

**ინტეგრირებული სერვის-ორიენტირებული სისტემის  
მონაცემთა ბაზის დაპროექტება და რეალიზაცია**

გია სურგულაძე, ნინო თოფურია, ირაკლი ბულია  
საქართველოს ტექნიკური უნივერსიტეტი

**რეზიუმე**

განიხილება კორპორაციათაშორისი ავტომატიზებული სისტემის მონაცემთა ბაზის ობიექტ-როლური მოდელირების ამოცანა და მისი გადაწყვეტა. შემოთავაზებულია სხვადასხვა საფინანსო სტრუქტურებს შორის განაწილებული, ლოგიკურად ერთიანი მონაცემთა ბაზის აგების კონცეფცია სერვის-ორიენტირებული არქიტექტურის საფუძველზე. შედეგები რეალიზებულია ვებ-აპლიკაციების სახით Ms Visual Studio.NET პლატფორმაზე Natural ORM Architect მოდელირების პროგრამული ინსტრუმენტისა და Ms SQL Server ბაზის გამოყენებით.

**საკვანძო სიტყვები:** კონცეპტუალური სქემა. ობიექტ-როლური მოდელირება. ORM. ERM. NORMA. SQL. DDL.

**1. შესავალი**

მართვის საინფორმაციო სისტემების დასაპროექტებლად აუცილებელია კონცეპტუალური მოდელის ანუ საპრობლემო სფეროს ER-მოდელის აგება, რომლის საფუძველზეც შემდგომ ხდება მონაცემთა განაწილებული რელაციური ბაზის სტრუქტურების დაპროექტება [1,2].

ინტერკორპორაციული სისტემის მაგალითისათვის განვიხილავთ ფინანსთა სამინისტროს შემოსავლების სამსახურისა და საფინანსო ბანკების ერთიან ელექტრონულ სისტემას, რომელც სერვის-ორიენტირებული არქიტექტურის (SOA) პრინციპით ფუნქციონირებს [3].

კონცეპტუალური მოდელირების განვითარებულ ტექნიკას წარმოადგენს ობიექტ-როლური მოდელირება (ORM), რომელიც პრაქტიკულად არის სემანტიკური მოდელირების ინსტრუმენტი ფაქტებზე დაყრდნობით [4]. ბუნებრივი ენის და ინტუიციური დიაგრამების (რომელთა შვესებაც ხდება მაგალითებით) გამოყენება და ასევე საპრობლემო სფეროს აღწერა ელემენტარული ფაქტების საფუძველზე, საგრძობლად ამარტივებს დაპროექტების პროცესს [2].

კონცეპტუალურის მოდელის ავტომატური დაპროექტებისათვის სტატიამში წარმოდგენილია ობიექტ-როლური მოდელირების მეთოდი (ORM). NORMA (Natural ORM Architect) არის ახალი მოდელირების ინსტრუმენტი ობიექტ-როლური დიაგრამების დასაპროექტებლად ORM2-ის ბაზაზე [2].

**2. ძირითადი ნაწილი**

თავდაპირველად ხდება საპრობლემო არის, კერძოდ კორპორაციათაშორისი არსებული სისტემის მოთხოვნილებათა ანალიზი და ტექნიკური დავალების განსაზღვრა. არსებული ბიზნეს-პროცესებისა და ბიზნეს-წესების საფუძველზე ჩამოყალიბდება ფაქტები, ანუ იმ კანონზომიერებათა ერთობლიობა, რომელიც უნდა აისახოს მონაცემთა ბაზის მოდელში. ამგვარად, მიღებული სემანტიკური ფაქტების სიმრავლის ფიზიკური გადატანით ავტომატიზებული დაპროექტების NORMA ინსტრუმენტით Ms Visual Studio.NET გარემოში, განისაზღვრება ჩვენი სისტემის ობიექტ-როლური მოდელი, რომლის მიხედვითაც შემდგომ აიგება ERM-დიაგრამა [2].

