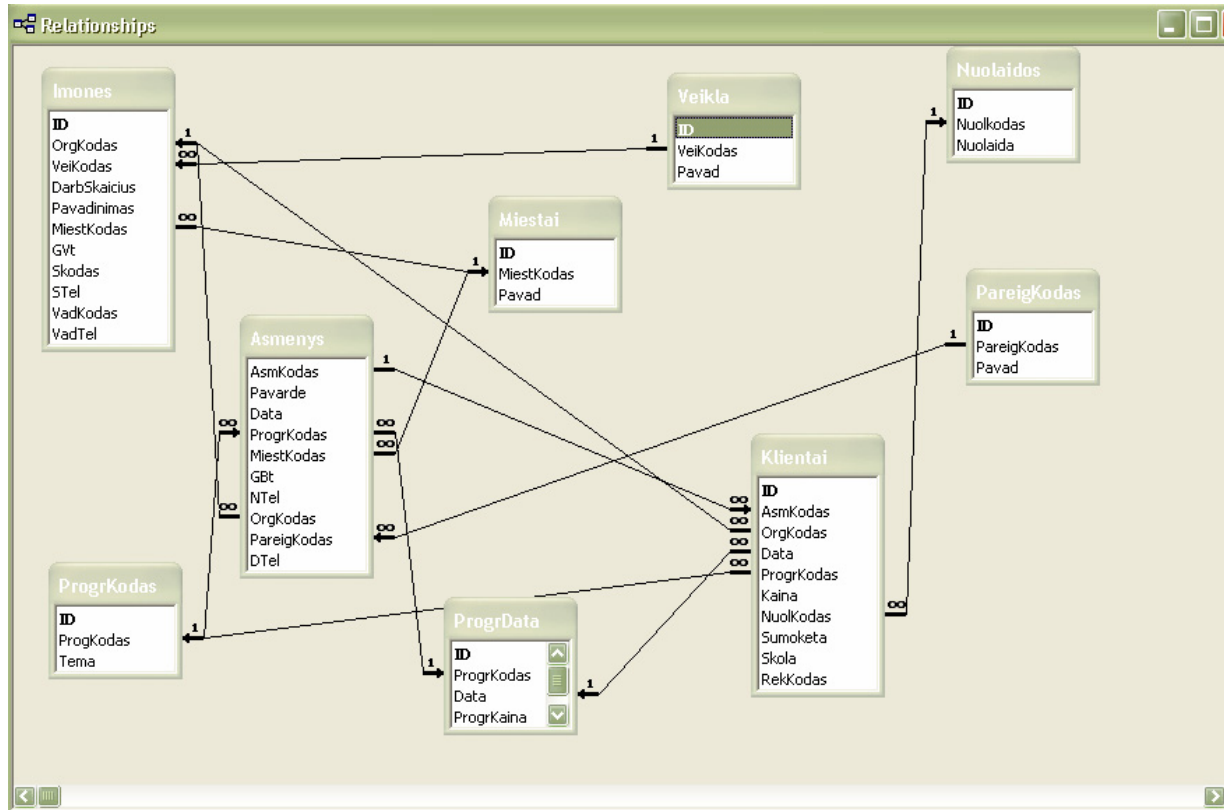
- Vardą (*role*), kuris rašomas prie atitinkamos dalies;
- kardinalumą (*degree/cardinality*), kuris gali įgauti dvi reikšmes: ‘daug’ (ryšio gale žymima trišakiu dar vadinamu ‘varnos pėda’) ir ‘vienas’ (žymima vientisa linija);
- asociacijos būtinumą (*optionality*): būtina asociacija (*mandatory*) žymima vientisa linija, nebūtina (*optional*) - brūkšnine linija).



21 pav. VMC esybių ryšių diagrama

6.3. Duomenų bazės ryšiai

Galutinai pasitelkę MS Access ir atsižvelgdami į anksčiau gautą esybių ryšių diagramą sukuriame DB „VMC klientai“ ryšius (22 pav.).



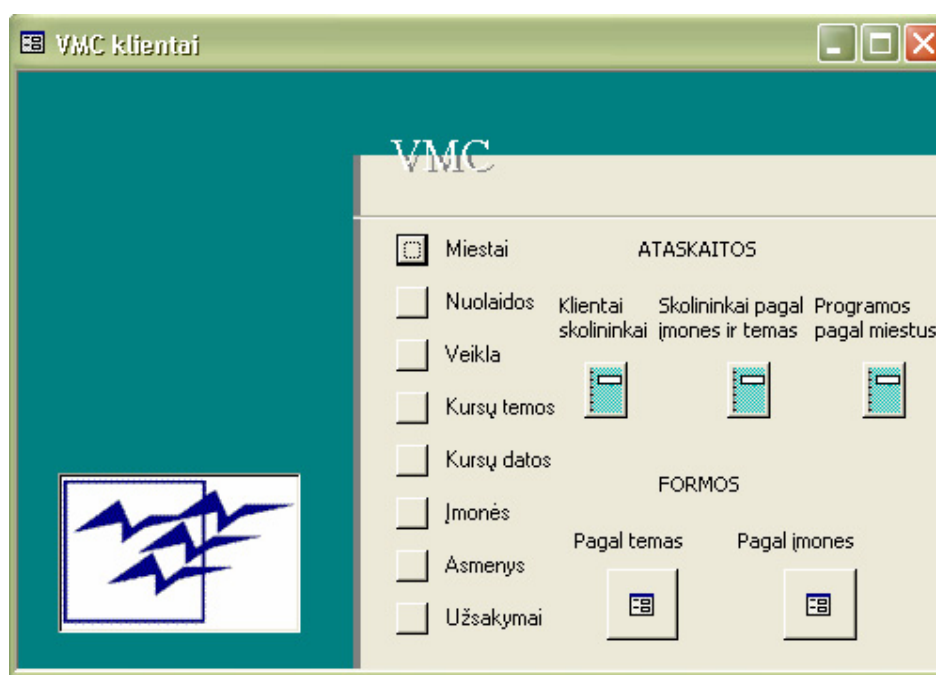
22 pav. DB „VMC klientai“ ryšiai

6.4. Duomenų bazės pagrindinė forma

Sudarome MS Access DB „VMC klientai“ pagrindinę formą (23 pav).

Žemiau pateikiamas DB „VMC klientai“ lentelių sąrašas (7 lentelė). Lenteles galima peržiūrėti priede Nr.1.Ten pat pateiktos ir autoriaus sukurtos formos, užklaustos ir ataskaitos.

Priede Nr2 pateikta DB„VMC klientai“ įdiegimo tvarka.



