Podstawy Baz Danych

System bazodanowy

Autorzy: *Jakub Jungiewicz*
Michał Zakrzewski

28 stycznia 2017

1 Założenia projektowe

Aktorzy:
- administrator
- klient prywatny
- klient grupowy
- pracownik firmy organizującej
- system

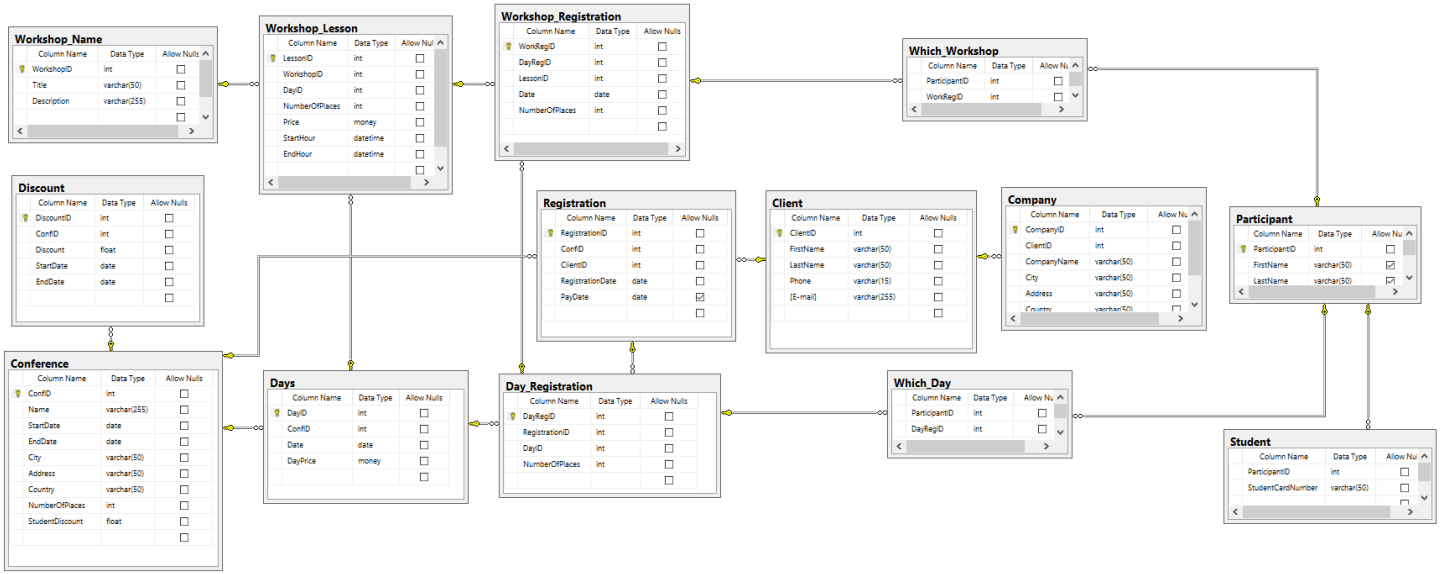
Funkcje systemu:
- rezerwacja miejsc na konferencje/warsztat - klient prywatny/grupowy
- rejestrowanie uczestników na konferencje/warsztat - klient prywatny/grupowy
- usuwanie uczestników z konferencji - klient prywatny/grupowy
- sprawdzanie ilosci wolnych miejsc na konferencji/warsztacie - klient prywatny/grupowy, pracownik firmy organizacyjnej
- sprawdzanie ceny za konferencje/warszta - klient prywatny/grupowy
- wybór dni w przypadku konferencji kilkudniowych - klient prywatny/grupowy
- wprowadzanie do systemu danych osobowych uczestników - klient prywatny/grupowy

- wgląd do statystyk dotyczących ilosci osob zapisanych na dany warsztat, konferencje - pracownik firmy organizującej
- wgląd do statystyk dotyczących dokonanych płatności - pracownik firmy organizującej
- dostęp do wszystkich widoków - pracownik firmy organizującej
- dodawanie konferencji oraz warsztatów - pracownik firmy organizującej

- przydzielenie uczestnikowi unikalnego identyfikatora imiennego (+ ew informacja o firmie) - system
- sprawdzaine czy dany uczestnik moze brać udział w danej konferencji - system
- sprawdzanie czy dany uczestnik moze brać udział w danym warsztacie - system
- sprawdzanie po 2 tyg od chwili rejestracji czy firma podała dane uczestników i wysłanie odpowiedniej wiadomosci - system
- anulowanie rezerwacji tydzien po niedokonaniu opłaty - system
- ustalanie wysokości opłaty w zależności od zniżek studenckich i/lub czasowych - system
- generowanie raportów list uczestników na każdy dzien i warsztat konferencji - system
- generowanie raportów z informacjami o płatnościach klientów - system
- generowanie raportów o użytkownikach, którzy najcześciej uczestniczą w konferencjach - system
- blokowanie zapisów na konferencje, które przekroczyly limit zapisów - system
- blokowanie zapisów na warsztaty, które przekroczyły limit zapisów - system
- blokowanie zapisów na warsztaty klientów, którzy nie sa zapisani na dany dzień konferencji - system

Administrator posiada pełny dostęp do wszystkich procedur i widoków.