საფინანსო ბანკების და შემოსავლების სამსახურის კორპორაციის ურთიერთობების საპრობლემო სფერო აღიწერება, მაგალითად, შემდეგი სახის ინფორმაციით: „ბანკში კლიენტის რეგისტრაცია ხდება ოპერატორის მიერ, „კლიენტების მართვის“ პროგრამის საშუალებით. ახალ კლიენტს ენიჭება სტატუსი „მხოლოდ ბიუჯეტური“. ამ კლიენტის შესახებ შეტყობინება ეგზავნება შემოსავლების

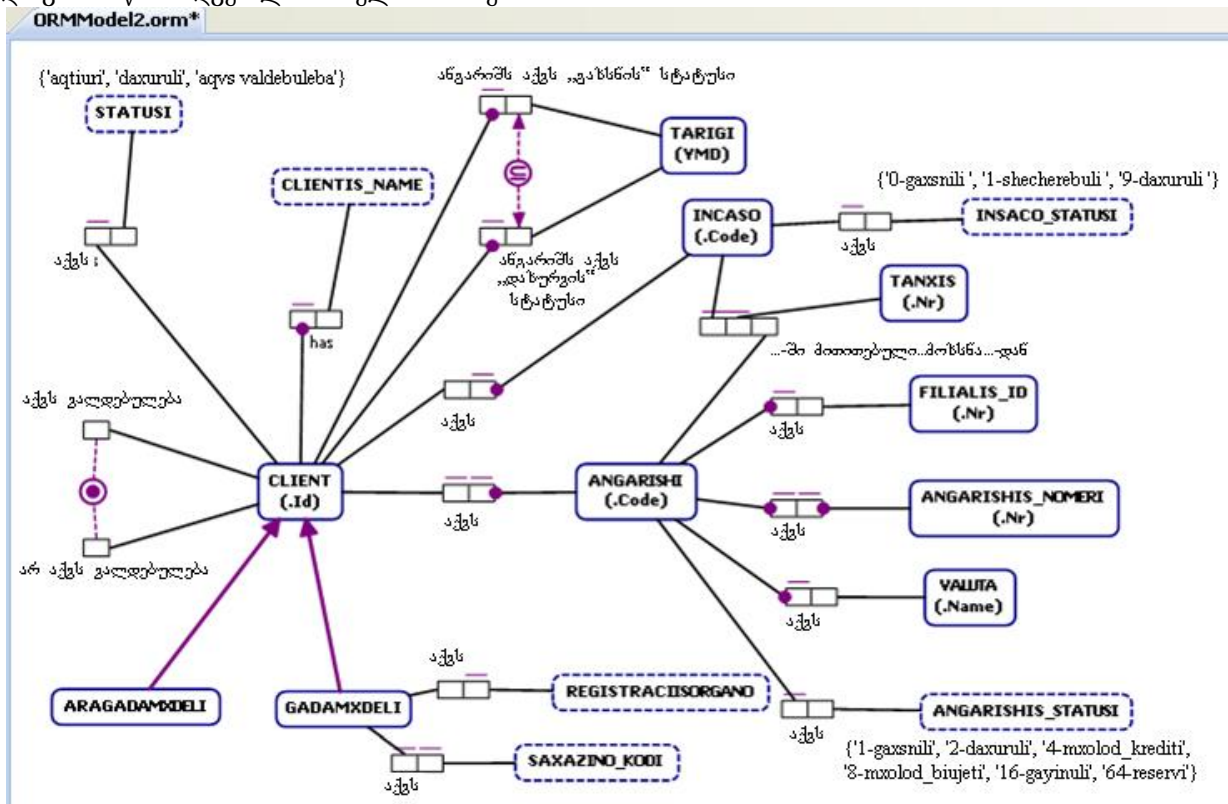
სამსახურს. შემოსავლების სამსახური ამოწმებს კლიენტის მონაცემებს. კერძოდ, განსაზღვრავს აქვს თუ არა ამ კლიენტს რაიმე ვალდებულება (დავალიანება ბიუჯეტის მიმართ ან სხვ.) და შემდეგ ბანკს უგზავნის შესაბამის საპასუხო შეტყობინებას. ამ შეტყობინების შესაბამისად ხდება კლიენტის სტატუსის შეცვლა „აქტიურზე“ (თუ არ აქვს ვალდებულება).