23.Pav. DB „VMC klientai“ pagrindinė forma

7 lentelė.DB lentelių sąrašas

Lentelių pavadinimas	Pastabos
Klientai	Duomenys apie VMC klientus (įmonės arba pavieniai asmenys)
Asmenys	Duomenys apie asmenis (klientai ir darbuotojai)
Įmonės	Duomenys apie įmones
Progdata	Kursų temų kodai ir numatoma kursų data , kaina , dėstytojas
ProgKodas	Kursų pavadinimas temų kodai
Miestai	Miestų pavadinimai ir pašto indeksai
PareigKodas	Pareigybių pavadinimai ir kodai
Veikla	Įmonių veiklos tipo pavadinimai ir kodai
Nuolaidos	Nuolaidų pavadinimai ir kodai
Reklamos	Reklamos pavadinimai ir kodai

7. Klientų registracijos Internetė modulis

7.1. PHP įdiegimas

Išarchyvuoti atsisiųstą zip bylą į laikinąjį aplanką ir nukopijuoti aplanke php-4.2.0-Win32 esančius aplankus bei bylas į C:\Php aplanką.

Dabar perkeltite C:\Php esančią bylą php4ts.dll į Windows sisteminio aplanko System32 pakatalogį.

Konfigūravimas. Dabar reikia pageduoti Apache konfigūracinę bylą ir įrašyti informaciją apie PHP modulį bei aprašyti naudosimą .php plėtinį. Atsidaryti

C:\Apache\conf\httpd.conf ir jo gale įrašyti:

```
LoadModule php4_module c:/php/sapi/php4apache.dll
```

```
AddType application/x-httpd-php .php
```

Toliau reikia "paleisti" "Apache" serverį. Per "Services, pasirinkus Apache, paspaudus ant jo dešinį klavišą ir pasirinkus "Start". Punkte "Description" matosi "Apache/1.3.24 (Win32)

PHP/4.2.0", punkte "Status" - "Started", "Startup Type" turėtų būti nurodytas "Automatic".

7.2. MySQL įdiegimas

Išskleisti atsisiųstą zip bylą į laikiną aplanką ir paleisti Setup.exe. Pagal nutylėjimą, bus nurodytas "C:\mysql. Geriausia nekeisti nes kitaip teks konfigūruoti .conf bylą.

Sekančiame dialogo lange pasirinkite įdiegimo tipą. Pasirinkite "Custom" tipą. Prasidės MySQL diegimas, o galiausiai reikės paspausti "Finish".

MySQL, kaip galite matyti, buvo įdiegtas "C:\mysql" kataloge.

Dabar reikia įdiegti MySQL kaip servisą. Komandinėje eilutėje surinkite:

```
cd c:\mysql\bin
```

```
mysqld-max-nt.exe -install
```

Dabar reikia MySQL serverį paleisti, o tai daroma per "Services" meniu.

Po to reikės šiek tiek sutvarkyti saugumą. MySQL sistemoje aukščiausias priėjimo teises turi vartotojas "root" ir yra įdiegiamas pagal nutylėjimą, tačiau be slaptažodžio, todėl privalu tą slaptažodį kuo skubiau uždėti. Komandinėje eilutėje, kurią išsikvietėte įdiegdami MySQL kaip servisą parašykite:

```
mysqladmin -u root -p password jusu_parinktas_root_slaptazodis
```

Dabar patikrinkite ar galima prisijungti prie MySQL serverio komandinėje eilutėje surinkus:

C:\mysql\bin>mysql -u root

Turėtumėte pamatyti tokį užrašą:

ERROR 1045: Access denied for user: 'root@localhost' (Using password: NO)

Jeigu matomas šis užrašas – prisijungimas pavyko sėkmingai.

7.3. Formų kūrimas

Duomenų bazės „VMC klientai“ klientų registravimui Internetu ir duomenų administravimui nuotoliniu būdu buvo sukurtas tinklalapis **registracija.html**.



24 pav. Dreamweaver MX

Tinklalapio ir formų kūrimui naudoju programą Dreamweaver MX (24.Pav.).

Šiame tinklalapyje registracija.html vartotojai gali užsiregistruoti į pasirinktus mokymo kursus.

Žemiau pateiktas formos programinis kodas ir tinklalapio vaizdas (25 pav.).

```
<form action="siusti.php" method="post" name="ivedimas" target="mainFrame">
<table width="467" style="border-collapse:collapse; font-size:10pt" bordercolor="#003366"
border="0" cellpadding="4">
<tr> <td width="145">Asmens kodas </td>
<td width="300"><input type="text" name="kodas" size=50 maxlength=11></td> </tr>
<tr> <td>Vardas</td>
<td><input type="text" name="vardas" size=50 maxlength=20></td></tr>
<tr><td>Pavardė</td>
<td><input type="text" name="pavarde" size=50 maxlength=20></td> </tr>
<tr><td>Data</td>
<td><input type="text" name="data" size=50 maxlength=11></td></tr>
<tr><td>Miestas</td>
<td><input type="text" name="miestas" size=50 maxlength=20></td></tr>
<tr><td>Namų tel. </td>
```

```

<td><input type="text" name="namtel"          size=50 maxlength=12></td> </tr>
<tr> <td>Darbo tel. </td>
<td><input type="text" name="darbtel" size=50 maxlength=12></td> </tr>
<tr><td>Kursai</td><td><select name="kurs">
  <option value = "we">Pasirinkite kursus
  <option value = "1">Kompiuteris iš vidaus
  <option value = "2">Kompl. komp. mok. programa suaugusiems
    <option value = "3">Komp. mok. mokiniams pradedantiesiems
    <option value = "4">Komp. mokymas mokiniams pažengusiesiems
    <option value = "6">Corel Draw maketavimo programa
    <option value = "7">Programavimo I lyg. Turbo Pascal progr. kalba
</select></td> </tr>
<tr> <td>Įmonė</td><td><input type="text" name="imone" size=50 maxlength=12></td></tr>
<tr> <td>Įmonės veiklos pobūdis </td><td><input type="text" name="pobudis" size=50
maxlength=12></td> </tr>
<tr><td>Pareigos</td><td><input type="text" name="pareigos" size=50
maxlength=12></td></tr></table><input name="Saugoti" type="submit" value="Siusti">
</form>

```

25 pav. VMC klientų registracijos tinklalapis

Įvedus duomenis ir paspaudus mygtuką „Siusti“ užkraunamas naujas atitinkamas PHP kalboje sukurta tinklapis, kuriame esantis skriptas persiunčia duomenis į MySQL duomenų bazę. Numatyta galimybė administratoriui keisti, šalinti ir redaguoti bazės duomenis nuotoliniu būdu. Tam sukurta nuoroda admin, kuri pateikia registracijos langą su vartotojo vardu ir slaptažodžiu. Įvedus teisingą slaptažodį paleidžiamas PHP programos kodas, kuris pateikia administratoriui visus iki tol į My SQL duomenų bazę įvestus duomenis (26 pav.). Visi sukurti PHP programų kodai pateikti priede Nr3.

