

საქართველოს ტექნიკური უნივერსიტეტი

ბია სურგულაძე, ნინო თოფურია,
გიორგი მელაძე

მონაცემთა ბაზის მართვის სისტემების
სტრუქტურების დაპროექტება:
ORM / ERM, Ms SQL Server



თბილისი – 2007

უაკ 681.3.06

დაპროგრამების ობიექტ-ორიენტირებული მიდგომისა და *UML* ტექნოლოგიის საფუძველზე შემოთავაზებულია ობიექტ-როლური მოდელირების *ORM*-ინსტრუმენტის გამოყენება არსთა-დამოკიდებულების მოდელის *ERM*-სქემის ავტომატიზებულ რეჟიმში მისაღებად. კატეგორიალური მიდგომისა და ლოგიკურ-აღგებრული მეთოდების გამოყენებით ხორციელდება საპრობლემო სფეროს მონაცემთა სტრუქტურებისა და ცოდნის ობიექტ-ორიენტირებული მოდელირება.

შემუშავებულია ამოცანათა კომპლექსი მონაცემთა ბაზების სტრუქტურების დასაპროექტებლად მრეწველობის, განათლების, ეკონომიკის და სხვ. სფეროებიდან. მათი გამოყენება მიზანშეწონილია მონაცემთა ბაზების დაპროექტებისა და დაპროგრამების *UML*-ტექნოლოგიის დისციპლინების შესწავლის მიზნით, აგრეთვე კვლევითი სამუშაოების ჩასატარებლად მონაცემთა ბაზების რაციონალური სტრუქტურებს მისაღებად.

განკუთვნილია ინფორმატიკისა და მართვის სისტემების ფაკულტეტის სპეციალობის მაგისტრანტებისა და სტუდენტებისათვის.

ს ა რ ჩ ე მ ი

- შესავალი	4
I თავი. თეორიული ნაწილი	5
1.1. რელაციური ბაზების სემანტიკური პრობლემები და მათი გაფართოებული მოდელები	5
1.2. კატეგორიული მიდგომა მონაცემთა ბაზებში და ობიექტ-ორიენტირებული ER-სქემის აგება	7
1.3. მეთოდოლოგია	13
1.3.1. მოთხოვნილებათა არაფორმალური წარმოდგენა (ტექნიკური დავალება)	13
1.3.2. პირველი ობიექტები მთავარ პრედიკატებში	14
1.3.3. აბსტრაქცია – ენის ლოგიკური პროცესი	15
1.3.4. მთავარიდან დამხმარე პრედიკატებისკენ	16
1.3.5. აბსტრაქცია – განზოგადება	19
1.3.6. მერეოლოგიური სტრუქტურები	20
II თავი. პრაქტიკული ნაწილი (ამოცანები)	23
2.1. ობიექტ-როლური მოდელირება (ORM) განაწილებული ოფის-ობიექტებისათვის	23
2.2. ამოცანის დასმა	27
2.3. ფაქტების მოზადება და ORM-დიაგრამის აგება საკონტრაქტო ამოცანისთვის	30
2.4. ORM-დიაგრამის აგება უნივერსიტეტის მაგალითზე	32
2.5. Ms SQL - ბაზისთვის DDL ფაილის მომზადება	37
2.6. ავტომატიზებული ანალიზის სისტემის დაპროექტება სატენდერო კომისიის ექსპერტებისათვის	43
2.7. ენერგოსისტემის ობიექტ-როლური მოდელი	49
2.8. ენერგობიექტების კონტროლის სისტემის მონაცემთა ბაზის ავტომატიზებული აგება	51
- ლიტერატურა	55

შესავალი

კომპიუტერული და საინფორმაციო ტექნოლოგიების განვითარების თანამედროვე ეტაპზე განსაკუთრებით აქტუალურია სამეცნიერო და საინჟინრო სამუშაოების შესრულება განაწილებული მართვის ავტომატიზებული სისტემების დასაპროექტებლად და სარეალიზაციოდ ობიექტ-ორიენტირებული მიდგომის საფუძველზე.

უნიფიცირებული მოდელირების ენის (UML) ინსტრუმენტების გამოყენებით მიიღწევა დაპროექტებისა და დაპროგრამების ურთულეს პროცესთა ავტომატიზაცია, რაც საგრძნობლად ამცირებს საპროექტო-საინჟინრო ეტაპების შესრულების დროს და ამადლებს საბოლოო პროდუქტის, მართვის სისტემის ხარისხს [1,2].

საუნივერსიტეტო განათლების სისტემა კომპლექსური და მეტად მნიშვნელოვანი ობიექტია მართვის პროცესების შემდგომი სრულყოფის თვალსაზრისით ახალი საინფორმაციო ტექნოლოგიების ბაზაზე [3,6,9,10,15].

ჩვენი ქვეყნის მრავალი დარგი, კერძოდ ენერგეტიკა, მრეწველობა, სოფლის მეურნეობა, ჯანმრთელობის დაცვა და ა.შ. ინფორმატიკისა და მართვის ამოცანების გადასაჭრელად ფართოდ იყენებს რელაციური ბაზების მართვის სისტემებს [7]. მათი შემდგომი სრულყოფა მათემატიკური მოდელირების სემანტიკური პრობლემების აღმოსაფხვრელად და ბაზების სტრუქტურების ავტომატიზებულ რეჟიმში დასაპროექტებლად – მეტად მნიშვნელოვანი საკითხებია.

წინამდებარე ნაშრომში შემოთავაზებულია აღნიშნული პრობლემატიკის გადაწყვეტის თანამედროვე თეორიული კონცეფცია და მისი პრაქტიკული რეალიზაცია ობიექტ-როლური მოდელის, არსთა-დამოკიდებულების სქემების, UML-ტექნოლოგიის ინსტრუმენტების (*MsVisio, Software, Database-ORM/ERM*) გამოყენებით [4,5,9].

წინის 1-ელი თავში გადმოცემულია დასმული პრობლემატიკის არსი და მისი გადაწყვეტის თეორიული ასპექტები და მეთოდები. შემოთავაზებულია ამ მიმართულების ფუძემდებლების, ამერიკელი, გერმანელი და სხვა ქვეყნების მეცნიერთა შედეგები.

მე-2 თავში აღიწერება კონკრეტული ამოცანები მონაცემთა რელაციური ბაზების მართვის სისტემების სტრუქტურების აგების პროცესების ავტომატიზაციისათვის. მათი გამოყენება მიზანშეწონილია მაგისტრანტებისა და მაღალი კურსის სტუდენტთა ჯგუფებთან პრაქტიკული და ლაბორატორიული სამუშაოების ჩასატარებლად.

I ტაში თეორიული ნაწილი

1.1. რელაციური ბაზების სემანტიკური პრობლემები და მათი გაფართოებული მოდელები

მონაცემთა ბაზა დისკრეტული დინამიკური სისტემაა, რომლის მდგომარეობები (რელაციები) მიეკუთვნება ფიქსირებულ მონაცემთა მოდელს. მონაცემთა მოდელი შეიძლება განვმარტოთ, როგორც ალგებრული სისტემა, რომელიც შედგება ლექსიკონური სიმრავლეების (საგნობრივი სფეროს ამსახველი სიტყვების ერთობლიობა) და მათ შორის დამოკიდებულებებისაგან, რომლებზეც შესაძლებელია განხორციელდეს სხვადასხვა სახის ოპერაცია.

მონაცემთა რელაციური მოდელი ფორმალური მათემატიკური ობიექტია და მისი საშუალებით საგნობრივი სფეროს არაფორმალურად აღწერილი თვისებების ასახვას მივყავართ რთულ სემანტიკურ პრობლემამდე. იგულისხმება სემანტიკის (შინაარსის) მათემატიკური მოდელირების სპეციფიკური პრობლემები. მათი სირთულე ძირითადად განისაზღვრება მონაცემთა ბაზის რეორგანიზაციის (განახლების) პროცედურების სირთულით, სტატიკური და დინამიკური შეზღუდვების (პრედიკატების) სისწორის შემოწმებით.

შეზღუდვები, გამოხატავს მონაცემთა ბაზის მდგომარეობის (რელაციები) ზოგად, აბსტრაქტულ თვისებებს ანუ მონაცემთა ბაზის სემანტიკას. შეზღუდვების გამოხატვის ყველაზე ბუნებრივი ხერხია გამონათქვამები I-რიგის პრედიკატების ენაზე, რომელიც განსაზღვრავს მონაცემთა ბაზის დასაშვებ მდგომარეობათა სიმრავლეს.

ამ გამონათქვამების ერთობლიობა მოდელიდან ამოსარჩევი ელემენტის სახელებთან ერთად (ატრიბუტების დასახელება) ქმნის მონაცემთა ბაზის ე.წ. სტატიკურ სქემას.

მონაცემთა ბაზა, გადადის რა ერთი მდგომარეობიდან მეორეში (რელაციური ცვალეზადობა), აღწერს მონაცემთა მოდელში გარკვეულ ტრანსფორმაციას. ყოველი მომდევნო მდგომარეობა შეიძლება დამოკიდებული იყოს მის წინა მდგომარეობაზე. ამ მდგომარეობებს შორის კავშირები აღიწერება მთლიანობის დინამიკური შეზღუდვებით (პრედიკატებით), რომლებიც ქმნის მონაცემთა ბაზის ე.წ. დინამიკურ სქემას.

მთლიანობის პრედიკატების ცოდნის საფუძველზე, რომელიც ერთ-ერთ მნიშვნელოვან ნაწილს ფუნქციონალური დამოკიდებულებები შეადგენს, ხდება სინამდვილის (საგნობრივი სფეროს) მოდელირება ნორმალურ ფორმათა თეორიის საშუალებით. ფორმირდება მონაცემთა ბაზის სქემა, რომელიც ამ დამოკიდებულებათა სასრულ ერთობლიობას შეადგენს. მაგრამ ფუნქციონალური დამოკიდებულებები ასახავს სინამდვილის შინაარსობრივი მხარის მხოლოდ ნაწილს; მონაცემთა ბაზის მოდიფიკაცია კი სემანტიკური ბუნების მატარებელია. აქედან გამომდინარე, არაა ცხადად წარმოდგენილი, თუ როგორ შეიძლება დამოკიდებულებათა ნორმალიზაციის სინტაქსური პროცედურებით გადაწყდეს სემანტიკური პრობლემები?!

საგნობრივი არის სტრუქტურისა და კანონზომიერებების მათემატიკური მოდელირებისას რელაციურ მოდელში გამოიყენება სიმრავლეთა თეორიის ელემენტები. ამიტომ ისმის კითხვა, რამდენად შეესაბამება მონაცემთა ბაზაში ფორმალურად ასახული ცნებები სიმრავლეთა თეორიის ცნებებს (ტერმინოლოგიის მათემატიკური სიზუსტის თვალსაზრისით).

რელაციური ბაზა გამოიხატება დამოკიდებულებათა ქვესქემების ერთობლიობის საშუალებით, ე.ი. გვაქვს რელაციების (ცხრილების) ერთობლიობა, რომლებშიც ასახულია ობიექტი (Ri), მათი თვისებები (Aj), კონკრეტული ურთიერთკავშირებით (Zk). კონკრეტული საგნობრივი სფეროდან გამომდინარე, რელაციური მოდელის აგების დროს ობიექტები, თვისებები და მნიშვნელობები შინაარსობრივად ურთიერთშენაცვლებადია, ე.ი. არ არსებობს მოდელირების საბოლოო ცალსახა ალგორთმი ამ საკითხის გადასაწყვეტად. ამასთანავე, რელაციური ბაზის მოდელში არაა ასახული ობიექტებს შორის კავშირების სემანტიკა. ხშირად კი, შინაარსობრივად ერთი და იგივე თვისება შეიძლება წარმოდგენილი იქნეს სინტაქსურად განსხვავებული ფორმებით (სინონიმები). სისტემას კი სჭირდება დამატებითი ინფორმაციული საშუალებები, რათა ეს მომენტები გაითვალისწინოს.

ჩვენ შემოვიფარგლეთ ზოგიერთი ძირითადი პრობლემური საკითხების ანალიზით, რომლებიც დამახასიათებელია რელაციული მოდელისათვის. შეიძლება აღინიშნოს, რომ ასეთი პრობლემების გადაჭრა შესაძლებელია როგორც რელაციური მოდელის შედარებით გართულებით (დამატებითი ინფორმაციის შემოტანით), ასევე პროგრამული და აპრატული საშუალებების სრულყოფით მომავალში.

რელაციური მოდელების სემანტიკური ანალიზი ფუნდამენტურად ჩაატარა ე.კოდმა და თავის ნაშრომში გადმოსცა ის ძირითადი მოთხოვნები, რომლებიც დამატებული უნდა იყოს რელაციურ ბაზაში. მისი აზრით სისტემა უნდა გაფართოვდეს ახალი ცოდნის დედუქციური გამოყვანის საშუალებებით არსებული ფაქტების საფუძველზე. იგი მნიშვნელოვნად თვლის აგრეთვე მონაცემთა ბაზის დამოკიდებულებებისა და მათი ძირითადი სტრუქტურების ეკრანზე გამოტანის გრაფიკულ საშუალებათა არსებობას, რაც ადამიანისთვის ფსიქოლოგიურ კომფორტს ქმნის. დასასრულად, მან აღნიშნა, რომ რელაციურ მოდელთა შემდგომი განვითარება უნდა მოიცავდეს დროის ასახვის საკითხსაც.

ახლა განვიხილოთ ზოგიერთი ე.წ. „სემანტიკურად გაფართოებული“ რელაციური მოდელი, რომელიც რეალიზებულია ზემოაღნიშნული მოსახრების საფუძველზე.

რელაციური მოდელის სემანტიკური გაფართოების საშუალებით კორტეჟი ხდება შედარებით ინფორმატიული, ვინაიდან იგი ჩვეულებრივ სალაპარაკო ენის სტრუქტურას მოიცავს. ე.ი. ამ შემთხვევაში წინადადების თითოეულ სიტყვას (არსებით სახელს, ატრიბუტს და მის მნიშვნელობას) აქვს თავისი როლი: სუბიექტი, ობიექტი, დამატება და ა. შ., გამოკვეთილია აგრეთვე მოქმედება (შემასმენელი), რომელიც ერთგვარად ხსნის დამოკიდებულების ატრიბუტთა ურთიერთმიმართებას.

1.2. კატეგორიული მიდგომა მონაცემთა ბაზებში და ობიექტ-ორიენტირებული ER-სქემის აგება

1.2.1. ძირითადი ცნებები

კატეგორიები, რომლებსაც აქ ვიხილავთ, არის ლოგიკური გრამატიკის შემადგენელი ნაწილი, რომელიც, როგორც ზოგადი ენის საძირკველი, ბევრად მდიდარია, ვიდრე ტრადიციული პრედიკატების ლოგიკა და რომელიც ცხადია დამოუკიდებელი უნდა იყოს ბუნებრივი ენის ემპირიული გრამატიკისგან.

პრედიკატების ლოგიკა არის ლოგიკური გრამატიკა დანართების შინაარსის ასაგებად, მაგრამ ძალზე ღარიბია და ნაკლებად განვითარებული - იგი არ ფლობს სინტაქსურ კატეგორიებს, რათა განასხვავოს ერთმანეთისგან მთავარი და არამთავარი პრედიკატები.

სალაპარაკო ენის წინადადება „ფილო არის ყავისფერი ძაღლი“ პრედიკატების ენაზე ჩაიწერება ასე:

ფილო ე ძალლი ^ ფილო ე ყავისფერი

ე=აღნიშვნა პრედიკატების ლოგიკაშიც და ზოგად ლოგიკურ ენაშიც არის კოპულა (კავშირი), ანუ დამხმარე საშუალება ენაში ასახვისათვის, რათა პრედიკატი (შემასმენელი) შეუსაბამოს არსს (საგანს, ობიექტს).

პრედიკატიზაცია არის ასიმეტრიული. კოპულას (ე) მარცხნივ ზის საკუთარი სახელი (ფილო) ან მახასიათებელი (აღნიშვნა).

მარჯვნივ ზის პრედიკატორი (არსის დასახელება – განსხვავებული გამოსახულება).

საკუთარი სახელები და მახასიათებლები არის ნომინატორები. ამგვარად, ნომინატორები, კოპულა და პრედიკატორები არის სამი კატეგორია, რომლებიც დასაწყისში შემოვიტანეთ.

Categorien(Nominatoren, Prädikatoren, Kopula)

განსხვავება მთავარ და დამხმარე პრედიკატორებს შორის პრედიკატულ ლოგიკაში წაშლილია, ვინაიდან პრედიკატულ-ლოგიკური გამოსახულება შეიძლება გულისხმობდეს წინადადებას „ფილო არის ძალღური ყავისფერი“

„ფილო ე (ყავისფერი) (ძალღი)“, სადაც ყავისფერი დამხმარე და ძალღი მთავარი პრედიკატორია.

