

## მონაცემთა ბაზის ტრიგერების დაპროექტება ნავთობკომპანიების ქსელისათვის

ნინო თოფურია - საქართველოს ტექნიკური უნივერსიტეტი,  
ელზა ბიწაძე, ზურაბ ხურცია - აკაკი წერეთლის სახელმწიფო  
უნივერსიტეტი

### რეზიუმე

განხილულია ნავთობკომპანიების ქსელის საპრობლემოს მონაცემთა ბაზის დაპროექტებისა და ტრიგერების მუშაობის პრინციპები. მონაცემთა ბაზის ER-მოდელის დასაპროექტებად შერჩეულია CASE\_Studio2, რომელიც საშუალებას მისცემს აპლიკაციის დამპროექტებელს თვალსაჩინოდ შეისწავლოს არსებული საინფორმაციო სისტემა, შეცვალოს იგი ახალი ამოცანების შესაბამისად. შემოთავაზებულია მონაცემთა ბაზის ტრიგერები, რომლებიც გამოიყენება სხვადასხვა სახის მოვლენების დასამუშავებლად და მონაცემთა მთლიანობის დაცვის უზრუნველსაყოფად.

**საკვანძო სიტყვები:** მონაცემთა ბაზა, ER-მოდელი. ტრიგერი. CASE ტექნოლოგია.

### 1. შესავალი

საინფორმაციო სისტემის დაპროექტების პირველი ნაბიჯია საგნობრივი არის ფორმალური აღწერა, საინფორმაციო სისტემის სრული, არაურთიერთ საწინააღმდეგო ფუნქციური და საინფორმაციო მოდელების აგება. ეს ლოგიკურად რთული, შრომატევადი და ხანგრძლივი სამუშაოა, რომელიც მასში მაღალკვალიფიციური სპეციალისტების მონაწილეობას საჭიროებს. უნდა აღინიშნოს, რომ საინფორმაციო სისტემის შექმნისა და ფუნქციონირების პროცესში მომხმარებელთა მოთხოვნები შეიძლება შეიცვალოს ან დაზუსტდეს, რაც ასეთი სისტემის დამუშავებას კიდევ უფრო ართულებს. მითითებულმა სირთულეებმა იმ პროგრამულ-ტექნოლოგიური საშუალებების სპეციალური კლასის, ე.წ. CASE-საშუალებების წარმოშობას შეუწყო ხელი, რომლებიც პროგრამული უზრუნველყოფის დამუშავების ეფექტურობის ამაღლებისთვისაა განკუთვნილი [1,2].

თანამედროვე CASE საშუალებათა უმეტესობა შეიცავს ER-მოდელის მონაცემთა აღწერის ინსტრუმენტებს, ამას გარდა შემუშავებულია მონაცემთა ბაზის ავტომატური გარდაქმნის პროცედურა ER-მოდელიდან ფიზიკურში და ფიზიკური მოდელიდან პროგრამულში (ddl), რომელიც კონკრეტულ მზმს-ს შეესაბამება. ყველა CASE-სისტემას გააჩნია მონაცემთა ბაზის შემუშავების პროცესის დოკუმენტირების გაძლიერების საშუალებები. ანგარიშების ავტომატური გენერატორები მიმდინარე პროექტის მდგომარეობის შესახებ ანგარიშის მომზადების საშუალებას იძლევიან, როგორც მონაცემთა ბაზების დეტალური აღწერისა და მათი გრაფიკული კავშირის სახეზე, ასევე მომზადებული სტანდარტული ბეჭდური ანგარიშების სახეზეც, რაც არსებითად აიოლებს პროექტის მართვას [3].

### 2. ძირითადი ნაწილი

ამჟამად, ტერმინი CASE (Computer Aided Software/System Engineering) ძალიან ფართო გაგებით გამოიყენება. პირველად კი ტერმინი CASE მხოლოდ პროგრამული უზრუნველყოფის დამუშავების ავტომატიზაციის საკითხებით შემოისაზღვრებოდა, ხოლო ახლა – მთლიანად რთული საინფორმაციო სისტემების დამუშავების პროცესებით.

CASE-საშუალებების ტერმინში იგულისხმება საინფორმაციო სისტემების დამუშავების პროგრამული საშუალებები, მათ შორის ინფორმაციაზე მოთხოვნილების ანალიზი და ფორმულირება, გამოყენებითი პროგრამული უზრუნველყოფის დაპროექტება და მონაცემთა ბაზა, კოდების გენერაცია, ტესტირება, დოკუმენტირება, ხარისხის უზრუნველყოფა, პროექტის მართვა და სხვა პროცესები. CASE-საშუალებები სისტემურ პროგრამულ უზრუნველყოფასა და ტექნიკურ საშუალებებთან ერთად, საინფორმაციო სისტემების დამუშავების სრულ გარემოს ქმნის.