Eil. Nr.	asmens kodas	vardas	pavardė	data	miestas	namų tel.	darbo tel.	kursai	įmonė	įmonės veiklos pobūdis	pareigos
1	545454545	Henrikas	Sinickas	2003 12 10	Panevėžys	456454		Kompiuteris iš vidaus	Panevėžio ko	mokymas	destytojas
2	364564554	Jonas	Jonaitis	2003 10 31	Vilnius	5646545		Corel Draw maketavimo programa	Lietuvos Aid	Spauda	korektorius
3	1455554544	Robertas	Sinickas	54645455	Kaunas	454564561		Komp. mokymas mokiniams pažengusiesiems	Naktinis ba	Laisvalaikis	administrato

26 pav. MySQL duomenų bazės redagavimo lentelė

Abu moduliai: DB „VMC klientai“ ir „Klientų registracija Internete“ gali dirbti savarankiškai. Tačiau yra numatyta galimybė abu šiuos modulius sujungti į vieną sistemą. Priede Nr4 pateikta MS ACCESS‘2000 duomenų bazės prijungimo My SQL virtualaus serverio tvarka. Tokiu būdu klientų duomenys užsiregistravus Internete ar administratoriaus redaguoti duomenys automatiškai bus perkelti į pagrindinę duomenų bazę.

8. IŠVADOS

Kauno technologijos universiteto Panevėžio filialo Vadybos technologijos ir verslo mokymo centro (VMC) klientų registravimui, apskaitai ir planavimui sukurta informacinė sistema „VMC klientai“. Sukurtas klientų registracijos Internetu modulis - internetinis puslapis su nuorodomis į reikiamas programas ir klientų registravimu bei duomenų kaupimu MySQL duomenų bazėje ir šios duomenų bazės nuotoliniu administravimu PHP programinės įrangos pagalba sukurtais programų kodais .su galimybe duomenis automatiškai perkelti iš My SQL duomenų bazės į pagrindinį informacinės sistemos „VMC klientai“ Access duomenų bazės modulį. Tiriamojo darbo metu atlikta:

1. atlikta VMC veiklos analizė;
2. sudaryti informacinių procesų modeliai- duomenų srautų diagramos;
3. atliktas informacinės sistemos reikalavimų specifikavimas-apibrėžti bendrieji, funkciniai reikalavimai sistemai, reikalavimai sąsajai, sistemos darbo reikalavimai;
4. sukurtas taikomųjų uždavinių modelis, kuris apima veiklos uždavinių modelį ir vartotojo informacinių poreikių modelį;
5. pasitelkus Provision Workbench įrangą buvo sukurti biznio panaudojimo atveju ir sistemos modeliai;
6. sukurtas informacinės sistemos architektūros modelis;
7. pasitelkus Provision Workbench įrangą buvo sukurti objektų, būsenų ir metodų modeliai;
8. Provision Workbench priemonėmis buvo atliktas projekto testavimas;
9. metodų modelio pagrindu buvo atliktas projekto prototipo generavimas Provision Workbench priemonėmis;
10. sukurta esybių ryšių diagrama ir sukurta duomenų bazė Access‘2000 aplinkoje;
11. sukurtas klientų registracijos Internetu modulis su galimybe duomenis automatiškai perkelti iš My SQL duomenų bazės į pagrindinį informacinės sistemos „VMC klientai“ Access duomenų bazės modulį.

9. LITERATŪRA

1. S.Gudas Organizacijos informacinių poreikių analizė. - Kaunas, Technologija, 2001.
2. V.Sekliuckis, G.Garšva, S.Gudas Duomenų bazės, Kaunas, Naujasis lankas, 2001.
3. Turban, E., McLean, E., Wetherbe, J. Information technology for management. – John Wiley&Sons, Inc., 1999.
4. K.C.Laudon, J.P.Laudon Management Information Systems. Organization and technology in the networked enterprise.- Prentice Hall, 2000. www.prenhall.com/laudon
5. Modern database systems. - Ed. by W.Kim. - Addison-Wesley Publ. Comp., - 1995.
6. S. Gudas.Veiklos analizė ir informacinių poreikių specifikavimas. - Kaunas, 2002.
7. P.P.Chen The Entity-Relationship Model: Towards a Unified view of Data, ACM Transactions on Database Systems, Vol.1. No.1, March 1976, pp.9 –36.
8. Sekliuckis V., Gudas S., Garšva G. Informacijos sistemos ir duomenų bazės. Vadovėlis. – Kaunas, Technologija, 2003, 318 p.
9. Gudas S. Organizacijų veiklos modeliavimas. Vadovėlis–Kaunas, Technologija, 2000, 134 p.
10. Gudas S. Organizacijos informacinių poreikių analizė. Mokomoji knyga – Kaunas, Technologija, 1998, 68 p.
11. Gudas S. Organizacijų valdymo modeliavimo sistema. Mokomoji knyga. – Kaunas, Technologija, 1998, 34 p.
12. Gudas S. Organizacinių sistemų valdymo modeliai ir algoritmai. Laboratorinis darbas "Informacinių ir technologinių procesų sąryšio modeliavimas" / Kaunas, Technologija, 1992, 29p.
13. Gudas S., Kriščiokaitienė G. Organizacinių sistemų valdymo modeliai ir algoritmai.Laboratoriniai darbai. 1 dalis / Kaunas, Technologija, 1992, 54 p.
14. Gudas S. Formalūs automatizuojamų objektų aprašymo metodai. Metodiniai nurodymai. – Kaunas, KPI, 1989, 38 p.
15. Atkočiūnienė E, Gudas S., Šakys V. Dialogas su duomenų banku. Mokymo priemonė. – Vilnius, 1985, 86 p.
16. Л.Томсон,Л.Веллинг.Разработка Web приложений на PHP и MySQL. М.2002, 655 стр.

SUMMARY

The programme of client's database and its relation with clients software

Master of scientific work Henrikas Sinickas,

Research adviser doc dr. Ojaras Purvinis

The information system „VMC klientai“ has been established for the client's registration and panning in Kaunas Technuological University, Panevėžys Departament, the centre of Management Technology and Busines training centre. The modulus of client's registration – the internet page with instruction to required programmes and client's registration as well as data compiling in MySQL database and with the same database distance administration, with a help of PHP software having been created programmes codes that could have a possibility to transfer data automatically from MySQL database into the main information system „VMC klientai“ Access database modulus.

10. PRIEDAI

10.1. Priedas Nr1. Duomenų bazės lentelės, formos ir ataskaitos

10.1.1. Duomenų bazės lentelės

Remiantis esybių diagrama projekte yra kuriamos duomenų lentelės. Lentelių turinį ir laukų pavadinimus apsprendžia normalizavimo uždavinyje gautos normalinės formos. Bendras lentelių sąrašas yra pateikiamas.