ელემენტარულ წინადადებაში (მარტივ გამონათქვამში) შეიძლება იყოს ერთი მთავარი და რამდენიმე დამხმარე პრედიკატორი, აქ ლოგიკური კავშირი არაა.

სინტაქსური კატეგორიები, არსებისა (ობიექტების) და მოქმედებებისგან (ოპერაციების) განსხვავებით პრედიკატულ ლოგიკას არ გააჩნია.

გამონათქვამიდან: „ფილო დარბის“ გვექნება „ფილო ე მორბენალი“, ან დამხმარე პრედიკატორით: „ფილო დარბის სწრაფად“ ვღებულობთ:

„ფილო ე სწრაფი ^ ფილო ე მორბენალი“.

მნიშვნელოვანია, რომ აქტიური კომპონენტები განვასხვავოთ სუფთა ობიექტებისაგან (არსებისგან).

გამოსახულებაში: „ფილო π სირბილი“ ან „ფილო π (სწრაფი)(სირბილი)“ მოქმედების კოპულით π (კეთება-tun) ნაჩვენებია, რომ მთავარი პრედიკატორის სახით გამოდის მოქმედების პრედიკატორი და არა ობიექტის პრედიკატორი.

ლოგიკურ გრამატიკაში, რომელსაც ჩვენ განვიხილავთ, არსებობს π-კოპულას შემდეგ ასევე მხოლოდ ერთი მოქმედების

პრედიკატორი, ვინაიდან რამდენიმე მოქმედება ერთდროულად ელემენტარულად ვერ შესრულდება.

რთული წინადადება, მაგალითად, რამდენიმე მოქმედების პრედიკატორით π -ს შემდეგ, მიიღება მარტივი წინადადებების ლოგიკური ნაწილაკით შეერთებით.

პრედიკატული ლოგიკა უნდა განვიხილოთ, როგორც ლოგიკური გრამატიკის გაფართოება.

კითხვა, თუ როგორ ამუშავებენ ელემენტარულ წინადადებებში რამდენიმე პრედიკატორ-არსს ერთი ε ან π კოპულას საშუალებით, შემდგომი მსჯელობის საგანია.

ლოგიკურ გრამატიკაში ჩვენ ვიყენებთ ლორენცის [21] მოსაზრებას.

მაგალითად, წინადადება: „მიულერი იღებს დეტალს საწყობიდან“.

დეტალი - პირდაპირი ობიექტია, რომელზეც მიუთითებს მოქმედების პრედიკატორი „იღებს“. პირდაპირ ობიექტს შეიძლება მოჰყვეს ირიბი ობიექტი, მაგალითად, „საწყობი“, რომელიც ადგილის გარემოებაა. იგი აღნიშვნა ანუ მახასიათებელია, რომლისთვისაც შეიძლება ჩაისვას საკუთარი სახელი (მაგალითად, ავრორა). დაიწერება „ავრორა = i საწყობი“, სადაც = არის განსაზღვრების ნიშანი, ხოლო Jota (i) არის მახასიათებლის ნიშანი.

ელემენტარული წინადადება ჩაიწერება ასე:

მიულერი	π	(აღება)	(დეტალი)	(i საწყობი)
[ნომინა- ტორი]		[მოქმედების პრედიკატორი]	[ობიექტის პრედიკატორი]	[მახასია- თებელი]

აქმდე განხილულ კატეგორიებს:

- ნომინატორი – პრედიკატორი;
- მთავარი პრედიკატორი – დამხმარე პრედიკატორი;
- ობიექტის პრედიკატორი – მოქმედების პრედიკატორი;
- არის კოპულა (ε) – კეთების კოპულა (π)

ლოგიკურ გრამატიკაში ემატება შემდეგი მნიშვნელოვანი კატეგორიები, რათა შესაძლებელი იყოს კონცეპტუალური სქემის მეთოდურად აგება [22].

ასეთ კატეგორიებს მიეკუთვნება უპირველეს ყოვლისა მსჯელობა ლოგიკური ენის დონეებზე, ანუ დიფერენცირება

კონკრეტულ და აბსტრაქტულ დონეებზე და მათ ობიექტებზე (ტიპების ლოგიკა).

შემოიტანება აგრეთვე კატეგორია „მთელი-ნაწილი“;

სქემების აგებისას დრო (ან დროის პუნქტი) არის ჩვეულებრივი პრედიკატორი, რომელიც „მანამ“, „შემდეგ“ და ა.შ. მიეთითება ორადგილიან პრედიკატორში „x, y ε before“ ან „x, y ε after“.

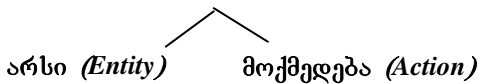
ამგვარად, შემოიტანება კიდევ შემდეგი კატეგორიები:

- აბსტარქცია – კონკრეტიზაციის: (ენის სხვადასხვა დონეზე მისაღწევად);
- კომპოზიცია – დეკომპოზიციის: „მთელი - ნაწილი“;
- ობიექტების ენა – მეტაენა: ობიექტების ენა ერთი ენაა (I დონე, მაგალითად, ინგლისური) რომლის შესახებაც საუბრობენ მეტაენაზე (II დონე, მაგალითად, ქართული);
- სქემის დრო - მიმართვის (ათვის) დრო.

სქემის დროსთვის იყენებენ ტერმინს „ინტენსიონალი“ (შინაარსი) და მიმართვის დროსთვის „ექსტენსიონალი“ (მოცულობა).

„ობიექტი“ არის ყველაზე მეტად განზოგადებული პრედიკატორი და არ გამოდგება განსხვავებულობის შესადარებლად. ესაა ერთი მთავარი სახელის ქვეშ მოქცეული მნიშვნელობები. მიღებულია ობიექტის სინონიმად „არსი“-ს გამოყენება (thing – entity). ამგვარად, ობიექტი ხასიათდება არსით და მოქმედებით (ქცევით – action).

Object



ნახ.1.1.

წინადადება: „მოქმედებები არის ობიექტები“ არაბუნებრივია, ვინაიდან ხშირად არსი აღიქმება ობიექტად და არა მოქმედება.

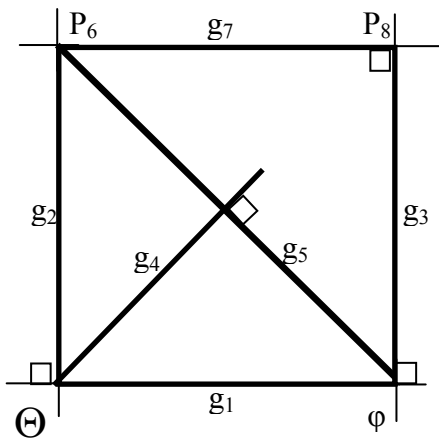
პრედიკატორ „ობიექტის“ უნივერსალობა დგება ეჭვის ქვეშ. თუ „ობიექტის“ მაგივრად გამოიყენება „არსი“, მაშინ გამონათქვამი „Actions are entities“ – ნორმალურად აღიქმება (ინგლისელ-ამერიკელების მიერ). რჩება შეკითხვა: ტერმინი „ობიექტორიენტულობა“ ნიშნავს მხოლოდ „არს-ორიენტულობას“, თუ ასევე „მოქმედება-ორიენტულობასაც“ ?.

დღეისათვის „მოქმედების“ (action) ნაცვლად იყენებენ ტერმინს მოვლენა (event). განმეორებად მოვლენათა ერთობლიობას უწოდებენ პროცესს (process).

ტერმინი „პროცედურა“, „მეთოდი“ არის ოპერაციათა სხვა ტიპები. ინფორმატიკაში მისთვის მიღებულია ტერმინი „ყოფაქცევა“ (behavior). ამგვარად, მე-9 კატეგორიაა:

არსთა სქემა - ყოფაქცევის სქემა (ერთად აღებული არის ობიექტის სქემა, ანუ არსთა დამოკიდებულების მოდელი). მონაცემთა და პროცესთა გამოცხადება (declaration) მათი განმეორებადი გამოყენების მიზნით ამის ტიპური მაგალითია.

განვიხილოთ მაგალითი არსთა და ყოფაქცევის სქემისთვის „კვადრატი“. ქცევის სქემისთვის მოიცემა კვადრატის კლასიკური კონსტრუქცია (ნახ.12).



ნახ.12.

არსის სქემისთვის კი გამოიყენება რელაცია „კვადრატი“ რელაციური მოდელის მიხედვით.

- ობიექტის სქემა: კვადრატი.

- ქცევის სქემა:

კონსტრუქცია-კვადრატი Θ და φ -თი.

$g_1 = \Theta \sqcup \varphi;$	$g_5 = \varphi \dashv g_4;$
$g_2 = \Theta \dashv g_1;$	$P_6 = g_2 \cap g_5;$
$g_3 = \varphi \dashv g_1;$	$g_7 = P_6 \dashv g_3;$
$g_4 = g_1 * g_2;$	$P_8 = g_3 \cap g_7;$

- არსთა სქემა: რელაცია-კვადრატი Θ და Φ ელემენტებით, როგორც პირველადი გასაღებებით (ნომონატორები):
Quadrat (Θ, Φ, P_6, P_8 , სიბრტყეთა შინაარსი, ფერები, ...)

END კვადრატი.

აღნიშვნები:

- $\Theta=(x_0, y_0)$ და $\Phi=(x_1, y_1)$ - ორი წინასწარმოცემული წერტილია;

- $P_6=(x_6, y_6), P_8=(x_8, y_8)$ - ორი აგებული წერტილი;

- \square - შემაერთებელი წრფე ;

- \perp - ორთოგონალური შეერთება ;

- $*$ - კუთხის გამყოფი (ბისექტრისა);

- \cap - გადაკვეთის წერტილი.

ობიექტური სქემის აგების უმაღლესი მიზანია სისრულე (სიზუსტე), ანუ ყველაფერი, რაც ეკუთვნის ობიექტებს, უნდა იქნეს აღწერადი როგორც არსთა-სქემა და მისაწვდომი ქცევის-სქემისთვის.

ზემოთ განხილული სქემა არაა სრული, ვინაიდან მასში არაა წარმოდგენილი ოპერაციები გაფერადებისა და სიბრტყეთა განსაზღვრისათვის.

ლოგიკური გრამატიკა, რომლის ნაწილიცაა პრედიკატების ლოგიკა, არის ზოგადი სპეციფიკაციების მედიუმი (საკომუნიკაციო საშუალება).

ამოსავალი წერტილია – ამოცანის დასმის წარმოდგენა, პროექტი ტექნიკურ დავალებათა შესახებ (ან მოვალეობათა რვეული). საბოლოო წერტილი (მიზანი) – მოქნილ ანგარიშთა სისტემა (დანახარჯების) მაღალი ეკონომიურობით. მათ შორის გზის აღწერა არაა მარტივი.

ამ საკითხების გადაწყვეტა ფილოსოფიურ სფეროს ეხება, რომელიც ენასთან და შემეცნებასთანაა დაკავშირებული. ამ სფეროში ნაშრომები ეკუთვნის კანტს, რომელმაც პირველმა შემოიტანა განსხვავება სქემასა და მის გამოსახულებას შორის. ფრეგემ შემოიტანა აბსტარაქტული თეორია და პრედიკატების ლოგიკა. რასელმა და ფრეგემ შექმნეს აღნიშვნათა თეორია. კარნაპმა განავითარა „ინტენსინალ-ექსტენსიონალობის“ თემა და ა.შ..

ჩვენი ამოცანაა აღნიშნული თემის გამოყენება ინფორმატიკის ობიექტ-ორიენტირებული მოდელირების სფეროში, კერძოდ მონაცემთა ბაზებსა და დაპროგრამებაში.

1.3. მეთოდოლოგია

1.3.1. მოთხოვნილებათა არაფორმალისტული წარმოდგენა (ტექნიკური დავალება)

ახალი დანართის (აპლიკაციის) აგება ხორციელდება ცნობილი, უკვე არსებული საშუალებებით, რათა შესაძლებელი გახდეს ახალი ინსტრუმენტის კონსტრუირება. როცა პროგრამულ უზრუნველყოფაზეა დებატები, ამ შემთხვევაში ყოველთვის ენის საშუალებებს გულისხმობენ. მაგალითად, არცერთი პროგრამისტი დღეს არ წერს პროგრამას უკვე არსებულის გამოყენების გარეშე.

Software Engineering ცნობილია “value-added” ტერმინით (იგულისხმება მომგებიანი ინფრასტრუქტურა). იგი ითვალისწინებს მთლიანის აგებას ბიჯურად (ნაწილ-ნაწილ). მეთოდოლოგიაში (ანუ ბიჯურ ტექნოლოგიაში) წესები ახალი საშუალებისათვის შემოიტანება ისეთი, რომლებიც უკვე განსაზღვრული იყო ადრე. ვინაიდან მონაცემები, უკვე არსებულ საშუალებათა შესახებ ტექნიკურ დავალებაში აყენებს ახალ ამოცანებს, ამიტომაც პროექტის აგების მეთოდური დაწესება უნდა იქნას გამოკვლეული. აქ იგულისხმება ტექნიკური დავალება, კატეგორიები და კრიტიკის უნარი (ხელოვნება განსხვავებათა შესახებ).

განვიხილოთ მაგალითი: მანქანების ქარხანაში შეკვეთების დამუშავება. ციტატა ტექნიკური დავალებიდან: „შეკვეთების დამუშავება გულისხმობს ახალი შეკვეთის ბარათების დაფიქსირებას (შეტანას სისტემაში) და უკვე შესრულებული შეკვეთების წაშლას სისტემიდან. შეკვეთაზე ბრძანების მომზადების წინ გამოკვლეული უნდა იქნას, თუა წესრიგში შემკვეთის საკრედიტო მონაცემები და შეკვეთაში მითითებული ნაწილები (დეტალები) თუ შეესაბამება ქარხნის (კონტრაქტის) დასაშვებ ვარიანტს (დეტალების სიას) . . .“.

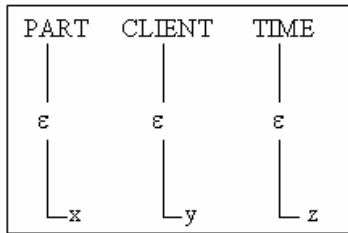
ჩვენი ამოცანაა ლოგიკური გრამატიკის გამოყენებით აღნიშნული ტექსტის რეკონსტრუირება (სტრუქტურის შეცვლა) ენის-კრტიკული თვალსაზრისით. ანუ უნდა ავაგოთ (დავაპროექტოთ) კონცეპტუალური ობიექტური სქემა ახალი მოთხოვნებისათვის. ამავე დროს უნდა გამოვიყენოთ არსებული საშუალებანი. პრობლემას სამეცნიერო დასაწეისისათვის „უკვე არსებულის“ მიხედვით, სამეცნიერო ისტორიას აქვს დიდი ხსნის ტრადიცია.

1.3.2. პირველი ობიექტები მთავარ პრედიკატებში

ტექნიკური დავალება არის არაფორმალისებულად გადმოცემული ტექსტი. კატეგორიული განხილვა, ანუ ტექნიკური დავალების ენის-კრიტიკული (ანალიზი) რეკონსტრუირება (სპეციფიცირება) მოიცავს კომპლექსურ პრობლემას.

I-ბიჯის ამოცანაა ტექნიკური დავალებიდან მთავარი გამონათქვამების გამოყოფა. რა არის მთავარი გამონათქვამი ? ეს დამოკიდებულია მიზნის დასმაზე ამოცანისათვის. ჩვენ გვინტერესებს გარკვევა, მაგ., შეკვეთების დამუშავების ამოცანაში „საზღვარგარეთელი კლიენტი“ და არა „კლიენტური საზღვარგარეთელი“. აქ „საზღვარგარეთელი“ არის არა-მთავარი (!) პრედიკატორი, ვინაიდან ჩვენ არ ვიხილავთ მოქალაქეთა ცნობათა ბიუროს ამოცანას.

მთავარი პრედიკატები, რომლებიც ჩვენ გვინტერესებს, არის: დეტალი, კლიენტი, შიკვითის-დრო (PART, CLIENT, TIME) და ა.შ. (ისინი ჩაიწერება ინგლისური დიდი ასოებით PART. . .) მათ გარეშე პრობლემა არ გადაწყდება. მთავარ პრედიკატორებს საწყის ფაზაზე აქვს ბირთვის (საწყისის) მნიშვნელობა.



ნახ.13.