კლიენტის ანგარიშებზე ინკასოს (ან ყადაღას) წარმოდგენისთვის შემოსავლების სამსახური აფორმირებს შეტყობინებას, რომელიც შეიცავს კლიენტის მონაცემებს და თანხის ოდენობას. ეს შეტყობინება ეგზავნება ბანკის ვებ-სერვისს. ოპერატორი ინკასოების (ან ყადაღების) მართვის მოდულში ნახულობს შემოსული ინკასოების სიას და ამოწმებს მათ. დადასტურებულ ინკასოებზე Windows სერვისი კლიენტის ანგარიშებს ანიჭებს „მხოლოდ ბიუჯეტურ“ სტატუსს“ და ა.შ.

ელემენტარული ფაქტების ერთობლიობა შეიძლება ასე წარმოვადგინოთ:

f1- კლიენტს აქვს გვარი; f2 - კლიენტს აქვს ანგარიში ; f3-კლიენტს აქვს ტიპი; f4-ანგარიშს აქვს ანგარიშის ნომერი ; f5-ანგარიშს აქვს ვალუტა; f6-კლიენტს აქვს ინკასო; f7-ინკასოს აქვს ინკასოს სტატუსი; f8-ანგარიშებს აქვს ანგარიშების სტატუსი; f9-ანგარიშს აქვს ფილიალი; f10-კლიენტს აქვს სტატუსი; f11-ანგარიშიდან ხდება ინკასოში მითითებული თანხის მოხსნა და ა.შ.

ამ ფაქტების საფუძველზე NORMA პროგრამული ინსტრუმენტით აგებული შესაბამისი ORM-დიაგრამა წარმოდგენილია 1-ელ ნახაზზე.



ნახ.1. ORM-მოდელის ფრაგმენტი

ORM-დიაგრამაზე ობიექტები გამოსახულია ელიფსებში, წყვეტილი ელიფსი აღნიშნავს მნიშვნელობას (value), პრედიკატები გამოსახულია მართკუთხედებში. გამოყენებულია ბინარული და ტერნარული პრედიკატები.

შეზღუდვები გამოსახულია შემდეგი აღნიშვნებით:

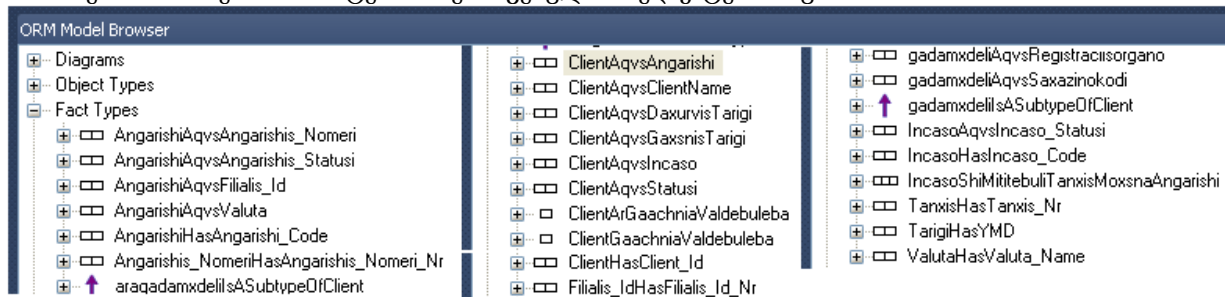
- იძულების შეზღუდვა: „თითოეულ კლიენტ აქვს გვარი“;
- შიგა უნიკალურობის შეზღუდვა ბინარული ფაქტის ტიპზე გვიჩვენებს, რომ „კლიენტის ანგარიში შეიძლება გახსნილი იყოს ბანკის მხოლოდ ერთ ფილიალში“.
- ↑ ქვეტიპის შეზღუდვა: კლიენტთან გამოყვავით ორი ქვეტიპი „გადამხდელი“ და „არაგადამხდელი“;
- ⊆ ქვესიმრავლის შეზღუდვა: თუ ბაზაში მითითებულია „ანგარიშის დახურვის თარიღი“, თავდაპირველად აუცილებლად უნდა იყოს მითითებული „ანგარიშის გახსნის თარიღი“;
- ⊙ წყვილის გამორიცხვის შეზღუდვა: „კლიენტს გააჩნია“ ან „არ გააჩნია ვალდებულება“; ორივე ფაქტის არსებობა ერთად გამორიცხულია.