	AsmKodas	Pavarde	Data	ProgrKodas	MiestKodas	GBt	NTel	OrgKodas	PareigKodas
+	12345678912	Purvinis Ojaras	2002.10.12	0020	5300	Kniaudiškių 54	984456877	152	400
+	12354548758	Kaziliūnas Petras	2002.04.25	0011	5300	Ramygalos 8-5	546656622	122	300
+	12356988787	Povilonienė Ona	2002.10.05	0012	5300	Raginėnų 45-26	593269587	120	600
+	41421222323	Jonaitis Jonas	2002.11.01	0013	5300	Šiaulių 60-1	415654645	121	100
+	44854746466	Povilonis Leonardas	2002.04.06	0010	4700	Joniskio 52-102	545565656	125	500
+	45124578788	Sigita Grybas	2002.11.12	0010	2700	Alyvų 60-45	545412554	121	200
+	45454564663	Snickas Henrikas	2002.10.25	0010	5300	Kniaudiškių 41-	545645656	123	100
+	45454585656	Kupšys Oswaldas	2002.09.15	0014	2200	Laisvės a. 45-4	569565226	121	200
+	47888989989	Elena Leikaitė	2002.11.12	0012	4700	Utenos 13-15	415879899	121	200
+	55875658921	Haris Poteris	2002.11.12	0010	4970	Laisvės al. 15-3	457878411	125	300
*			2002.11.12		0				

27 pav. Lentelė Nr.1. Asmenys

Šioje lentelėje visi reikalingi kodai pasirenkami iš atitinkamų lentelių (LOOKUP).

	ID	AsmKodas	OrgKodas	Data	ProgrKodas	Kaina	NuolKodas	Sumoketa	Skola	RekKo
▶	1	12345678912	152	2002.04.06	0020	500,00 Lt	1	400,00 Lt	100,00 Lt	11
	2	12354548758	120	2002.07.05	0011	540,00 Lt	1	300,00 Lt	300,00 Lt	10
	3	44854746466	125	2002.04.06	0015	250,00 Lt	0	250,00 Lt	0,00 Lt	10
	4	45454585656	121	2002.10.05	0014	100,00 Lt	0	100,00 Lt	0,00 Lt	14
	5	41421222323	121	2002.08.21	0013	500,00 Lt	1	200,00 Lt	300,00 Lt	11
	6	12354548758	122	2002.06.12	0011	300,00 Lt	1	200,00 Lt	100,00 Lt	10
	7	12356988787	120	2002.02.07	0012	200,00 Lt	0	200,00 Lt	0,00 Lt	12
	8	45454564663	123	2002.05.26	0010	300,00 Lt	1	300,00 Lt	0,00 Lt	10
*	umber)		0			0,00 Lt	0	0,00 Lt	0,00 Lt	

28 pav. Lentelė Nr.3. Klientai.

	ID	OrgKod	VeikKodas	DarbSkaicius	Pavadinimas	MiestKodas	GVt	Skodas	STel	VadKodas	VadTel
▶	1	120	454	5456	"Ekranas" g-kl	5300	Senamiesčio 120	0	545454554	545487787	545454445
+	2	121	412	8665	"Kalnapilio" g-k	5300	Respublikos 150	0	545454545	454555556	545545454
+	3	125	456	6542	"Sernos" g-kl	5300	respublikos 54	545456666	455454555	545454545	455545454
+	4	123	455	6221	"Utenos" g-kl	4700	Ažuolų 54	545545454	545454545	545454545	545545445
+	5	124	544	1233	"Nevėžio" g-kl	5300	Respublikos 45	545545545	878787898	544554545	455454545
+	6	122	400	10	"Grafelio" pardu	5300	Respublikos 11	0	455556666	454556646	456987887
+	7	129	401	50	"Maxima"	2700	Kauno 21	24545854	4478745	545445454	54545454
+	8	130	412	523	"Mėšos" g-kl	2200	Vilniaus 58	545454545	545454445	566566666	656566656
+	9	152	500	120	KTU Panevėžio	5300	Laisvės a. 23	554611211	545555555	545689899	797997977
*	nber)					0		0			

29 pav. Lentelė Nr. 2. Įmonės

Šioje lentelėje visi reikalingi kodai pasirenkami iš atitinkamų lentelių (LOOKUP).

10.1.2. Duomenų bazės formos

ID	AsmKodas	OrgKodas	Data	ProgrKod	Kaina	NuolKoda	Sumoketa	Skola	Re.
1	1,2E+10	152	4.06	0020	100,00 Lt	1	100,00 Lt	03,00 Lt	11
2	1,2E+10	120	7.05	0011	140,00 Lt	1	100,00 Lt	103,00 Lt	10
3	4,5E+10	125	4.06	0015	150,00 Lt	0	150,00 Lt	3,00 Lt	10
4	4,5E+10	121	0.05	0014	00,00 Lt	0	00,00 Lt	3,00 Lt	14
5	4,1E+10	121	8.21	0013	100,00 Lt	1	100,00 Lt	103,00 Lt	11
6	1,2E+10	122	6.12	0011	100,00 Lt	1	100,00 Lt	03,00 Lt	10
7	1,2E+10	120	2.07	0010	100,00 Lt	0	50,00 Lt	3,00 Lt	12
8	4,5E+10	123	5.26	0010	100,00 Lt	1	100,00 Lt	3,00 Lt	10
*	umber)	0			0,00 Lt	0	0,00 Lt	3,00 Lt	

Record: 1 of 8

30 pav Forma Nr. 1 Klientai.

Užsakymai	
Atsakingas asmuo	
Įmonės pavadinimas	'Ekrano" g-kla
Veiklos pobūdis	454
Darbuotojų skaičius	5456
Miestas	5300
Adresas	Senamiesčio 120
Sekretorės tel.	545454554
Vadovo tel.	545454445
Record: 1 of 9	

31 pav. Forma Nr.2. Užsakymai.

Šioje formoje paspaudus mygtuką „Atsakingas asmuo“ atsidaro atitinkama forma:

Asakingas asmuo

Data: 2002.10.05 Darbovietė: 120
 Asmens kodas: 12356988787 Pareigos: 600
 Pavardė, vardas: Povilonienė Ona Darbo tel.: 546657777
 Miesto kodas: 5300
 Adresas: Raginėnų 45-26
 Namų tel.: 593269587

Naujas įrašas Pašalinti įrašą Įrašyti

Apmokėjimas

ProgrKodas	Kaina	NuolKodas	Sumoketa
▶ 0010	200,00 Lt	0	150,00 Lt
*	0,00 Lt	0	0,00 Lt

Record: 1 of 1

Gryžti STOP

Record: 1 of 1 (Filtered)

33 pav. Forma Nr.2A. Atsakingas asmuo.

PASTABA. Forma Nr 2 skirta pagrindinių duomenų įvedimui.

10.1.3. Duomenų bazės užklausos

Skolos : Select Query

	Įmonės pavadinimas	Kaina	Sumoketa	Skola
▶	"Ekranas" g-kla	200,00 Lt	150,00 Lt	50,00 Lt
	"Ekranas" g-kla	540,00 Lt	300,00 Lt	240,00 Lt
	"Grafelio" parduotuvė	300,00 Lt	200,00 Lt	100,00 Lt
	"Semos" g-kla	100,00 Lt	100,00 Lt	0,00 Lt
	"Semos" g-kla	500,00 Lt	200,00 Lt	300,00 Lt
	"Sonex" firma	250,00 Lt	150,00 Lt	1.000,00 Lt
	"Utenos" g-kla	300,00 Lt	300,00 Lt	0,00 Lt

Record: 1 of 8

34 pav. Atrenkamoji užklausa Nr.1. Skolos.

