



**LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE**

**UZ METADATIEM BALSTĪTA LR CENTRĀLĀS
STATISTIKAS PĀRVALDES DATU APSKATES
SISTĒMA**

Kvalifikācijas darbs

Autors: Austra Jērāne

Stud. apl. aj09165

Darba vadītājs: Dr dat. Ģirts Karnītis

Rīga, 2011

Anotācija

Kvalifikācijas darbā izstrādātais programmatūras produkts ir uz metadatiem balstīta datubāzes saskarne, kura tiek veidota neatkarīgi no izvēlētas datubāzes struktūras.

Programmatūras galvenais mērķis ir SQL vaicājuma ģenerēšana balstoties uz lietotāja izvēlētiem datu avotiem un datu ierobežojumiem.

Darbā ir iekļauta programmatūras prasību specifikācija, programmatūras projektējuma apraksts, informācija par kvalitātes nodrošināšanu, konfigurāciju pārvaldību, projekta organizācijas pārvaldību, kā arī testēšanas dokumentācija un darbietilpības novērtējums. Pielikumā ir pievienots programmatūras kods.

Abstract

In qualification thesis deveoled tool is matadata based interface of database, which is developed independently of selected database structure. The main functionality of the software is a SQL query generation, based on user-selected data sources and data limitations.

The paper included the software requirement specification, sowftware design description, information on quality assurance, configuration management, project organization management, and testing documentation, and labor intensity assessment. Attached a software code.

Saturs

1	DEFINĪCIJAS, SAĪSINĀJUMI UN AKRONĪMI	6
2	PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA	8
2.1	IEVADS	8
2.1.1	<i>Nolūks</i>	8
2.1.2	<i>Darbības sfēra</i>	8
2.1.3	<i>Saistība ar citiem dokumentiem</i>	8
2.1.4	<i>Dokumenta pārskats</i>	8
2.2	VIŠPĀRĒJS APRAKSTS	9
2.2.1	<i>Produkta perspektīva</i>	9
2.2.2	<i>Sistēmas funkciju apraksts</i>	9
2.2.3	<i>Lietotāju raksturiezīmes</i>	10
2.2.4	<i>Vispārējie ierobežojumi</i>	10
2.2.5	<i>Pieņēmumi un atkarības</i>	10
2.3	FUNKCIONĀLĀS PRASĪBAS.....	10
2.3.1	<i>Moduļa pievienošana</i>	10
2.3.2	<i>Moduļa rediģēšana</i>	12
2.3.3	<i>Vērtību saglabāšana datubāzē</i>	13
2.3.4	<i>Reģistra moduļa dzēšana</i>	14
2.3.5	<i>Filtra attēlošana</i>	15
2.3.6	<i>Informācijas attēlošana kontroļos</i>	15
2.3.7	<i>Datu atlase</i>	16
2.3.8	<i>SQL teikuma ģenerēšana</i>	17
2.3.9	<i>Ierobežojumu loga attēlošana</i>	19
2.4	ĀRĒJĀ SASKARNE	20
2.4.1	<i>Lietotāja saskarnes prasības</i>	20
2.4.2	<i>Aparatūras saskarne</i>	21
2.4.3	<i>Programmatūras saskarne</i>	21
2.5	NEFUNKCIONĀLĀS PRASĪBAS.....	21
2.5.1	<i>Veiktspējas prasības</i>	21
2.5.2	<i>Datu drošība</i>	21
2.5.3	<i>Pieejamība</i>	22
2.5.4	<i>Aizsardzība pret nesakncionētu piekļuvi</i>	22
3	PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS	23
1.1.	IEVADS	23
3.1.1	<i>Nolūks</i>	23
3.1.2	<i>Darbības sfēra</i>	23
3.1.3	<i>Saistības ar citiem dokumentiem</i>	23
1.2.	DEKOMPOZĪCIJAS APRAKSTS	23
3.1.4	<i>Moduļu dekompozīcija</i>	23
3.1.5	<i>Datu dekompozīcija</i>	24
3.1.6	<i>Atkarības aprsksts</i>	27
4	KVALITĀTES NODROŠINĀŠANA	29

5	KONFIGURĀCIJAS PĀRVALDĪBA.....	30
6	PROGRAMMATŪRAS PROJEKTA ORGANIZĀCIJA.....	31
7	TESTĒŠANAS DOKUMENTĀCIJA.....	32
7.1	TESTĒŠANAS PROCESA APRAKSTS	32
7.2	SISTĒMAS TESTĒŠANA	32
7.2.1	Moduļa pievienošana	32
7.2.2	Moduļa rediģēšana	33
7.2.3	Vērtību saglabāšana datubāzē.....	33
7.2.4	Moduļa dzēšana	34
7.2.5	Filtra attēlošana	34
7.2.6	SQL teikuma ģenerēšana (WHERE daļa)	35
7.2.7	SQL teikuma ģenerēšana (SELECT un FROM daļa).....	37
7.2.8	Informācijas attēlošana kontroļos	38
7.2.9	Datu atlase.....	39
7.2.10	Ierobežojumu loga attēlošana.....	40
8	DARBIETILPĪBAS NOVĒRTĒJUMS.....	41
9	TRASĒJAMĪBAS TABULA.....	42
10	PIELIKUMS.....	43
10.1	EKRĀNFORMAS.....	43
10.1.1	Sadaļa “Moduļi”.....	43
10.1.2	Sadaļa “Dati”.....	44
10.1.3	Ierobežojumu logs.....	44
10.2	PROGRAMMAS KODS	45

1 Definīcijas, saīsinājumi un akronīmi

* izmantota tiešsaistē pieejama akadēmiskā terminu datubāze "AkadTerm"

(<http://termini.lza.lv/term.php?term=label&list=&lang=>)

Datubāzes administrators / administrators - persona vai personu grupa, kas atbild par datu bāzes uzturēšanu, ekspluatāciju, attīstību, drošību un integritāti.

Datu avots (no angļu: *Data source*) - datu apstrādes sistēmas komponents, kas ģenerē pārsūtāmos datus.

Datu kopa (no angļu: *Data set*) - fizikālo ierakstu apvienojums datu vidē, kam ir savs identifikators un kas atbilst vienam no datu apstrādes sistēmas, datu izvietojanas, glabāšanas un piekļuves organizācijas veidiem.

Datu režģis (no angļu: *Data grid*) – kvadrātveida laukums, kurā tiek attēloti dati tabulas formātā.

Izvēlnes rūtiņa (no angļu: *Check Box*) - kvadrātveida laukums, kam pierakstīts teksts, kas norāda iespējamo izvēli. Kad lietotājs izdarījis šo izvēli, tad izvēles rūtiņā parādās rakstzīme "x".

Kombinētais lodziņš (no angļu: *Combo Box*) - lodziņš, kas apvieno ievadlauka un sarakstlodziņa iespējas. Sarakstlodziņš satur izvēles, ko lietotājs var ritināt un atlasīt, lai aizpildītu ievadlauku.

Lietotājs – persona, kurai ir tiesības atlasīt/rediģēt datus un veidot atskaites.

LR CSP – Latvijas Republikas Centrālā Statistikas pārvalde.

Metadati (no grieķu *meta* + latīņu *data*) jeb "dati par datiem" ir informācija, kas apraksta kādu citu datu kopu.

PPA – programmatūras projektējuma apraksts.

PPS – programmatūras prasību specifikācija.

Radiopogas (no angļu: *Radio button*) – pogu virkne ekrānā. Tā pieļauj tikai vienas pogas izvēli. Ja kāda poga tiek izvēlēta, citas pogas izvēle tiek automātiski likvidēta.

Ritjosla (no angļu: *Scroll Bar*) - taisnstūrveida josla loga ietvara labajā vai apakšējā malā, ko izmanto loga satura pārvietošanai vertikālā vai horizontālā virzienā.

Sarakstlodziņš (no angļu: *List Box*) - lodziņš ar ritināmām izvēlēm, no kurām lietotājs var atlasīt vienu.

Savienojuma virkne (no angļu: *Connection String*) – ar semikolu atdalīta argumentu virkne, kas nosaka datubāzes atrašanās vietu un savienojuma izveidošanas noteikumus.

Spiedpoga (poga) - lielāka izmēra poga, kas dialoglodziņā iniciē lietotāja izvēlētas darbības.

SQL (no angļu: *Structured Query Language* "strukturēto vaicājumu valoda") - vaicājumu valoda, kas paredzēta datu manipulēšanai relāciju datubāžu pārvaldības sistēmās.

SQL vaicājums (no angļu: *SQL query*) - instrukciju kopa, ko izmanto darbam ar datiem. Vaicājumu palaiž, lai izpildītu šīs instrukcijas.

Teksta lauks (no angļu: *Text Box*) - lodziņš displeja ekrānā, kas iezīmē laukumu, kurā lietotājs var ievadīt tekstu.

Ziņojuma lodziņš/dialoglodziņš (no angļu: *Message Box*) – lodziņš, kas parādās displeja ekrānā kādas programmas izpildes gaitā un parasti informē par kļūdām vai brīdina par lietotāja veikto darbību iespējamām sekām.

2 Programmatūras prasību specifikācija

2.1 Ievads

2.1.1 Nolūks

Šis dokuments paredzēts programmatūras prasību specificēšanai un to detalizētai aprakstīšanai. Programmproduktā tiks realizētas tikai un vienīgi tās prasības, kas ir iekļautas šajā dokumentā.

Dokumenta mērķauditorija ir programmatūras izstrādātājs, kā arī programmatūras pasūtītājs. Šajā dokumentā aprakstītās prasības tiks izmantotas par pamatu rakstot programmatūras projektējuma aprakstu (turpmāk PPA).

2.1.2 Darbības sfēra

Izstrādājamā programmatūra ir datubāzes saskarne, kura ļauj integrēt jaunus reģistru modulus un atlasīt datus no iepriekš norādītiem datu avotiem. Programmatūrai būtu jādarbojas neatkarīgi no tā, kādas datubāzes tai ir piesaistītas. Izstrādājamā sistēma ir apakšsistēma lielākai sistēmai, kurā tiks iekļauta arī lietotāju autentifikācija, atskaišu veidošana un citas funkcijas. Lai nodrošinātu sistēmas darbību, tiks izmantota atsevišķa datubāze, kurā glabāsies sistēmas informācija (piemēram, informācija par piesaistītajām datubāzēm, izveidotajiem moduļiem u.c.).

2.1.3 Saistība ar citiem dokumentiem

Šis dokuments veidots, vienojoties izstrādātājam un pasūtītājam, precizējot sistēmas aprakstu un programmatūras tvērumu. Šis dokuments tiks ņemts par pamatu programmatūras projektējuma aprakstam un lietotāja dokumentācijai. Dokumenta sastādīšanā izmantots standarts LVS 68:1996 „Programmatūras prasību specifikācijas ceļvedis”.

2.1.4 Dokumenta pārskats

Dokuments organizēts 5 nodaļās.

1. Ievads - īsumā aprakstīts dokumenta mērķis, kā arī izstrādājamās sistēmas galvenā koncepcija.

2. Vispārējais apraksts – ietver informāciju par izstrādājamā produkta vispārējo funkcionalitāti, kā arī produkta perspektīvu un lietotāju raksturiezīmēm. Apkopota informācija arī par sistēmas ierobežojumiem, pieņēmumiem un atkarībām.
3. Funkcionālās prasības – satur detalizētu informāciju par visām sistēmas funkcijām, ievaddatiem, to pārbaudi, operāciju secību un rācību uz nepardzētām (kļūdu) situācijām.
4. Ārējā saskarne – apkopota informācija par lietotāja saskarni, aparatūras saskarni, programmatūras saskarni un sakaru saskarni.
5. Nefunkcionālās prasības – satur informāciju par datu drošību, pieejamību un aizsardzību pret nesankcionētu piekļuvi kā arī veiktspējas prasībām.

2.2 Vispārējs apraksts

Šajā sadaļā aprakstītas vispārīgās prasības sistēmai.

2.2.1 Produkta perspektīva

Sistēma tiek izstrādāta LR Centrālās Statistikas pārvaldes vajadzībām.

Sistēma tiek veidota kā lielākas sistēmas daļa. Tā ir cieši saistīta ar citu informācijas sistēmu (reģistru) darbu, un izmanto tajās reģistrētos datus.

Programmprodukts izmanto atsevišķu datubāzi, kurā tiek saglabāta sistēmas informācija, taču daļa no sistēmai nepieciešamās informācijas tiek ņemta no ārējās datubāzes. Sistēma ir atkarīga no SQL Server darbības, kā arī no severa, uz kura tā ir uzstādīta, darbības.

2.2.2 Sistēmas funkciju apraksts

2.2.2.1 Moduļu pārvaldība

Lai varētu darboties ar datubāzes datiem, nepieciešams veidot moduļus, kuros jānorāda, ar kuriem datubāzes objektiem lietotājs darbosies. Jānodrošina arī moduļu rediģēšana un dzēšana. Rediģējot moduli nepieciešams precīzi attēlot izveidotā moduļa shēmu un jābūt iespējai tam pievienot jaunas tabulas, kolonnas un noņemt iepriekš pievienotās. Pievienojot vai rediģējot moduli nepieciešams nodrošināt to, lai naverētu tikt pievienotas tabulas, kurās nav piesaistīta neviena kolonna, kā arī to, lai nevarētu tikt pievienoti moduļi ar vienādiem nosaukumiem. Rediģējot moduli

nosaukums drīkst tikt mainīts, taču tas nedrīkst sakrist ar citu, jau pievienotu moduļa nosaukumu.

2.2.2.2 Datu atlase

Lai veiktu datu atlasīšanu, nepieciešams norādīt kurš modulis tiks izmantots. Izvēloties moduli jābūt iespējai norādīt parametrus pēc kuriem atlasīt datus. Jābūt iespējai izmantot Microsoft SQL Server Management Studio vidē pieejamos operatorus (“>”, “<”, “<>”, “=”, “>=”, “<=”, NOT, AND, OR, BETWEEN, LIKE) un speciālos simbolus (“%” un “_”). Ja izvēlētajā tabulā satur ārējās atslēgas, tad jābūt iespējai tās aizstāt ar datiem. Jābūt iespējai apskatīt sistēmas izveidoto SQL pieprasījumu.

2.2.3 Lietotāju raksturiezīmes

Tā kā sistēma ir paredzēta tikai reģistrētiem lietotājiem, tad netiek izskatīta iespēja, ka sistēmu varētu izmantot neregistrēts lietotājs. Tiek pieņemts, ka sistēmas lietotāju zināšanas informācijas tehnoloģijās ir labas un sistēmas administrators (kurš pārvalda sistēmas lietotājus) nav nepieciešams, kā arī nav nepieciešami sīki paskaidrojumi par sistēmas darbību. Tiek pieņemts, ka lietotājs, kurš darbojas ar sistēmu, labi pārzina SQL Server sintaksi.

2.2.4 Vispārējie ierobežojumi

Sistēmai jādarbojas uz visām Windows versijām sākot ar Windows2000.

2.2.5 Pieņēmumi un atkarības

Tiek pieņemts, ka sistēmas lietotāji pārzina SQL Server darbības principus, kā arī SQL pieprasījumu sintaksi.

2.3 Funkcionālās prasības

Šajā sadaļā aprakstītas konkrētas prasības sistēmai un tās funkcionalitātei.

2.3.1 Moduļa pievienošana

Identifikators: F.1.3.1.
Mērķis
Funkcija tiks izmantota, lai pievienotu jaunu moduli no reģistra.
Ievaddati

Poga "Jauns"

Poga "Saglabāt"

Kombinētais lodziņš "Datubāze"

Moduļa nosaukums 1-50 simboli (latīņu alfabēta lielie un mazie burti, cipari 0-9)

Izvēlnes rūtiņa "Atzīmēt visu"

Koka skata kontrolis (ar izvēlnes rūtiņām)

Sarakstlodziņš "Moduļi"

Apstrāde

Nospiežot pogu "Jauns"

- kombinētais lodziņš tiek attēlots kā redzams;
- no metadatiem tiek ielasīti datubāzu nosaukumi;
- datubāzu nosaukumi tiek attēloti kombinētajā lodziņā, no kura iespējams izvēlēties vienu datubāzi;

Izvēloties vērtību no kombinētā lodziņa koka skata kontrolī (Tree View) tiek attētotas datubāzē esošās tabulas. Izvēloties tabulu tiek ielasītas tabulā esošās kolonnas, kuras iespējams atzīmēt, lai pievienotu modulim. Ja neviena kolonna nav atzīmēta, tad atzīme tiek izņemta arī pie tabulas, savukārt atkārtoti atzīmējot kādu tabulu, tiek atzīmētas visas tai piesaistītās kolonnas. Šādi tiek nodrošināts tas, ka tabulas bez piesaistītām kolonnām nevar tikt pievienotas.

Atzīmējot izvēlnes rūtiņu "Atzīmēt visu", tiek atzīmētas visas koka skatā ielasītās tabulas un kolonnas. Izņemot atzīmi šajā rūtiņā, atzīme pie visām tabulām un kolonnām tiek atcelta.

Nospiežot pogu "Saglabāt"

- tiek pārbaudīts, vai ir ievadīts nosaukums. Ja nē, tad izvada kļūdas paziņojumu #1. Ja kļūdas paziņojums netiek atgriezts, tad pāriet pie nākamā punkta;
- tiek pārbaudīts, vai ievadītais nosaukums ir atbilstošā garumā. Ja nav, tad tiek izvadīts kļūdas paziņojums #2. Ja kļūdas paziņojums netiek atgriezts, tad tiek izpildīta nākamā darbība;
- tiek pārbaudīts, vai sistēmas datubāzē šāds nosaukums neeksistē. Ja eksistē, tad tiek izvadīts kļūdas paziņojums #3. Ja kļūdas paziņojums netiek atgriezts, tad pilda nākamo darbību;
- tiek pārbaudīts vai ir atzīmēta vismaz viena tabula. Ja nav, tad tiek izvadīts kļūdas paziņojums #4. Ja kļūdas paziņojums netiek izvadīts, tiek izpildīts

<p>nākamais punkts;</p> <ul style="list-style-type: none"> • sistēmas datubāzē tiek saglabāts moduļa nosaukums, tam piesaistītas tabulas un kolonnas un datubāzes nosaukums, kurai pieder attiecībie datubāzes objekti. • sarakstlodziņam tiek pievienots jauns modulis. • tiek izvadīts apstiprinošs paziņojums. • Tiek izsaukta funkcija „Informācijas attēlošana kontroljos”
Izvad dati
<p>Ziņojuma lodziņš “Modulis ir pievienots!” Pievienots ieraksts sarakstlodziņā „Moduļi”</p>
Kļūdu paziņojumi
<ol style="list-style-type: none"> 1. “Lūdzu ievadiet nosaukumu!” 2. “Nosaukums nedrīkst būt garāks par 50 simboliem!” 3. “Šāds nosaukums jau eksistē!” 4. “Nav atzīmēta neviena tabula!”

2.3.2 Moduļa rediģēšana

Identifikators: F.1.3.2.
Mērķis
Funkcija tiks izmantota, lai rediģētu iepriekš izveidotu moduli.
Ievaddati
<p>Sarakstlodziņš „Moduļi” Moduļa nosaukums 1-50 simboli (latīņu alfabēta lielie un mazie burti, cipari 0-9) Poga “Saglabāt” Koka skata kontrolis</p>
Apstrāde
<p>Izvēloties moduli no sarakstlodziņa</p> <ul style="list-style-type: none"> • koka skatā tiek attēlotas visas tabulas un kolonnas, kuras pieder modulim piesaistītajai datubāzei; • tabulām un kolonnām, kuras ir piesaistītas modulim, izvēlnes rūtiņa ir atzīmēta. <p>Nospiežot pogu “Saglabāt”</p> <ul style="list-style-type: none"> • tiek pārbaudīts, vai ir ievadīts nosaukums. Ja nē, tad izvada kļūdas paziņojumu

<p>#1. Ja kļūdas paziņojums netiek atgriezts, tad pāriet pie nākamā punkta;</p> <ul style="list-style-type: none"> • tiek pārbaudīts, vai ievadītais nosukums ir atbilstošā garumā. Ja nav, tad tiek izvadīts kļūdas paziņojums #2. Ja kļūdas paziņojums netiek atgriezts, tad tiek izpildīta nākamā darbība; • ja nosaukums ir ticis izmainīts, tad tiek pārbaudīts, vai sistēmas datubāzē šāds nosaukums neeksistē. Ja eksistē, tad tiek izvadīts kļūdas paziņojums #3. Ja kļūdas paziņojums netiek atgriezts, tad pilda nākamo darbību; • tiek pārbaudīts vai ir atzīmēta vismaz viena tabula. Ja nav, tad tiek izvadīts kļūdas paziņojums #4. Ja kļūdas paziņojums netiek izvadīts, tiek izpildīts nākamais punkts; • sistēmas datubāzē tiek saglabāts rediģētais moduļa nosaukums; • ja ir atzīmēta tabula vai kolonna, kura vēl nav sistēmas datubāzē, tad tā tiek pievienota; • starptabulā tiek saglabāta jaunā datubāzes shēma; • tiek izvadīts apstiprinošs paziņojums • tiek izsaukta funkcija „Informācijas attēlošana kontrolēs”
Izvaddati
Ziņojuma lodziņš “Modulis ir saglabāts” Atjaunota informācija sarakstlodziņā „Moduļi”
Kļūdu paziņojumi
1. “Lūdzu ievadiet nosaukumu!” 2. “Nosaukums nedrīkst būt garāks par 50 simboliem!” 3. “Šāds nosaukums jau eksistē!” 4. “Nav atzīmēta neviena tabula!”