კატეგორიული თვალსაზრისით უნდა უყოს მოფიქრებული, რომ ჩვენ ვიმყოფებით სქემის-დროში (schma-time) და არა მიმართვის-დროში (referent-time). მაგალითად, „მიუღერი ε კლიენტი“ - ეს არაა განზოგადება, პირიქით „y ε კლიენტი“ არის y-ნომინატორული ცვლადით ყველა შემთხვევისთვის განზოგადებული. მიუღერი მხოლოდ კონკრეტულ მომენტში გააქტიურდება. ე. ი. y - ხდება მიუღერი.

CLIENT და CNR ურთიერთხანაცვლებადია (არ იცვლება ჭეშმარიტება გამონათქვამისა $A_R(X)$).

ρ -აბსტრაქციის თვალსაზრისით არის ეკვივალენტობის რელაცია, რომლისთვისაც სამართლიანია რეფლექსურობის, სიმეტრიულობის და ტრანზიტულობის თვისებები.

განვიხილოთ ექსტენსიონალური აბსტრაქცია მონაცემთა ტიპების მაგალითზე. მოცემულია ორი x და y ცვლადი. უნივერსალობის კვანტორით ჩაიწერება:

$$\forall_{x,y} x \sim_B y \rightarrow (A(x) \leftrightarrow A(y)).$$

სამართლიანია გამოსახულება:

$$\tau x = \tau y = \forall_A (A(x) \leftrightarrow A(y)).$$

ეკვივალენტობის რელაცია \sim_B არის დამუშავების თვალსაზრისით თანაბარი. τ -აბსტრაქტორია ტიპებისთვის, როგორც მაგ., \in - სიმრავლისთვის.

ტიპების სტანდარტული პრედიკატორებია: int, float, char და ა.შ.

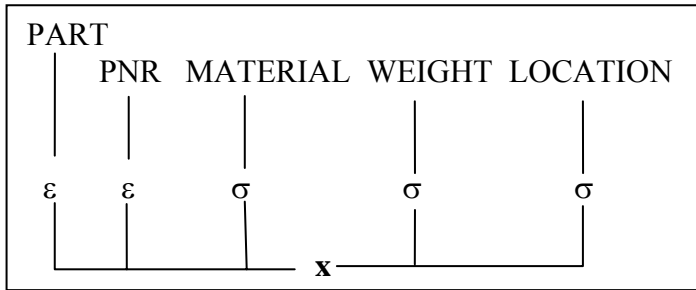
1.3.4. მთავარიდან დამხმარე პრედიკატებისკენ

საგნობრივი სფეროს მოდელირებისას კატეგორიული მიდგომის საფუძველზე საჭიროა ძირითადი და დამხმარე პრედიკატორების გამოვლენა. მათი ასეთი დაყოფა დამოკიდებულია ამოცანის დასმაზე ტექნიკურ დავალებაში. ზოგადად, მეთოდულად ჯერ მთავარ, შემდეგ კი დამხმარე პრედიკატორებს გამოყოფენ. ეს უკანასკნელნი მოდელში აღწერს ობიექტის არამთავარ, ანუ დამატებით მახასიათებლებს.

მაგალითად, წინადადებაში „დეტალს აქვს ხის-ნაწილი“, შემოიტანება ახალი σ -კოპულა, რომელიც „აქვს“ ოპერაციას ასახავს. ამგვარად, გვექნება:

„დეტალს σ ხის-ნაწილი“.

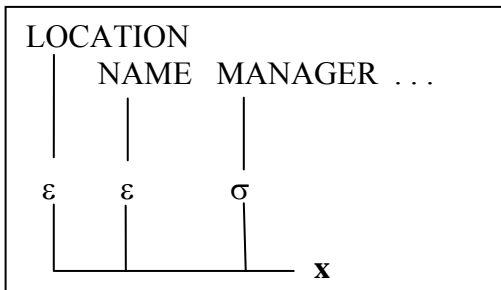
წინა პარაგრაფში განხილული სქემა, ამჯერად დამატებითი პრედიკატორებით გაფართოების საფუძველზე შეიძლება ასე წარმოვადგინოთ:



ნახ.1.5

სადაც დამხმარე პრედიკატორებია: MATERIAL-მასალა, WEIGHT-წონა და LOCATION –საწყოების მდებარეობა.

თუ საწყოებს განვიხილავთ როგორც მთავარ პრედიკატორს, მაშინ მისთვის სხვა დამხმარე პრედიკატორები გაჩნდება, მაგ., ნახაზი 1.6.



ნახ.1.6

დამხმარე პრედიკატორების საშუალებით პრედიკატული სქემის გაფართოების პროცესს ლიტერატურაში მოიხსენებენ აგრეთვე რელაციური სქემის სახით. ესაა მონაცემთა რელაციური ბაზის კონცეპტუალური სქემის საფუძველი. დამხმარე პრედიკატორების გამოყენებას კავშირი აქვს მონაცემთა ნორმალიზაციის თეორიასთან, კერძოდ ესაა წესების სისტემა დამხმარე პრედიკატორების შედგენილობის განსაზღვრისათვის.

რელაცია, ჩვეულებისამებრ თეორიაში განიხილება როგორც ატრიბუტთა (დამხმარე კატეგორიების) დეკარტული ნამრავლი:

$PART \subseteq PNR \times MATERIAL \times WEIGHT \times LOCATION$

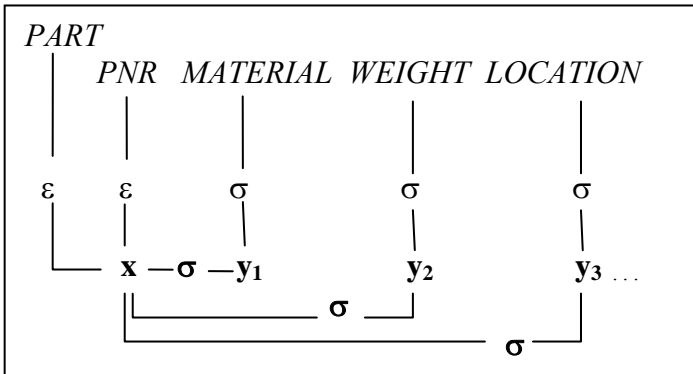
რელაციები – მრავალადგილიანი სიმრავლეებია, რომლებიც ერთადგილიანი სიმრავლეებისგან, ანუ კლასებისგან განსხვავდება. რელაციები და კლასები, ანუ სიმრავლეები კონსტრუქციულად აბსტრაქციის საფუძველზე ობიექტის მდგომარეობათა მოწესრიგებულ-კორტეჟთა ერთობლიობაა (ნახ.1.7).

PART

<i>PNR</i>	<i>MATERIAL</i>	<i>WEIGHT</i>	<i>LOCATION</i>
123	ფოლადი	24,5	თბილისი
124	X _j	10,25	ბათუმი
135	პლასტმასი	27,55	თბილისი
...

ნახ.1.7.

შეიძლება ჩაიწეროს: „x_j ∈ MATERIAL“, „10,25 ∈ WEIGHT“ და ა.შ. დამხმარე პრედიკატორების შიგნით, იერარქიული თვალსაზრისით, ჩნდება მთავარი პრედიკატორები (ნახ.1.8):



ნახ.1.8

ასეთი სქემა აღარაა ელემენტარული. ზოგიერთი დამხმარე პრედიკატორი ლოგიკურად შეიძლება კონიუნქციის სახით ჩაიწეროს:

$$x \in (y_1)(PART) \wedge x \in (y_2)(PART).$$

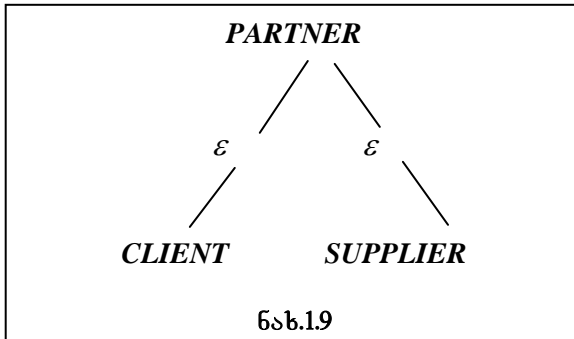
1.3.5. აბსტრაქცია – განზოგადება

საგნობრივი სფეროს აღწერისას ხდება მისი ობიექტების კლასიფიკაცია, მათი დაყოფა და მოწესრიგება (კლასი, ქვეკლასი და ა.შ.). ანუ ხდება გარკვეული სუბორდინაცია, კონკრეტულიდან აბსტრაქტულზე ასვლა გარკვეული მემკვიდრეობითი თვისებების საფუძველზე. ამ თვალსაზრისით განიხილება სახეობათა (ქვეკლასის) პრედიკატორები და კლასის (სუპერკლასის) პრედიკატორები.

მაგალითად, საპრობლემო სფეროსთვის „პროდუქციის წარმოება“ არსები კლიენტი (პროდუქციის მყიდველი - CLIENT) და მიმწოდებელი (ნედლეულის მიმწოდებელი- SUPPLIER) შეიძლება შეესაბამებოდეს სახეობათა პრედიკატორებს, მაშინ მათთვის უფრო განზოგადოებული ცნება წარმოებულს-პარტნიორი (PARTNER) იქნება კლასის პრედიკატორი, რომელიც ორივეს მოიცავს (ნახ.2.9).

$$\wedge_x x \in \text{CLIENT} \rightarrow x \in \text{PARTNER}$$

$$\wedge_x x \in \text{SUPPLIER} \rightarrow x \in \text{PARTNER}$$



ნახ.19

ახალი ცნებების შემოტანა ასევე ლოგიკური კავშირების გამოყენებითაა შესაძლებელი. მაგალითად, კლიენტი ანუ დამკვეთი არის ამავე დროს მიმწოდებელიც. ეს ნიშნავს, რომ პროდუქციის მწარმოებელი ფირმა, ერთის მხრივ უკვეთავს ნედლეულს სხვა ფირმას (არის დამკვეთი) და მეორეს მხრივ, ის ამზადებს პროდუქციას და მიაწვდის კლიენტს (არის მიმწოდებელი). ასეთი წესები შეიძლება ლოგიკურად ჩავწეროთ შემდეგნაირად:

$$\wedge_x x \in C \vee x \in P \rightarrow x \in S,$$

$$\wedge_x x \in C \wedge x \in P \rightarrow x \in S,$$

სადაც \vee - ლოგიკური დიზიუნქცია, ხოლო \wedge - ლოგიკური კონიუნქციაა.

ამგვარად, აბსტრაქცია ახალი ობიექტების შემოტანის ერთ-ერთი გზაა კლასიფიკაციის სისტემაში, რომლის საფუძველსაც მემკვიდრეობითობის თვისებები წარმოადგენს კლასებს შორის.

1.3.6. მერეოლოგიური სტრუქტურები

გარდა ამისა არსებობს ობიექტების და კლასთაშორის კავშირების ასახვის მეორე გზაც, რომელსაც მერეოლოგიურს, ანუ „მთელი-ნაწილი“-დამოკიდებულების ლოგიკას უწოდებენ. აქ ყურადღება მახვილდება სტრუქტურულ შედგენილობაზე, ანუ აგრეგაციაზე. მაგალითად, პროდუქცია შედგება სხვადასხვა ნაწილისგან.

„მთელი-ნაწილი“-დამოკიდებულებათა ლოგიკის ფუძემდებელია პოლონელი მათემატიკოსი ლეზნიევსკი (1916 წ.) პრედიკატულ-აბსტრაქტული ლოგიკისგან განსხვავებით, რომელიც მაგალითად, რასელის „მათემატიკის პრინციპებიდანაა“ ცნობილი, მერეოლოგიური ენა „მთელი-ნაწილი“-რელაციასთან (\leq) ერთად ცნობს მხოლოდ კატეგორიებს: „სახელი“ და „გამონათქვამი“.

მთელი/ნაწილი – რელაცია ($x \leq y$, x არის y -ის ნაწილი) სამართლიანია მხოლოდ ლოგიკური ტიპისათვის, როგორც ნაწილისთვის და განსხვავდება ელემენტი/სიმრავლე – რელაციისაგან \in , რომელიც პრედიკაციასა და აბსტრაქციას ითვალისწინებს. ობიექტებისათვის არსებობს სახელები, რომლებიც როგორც კონსტანტები ან ცვლადებია შედგენილ ობიექტებში.

მაგალითად, გამონათქვამი „123 \in CLIENT“ (123 არის კლიენტი) შეიცვლება მერეოლოგიური თვალსაზრისით გამონათქვამით „123 \leq CUSTOMERS“, „123 არის ნაწილი კლიენტურისა“. აქ „კლიენტურა“ შედგენილი ნაწილების სახელია.

$$\text{CUSTOMERS} = t_z \wedge_x x \leq z.$$

ყველა x , რომლებიც ერთტიპური z -ის ნაწილია, არის CUSTOMERS.

ჩვენ განვიხილეთ ორი ლოგიკა: პრედიკატულ-აბსტრაქტული და მერეოლოგიური.

პედიკატულ-აბსტრაქტული ლოგიკა ცნობს ორი სახის ობიექტებს, კონკრეტულებს და აბსტრაქტულებს. მას ასევე რეალისტურ ლოგიკასაც უწოდებენ.

მერეოლოგია არის ნომინალიზმის ლოგიკა (ფილოსოფიური მიმართულება, რომელიც უარყოფს ზოგადი ცნებების არსებობას). მისი კონცეფციით არსებობს მხოლოდ კონკრეტული ობიექტები და მათი სტრუქტურები. მათ შეესაბამება სახელები და გამონათქვამები.

ინფორმატიკოსებისათვის ელემენტარული გამოსახულებანი:

$$x \in P \text{ და } x \leq P$$

უნდა ჩავთვალოთ დუალურად.

ნომინალისტ-ინფორმატიკოსები სააზროვნოდ იყენებენ ხისმაგვარ სტრუქტურას, რომელიც x -შივილი- და y -მშობელი კვანძებისგან შედგება, სადაც სამართლიანია: $x \leq y$.

\leq -თვის ასევე სამართლიანია შემდეგი წესები:

- რეფლექსურობის: $\wedge_x x \leq x$.

- ტრანზიტულობის: $\wedge_{x,y} (x \leq y \wedge y \leq z \rightarrow x \leq z)$

- <საკუთარი სახელი> = $k_x A(x)$. მთელი x , რომელიც აკმაყოფილებს $A(x)$ -გამონათქვამს.

2.10 ნახაზზე ნაჩვენებია მერეოლოგიური სტრუქტურის მაგალითი პროდუქციის შეკვეთების ამოცანისათვის.

ორი u და p ობიექტის წარმოებული ($u * p$) არის ახალი ობიექტი w , რომელსაც იგივე შედგენილობა აქვს, რაც u -ს და p -ს ერთად.

R -შეკვეთისათვის ლოგიკურად შეიძლება დაიწეროს:

$$R = t_p \vee_{y,z} y \leq p \wedge_x z \leq p.$$

ფაქტიური შეკვეთისათვის $u * p$ - წარმოებულისთვის იქნება:

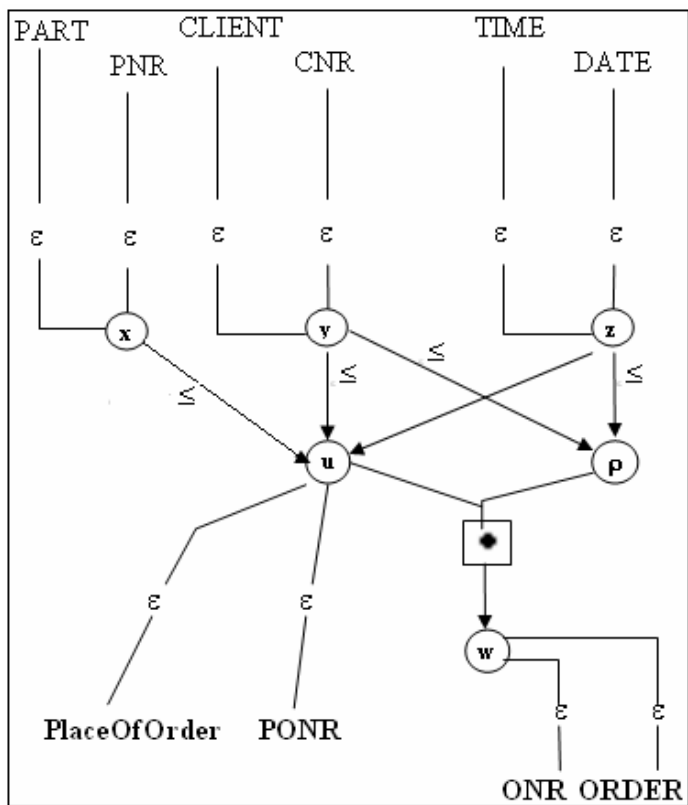
$$u * p = t_w \wedge_r (r \leq w) \longleftrightarrow (r \leq u \wedge r \leq p).$$

დამხმარე ცვლადი p ჩვენს შემთხვევაში აირჩევა ისე, რომ r -თავისუფალ ცვლადებს შეეძლოთ ობიექტების მიღება მხოლოდ y და z -სგან.

w -შეკვეთას და u -შეკვეთის ფოსტას აქვს ერთნაირი CNR (y) და DATE (z).