ჩვენ მიერ პროგრამულ ინსტრუმენტში შეტანილ ფაქტებს ობიექტებისა და მათი სემანტიკური კავშირების შესახებ აქვს 1-ელ ცხრილში ნაჩვენები სახე.

ცხრ.1

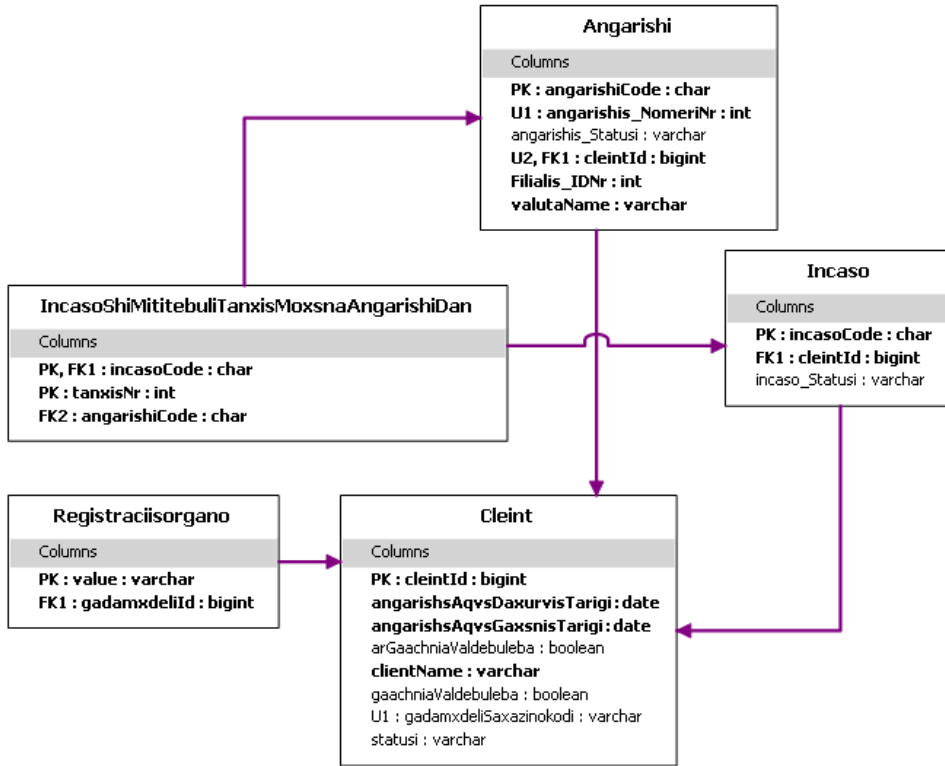
<p><b>Fact Types:</b>                  Client aqvs ClientName.  <b>Each</b> Client aqvs <b>exactly one</b> ClientName.  <b>It is possible that more than one</b> Client aqvs <b>the same</b> ClientName.                  Gadamxdeli <b>is an entity type</b>.  <b>Reference Scheme:</b> Client has Client_Id.  <b>Reference Mode:</b> .Id.</p>	<p><b>Fact Types:</b>                  Gadamxdeli aqvs Registraciisorgano.  <b>For each</b> Registraciisorgano, <b>at most one</b> gadamxdeli aqvs <b>that</b> Registraciisorgano.  <b>It is possible that the same</b> gadamxdeli aqvs <b>more than one</b> Registraciisorgano.                  Tarigi <b>is an entity type</b>.  <b>Reference Scheme:</b> Tarigi has YMD.  <b>Reference Mode:</b> YMD.</p>
<p><b>Fact Types:</b>                  Each gadamxdeli is an instance of Client.                  Gadamxdeli aqvs Saxazinokodi.                  Gadamxdeli aqvs Registraciisorgano.                  Aragamxdeli <b>is an entity type</b>.  <b>Reference Scheme:</b> Client has Client_Id.  <b>Reference Mode:</b> .Id.</p>	<p><b>Fact Types:</b>                  Gadamxdeli aqvs Saxazinokodi.  <b>Each</b> gadamxdeli aqvs <b>at most one</b> Saxazinokodi.  <b>For each</b> Saxazinokodi, <b>at most one</b> gadamxdeli aqvs <b>that</b> Saxazinokodi.                  Registraciisorgano <b>is a value type</b>.  <b>Portable data type:</b> Text: Variable Length.</p>
<p><b>Fact Types:</b>                  Each aragamxdeli is an instance of Client.                  Saxazinokodi <b>is a value type</b>.  <b>Portable data type:</b> Text: Variable Length.</p>	<p><b>Fact Types:</b>                  Filialis_Id has Filialis_Id_Nr.                  Angarishi aqvs Filialis_Id.                  Angarishi aqvs Filialis_Id.  <b>Each</b> Angarishi aqvs <b>exactly one</b> Filialis_Id.  <b>It is possible that more than one</b> Angarishi aqvs <b>the same</b> Filialis_Id.                  Angarishis_Nomeri <b>is an entity type</b>.  <b>Reference Scheme:</b> Angarishis_Nomeri has Angarishis_Nomeri_Nr.  <b>Reference Mode:</b> .Nr.</p>
და ა.შ.	