2.3.3 Vērtību saglabāšana datubāzē

Identifikators: F.1.3.3.
Mērķis
Tiks izmantota, lai pievienojot vai rediģējot moduli, saglabātu pievienoto objektu shēmu.
Ievaddati
Moduļa nosaukums Datubāzes nosaukums

Tabulas nosaukums
Kolonnas nosaukums
Apstrāde
Katrai tabulai un katrai kolonnai tiek pārbaudīts, vai tā ir atzīmēta <ul style="list-style-type: none"> • ja ir atzīmēta, tad pārbauda, vai šāda tabula (vai kolonna) jau eksistē datubāzē; • ja eksistē, tad tiek pievienots jauns ieraksts datubāzes starptabulā; • ja neeksistē, tad tā tiek pievienota datubāzei, kā arī tiek pievienots jauns ieraksts starptabulā.
Izvaddati
Tabulas sistēmas datubāzē ar attiecīgajiem ierakstiem.
Kļūdu paziņojumi

2.3.4 Reģistra moduļa dzēšana

Identifikators: F.1.3.4.
Mērķis
Funkcija tiks izmantota, lai dzēstu iepriekš izveidotu moduli.
Ievaddati
Poga “Dzēst” Sarakstlodziņš “Moduļi” Koka skata kontrolis
Apstrāde
Nospiežot pogu “Dzēst” <ul style="list-style-type: none"> • tiek pārbaudīts, vai ir atzīmēts kāds modulis. Ja nav, tad izvada kļūdas paziņojumu #1. Ja ir atzīmēts, tad izpilda nākamās darbības; • tiek dzēstas vērtības no starptabulas (moduļa ID, tabulas ID, kolonnas ID); • tiek dzēstas vērtības no tabulas, kura satur informāciju par pievienotajiem moduļiem; • tiek izvadīts apstiprinošs paziņojums; • no sarakstlodziņa tiek dzēsts atzīmētais modulis.
Izvaddati
Paziņojums “Modulis ir dzēsts”

Kļūdu paziņojumi
1. "Nav atzīmēts modulis!"

2.3.5 Filtra attēlošana

Identifikators: F.1.3.5.
Mērķis
Funkcija tiks izmantota, lai pievienotam modulim būtu iespējams norādīt datu atlasē kritērijus. Funkcija izveido teksta laukus, kuros iespējams norādīt atlasē nosacījumus.
Ievaddati
Moduļa nosaukums un tabulas (vai kolonnas) nosaukums
Apstrāde
<p>Izvēloties objektu no moduļu koka, tiek pārbaudīts</p> <ul style="list-style-type: none"> vai izvēlētais objekts ir tabula. Ja ir, tad katrai tabulai piesaistītajai kolonnai tiek izveidots izveidots jauns kontrolis, kas sastāv no kolonnas nosaukuma un teksta lauka; ja izvēlētais objekts ir kolonna, tad tiek pievienots viens kontrolis, kurš atbilst izvēlētajai kolonnai. <p>Izvēloties citu objektu, iepriekš pievienotie kontroļi tiek dzēsti.</p> <p>Ja nepieciešams pievienot vairāk kā desmit kontroļus, tad tie tiek izvietoti vairākos stabiņos (katrā ne vairāk kā desmit) .</p>
Izvaddati
Tiek attēlots filtrs, kas atbilst izvēlētajam objektam un sastāv no kolonnu nosaukumiem un teksta laukiem, kuros iespējams rakstīt SQL WHERE nosacījumu.
Kļūdu paziņojumi
Ja struktūra datubāzei, no kuras nepieciešams atlasīt datus, ir mainīta pēc moduļa izveidošanas, tad var tikt atgriezts atbilstošs SQL Server kļūdas paziņojums.

2.3.6 Informācijas attēlošana kontroļos

Identifikators: F.1.3.6.
Mērķis
Funkcija tiks izmantota, lai uzreiz pēc moduļa pievienošanas/ dzēšanas/ rediģēšanas tiktu atjaunota informācija kontroļos, kuros tiek attēloti moduļu nosaukumi un saturs.

Ievaddati
Kontroļa nosaukums
Apstrāde
<p>Izsaucot funkciju tiek</p> <ul style="list-style-type: none"> • nodzēsti visi tam iepriekš pievienotie elementi; • no datubāzes nolasīti visi moduļu nosaukumi un to shēma; • tiek pievienoti moduļu nosaukumi, tabulas, kolonnas.
Izvaddati
Kontrolis, ar elementiem, kas atbilst datubāzē saglabātajām vērtībām.
Kļūdu paziņojumi

2.3.7 Datu atlase

Identifikators: 1.3.7.
Mērķis
Funkcija tiks izmantota, lai atlasītu datus atbilstoši lietotāja norādītajiem parametriem.
Ievaddati
<p>Poga “Atlasīt”</p> <p>Poga “SQL pieprasījums”</p> <p>Poga “Aizstāt ārējās atslēgas”</p>
Apstrāde
<p>Izvēloties objektu no moduļu koka tiek izsaukta funkcija “SQL teikuma ģenerēšana”, kuras atgrieztie dati tiek izvadīti datu režģī</p> <ul style="list-style-type: none"> • ja izvēlētais objekts ir tabula, tad tiek atlasītas visas kolonnas, kuras šajā modulī ir pievienotas izvēlētajai tabulai; • ja izvēlētais objekts ir kolonna, tad tiek atlasīti visi dati, kurus satur šī kolonna; • tiek ģenerēts SQL teikums, kurš sastāv no SELECT un FROM daļām. <p>Nospiežot pogu “Atlasīt” tiek izsaukta funkcija “SQL teikuma ģenerēšana” tāpat kā augstāk minēts, taču SQL teikumam tiek pievienota WHERE daļa ievērojot filtrā norādītos nosacījumus.</p> <p>Datu režģī nospiežot peles kreiso taustiņu uz kolonnas nosaukuma, dati tiek sakārtoti</p>

augošā secībā pēc šīs kolonnas. Nospiežot atkārtoti dati tiek sakārtoti pretējā secībā. Nospiežot pogu “SQL pieprasījums” tiek atvērts jauns logs ar uzģenerēto SQL teikumu.

Nospiežot pogu “Aizstāt ārējās atslēgas” tiek atvērts logs ar kolonnu nosaukumiem, kuri var tik lietoti, lai aizstātu tabulā esošās ārējās atslēgas. Atkārtoti nospiežot pogu, logā tiek attēlotas tikai tie kolonnu nosaukumi, kuri vēl nav pievienoti datu režģī. Ja tabulā nav nevienas ārējās atslēgas, tad logā netiek attēlots neviens ieraksts.

Ja datu režģī nav datu pogas “Aizstāt ārējās atslēgas” nospiešanas brīdī, tad tiek izdrukāts kļūdas paziņojums #1.

Izvaddati

Datu režģī tiek attēloti atlasītie dati atbilstoši uzģenerētajam SQL teikumam.

Kļūdu paziņojumi

1. “Nav datu!”

Ja struktūra datubāzei, no kuras nepieciešams atlasīt datus, ir mainīta pēc moduļa izveidošanas, tad var tikt atgriezts atbilstošs SQL Server kļūdas paziņojums.

2.3.8 SQL teikuma ģenerēšana

2.3.8.1 SELECT un FROM daļas ģenerēšana

Identifikators: 1.3.8.

Mērķis

Funkcija tiks izmantota, lai lietotāja ievadītos parametrus datu atlasei pārveidotu par SQL teikumu. SQL teikums nedrīkst atgriezt kļūdas paziņojumu.

Ievaddati

Poga “Atlasīt”

Poga “SQL teikums”

Apstrāde

Izvēloties no moduļu koka vai nospiežot pogu “Atlasīt”

- no datubāzē saglabātās moduļa shēmas tiek noasīts datubāzes nosaukums, kurai pieder modulim piesaistītās tabulas;
- atbilstoši datubāzes nosaukumam tiek izveidota savienojuma virkne (*connection string*);
- no datubāzes tiek nolasīta moduļa shēma;

- izmantojot tabulas nosaukumu, tiek ģenerēta FROM daļa;
- SELECT daļa tiek aizpildīta izmantojot kolonnu nosaukumus, kuri piesaistīti izvēlētajai daļai. Moduļa pievienošanas funkcijā jābūt novērstam tam, ka var tikt pievienota tabula bez piesaistītām kolonnām;
- ja kāda no ārējām atslēgām ir aizstāta ar datiem, tad tiek ģenerēta JOIN daļa, nolaset attiecīgās vērtības no tabulas, kurā glabājas informācija par ierobežojumiem (*constraints*).

Ja funkcija tiek izsaukta nospiežot pogu, tad

- katram no filtrā esošajiem teksta laukiem tiek pārbaudīts, vai tajā ir ievadīta kāda vērtība. Ja vērtība ir ievadīta, tad tiek pārbaudīts, vai tā satur kādu no iepriekš definētajiem operatoriem;
- katram teksta laukam, kurā ir ievadīta kāda vērtība, tiek izsaukta funkcija “WHERE daļas ģenerēšana”

Datu režģī tiek attēloti SQL pieprasījuma atgrieztie dati.

Izvaddati

Datu tabula atbilstoši izveidotajam SQL teikumam.

Kļūdu paziņojumi

Ja struktūra datubāzei, no kuras nepieciešams atlasīt datus, ir mainīta pēc moduļa izveidošanas, tad var tikt atgriezts atbilstošs SQL Server kļūdas paziņojums.

2.3.8.2 WHERE daļas ģenerēšana

Identifikators: 1.3.9

Mērķis

Funkcija tiks izmantota, lai lietotāja ievadītos parametrus datu atlasei pārveidotu par SQL teikumu. SQL teikums nedrīkst atgriezt kļūdas paziņojumu.

Ievaddati

Poga “Atlasīt”

Poga “SQL teikums”

Apstrāde

Izsaucot šo funkciju tiek pārbaudīts, vai teksta lauks satur kādu no iepriekš definētiem operatoriem. Ja nesatur, visi operandi tiek sasaistīti ar operatoru “and”.

- ja satur operators “>”, “<”, “<>”, “<=”, “>=”, tad tiek ģenerēts WHERE nosacījums izmantojot ievadīto operatoru. Ja ievadītā vērtība neatbilst SQL

<p>sintaksei, tiek atgriezts kļūdas paziņojums #1.</p> <ul style="list-style-type: none"> • ja satur operatorus “or” vai “not” tad SQL pieprasījums tiek ģenerēts atbilstoši SQL sintaksei, izmantojot ievadīto operatoru. Ja vērtības ir ievadītas vairākos teksta laukos, tad tās kopā tiek sasaistītas ar operatoru “and” • ja satur operatorus “%”, „_” vai “between”, tad WHERE nosadījumam tiek pievienots operators “like”. Ja ievadītā vērtība satur operatoru “between”, tad tiek pārbaudīts, vai ir ievadīti tieši divi operandi un starp tiem operators “and”. Ja nav, tad tiek atgriezts kļūdas paziņojums #2. <p>Nospiežot pogu “SQL teikums”</p> <ol style="list-style-type: none"> 1. iespējams apskatīt SQL teikumu.
Izvaddati
Datu tabula atbilstoši izveidotajam SQL teikumam.
Kļūdu paziņojumi
<ol style="list-style-type: none"> 1. “Kļūda!” 2. “Nosacījumam jābūt formā 'BETWEEN aaa AND zzz'”

2.3.9 Ierobežojumu loga attēlošana

Identifikators: F.1.3.10.
Mērķis
Funkcija tiks izmantota, attēlotu logu, kurā lietotājs var izvēlēties ar kādām kolonnām aizstāt ārējo atslēgu kolonnas.
Ievaddati
<p>Datu tabula</p> <p>Datu kolonna</p> <p>Poga “Aizstāt ārējās atslēgas”</p> <p>Poga “Apstiprināt”</p>
Apstrāde

Nospiežot pogu “Aizstāt ārējās atslēgas”

- katrai datu režģī pievienotajai kolonnai tiek pārbaudīts, vai tā satur ārējās atslēgas, ja satur, tad no datubāzes tiek nolasīts atbilstošās primārās atslēgas tabulas un kolonnas nosaukums.
- ierobežojumu logā tiek izdrukāts ārējās atslēgas kolonnas nosaukums un kolonnas (kā izvēlnes rūtiņas) no tabulas, kurā atrodas primārā atslēga (izņemot kolonnu, kura atbilst primārajai atslēgai).

Ja poga “Aizstāt ārējās atslēgas” tiek nospiesta atkārtoti tai pašai tabulai, tad jau pievienotās kolonnas netiek attēlotas ierobežojumu logā.

Nospiežot pogu “Apstiprināt”

- datu režģim tiek pievienotas izvēlētās kolonnas.
- filtram tiek pievienoti elementi atbilstoši izvēlētajām kolonnām.
- No ierobežojumu loga tiek noņemtas izvēlnes rūtiņas, kuras ir atzīmētas.

Izvaddati

Datu režģis ar lietotāja izvēlētajām kolonnām.

Filtrs ar elementiem, kas atbilst datu režģī esošajām kolonnām.

Kļūdu paziņojumi

Ja struktūra datubāzei, no kuras nepieciešams atlasīt datus ir mainīta pēc moduļa izveidošanas, tad var tikt atgriezts atbilstošs SQL Server kļūdas paziņojums.

2.4 Ārējā saskarne

2.4.1 Lietotāja saskarnes prasības

SQL vaicājumu ģenerēšana izmantojot lietotāja saskarni ir sistēmas galvenā pamatfunkcija. Ar lietotāju saskarnes palīdzību jāspēj uzģenerēt SQL vaicājumu, kurš satur šādus SQL valodas pamatvārdus:

- SELECT;
- FROM ;
- WHERE;
- JOIN.

Elementi formā jāizvieto tā, lai lietojot taustiņu ‘Tab’ pārejai uz nākošo elementu, to apietu virzienā no kreisās uz labo pusi un no augšas uz leju. Vadības spiedpogas izvieto loga kreisajā pusē.

Sistēmas galvenajam logam jāsaturs izvēlņu joslu. Tas var saturēt arī ritjoslu. Izvēlņu hierarhijā nevar būt vairāk kā trīs līmeņi. Vienai izvēlei jāsaturs vismaz divas izvēles. Spiedpogu apzīmējumiem jālieto darbības vārda nenoteksme, piemēram, ‘saglabāt’, ‘apstiprināt’. Visām pogām un izvēlnēm jābūt pieejamām ne tikai ar peli, bet arī ar klaviatūras palīdzību, izmantojot taustiņu ‘Tab’, pie tam pieejas secībai jābūt funkcionāli loģiskai, vēlams no kreisās uz labo pusi un no augšas uz leju. Kļūdu paziņojumi var tikt izvadīti gan ziņojumu lodziņā, gan formā, ar kuru notiek darbība. Vēlams augšējā malā, zem izvēlņu joslas. Tekstam jābūt sarkanā krāsā.

Lietotāja saskarnei jāsaturs no šādām sadaļām:

- Dati (apakšsadaļas: moduļu koks, filtrs, datu režģis) – iespējams veikt visas nepieciešamās darbības ar datiem.
- Moduļi – iespējams pievienot, rediģēt, dzēst moduļus.

2.4.2 Aparatūras saskarne

Sistēmai jānodrošina ne vairāk kā 25 lietotāju vienlaicīgs pieslēgums.

2.4.3 Programmatūras saskarne

Lai varētu darboties ar sistēmu ir jābūt reģistrētam lietotājam SQL serverī, jo sistēma izmanto SQL servera lietotājvārdu un paroli. Reģistrēties, izmantojot sistēmu, nav iespējams.

2.5 Nefunkcionālās prasības

2.5.1 Veiktspējas prasības

Sistēmas veiktspējas prasības tiek noteiktas sistēmas kopējā datu apjoma apstrādei, lietotāju pieprasījumu apstrādei un datu apmaiņai ar citām sistēmām.

Lietotāju veiktos datu pieprasījumus, izņemot datu atlases (meklēšanas) pieprasījumus, sistēma apkalpo triju sekunžu laikā. No tiem 90% pieprasījumu sistēma apkalpot vienas sekundes laikā.

Sistēmas darbības ātrumu ietekmē SQL Server darbības ātrums un apstrādājamo datu apjoms.

2.5.2 Datu drošība

Tā kā izstrādājamā sistēma ir paredzēta, lai atvieglotu lietotāju darbu ar datiem, tad rezerves kopiju veidošana nav iekļauta sistēmas funkcijās. Tas tiek darīts ārpus šīs sistēmas.

Datu drošība tiek nodrošināta jau SQL Server līmenī, līdz ar to šajā sistēmā nav vajadzības ieviest papildus drošības atribūtus.

2.5.3 Pieejamība

Sistēmai jābūt pieejamai jebkurā laikā, uz jebkura datora, uz kura tā ir uzstādīta. Tā pieejama tikai lietotājiem, kuriem ir piešķirts SQL Server lietotājvārds un parole.

2.5.4 Aizsardzība pret nesankcionētu piekļuvi

Sistēmu paredzēts izmantot tikai uzņēmuma iekšienē, līdz ar to uzņēmums, kurš izmanto šo sistēmu pats ir atbildīgs par aizsardzību pret nesankcionētu piekļuvi.

3 Programmatūras projektējuma apraksts

1.1. Ievads

3.1.1 Nolūks

Šis dokuments ir paredzēts, lai aprakstītu sistēmas projektējuma informāciju. Dokumenta uzdevums ir atvieglot sistēmas projektēšanu un nodrošināt projektējuma pieeju gan sistēmas izstrādātājiem, gan uzturētāju pārstāvjiem.

3.1.2 Darbības sfēra

Izstrādājamā programmatūra ir datubāzes saskarne, kura ļauj darboties ar datubāzes datiem, neatkarīgi no tās struktūras. Tajā ir paredzēts nodrošināt datu atlasīšanu, izmantojot iepriekš definētus moduļus, kuros tiek norādīts, ar kādiem datubāzes objektiem notiks darbs. Lai to nodrošinātu, tiks izmantota atsevišķa datu bāze, kurā tiks glabāta informācija par moduļiem un tiem piesaistītajām tabulām.

3.1.3 Saistības ar citiem dokumentiem

Programmatūras projektējuma apraksts ir izstrādāts ievērojot programmatūras prasību specifikāciju. Programmatūras projektējuma apraksts ir noformēts ievērojot standarta LVS 72:1996 Ieteicamā prakse programmatūras projektējuma rakstīšanai prasības.

1.2. Dekompozīcijas apraksts

3.1.4 Moduļu dekompozīcija

3.1.4.1 Modulis "MY_DATA"

Identifikators:	M.2.1.4.1.
Funkcijas:	<ul style="list-style-type: none">• moduļa pievienošana• vērtību saglabāšana datubāzē• moduļa rediģēšana• moduļa dzēšana
Apraksts:	Šajā modulī esošās funkcijas nodrošina jauna reģistru moduļa integrēšanu sistēmā, kā arī tā rediģēšanu un dzēšanu.