საკუთარი სახელი W შეკვეთისათვის მაშინ მოიცემა, როდესაც $u * p$ პროდუქცია არსებობს.

$$W = \vee_{u,p} u * p.$$



ნახ.1.10. მერეოლოგიური სტრუქტურის ფრაგმენტი ამოცანისათვის „პროდუქციის შეკვეთა“

II თავი

პრაქტიკული ნაწილი (ამოცანები)

2.1. ობიექტ-როლური მოდელირება (ORM) განაწილებული ოფის-ობიექტებისათვის

ცნობილია, რომ მონაცემთა ბაზის აპლიკაციის ხარისხი კრიტიკულადაა დამოკიდებული მის დაპროექტებაზე. ინფორმაციული სისტემების განსაზღვრა ყველაზე უკეთ კონცეპტუალურ დონეზე ხდება, სადაც გამოიყენება ისეთი კონცეფციები და ენა რომელიც ადვილი გასაგებია ადამიანისათვის. კონცეპტუალური დაპროექტება შეიძლება შეიცავდეს მონაცემებს, პროცესებს და ქცევით პერსპექტივებს, ხოლო მონაცემთა ბაზების მართვის სისტემა (DBMS) დაპროექტებისას გამოიყენებდა ცნობილი ლოგიკური მოდელებიდან (რელაციური, იერარქიული, ქსელური, ობიექტ-ორიენტირებული და ა.შ.) ერთ-ერთს.

მონაცემთა ბაზის დაპროექტება მოიცავს საპრობლემო არის ფორმალური მოდელის აგებას ანუ მისმ აღწერის (საუბრის) ფორმირებას (*universe of discourse Uod*) სათანადოდ ამის გაკეთება კი დამოკიდებულია *Uod*-ის კარგ ცოდნაზე. ობიექტ-როლური მოდელირება (ORM) ამარტივებს დაპროექტების პროცესს, იყენებს რა ბუნებრივ, სალაპარაკო ენას, ასევე ინტუიციურ დიაგრამებს, რომელთა შევსებაც შეიძლება მაგალითების საშუალებით და შესაძლებელია ინ ორმაციის შემოწმება მარტივ, ელემენტარულ ფაქტებზე დაყრდნობით. ვინაიდან მოდელი გამოსახულია ისეთ ბუნებრივ ტერმინებში, როგორცაა ობიექტი და როლი, იგი უზრუნველყოფს მოდელირების კონცეპტუალურ მიდგომას[17-19].

ობიექტ-როლური მოდელირების ადრეული ვერსია 1970-იან წლებში გამოჩნდა ევროპაში. აქ აღწერილი მეთოდი დამუშავებულია ავსტრალიასა და აშშ-ში. ასოციაციური ენა *FORML (Formal Object-Role Modeling Language)* თანმხლებია *Microsoft Visio for Enterprise Architect (VEA)*-სი, რომელიც *Visual Studio.NET*-ის შემადგენელი ნაწილია.

კონცეპტუალური მოდელირება მიიღწევა არსთა დამოკიდებულების (*ER*) მოდელითაც თუმცა (*ER*) მოდელი შეიძლება გამოვიყენოთ მას შემდეგ, რაც დაპროექტების პროცესი დამთავრებულია. იგი ნაკლებადაა შესაფერისი ფორმულირებისათვის, განალებასა და პროექტის შემდგომი გაფართოებისათვის. *ER*-დიაგრამა შორსაა ბუნებრივი ენისაგან, ვერ ხერხდება შევსება ამა

თუ იმ მოვლენის ფაქტით, დამალულია ინფორმაცია იმ სემანტიკური დომენების შესახებ, რომლებიც ქმნის მოდელს.

ამრიგად, კონცეპტუალური მოდელირების განვითარებულ ტექნიკას წარმოადგენს ობიექტ-როლური მოდელირება. სწორედ *ORM*-ს შეუძლია უზრუნველყოს სხვადასხვა პროფესიის ადამიანების შეთანხმებული მუშაობა, რომელთა მომზადების დონე ინფორმაციული სისტემების დაპროექტების სფეროში შეიძლება მნიშვნელოვად განსხვავდებოდეს.

ზემოაღწერილი მიზეზებიდან გამომდინარე, კონცეპტუალური მოდელირებისათვის ჩვენ ვირჩევთ *ORM*-ს. საინფორმაციო სისტემების ცხოვრების ციკლი მოიცავს რამდენიმე სტადიას: ტექნიკურ-ეკონომიკური დასაბუთება, მოთხოვნათა ანალიზი, მონაცემებისა და ოპერაციების კონცეპტუალური დაპროექტება; ლოგიკური დაპროექტება; გარე დაპროექტება; მაკეტირება; შიგა დაპროექტება და შესრულება; ტესტირება და შესწორების შეტანა; მომსახურება (თანხლება).

ORM-ის კონცეპტუალური მოდელირების სქემის პროცედურა (*CSDP*) ყურადღებას ამახვილებს მონა-ცემების ანალიზზე და დაპროექტებაზე. კონცეპტუალური სქემა აღწერს აპლიკაციის ინფორმაციულ სტრუქტურას: ფაქტების ტიპები, რომლებიც წარმოადგენს ინტერესის სფეროს; მასზე არსებული შეზღუდვები და შესაძლოა წარმოქმნის წესები, რათა მივიღოთ ესა თუ ის ფაქტი სხვა ფაქტებიდან.

მრავალმასშტაბიანი აპლიკაციისათვის *Uod*-ი ყოფს მას მოხერხებულ მოდულებად, *CSDP*-ს მიმართავს თითოეული მათგანი და შემდეგ ხდება მიღებული კონცეპტუალური ქვესქემების გაერთიანება ერთ გლობალურ კონცეპტუალურ სქემად. თვითონ *CSDP*-ი შედგება შვიდი ბიჯისაგან. იხ. ცხრილი 1.1.

ბიჯი 1 – *CSDP*-ის ყველაზე მნიშვნელოვანი სტადიაა, სადაც ხდება სხვადასხვა სახის ინფორმაციის შეგროვება, ბუნებრივ სალაპარაკო ენაზე. ასეთი ინფორმაცია ხშირად არის შემაჯავლი და გამომავალი ფორმები, შეიძლება იყოს ხელნაწერი. წინააღმდეგ შემთხვევაში მოდელის დამპროექტებელი მუშაობს უშუალოდ კლიენტთან, რათა ზუსტად ჩამოყალიბდეს, თუ რა მოეთხოვება სისტემას. აუცილებელია ექსპერტის არსებობა, რომელიც იცნობს აპლიკაციას.

1. ელემენტარული ფაქტების ფორმირება და მათი ადეკვატურობის შემოწმება;
2. ფაქტების ტიპებისათვის დიაგრამის აგება და სისრულის შემოწმება;
3. იმ ობიექტთა ტიპების შემოწმება, რომლებიც უნდა გაერთიანდეს და მათი მათემატიკური წარმომავლობის დაფიქსირება;
4. დაემატოს უნიკალურობის შეზღუდვა და შემოწმდეს ფაქტების ტიპების ოპერანდების რაოდენობა;
5. დაემატოს როლების იძულებითი შეზღუდვები და შემოწმდეს მათი ლოგიკური წარმომავლობა;
6. დაემატოს ელემენტები, სიმრავლეთა შედარება და ქვეტაპის შეზღუდვები;
7. დაემატოს სხვა შეზღუდვები და მოხდეს საბოლოო შემოწმება.

ეს ფაქტი ასე შეიძლება ჩაიწეროს:

- f1 თანამშრომელს, ნომრით 35, აქვს გვარი ‘პეტრიაშვილი’
- f2 თანამშრომელი, ნომრით 7, მუშაობს კონტრაქტით თარიღამდე ‘12.31.06’

თითოეული ფაქტი არის ბინარული დამოკიდებულება ორ ობიექტს შორის. მუქი შრიფტით გამოყოფილია ლოგიკური პრედიკატი, რომელიც ახდენს ობიექტების იდენტიფიცირებას ნაჩვენებია კურსივით. იმ შემთხვევაში თუ განისაზღვრება ობიექტის მხოლოდ ერთი თვისება, საქმე გვაქვს ერთადგილიან პრედიკატთან (unary fact). პრედიკატს შეიძლება ჰქონდეს (1,2,3,..) ოპერანდი, თუმცა რადგან პრედიკატი ელემენტარულია 3-4 ოპერანდზე მეტი იშვიათად გვხვდება. უმრავლეს შემთხვევაში პრედიკატი არის ორობითი. ასეთი პრედიკატებისათვის არსებობს ინვერსული პრედიკატი. ისე, რომ ფაქტი შეიძლება წავიკითხოთ ორივე მიმართულებით.

ბიჯი 2 – აქ ხდება ფაქტების ტიპებისათვის დიაგრამის აგება. ობიექტები გამოსახება ელიფსებით, პრედიკატები მართკუთხედებით, მნიშვნელობის ტიპი გამოსახება წვეტილი ელიფსით. პრედიკატი იკითხება მარცხნიდან-მარჯვნივ და ზემოდან-ქვემოთ, მანამ სანამ არ შეხვდება ნიშანი “<<”, რომელიც ცვლის წაკითხვის მიმართულებას საწინააღმდეგო მიმართულებით. შემდეგ ბიჯებზე ხდება შეზღუდვების დაწესება.

ORM დიაგრამაში გამოყენებული შეზღუდვები:

იძულების შეზღუდვები:



ობიექტი ასრულებს ზუსტად გარკვეულ როლს.
როლების დიზუნქცია არის იძულებითი. თითოეული ობიექტი
ობიექტების ტიპების ნაკრებიდან უნდა ასრულებდეს მხოლოდ
ერთ როლს.

უნიკალურობის შეზღუდვები:



ერთ ან მეტ როლში მონაწილეობა ხდება არა
უმეტეს ერთხელ.

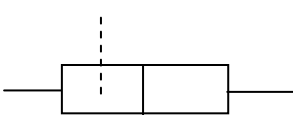


როლების გარე უნიკალურობის შეზღუდვა.
წყვილის გამორიცხვის შეზღუდვა.

სიმრავლეების შედარების შეზღუდვები:

- \subseteq პირველი ობიექტის სიმრავლე ყოველთვის უნდა იყოს
მეორის ქვესიმრავლე.
- $=$ პირველი ობიექტის სიმრავლე ყოველთვის
უნდა იყოს მეორის ტოლი.
- $\not\subseteq$ პირველი ობიექტის სიმრავლე არ შედის მეორეში.
სისშირის შეზღუდვა.

≤ 2



ობიექტმა რამდენჯერ შეიძლება შეასრულოს
ეს როლი

ბუდის ტიპის ობიექტი.

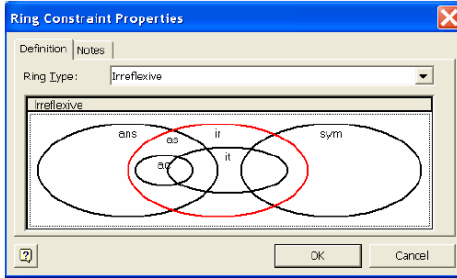


ობიექტი თამაშობს მხოლოდ ერთ როლს და ეს
როლი არ არის სავალდებულო.



ქვეტიპი.
ერთი ობიექტი არის მეორის ქვეტიპი

წრიული შეზღუდვები



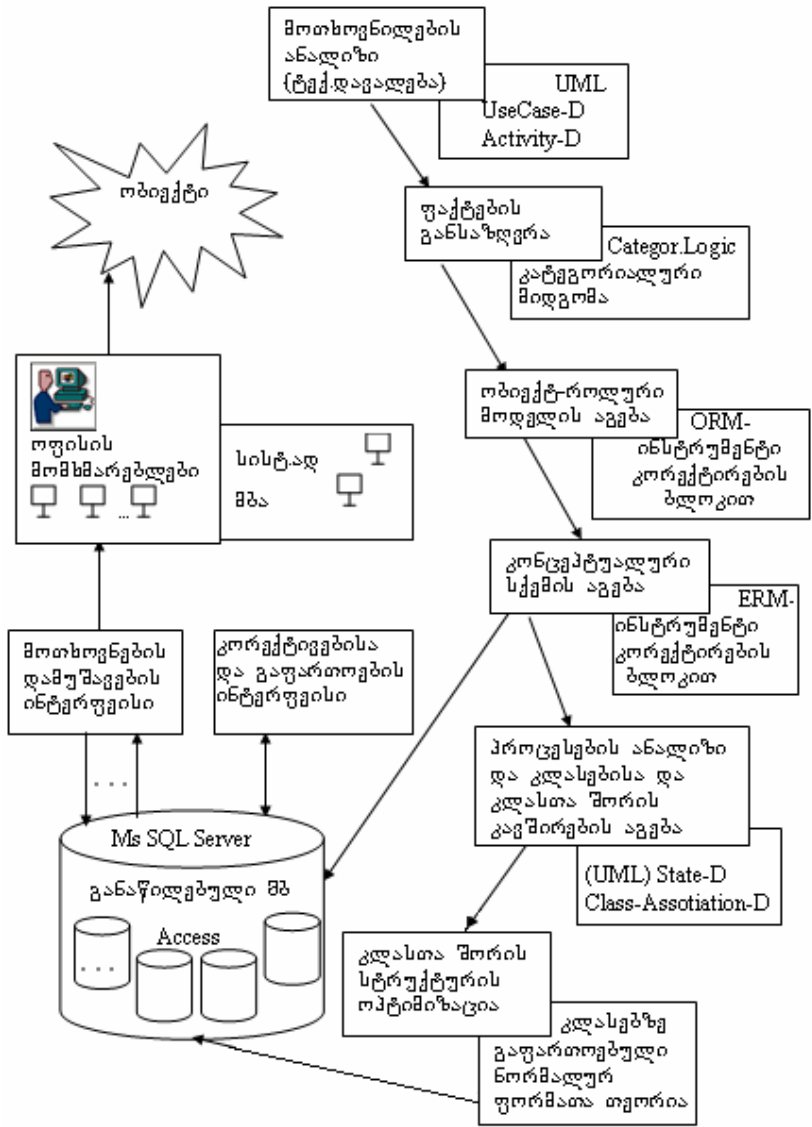
ანტირეფლექსურობა	Oir	iff for all $x, \sim xRx$
სიმეტრიულობა		$Osym$ iff for all $x, y, xRy \rightarrow yRx$
ასიმეტრიულობა		Oas iff for all $x, y, xRy \rightarrow \sim yRx$
ანტისიმეტრიულობა	$Oans$	iff for all $x, y, x \neq y \ \& \ xRy \rightarrow \sim yRx$
ანტიტრანზიტულობა	Oit	iff for all $x, y, z, xRy \ \& \ yRz \rightarrow \sim xRz$
აციკლურობა	Oac	iff for all $x, y, z, xRy \ \& \ yRz \rightarrow \sim zR$

2.2. ამოცანის დასმა

განიხილება განაწილებული ოვის-ობიექტების მონაცემთა ბაზების ობიექტ-ორიენტირებული მოდელირებისა და ობიექტ-ორიენტირებული დაპროექტების ავტომატიზებული პროცესების დამუშავების ამოცანა, რაც საგრძნობლად ამცირებს სისტემების ინფორმაციული და პროგრამული პაკეტების აგების დროს და, ამასთანავე ორიენტირებული გამოყენებითი სფეროს მომხმარებელზე.

2.1 ნახაზზე მოცემულია საპრობლემო სფეროს მოდელირებისა და მონაცემთა ბაზაში ავტომატიზებული ასახვის ამოცანის ძირითადი ეტაპებისა და მათი რეალიზაციის ინსტრუმენტული საშუალებების სქემა.

ცოდნა, რომელიც საპრობლემო სფეროს შესახებ გააჩნია მომხმარებელს, სპეციალური ინტერფეისების საშუალებით, რომელთა საფუძველს ფორმალური ენის გრამატიკის კატეგორიები და ლოგიკურ-ალგებრული მეთოდები შეადგენს, გადაეცემა ობიექტ-როლური მოდელირების კომპიუტერულ პროგრამას.



ნახ.2.1.

მისი დახმარებით აიკვება სემანტიკური სტრუქტურები *ORM*-დიაგრამათა სახით. შემდგომ ეტაპზე ავტომატიზებული პროცედურების გამოყენებით ფორმირდება არსთა-დამოკიდებულების მოდელი, ანუ *ER*-დიაგრამები, ცხრილებითა და ატრიბუტებით. მის საფუძველზე ავტომატურად გენერირდება *.DDL* ფაილები, რომლებიც შემდგომ სტრუქტურულად მოთავსდება *Ms SQL Server* მონაცემთა განაწილებულ ბაზაში.