მე-2 ნახაზზე ჩანს სისტემის მიერ აგებული პრედიკატების ნუსხა.



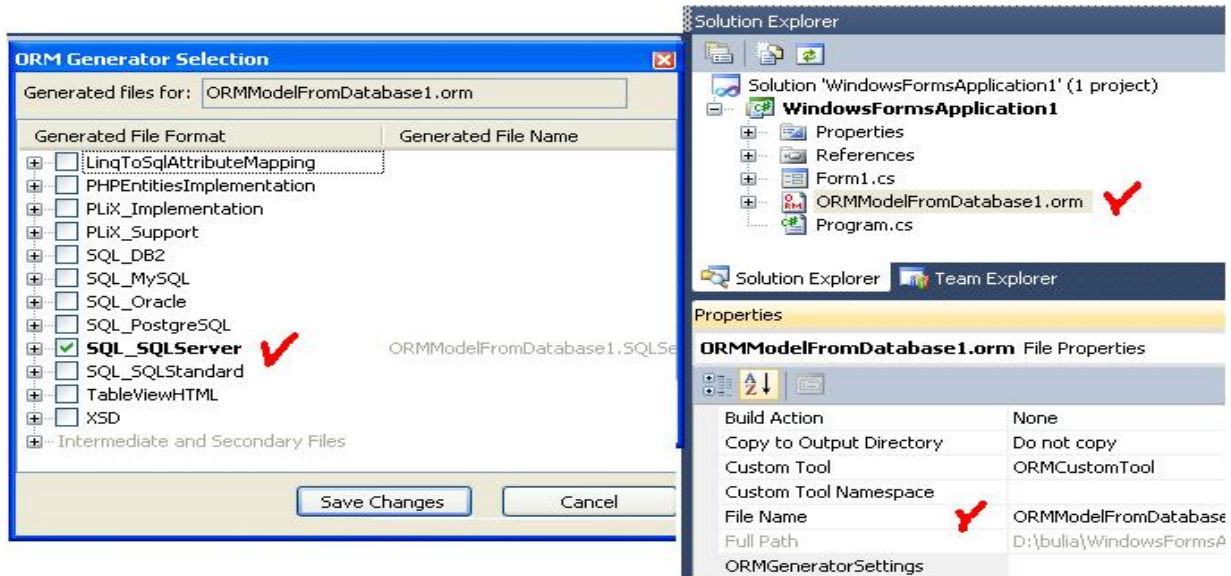
ნახ.2

MsVisual\_Studio.NET გარემოში NORMA პროგრამული პაკეტის საშუალებით ORM-დიაგრამიდან ავტომატურად ვღებულობთ ERM არსთა დამოკიდებულებათა მოდელს (ნახ.2).