3.1.4.2 Modulis “DATA”

Identifikators:	M.2.1.4.2.
Funkcijas:	<ul style="list-style-type: none">• informācijas attēlošana moduļu sarakslodziņā• informācija attēlošana moduļu koka skata kontrolī• informācijas attēlošana tabulu koka skata kontrolī• moduļa shēmas attēlošana
Apraksts:	Šajā modulī esošās funkcijas nodrošina informācijas attēlošanu un atjaunošanu kontroļos.

3.1.4.3 Modulis “SQL_QUERY”

Identifikators:	M.2.1.4.3.
Funkcijas:	<ul style="list-style-type: none">• SQL teikuma SELECT daļa• SQL teikuma FROM daļa• SQL teikuma WHERE daļa
Apraksts:	Šajā modulī esošās funkcijas nodrošina SQL teikuma ģenerēšanu.

3.1.4.4 Modulis “INTERFACE”

Identifikators:	M.2.1.4.4.
Funkcijas:	<ul style="list-style-type: none">• filtra izveidošana• ierobežojumu loga attēlošana• izvēlnes rūtiņu pievienošana• iezīmju (label) pievienošana
Apraksts:	Šajā modulī esošās darbības veic visas nepieciešamās darbības, lai dinamiski tiktu attēlota lietotāja saskarne.

3.1.5 Datu dekompozīcija

3.1.5.1 Tabula “my_modules”

Tabula paredzēta, lai glabātu datus par izveidotajiem moduļiem. Lauks “module_id” tiek ģenerēts automātiski. Lauks “name” tiek nolasīts no lietotāja ievadītās vērtības. Ja modulis tiek rediģēts, tad “name” var tikt mainīts. Datubāzes nosaukums tiek saglabāts izmantojot vērtību, kuru lietotājs ir izvēlējis kabinētajā lodziņā, kas atbilst datubāzēm. Dzēšot moduli ieraksts tiek dzēsts fiziski.

Lauks	Tips	Atslēgas	Tukšs	Komentārs
module_id	INT	Primārā(unikāla)	Nē	Unikāls moduļa identifikators.
name	NVARCHAR(50)		Nē	Unikāls moduļa nosaukums.
db_name	NVARCHAR(50)		Jā	Datubāzes nosaukums.

3.1.5.2 Tabula “my_tables”

Šajā tabulā tiek glabāta informācija par tabulām, kuras sistēmā ir izmantotas. Lauks “table_id” tiek ģenerēts automātiski. “name” tiek nolasīts no metadatiem. Tā kā dažādās datubāzēs var atrasties tabulas ar vienādiem nosaukumiem, tad nepieciešams glabāt arī informāciju par datubāzi, kurā atrodas konkrētā tabula. Izmantojot sistēmu datus no šīs tabulas dzēst nav iespējams.

Lauks	Tips	Atslēgas	Tukšs	Komentārs
table_id	INT	Primārā(unikāla)	Nē	Unikāls tabulas identifikators.
name	NVARCHAR(50)		Nē	Tabulas nosaukums.
db_name	NVARCHAR(50)			Datubāzes nosaukums

3.1.5.3 Tabula “my_columns”

Tabulā tiek glabāta informācija par kolonnām, kuras ir izmantotas. Lauks “column_id” tiek ģenerēts automātiski. Kolonnas nosaukumi un datu tipi tiek nolasīti no metadatiem un saglabāti šajā tabulā. Lauka “table_id” vērtība tiek nolasīta no tabulas “my_tables”. Izmantojot sistēmu datus no šīs tabulas dzēst nav iespējams.

Lauks	Tips	Atslēgas	Tukšs	Komentārs
column_id	INT	Primārā(unikāla)	Nē	Unikāls kolonnas identifikators.
name	NVARCHAR(50)		Nē	Kolonnas nosaukums.
table_id	INT	Ārējā atslēga	Nē	Atbilst primārajai atslēgai tabulā “my_tables”.

data_type	NVARCHAR(50)		jā	Norāda datu tipu.
-----------	--------------	--	----	-------------------

3.1.5.4 Tabula “module_table”

Šī tabula nodrošina iespēju atjaunot un uzturēt izveidoto moduļu shēmu. “module_id” tiek nolasīts no tabulas “my_modules”, “table_id” no tabulas “my_table” un “column_id” no tabulas “my_column”. Rediģējot vai dzēšot moduli vērtības var tik dzēstas no šīs tabulas.

Lauks	Tips	Atslēgas	Tukšs	Komentārs
module_id	INT	Ārējā atslēga	Nē	Atbilst primārajai atslēgai tabulā “my_modules”
table_id	INT	Ārējā atslēga	Nē	Atbilst primārajai atslēgai tabulā “my_tables”
column_id	INT	Ārējā atslēga	Nē	Atbilst primārajai atslēgai tabulā “my_column”

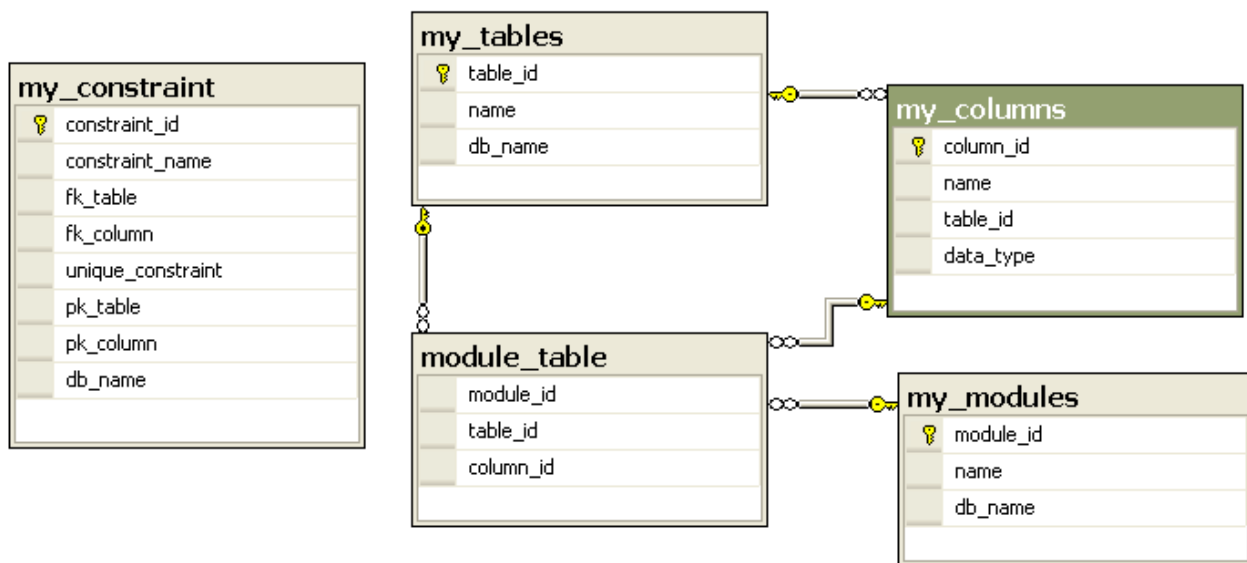
3.1.5.5 Tabula “my_constraint”

Šajā tabulā tiek glabāta informācija par ierobežojumiem, kuri atbilst konkrētai tabulai. Visi lauki, izņemot “constraint_id” tiek nolasīti no metadatiem. Tabula nav sasaistīta ar citām sistēmas tabulām, jo sistēmas darbības nodrošināšanai tas nav nepieciešams. Tā kā programmatūras darbības laikā nav iespējas mainīt kolonnu un tabulu nosaukumus, tad izmantojot tabulu, kolonnu un datubāzes nosaukumus ir iespējams viennpziīmīgi identificēt tabulā esošos objektus.

Lauks	Tips	Atslēgas	Tukšs	Komentārs
constraint_id	INT	Primārā atslēga (unikāla)	Nē	Unikāls atslēgas identifikators
constraint_name	NVARCHAR(50)		Nē	Ārējās atslēgas ierobežojuma nosaukums.
fk_table	NVARCHAR(50)		Jā	Tabulas nosaukums,

				kura norāda ārējo atslēgu.
fk_column	NVARCHAR(50)		Jā	Kolonna, kura ir ārējā atslēga.
unique_constraint	NVARCHAR(50)		Jā	Atbilstošās primārās atslēgas ierobežojuma nosaukums
pk_table	NVARCHAR(50)		Jā	Tabulas nosaukums, kurā atrodas primārās atslēgas kolonna.
pk_column	NVARCHAR(50)		Jā	Kolonnas nosaukums, kura norāda primāro atslēgu.
db_name	NVARCHAR(50)		Jā	Datubāzes nosaukums.

3.1.6 Atkarības aprsksts



4 Kvalitātes nodrošināšana

Kvalitātes nodrošināšanai tika izstrādāta Latvijas standartiem atbilstoša programmatūras prasību specifikācija un programmatūras projektējuma apraksts, kā arī sastādīta testēšanas dokumentācija. Sarežģītākām funkcijām pirms to realizācijas tika veikts projektējums. Programmatūras izstrādes laikā iespēju robežās tika ievērots programmēšanas labais stils. Programmas kods tika komentēts, lai veicot izmaiņas vai atklūdojot programmatūras kods būtu pārskatāms.

Programmatūra tika izstrādāta ar Microsoft Visual Studio 2010 izstrādes rīku. Šī izstrādes vide palīdz optimizēt un nepieļaut sintakses kļūdas izveidotajā kodā kā arī novērst atkārtotu mainīgo lietošanu un citas nepilnības, par tām paziņojot izstrādātājam kļūdu paziņojumos. Koda veidošanas procesā regulāri tika veikta testēšana, testējot katru jaunizveidoto funkciju, lai nodrošinātu pareizu programmas darbību.

Programmprodukta izstrādes beigu fāzē, tika veikta pilnīga programmas testēšana un izveidota testēšanas dokumentācija, kur minētas testētās lietas, to mērķis un sasniegtie rezultāti. Tādējādi tika samazināta kļūdu iespējamība izstrādātajā programmproduktā.

5 Konfigurācijas pārvaldība

Tā kā pie konkrētās projekta daļas strādāja tikai viens programmētājs, tad netika izmantots atsevišķs konfigurāciju pārvaldības rīks. Programmatūras izstrādes procesā manuāli tika veikta rezerves kopiju veidošana uz zibatmiņas un cietā diska, veidojot datnes, kuru nosaukumi atbilst kopijas veidošanas datumiem, lai izvairītos no iespējamās datu pazušanas, kā arī atvieglotu sekošanu līdz darba izpildei. Rezerves kopijas tika veidotas pēc veiksmīgas funkcijas vai lietotnes izveidošanas, lai nepieciešamības gadījumā varētu atgriezties pie iepriekšējās fāzes.

6 Programmatūras projekta organizācija

Sistēmas izstrādes sākumposmā tika izveidota programmatūras prasību specifikācija, kurā bija apkopotas svarīgākās lietas, kas nepieciešamas lai nodrošinātu sistēmas darbību. Projektu centros izstrādāju pēc ūdenskrituma metodes, taču tā kā iepriekš nav bijusi liela pieredze apkopojot programmatūras prasības, tad tika ieviesti daži principi no agile metodes, kas pieļauj prasību precizēšanu izstrādes laikā, līdz ar to nodrošina to, ka programmatūras produktā netiek ieviestas nevajadzīgas vai nepilnīgas funkcijas.

Pie dokumentā aprakstītās sistēmas izveides strādāja viens programmētājs, kas veica sistēmas projektēšanu, datubāzes projektēšanu, programmēšanu un testēšanu.

7 Testēšanas dokumentācija

7.1 Testēšanas procesa apraksts

Šajā dokumentā ir aprakstīti sistēmas vienībtēšanas rezultāti. Testēšana veikta pēc melnās kastes metodes. Testi pārklāj visus ievadlaukus. Tika testēti robežgadījumi, uzskaitīto testu rezultāti ir savstarpēji atšķirīgi. Vienību testēšanas dokumentācija attēlota sekojošajās tabulās.

Testi tika veikti paralēli programmēšanai, līdz ar to visas pamanītās kļūdas tika ātri un operatīvi novērstas, pēctam sekoja atkārtotā testēšana, lai pārbaudītu, vai kautkas nav nejauši palicis nepamanīts.

7.2 Sistēmas testēšana

7.2.1 Moduļa pievienošana

	Situācijas apraksts un ievaddati	Sagaidāmais rezultāts	Statuss
1.	Nav ievadīts moduļa nosaukums un nav atzīmēta neviena tabula	Tiek izdrukāts kļūdas paziņojums „Ievadiet nosaukumu!. Nav atzīmēta neviena tabula”	+
2.	Nav ievadīts moduļa nosaukums, bet ir atzīmēta vismaz viena tabula.	Tiek izdrukāts kļūdas paziņojums „Ievadiet nosaukumu”.	+
3.	Ir ievadīts nosaukums un atzīmēta vismaz viena tabula, bet ievadītais nosaukums ir garāks par 50 simboliem.	Tiek izdrukāts kļūdas paziņojums „Nosaukums nedrīkst būt garāks par 50 simboliem”	+
4.	Ir ievadīts nosaukums, kurš ir garāks par 50 simboliem un nav atzīmēta neviena tabula.	Tiek izdrukāts kļūdas paziņojums „Nav atzīmēta neviena tabula. Nosaukums nedrīkst būt garāks par 50 simboliem.”	+
5.	Tiek atzīmēta tabula	Visas tabulai piesaistītās kolonnas tiek atzīmētas.	+
6.	Tiek noņemta atzīme tabulai	Visām kolonnām tiek noņemta atzīme	+
7.	Tiek noņemta atzīme vienīgajai kolonnai (konkrētā tabulā)	Tabulai atzīme tiek noņemta.	+

8.	Tiek pievienota tabula (kolonna) kura jau ir piesaistīta iepriekš pievienotam modulim (tiek izmantota viena un tā pati datubāze)	Tabula (kolonna) datubāzē no jauna netiek pievienota.	+
9.	Tiek ievadīts moduļa nosaukums, kurš jau eksistē sistēmas datubāzē	Tiek izdrukāts kļūdas paziņojums „Šāds nosaukums jau eksistē”	+

7.2.2 Moduļa rediģēšana

	Situācijas apraksts un ievaddati	Sagaidāmais rezultāts	Statuss
10.	Tiek mainīts moduļa nosaukums. Izmainītais nosaukums sakrīt ar kāda cita moduļa nosaukumu.	Tiek izdrukāts kļūdas paziņojums „Šāds nosaukums jau eksistē”	+
11.	Ir noņemta atzīme visām tabulām.	Tiek izdrukāts paziņojums “Nav atzīmēta neviena tabula”	+
12.	Tiek atzīmēta tabula	Visas tabulai piesaistītās kolonnas tiek atzīmētas.	+
13.	Tiek noņemta atzīme tabulai	Visām kolonnām tiek noņemta atzīme	+
14.	Tiek noņemta atzīme vienīgajai kolonnai (konkrētā tabulā)	Tabulai atzīme tiek noņemta.	+

7.2.3 Vērtību saglabāšana datubāzē

	Situācijas apraksts un ievaddati	Sagaidāmais rezultāts	Statuss
15.	Tiek pievienota kolonnas ar vienādiem nosaukumiem, taču tās pieder dažādām tabulām.	Datubāzē tiek veidots ieraksts katrai unikālajai kolonnai.	+
16.	Tiek pievienotas tabulas ar vienādiem nosaukumiem, taču tās pieder atšķirīgām datubāzēm.	Sistēmas datubāzē tiek veidots atbilstošs ieraksts katrai unikālajai tabulai.	+

17.	Ir atzīmēta kolonna, kura vēl nav saglabāta sistēmas datubāzē.	Ja tabula, kurai pieder konkrētā kolonna jau eksistē datubāzē, tad izveido jaunu ierakstu tabulā „my_tables”. Ja šādas tabulas sistēmas datubāzē nav, tad tiek izveidots jauns ieraksts tabulā „my_tables” un tabulā „my_columns”.	+
18.	Ir noņemta atzīme no kolonnas, kura iepriekš bija pievienota modulim.	No tabulas „module_table” tiek dzēsts ieraksts, kas atbilst konkrētajai kolonnai un konkrētajam modulim.	+
19.	Ir noņemta atzīme no tabulas, kura iepriekš bija pievienota modulim.	No tabulas „module_table” tiek dzēsti visi ieraksti, kuri atbilst konkrētajai tabulai un konkrētajam modulim.	+
20.	Ir atzīmēta tabula, kura vēl nav saglabāta sistēmas datubāzē.	Tabulā my_tables tiek izveidots jauns ieraksts, kas atbilst atzīmētajai tabulai. Tabulā „module_table” tiek pievienots jauns ieraksts, kas atbilst pievienotajai tabulai un tai piesaistītajām kolonnām.	+

7.2.4 Moduļa dzēšana

	Situācijas apraksts un ievaddati	Sagaidāmais rezultāts	Statuss
21.	Ir atzīmēts elements moduļu sarakstlodziņā.	No tabulas „my_modules” tiek dzēsts ieraksts, kas atbilst atzīmētajam modulim. Tiek izdrukāts apstiprinošs paziņojums.	+
22.	Nav atzīmēts elements moduļu sarakstlodziņā.	Tiek izdrukāts kļūdas paziņojums „Nav atzīmēts neviens modulis”	+

7.2.5 Filtra attēlošana

	Situācijas apraksts un ievaddati	Sagaidāmais rezultāts	Statuss
23.	Jāizveido filtrs tabulai ar vienu kolonnu.	Tiek pievienots kolonnas nosaukums un teksta lauks.	+
24.	Jāizveido filtrs 10 (vai mazāk) elementiem.	Filtra elementi tiek izkārtoti vienā kolonnā.	+

25.	Jāizveido ar filtrs vairāk kā 10 elementiem.	Filtra elementi tiek izkārtoti vairākās kolonnās. Katrā kolonnā jābūt tieši 10 elementi, izņemot pēdējo.	+
-----	--	--	---

7.2.6 SQL teikuma ģenerēšana (WHERE daļa)

	Situācijas apraksts un ievaddati	Sagaidāmais rezultāts	Statuss
26.	Nav ievadīta vērtība nevienā no teksta laukiem.	Tiek ģenerēts SQL pieprasījums bez WHERE daļas	+
27.	Ir ievadīta vērtība vienā teksta laukā un tā nesatur iepriekš definētos operatorus.	SQL teikumam tiek pievienota WHERE daļa, kurā norādīta ievadītā vērtība atbilstoši SQL sintaksei, izmantojot operetoru “=”.	+
28.	Ir ievadīta vērtība vairākos teksta laukos, bet neviena nesatur iepriekš definētos operatorus.	SQL teikumam tiek pievienota WHERE daļa un visi ievadītie operanti tiek sasaistīti ar operatoru “and”, atbilstoši SQL sintaksei.	+
29.	Ir ievadīta vērtība vienā teksta laukā un tā satur vairākus operandus, starp kuriem ir operators „or”	Tiek ģenerēts SQL teikums, kuram WHERE nosacījumā norādīta ievadītā izteiksme atbilstoši SQL sintaksei.	+
30.	Ir ievadītas vērtības vairākos teksta laukos un vismaz viena no tām satur operatoru „or” (vai vairākus) un tie ir izkārtoti loģiskā secībā , bet nesatur kādu citu no operatoriem.	SQL teikuma WHERE nosacījums tiek ģenerēts formā SELECT <kolonnas nosaukums1, kolonnas nosaukums2...> FROM <tabulas nosaukums> WHERE <kolonnas1 nosaukums> = <operands1> OR <kolonnas1 nosaukums> = <operands2> AND <kolonnas2 nosaukums> = <operands3> OR <kolonnas2 nosaukums> = <operands4>	+
31.	Kādān no teksta laukiem ievadītā vērtība satur vairākus ooperatorus “or” pēc kārtas.	Blakus esošie, vienādie operatori tiek aizstāti ar vienu un tiek veidots SQL teikums atbilstoši SQL sintaksei.	+