როგორც აღვნიშნეთ, აქ პრობლემურად ისმება, *ER*-მოდელის შესაბამისი კლასების დიაგრამის „სტრუქტურის ოპტიმიზაციის“ საკითხი ანუ მონაცემთა ბაზების დაპროექტების თეორიის ტერმინით - ნორმალიზაციის ამოცანა.

აქ იგულისხმება შემდეგი: უნდა განისაზღვროს თითოეული კლასის მონაცემთა ოპტიმალური სტრუქტურა (მონაცემთა მოდელი), კლასთა შორის კავშირების გათვალისწინებით და დამოკიდებულებათა რელაციების ნორმალიზაციის მეთოდების გამოყენებით.

ოპტიმიზაციის კრიტერიუმად შეიძლება განვიხილოთ რელაციური ცხრილების ინფორმაციული და ინდექსური მონაცემების მოცულობათა მინიმიზაცია, და მათთან მიმართვისა და განახლების დროის შემცირება. მონაცემთა ერთიანი ლოგიკური სტრუქტურების მთლიანობის უზრუნველყოფით (დაცვით). ამისათვის შევიმუშავეთ სპეციალური ალგორითმი.

ნაშრომი ეფუძნება *UML*-ტექნოლოგიისა და მონაცემთა ლოგიკური სტრუქტურების რელაციურ დამოკიდებულებათა თეორიის კლასიკურ საკითხებს. ასეთი ინტეგრირებული მიდგომა დასმული ამოცანის გადასაწყვეტად უზრუნველყოფს, ერთის მხრივ, მონაცემთა სტრუქტურების ოპტიმიზაციას ნორმალურ ფორმათა თეორიის საფუძველზე და, მეორეს მხრივ, დაპროექტების ობიექტ-ორიენტირებული პრინციპების მხარდაჭერას *Case*-ტექნოლოგიების გამოყენებით (დაპროგრამების ჯგუფური ტექნოლოგიები), რაც აქტუალური და მნიშვნელოვანი ამოცანაა.

აღნიშნული საკითხების თეორიული გადაწყვეტა და მათი პრაქტიკული რეალიზაცია ასახულია წიგნის მომდევნო თავებში.

2.3. ფაქტების მოზადება და *ORM*-დიაგრამის აგება საკონტრაქტო ამოცანისთვის

პირველ თავში ჩვენ ზოგადად აღწერეთ კონცეპტუალური მოდელირებისათვის *ORM*-მეთოდის გამოყენება.

საინფორმაციო სისტემების ცხოვრების ციკლი, როგორც ცნობილია მოიცავს რამდენიმე სტადიას: ტექნიკურ-ეკონომიკური დასაბუთება, მოთხოვნათა ანალიზი, მონაცემებისა და ოპერაციების კონცეპტუალური დაპროექტება; ლოგიკური დაპროექტება; გარე დაპროექტება; მაკეტირება; შიდა დაპროექტება და შესრულება; ტესტირება და შესწორების შეტანა; მომსახურება (თანხლება).

ORM-ის კონცეპტუალური მოდელირების სქემის პროცედურა (*CSDP*) ყურადღებას ამახვილებდა მონაცემთა ანალიზზე და დაპროექტებაზე. კონცეპტუალური სქემა აღწერს აპლიკაციის ინფორმაციულ სტრუქტურას: ფაქტების ტიპები, რომლებიც წარმოადგენს ინტერესის სფეროს; მასზე არსებული შეზღუდვები და შესაძლოა წარმოქმნის წესები, რათა მივიღოთ ესა თუ ის ფაქტი სხვა ფაქტებიდან. ახლა განვიხილოთ მაგალითები.

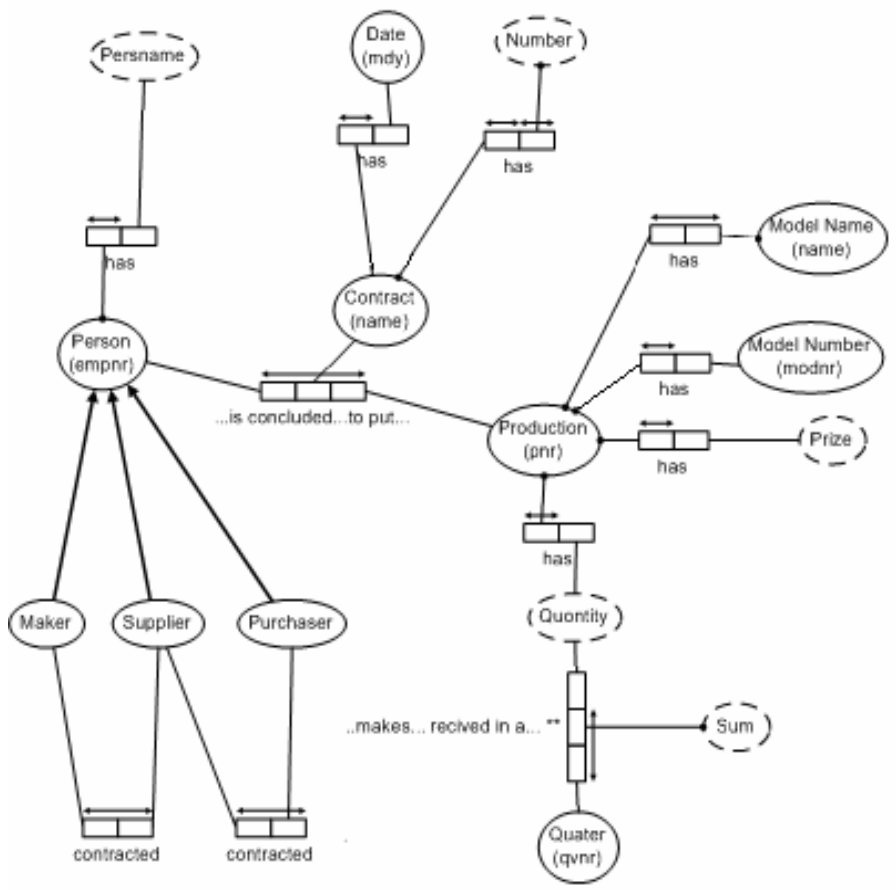
ოფის-ობიექტებზე ერთ-ერთი მნიშვნელოვანი ამოცანაა კონტრაქტების გაფორმება პროდუქციის დამკვეთ და მიმწოდებელ ორგანიზაციებს შორის.

განვიხილოთ ამ ამოცანისათვის შესაბამისი *ORM*-დიაგრამის აგების პროცესი. ამისათვის „კონტრაქტის დადების“ პროცედურის ტექსტური (შინაარსობრივი) ფორმიდან უნდა შევარჩიოთ და ჩამოვაყალიბოთ ფაქტების ერთობლიობა.

მაგალითად, ოფის-ობიექტზე „კონტრაქტის დადების“ ამოცანისთვის გვექნება შემდეგი ფაქტები:

- f1 - პერსონას (სუბიექტს) აქვს სახელი;
- f2 - დამამზადებელი არის სუბიექტი;
- f3 - მიმწოდებელი არის სუბიექტი;
- f4 - მყიდველი არის სუბიექტი;
- f5 - სუბიექტმა დადო კონტრაქტი პროდუქციის შესყიდვაზე;
- f6 - კონტრაქტს აქვს ნომერი;
- f7 - კონტრაქტს აქვს სახელი;
- f8- კონტრაქტს აქვს თარიღი;
- f9 - პროდუქციას აქვს მოდელის დასახელება;
- f10 - პროდუქციას აქვს მოდელის ფასი;
- f11 - პროდუქციას აქვს მოდელის ნომერი;
- f12 - პროდუქციას აქვს რაოდენობა;
- f13 - პროდუქცია გარკვეული რაოდენობით და გარკვეული ფასით გაიყიდა კვარტალში;
- f14 - თანხა არის ფასი, გამრავლებული რაოდენობაზე.

ORM-დიაგრამაზე გამოსათვლელი ფაქტის ტიპი აღინიშნება „*“-ით. ოფის-ობიექტზე კონტრაქტის დადების *ORM*-დიაგრამა ნაჩვენებია 2.2 ნახაზზე.



ნახ.2.2. „კონტრაქტის“ ORM-მოდელის ფრაგმენტი

2.4. ORM-დიაგრამის აგება უნივერსიტეტის მაგალითზე

საუნივერსიტეტო მართვის სისტემის საპრობლემო სფეროს მაგალითზე საჭიროა მონაცემთა ისეთი განაწილებული ბაზის დაპროექტება და აგება, რომელიც მომავალში მიუერთდება ქვეყნის ერთიანი საგანმანათლებლო სისტემის მონაცემთა საცავს.

ასეთი საინფორმაციო სისტემის აგებისას აუცილებელია მიღებული საერთაშორისო სტანდარტების გათვალისწინება, რაც მნიშვნელოვნად შეუწყობს ხელს მის შემდგომ მასშტაბირებას და განვითარებას.

თანამედროვე სტანდარტს (უნიფიცირებული მოდელირების ენას, წესებს) წარმოადგენს *UML*-ტექნოლოგია, ხოლო მის ერთ-ერთ გავრცელებულ ინსტრუმენტს *Ms Visio*.

მოცემული პარაგრაფის მიზანია ამ ინსტრუმენტების გამოყენებით მონაცემთა ბაზის კონცეპტუალური სქემის ასაგებად წინა პროცედურების მომზადება.

საუნივერსიტეტო მართვის სისტემის, კერძოდ მისი მონაცემთა განაწილებული ბაზის დასაპროექტებლად, როგორც ეს *UML*-ტექნოლოგიით მოითხოვება, საჭიროა საპრობლემო სფეროს შესწავლა და მისი ტექსტური (არაფორმალური) აღწერა, აქ არსებული მიზნების, ამოცანების, ობიექტების, ფაქტორების და სხვა ელემენტების, აგრეთვე მათი ურთიერთკავშირების დეტალური ჩამოყალიბებით.

ესაა ცოდნა სამართავი ობიექტის შესახებ, მისი სტატიკური (მდგომარეობათა სიმრავლე) და დინამიკური (ქცევათა სიმრავლე) მოდულებით. თუ ობიექტ-ორიენტირებული მოდელირების ტერმინებით ვისარგებლებთ, დასმული ამოცანის გადაწყვეტის „გასაღებს“ კლასების, ობიექტების, კლასთაშორისი კავშირების, ობიექტ-როლური და არსთა-დამოკიდებულების მოდულებისა და სხვა სახის დიაგრამების აგება წარმოადგენს. ხოლო შემდეგ, კლასთა-ასოციაციებისა და არსთა-დამოკიდებულების დიაგრამათა საფუძველზე განხორციელდება მიზნობრივი სისტემის პროგრამული კოდების რეალიზაციის ავტომატიზებული პროცესი.

განვიხილოთ ეს საკითხები დეტალურად უნივერსიტეტის საპრობლემო სფეროს მაგალითზე. არაფორმალური აღწერის

ობიექტებია (ტერმინთა ლექსიკონი): ფაკულტეტები, კათედრები, სტუდენტები, ლექტორები (როგორც თანამშრომლები, ისე მოწვეულები, საშტატო და საათობრივი დატვირთვით), საგნები (აკადემიური დისციპლინები), რომლებიც იკითხება კათედრებისა და სპეციალობების მიხედვით. სასწავლო გეგმები, სილაბუსები (პროგრამები), ლექციები, პრაქტიკული და ლაბორატორიული სამუშაოები, გამოცდები, ტესტირება და ა.შ.

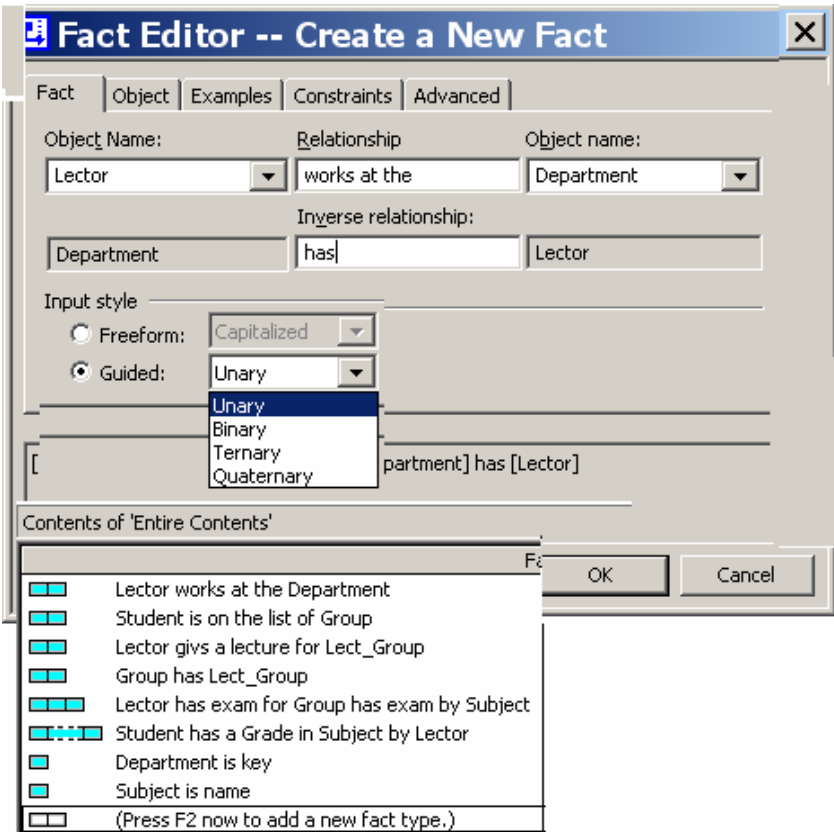
მისი ფაქტების ერთობლიობა იქნება:

- f1 - თანამშრომელს (ლექტორს) აქვს სახელი.**
 - f2 - თანამშრომელი მუშაობს დეპარტამენტში.**
 - f3 - თანამშრომელს დაკავებული აქვს ოთახი.**
 - f4 - თანამშრომელი მუშაობს კონტრაქტით თარიღამდე.**
 - f5 - თანამშრომელს უკავია თანამდებობა**
 - f6 - თანამშრომელმა მიიღო ხარისხი უნივერსიტეტიდან.**
 - f7 - თანამშრომელი ამოწმებს თანამშრომელს.**
 - f8 - თანამშრომელს აქვს წოდება**
 - f9 - თანამშრომელი ასწავლის საგანს.**
 - f10 - თანამშრომელი ატარებს რეიტინგს.**
 - f11 - პროფესორი ხელმძღვანელობს დეპარტამენტს.**
 - f10 - დეპარტამენტს აქვს ბიუჯეტი.**
 - f11 - პროფესორს კავია თანამდებობა კათედრაზე.**
 - f12 - მასწავლებელი არის თანამშრომელი.**
 - f13 - პროფესორი არის თანამშრომელი**
- და ა შ.**

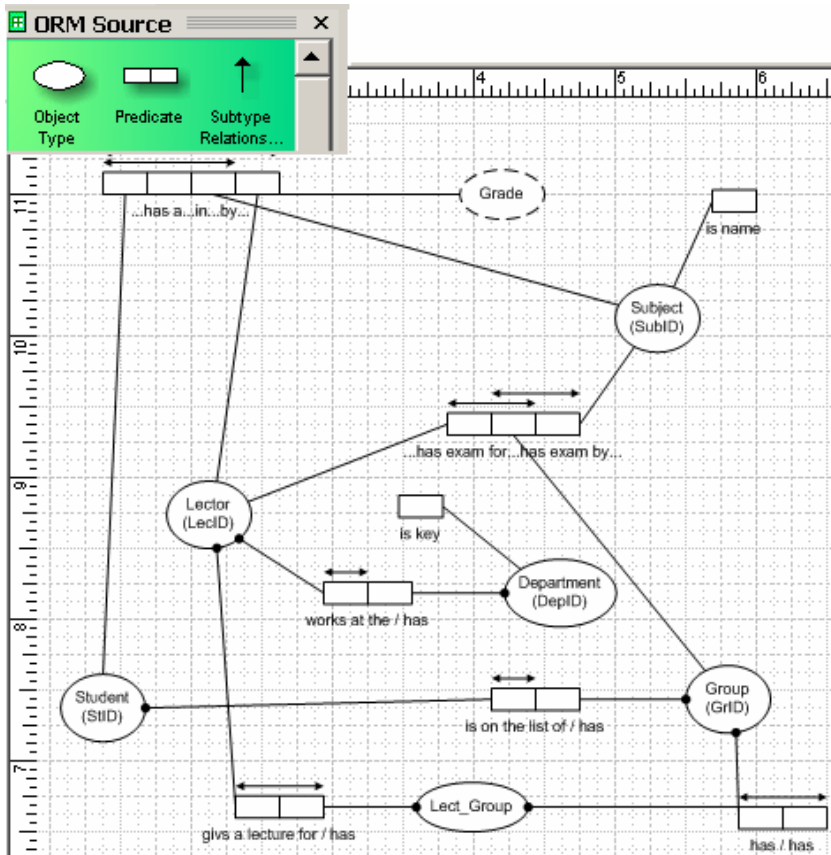
ზემოაღწერილი წესების თანახმად საჭიროა აიგოს *ORM*-დიაგრამა. ამისათვის ფაქტ-შეზღუდვები, რომლებშიც ასახულია საპრობლემო სფეროს შესახებ ცოდნა (კლასებისა და ობიექტების ძირითადი ტერმინები და ქცევის წესები), ჩვენს მიერ გადაიტანება

Ms_Visio ინსტრუმენტის საშუალებით ობიექტ-როლურ მოდელში (ნახ. 2.3-ა და ბ).