Klijantai_Crosstab : Crosstab Query

ProgrKoda:	Total Of Kaina	120	121	122	123	125	152
▶ 0010	500,00 Lt	200,00 Lt			300,00 Lt		
0011	840,00 Lt	540,00 Lt		300,00 Lt			
0013	500,00 Lt		500,00 Lt				
0014	100,00 Lt		100,00 Lt				
0015	250,00 Lt					250,00 Lt	
0020	500,00 Lt						500,00 Lt

Record: 1 of 6

35 pav. Kryžminė užklausa Nr.2. Sumokėta pagal įmonių kodus.

AsmKodas	Kaina	ProgrKoda:	PVM
12345678912	500,00 Lt	0020	90
12354548758	540,00 Lt	0011	97,2
44854746466	250,00 Lt	0015	45
45454585656	100,00 Lt	0014	18
41421222323	500,00 Lt	0013	90
12354548758	300,00 Lt	0011	54
12356988787	200,00 Lt	0010	36
45454564663	300,00 Lt	0010	54
* 0	0,00 Lt		

Record: 1 of 8

36 pav. Skaičiuojamoji užklausa Nr.3. Su PVM skaičiavimu.

10.1.4. Duomenų bazės ataskaitos

<i>Klientai Skolininkai</i>		
Zem.	Įmonės pavadinimas	Suma, QTDeta
Šilainiai apskr.		
	UAB "Klaipėdos paštas"	105,66 Lt
Summary for "Klaipėdos paštas" n. ataskaitos		105,66 Lt
Sum		105,66 Lt
Freiza		
	"Sveikof" gėlių	105,66 Lt
Summary for "Klaipėdos paštas" n. ataskaitos		105,66 Lt
Sum		105,66 Lt
Informacijos techn. įstaiga		
	"Sveikof" gėlių	6,66 Lt
	"Sveikof" žvakių	105,66 Lt
	"Sveikof" žvakių	105,66 Lt
	"Sveikof" gėlių	105,66 Lt
	"Sveikof" gėlių	105,66 Lt
Summary for "Klaipėdos paštas" n. ataskaitos		228,66 Lt
Sum		228,66 Lt
Adriatikas		
	"Sveikof" gėlių	105,66 Lt
Summary for "Klaipėdos paštas" n. ataskaitos		105,66 Lt
Sum		105,66 Lt
Adriatikas		
	KTU Pasažierių	105,66 Lt
Summary for "Klaipėdos paštas" n. ataskaitos		105,66 Lt
Sum		105,66 Lt
UAB "Laidis"		
	"Sveikof" gėlių	105,66 Lt
Summary for "Klaipėdos paštas" n. ataskaitos		105,66 Lt
Sum		105,66 Lt
Grand Total		1.090,00 Lt

2007 m. gruodžio 31 d. Page 1 of 1

Page: 1

37 pav. Ataskaita Nr. 1. Klientai skolininkai.

Klientai

Klientai

<i>Tema</i>	<i>Objektas</i>	<i>Įmonės pavadinimas</i>	<i>Paspyktas</i>
Kitas	120	Kitas	120
	120	Kitas	120
Kitas	120	Kitas	120
Kitas	120	Kitas	120
	120	Kitas	120
Kitas	120	Kitas	120
Kitas	120	Kitas	120
Kitas	120	Kitas	120

2007 m. gruodis 20 d. Page: 1 of 1

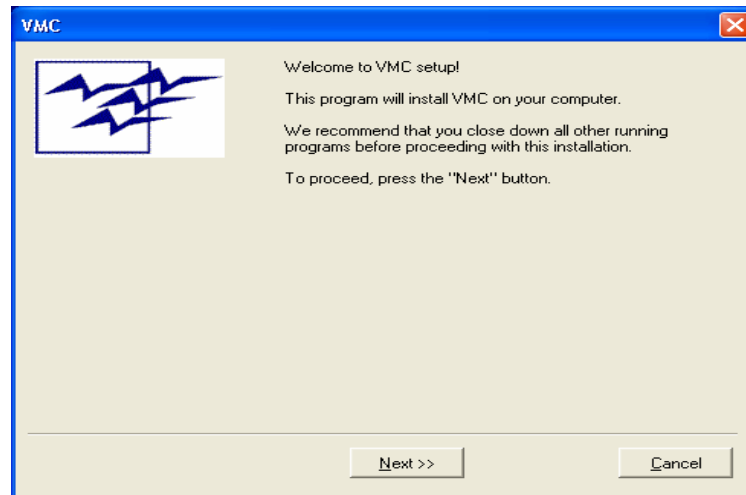
Page: 1

38 pav. Ataskaita Nr. 2. Klientai-įmonės pagal kursų temas.

10.2. Priedas Nr.2

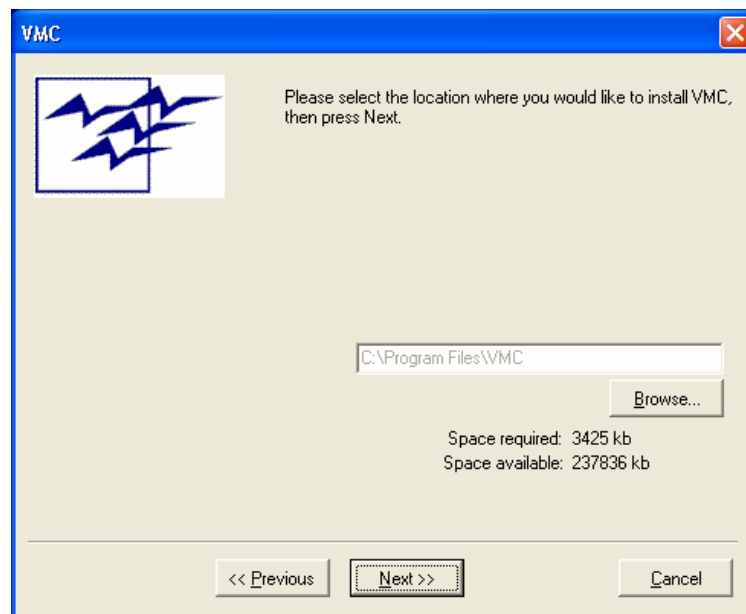
DB „VMC klientai“ įdiegimas

Paleisti įdiegimo programą VMCSetsup.exe ir atsidariusiame lange paspausti mygtuką Next.



39 pav. Įdiegimo 1-as žingsnis

Po to sekančiame lange nurodyti aplanką, į kurią bus perkeltos DB „VMC klientai“ bylos.



40 pav. Įdiegimo 2-as žingsnis

Po to sekančiame lange nurodyti ar reikalingos programos ikonos ir paspausti mygtuką Finish..