32.	Kādā no teksta laukiem ievadītā vērtība sākas ar simbolu virkni “or_”.	Tiek izdrukāts kļūdas paziņojums “Kļūda!”	+
33.	Kādā no teksta laukiem ievadītā vērtība beidzas ar simbolu virkni “_or”	Tiek izdrukāts kļūdas paziņojums “Kļūda!”	+
34.	Kādā no teksta laukiem ievadītā vērtība formā “not aaa or bbb”.	Tiek ģenerēts WHERE nosacījums atbilstoši ievadītajai vērtībai un SQL sintaksei, izmantojot operatoru “=”.	+
35.	Kādā no teksta laukiem ievadītā vērtība formā “aaa or not b”.	Tiek ģenerēts WHERE nosacījums atbilstoši ievadītajai vērtībai un SQL sintaksei, izmantojot operatoru “=”.	+
36.	Kādā no teksta laukiem ievadītā vērtība formā “not aaa or not bbb”	Tiek ģenerēts WHERE nosacījums atbilstoši ievadītajai vērtībai un SQL sintaksei, izmantojot operatoru “=”.	+
37.	Kādā no teksta laukiem ievadītā vērtība formā “not z% or not %g”	Tiek ģenerēts WHERE nosacījums atbilstoši ievadītajai vērtībai un SQL sintaksei, izmantojot operatoru “LIKE”.	+
38.	Kādā no teksta laukiem ievadītā vērtība formā “not z% or %g”	Tiek ģenerēts WHERE nosacījums atbilstoši ievadītajai vērtībai un SQL sintaksei, izmantojot operatoru “LIKE”.	+
39.	Kādā no teksta laukiem ievadītā vērtība formā “z% or a%”	Tiek ģenerēts WHERE nosacījums atbilstoši ievadītajai vērtībai un SQL sintaksei, izmantojot operatoru “LIKE”.	+
40.	Kādā no teksta laukiem ievadītā vērtība formā “not aaa and not bbb”	Tiek ģenerēts WHERE nosacījums atbilstoši ievadītajai vērtībai un SQL sintaksei, izmantojot operatoru “=”.	+
41.	Kādā no teksta laukiem ievadītā vērtība formā “not aaa and bbb”	Tiek ģenerēts WHERE nosacījums atbilstoši ievadītajai vērtībai un SQL sintaksei, izmantojot operatoru “LIKE”.	+
42.	Kādā no teksta laukiem ievadītā vērtība formā “not %aaaa”	Tiek ģenerēts WHERE nosacījums atbilstoši ievadītajai vērtībai un SQL sintaksei, izmantojot operatoru “LIKE”.	+

43.	Kādā no teksta laukiem ievadīta vērtība formā “not z% and %g”	Tiek ģenerēts WHERE nosacījums atbilstoši ievadītajai vērtībai un SQL sintaksei, izmantojot operatoru “LIKE”.	+
44.	Kādā no teksta laukiem ievadīta vērtība formā “aaa and not b”	Tiek ģenerēts WHERE nosacījums atbilstoši ievadītajai vērtībai un SQL sintaksei, izmantojot operatoru “=”.	+
45.	Kādā no teksta laukiem ievadīta vērtība formā “a% and %g”	Tiek ģenerēts WHERE nosacījums atbilstoši ievadītajai vērtībai un SQL sintaksei, izmantojot operatoru “LIKE”.	+
46.	Kādā no teksta laukiem ievadīta vērtība formā “not %aaaa”	Tiek ģenerēts WHERE nosacījums atbilstoši ievadītajai vērtībai un SQL sintaksei, izmantojot operatoru “LIKE”.	+
47.	Kādā no teksta laukiem ievadīta vērtība formā “not z% or %g”	Tiek ģenerēts WHERE nosacījums atbilstoši ievadītajai vērtībai un SQL sintaksei, izmantojot operatoru “LIKE”.	+
48.	Ir ievadīta vērtība vismaz vienā teksta laukā un tā satur operatoru „and” bet nesatur citus operatorus, piemēram, “aaa and bbb”	Dotajam piemēram atbilstošs WHERE nosacījums būtu šāds: “where name = ‘aaa’ and name = ‘bbb’, taču tas nekad neatriezīs nevienu ierakstu no datubāzes.	+
49.	Ir ievadītas korektas vērtības (tādas, kuras neatgriež kļūdas paziņojumu) vairākos teksta laukos un tās satur dažādus operatorus.	Tiek ģenerēts WHERE nosacījums atbilstoši SQL sintaksei un visas ievadītās vērtības savā starpā tiek sasaistītas ar operatoru “and”.	+
50	Ir ievadīti dažādi operatori, kas nevar likt lietoti vienlaicīgi vienam laukam, piemēram “between” un “<”.	Tiek izvadīts SQL server atgrieztais kļūdas paziņojums.	+

7.2.7 SQL teikuma ģenerēšana (SELECT un FROM daļa)

	Situācijas apraksts un ievaddati	Sagaidāmais rezultāts	Statuss
51.	Nepieciešams atlasīt vienu kolonnu.	Tiek ģenerēts SQL teikums formā “SELECT <kolonnas_nosaukums> FROM <tabulas_nosaukums>”	+

52.	Nepieciešams atlasīt vairākas kolonnas no vienas tabulas.	Tiek ģenerēts SQL teikums formā “SELECT <kolonnas1_nosaukums>, <kolonnas2_nosaukums> ... FROM <tabulas_nosaukums>”	+
53.	Nepieciešams atlasīt datus no vairākām tabulām.	Tiek ģenerēts SQL teikums formā “SELECT <tabulas1_nosaukums>.<kolonnas1_nosaukums>, <tabulas1_nosaukums>.<kolonnas2_nosaukums> ... FROM <tabulas1_nosaukums> JOIN <tabulas2_nosaukums> ON <tabulas1_nosaukums>.<tabulas1_primārā_atslēga> = <tabulas2_ārējā_atslēga>”	+

7.2.8 Informācijas attēlošana kontroļos

	Situācijas apraksts un ievaddati	Sagaidāmais rezultāts	Statuss
54.	Programma tiek atvērta.	Sadaļā “Dati” koka skata kontrolī tiek attēloti izveidotie moduļi ar tiem piesaistītajām tabulām un kolonnām. Sadaļā “Moduļi” moduļu sarakstlodziņā tiek attēloti moduļu nosaukumi.	+
55.	Sadaļā “Moduļi” tiek nospiesta poga “Jauns”.	Kombinētais lodziņš “Datubāzes” tiek attēlots kā redzams. Tajā tiek ielasīti visu pieejamo datubāzu nosaukumi.	+
56.	Sadaļā “Moduļi” tiek nospiesta poga “Saglabāt”, lai pievienotu jaunu moduli.	Sadaļā “Moduļi” moduļu sarakstlodziņā tiek pievienots izveidotais modulis.	+
		Sadaļā “Dati” moduļu koka skatā tiek pievienots jauns modulis.	+
57.	Sadaļā moduļi, no moduļu sarakstlodziņa tiek izvēlēts kāds	Kombinētais lodziņš “Datubāzes” tiek attēlots kā neredzams.	+

	modulis.	Precīzi tiek attēlota pievienotā moduļa shēma (tiek attēlotas visas attiecīgajā datubāzē esošās tabulas un kolonnas. Tās, kuras ir pievienotas modulim, tiek atzīmētas)	+
58.	Sadaļā “Moduļi” tiek nospiesta poga “Saglabāt”, lai saglabātu rediģētu moduli..	Ja moduļa nosaukums ir izmainīts, tad gan sadaļā “Moduļi”, gan sadaļā “Dati” tiek attēlots jaunais nosaukums.	+
59.	Sadaļā “Moduļi” tiek nospiesta poga “Dzēst”.	Sadaļā “Moduļi” no moduļu sarakstlodziņā tiek noņemts izdzēstais modulis.	+
		Sadaļā “Dati” no moduļu koka skatā tiek noņemts izdzēstais modulis.	+

7.2.9 Datu atlase

Moduļa testēšana tika veikta salīdzinot sistēmas uzģenerētā SQL pieprasījuma atgrieztos datus un SQL Server Management Studio atgrieztos datus pie vienādiem nosacījumiem.

	Situācijas apraksts un ievaddati	Sagaidāmais rezultāts	Statuss
60.	Jāatlasa visi dati no vienas tabulas (atbilstoši tam, kādas kolonnas ir norādītas, pievienojot moduli)	SQL Server Management Studio un sistēmas ģenerētā pieprasījuma atgrieztie dati ir identiski.	+
61.	Jāatlasa visi dati no vienas kolonnas.	SQL Server Management Studio un sistēmas ģenerētā pieprasījuma atgrieztie dati ir identiski.	+
62.	Jāatlasa visi dati no divām tabulām.	SQL Server Management Studio un sistēmas ģenerētā pieprasījuma atgrieztie dati ir identiski.	+
63.	Jāatlasa visi dati no vairāk kā divām tabulām.	SQL Server Management Studio un sistēmas ģenerētā pieprasījuma atgrieztie dati ir identiski.	+

64.	Jāatlasa dati no vienas tabulas, ievērojot “where” nosacījumu.	SQL Server Management Studio un sistēmas ģenerētā pieprasījuma atgrieztie dati ir identiski.	+
65.	Jāatlasa dati no vairākām tabulām ievērojot “where” nosacījumu.	SQL Server Management Studio un sistēmas ģenerētā pieprasījuma atgrieztie dati ir identiski.	+

7.2.10 Ierobežojumu loga attēlošana

	Situācijas apraksts un ievaddati	Sagaidāmais rezultāts	Statuss
66.	Jāizveido ierobežojumu logs tabulai, kurā nav nevienas ārējās atslēgas.	Tiek atvērts ierobežojumu logs, kurā nav neviena cita kontrole, izņemot pogu “Apstiprināt”	+
67.	Jāizveido ierobežojumu logs tabulai, kurā ir viena ārējā atslēga.	Tiek atvērts ierobežojumu logs, kurā ir viena kolonna ar elementiem. Kolonnas pirmais elements apzīmē kolonnu, kura norāda ārējo atslēgu. Pārējie elementi atbilst tabulas kolonnām, kurā atrodas primārā atslēga.	+
68.	Jāizveido ierobežojumu logs tabulai, kurā ir vairākas ārējās atslēgas.	Tiek atvērts ierobežojumu logs, kurā ir vairākas kolonnas ar elementiem.	+
69.	Jāizveido ierobežojumu logs tabulai, kurai kāda no ārējām atslēgām jau ir aizstāta ar kādu no primārās atslēgas tabulas kolonnām.	Tiek atvērts ierobežojumu logs ar tikai tām kolonnām, kuras vēl nav pievienotas.	+
70.	Jāizveido ierobežojumu logs tabulai, kurai visas ārējās atslēgas ir aizstātas ar visām kolonnām no primāro atslēgu tabulām.	Tiek atvērts ierobežojumu logs, kurā nav neviena cita kontrole, izņemot pogu “Apstiprināt”	+

8 Darbietilpības novērtējums

Programmatūras atribūti:

Nepieciešamā sistēmas atkarība: programmatūras strādāt nespējas gadījumā atkopšanās no sistēmas datu zuduma lietotāju apdraud minimāli. Sistēmas darbību atjaunot ir salīdzinoši vienkārši.

Datubāzes izmērs: mazs (5 tabulas)

Produkta sarežģītība: Vidēja

Produkta uzticamība: Virs vidējā

Izstrādes servera atribūti

Izpildes laiks: vidējs

Aizņemtā vieta uz datu nesēja: zema

Pieprasījuma apstrādes laiks: Pieprasījuma apstrāde ir atkarīga no SQL servera darbības ātruma un ir atkarīga no apstrādājamo datu apjoma.

Personāla atribūti

Pieredze aplikāciju izstrādē: zema

Izstrādes vides izmantošanas pieredze: zema (< 1 gads)

Pieredze izmantotajā programmēšanas valodā: zema (~ 4 mēneši)

Projekta atribūti

Izmantotas mūsdienīgas programmēšanas metodes: vidēji

Izmantotie programmatūras rīki: Visual Studio 2010, SQL Server Managemet Studio

Nepieciešamais izstrādes laiks: nav apstākļu, kas liktu paātrināt izstrādi

Lai aprēķinātu darbietilpību, izmantoju tīmeklī pieejamu COCOMO darbietilpības kalkulatoru.

(http://sunset.usc.edu/research/COCOMOII/cocomo81_pgm/cocomo81.html)

Darbietilpība: **4,53 personmēneši**

Reālais izstrādes laiks: **3 mēneši**

Koda rindiņas: **1502**

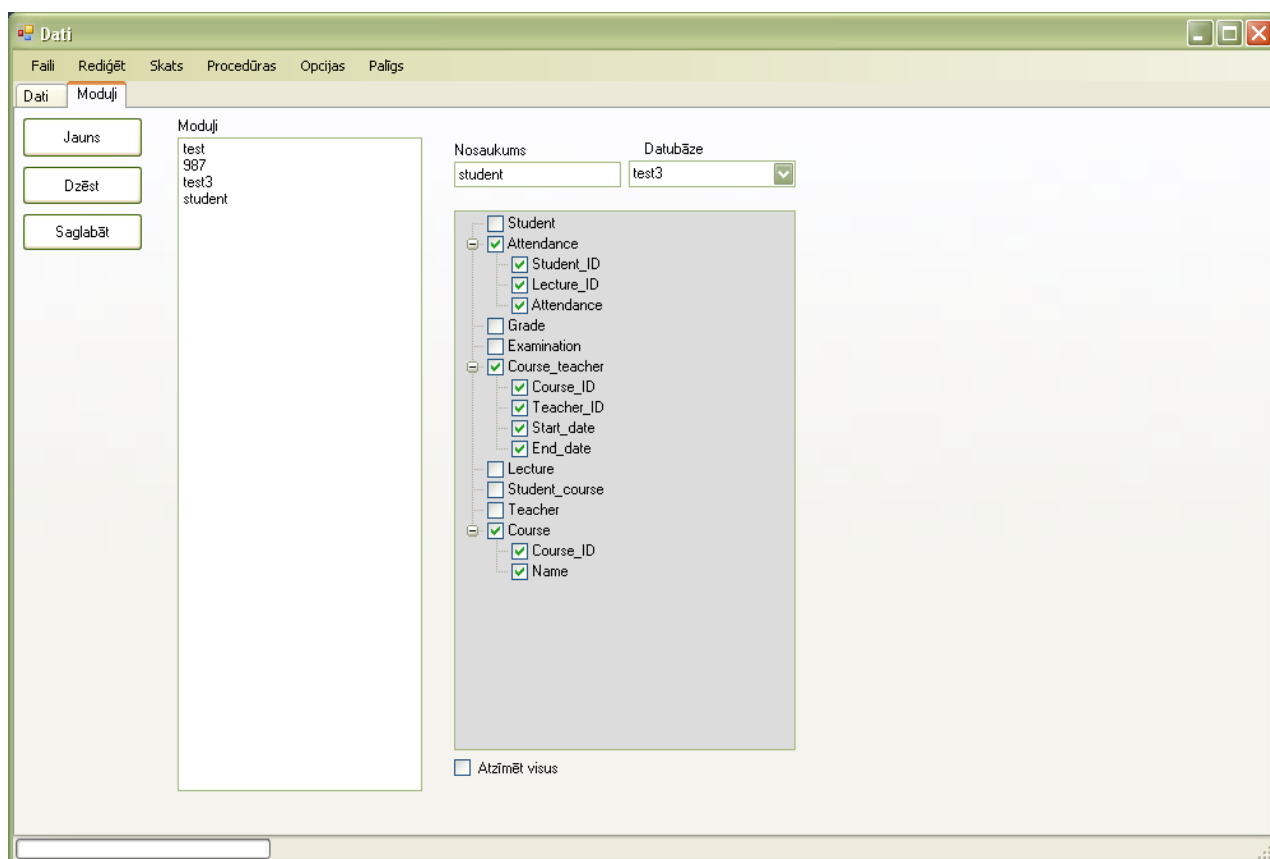
9 Trasējāmības tabula

NR .P. K	Funkcionālā prasība		Modulis		TESTPIEMĒRA NR.
	Identifikators	Nosaukums	Identifikators	Nosaukums	
1.	F.1.3.1.	Moduļa pievienošana	M.2.1.4.1.	“MY_DATA”	1., 2., 3., 4., 5., 6., 7., 8., 9.
2.	F.1.3.2.	Moduļa rediģēšana	M.2.1.4.1.	“MY_DATA”	10., 11., 12., 13., 14.
3.	F.1.3.3.	Vērtību saglabāšana datubāzē	M.2.1.4.1.	“MY_DATA”	15., 16., 17., 18., 19., 20.
4.	F.1.3.4.	Moduļa dzēšana	M.2.1.4.1.	“MY_DATA”	21., 22.
5.	F.1.3.5.	Filtra attēlošana	M.2.1.4.4.	“INTERFACE”	23., 24., 25.
6.	F.1.3.6.	Informācijas attēlošana kontroljos	M.2.1.4.2.	“DATA”	54., 55., 56., 57., 58., 59.
7.	F.1.3.7.	Datu atlase	M.2.1.4.3.	“SQL_QUERY”	60., 61., 62., 63., 64., 65.
8.	F.1.3.8.	SQL teikuma ģenerēšana (SELECT, FROM daļas ģenerēšana)	M.2.1.4.3.	“SQL_QUERY”	51., 52., 53.
9.	F.1.3.9.	SQL teikuma ģenerēšana(WHERE daļas ģenerēšana)	M.2.1.4.3.	“SQL_QUERY”	26., 27., 28., 29., 30., 31., 32., 33., 34., 35., 36., 37., 38., 39., 40., 41., 42., 43., 44., 45., 46., 47., 48., 49., 50.
10.	F.1.3.10.	Ierobežojumu loga attēlošana	M.2.1.4.4.	“INTERFACE”	66., 67., 68., 69., 70.