CASE-ტექნოლოგია საინფორმაციო სისტემის დაპროექტების მეთოდოლოგიური და ინსტრუმენტული საშუალებების ნაკრებია, რომელიც საშუალებას გვაძლევს თვალსაჩინო ფორმით მოვახდინოთ საგნობრივი მოდელირება, საინფორმაციო სისტემის დამუშავების ყველა ეტაპზე გავაანალიზოთ ეს მოდელი, დავამუშაოთ მომხმარებლის ინფორმაციული მოთხოვნილების თანახმად მიღებული შესაბამისი დანართები.

### 2.1. მონაცემთა ბაზის დაპროექტება

მონაცემთა ბაზის დასაპროექტებლად აუცილებელია კონცეპტუალური სქემის შექმნა (ER - მოდელი), რომელიც მონაცემთა ბაზის საფუძველია.

შემუშავებული სქემისთვის გენერირდება პროგრამული კოდი ცხრილების, ინდექსების, კავშირების, არსებული პროცედურების, მომხმარებლებისა და მბმს-ს სხვა კომპონენტების სრულად შესაქმნელად. ამას გარდა, გათვალისწინებულია ER-დიაგრამათა გენერაციის შესაძლებლობაც, არსებული მონაცემთა ბაზისათვის (Reverse Engineering), მბ-სთან პირდაპირ დასაკავშირებლად, ODBC ან ADO დრაივერებით. პროგრამა ასევე აყალიბებს HTML-ის ან RTF-ის დეტალურ დოკუმენტაციას, უზრუნველყოფს DFD დიაგრამათა ექსპორტს XML ფორმატში, ასევე ახდენს ნიმუშების რედაქტირებას და სხვა ფუნქციების შესრულებას.

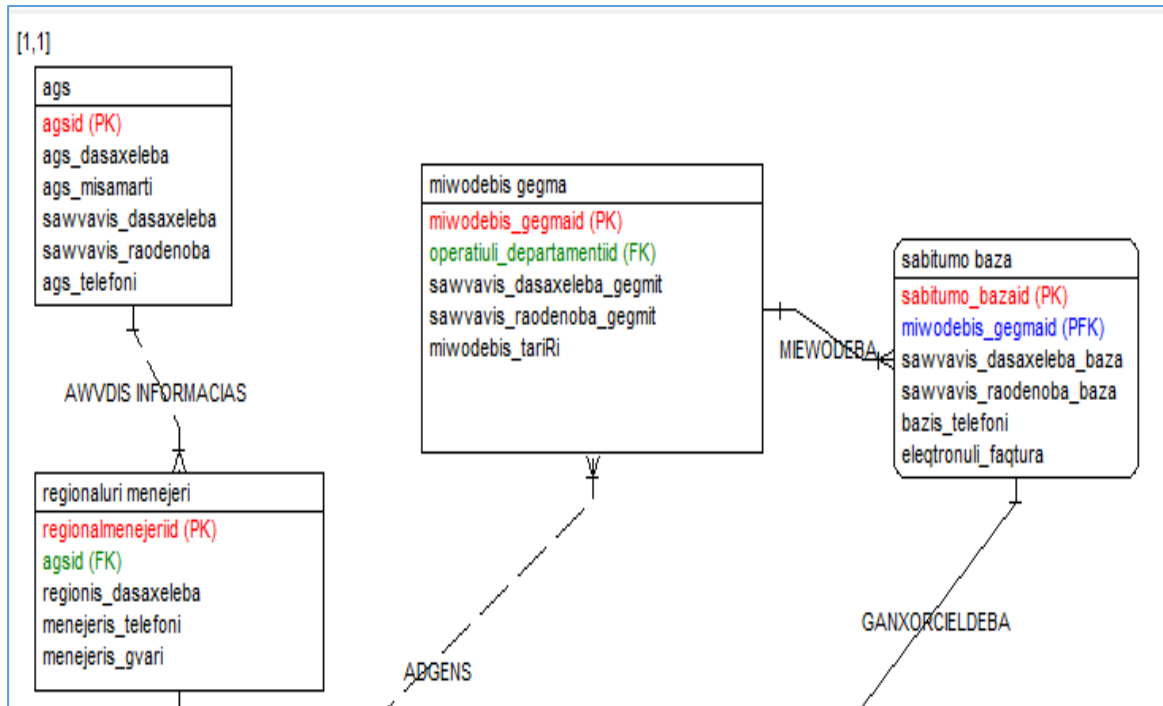
CASE\_Studio2-ს აქვს მონაცემთა ბაზის სრული ფიზიკური მოდელის აგების შესაძლებლობა, ასევე მბმს-ს ყოველი მოდელიდან ინდივიდუალური თვისებების გამოყენებით – სახეებისა და ატრიბუტთა თვისებების (სტანდარტული მბ და მომხმარებელი), გასაღებების (პირველადი და გარე) აღწერის შესაძლებლობების, კავშირების, მითითებული მთლიანობის პირობათა შესრულების, მომხმარებლების და სხვა ჯგუფების (როლების), შენახული პროცედურების აღწერის შესაძლებლობა. შემდეგში შესაძლებელი გახდება სხვა მბმს-სთვის ფიზიკური მოდელის კონვერტაციის შესრულება.