10.3. Priedas Nr.3

PHP programų kodai

10.3.1. Kliento duomenų siuntimas į My SQL duomenų bazę

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>duomenys išsaugoti</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1257">
</head>
<body>
<h2>duomenys išsaugoti...</h2>
<?
switch ($kurs) {
case "1" :
    $kursai = "Kompiuteris iš vidaus";
break;
case "2" :
    $kursai = "Kmpl. komp. mok. programa suaugusiems";
break;
case "3" :
    $kursai = "Komp. mok. mokiniams pradedantiesiems";
break;
case "4" :
    $kursai = "Komp. mokymas mokiniams pažengusiems";
break;
case "5" :
    $kursai = "Komp. mok. mokiniams pradedantiesiems";
break;
case "6" :
    $kursai = "Corel Draw maketavimo programa";
break;
case "7" :
    $kursai = "Programavimo I lyg. Turbo Pascal progr. kalba";
break;
}
$con=@mysql_connect("localhost", "root", "slaptazodis") or die ("Nepavyko prisijungti prie duomenų
serverio.");
mysql_select_db(db_panko);
mysql_query("INSERT INTO henri VALUES (',$kodas', '$vardas', '$pavarde', '$data',
'$miestas', '$namtel', '$darbtel', '$kursai', '$imone', '$pobudis', '$pareigos')");
?>
</body>
</html>

```

10.3.2. Administratoriaus prisijungimas ir duomenų pateikimas iš My SQL duomenų bazės

```

<?
$vardas = "henri";
$slaptazodis = "henri";
function klaida ($klaida) {

```

```

echo $klaida."<br>";
exit;
}
if ( (!isset($PHP_AUTH_USER)) || ! (($PHP_AUTH_USER == $vars) && ( $PHP_AUTH_PW ==
"$slaptazodis" ) ) ) {
header("WWW-Authenticate: Basic enter=\\"Form2txt admin\\"");
header("HTTP/1.0 401 Unauthorized");
klaida("Informacija jums neprieinama !!!");
}
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>peržiūra</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1257">
</head>
<body>
<?
$paieska=@mysql_connect("localhost", "root", "slaptazodis") or die ("Nepavyko prisijungti prie
duomenu serverio.");
mysql_select_db(db_panko);
$uzklausa= mysql_query("SELECT * FROM henri");
?>
<table width="100%" style="border-collapse:collapse; font-size:10pt" bordercolor="#003366"
border="1" cellpadding="4" >
<tr>
<td width="7%">Eil. Nr.</td>
<td width="12%">asmens kodas</td>
<td width="10%">vardas</td>
<td width="10%">pavardė</td>
<td width="7%">data</td>
<td width="8%">miestas</td>
<td width="7%">namų tel.</td>
<td width="6%">darbo tel.</td>
<td width="7%">kursai</td>
<td width="7%">įmonė</td>
<td width="8%">įmonės veiklos pobūdis</td>
<td width="11%">pareigos</td>
</tr>
<?
$cid=1;
while ( $row = mysql_fetch_array($uzklausa) ) {
$zkodas=$row['asmens_kodas'];
$vardas=$row['vardas'];
$pavarde=$row['pavarde'];
$data=$row['data'];
$miestas=$row['miestas'];
$namutel=$row['namu_tel'];
$darbotel=$row['darbo_tel'];
$nr=$row['id_nr'];
$k=$row['kursai'];
$im=$row['imone'];
$pob=$row['pobudis'];
$par=$row['pareigos'];
echo "<tr><td>$cid</td>";

```

```

echo "<td><a
href=\"duomenų_keitimas.php?id=$nrs\">$kodas</a>
</td><td>$vardas</td><td>$pavarde</td><td>$data</td><td>$miestas</td><td>$namutel</td><td>$dar
botel</td>
<td>$k</td><td>$im</td><td>$pob</td><td>$par</td></tr>";
$Sid ++;
}
?>
</table>
</body>
</html>

```

10.3.3. Duomenų keitimas My SQL duomenų bazėje

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1257">
</head>
<body><h2>Duomenys pakeisti...</h2>
<?
switch($kursai)
{
    case "z" :
        $kursas = "$kz";
        break;
        case "I" :
        $kursas = "Kompiuteris iš vidaus";
        break;

        case "II" :
        $kursas = "Kompl. komp. mok. programa suaugusiems";
        break;
        case "III" :
        $kursas = "Komp. mok. mokiniams pradedantiesiems";
        break;
        case "IV" :
        $kursas = "Komp. mokymas mokiniams pažengusiesiems";
        break;
        case "VI" :
        $kursas = "Corel Draw maketavimo programa";
        break;
        case "VII" :
        $kursas = "Programavimo I lyg. Turbo Pascal progr. kalba";
        break;
}
$zduotis=@mysql_connect("localhost", "root", "slaptazodis") or die ("Nepavyko prisijungti prie
duomenų serverio.");
mysql_select_db(db_panko);
mysql_query("UPDATE henri SET  henri.asmens_kodas='$kodas', henri.vardas='$vardas',
henri.pavarde='$pavarde', henri.data='$data', henri.miestas='$miestas',
henri.namu_tel='$namutel', henri.darbo_tel ='$darbo_tel', henri.kursai ='$kursas', henri.imone ='$imone',
henri.pobudis ='$pobudis',
henri.pareigos ='$pareigos' WHERE henri.id_nr='$id'");
?></body></html>

```

10.3.4. Duomenų pašalinimas iš My SQL duomenų bazės

<?

```
$paieska=@mysql_connect("localhost", "root", "slaptazodis") or die ("Nepavyko prisijungti prie duomenų serverio.");
```

```
mysql_select_db(db_panko);
```

```
mysql_query("DELETE FROM henri WHERE henri.id_nr = '$id' LIMIT 1");
```

?>

<h3>delete complite...</h3>

10.4. Priedas Nr.4

MS ACCESS'2000 duomenų bazės prijungimas prie My SQL virtualaus serverio

10.4.1. MySQL ODBC Tvarkyklės

MySQL ODBC tvarkyklės jums leidžia prisijungti prie veikiančio MySQL esančio Virtualiame Privačiame Serveryje iš jūsų personalinio kompiuterio ir importuoti/eksportuoti duomenų bazes.

Diegimas

Apačioje nurodyta eiga turi būti atlikta jūsų kompiuteryje:

1. Jeigu jūs naudojate MS Access 2000, gali būti reikalinga atnaujinti jūsų MS Jet 4.0 duomenų bazės varikliuką prieš tęsiant. Nuorodos pateiktos žemiau:
 - [MS Product Support Services: Jet 4.0 Bug Explanation](#)
 - [MS Product Support Services: Jet 4.0 Service Pack 5](#)
2. Įdiegti reikalingą MySQL ODBC tvarkyklę į jūsų kompiuterį. Parsisiųskite tinkamą tvarkyklę, pateiktą žemiau:
 - [Windows 95, 98, ME](#)
 - [Windows NT/2000](#)

Išarchyvuokite tvarkyklę ir paleiskite tvarkyklės diegimo programą.

3. Sukonfigūruokite MySQL tvarkyklę naudojimui.

- Windows 2000

Iš *Start* meniu, pasirinkite *Settings* tuomet *Control Panel*. Du kart spauskite ant *Administrative Tools* ir du kart ant *Data Sources (ODBC)*.

- Windows 95/98/NT

Iš *Start* meniu, pasirinkite *Settings* tuomet *Control Panel*. Du kart ant *ODBC Sources* piktogramos.