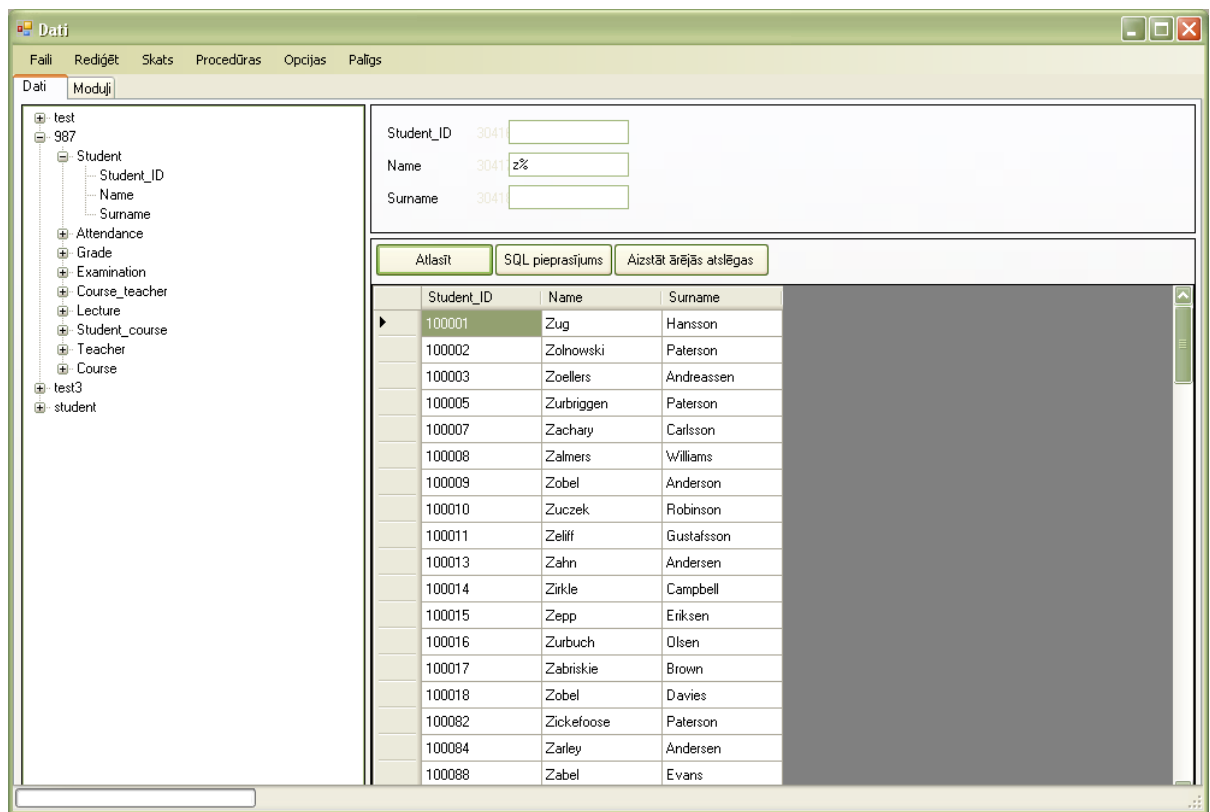
10 Pielikums

10.1 Ekrānformas

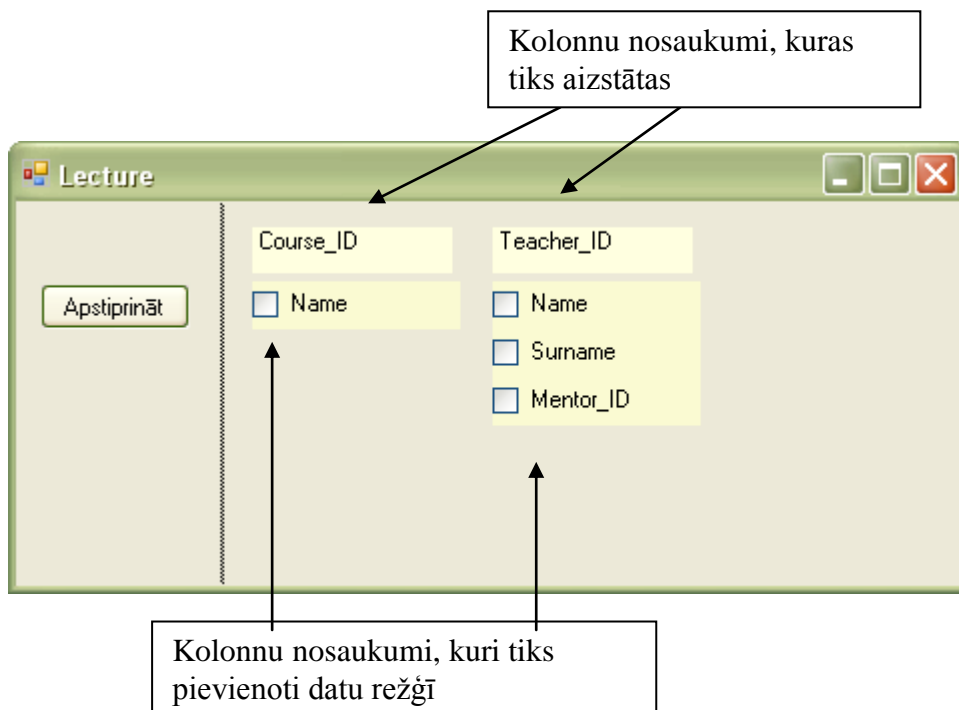
10.1.1 Sadaļa "Moduļi"



10.1.2 Sadaļa "Dati"



10.1.3 Ierobežojumu logs



10.2 Programmas kods

```
Imports System.Data.SqlClient

Module DB_DATA
    'saglabā datus tabulā 'my_constraint'
    Public Sub table_constraints(ByRef column_name As String)
        Try
            'cn1.Open()
            Dim cm19 As New SqlCommand
            cm19.CommandType = CommandType.Text
            cm19.Connection = Form2.cn
            cm19.CommandText = "select a.CONSTRAINT_NAME, a.TABLE_NAME,
a.TABLE_CATALOG, b.UNIQUE_CONSTRAINT_name, c.COLUMN_NAME, c.TABLE_NAME from " _
                & Form2.database_combobox.Text &
                ".INFORMATION_SCHEMA.TABLE_CONSTRAINTS a, " _
                & Form2.database_combobox.Text &
                ".INFORMATION_SCHEMA.REFERENTIAL_CONSTRAINTS b, " _
                & Form2.database_combobox.Text &
                ".INFORMATION_SCHEMA.KEY_COLUMN_USAGE c " _
                & "where a.table_name = '" & column_name & "' and
a.CONSTRAINT_TYPE = 'foreign key' and c.CONSTRAINT_NAME = a.CONSTRAINT_NAME and
c.CONSTRAINT_NAME = b.CONSTRAINT_NAME "
            Dim constraint As SqlDataReader = cm19.ExecuteReader
            While constraint.Read
                Dim cm111 As New SqlCommand
                cm111.CommandType = CommandType.Text
                cm111.Connection = Form2.cn2
                cm111.CommandText = "select table_name, column_name from " &
Form2.database_combobox.Text & ".INFORMATION_SCHEMA.KEY_COLUMN_USAGE " _
                    & " where constraint_name = '" & constraint.GetValue(3) & "'"
                Dim prim_constraint As SqlDataReader = cm111.ExecuteReader
                prim_constraint.Read()
                Dim cm18 As New SqlCommand
                cm18.CommandType = CommandType.Text
                cm18.Connection = Form2.cn1
                cm18.CommandText = "insert into my_constraint values('" &
RTrim(LTrim(constraint.GetValue(0))) & "', '" &
RTrim(LTrim(constraint.GetValue(1))) & "', '" & RTrim(LTrim(constraint.GetValue(4)))
& "', '" & RTrim(constraint.GetValue(3)) & "', '" &
RTrim(LTrim(prim_constraint.GetValue(0))) & "', '" &
RTrim(LTrim(prim_constraint.GetValue(1))) & "', '" &
RTrim(LTrim(constraint.GetValue(2))) & "'" )"
                cm18.ExecuteNonQuery()
                prim_constraint.Close()
            End While
            constraint.Close()
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        Finally
            'cn1.Close()
        End Try
    End Sub

    'pievienojot jaunu moduli saglabā vērtības sistēmas datubāzē
    Public Sub save_new_module()
        'saglabā datubāzē tabulu, kolonnu nosaukumus
        Dim Tree As TreeView = Form2.TreeView1
        If Form2.TreeView1.Visible = False Then
            Tree = Form2.TreeView2
        End If
        Dim table_text As String
        Dim table_read As String
        Dim title As String
        Dim cm1 As New SqlCommand
    
```

```

cm1.CommandType = CommandType.Text
cm1.Connection = Form2.cn
Dim cm3 As New SqlCommand
cm3.CommandType = CommandType.Text
cm3.Connection = Form2.cn3
cm3.CommandText = "select * from my_modules"
Dim cm4 As New SqlCommand
cm4.CommandType = CommandType.Text
cm4.Connection = Form2.cn3
Dim cm5 As New SqlCommand
cm5.CommandType = CommandType.Text
cm5.Connection = Form2.cn3
Dim cm6 As New SqlCommand
cm6.CommandType = CommandType.Text
cm6.Connection = Form2.cn4
Dim cm7 As New SqlCommand("select * from my_columns", Form2.cn2)
cm7.CommandType = CommandType.Text
cm7.Connection = Form2.cn2
Dim cm8 As New SqlCommand
cm8.CommandType = CommandType.Text
cm8.Connection = Form2.cn4
Form2.error_Label.Text = ""
'ja modulīs tiek rediģets, tad nolasa modulja_id
Dim module_id2 As String
Dim db_name As String

If Form2.TreeView2.Visible = True Then
    Dim cm20 As New SqlCommand
    cm20.CommandType = CommandType.Text
    cm20.Connection = Form2.cn
    cm20.CommandText = "select * from my_modules where name = '" &
Form2.ListBox1.SelectedItem & "'"
    Try
        Form2.cn.Open()
        Dim module_id_db As SqlDataReader = cm20.ExecuteReader
        While module_id_db.Read
            module_id2 = module_id_db.GetValue(0)
            db_name = module_id_db.GetValue(2)
        End While
        module_id_db.Close()
    Catch ex As Exception
        MessageBox.Show(ex.Message)
    Finally
        Form2.cn.Close()
    End Try
End If
Dim sk4 As Integer = 0
Dim parent As TreeNode

For Each parent In Tree.Nodes
    If parent.Checked Then
        sk4 = sk4 + 1
    End If
    'Dim child As TreeNode
    'child = parent.Nodes
    For Each TreeNode In parent.Nodes
        If TreeNode.Checked Then
            sk4 = sk4 + 1
        End If
    Next
Next
If Form2.TextBox1.Text = "" And sk4 = 0 Then
    Form2.error_Label.Text = "Ievadiet nosaukumu!" & vbNewLine & "Nav
atzimēta neviena tabula!"
Else
    If sk4 = 0 And Form2.TextBox1.Text.Length > 50 Then

```

```

        Form2.error_Label.Text = "Nav atzīmēta neviena tabula! " &
vbNewLine & "Nosaukums nedrīkst būt garāks par 50 simboliem! "
    Else
        If sk4 = 0 Then
            Form2.error_Label.Text = "Nav atzīmēta neviena tabula! " &
vbNewLine & ""
        Else
            If Form2.TextBox1.Text = "" Then
                Form2.error_Label.Text = "Ievadiet nosaukumu! " & vbNewLine
& ""
            Else
                If Form2.TextBox1.Text.Length > 50 Then
                    Form2.error_Label.Text = "Nosaukums nedrīkst būt garāks
par 50 simboliem! " & vbNewLine & ""
                Else
                    Try
                        Form2.cn.Open()
                        Form2.cn3.Open()
                        Form2.cn4.Open()
                        Form2.cn5.Open()
                        Form2.cn1.Open()
                        title = Form2.TextBox1.Text
                        Dim text As SqlDataReader = cm3.ExecuteReader
                        While text.Read
                            If Form2.TreeView1.Visible = True Then
                                If RTrim(text.GetValue(1)).ToLower = title
Then
                                    Form2.error_Label.Text = "Šāds
nosaukums jau eksistē! " & vbNewLine & ""
                                    Form2.cn5.Close()
                                    Form2.cn2.Close()
                                    Form2.cn4.Close()
                                    Form2.cn3.Close()
                                    Form2.cn.Close()
                                    GoTo line3 'ja nosaukums jau eksistē,
tad iet uz beigām
                                End If
                            ElseIf Form2.TreeView2.Visible = True Then
                                If Not text.GetValue(0) = module_id2 Then
                                    If RTrim(text.GetValue(1)).ToLower =
title Then
                                        'MessageBox.Show(text.GetValue(0) &
" " & module_id2 & vbNewLine & RTrim(text.GetValue(1)).ToLower & " " & title)
                                        Form2.error_Label.Text = "Šāds
nosaukums jau eksistē!! " & vbNewLine & ""
                                        Form2.cn5.Close()
                                        Form2.cn2.Close()
                                        Form2.cn4.Close()
                                        Form2.cn3.Close()
                                        Form2.cn.Close()
                                        GoTo line3 'ja nosaukums jau
eksistē, tad iet uz beigām
                                    End If
                                End If
                            End If
                        End While
                        text.Close()
                        If Form2.TreeView1.Visible = True Then
                            cm4.CommandText = "insert into my_modules
values ('" & title & "', '" & Form2.database_combobox.Text & "')"
                        Else
                            cm4.CommandText = "update my_modules set name =
'" & Form2.TextBox1.Text & "' where module_id = '" & module_id2 & "'"
                            'MessageBox.Show("nosaukums ir izmainiits")
                        End If
                        cm4.ExecuteNonQuery()

```

```

'saglabā kolonnas un tabulas
If Form2.TreeView1.Visible = True Then
    Dim obj As TreeNode
    For Each obj In Form2.TreeView1.Nodes
        If obj.Checked = True Then
            table_text = CStr(obj.Text)
            Dim cm2 As New SqlCommand("select *
from my_tables", Form2.cn3)

            cm2.CommandType = CommandType.Text
            cm2.Connection = Form2.cn3
            Dim table As SqlDataReader =
cm2.ExecuteReader

            While table.Read
                table_read =
RTrim(table.GetValue(1))
                If (RTrim(table_text).ToLower =
RTrim(table_read).ToLower And Form2.database_combobox.Text =
RTrim(table.GetValue(2))) Then
                    GoTo Line2
                End If
            End While
            cm1.CommandText = "insert into
my_tables values (' & table_text & ', ' & Form2.database_combobox.Text & ')"
            cm1.ExecuteNonQuery()
            Form2.cn2.Open()
            table_constraints(table_text)
            Form2.cn2.Close()

line2:
            table.Close()
            'cn2.Close()
        End If
    Next
End If
If Form2.TreeView1.Visible = True Then
    For Each obj In Form2.TreeView1.Nodes
        Form2.cn2.Open()
        If obj.Checked Then
            table_text = CStr(obj.Text)
            Dim cm2 As New SqlCommand
            cm2.CommandType = CommandType.Text
            cm2.Connection = Form2.cn3
            If Form2.TreeView1.Visible = False Then
                cm2.CommandText = "select * from
my_tables where name = ' & RTrim(table_text) & ' and db_name = ' & db_name & '"
            Else
                cm2.CommandText = "select * from
my_tables where name = ' & RTrim(table_text) & ' and db_name = ' &
Form2.database_combobox.Text & '"

            End If
            Dim table As SqlDataReader =
cm2.ExecuteReader()

            table.Read()
            Dim table_id As String =
table.GetValue(0)

            table.Close()
            Dim child As TreeNode
            For Each child In obj.Nodes
                If child.Checked = True Then
                    Dim column As SqlDataReader =
cm7.ExecuteReader

                    While column.Read
                        If
RTrim(column.GetValue(1)).ToLower = RTrim(child.Text).ToLower And
column.GetValue(2) = table_id Then
                            column.Close()
                            GoTo line4

```

```

        End If
        'table_id_2 =

table.GetValue(0)

CommandType.Text

        cm08.Connection = Form2.cn
        cm08.CommandText = "select *
from " & Form2.database_combobox.Text & ".INFORMATION_SCHEMA.columns where
column_name = '" & child.Text & "' and table_name = '" & child.Parent.Text & "'"
        Dim column_type As

SqlDataReader = cm08.ExecuteReader

        column_type.Read()
        cm8.CommandText = "insert into
my_columns values ('" & child.Text & "', " & table_id & ", '" &
column_type.GetValue(7) & "')"

        cm8.ExecuteNonQuery()
        column_type.Close()
        column.Close()

line4:
        End If
    Next

    End If
    Form2.cn2.Close()
Next
End If
'saglabā vērtības starptabulā
'MessageBox.Show(title)
If Form2.TreeView1.Visible = True Then
    cm5.CommandText = "select * from my_modules
where name = '" & title & "'"
    Dim title_read As SqlDataReader =
cm5.ExecuteReader

    title_read.Read()
    Dim title_text_int As Integer =
title_read.GetValue(0) 'moduļa nosaukuma ID
    Dim title_text_str As String =
title_read.GetValue(1) 'moduļa nosaukums
    title_read.Close()
    'MessageBox.Show(title_text_int)
    For Each obj In Form2.TreeView1.Nodes
        If obj.Checked Then
            table_text = CStr(obj.Text).ToLower
            Dim cm2 As New SqlCommand
            If Form2.TreeView1.Visible = True Then
                cm2.CommandText = "select * from
my_tables where db_name = '" & Form2.database_combobox.Text & "'"
            Else
                cm2.CommandText = "select * from
my_tables where db_name = '" & Form2.database_combobox.Text & "'"
            End If
            cm2.CommandType = CommandType.Text
            cm2.Connection = Form2.cn3

            Dim table2 As SqlDataReader =

cm2.ExecuteReader

            While table2.Read
                Dim cm9 As New SqlCommand("select *
from my_columns where table_id = '" & table2.GetValue(0) & "'", Form2.cn5)
                cm9.CommandType = CommandType.Text
                Dim column As SqlDataReader =

cm9.ExecuteReader

                Dim value As String =

                While column.Read

```

```

Dim child As TreeNode
For Each child In obj.Nodes
    If child.Checked Then
        If
RTrim(table_text.ToLower) = RTrim(table2.GetValue(1).ToLower) And
RTrim(table2.GetValue(2)) = Form2.database_combobox.Text Then
            If
RTrim(child.Text.ToLower) = RTrim(column.GetValue(1)).ToLower And
table2.GetValue(0) = column.GetValue(2) Then

'MessageBox.Show("no formas - " & child.Text & " no BD - " &
RTrim(column.GetValue(1)).ToLower & vbNewLine & " no my_table - " &
table2.GetValue(0) & "no my_column - " & column.GetValue(2))
cm6.CommandText
= " insert into module_table values ('" & title_text_int & "' , '" & value & "','"
& column.GetValue(0) & "')"

cm6.ExecuteScalar()

            End If
        End If
    End If
Next
End While
column.Close()
line1: End While
table2.Close()
End If
Next
MessageBox.Show("Modulis ir pievienots")
End If
Catch ex As Exception
    MessageBox.Show(ex.Message)
Finally
    Form2.cn1.Close()
    Form2.cn5.Close()
    Form2.cn2.Close()
    Form2.cn4.Close()
    Form2.cn3.Close()
    Form2.cn.Close()
End Try
If Form2.TreeView2.Visible = True Then
    redig_module()
    MessageBox.Show("Modulis ir saglabāts")
    Tree.Nodes.Clear()
    Form2.TextBox1.Text = ""
End If
End If
End If
End If
line3: End If
Form2.CheckBox2.Checked = False
listbox_refresh()
treeview3_data()
End Sub

'redigējot moduli saglabā vērtības datubāzē. ja nepieciešams, tad dzēš vērtības
Private Sub redig_module()
    ' noskaidro datubaazi un modulja id
    Dim cm1 As New SqlCommand("select * from my_modules where name = '" &
Form2.TextBox1.Text & "'", Form2.cn)
    cm1.CommandType = CommandType.Text
    cm1.Connection = Form2.cn
    Dim module_id As String
    Dim db_name As String
    Try

```

```

Form2.cn.Open()
Dim mod_id As SqlDataReader = cm1.ExecuteReader
mod_id.Read()
module_id = mod_id.GetValue(0)
db_name = mod_id.GetValue(2)
mod_id.Close()
Catch ex As Exception
    MessageBox.Show(ex.Message)
Finally
    Form2.cn.Close()
End Try
Try
    Dim parent1 As TreeNode
    Form2.cn.Open()
    Form2.cn2.Open()
    Form2.cn1.Open()
    Form2.cn3.Open()
    Form2.cn4.Open()
    Form2.cn5.Open()
    For Each parent1 In Form2.TreeView2.Nodes
        Dim cm5 As New SqlCommand
        cm5.Connection = Form2.cn3
        cm5.CommandType = CommandType.Text
        cm5.CommandText = "select * from my_tables where db_name = '" &
db_name & "'"
        Dim my_tables As SqlDataReader = cm5.ExecuteReader
        If parent1.Checked = True Then
            'paarbauda vai atziimeetaas tabulas un kolonnas jau ir
            datubaazee
            While my_tables.Read
                'ja tabula jau ir saglabaata my_tables
                'MessageBox.Show(RTrim(my_tables.GetValue(1)).ToLower & "
" & RTrim(parent1.Text).ToLower)
                'MessageBox.Show(RTrim(my_tables.GetValue(2)).ToLower & "
" & db_name)
                If RTrim(my_tables.GetValue(1)).ToLower =
RTrim(parent1.Text).ToLower And RTrim(my_tables.GetValue(2)).ToLower =
RTrim(db_name).ToLower Then
                    'MessageBox.Show("tabula jau ir " & parent.Text)
                    Dim child1 As TreeNode
                    For Each child1 In parent1.Nodes
                        If child1.Checked = True Then
                            Dim cm2 As New SqlCommand
                            cm2.CommandType = CommandType.Text
                            cm2.Connection = Form2.cn1
                            cm2.CommandText = "select * from my_columns
where table_id = '" & my_tables.GetValue(0) & "'"
                            Dim my_columns As SqlDataReader =
cm2.ExecuteReader
                            While my_columns.Read
                                If RTrim(my_columns.GetValue(1)).ToLower =
RTrim(child1.Text).ToLower Then
                                    'MessageBox.Show(child.Text & "
shitaads child jau ir")
                                    my_columns.Close()
                                    GoTo line3
                                End If
                            End While
                            Dim cm08 As New SqlCommand
                            cm08.CommandType = CommandType.Text
                            cm08.Connection = Form2.cn
                            cm08.CommandText = "select * from " & db_name &
".INFORMATION_SCHEMA.columns where column_name = '" & child1.Text & "' and
table_name = '" & child1.Parent.Text & "'"
                            Dim column_type As SqlDataReader =
cm08.ExecuteReader