მომდევნო ეტაპზე განხორციელება ობიექტ-როლური მოდელის ავტომატური გადაყვანა არსთა-დამოკიდებულების მოდელში და, ბოლოს, მონაცემთა ბაზის სერვერში განთავსება. ამ საკითხებს ჩვენ წიგნის მომდევნო თავებში დავუბრუნდებით.



ნახ.2.3-ა. ფაქტების ფორმალური ჩაწერა



ნახ.2.3-ბ. ORM-დიაგრამა MsVisio-გარემოში

2.5. Ms SQL - ბაზისთვის .DDL ფაილის მომზადება

საუნივერსიტეტო მართვის სისტემის საპრობლემო სფეროს მაგალითზე საჭიროა მონაცემთა ისეთი განაწილებული ბაზის დაპროექტება და აგება, რომელიც მომავალში მიუერთდება ქვეყნის ერთიანი საგანმანათლებლო სისტემის მონაცემთა საცავს.

ასეთი საინფორმაციო სისტემის აგებისას აუცილებელია მიღებული საერთაშორისო სტანდარტების გათვალის-წინება, რაც მნიშვნელოვნად შეუწევობს ხელს მის შემდგომ მასშტაბირებას და განვითარებას.

პროგრამული პაკეტების შექმნის ასეთ სტანდარტს (უნიფიცირებული მოდელირების ენას, წესებს) წარმოადგენს UML-ტექნოლოგია, ხოლო მის ერთ-ერთ გავრცელებულ ინსტრუმენტს Ms_Visio.

წინამდებარე ნაშრომის მიზანია ამ ინსტრუმენტების გამოყენებით მონაცემთა ბაზის კონცეპტუალური და ლოგიკური სტრუქტურების დამუშავება, შემდგომ კი მისი ფიზიკური ორგანიზაცია, რეალიზაცია მაგალითად, SQL Server მონაცემთა ბაზების მართვის სისტემაში.

საუნივერსიტეტო მართვის სისტემის, კერძოდ მისი მონაცემთა განაწილებული ბაზის დასაპროექტებლად, როგორც ეს UML-ტექნოლოგიით მოითხოვება, საჭიროა საპრობლემო სფეროს შესწავლა და მისი ტექსტური (არაფორმალური) აღწერა, აქ არსებული მიზნების, ამოცანების, ობიექტების, ფაქტორების და სხვა ელემენტების, აგრეთვე მათი ურთიერთკავშირების დეტალური ჩამოყალიბებით.

ესაა ცოდნა სამართავი ობიექტის შესახებ, მისი სტატიკური (მდგომარეობათა სიმრავლე) და დინამიკური (ქცევათა სიმრავლე) მოდელებით.

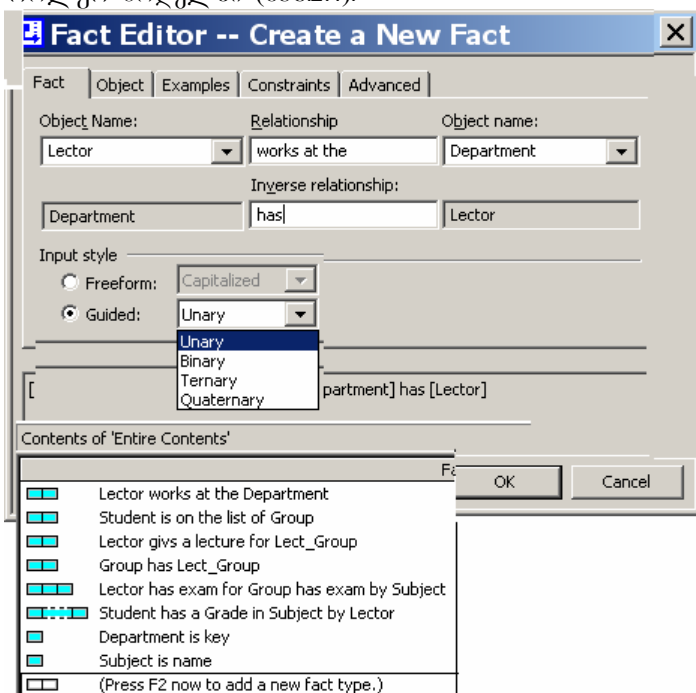
თუ ობიექტ-ორიენტირებული მოდელირების ტერმინებით ვისარგებლებთ, დასმული ამოცანის გადაწყვეტის „გასაღებს“ კლასების, ობიექტების, კლასთაშორისი კავშირების, ობიექტ-როლური და არსთა-დამოკიდებულების მოდელებისა და სხვა სახის დიაგრამების აგება წარმოადგენს.

შემდეგ კი კლასთა-ასოციაციებისა და არსთა-დამოკიდებულების დიაგრამათა საფუძველზე განხორციელდება მიზნობრივი სისტემის პროგრამული კოდების რეალიზაციის ავტომატიზებული პროცესი.

განვიხილოთ ეს საკითხები დეტალურად უნივერსიტეტის საპრობლემო სფეროს მაგალითზე.

არაფორმალზებული აღწერის ობიექტებია (ტერმინთა ლექსიკონი): ფაკულტეტები, კათედრები, სტუდენტები, ლექტორები (როგორც თანამშრომლები, ისე მოწვეულები, საშტატო და საათობრივი დატვირთვით), საგნები (აკადემიური დისციპლინები), რომლებიც იკითხება კათედრებისა და სპეციალობების მიხედვით. სასწავლო გეგმები, სილაბუსები (პროგრამები), ლექციები, პრაქტიკული და ლაბორატორიული სამუშაოები, გამოცდები, ტესტირება და ა.შ.

მეორე ეტაპზე საპრობლემო სფეროს შესახებ ცოდნა (კლასებისა და ობიექტების ძირითადი ტერმინები და ქცევის წესები) ჩვენს მიერ გადაიტანება Ms_Visio ინსტრუმენტის საშუალებით ობიექტ-როლურ მოდელში (ნახ.2.4).



ნახ. 2.4

ობიექტებისა და მათ შორის კავშირების (წესების) მომდევნო ეტაპზე განხორციელდება ობიექტ-როლური მოდელის ავტომატური გადაყვანა არსთა-დამოკიდებულების მოდელში.

ამისათვის Ms_Visio ინსტრუმენტის (ORM-source) მენიუდან გამოვიძახებთ სტრიქონს :

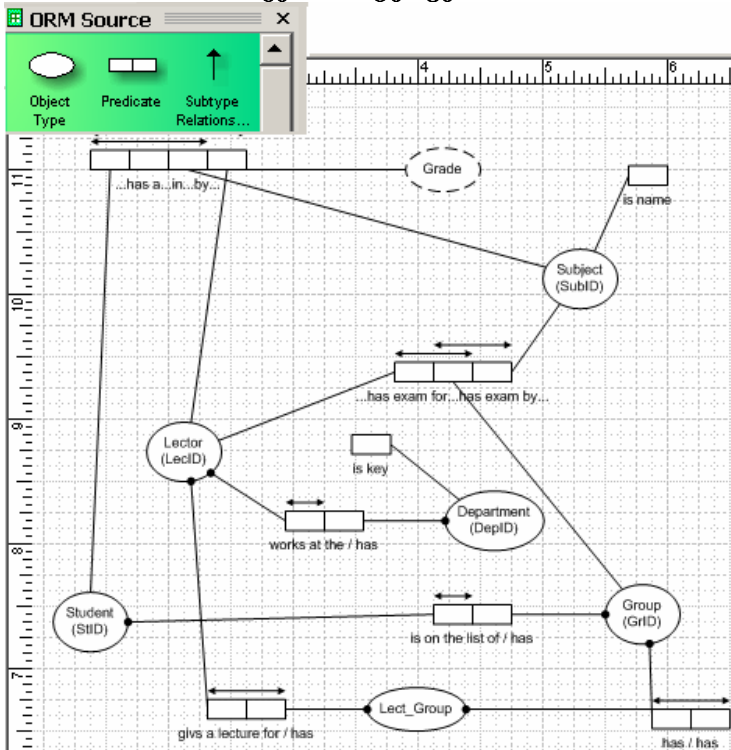
File | New | Database | Database Model Diagram

და ახალი ფანჯრის სუფთა ფურცელზე გადმოვიტანთ წინა ეტაპზე აგებულ ORM-დიაგრამის შესაბამის ER-დიაგრამას. ამისათვის აქ მენიუდან მივუთითებთ:

Database | Project | Add Existing Document

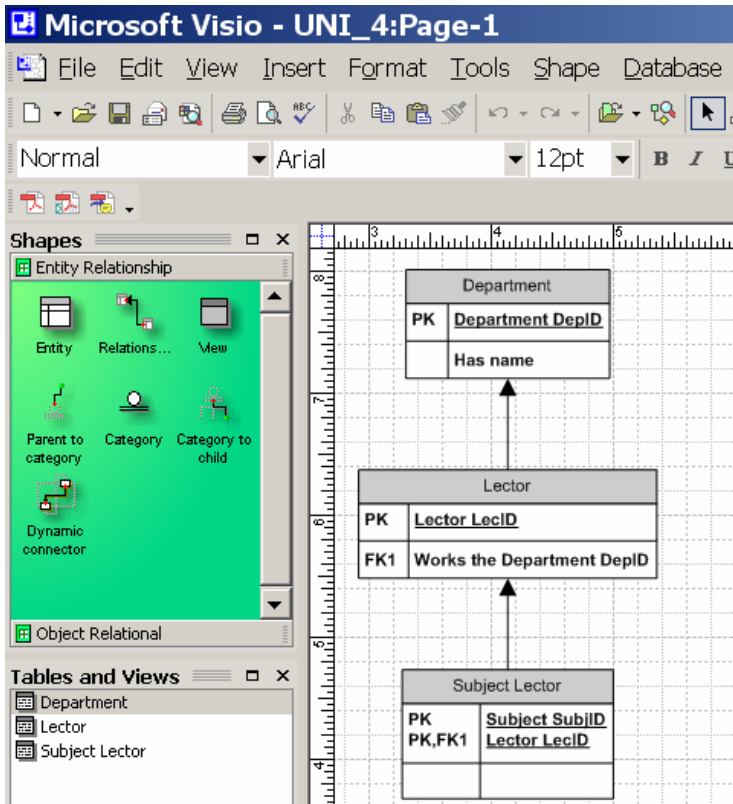
2.5 ნახაზზე ნაჩვენებია ფრაგმენტი ასეთი გზით მიღებული შედეგისათვის.

აგების ინტერფეისი



ნახ.2.5. ობიექტ-როლური მოდელის მაგალითი

2.6 ნახაზზე ჩანს სამი ცხრილი : ლექტორი, კათედრა და საგანი, აგრეთვე მათ შორის კავშირები. თუ ყველა ცხრილს გამოვიყენებთ ამ ნახაზიდან, მაშინ მივიღებთ რეალურ, მრავალცხრილურ კავშირებს.



ნახ.2.6. არსთა-დამოკიდებულების მოდელის ფრაგმენტი

მონაცემთა ბაზის აგების ავტომატიზაციის შემდეგი პროცედურა მდგომარეობს შუალედური ტექსტური ტიპის DLL–ფაილის მომზადებაში, რომელიც მომავალში SQL Server მონაცემთა ბაზების მართვის სისტემამ უნდა გამოიყენოს.

ქვემოთ მოცემულია ავტომატურად გენერირებადი DDL-ფაილის ტექსტის ფრაგმენტი.

```
/* Driver Used : Microsoft Visual Studio - Microsoft SQL Server Driver.
*/
/* Create DB_Uni database. */
use master
go
create database "DB_Uni"
go
use "DB_Uni"
go
/* Create new table "Subject Lector". */
/* "Subject Lector" : Subject has a Lector / Lector gives a Subject */
/* "Subject SubjID" : Subject has a Lector */
/* "Lector LecID" : Lector gives a Subject */
create table "Subject Lector" (
    "Subject SubjID" char(10) not null,
    "Lector LecID" char(10) not null)
go
alter table "Subject Lector"
    add constraint "Subject Lector_PK" primary key ("Subject SubjID",
    "Lector LecID")
go
/* Create new table "Lector". */
/* "Lector" : Table of Lector */
/* "Lector LecID" : Lector is identified by LecID */
/* "Works the Department DepID" : Department has Lector */
create table "Lector" (
    "Lector LecID" char(10) not null,
    "Works the Department DepID" char(10) not null)
go
alter table "Lector"
    add constraint "Lector_PK" primary key ("Lector LecID")
go
/* Create new table "Department". */
/* "Department" : Table of Department */
/* "Department DepID" : Department is identified by DepID */
/* "Has name" : Department has name */
create table "Department" (
    "Department DepID" char(10) not null,
```

```

        "Has name" bit not null)
go
alter table "Department"
    add constraint "Department_PK" primary key ("Department DepID")
go

/* Add foreign key constraints to table "Subject Lector". */
alter table "Subject Lector"
    add constraint "Lector_Subject Lector_FKI" foreign key (
        "Lector LecID")
    references "Lector" (
        "Lector LecID")
go
/* Add foreign key constraints to table "Lector". */
alter table "Lector"
    add constraint "Department_Lector_FKI" foreign key (
        "Works the Department DepID")
    references "Department" (
        "Department DepID")
go
/* This is the end of the Microsoft Visual Studio generated SQL DDL script. */

```

ახალი საინფორმაციო ტექნოლოგიების გამოყენებით, რომლებიც მოდელირების უნიფიცირებული ენის (UML) ინსტრუმენტებზეა ორიენტირებული, შესაძლებელია მოდელირებისა და დაპროგრამების პროცესების ავტომატიზაცია, რაც საბოლოო ჯამში შესაძლებელს ხდის მონაცემთა ბაზების მართვის სისტემების (მაგალითად, Ms SQL Server, SyBase, Oracle, ADO.NET) და დაპროგრამების საინტერფეისო ენების (მაგალითად, C#.NET, VB.NET) ინტეგრირებული გამოყენებით ვიზუალური მოდელირების პაკეტებთან ერთად (მაგალითად, Ms Visio, Rational Rose, Paradigm Plus) დაპროექტდეს და რეალიზებულ იქნას ხარისხიანად და სწრაფად მართვის ავტომატიზებული სისტემების Windows- და Web-დანართები.

2.6. ავტომატიზებული ანალიზის სისტემის დაპროექტება სატენდერო კომისიის ექსპერტებისათვის

აღწერილია სატენდერო პროცესის ხელშემწყობი კომპიუტერული სისტემა, რომლის მიზანია სატენდერო კომისიის ექსპერტებისათვის აუცილებელი საინფორმაციო ბაზისა და მისი ოპერატიული ავტომატიზებული ანალიზის ჩატარების განხორციელება. კონცეპტუალური მოდელი დაპროექტებულია *ORM*-დიაგრამისა და მისი შესაბამისი *ER*-მოდელის საშუალებით. მონაცემთა ბაზა რეალიზებულია *Ms SQL* სერვერზე.

დღეს, მეტად აქტუალური გახდა ტენდერის ჩატარება ნებისმიერ სფეროში, მშენებლობა იქნება ეს, სარემონტო სამუშაოები, მომარაგება, ამა თუ იმ პროდუქციის შესყიდვა, თუ რომელიმე სხვა პროექტის განხორციელება. ტენდერს აცხადებს ორგანიზაცია, რომელსაც სურს გარკვეული მოცულობის ამა თუ იმ სამუშაოს შესრულება. ფირმები, რომელთაც სურს ამ სამუშაოს შესრულება, აკეთებენ ოფიციალურ განაცხადს ტენდერში მონაწილეობის მისაღებად.