ნახ.2 ER-დიაგრამის ფრაგმენტი

შემდეგ ავტომატურად მიიღება SQL Server მონაცემთა ბაზის ფიზიკური სტრუქტურა. მე-3 ნახაზზე ნაჩვენებია MsVisualStudio.Net გარემოში



ნახ.3. DDL-ფაილის გენერაცია

სისტემის მიერ ავტომატურად გენერირდება DDL-ფაილი, რომლის ფრაგმენტი მოცემულია ქვემოთ:

— ERM დაფარვების შესაბამისი DDL ფაილის ტექსტი -----

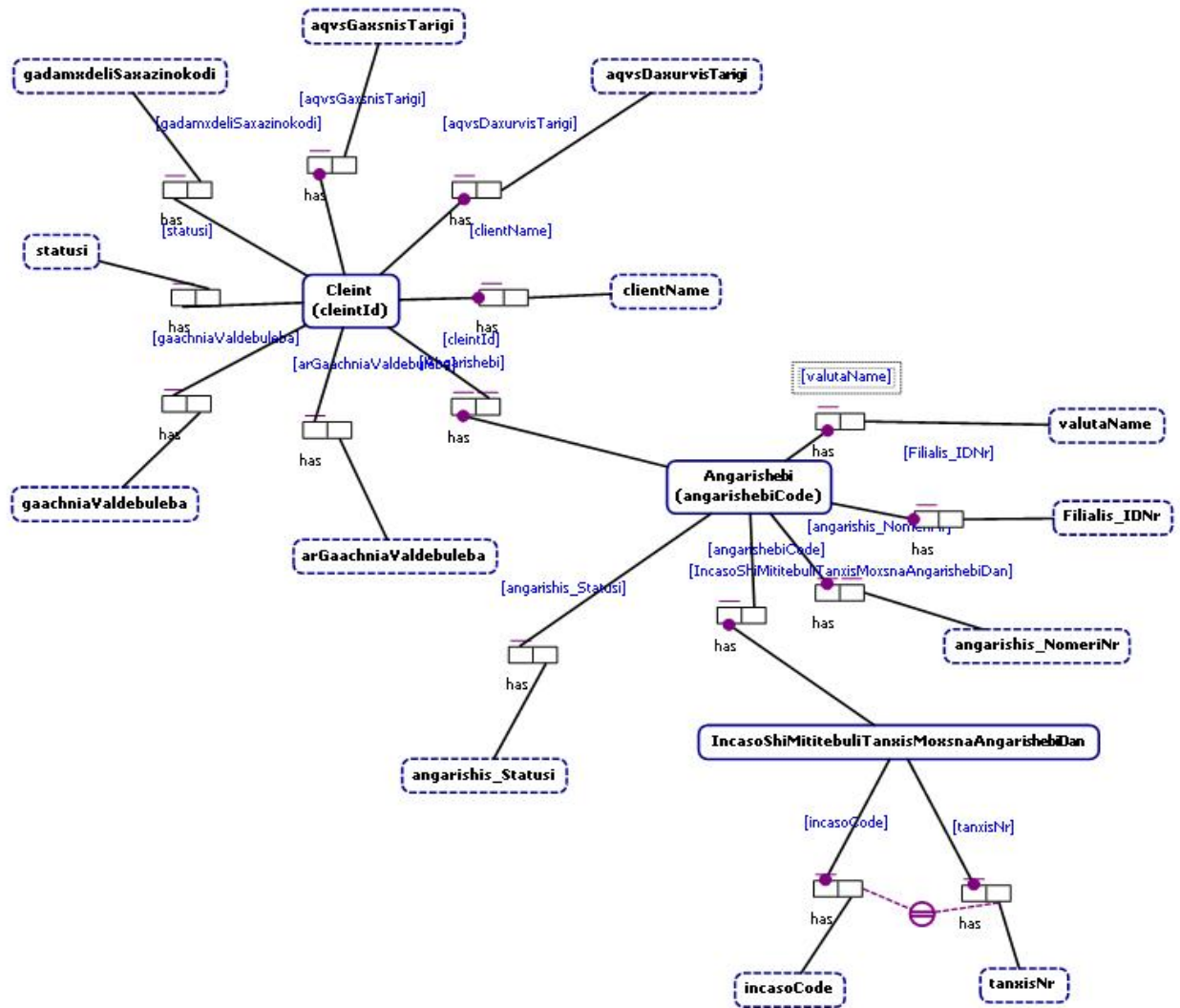
```

CREATE SCHEMA ORMModel1
GO
CREATE TABLE ORMModel1.Cleint
(
    cleintId BIGINT CHECK (cleintId IN (N'aqtiuri', N'daxuruli', N'aqvs valdebuleba'))
                                                    NOT NULL,
    clientName NATIONAL CHARACTER VARYING(MAX) NOT NULL,
    aqvsDaxurvisTarigi DATETIME NOT NULL,
    aqvsGaxsnisTarigi DATETIME NOT NULL,
    gadamxdeliSaxazinokodi NATIONAL CHARACTER VARYING(MAX),
    statusi NATIONAL CHARACTER VARYING(MAX) CHECK (statusi IN (N'aqtiuri', N'daxuruli',
N'aqvs valdebuleba')),
    gaachniaValdebuleba BIT,
    arGaachniaValdebuleba BIT,
    CONSTRAINT Cleint_PK PRIMARY KEY(cleintId)
)
GO
CREATE VIEW ORMModel1.Cleint_UC (gadamxdeliSaxazinokodi)
WITH SCHEMABINDING
AS
    SELECT gadamxdeliSaxazinokodi
    FROM ORMModel1.Cleint
    WHERE gadamxdeliSaxazinokodi IS NOT NULL
GO

CREATE UNIQUE CLUSTERED INDEX Cleint_UCIndex ON
ORMModel1.Cleint_UC(gadamxdeliSaxazinokodi)
GO
CREATE TABLE ORMModel1.Angarishebi
(
    angarishebiCode NATIONAL CHARACTER(4000) NOT NULL,