```

```

        column_type.Read()
        Dim cm8 As New SqlCommand
        cm8.CommandType = CommandType.Text
        cm8.Connection = Form2.cn
        cm8.CommandText = "insert into my_columns
values('" & child1.Text & "', '" & RTrim(my_tables.GetValue(0)) & "', '" &
column_type.GetValue(7) & "')"
        cm8.ExecuteNonQuery()
        'MessageBox.Show("pievienoju tabulai my_columns
" & child1.Text & " (" & RTrim(my_tables.GetValue(1)) & ")")
        column_type.Close()
        my_columns.Close()
    End If
line3:    Next
        my_tables.Close()
        GoTo line1
    End If
End While
Dim cm7 As New SqlCommand
cm7.CommandType = CommandType.Text
cm7.Connection = Form2.cn
cm7.CommandText = "insert into my_tables values ('" &
parent1.Text & "', '" & db_name & "')"
cm7.ExecuteNonQuery()
table_constraints(parent1.Text)
Dim cm9 As New SqlCommand
cm9.CommandType = CommandType.Text
cm9.Connection = Form2.cn4
cm9.CommandText = "select * from my_tables where name = '" &
parent1.Text & "' and db_name = '" & db_name & "'"
Dim my_table1 As SqlDataReader = cm9.ExecuteReader
my_table1.Read()
For Each child1 In parent1.Nodes
    If child1.Checked = True Then
        Dim cm08 As New SqlCommand
        cm08.CommandType = CommandType.Text
        cm08.Connection = Form2.cn
        cm08.CommandText = "select * from " & db_name &
".INFORMATION_SCHEMA.columns where column_name = '" & child1.Text & "' and
table_name = '" & child1.Parent.Text & "'"
        Dim column_type As SqlDataReader = cm08.ExecuteReader
        column_type.Read()
        Dim cm10 As New SqlCommand
        cm10.CommandType = CommandType.Text
        cm10.Connection = Form2.cn1
        cm10.CommandText = "insert into my_columns values('" &
child1.Text & "', '" & my_table1.GetValue(0) & "', '" & column_type.GetValue(7) &
"')"
        'MessageBox.Show("pievienoju tabulai my_columns
child1.text & " (" & my_table1.GetValue(1) & ")")
        cm10.ExecuteNonQuery()
        column_type.Close()
    End If
Next
my_table1.Close()
Else
End If
line1:    my_tables.Close()
Next
Catch ex As Exception
    MessageBox.Show(ex.Message)
Finally
    Form2.cn5.Close()
    Form2.cn4.Close()
    Form2.cn3.Close()
    Form2.cn1.Close()

```

```

Form2.cn2.Close()
Form2.cn.Close()
End Try
'paarbauda vai jau nav atziimeets starptabulaa
Try
Form2.cn.Open()
Form2.cn1.Open()
Form2.cn2.Open()
Form2.cn3.Open()
Form2.cn4.Open()
Form2.cn5.Open()
Dim parent As TreeNode
Dim child As TreeNode
For Each parent In Form2.TreeView2.Nodes
Dim cm11 As New SqlCommand
cm11.CommandType = CommandType.Text
cm11.Connection = Form2.cn
cm11.CommandText = "select * from my_tables where name = '" &
RTrim(parent.Text) & "' and db_name = '" & db_name & "'"
Dim my_table As SqlDataReader = cm11.ExecuteReader
Dim cm12 As New SqlCommand
cm12.CommandType = CommandType.Text
cm12.Connection = Form2.cn1
cm12.CommandText = "select table_id, column_id from module_table
where module_id = " & module_id & ""
Dim module_table As SqlDataReader = cm12.ExecuteReader
my_table.Read()
If parent.Checked = True Then
For Each child In parent.Nodes
If child.Checked = True Then
Dim cm14 As New SqlCommand
cm14.CommandType = CommandType.Text
cm14.Connection = Form2.cn3
cm14.CommandText = "select * from my_columns where name
= '" & RTrim(child.Text) & "' and table_id = '" & my_table.GetValue(0) & "'"
Dim my_column As SqlDataReader = cm14.ExecuteReader
my_column.Read()
While module_table.Read
If module_table.GetValue(0) = my_table.GetValue(0)
Then
If module_table.GetValue(1) =
my_column.GetValue(0) Then
my_column.Close()
'module_table.Close()
GoTo line2
End If
End If
End While
Dim cm13 As New SqlCommand
cm13.CommandType = CommandType.Text
cm13.Connection = Form2.cn2
'my_column.Read()
cm13.CommandText = "insert into module_table values ('"
& module_id & "', '" & my_table.GetValue(0) & "', '" & my_column.GetValue(0) & "'"
cm13.ExecuteNonQuery()
'MessageBox.Show("pievienoju tabulai module_table " &
my_table.GetValue(1) & " un " & my_column.GetValue(1))
my_column.Close()
Else
'if child.checked = false
While module_table.Read
Dim cm18 As New SqlCommand
cm18.CommandText = CommandType.Text
cm18.Connection = Form2.cn5
cm18.CommandText = "select * from my_columns "
Dim column As SqlDataReader = cm18.ExecuteReader

```

```

        While column.Read
            If RTrim(column.GetValue(1)).ToLower =
RTrim(child.Text).ToLower Then
                Dim cm16 As New SqlCommand
                cm16.CommandType = CommandType.Text
                cm16.Connection = Form2.cn2
                cm16.CommandText = " delete from
module_table where column_id = " & column.GetValue(0) & " and module_id = " &
module_id & " and table_id = " & my_table.GetValue(0) & ""
                'MessageBox.Show(" izñemu no starptabulas
kolonna " & column.GetValue(1))
                cm16.ExecuteNonQuery()
                column.Close()
                GoTo line2
            End If
        End While
        column.Close()
    End While
    'my_table.Close()
    End If
line2:
    Next
Else
    'if parent.checked = false
    While module_table.Read
        Dim cm17 As New SqlCommand
        cm17.CommandType = CommandType.Text
        cm17.Connection = Form2.cn4
        cm17.CommandText = "select * from my_tables where db_name =
"" & db_name & ""
        Dim table As SqlDataReader = cm17.ExecuteReader
        While table.Read
            If RTrim(table.GetValue(1)) = RTrim(parent.Text) Then
                If module_table.GetValue(0) = my_table.GetValue(0)
Then
                    'MessageBox.Show(module_table.GetValue(0) & "
" & my_table.GetValue(0))
                    Dim cm15 As New SqlCommand
                    cm15.CommandType = CommandType.Text
                    cm15.Connection = Form2.cn2
                    cm15.CommandText = " delete from module_table
where table_id = " & my_table.GetValue(0) & " and module_id = " & module_id & ""
                    'MessageBox.Show("izñemu no starptabulas tabula
" & my_table.GetValue(1))
                    cm15.ExecuteNonQuery()
                End If
            End If
        End While
        table.Close()
    End While
    module_table.Close()
    End If
    module_table.Close()
    my_table.Close()
Next
Catch ex As Exception
    MessageBox.Show(ex.Message)
Finally
    Form2.cn5.Close()
    Form2.cn4.Close()
    Form2.cn3.Close()
    Form2.cn2.Close()
    Form2.cn1.Close()
    Form2.cn.Close()
End Try
End Sub

```

```

Public Sub delete_module()
    'poga "Dzēst"
    If Not Form2.ListBox1.SelectedItem = "" Then
        Dim nosaukums As String
        Dim module_id As String
        nosaukums = Form2.ListBox1.SelectedItem
        Dim cm2 As New SqlCommand("select * from my_modules where name = '" &
nosaukums & "'", Form2.cn3)
        cm2.CommandType = CommandType.Text
        cm2.Connection = Form2.cn3
        Dim cm3 As New SqlCommand
        cm3.CommandType = CommandType.Text
        cm3.Connection = Form2.cn
        Dim cm4 As New SqlCommand
        cm4.CommandType = CommandType.Text
        cm4.Connection = Form2.cn2
        'MessageBox.Show(nosaukums)
        Try
            Form2.cn3.Open()
            Form2.cn.Open()
            Form2.cn2.Open()
            Dim my_module As SqlDataReader = cm2.ExecuteReader
            my_module.Read()
            module_id = my_module.GetValue(0)
            my_module.Close()
            cm3.CommandText = "delete from module_table where module_id = " &
module_id & ""
            cm3.ExecuteNonQuery()
            cm4.CommandText = "delete from my_modules where module_id = " &
module_id & ""
            cm4.ExecuteNonQuery()
            Form2.ListBox1.Items.Remove(nosaukums)
            Form2.TreeView2.Nodes.Clear()
            MessageBox.Show("Modulis ir izdzēsts")
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        Finally
            Form2.cn2.Close()
            Form2.cn.Close()
            Form2.cn3.Close()
        End Try
    Else
        MessageBox.Show("Nav atzīmēts modulis")
    End If
End Sub
End Module

```

```
Imports System.Data.SqlClient
```

```
Module DB_DATA
```

```
'saglabā datus tabulā 'my_constaint'
```

```
Public Sub table_constraints(ByRef column_name As String)
```

```
Try
```

```
'cn1.Open()
```

```
Dim cm19 As New SqlCommand
```

```
cm19.CommandType = CommandType.Text
```

```
cm19.Connection = Form2.cn
```

```
cm19.CommandText = "select a.CONSTRAINT_NAME, a.TABLE_NAME,
```

```
a.TABLE_CATALOG, b.UNIQUE_CONSTRAINT_name, c.COLUMN_NAME, c.TABLE_NAME from " _
```

```
& Form2.database_combobox.Text &
```

```
".INFORMATION_SCHEMA.TABLE_CONSTRAINTS a, " _
```

```
& Form2.database_combobox.Text &
```

```
".INFORMATION_SCHEMA.REFERENTIAL_CONSTRAINTS b, " _
```

```
& Form2.database_combobox.Text &
```

```
".INFORMATION_SCHEMA.KEY_COLUMN_USAGE c " _
```

```

        & "where a.table_name = '" & column_name & "' and
a.CONSTRAINT_TYPE = 'foreign key' and c.CONSTRAINT_NAME = a.CONSTRAINT_NAME and
c.CONSTRAINT_NAME = b.CONSTRAINT_NAME "
    Dim constraint As SqlDataReader = cm19.ExecuteReader
    While constraint.Read
        Dim cm111 As New SqlCommand
        cm111.CommandType = CommandType.Text
        cm111.Connection = Form2.cn2
        cm111.CommandText = "select table_name, column_name from " &
Form2.database_combobox.Text & ".INFORMATION_SCHEMA.KEY_COLUMN_USAGE " _
        & " where constraint_name = '" & constraint.GetValue(3) & "'"
        Dim prim_constraint As SqlDataReader = cm111.ExecuteReader
        prim_constraint.Read()
        Dim cm18 As New SqlCommand
        cm18.CommandType = CommandType.Text
        cm18.Connection = Form2.cn1
        cm18.CommandText = "insert into my_constraint values('" &
RTrim(LTrim(constraint.GetValue(0))) & "', '" &
RTrim(LTrim(constraint.GetValue(1))) & "', '" & RTrim(LTrim(constraint.GetValue(4)))
& "', '" & RTrim(constraint.GetValue(3)) & "', '" &
RTrim(LTrim(prim_constraint.GetValue(0))) & "', '" &
RTrim(LTrim(prim_constraint.GetValue(1))) & "', '" &
RTrim(LTrim(prim_constraint.GetValue(2))) & "'" )"
        cm18.ExecuteNonQuery()
        prim_constraint.Close()
    End While
    constraint.Close()
Catch ex As Exception
    MessageBox.Show(ex.Message)
Finally
    'cn1.Close()
End Try
End Sub

'pievienojot jaunu moduli saglabā vērtības sistēmas datubāzē
Public Sub save_new_module()
    'saglabā datubāzē tabulu, kolonnu nosaukumus
    Dim Tree As TreeView = Form2.TreeView1
    If Form2.TreeView1.Visible = False Then
        Tree = Form2.TreeView2
    End If
    Dim table_text As String
    Dim table_read As String
    Dim title As String
    Dim cm1 As New SqlCommand
    cm1.CommandType = CommandType.Text
    cm1.Connection = Form2.cn
    Dim cm3 As New SqlCommand
    cm3.CommandType = CommandType.Text
    cm3.Connection = Form2.cn3
    cm3.CommandText = "select * from my_modules"
    Dim cm4 As New SqlCommand
    cm4.CommandType = CommandType.Text
    cm4.Connection = Form2.cn3
    Dim cm5 As New SqlCommand
    cm5.CommandType = CommandType.Text
    cm5.Connection = Form2.cn3
    Dim cm6 As New SqlCommand
    cm6.CommandType = CommandType.Text
    cm6.Connection = Form2.cn4
    Dim cm7 As New SqlCommand("select * from my_columns", Form2.cn2)
    cm7.CommandType = CommandType.Text
    cm7.Connection = Form2.cn2
    Dim cm8 As New SqlCommand
    cm8.CommandType = CommandType.Text
    cm8.Connection = Form2.cn4

```

```

Form2.error_Label.Text = ""
'ja modulis tiek redigets, tad nolasa modulja_id
Dim module_id2 As String
Dim db_name As String

If Form2.TreeView2.Visible = True Then
    Dim cm20 As New SqlCommand
    cm20.CommandType = CommandType.Text
    cm20.Connection = Form2.cn
    cm20.CommandText = "select * from my_modules where name = '" &
Form2.ListBox1.SelectedItem & "'"
    Try
        Form2.cn.Open()
        Dim module_id_db As SqlDataReader = cm20.ExecuteReader
        While module_id_db.Read
            module_id2 = module_id_db.GetValue(0)
            db_name = module_id_db.GetValue(2)
        End While
        module_id_db.Close()
    Catch ex As Exception
        MessageBox.Show(ex.Message)
    Finally
        Form2.cn.Close()
    End Try
End If
Dim sk4 As Integer = 0
Dim parent As TreeNode

For Each parent In Tree.Nodes
    If parent.Checked Then
        sk4 = sk4 + 1
    End If
    'Dim child As TreeNode
    'child = parent.Nodes
    For Each TreeNode In parent.Nodes
        If TreeNode.Checked Then
            sk4 = sk4 + 1
        End If
    Next
Next
If Form2.TextBox1.Text = "" And sk4 = 0 Then
    Form2.error_Label.Text = "Ievadiet nosaukumu!" & vbNewLine & "Nav
atzimēta neviena tabula!"
Else
    If sk4 = 0 And Form2.TextBox1.Text.Length > 50 Then
        Form2.error_Label.Text = "Nav atzīmēta neviena tabula! " &
vbNewLine & "Nosaukums nedrīkst būt garāks par 50 simboliem! "
    Else
        If sk4 = 0 Then
            Form2.error_Label.Text = "Nav atzīmēta neviena tabula! " &
vbNewLine & ""
        Else
            If Form2.TextBox1.Text = "" Then
                Form2.error_Label.Text = "Ievadiet nosaukumu! " & vbNewLine
& ""
            Else
                If Form2.TextBox1.Text.Length > 50 Then
                    Form2.error_Label.Text = "Nosaukums nedrīkst būt garāks
par 50 simboliem! " & vbNewLine & ""
                Else
                    Try
                        Form2.cn.Open()
                        Form2.cn3.Open()
                        Form2.cn4.Open()
                        Form2.cn5.Open()
                        Form2.cn1.Open()
                    End Try
                End If
            End If
        End If
    End If
End If

```

```

        title = Form2.TextBox1.Text
        Dim text As SqlDataReader = cm3.ExecuteReader
        While text.Read
            If Form2.TreeView1.Visible = True Then
                If RTrim(text.GetValue(1)).ToLower = title
                    Then
                        Form2.error_Label.Text = "Šāds
nosaukums jau eksistē! " & vbNewLine & ""
                        Form2.cn5.Close()
                        Form2.cn2.Close()
                        Form2.cn4.Close()
                        Form2.cn3.Close()
                        Form2.cn.Close()
                        GoTo line3 'ja nosaukums jau eksistē,
tad iet uz beigām
                    End If
                ElseIf Form2.TreeView2.Visible = True Then
                    If Not text.GetValue(0) = module_id2 Then
                        If RTrim(text.GetValue(1)).ToLower =
title Then
                            'MessageBox.Show(text.GetValue(0) &
" " & module_id2 & vbNewLine & RTrim(text.GetValue(1)).ToLower & " " & title)
                            Form2.error_Label.Text = "Šāds
nosaukums jau eksistē!! " & vbNewLine & ""
                            Form2.cn5.Close()
                            Form2.cn2.Close()
                            Form2.cn4.Close()
                            Form2.cn3.Close()
                            Form2.cn.Close()
                            GoTo line3 'ja nosaukums jau
eksistē, tad iet uz beigām
                        End If
                    End If
                End If
            End While
            text.Close()
            If Form2.TreeView1.Visible = True Then
                cm4.CommandText = "insert into my_modules
values ('" & title & "', '" & Form2.database_combobox.Text & "')"
            Else
                cm4.CommandText = "update my_modules set name =
'" & Form2.TextBox1.Text & "' where module_id = '" & module_id2 & "'"
                'MessageBox.Show("nosaukums ir izmainiits")
            End If
            cm4.ExecuteNonQuery()
            'saglabā kolonnas un tabulas
            If Form2.TreeView1.Visible = True Then
                Dim obj As TreeNode
                For Each obj In Form2.TreeView1.Nodes
                    If obj.Checked = True Then
                        table_text = CStr(obj.Text)
                        Dim cm2 As New SqlCommand("select *
from my_tables", Form2.cn3)
                        cm2.CommandType = CommandType.Text
                        cm2.Connection = Form2.cn3
                        Dim table As SqlDataReader =
cm2.ExecuteReader
                        While table.Read
                            table_read =
RTrim(table.GetValue(1))
                            If (RTrim(table_text).ToLower =
RTrim(table_read).ToLower And Form2.database_combobox.Text =
RTrim(table.GetValue(2))) Then
                                GoTo Line2
                            End If
                        End While
                    End If
                End While
            End If
        End While
    End Sub

```

```

        cm1.CommandText = "insert into
my_tables values (' & table_text & ', ' & Form2.database_combobox.Text & ')"
        cm1.ExecuteNonQuery()
        Form2.cn2.Open()
        table_constraints(table_text)
        Form2.cn2.Close()

line2:
        table.Close()
        'cn2.Close()
    End If
Next
End If
If Form2.TreeView1.Visible = True Then
    For Each obj In Form2.TreeView1.Nodes
        Form2.cn2.Open()
        If obj.Checked Then
            table_text = CStr(obj.Text)
            Dim cm2 As New SqlCommand
            cm2.CommandType = CommandType.Text
            cm2.Connection = Form2.cn3
            If Form2.TreeView1.Visible = False Then
                cm2.CommandText = "select * from
my_tables where name = ' & RTrim(table_text) & ' and db_name = ' & db_name & '"
            Else
                cm2.CommandText = "select * from
my_tables where name = ' & RTrim(table_text) & ' and db_name = ' &
Form2.database_combobox.Text & '"

            End If
            Dim table As SqlDataReader =
cm2.ExecuteReader()

            table.Read()
            Dim table_id As String =
table.GetValue(0)

            table.Close()
            Dim child As TreeNode
            For Each child In obj.Nodes
                If child.Checked = True Then
                    Dim column As SqlDataReader =
cm7.ExecuteReader

                    While column.Read
                        If
RTrim(column.GetValue(1)).ToLower = RTrim(child.Text).ToLower And
column.GetValue(2) = table_id Then

                            column.Close()
                            GoTo line4
                        End If
                        'table_id_2 =

table.GetValue(0)

                    End While
                    Dim cm08 As New SqlCommand
                    cm08.CommandType =
CommandType.Text

                    cm08.Connection = Form2.cn
                    cm08.CommandText = "select *
from " & Form2.database_combobox.Text & ".INFORMATION_SCHEMA.columns where
column_name = ' & child.Text & ' and table_name = ' & child.Parent.Text & '"
                    Dim column_type As
SqlDataReader = cm08.ExecuteReader

                    column_type.Read()
                    cm8.CommandText = "insert into
my_columns values (' & child.Text & ', ' & table_id & ', ' &
column_type.GetValue(7) & ')"

                    cm8.ExecuteNonQuery()
                    column_type.Close()
                    column.Close()

line4:
                End If
            End If
        End If
    End For
End If