უმეტესად ასეთი საქმიანობით სხვადასხვა ტიპის სპეციალიზებული ფირმებია დაინტერესებული. ტენდერში მონაწილეობის მისაღებად აუცილებელია საჭირო იურიდიული დოკუმენტაციის შეკრება სხვადასხვა დაწესებულებებიდან, ასევე აუცილებელია სატენდერო მოსაკრებლის გადახდა ტენდერის მოთხოვნებში მითითებულ ბანკის ანგარიშზე. ტენდერში მონაწილეობის მიღების მიზნით შესასრულებელი სამუშაო დეტალურად აღწერილია *UML Activity*-დიაგრამაზე (ნახ.2.7).

კონცეპტუალური მოდელის დაპროექტებისათვის გამოყენებულია ობიექტ-როლური მოდელირება (*ORM*), რომელიც კონცეპტუალური მოდელირების განვითარებულ ტექნიკას წარმოადგენს. ობიექტ-როლური მოდელირება მიახლოებულია ბუნებრივ სალაპარაკო ენასთან. ესაა მოდელირება ფაქტების საფუძველზე, სადაც საპრობლემო არე განიხილება, როგორც ობიექტების ერთობლიობა, რომლებიც თამაშობს გარკვეულ „როლებს“. ობიექტ-როლური მოდელირების მეთოდი აქტუალურია და ფართოდ გამოიყენება საზღვარგარეთაც. აღნიშნული

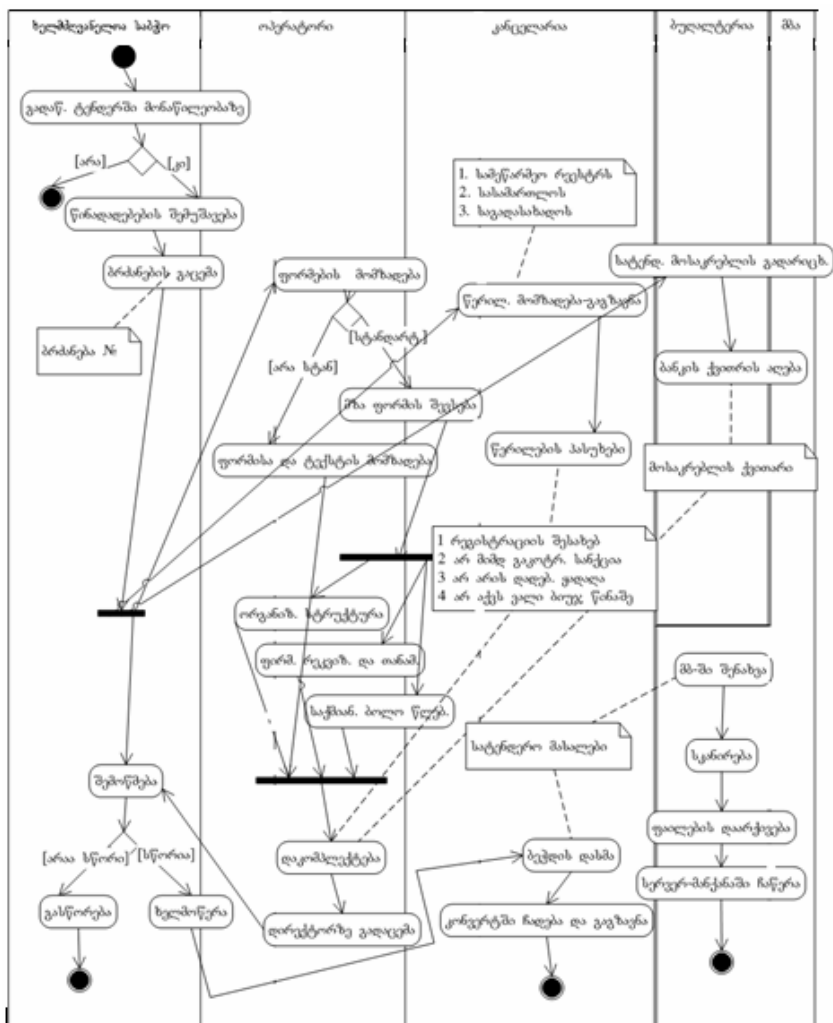
ინსტრუმენტული საშუალებანი ემსახურება მონაცემთა ბაზის დაპროექტების ავტომატიზაციას.

UML-ტექნოლოგიის საფუძველზე პროგრამული პროდუქტების შექმნა მოითხოვს საკვლევი ობიექტის მოთხოვნილებათა განსაზღვრის, ობიექტ-ორიენტირებული ანალიზის (ოო), ოო-დაპროექტებისა და რეალიზაციის (ოო-დაპროგრამების) ეტაპების განხორციელებას.

ობიექტ-როლური მოდელირების მეთოდი გამოიყენება საგნობრივი სფეროს კვლევის ეტაპზე და ახორციელებს ექსპერტების (დამპროექტებლების) მიერ გარკვეული ცოდნის, ფაქტების ფიქსირებას, რომლებიც აუცილებელია ORM-დიაგრამის (ობიექტ-როლური მოდელის) ასაგებად.

სატენდერო ამოცანების გადაწყვეტის ხელშემწყობი კომპიუტერული სისტემის აგების მიზნით შეიძლება შემდეგი ზოგადი ფაქტების ჩამოყალიბება, რაც შეესაბამება 2.7 ნახაზზე მოცემულ აქტიურობათა დიაგრამას:

„არსებობს ფირმა რომელიც აცხადებს ტენდერს, აგრეთვე ფირმა (ფირმები), რომელიც მონაწილეობს ტენდერში, და ბოლოს - გამარჯვებული ფირმა. ფირმას აქვს რეკვიზიტები და ჰყავს მენეჯერი, აქვს საკუთარი ანგარიშის ნომერი ბანკში. ტენდერში მონაწილეობის მისაღებად საჭირო დოკუმენტები თავსდება კონვერტში, ილუქება და იგზავნება ტენდერის მოთხოვნებში მითითებულ მისამართზე. აქვე მითითებულია ტენდერის ჩატარების თარიღი და დრო. ტენდერის ჩატარების დროს ყველა მონაწილე ორგანიზაციის წარმომადგენელი, ვალდებულია გამოცხადდეს მითითებულ მისამართზე. სატენდერო კომისიის წევრები საჯაროდ გახსნიან დალუქულ კონვერტებს, შეამოწმებენ იურიდიულად საჭირო დოკუმენტაციის არსებობას და შეადგენენ ოქმს ტენდერის დაწყებისა და მასში მონაწილე ორგანიზაციების შესახებ.



ნახ.2.7

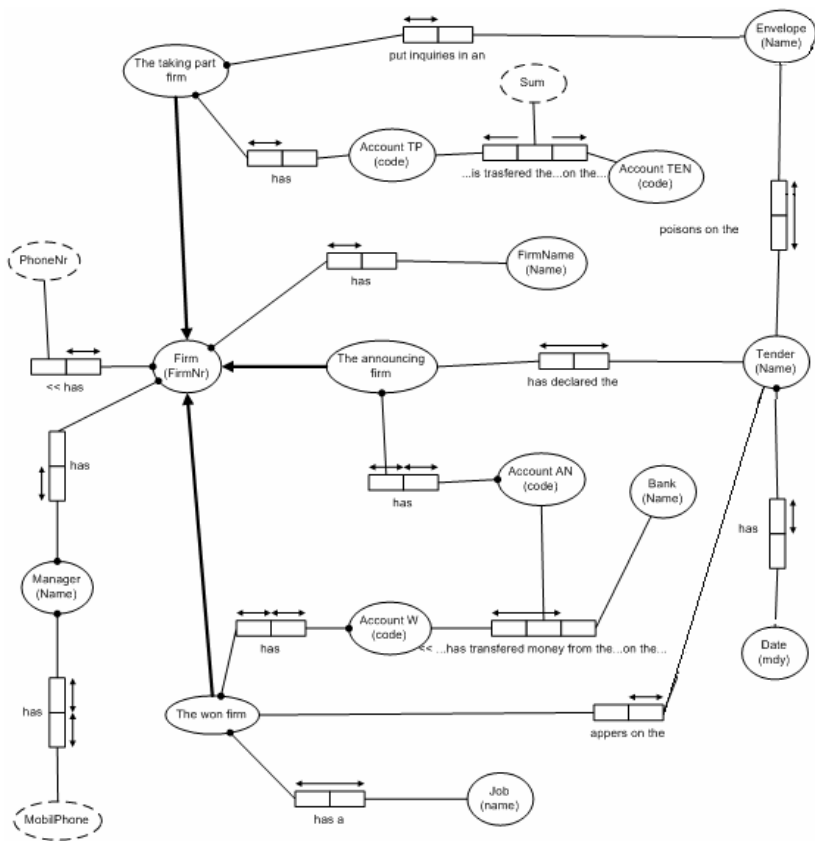
თუ დოკუმენტაციას აკლია რომელიმე აუცილებელი დოკუმენტი, ან არასწორადაა წარმოდგენილი, ასეთი ფირმა მოიხსნება ტენდერიდან. მომდევნო პერიოდში ტენდერის კომისიის წევრები დეტალურად გაეცნობიან შემოთავაზებულ წინადადებებს, გადაამოწმებენ საბუთებს, მსჯელობის შემდეგ ამოირჩევენ საუკეთესო (მისაღებ) წინადადებას და შემდგომ დაასახელებენ ტენდერში გამარჯვებულ ორგანიზაციას. დამარცხებულ ფირმებს უფლება აქვს სასამართლოში გაასაჩივროს ტენდერის ჩამტარებული ფირმის გადაწყვეტილება, რაც ხდება შემდგომი იურიდიული დავის საგანი. გამარჯვებული ფირმის ანგარიშზე გადაირიცხება სამუშაოს დაწყებისათვის აუცილებელი თანხა და ა.შ.“

ასეთი არაფორმალური აღწერიდან ფორმალურიზებულზე გადასასვლელად არის საჭირო სწორედ ფაქტების გამოკვეთა, რომლებიც შემდგომ ობიექტ-როლური დიაგრამისთვის გამოგვადგება. 2.8 ნახაზზე მოცემულია „ტენდერის ჩატარების“ პროცესის ფაქტების გადატანა ORM-დიაგრამაზე.

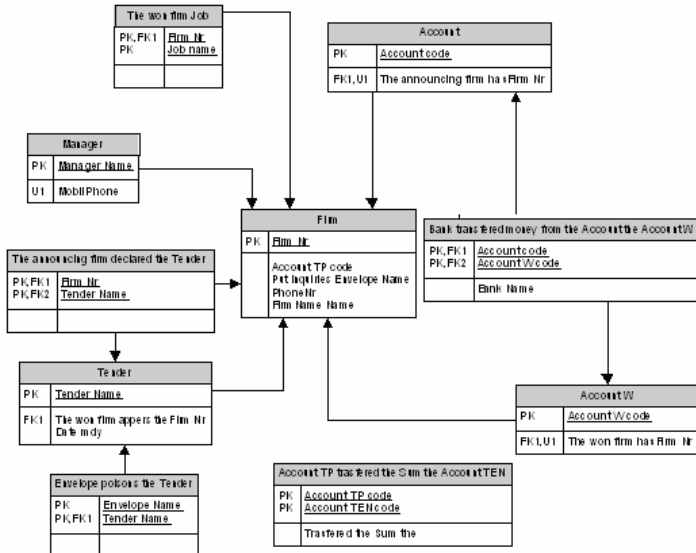
Ms Studio.Net პროგრამული პაკეტი, კერძოდ Ms Visio საშუალებას გვაძლევს ORM-დიაგრამიდან ავტომატურად ავაგოთ საპრობლემო სფეროს კონცეპტუალური მოდელი, ER-დიაგრამა, რომელიც ჩვენი შემთხვევისთვის მოცემულია 2.9 ნახაზზე.

ERM (Entyty Relation Model)–ის საფუძველზეც აიგება რელაციური მონაცემთა ბაზების ლოგიკური სტრუქტურა. მონაცემთა ბაზის დაპროექტების მიზნით ვიყენებთ SQL-სერვერს.

2.10 ნახაზზე მოცემულია ტენდერის ჩატარების ხელშემწყობი კომპიუტერული სისტემის TENDER-მონაცემთა ბაზის ცხრილების ფრაგმენტი.



656.2.8



6sb.2.9

SQL Server Enterprise Manager

File Window Help

Design Table 'Firm participating in the tender' in 'tender'

Column Name	Data Type	Length	Allow Nulls
firmcode	int	4	
saxeli	char	10	✓
qalaqi	char	10	✓
misamarTi	char	53	✓
tel	char	10	✓
Faqsi	char	19	✓

Design Table 'List of firm' in 'tender' on 'USER-E0AD098E2'

Column Name	Data Type	Length	Allow Nulls
firmcode	int	4	✓
dasaxeleba	char	10	✓
fasi	money	8	✓

Design Table 'Prize list' in 'tender' on 'USER-E0AD098E2F'

Column Name	Data Type	Length	Allow Nulls
dasaxeleba	char	10	
modeli	char	10	
raodenoba	char	10	✓
fasi	money	8	✓

6sb.2.10

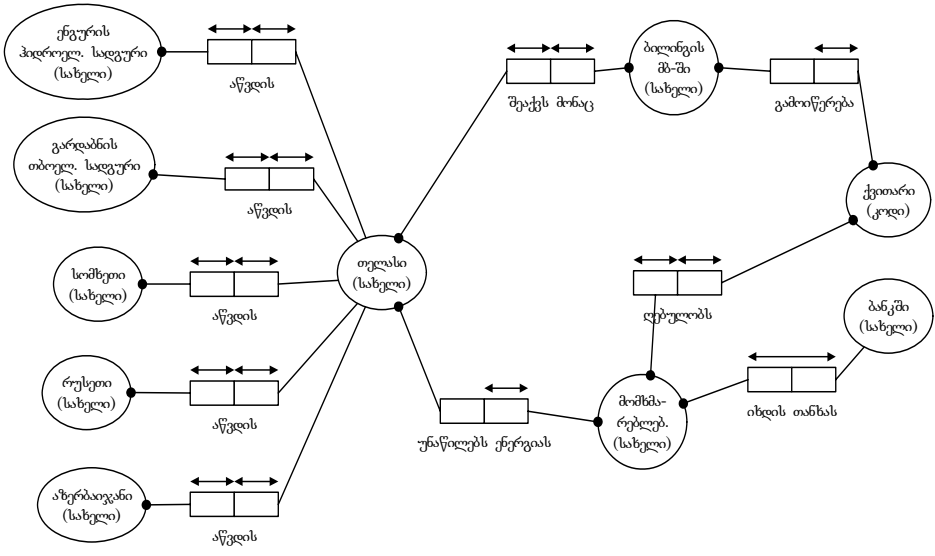
2.7. ენერგოსისტემის ობიექტ-როლური მოდელი

თანამედროვე ენერგეტიკული ობიექტების ქსელი უდავოდ მიეკუთვნება რთული და დიდი სისტემების კლასს, რომლისთვისაც დამახასიათებელია ძნელადფორმალიზებადი მართვის ამოცანების, დიდი მოცულობის საინფორმაციო ბაზების და გადაწყვეტილების მიღების რთული ალგორითმების არსებობა. ყოველივე ეს კი აუცილებლად მოითხოვს ენერგეტიკაში მართვის ახალი სისტემების გამოყენებას.

ენერგოსისტემის საწარმოთა და ორგანიზაციათა მართვის ავტომატიზებული სისტემების ქსელის აგების კონცეფციამ, ერთის მხრივ, მცირე და საშუალო კერძო ბიზნესის სექტორის გაფართოების გამო, და მეორეს მხრივ, მკაცრი კონკურენციის პირობებში მათი მოქმედების არეალის საერთაშორისო ბიზნესის მოთხოვნების ჩარჩოებში მოქცევამ, მეტად მნიშვნელოვანი ფუნქცია შეიძინა. განსაკუთრებით საყურადღებოა საქმიანი პროცესების ტექნოლოგიური ციკლების თანმხლები საინფორმაციო ნაკადების მართვის მექანიზმების დახვეწა, უნიფიცირებული დოკუმენტებისა და დოკუმენტბრუნვის პრობლემათა გათვალისწინებით, ახალი საკანონმდებლო აქტებისა და ინტერნეტ-ინტრანეტის პირობებში.

ჩვენი მიზანია ორგანიზაციული და ბიზნესის მართვის განაწილებული ენერგოსისტემის ოფის-ობიექტებისათვის საქმიანი პროცესების წარმოების უნიფიცირებული დოკუმენტბრუნვის ტექნოლოგიის შემუშავება და პროგრამული რეალიზაცია, თანამედროვე ობიექტ-ორიენტირებული დაპროგრამების მეთოდებისა და კლიენტ-სერვერული არქიტექტურის საფუძველზე.