    angarishebiNomeriNr INTEGER NOT NULL,
    cleintId BIGINT CHECK (cleintId IN (N'aqtiuri', N'daxuruli', N'aqvs valdebuleba'))
                                                    NOT NULL,
    Filialis_IDNr INTEGER NOT NULL,
    valutaName NATIONAL CHARACTER VARYING(MAX) NOT NULL,
    angarishebiStatusi NATIONAL CHARACTER VARYING(MAX) CHECK (angarishebiStatusi IN
(N'1-gaxsnili', N'2-daxuruli', N'4-mxolod krediti', N'8-mxolod biujeti', N'16-
gayinuli', N'64-reservi')),
    CONSTRAINT Angarishebi_PK PRIMARY KEY(angarishebiCode),
    CONSTRAINT Angarishebi_UC1 UNIQUE(angarishebiNomeriNr),
    CONSTRAINT Angarishebi_UC2 UNIQUE(cleintId)
)
GO
CREATE TABLE ORMModel1.Incaso
(
    incasoCode NATIONAL CHARACTER(4000) NOT NULL,
    cleintId BIGINT CHECK (cleintId IN (N'aqtiuri', N'daxuruli', N'aqvs valdebuleba'))
                                                    NOT NULL,
    incaso_Statusi NATIONAL CHARACTER VARYING(MAX) CHECK (incaso_Statusi IN (N'0-
gaxsnili', N'2-shecherebuli', N'9-daxuruli')),
    CONSTRAINT Incaso_PK PRIMARY KEY(incasoCode)
)
GO
ALTER TABLE ORMModel1.Angarishebi ADD CONSTRAINT Angarishebi_FK FOREIGN KEY (cleintId)
REFERENCES ORMModel1.Cleint (cleintId) ON DELETE NO ACTION ON UPDATE NO ACTION
GO
ALTER TABLE ORMModel1.Incaso ADD CONSTRAINT Incaso_FK FOREIGN KEY (cleintId) REFERENCES
ORMModel1.Cleint (cleintId) ON DELETE NO ACTION ON UPDATE NO ACTION
GO
. . .