```

```

        Next
    End If
    Form2.cn2.Close()
Next
End If
'saglabā vērtības starptabulā
'MessageBox.Show(title)
If Form2.TreeView1.Visible = True Then
    cm5.CommandText = "select * from my_modules
where name = '" & title & "'"
cm5.ExecuteReader
    Dim title_read As SqlDataReader =
    title_read.Read()
    Dim title_text_int As Integer =
    title_read.GetValue(0) 'moduļa nosaukuma ID
    Dim title_text_str As String =
    title_read.GetValue(1) 'moduļa nosaukums
    title_read.Close()
    'MessageBox.Show(title_text_int)
    For Each obj In Form2.TreeView1.Nodes
        If obj.Checked Then
            table_text = CStr(obj.Text).ToLower
            Dim cm2 As New SqlCommand
            If Form2.TreeView1.Visible = True Then
                cm2.CommandText = "select * from
my_tables where db_name = '" & Form2.database_combobox.Text & "'"
            Else
                cm2.CommandText = "select * from
my_tables where db_name = '" & Form2.database_combobox.Text & "'"
            End If
            cm2.CommandType = CommandType.Text
            cm2.Connection = Form2.cn3

            Dim table2 As SqlDataReader =

cm2.ExecuteReader
            While table2.Read
                Dim cm9 As New SqlCommand("select *
from my_columns where table_id = '" & table2.GetValue(0) & "'", Form2.cn5)
                cm9.CommandType = CommandType.Text
                Dim column As SqlDataReader =

cm9.ExecuteReader
                Dim value As String =

RTrim(table2.GetValue(0)).ToLower
                While column.Read
                    Dim child As TreeNode
                    For Each child In obj.Nodes
                        If child.Checked Then
                            If
RTrim(table_text.ToLower) = RTrim(table2.GetValue(1).ToLower) And
RTrim(table2.GetValue(2)) = Form2.database_combobox.Text Then
                                If
RTrim(child.Text.ToLower) = RTrim(column.GetValue(1)).ToLower And
table2.GetValue(0) = column.GetValue(2) Then
                                    'MessageBox.Show("no formas - " & child.Text & " no BD - " &
RTrim(column.GetValue(1)).ToLower & vbNewLine & " no my_table - " &
table2.GetValue(0) & "no my_column - " & column.GetValue(2))
                                        cm6.CommandText
= " insert into module_table values ('" & title_text_int & "', '" & value & "', '"
& column.GetValue(0) & "'"
cm6.ExecuteScalar()
                                    End If
                                End If
                            End If
                        End If
                    End If
                End If
            End If
        End If
    End If
End If

```

```

        Next
        End While
        column.Close()
line1:    End While
        table2.Close()
        End If
        Next
        MessageBox.Show("Modulis ir pievienots")
        End If
    Catch ex As Exception
        MessageBox.Show(ex.Message)
    Finally
        Form2.cn1.Close()
        Form2.cn5.Close()
        Form2.cn2.Close()
        Form2.cn4.Close()
        Form2.cn3.Close()
        Form2.cn.Close()
    End Try
    If Form2.TreeView2.Visible = True Then
        redig_module()
        MessageBox.Show("Modulis ir saglabāts")
        Tree.Nodes.Clear()
        Form2.TextBox1.Text = ""
    End If
End If
End If
End If
End If
line3: End If
        Form2.CheckBox2.Checked = False
        listBox_refresh()
        treeview3_data()
    End Sub

'rediģējot moduli saglabā vērtības datubāzē. ja nepieciešams, tad dzēš vērtības
Private Sub redig_module()
    'noskaidro datubaazi un modulja id
    Dim cm1 As New SqlCommand("select * from my_modules where name = '" &
Form2.TextBox1.Text & "'", Form2.cn)
    cm1.CommandType = CommandType.Text
    cm1.Connection = Form2.cn
    Dim module_id As String
    Dim db_name As String
    Try
        Form2.cn.Open()
        Dim mod_id As SqlDataReader = cm1.ExecuteReader
        mod_id.Read()
        module_id = mod_id.GetValue(0)
        db_name = mod_id.GetValue(2)
        mod_id.Close()
    Catch ex As Exception
        MessageBox.Show(ex.Message)
    Finally
        Form2.cn.Close()
    End Try
    Try
        Dim parent1 As TreeNode
        Form2.cn.Open()
        Form2.cn2.Open()
        Form2.cn1.Open()
        Form2.cn3.Open()
        Form2.cn4.Open()
        Form2.cn5.Open()
        For Each parent1 In Form2.TreeView2.Nodes
            Dim cm5 As New SqlCommand

```

```

cm5.Connection = Form2.cn3
cm5.CommandType = CommandType.Text
cm5.CommandText = "select * from my_tables where db_name = '" &
db_name & "'"
Dim my_tables As SqlDataReader = cm5.ExecuteReader
If parent1.Checked = True Then
    'paarbauda vai atziimeetaas tabulas un kolonnas jau ir
    datubaazee
        While my_tables.Read
            'ja tabula jau ir saglabaata my_tables
            'MessageBox.Show(RTrim(my_tables.GetValue(1)).ToLower & "
" & RTrim(parent1.Text).ToLower)
            'MessageBox.Show(RTrim(my_tables.GetValue(2)).ToLower & "
" & db_name)
            If RTrim(my_tables.GetValue(1)).ToLower =
RTrim(parent1.Text).ToLower And RTrim(my_tables.GetValue(2)).ToLower =
RTrim(db_name).ToLower Then
                'MessageBox.Show("tabula jau ir " & parent.Text)
                Dim child1 As TreeNode
                For Each child1 In parent1.Nodes
                    If child1.Checked = True Then
                        Dim cm2 As New SqlCommand
                        cm2.CommandType = CommandType.Text
                        cm2.Connection = Form2.cn1
                        cm2.CommandText = "select * from my_columns
where table_id = '" & my_tables.GetValue(0) & "'"
                        Dim my_columns As SqlDataReader =
cm2.ExecuteReader
                        While my_columns.Read
                            If RTrim(my_columns.GetValue(1)).ToLower =
RTrim(child1.Text).ToLower Then
                                'MessageBox.Show(child.Text & "
shitaads child jau ir")
                                my_columns.Close()
                                GoTo line3
                            End If
                        End While
                        Dim cm08 As New SqlCommand
                        cm08.CommandType = CommandType.Text
                        cm08.Connection = Form2.cn
                        cm08.CommandText = "select * from " & db_name &
".INFORMATION_SCHEMA.columns where column_name = '" & child1.Text & "' and
table_name = '" & child1.Parent.Text & "'"
                        Dim column_type As SqlDataReader =
cm08.ExecuteReader
                        column_type.Read()
                        Dim cm8 As New SqlCommand
                        cm8.CommandType = CommandType.Text
                        cm8.Connection = Form2.cn
                        cm8.CommandText = "insert into my_columns
values('" & child1.Text & "', '" & RTrim(my_tables.GetValue(0)) & "', '" &
column_type.GetValue(7) & "'"
                        cm8.ExecuteNonQuery()
                        'MessageBox.Show("pievienoju tabulai my_columns
" & child1.Text & " (" & RTrim(my_tables.GetValue(1)) & ")")
                        column_type.Close()
                        my_columns.Close()
                    End If
                Next
            my_tables.Close()
            GoTo line1
        End If
    End While
    Dim cm7 As New SqlCommand
    cm7.CommandType = CommandType.Text
    cm7.Connection = Form2.cn

```

```

        cm7.CommandText = "insert into my_tables values ('" &
parent1.Text & "',''" & db_name & "')"
        cm7.ExecuteNonQuery()
        table_constraints(parent1.Text)
        Dim cm9 As New SqlCommand
        cm9.CommandType = CommandType.Text
        cm9.Connection = Form2.cn4
        cm9.CommandText = "select * from my_tables where name = '" &
parent1.Text & "' and db_name = '" & db_name & "'"
        Dim my_table1 As SqlDataReader = cm9.ExecuteReader
        my_table1.Read()
        For Each child1 In parent1.Nodes
            If child1.Checked = True Then
                Dim cm08 As New SqlCommand
                cm08.CommandType = CommandType.Text
                cm08.Connection = Form2.cn
                cm08.CommandText = "select * from " & db_name &
".INFORMATION_SCHEMA.columns where column_name = '" & child1.Text & "' and
table_name = '" & child1.Parent.Text & "'"
                Dim column_type As SqlDataReader = cm08.ExecuteReader
                column_type.Read()
                Dim cm10 As New SqlCommand
                cm10.CommandType = CommandType.Text
                cm10.Connection = Form2.cn1
                cm10.CommandText = "insert into my_columns values('" &
child1.Text & "',''" & my_table1.GetValue(0) & "',''" & column_type.GetValue(7) &
"')"
                'MessageBox.Show("pievienoju tabulai my_columns " &
child1.text & " (" & my_table1.GetValue(1) & ")")
                cm10.ExecuteNonQuery()
                column_type.Close()
            End If
        Next
        my_table1.Close()
    Else
    End If
line1: my_tables.Close()
Next
Catch ex As Exception
    MessageBox.Show(ex.Message)
Finally
    Form2.cn5.Close()
    Form2.cn4.Close()
    Form2.cn3.Close()
    Form2.cn1.Close()
    Form2.cn2.Close()
    Form2.cn.Close()
End Try
'paarbauda vai jau nav atziimeets starptabulaa
Try
    Form2.cn.Open()
    Form2.cn1.Open()
    Form2.cn2.Open()
    Form2.cn3.Open()
    Form2.cn4.Open()
    Form2.cn5.Open()
    Dim parent As TreeNode
    Dim child As TreeNode
    For Each parent In Form2.TreeView2.Nodes
        Dim cm11 As New SqlCommand
        cm11.CommandType = CommandType.Text
        cm11.Connection = Form2.cn
        cm11.CommandText = "select * from my_tables where name = '" &
RTrim(parent.Text) & "' and db_name = '" & db_name & "'"
        Dim my_table As SqlDataReader = cm11.ExecuteReader
        Dim cm12 As New SqlCommand

```

```

        cm12.CommandType = CommandType.Text
        cm12.Connection = Form2.cn1
        cm12.CommandText = "select table_id, column_id from module_table
where module_id = " & module_id & ""
        Dim module_table As SqlDataReader = cm12.ExecuteReader
        my_table.Read()
        If parent.Checked = True Then
            For Each child In parent.Nodes
                If child.Checked = True Then
                    Dim cm14 As New SqlCommand
                    cm14.CommandType = CommandType.Text
                    cm14.Connection = Form2.cn3
                    cm14.CommandText = "select * from my_columns where name
= '" & RTrim(child.Text) & "' and table_id = '" & my_table.GetValue(0) & "'"
                    Dim my_column As SqlDataReader = cm14.ExecuteReader
                    my_column.Read()
                    While module_table.Read
                        If module_table.GetValue(0) = my_table.GetValue(0)
Then
                            If module_table.GetValue(1) =
my_column.GetValue(0) Then
                                my_column.Close()
                                'module_table.Close()
                                GoTo line2
                            End If
                        End If
                    End While
                    Dim cm13 As New SqlCommand
                    cm13.CommandType = CommandType.Text
                    cm13.Connection = Form2.cn2
                    'my_column.Read()
                    cm13.CommandText = "insert into module_table values ('"
& module_id & "',''" & my_table.GetValue(0) & "',''" & my_column.GetValue(0) & "'"")"
                    cm13.ExecuteNonQuery()
                    'MessageBox.Show("pievienoju tabulai module_table " &
my_table.GetValue(1) & " un " & my_column.GetValue(1))
                    my_column.Close()
                Else
                    'if child.checked = false
                    While module_table.Read
                        Dim cm18 As New SqlCommand
                        cm18.CommandText = CommandType.Text
                        cm18.Connection = Form2.cn5
                        cm18.CommandText = "select * from my_columns "
                        Dim column As SqlDataReader = cm18.ExecuteReader
                        While column.Read
                            If RTrim(column.GetValue(1)).ToLower =
RTrim(child.Text).ToLower Then
                                Dim cm16 As New SqlCommand
                                cm16.CommandType = CommandType.Text
                                cm16.Connection = Form2.cn2
                                cm16.CommandText = " delete from
module_table where column_id = " & column.GetValue(0) & " and module_id = " &
module_id & " and table_id = " & my_table.GetValue(0) & ""
                                'MessageBox.Show(" izņemu no starptabulas
kolonna " & column.GetValue(1))

                                cm16.ExecuteNonQuery()
                                column.Close()
                                GoTo line2
                            End If
                        End While
                    End While
                    column.Close()
                End While
                'my_table.Close()
            End If
        End If
    Next
line2:

```

```

Else
    'if parent.checked = false
    While module_table.Read
        Dim cm17 As New SqlCommand
        cm17.CommandType = CommandType.Text
        cm17.Connection = Form2.cn4
        cm17.CommandText = "select * from my_tables where db_name =
'" & db_name & "'"
        Dim table As SqlDataReader = cm17.ExecuteReader
        While table.Read
            If RTrim(table.GetValue(1)) = RTrim(parent.Text) Then
                If module_table.GetValue(0) = my_table.GetValue(0)
Then
                    'MessageBox.Show(module_table.GetValue(0) & "
" & my_table.GetValue(0))
                    Dim cm15 As New SqlCommand
                    cm15.CommandType = CommandType.Text
                    cm15.Connection = Form2.cn2
                    cm15.CommandText = " delete from module_table
where table_id = " & my_table.GetValue(0) & " and module_id = " & module_id & "
" & my_table.GetValue(1))
                    'MessageBox.Show("iznemu no starptabulas tabula
cm15.ExecuteNonQuery()
                End If
            End If
        End While
        table.Close()
    End While
    module_table.Close()
End If
module_table.Close()
my_table.Close()
Next
Catch ex As Exception
    MessageBox.Show(ex.Message)
Finally
    Form2.cn5.Close()
    Form2.cn4.Close()
    Form2.cn3.Close()
    Form2.cn2.Close()
    Form2.cn1.Close()
    Form2.cn.Close()
End Try
End Sub

Public Sub delete_module()
    'poga "Dzēst"
    If Not Form2.ListBox1.SelectedItem = "" Then
        Dim nosaukums As String
        Dim module_id As String
        nosaukums = Form2.ListBox1.SelectedItem
        Dim cm2 As New SqlCommand("select * from my_modules where name = '" &
nosaukums & "'", Form2.cn3)
        cm2.CommandType = CommandType.Text
        cm2.Connection = Form2.cn3
        Dim cm3 As New SqlCommand
        cm3.CommandType = CommandType.Text
        cm3.Connection = Form2.cn
        Dim cm4 As New SqlCommand
        cm4.CommandType = CommandType.Text
        cm4.Connection = Form2.cn2
        'MessageBox.Show(nosaukums)
        Try
            Form2.cn3.Open()
            Form2.cn.Open()
            Form2.cn2.Open()

```

```

        Dim my_module As SqlDataReader = cm2.ExecuteReader
        my_module.Read()
        module_id = my_module.GetValue(0)
        my_module.Close()
        cm3.CommandText = "delete from module_table where module_id = " &
module_id & ""
        cm3.ExecuteNonQuery()
        cm4.CommandText = "delete from my_modules where module_id = " &
module_id & ""
        cm4.ExecuteNonQuery()
        Form2.ListBox1.Items.Remove(nosaukums)
        Form2.TreeView2.Nodes.Clear()
        MessageBox.Show("Modulis ir izdzēsts")
    Catch ex As Exception
        MessageBox.Show(ex.Message)
    Finally
        Form2.cn2.Close()
        Form2.cn.Close()
        Form2.cn3.Close()
    End Try
Else
    MessageBox.Show("Nav atzīmēts modulis")
End If
End Sub
End Module

Imports System.Data.SqlClient
Imports System.IO

Module SQL_query
    Public sql_all As String
    Private sql_where As String = ""
    Public Sub sql_select2()
        sql_all = ""
        Dim sql_cn As New SqlConnection
        Form2.DataGridView1.Columns.Clear()
        Dim sql_select2 As String = "SELECT "
        Dim sql_from2 As String = " FROM "
        Dim sql_join As String = " "
        Dim control As UserControl1
        Dim table_from As String = ""
        For Each control In Form2.SplitContainer1.Panel1.Controls
            sql_select2 = sql_select2 & " " & control.table.Text & "." &
control.Label1.Text & ","
            'If Not control.table.Text = table_from Then
            Try
                Form2.cn.Open()
                Dim cm91 As New SqlCommand
                cm91.CommandType = CommandType.Text
                cm91.Connection = Form2.cn
                cm91.CommandText = "select * from my_constraint where fk_table = '"
& Form2.selected_table_text & "' and db_name = '" & Form2.selected_db & "'"
                Dim constr As SqlDataReader = cm91.ExecuteReader
                While constr.Read
                    If control.Label1.Text = RTrim(LTrim(constr.GetValue(3))) Then
                        'sql_from2 = sql_from2 & " " & constr.GetValue(2) & " JOIN
" & constr.GetValue(5) & " ON " _
                        '
                        & constr.GetValue(2) & "." & constr.GetValue(3) & " =
" & constr.GetValue(5) & "." & constr.GetValue(6) & ","
                        sql_join = sql_join & " JOIN " & constr.GetValue(5) & " ON
" _
                        & constr.GetValue(2) & "." & constr.GetValue(3) & " = "
& constr.GetValue(5) & "." & constr.GetValue(6) & " "
                    End If
                End While
                'MessageBox.Show(sql_from2)
            
```

```

        constr.Close()
    Catch ex As Exception
        MessageBox.Show(ex.Message)
    Finally
        Form2.cn.Close()
    End Try
    'sql_from2 = sql_from2 & " " & control.table.Text & ","
    'End If
    'table_from = control.table.Text
Next
sql_from2 = sql_from2 & " " & Form2.selected_table_text & " " & sql_join
sql_select2 = sql_select2.Remove(sql_select2.Count - 1, 1)
sql_from2 = sql_from2.Remove(sql_from2.Count - 1, 1)
sql_all = sql_all & sql_select2 & sql_from2
'MessageBox.Show(sql_all)

Dim source As String = "Data Source=SERVER\SQLEXPRESS; "
Dim catalog As String = "Initial Catalog= " & Form2.selected_db & "; "
Dim security As String = "Integrated Security=True "
sql_cn.ConnectionString = source + catalog + security
sql_all = sql_select2 + sql_from2
Try
    sql_cn.Open()
    Dim ad As New SqlDataAdapter()
    Dim ds As New DataSet
    ds.Tables.Add("Table1")
    sql_all = sql_all + sql_where
    ad.SelectCommand = New SqlCommand(sql_all, sql_cn)
    ad.Fill(ds.Tables(0))
    Form2.DataGridView1.DataSource = ds.Tables(0)
Catch ex As Exception
    MessageBox.Show(ex.Message)
Finally
    sql_cn.Close()
End Try
End Sub
Public Sub sql_select()
    sql_all = ""
    Form2.DataGridView1.Columns.Clear()
    Dim sql_select As String = "SELECT "
    Dim sql_from As String = " FROM "
    Dim xnode As String = Form2.selected_table_text
    Dim sql_cn As New SqlConnection
    If Not xnode = "" Then
        If Form2.selected_node_level = 1 Then
            sql_from = sql_from + xnode
            Dim control4 As UserControl1
            For Each control4 In Form2.SplitContainer1.Panel1.Controls
                sql_select = sql_select + " " & control4.table.Text & "." &
control4.Label1.Text & ","
            Next
            'sql_select = sql_select & ","
            sql_select = sql_select.Remove(sql_select.Count - 1, 1)
            sql_all = sql_select + sql_from
            'MessageBox.Show(sql_all)
        ElseIf Form2.selected_node_level = 2 Then
            'Dim control5 As UserControl1
            'For Each control5 In Form2.SplitContainer1.Panel1.Controls
            sql_select = sql_select + Form2.selected_column_text
            sql_from = sql_from + Form2.selected_table_text
            sql_all = sql_select + sql_from
            'MessageBox.Show(sql_all)
            'Next
        Else
            GoTo line7
        End If
    End If