1-ელ ნახაზზე მოცემულია მონაცემთა ბაზის ER-მოდელი, დაპროექტებული CASE Studio2-ის საშუალებით ნავთობკომპანიის მაგალითზე.

### 2.2. მონაცემთა ბაზის ტრიგერები

ტრიგერი არის შენახვადი პროცედურის სპეციალური ტიპი, რომელიც სერვერის მიერ ავტომატურად გაიშვება ამა თუ იმ ცხრილზე მოქმედებების დროს. ის ანალოგიურია Visual Basic -ში მოვლენების დამუშავების პროცედურებისა.

ტრიგერები უზრუნველყოფს ბაზებში მონაცემთა მთლიანობას. მათი დახმარებით შესაძლებელია მეორადი ცხრილებიდან მონაცემთა ავტომატური წაშლა, მასთან დაკავშირებული მონაცემების წაშლისას პირველად ცხრილებში.



ნახ.1. მზ-ის არსთა დამოკიდებულებათა მოდელის ფრაგმენტი

შემოთავაზებული გვაქვს SQL Server- ის ორი სახის ტრიგერი:

1. ტრიგერები რომელიც ცხრილთან ჩატარებული მოვლენის შემდეგ სრულდება;
2. ტრიგერები რომლებიც ცხრილთან დაკავშირებით, მოვლენების ნაცვლად, სრულდება. ამ შემთხვევაში მოვლენა (ჩანაწერთა დამატება, შეცვლა ან წაშლა) არ სრულდება, მათ ნაცვლად SQL ბრძანებები შესრულდება, რომელიც ტრიგერის შიგნით მოცემულია;

ტრიგერების პირველი ტიპი გამოიყენება ცხრილის მოვლენების დამუშავებისათვის, ხოლო მეორე – მონაცემთა მთლიანობის უზრუნველყოფისათვის, ე.ი მეორადი ცხრილებიდან მონაცემთა წაშლისათვის, პირველადი ცხრილებიდან მათთან დაკავშირებული მონაცემების წაშლის დროს.

ნავთობკომპანიის მონაცემთა ბაზაში გამოყენებული ტრიგერების აღწერა მოცემულია ლისტინგებში:

### აგს დამატების ინდიკატორი

```
USE [elza3]
GO
/***** Object: Trigger [dbo].[damatebis indikatori]    Script Date: 19.04.2018
9:33:26 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Author,,Name>
-- Create date:    <Create Date,,>
-- Description:    <Description,,>
-- =====
ALTER TRIGGER [dbo].[damatebis indikatori]
```

```

        ON [dbo].[ags]
    AFTER INSERT
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for trigger here
    PRINT 'canaweri daemata'

END

```

### აგს შეცვლა

```

USE [elza3]
GO
/***** Object: Trigger [dbo].[secvli indikator]    Script Date: 19.04.2018 9:34:30
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Author,,Name>
-- Create date:    <Create Date,,>
-- Description:    <Description,,>
-- =====
ALTER TRIGGER [dbo].[secvli indikator]
    ON [dbo].[ags]
    AFTER UPDATE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for trigger here
    Print 'ჩანაწერი შეიცვალა'

END

```

### რეგიონალური მენეჯერის დამატება

```

USE [elza3]
GO
/***** Object: Trigger [dbo].[ damatebis indikator]    Script Date: 19.04.2018
9:35:12 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Author,,Name>
-- Create date:    <Create Date,,>
-- Description:    <Description,,>
-- =====
ALTER TRIGGER[dbo].[ damatebis indikator]
    ON [dbo].[regionaluri_menejeri]
    AFTER INSERT
AS
BEGIN

```

```
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

-- Insert statements for trigger here
PRINT 'canaweri daemata'
END
```