```

ORM2 ნოტაცია MsVisualStudio.Net პაკეტის საშუალებით ითვალისწინებს რევერსული დაპროექტების პროცესსაც. შესაძლებელია ERM მოდელიდან აღვადგინოთ ORM მოდელი და გამოვიკვლიოთ მათი სინტაქსურ-სემანტიკური სისრულე. შემდეგ შევადაროთ მიღებული და ადრე დაპროექტებული დიაგრამები ერთმანეთს. ექსპერტული შეფასების საფუძველზე შესაძლებელი იქნება ცდომილებების მიგნება და შემდეგ მათი კორექტირება. რევერსული დაპროექტების შედეგი, ანუ რეალურად მომუშავე SQL Server ბაზის DDL-ფაილიდან მიღებული ORM დიაგრამა გამოსახულია მე-4 ნახაზზე.



ნახ.4. რევერსული გზით მიღებული ORM – დიაგრამის ფრაგმენტი

მიღებული დიაგრამის ანალიზმა ცხადყო, რომ საჭიროა ORM-დიაგრამაში გარკვეული ცვლილებების შეტანა. კერძოდ, აუცილებელია ტერნარული პრედიკატის შეცვლა. ჩასამატებელია ტოლობის შეზღუდა, რომელიც გვიჩვენებს, რომ „ინკასოში“ აუცილებელია მითითებული იყოს თანხის რაოდენობა და ა.შ.

### 3. დასკვნა

თანამედროვე ინფორმაციული ტექნოლოგიების, კერძოდ CASE-ინსტრუმენტების გამოყენებით კორპორაციული მენეჯმენტისა და ინტერკორპორაციული ვებ-აპლიკაციების აგების პროცესში მნიშვნელოვნად უმჯობესდება პროგრამული უზრუნველყოფის ხარისხი და საგრძნობლად მცირდება დაპროექტების, მისი იმპლემენტაციისა და რეინჟინერინგის პერიოდები. ნაშრომში შემოთავაზებული კონცეფციის საფუძველზე შესაძლებელია დიდი საინფორმაციო სისტემების მონაცემთა ბაზების სტრუქტურების დაპროექტების პროცესის ავტომატიზება და აგება, აგრეთვე მისი შემდგომი რესტრუქტურისაციის პრობლემის მოქნილად გადაწყვეტა.

#### ლიტერატურა:

1. ჩოგვაძე გ., სურგულაძე გ., შონია ო. მონაცემთა და ცოდნის ბაზების აგების საფუძველები. სტუ. თბ., 1996
2. ვედეკინდი ჰ., სურგულაძე გ., თოფურია ნ. განაწილებული ოფის-სისტემების მონაცემთა ბაზების დაპროექტება და რეალიზაცია UML-ტექნოლოგიით. სტუ. თბ., 2006
3. ბულია ი. თანამედროვე სისტემებში ინტეგრაციის, მონაცემთა გადაცემის და დამუშავების ტექნოლოგიები. სტუ. შრ.კრ. „მას“-N2(11), თბ., 2011, გვ. 139–144
4. Halpin T., ORM 2 Graphical Notation, Neumont University, 2005. [http://www.orm.net/pdf/ORM2\\_TechReport1.pdf](http://www.orm.net/pdf/ORM2_TechReport1.pdf)

### DESIGNING AND IMPLEMENTATION OF DATABASES FOR INTERCORPORATE SERVICE-ORIENTED SYSTEMS

Topuria N.S., Surguladze G.G., Bulia I.  
Georgian Technical University

#### Summary

The article deals with the object-role modeling problem for inter-corporate automated systems and its implementation is considered in the article. The creation concept for the distributed databases for various financial structures on the basis of Service-oriented architecture is offered. The web-application forms on Ms Visual Studio.NET platform by The NORMA software and Ms SQL Server package are realized as a result.

### ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ БАЗ ДАННЫХ ИНТЕРКОРПОРАТИВНОЙ СЕРВИС-ОРИЕНТИРОВАННОЙ СИСТЕМЫ

Сургуладзе Г.Г., Топурия Н.Ш., Булия И.  
Грузинский Технический Университет

#### Резюме

Рассматривается задача объектно-ролевого моделирования баз данных для интеркорпоративной автоматизированной системы и ее реализация. Предлагается концепция построения распределенных баз данных для различных финансовых структур на базе сервис-ориентированной архитектуры. Результаты реализованы в виде веб-аппликаций на платформе Ms Visual Studio.NET с помощью программного инструмента Natural ORM Architect и пакета Ms SQL Server.