```

```

Dim source As String = "Data Source=SUPERPC\SQLEXPRESS; "
Dim catalog As String = "Initial Catalog= " + Form2.selected_db + "; "
Dim security As String = "Integrated Security=True "
sql_cn.ConnectionString = source + catalog + security
Try
    sql_cn.Open()
    Dim ad As New SqlDataAdapter()
    Dim ds As New DataSet
    ds.Tables.Add("Table1")
    sql_all = sql_all + sql_where
    ad.SelectCommand = New SqlCommand(sql_all, sql_cn)
    ad.Fill(ds.Tables(0))
    Form2.DataGridView1.DataSource = ds.Tables(0)
Catch ex As Exception
    MessageBox.Show(ex.Message)
Finally
    sql_cn.Close()
End Try
'filter()
Else
    'MessageBox.Show("Nav atzīmēta tabula")
line7: End If
End Sub
Public Sub sql_where_9()
    Dim control As UserControl1
    Dim temp_name As String = ""
    Dim skk As Integer = 1
    sql_where = sql_where + " WHERE "
    Dim temp_where As String = ""
    For Each control In Form2.SplitContainer1.Panel1.Controls
        If Not control.TextBox1.Text = "" Then
            If control.TextBox1.Text.Contains(" and ") And Not
control.TextBox1.Text.Contains("between") Then
                Dim line2 As Array
                line2 = control.TextBox1.Text.Split(" ")
                Dim ind As Integer = 0
                While ind < line2.Length
                    If control.TextBox1.Text.Contains(" not ") Or
Left(control.TextBox1.Text, 4) = "not " Then
                        If Not LTrim(RTrim(line2(ind).ToString)) = "and" Then
                            If Not LTrim(RTrim(line2(ind).ToString)) = "not"
Then
                                If control.TextBox1.Text.Contains("%") Or
control.TextBox1.Text.Contains("_") Then
                                    temp_where = temp_where & " " &
control.table.Text & "." & control.Label1.Text & " LIKE " & "" &
line2(ind).ToString & " and "
                                Else
                                    temp_where = temp_where & " " &
control.table.Text & "." & control.Label1.Text & " = " & "" & line2(ind).ToString
& " and "
                                End If
                            Else
                                temp_where = temp_where & " not "
                            End If
                        End If
                    End If
                    ind = ind + 1
                Else
                    If Not LTrim(RTrim(line2(ind).ToString)) = "and" Then
                        If control.TextBox1.Text.Contains("%") Or
control.TextBox1.Text.Contains("_") Then
                            temp_where = temp_where & " " &
control.table.Text & "." & control.Label1.Text & " LIKE " & "" &
line2(ind).ToString & " and "
                        Else
                    
```

```

temp_where = temp_where & " " &
control.table.Text & "." & control.Label1.Text & " = " & "" & line2(ind).ToString
& "' and "
        End If
    End If
    ind = ind + 1
End If
End While
sql_where = sql_where + temp_where
sql_where = sql_where.Remove(sql_where.Count - 4, 4)
sql_where = sql_where & " and "

If control.TextBox1.Text.Contains("%") Or
control.TextBox1.Text.Contains(">") Then

End If
Else
If Left(control.TextBox1.Text, 3) = "or " Then
'MessageBox.Show("Kļūda!")
GoTo line2
End If
If Right(control.TextBox1.Text, 3) = " or" Then
'MessageBox.Show("Kļūda!")
GoTo line2
End If
If control.TextBox1.Text.Contains(" or ") Then
Dim line As Array
line = control.TextBox1.Text.Split(" ")
If line.Length < 2 Then
GoTo line2
End If
Dim length As Integer = line.Length
If line(0) = RTrim(LTrim("or")) Then
GoTo line2
End If
Dim i As Integer = 0
While i < length
If Not LTrim(RTrim(line(i).ToString)) = "or" Then
If Not LTrim(RTrim(line(i).ToString)) = "not" Then
If Not control.TextBox1.Text.Contains("%") Then
temp_where = temp_where & " " &
control.table.Text & "." & control.Label1.Text & " = " & "" & line(i).ToString &
"' or "
Else
temp_where = temp_where & " " &
control.table.Text & "." & control.Label1.Text & " LIKE " & "" & line(i).ToString
& "' or "
End If
Else
temp_where = temp_where & " not "
End If
End If
i = i + 1
End While
sql_where = sql_where + temp_where
sql_where = sql_where.Remove(sql_where.Count - 4, 4)
sql_where = sql_where & " and "
'End If
Else
If control.TextBox1.Text.Contains(">") Or
control.TextBox1.Text.Contains("<") Then
Dim sign As String = ""
If Left(control.TextBox1.Text, 1) = ("<") Then
If Left(control.TextBox1.Text, 2) = ("<>") Then
sign = "<>"
Else

```

```

        If Left(control.TextBox1.Text, 2) = ("<=") Then
            sign = ">="
        Else
            sign = "<"
        End If
    End If
Else
    If Left(control.TextBox1.Text, 1) = (">") Then
        If Left(control.TextBox1.Text, 2) = (">=") Then
            sign = ">="
        Else
            sign = ">"
        End If
    Else
        GoTo line2
    End If
End If

Dim value As Array
value = control.TextBox1.Text.Split(" ")
'MessageBox.Show(value(1))
If Not Right$(sql_where, 4) = "or " Then
    sql_where = sql_where & control.table.Text & "." &
control.Label1.Text & " " & sign & " '" & RTrim(LTrim(value(1))) & "' and "
Else
    sql_where = sql_where.Remove(sql_where.Count - 4,
4)

    sql_where = sql_where & "and"
    sql_where = sql_where & control.table.Text & "." &
control.Label1.Text & " '" & sign & " '" & RTrim(LTrim(value(1))) & "' and "
    End If
Else
    If control.TextBox1.Text.Contains("%") Or
control.TextBox1.Text.Contains("_") Then
        ' MessageBox.Show(Left(control.TextBox1.Text, 4))
        If Not Right$(sql_where, 4) = "or " Then
            If Not Left$(control.TextBox1.Text, 4) = "not "
Then
                sql_where = sql_where & control.table.Text
& "." & control.Label1.Text & " LIKE '" & control.TextBox1.Text & "' and "
            Else
                Dim line3 As Array
                line3 = control.TextBox1.Text.Split(" ")
                sql_where = sql_where & control.table.Text
& "." & control.Label1.Text & " NOT LIKE '" & line3(1) & "' and "
                End If
            Else
                If Not Left(control.TextBox1.Text, 4) = "not"
Then
                    sql_where =
sql_where.Remove(sql_where.Count - 4, 4)
                    sql_where = sql_where & "and"
                    sql_where = sql_where & control.table.Text
& "." & control.Label1.Text & " LIKE '" & control.TextBox1.Text & "' and "
                Else
                    Dim line3 As Array
                    line3 = control.TextBox1.Text.Split(" ")
                    sql_where = sql_where & control.table.Text
& "." & control.Label1.Text & " NOT LIKE '" & line3(1) & "' and "
                    End If
                End If
            Else
                If control.TextBox1.Text.Contains("between") Then
                    If control.TextBox1.Text.Contains(" and") Then
                        Dim line2 As Array
                        line2 = control.TextBox1.Text.Split(" ")

```

```

        If line2.Length > 3 Then
            If line2(0) = "between" And line2(2) =
"and" Then
                Dim ind2 As Integer = line2.Length
                - 1
                sql_where = sql_where &
control.table.Text & "." & control.Label1.Text & " " & line2(0) & " '" & line2(1) &
"' " & line2(2) & " '" & line2(3) & " "
            Else
                MessageBox.Show("Nosacījumam jābūt
formā 'BETWEEN aaa AND zzz' ", "Info", MessageBoxButtons.OK,
MessageBoxIcon.Information)
                sql_where = ""
            End If
        End If
    Else
        MessageBox.Show("Nosacījumam jābūt formā
'BETWEEN aaa AND zzz' ", "Info", MessageBoxButtons.OK, MessageBoxIcon.Information)
        sql_where = ""
    End If
    Else
        sql_where = sql_where & control.table.Text &
"." & control.Label1.Text & " = '" & control.TextBox1.Text & "' and "
    End If
    End If
    End If
    End If
    End If
    End If
    Next
    'MessageBox.Show(sql_where)
    If Not sql_where = "" Then
        sql_where = sql_where.Remove(sql_where.Count - 4, 4)
    End If
    If sql_where.Count < 8 Then
        sql_where = ""
    End If
    'MessageBox.Show(sql_where)
    sql_all = sql_all + sql_where
    If sql_where.Contains(" AND ") And sql_where.Contains(" OR ") Then
line2:    MessageBox.Show("Kļūda!")
        sql_where = ""
    End If
    Dim contr As UserControl1
    For Each contr In Form2.SplitContainer1.Panel1.Controls
        If contr.Tag = "1" Then
            sql_select2()
            GoTo line1
        End If
    Next
    sql_select()
    'MessageBox.Show(sql_all)
line1:    sql_where = ""
    End Sub

End Module

Imports System.Data.SqlClient
Module data

    'funkcija nodrošina, ka informācija sarakstlodziņā tiek izmainīta, ja tiek
pievienots vai dzēsts kāds modulis.
    'nodrošina, ka tiek ielasīti datubāzu nosaukumi.
    Public Sub listbox_refresh()
        Dim cm1 As New SqlCommand("select name FROM sys.master_files where
database_id > '4' and file_id = '1'", Form2.cn1)

```

```

cm1.CommandType = CommandType.Text
Dim cm2 As New SqlCommand("select * from my_modules", Form2.cn3)
cm2.CommandType = CommandType.Text
If Form2.database_combobox.Items.Count = 0 Then
    Try
        Form2.cn1.Open()
        Dim Database As SqlDataReader = cm1.ExecuteReader
        While Database.Read
            Form2.database_combobox.Items.Add(RTrim(Database.GetValue(0)))
        End While
        Database.Close()
    Catch ex As Exception
        MessageBox.Show(ex.Message)
    Finally
        Form2.cn1.Close()
    End Try
End If
Try
    Form2.cn3.Open()
    Form2.ListBox1.Items.Clear() 'lai uzreiz tiktu atjaunināti tikko
pievienotie moduļi
    Form2.TreeView3.Nodes.Clear()
    Dim module_name As SqlDataReader = cm2.ExecuteReader
    While module_name.Read
        Form2.ListBox1.Items.Add(RTrim(module_name.GetValue(1)))
        Form2.TreeView3.Nodes.Add(RTrim(module_name.GetValue(1)))
    End While
    module_name.Close()
Catch ex As Exception
    MessageBox.Show(ex.Message)
Finally
    Form2.cn3.Close()
End Try
End Sub

'funkcija nodrošina datu attēlošanu sadaļā 'dati', koka skata konrolī
Public Sub treeview3_data()
    Dim parent As TreeNode
    For Each parent In Form2.TreeView3.Nodes
        parent.Nodes.Clear()
        Dim cm1 As New SqlCommand("select * from my_modules where name = '" &
parent.Text & "'", Form2.cn)
        cm1.CommandType = CommandType.Text
        Try
            Form2.cn.Open()
            Form2.cn2.Open()
            Form2.cn3.Open()
            Form2.cn4.Open()
            Form2.cn5.Open()
            Dim module_name As SqlDataReader = cm1.ExecuteReader
            module_name.Read()
            Dim cm2 As New SqlCommand("select distinct module_id, table_id from
module_table where module_id = '" & module_name.GetValue(0) & "'", Form2.cn2)
            cm2.CommandType = CommandType.Text
            Dim table_id As SqlDataReader = cm2.ExecuteReader
            While table_id.Read
                Dim cm3 As New SqlCommand("select * from my_tables where
table_id = '" & table_id.GetValue(1) & "'", Form2.cn3)
                cm3.CommandType = CommandType.Text
                Dim table As SqlDataReader = cm3.ExecuteReader
                table.Read()
                Dim xparent As New TreeNode
                xparent.Text = RTrim(table.GetValue(1))
                xparent.Tag = table.GetValue(0)
                Dim child As TreeNode
                child = parent.Nodes.Add(xparent.Text)
            End While
        End Try
    End For
End Sub

```

```

        Dim cm4 As New SqlCommand("select * from module_table where
table_id = '" & RTrim(table_id.GetValue(1)) & "' and module_id = '" &
module_name.GetValue(0) & "'", Form2.cn4)
        cm4.CommandType = CommandType.Text
        Dim column As SqlDataReader = cm4.ExecuteReader
        While column.Read
            Dim cm5 As New SqlCommand("select * from my_columns where
column_id = '" & column.GetValue(2) & "'", Form2.cn5)
            cm5.CommandType = CommandType.Text
            Dim column_name As SqlDataReader = cm5.ExecuteReader
            column_name.Read()
            Dim xchild As New TreeNode
            xchild.Text = RTrim(column_name.GetValue(1))
            xchild.Tag = column_name.GetValue(0)
            child.Nodes.Add(xchild)
            column_name.Close()
        End While
        column.Close()
        table.Close()
    End While
    module_name.Close()
    table_id.Close()
Catch ex As Exception
    MessageBox.Show(ex.Message)
Finally
    Form2.cn5.Close()
    Form2.cn4.Close()
    Form2.cn3.Close()
    Form2.cn2.Close()
    Form2.cn.Close()
End Try
'parent.Expand()
Next
End Sub

'funkcija tiek izmantota pievienojot jaunu moduli. atzīmējot tabulu tiek
nolasītas attiecīgās kolonnas un pievienotas kontroļi
Public Sub treeView1_AfterCheck(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.TreeViewEventArgs)
    Dim xnode As TreeNode = e.Node
    If xnode.Level = 1 Then
        Dim count As Integer = 0
        Dim parent As TreeNode
        parent = xnode.Parent
        Dim child As TreeNode
        For Each child In parent.Nodes 'lai uzzinātu atzīmēto elementu skaitu
            If child.Checked = True Then
                count = count + 1
            End If
        Next
        If count = 0 Then
            parent.Checked = False
        End If
    End If
    If xnode.Level = 0 Then
        Dim database_name As String =
Form2.database_combobox.SelectedItem.ToString
        Dim table_name As String = xnode.Text

        Dim cm2 As New SqlCommand("select COLUMN_NAME from " & database_name &
".INFORMATION_SCHEMA.COLUMNS where TABLE_NAME = '" & table_name & "'", Form2.cn1)
        cm2.CommandType = CommandType.Text
        If xnode.Checked Then
            Try
                Form2.cn1.Open()
                Dim Column_name As SqlDataReader = cm2.ExecuteReader
            
```

```

        While Column_name.Read
            Dim newNode As TreeNode = New
TreeNode(Column_name.GetValue(0))
            xnode.Nodes.Add(newNode)
        End While
        Column_name.Close()

    Catch ex As Exception
        MessageBox.Show(ex.Message)
    End Try
    Form2.cn1.Close()
    xnode.Expand()
    'Lai uzreiz tiktu atzīmētas tabulai piederošās kolonnas
    Dim child As TreeNode
    For Each child In xnode.Nodes
        child.Checked = True
    Next
Else
    xnode.Nodes.Clear()
End If
End Sub

'funkcija nepieciešama, lai būtu iespējams rediģēt moduli.
'no metadatiem nolasa visas kolonnas un tabulas
'no sistēmas datubāzes tiek nolasītas pievienotās tabulas un kolonnas
Public Sub module_select_for_redig()
    Form2.TreeView1.Visible = False
    Form2.TreeView2.Visible = True
    Form2.database_combobox.Visible = False
    Form2.Label_datubaze.Visible = False
    Form2.TreeView2.Nodes.Clear()
    Dim name As String = Form2.ListBox1.SelectedItem
    Form2.TextBox1.Text = name
    If Not name = "" Then
        Dim cm9 As New SqlCommand("select * from my_modules where name = '" &
Form2.ListBox1.SelectedItem & "'", Form2.cn3)
        cm9.CommandType = CommandType.Text
        Dim database_name As String
        'ielasa visas tabulas un kolonnas
        Try
            Form2.cn3.Open()
            Form2.cn.Open()
            Form2.cn1.Open()
            Dim database As SqlDataReader = cm9.ExecuteReader
            database.Read()
            database_name = database.GetValue(2)
            Dim module_id As String = database.GetValue(0)
            database.Close()
            Dim sk As Integer = -1
            Dim cm3 As New SqlCommand("select table_name from " & database_name
& ".INFORMATION_SCHEMA.tables", Form2.cn)
            cm3.CommandType = CommandType.Text
            Dim table As SqlDataReader = cm3.ExecuteReader
            While table.Read
                Form2.TreeView2.Nodes.Add(RTrim(table.GetValue(0)))
                sk = sk + 1
                Dim cm5 As New SqlCommand("select COLUMN_NAME from " &
database_name & ".INFORMATION_SCHEMA.COLUMNS where TABLE_NAME = '" &
table.GetValue(0) & "'", Form2.cn1)
                cm5.CommandType = CommandType.Text
                Dim column As SqlDataReader = cm5.ExecuteReader
                While column.Read()
                    Dim xnode As TreeNode
                    xnode = Form2.TreeView2.Nodes.Item(sk)
                    xnode.Nodes.Add(RTrim(column.GetValue(0)))
                End While
            End While
        End Try
    End If
End Sub

```

```

        End While
        column.Close()
    End While
    table.Close()
Catch ex As Exception
    MessageBox.Show(ex.Message)
Finally
    Form2.cn1.Close()
    Form2.cn.Close()
    Form2.cn3.Close()
End Try
'atzimē tabulas
Dim parent As TreeNode
For Each parent In Form2.TreeView2.Nodes
    Try
        Form2.cn4.Open()
        Form2.cn3.Open()
        Form2.cn.Open()
        Form2.cn2.Open()
        Form2.cn5.Open()
        Dim module_id_2 As String
        Dim module_id_db As SqlDataReader = cm9.ExecuteReader
        module_id_db.Read()
        module_id_2 = module_id_db.GetValue(0)
        module_id_db.Close()
        Dim database As SqlDataReader = cm9.ExecuteReader
        database.Read()
        database_name = database.GetValue(2)
        database.Close()
        Dim cm1 As New SqlCommand("select * from my_tables where
db_name = '' & database_name & '' and name = '' & parent.Text & ''", Form2.cn4)
        cm1.CommandType = CommandType.Text
        Dim my_table As SqlDataReader = cm1.ExecuteReader
        While my_table.Read
            Dim cm2 As New SqlCommand("select * from module_table where
module_id = '' & module_id_2 & '' and table_id = '' & my_table.GetValue(0) & ''",
Form2.cn)
            cm2.CommandType = CommandType.Text
            Dim check_table As SqlDataReader = cm2.ExecuteReader
            While check_table.Read()
                If check_table.GetValue(1) = my_table.GetValue(0) Then
                    'MessageBox.Show(parent.Text)
                    parent.Checked = True
                    parent.Expand()
                End If
            End While
            check_table.Close()
            'atziimee kolonnas
            If parent.Checked = True Then
                Dim child As TreeNode
                For Each child In parent.Nodes
                    Dim cm3 As New SqlCommand("select * from my_columns
where name = '' & child.Text & '' and table_id = '' & my_table.GetValue(0) & ''",
Form2.cn2)
                    cm3.CommandType = CommandType.Text
                    Dim column As SqlDataReader = cm3.ExecuteReader
                    While column.Read
                        Dim cm4 As New SqlCommand("select * from
module_table where module_id = '' & module_id_2 & '' and table_id = '' &
my_table.GetValue(0) & '' and column_id = '' & column.GetValue(0) & ''", Form2.cn5)
                        cm4.CommandType = CommandType.Text
                        Dim check_column As SqlDataReader =
cm4.ExecuteReader
                        While check_column.Read
                            If check_column.GetValue(2) =
column.GetValue(0) Then

```

```
        child.Checked = True
        End If
    End While
    check_column.Close()
End While
column.Close()
Next
End If
End While
my_table.Close()
Catch ex As Exception
    MessageBox.Show(ex.Message)
Finally
    Form2.cn5.Close()
    Form2.cn2.Close()
    Form2.cn.Close()
    Form2.cn3.Close()
    Form2.cn4.Close()
End Try
Next
End If
End Sub

End Module
```

Kvalifikācijas darbs „**Uz metadatiem balstīta LR Centrālās Statistikas pārvaldes datu apskates sistēma**” izstrādāts Latvijas Universitātes Datorikas fakultātē.

Ar savu parakstu apliecinu, ka darbs izstrādāts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: *Jērāne Austra* _____ .05.2011.

Rekomendēju darbu aizstāvēšanai

Darba vadītājs: *Dr.sc.comp. Ģirts Karnītis* _____ .05.2011.

Recenzents: *M.dat. Kārlis Gobleja*

Darbs iesniegts 26.05.2011.

Kvalifikācijas darbu pārbaudījumu komisijas sekretārs: *Dainis Dosbergs* _____

Darbs aizstāvēts kvalifikācijas darbu pārbaudījuma komisijas sēdē

____.06.2011. prot. Nr. _____, vērtējums _____ (_____)

Komisijas sekretārs(-e): _____