### რეგიონალური მენეჯერი შეცვლა

```
USE [elza3]
GO
/***** Object: Trigger [dbo].[Shecvlis indikator]    Script Date: 19.04.2018
9:36:37 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Author,,Name>
-- Create date:    <Create Date,,>
-- Description:    <Description,,>
-- =====
ALTER TRIGGER [dbo].[Shecvlis indikator]
ON [dbo].[regionaluri_menejeri]
AFTER UPDATE
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

-- Insert statements for trigger here
Print 'Canaweri shecvla'

END
```

### ოპერტიული დეპარტამენტი შეცვლა

```
USE [elza3]
GO
/***** Object: Trigger [dbo].[Secvlis indikator]    Script Date: 19.04.2018 9:39:10
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER TRIGGER [dbo].[Secvlis indikator]
ON [dbo].[operatiuli_departamenti]
AFTER INSERT
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.

-- Insert statements for trigger here
PRINT 'ჩანაწერის შეცვლა'
SET NOCOUNT ON;

END
```

## მოწოდების გეგმის შეცვლა

```

USE [elza3]
GO
/***** Object: Trigger [dbo].[Secvla]    Script Date: 19.04.2018 9:40:57 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Author,,Name>
-- Create date:    <Create Date,,>
-- Description:    <Description,,>
-- =====
ALTER TRIGGER [dbo].[Secvla]
  ON [dbo].[miwodebis_gegma]
  AFTER UPDATE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for trigger here
    Print 'ჩანაწერი შეიცვალა'
END

```

## საბითუმო ბაზის შეცვლა

```

USE [elza3]
GO
/***** Object: Trigger [dbo].[bazashecvla]    Script Date: 19.04.2018 9:42:55 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Author,,Name>
-- Create date:    <Create Date,,>
-- Description:    <Description,,>
-- =====
ALTER TRIGGER [dbo].[bazashecvla]
  ON [dbo].[sabitumo_baza]
  AFTER UPDATE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for trigger here
    Print 'ჩანაწერი შეიცვალა'
END

```

## 3. დასკვნა

საინფორმაციო სისტემის დამუშავების დროს ანალიზი და დაპროექტება საკმაოდ შრომატევადი პროცესია, ამიტომ CASE-ინსტრუმენტების სწორად შერჩევა განსაზღვრავს

ტექნიკური გადაწყვეტილებისა და საპროექტო დოკუმენტაციის მომზადების ხარისხს. ნავთობკომპანიის ქსელის მონაცემთა ბაზის ER-მოდელის დასაპროექტებულად შერჩეულა CASE\_Studio2, ხოლო SQL-ენაზე დაპროგრამებული მონაცემთა ბაზის ტრიგერები საშუალებას მოგცემს განვსაზღვროთ ცხილებში მონაცემების მთლიანობა, სანდოობა და მათი ურთიერთსაწინააღმდეგო შინაარსი.

#### ლიტერატურა - References - Литература:

1. Surguladze G., Turkia E., Topuria N., Lominadze T., Giutashvili M. (2012). Automation of Business-Processes of an Election System. IV-Intern.Conf. “Problems of Cybernetics and Informatics“ (PCI’ 2012). Baku, Azerbaijan.
2. Dias G.K.A. (2017). Evolvement of Computer Aided Software Engineering (CASE) Tools: A User Experience, International Journal of Computer Science and Software Engineering (IJCSSE), Volume 6, Issue 3.
3. Coronel C., Morris S., (2014). Database Systems: Design, Implementation, & Management. Cengage Learning.

### DESIGNING TRIGGERS OF DATABASES FOR THE OIL COMPANIES NETWORK

Topuria Nino - Georgian Technical University,  
Bitsadze Elza, Khurtsia Zurab – Akaki Tsereteli State  
University (Kutaisi)

#### Summary

The article reviews the principles of database design and triggers for the oil companies network. ER-model of database is designed on the bases of CASE Studio 2, which will enable project developers to examine existing information systems, replacing it with new tasks. The database triggers are used for processing various events and ensuring data integrity.

### СОЗДАНИЕ ТРИГГЕРОВ БАЗЫ ДАННЫХ ДЛЯ СЕТИ НЕФТЯНЫХ КОМПАНИЙ

Топурия Н. – Грузинский Технический Университет,  
Бицадзе Э., Хурция З. – Государственный Университет  
им. Ак. Церетели (Кутаиси)

#### Резюме

Рассматриваются вопросы проектирования баз данных и внедрение триггеров для сети нефтяных компаний. ER-модель базы данных разработана на базе CASE Studio 2, которая позволит разработчикам наглядно изучать информационные системы, заменять их в соответствии поставленных целей. Триггеры базы данных используются для обработки различных событий и обеспечения целостности данных.