2.11 ნახაზზე მოცემულია განაწილებული ენერგოობიექტების ზოგადი ORM-დიაგრამის ფრაგმენტი, რომელიც აერთიანებს ელენერგიის მწარმოებლებს, მის დამკვეთებს (მაგ., თელასი) და საბოლოო აბონენტებს (იურიდიულ და ფიზიკურ მომხმარებლებს).



ნახ.2.11. ORM მოდელის ფრაგმენტი

ენერგეტიკის ოფის-ობიექტებზე საქმიანი პროცესების თანმხლები უნიფიცირებული და ინდივიდუალური დოკუმენტაციისა და დოკუმენტბრუნვის სისტემის დიაგნოსტიკური ანალიზი, ფუნქციურ განყოფილებათა ურთიერთკავშირის სტრუქტურებისა და მართვის პროცესში მონაწილე პირთა „როლების“ გათვალისწინებით.

მოცემულ პარაგრაფში დამუშავებულია განაწილებული ენერგეტიკული ოფის-სისტემის მონაცემთა ბაზის ფიზიკური სტრუქტურები, მათი მოდიფიკაციის ალგორითმები და ადმინისტრირების ფუნქციები, კლიენტ-სერვერული არქიტექტურის გამოყენებით. შემოთავაზებულია ენერგეტიკული ოფის-სისტემის მომხმარებელთა ინტერფეისები, მათი როლების შესაბამისად.

მართვის ავტომატიზებული სისტემის ამოცანები განიხილება განაწილებული ენერგოსისტემის მთლიან სპექტრში, დაწყებული, მაგალითად, ენგურჰესის ჰიდროელექტროსადგურზე ელენერგის გამომშავების აღრიცხვიდან, დამთავრებული თელასის სააბონენტო გადასახადების აღრიცხვისა და კონტროლის ამოცანებით, ინტერნეტული მოხმარებისათვის.

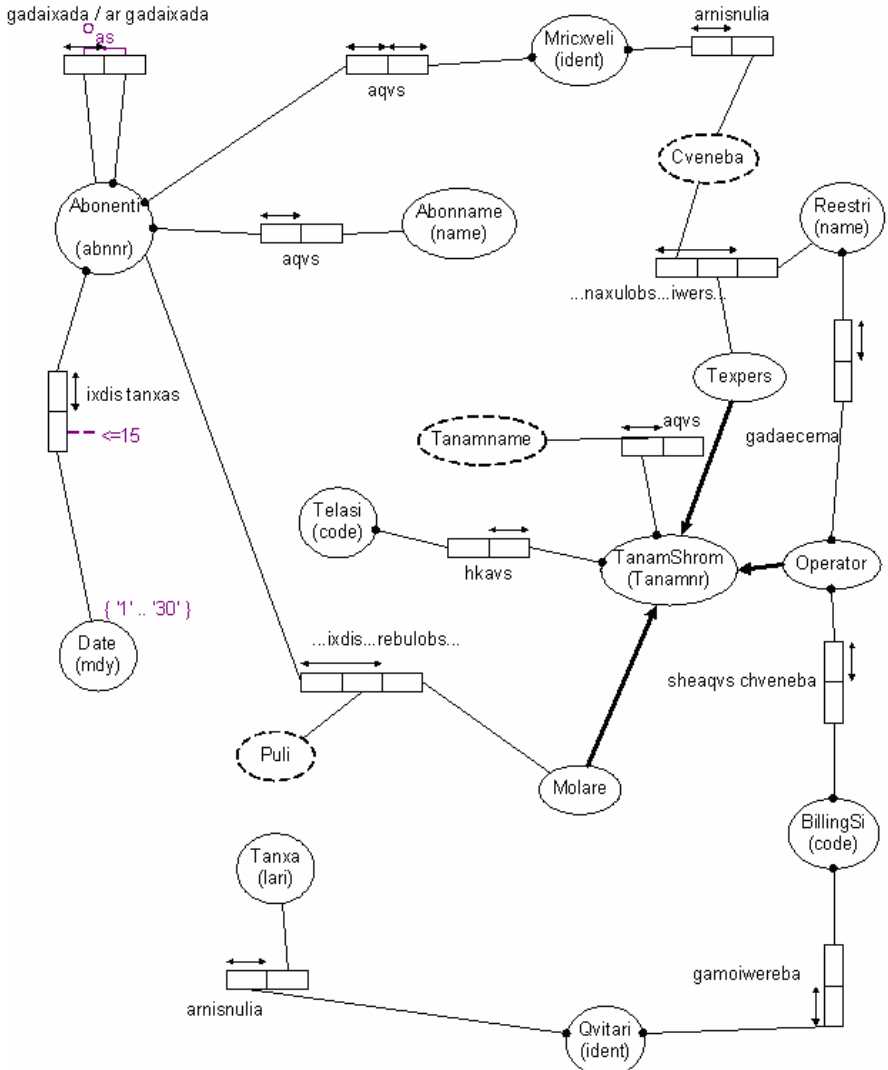
2.8. ენერგობიექტების კონტროლის სისტემის მონაცემთა ბაზის ავტომატიზებული აგება

განხილულია ენერგეტიკის სფეროში მომუშავე პერსონალისათვის მონაცემთა ბაზების ავტომატიზებული დაპროექტების ამოცანა. მისი გადაწყვეტა საშუალებას აძლევს ენერგეტიკოს-სპეციალისტს მონაცემთა ბაზის სტრუქტურა ააგოს კომპიუტერის დახმარებით საკუთარი ცოდნისა და გამოცდილების საფუძველზე ენერგო-სფეროში.

მოდელირების პროცესის წარმართვისათვის მას ესაჭიროება კონკრეტული ამოცანის ირგვლივ გარკვეული არსებული კანონზომიერებების (ფაქტების) ტექსტური ფორმით ჩაწერა. მაგალითად:

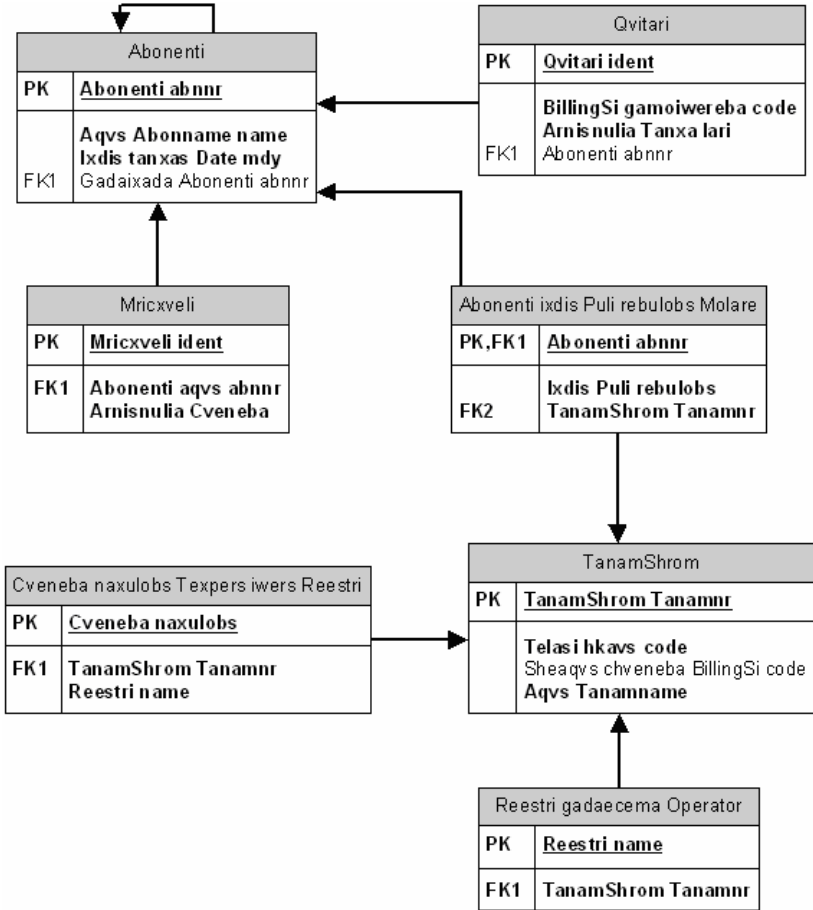
- f1 - აბონენტს აქვს გვარი;
 - f2 - აბონენტს აქვს მრიცხველი;
 - f3 - აბონენტს იხდის თანხას ყოველი თვის 15 რიცხვამდე;
 - f4 - მრიცხველზე აღნიშნულია მოხმარებული ელ.ენერგია კვტ/სთ;
 - f5 - არსებობენ აბონენტები, რომლებმაც გადაიხადეს ან არ გადაიხადეს მოხმარებული ელ.ენერგიის საფასური;
 - f6 - თელასში მუშაობენ თანამშრომლები;
 - f7 - თანამშრომელს აქვს გვარი;
 - f8 - ოპერტორი არის თანამშრომელი;
 - f9 - ტექნიკური პერსონალი არის თანამშრომელი;
 - f10 - მოლარე არის თანამშრომელი;
 - f11 - ტექნიკური პერსონალი აღნიშნავს აბონენტთა მრიცხველების ჩვენებებს რეესტრში;
 - f12 - რეესტრში არსებული მონაცემები ოპერატორს შეაქვს ბილინგის მონაცემთა ბაზაში;
 - f13 - ბილინგის მონაცემთა ბაზიდან გამოიწერება ქვითარი;
 - f14 - ქვითარზე აღნიშნულია თანხა ;
 - f15 - ქვითარს ღებულობს აბონენტი ;
 - f16 - აბონენტი იხდის ფულს მოლარესთან ;
- და ა.შ.

ზემოთ ჩამოთვლილი ფაქტების შესაბამისი *ORM (Object-Role Model)*–დიაგრამა, რომელიც აიგება *MsVisio* მოდელირების ინსტრუმენტით, მოცემულია 2.12 ნახაზზე. რაც მეტი ფაქტია აღწერილი, მით უფრო ზუსტია ობიექტის საშედეგო სემანტიკური (კონცეპტუალური) მოდელი.



ნახ.2.12. ORM-დიაგრამის ფრაგმენტი

შემდგომ ეტაპზე ORM-დიაგრამიდან ავტომატურად ვაპროექტებთ ERM (Entity Relationship Model) დიაგრამას (ნახ.2.13), რომლის საფუძველზეც აიგება რელაციურ მონაცემთა ბაზების ლოგიკური სტრუქტურა.



ნახ.2.13. ERM-დიაგრამის ფრაგმენტი

მესამე ეტაპზე ER-მოდელის საფუძველზე ვახდენთ პროგრამული სახით მონაცემთა აღწერის .DDL ფაილის ავტომატურ გენერირებას (ნახ.2.14).

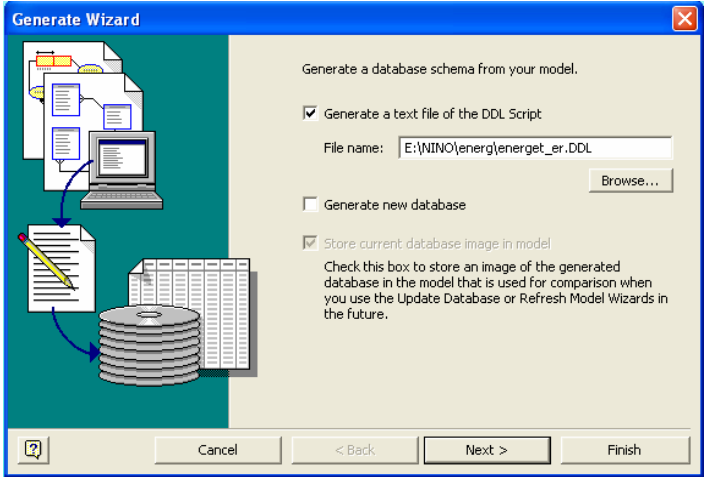
ქვემოთ მოცემულია .DDL ფაილის კოდის ფრაგმენტი:

```
/* Create new table "Reestri gadaecema Operator". */
/* "Reestri gadaecema Operator" : Table of Reestri gadaecema Operator */
/* "Reestri name" : Reestri identifies Reestri gadaecema Operator */
/* Role two (Reestri) of fact: Reestri gadaecema Operator is identified by */
/* Role two (Reestriname) of fact: Reestri is identified by */
/* "TanamShrom Tanamn" : TanamShrom Operator has Reestri gadaecema
Operator */

create table
    "Reestri gadaecema Operator" ("Reestri name" char(10) not null,
    "TanamShrom Tanamn" char(10) not null)
go
alter table "Reestri gadaecema Operator"
add constraint "Reestri gadaecema Operator_PK" primary key ("Reestri
name")

go
create table "Abonenti ixdis Puli rebulobs Molare" (
    "Abonenti abn" char(10) not null,
    "Ixdis Puli rebulobs" char(10) not null,
    "TanamShrom Tanamn" char(10) not null)
go
alter table "Abonenti ixdis Puli rebulobs Molare"
add constraint "Abonenti ixdis Puli rebulobs Molare_PK" primary
key ("Abonenti abn")

go
.....
```



ნახ.2.14

ამგვარად, კომპიუტერულ სისტემაში მონაცემთა ბაზების სტრუქტურები და ცხრილთაშორისი კავშირები მზადაა კონკრეტული მონაცემების შესატანად. აუცილებლობის შემთხვევაში შესაძლებელია მიღებული მონაცემთა ბაზის სტრუქტურის მანუალური მოდიფიკაცია.

ლიტერატურა

1. Booch G.,Rumbaugh J., Jacobson I., *The Unified Modeling Language User Guide*. Addison Wesley Longman, 2004.
2. სურგულაძე გ., კელეკინდი ჰ., თოფურია ნ. *განაწილებული ოფის-სისტემების მონაცემთა ბაზების დაპროექტება და რეალიზაცია UML-ტექნოლოგიით*. მონოგრაფია. სტუ, თბილისი. 2006.
3. Halpin T.A., *Information Modeling and relational Databases*, Morgan Kaufmann Publishers, San Francisco, 2001. www.mkp.com/books_catalog/catalog.asp/ISBN=1-55860-672-6.
4. Halpin, T.A., *Microsoft's new database modeling tool: Part 8*, www.orm.net

5. Ovchinnikov V., Vahromeev Y. *A Declarative Concept-Based Query Language as a mean for Relational Database Querying*, *Journal of Conceptual Modeling* – www.inconcept.com/jcm, 2005.

6. სურგულაძე გ., დოლიძე თ., ყვავაძე ლ. *კომპონენტურ-ვიზუალური დაპროგრამება*. სტუ. თბილისი, 2006.

7. სურგულაძე გ., შონია ო., ყვავაძე ლ. *მონაცემთა ბაზების მართვის სისტემები: Ms Access, SQL Server, InterBase, Oracle, Corba*. სტუ, თბილისი, 2004.

8. სურგულაძე გ., თოფურია ნ., ილიდიზი ი. *კათედრის მართვის ავტომატიზებული სისტემის დაპროექტება და აგება UML-ტექნოლოგიის გამოყენებით*. სტუ-შრ.კრ. №4(437), 2001.

9. სურგულაძე გ. *დაპროგრამების ვიზუალური მეთოდები და ინსტრუმენტები: UML, Ms VISIO, C++Builder*. სტუ, თბ., 2005.

10. Николаишвили В., Сургуладзе Г. Топурия Н., Кашибадзе М. *Категориальный подход разработки абстрактных моделей данных для объектно-ориентированных, реляционных баз данных*. Тез. Докл. Интерн. Конф. Киев, 2006.

11. ყიფშიძე მ, ჯამარაშვილი ვ., არაბიძე გ. *საქართველოს სათბობ-ენერგეტიკული კომპლექსის პრობლემები და განვითარების პერსპექტივები*. სტუ, თბ., 2004.

12. არაბიძე ვ., სურგულაძე გ., თოფურია ნ. *ელექტროენერჯის განაწილების მართვის ავტომატიზებული სისტემის აგება ობიექტ-როლური მოდელირების საფუძველზე*. სტუ შრ.კრ. №4(490), 2006.

13. თოფურია ნ. *ელექტროენერჯის ტექნიკური დანაკარგების კონტროლის კომპიუტერული სისტემის აგება ORM/ERM ბაზაზე*. სტუ შრ.კრ., მას“ №1(2), 2007.

14. სურგულაძე გ., თოფურია ნ., ყვავაძე ლ. *განაწილებული მონაცემთა ბაზების აგების ავტომატიზაცია .NET ვარემოში*. სტუ შრ.კრ., მას“ №1(2), 2007.

15. გოციანიშვილი ვ., ფრანგიშვილი ა., სურგულაძე გ. *ინფორმატიკა, პროგრამული ტექნოლოგიები და მათი განვითარების და სწავლების თანამედროვე მიმართულებანი*. სტუ შრ.კრ., მას“ №1(2), 2007.