Jums reiks pasirinkti ar jūs norite konfigūruoti ODBC tvarkyklę naudojimui vienam vartotojui ar naudojimui kiekvienam vartotojui kompiuteryje. Pirmoje skiltyje, „User DSN“ yra tikslai specifiniam vartotojui ir gali būti panaudotas specifiniam kompiuteriui. Antroje skiltyje, „System DSN“ yra naudojamas konfigūruoti ODBC tvarkyklę visiems vartotojams jūsų kompiuteryje. Priklausomai ką jūs pasirinksite naudoti, jums reiks paspausti mygtuką pridėti „add“ esantį dešinėje pusėje. Spaudžiant „pridėti“ mygtuką jūs būsite paklausti pasirinkti tvarkykles duomenų srautui. Jūs turėsite surasti MySQL sąrašę. Pasirinkite MySQL ir spauskite pabaiga „Finish“.

4. pagal nutylėjimą atsiras TDX MySQL tvarkyklės konfigūracijos langas. Jūs turėtumėte užpildyti langelius su paskirta informacija:

a. Windows DNS Name

Parašykite vardą šitai ypatingai tvarkyklei kurią jūs naudosite dėl MySQL. Vardas gali būti bet kas ką jūs pasirinksite. (Pvz.: everyoneMySQL)

b. Server

Čia yra Virtualaus Privataus Serverio vardas kur bus tiekama jūsų duomenų bazė. (Pvz.: myserver.net)

c. MySQL Database Name

Čia reikalingas vardas MySQL duomenų bazės prie kurios jūs jungsitės.

d. User

Čia bus reikalingas vartotojo vardas dėl MySQL duomenų bazės prie kurios jūs jungsitės.

e. Password

Čia MySQL vartotojo slaptažodis reikalingas apatiniam laukeliui.

f. Port (if not 3306)

Jei jūs esate už ugniasienės tai jums reikės atidaryti 3306 portą arba kitą portą, kad viskas dirbtų be priekaištų.

or it will not work correctly.

10.4.2.MS Access duomenų bazių eksportavimas į MySQL

Naudojant šias Windows MySQL ODBC tvarkykles jūs galite eksportuoti [Microsoft Access](#) duomenų bazes iš jūsų personalinio kompiuterio tiesiai į MySQL duomenų bazę serveryje.

1. [Idiekite MySQL](#) jūsų serveryje. Žinokite kad jums reiktų pasiaiškinti MySQL dokumentaciją dėl informacijos naudojant MySQL.
2. Pridėkite vartotoją su slaptažodžiu MySQL duomenų bazei. Žiūrėkite reikalingą skirsnį [MySQL Reference Manual](#):
 - a. [User Account Management](#)
3. Atidarykite MS Access ir sukurkite arba pažymėkite duomenų bazę lentelių poskyryje kurią jūs norite perkelti į serverį. Kai pažymėsite norimą lentelę, spauskite *Export...* per *File* meniu.
4. Pasirodžius *Save Table* lentelei. Čia galite pasirinkti pakeisti *Save as type* į *ODBC Databases* ir spausti **Export**.
5. Pasirodžius *Export* ekranui. *Export Addresses to:* laukelis turi būti vardas kuriuo norite iškviesti specifinę lentelę serveryje.
6. Jūs dabar turėtumėte išvysti *Select Data Source* ekraną. Pasirinkite *Machine Data Source* skirsnį ir paspauskite DSN sukonfigūruotą 4 žingsnelyje 4.
7. Lentelė turėtų būti perkelta į Virtualų Privatą Serverį į nurodytą MySQL. Kad tai patikrinti, prisijunkite prie Virtualaus Privataus Serverio per [Telnet ar SSH](#) ir susiraskite lentelę. Pvz.: taip gali atrodyti komanda:
`% mysql -u myusername new database name`

10.4.3. MySQL duomenų bazių importavimas į MS Access

Tai taip pat įmanoma importuoti MySQL duomenų bazes iš Virtualaus Privataus Serverio į MS Access duomenų bazę jūsų kompiuteryje.

1. Jūsų esamoje duomenų bazėje, eikite į *Tables* skirsnį ir spauskite dešinį pelės klavišą. Čia galėsite pasirinkti *Import*. Import ekrane pasirinkite *ODBC Databases* laukelyje *Files of type*.
2. Tai iškvies *Select Data Source* ekraną. Pasirinkite *Machine Data Source* skirsnį ir pasirinkite duomenų srauto vardą kurį jau teko naudoti.
3. Pasirodys *Import Objects* ekranas ir jūs galėsite pasirinkti galimas lenteles esančias Virtualiame Privačiame Serveryje. Pasirinkite norimą lentelę ir spauskite **OK**.

10.5. Priedas Nr.5

IS projekto testavimo rezultatai

10.5.1. Panaudojimo modelio testavimo rezultatai

Notebook: vmc Use Case Modeler (1) VMC_CASE Activity: Apmoketi kursas		
<input type="checkbox"/>	<input checked="" type="radio"/> Descriptions	no value for Description
<input type="checkbox"/>	<input checked="" type="radio"/> Properties	no value for Activity Type
<input type="checkbox"/>	<input checked="" type="radio"/> Properties	no value for Planning Horizon
<input type="checkbox"/>	<input checked="" type="radio"/> Properties	no value for Annual Cost
<input type="checkbox"/>	<input checked="" type="radio"/> Properties	no value for Unit of Work
<input type="checkbox"/>	<input checked="" type="radio"/> Properties	no value for InQueue Time Unit
<input type="checkbox"/>	<input checked="" type="radio"/> Properties	no value for InProcess Time Unit
<input type="checkbox"/>	<input checked="" type="radio"/> Properties	no value for Actual Process Time Unit
<input type="checkbox"/>	<input checked="" type="radio"/> Properties	no value for OutQueue Time Unit
<input type="checkbox"/>	<input checked="" type="radio"/> Properties	no value for InQueue Time
<input type="checkbox"/>	<input checked="" type="radio"/> Properties	no value for InProcess Time
<input type="checkbox"/>	<input checked="" type="radio"/> Properties	no value for Actual Process Time
<input type="checkbox"/>	<input checked="" type="radio"/> Properties	no value for OutQueue Time
<input type="checkbox"/>	<input checked="" type="radio"/> Properties	no value for Full Time Equivalent Headcount
Activity: Atsaukti kursas		
<input type="checkbox"/>	<input checked="" type="radio"/> Descriptions	no value for Description
<input type="checkbox"/>	<input checked="" type="radio"/> Properties	no value for Activity Type
<input type="checkbox"/>	<input checked="" type="radio"/> Properties	no value for Planning Horizon
<input type="checkbox"/>	<input checked="" type="radio"/> Properties	no value for Annual Cost

41 pav. Panaudojimo modelio testavimo rezultatai

10.1.2. Darbų sekų modelio testavimo rezultatai

Notebook: vmc Workflow Modeler (1) VMC_ryšiai Activity: Kursai		
<input type="checkbox"/>	<input checked="" type="radio"/> Descriptions	no value for Description
<input type="checkbox"/>	<input checked="" type="radio"/> Properties	no value for Activity Type
<input type="checkbox"/>	<input checked="" type="radio"/> Properties	no value for Planning Horizon
<input type="checkbox"/>	<input checked="" type="radio"/> Properties	no value for Annual Cost
<input type="checkbox"/>	<input checked="" type="radio"/> Properties	no value for Unit of Work
<input type="checkbox"/>	<input checked="" type="radio"/> Properties	no value for InQueue Time Unit
<input type="checkbox"/>	<input checked="" type="radio"/> Properties	no value for InProcess Time Unit
<input type="checkbox"/>	<input checked="" type="radio"/> Properties	no value for Actual Process Time Unit
<input type="checkbox"/>	<input checked="" type="radio"/> Properties	no value for OutQueue Time Unit
<input type="checkbox"/>	<input checked="" type="radio"/> Properties	no value for InQueue Time
<input type="checkbox"/>	<input checked="" type="radio"/> Properties	no value for InProcess Time
<input type="checkbox"/>	<input checked="" type="radio"/> Properties	no value for Actual Process Time
<input type="checkbox"/>	<input checked="" type="radio"/> Properties	no value for OutQueue Time
<input type="checkbox"/>	<input checked="" type="radio"/> Properties	no value for Full Time Equivalent Headcount
Activity: Kursai*		
<input type="checkbox"/>	<input checked="" type="radio"/> Descriptions	no value for Description
<input type="checkbox"/>	<input checked="" type="radio"/> Properties	no value for Activity Type
<input type="checkbox"/>	<input checked="" type="radio"/> Properties	no value for Planning Horizon

42 pav. Darbų sekų modelio testavimo rezultatai

10.1.3. BIM modelio testavimo palyginimo metodu rezultatai

Business Interaction (23)	Equivalent to target Business Interaction: Organization:
<input type="checkbox"/> Buhalterija (carrying Ataskaitos) ValdyMatch	Equivalent to target Business Interaction: Organization: Buhalterija (carrying At
<input type="checkbox"/> Buhalterija (carrying Saskaitos) BankMatch	Equivalent to target Business Interaction: Organization: Buhalterija (carrying S.
<input type="checkbox"/> Buhalterija (carrying Saskaitos) FizinaMatch	Equivalent to target Business Interaction: Organization: Buhalterija (carrying S.
<input type="checkbox"/> Buhalterija (carrying Saskaitos) ImoneMatch	Equivalent to target Business Interaction: Organization: Buhalterija (carrying S.
<input type="checkbox"/> Fiziniai_Asmenys (carrying Lesos) BankMatch	Equivalent to target Business Interaction: Market: Fiziniai_Asmenys (carrying L.
<input type="checkbox"/> Fiziniai_Asmenys (carrying Uzsakymai) Match	Equivalent to target Business Interaction: Market: Fiziniai_Asmenys (carrying U
<input type="checkbox"/> Imones (carrying Lesos) Bankas Match	Equivalent to target Business Interaction: Market: Imones (carrying Lesos) Ma
<input type="checkbox"/> Imones (carrying Uzsakymai) UzsakymMatch	Equivalent to target Business Interaction: Market: Imones (carrying Uzsakyma
<input type="checkbox"/> Kitos_Mokymo_Organizacijos (carryinMatch	Equivalent to target Business Interaction: Market: Kitos_Mokymo_Organizacij
<input type="checkbox"/> Kitos_Mokymo_Organizacijos (carryinMatch	Equivalent to target Business Interaction: Market: Kitos_Mokymo_Organizacij
<input type="checkbox"/> Marketingo_Reklamos_sk (carrying AMatch	Equivalent to target Business Interaction: Organization: Marketingo_Reklamos
<input type="checkbox"/> Marketingo_Reklamos_sk (carrying FMatch	Equivalent to target Business Interaction: Organization: Marketingo_Reklamos
<input type="checkbox"/> Marketingo_Reklamos_sk (carrying FMatch	Equivalent to target Business Interaction: Organization: Marketingo_Reklamos
<input type="checkbox"/> Programu_kurimas (carrying AtaskaitMatch	Equivalent to target Business Interaction: Organization: Programu_kurimas (ca
<input type="checkbox"/> Programu_kurimas (carrying Temos) MMatch	Equivalent to target Business Interaction: Organization: Programu_kurimas (ca
<input type="checkbox"/> Uzsakymu_skyrius (carrying AtaskaitMatch	Equivalent to target Business Interaction: Organization: Uzsakymu_skyrius (ca
<input type="checkbox"/> Valdyba (carrying Nurodymai) BuhalteMatch	Equivalent to target Business Interaction: Organization: Valdyba (carrying Nur
<input type="checkbox"/> Valdyba (carrying Nurodymai) MarketMatch	Equivalent to target Business Interaction: Organization: Valdyba (carrying Nur
<input type="checkbox"/> Valdyba (carrying Nurodymai) ProgranMatch	Equivalent to target Business Interaction: Organization: Valdyba (carrying Nur

43 pav. BIM modelio testavimo rezultatai

10.6. Priedas Nr.6 Projekto prototipo generavimo rezultatai

The image displays three screenshots of software forms used for data entry in a project prototype generation system.

ProVision Wo...: A form with a grid of input fields for various attributes:

Temos	ID
Imones	ASmKodas
Uzsakymai	OrgKodas
Muskinigdas_ asmuo	Data
Nuolaidos	Kaina
Klijantai	Miestai
Kasa	Fiziniai_asmuo

Uzsakymai: A form for order entry with the following fields:

- Nr (Number)
- Temos kodas_Temos (Topic code)
- Fiziniai_asmenys (Physical person)
- Imones kodas_Imones (Company code)
- Buttons: Uzsakymai, Uzsakymu Nr, Mokestis

Fiziniai_asmenys: A form for physical person data with the following fields:

- Asmenskodas_Klijantai (Person code)
- Miesto kodas_Miestai (City code)
- Uzsakymai (Order)
- Buttons: Pavarde, Adresas, Darboviete, Tel.Nr.

The 'Imones' form window has a blue title bar with the text 'Imones' and standard window controls. It features a text input field labeled 'Nr' on the left. To the right of the input field are two stacked buttons: 'Imones pavadinimas_Uzskumai' and 'Imones pavadinimas_Klientai'. At the bottom of the window, there is a row of five buttons: 'Tel.Nr', 'Adresas', 'Imones kodas', 'Pavadinimas', and 'Miestas'.

The 'Klientai' form window has a blue title bar with the text 'Klientai' and standard window controls. It features a text input field labeled 'Nr' on the left. To the right of the input field are four stacked buttons: 'Nuolaidos kodas_Nuolaidos', 'Pavarde, v_Fiziniai asmenys', '_Temos', and 'Imones kodas_Imones'. At the bottom of the window, there is a row of seven buttons: 'Pavarde', 'Tel.Nr', 'ASMENIS kodas', 'NUOLAIDOS kodas', 'Mokestis', 'Sumoketa', and 'Tema'.

44 pav. Projekto prototipo generavimo rezultatai