2 Diagram


Workshop_Name
Column Name Data Type Allow Nulls
WorkshopID int
Title varchar(50)
Description varchar(255)
Workshop_Lesson
Column Name Data Type Allow Nul
LessonID int
WorkshopID int
DayID int
NumberOfPlaces int
Price money
StartHour datetime
EndHour datetime
Workshop_Registration
Column Name Data Type Allow Nulls
WorkRegID int
DayRegID int
LessonID int
Date date
NumberOfPlaces int
Which_Workshop
Column Name Data Type Allow Nu
ParticipantID int
WorkRegID int
Discount
Column Name Data Type Allow Nulls
DiscountID int
ConfID int
Discount float
StartDate date
EndDate date
Registration
Column Name Data Type Allow Nulls
RegistrationID int
ConfID int
ClientID int
RegistrationDate date
PayDate date
Client
Column Name Data Type Allow Nulls
ClientID int
FirstName varchar(50)
LastName varchar(50)
Phone varchar(15)
[E-mail] varchar(255)
Company
Column Name Data Type Allow Nu
CompanyID int
ClientID int
CompanyName varchar(50)
City varchar(50)
Address varchar(50)
Participant
Column Name Data Type Allow Nu
ParticipantID int
FirstName varchar(50)
LastName varchar(50)
Conference
Column Name Data Type Allow Nulls
ConfID int
Name varchar(255)
StartDate date
EndDate date
City varchar(50)
Address varchar(50)
Country varchar(50)
NumberOfPlaces int
StudentDiscount float
Days
Column Name Data Type Allow Nulls
DayID int
ConfID int
Date date
DayPrice money
Day_Registration
Column Name Data Type Allow Nulls
DayRegID int
RegistrationID int
DayID int
NumberOfPlaces int
Which_Day
Column Name Data Type Allow Nu
ParticipantID int
DayRegID int
Student
Column Name Data Type Allow Nu
ParticipantID int
StudentCardNumber varchar(50)

3 Tabele

3.1 dbo.Client

```
USE [zakrzews_a]
GO

/****** Object:  Table [dbo].[Client]    Script Date:
    2017-01-25 23:28:33 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

SET ANSI_PADDING ON
GO

CREATE TABLE [dbo].[Client](
        [ClientID] [int] IDENTITY(1,1) NOT NULL,
        [FirstName] [varchar](50) NOT NULL,
        [LastName] [varchar](50) NOT NULL,
        [Phone] [varchar](15) NOT NULL,
        [E-mail] [varchar](255) NOT NULL,
 CONSTRAINT [PK_Client] PRIMARY KEY CLUSTERED
(
        [ClientID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

SET ANSI_PADDING OFF
GO
```

3.2 dbo.Company

```
USE [zakrzews_a]
GO

/****** Object:  Table [dbo].[Company]    Script Date:
    2017-01-25 23:33:49 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

SET ANSI_PADDING ON
GO

CREATE TABLE [dbo].[Company](
        [CompanyID] [int] IDENTITY(1,1) NOT NULL,
        [ClientID] [int] NOT NULL,
        [CompanyName] [varchar](50) NOT NULL,
        [City] [varchar](50) NOT NULL,
        [Address] [varchar](50) NOT NULL,
        [Country] [varchar](50) NOT NULL,
 CONSTRAINT [PK_Company] PRIMARY KEY CLUSTERED
(
        [CompanyID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

SET ANSI_PADDING OFF
GO

ALTER TABLE [dbo].[Company]  WITH CHECK ADD  CONSTRAINT [
    FK_Company_Client] FOREIGN KEY([ClientID])
REFERENCES [dbo].[Client] ([ClientID])
GO

ALTER TABLE [dbo].[Company] CHECK CONSTRAINT [
    FK_Company_Client]
GO
```

3.3 dbo.Conference

```
USE [zakrzews_a]
GO

/****** Object:  Table [dbo].[Conference]    Script Date:
    2017-01-25 23:34:07 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

SET ANSI_PADDING ON
GO

CREATE TABLE [dbo].[Conference](
        [ConfID] [int] IDENTITY(1,1) NOT NULL,
        [Name] [varchar](255) NOT NULL,
        [StartDate] [date] NOT NULL,
        [EndDate] [date] NOT NULL,
        [City] [varchar](50) NOT NULL,
        [Address] [varchar](50) NOT NULL,
        [Country] [varchar](50) NOT NULL,
        [NumberOfPlaces] [int] NOT NULL,
        [StudentDiscount] [float] NOT NULL,
 CONSTRAINT [PK_Conference] PRIMARY KEY CLUSTERED
(
        [ConfID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

SET ANSI_PADDING OFF
GO

ALTER TABLE [dbo].[Conference]  WITH CHECK ADD
    CONSTRAINT [CheckNumberOfPlaces_C] CHECK  (([
    NumberOfPlaces]>(0)))
GO

ALTER TABLE [dbo].[Conference] CHECK CONSTRAINT [
    CheckNumberOfPlaces_C]
```

```
GO

ALTER TABLE [dbo].[Conference] WITH CHECK ADD
    CONSTRAINT [CheckStartEnd_C] CHECK (([StartDate]<=[
    EndDate]))
GO

ALTER TABLE [dbo].[Conference] CHECK CONSTRAINT [
    CheckStartEnd_C]
GO
```

3.4 dbo.DayRegistration

```
USE [zakrzews_a]
GO

/****** Object:  Table [dbo].[Day_Registration]    Script
    Date: 2017-01-25 23:34:21 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Day_Registration](
        [DayRegID] [int] IDENTITY(1,1) NOT NULL,
        [RegistrationID] [int] NOT NULL,
        [DayID] [int] NOT NULL,
        [NumberOfPlaces] [int] NOT NULL,
 CONSTRAINT [PK_Day_Registration] PRIMARY KEY CLUSTERED
(
        [DayRegID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Day_Registration]  WITH CHECK ADD
    CONSTRAINT [FK_Day_Registration_Days] FOREIGN KEY([
    DayID])
REFERENCES [dbo].[Days] ([DayID])
GO

ALTER TABLE [dbo].[Day_Registration] CHECK CONSTRAINT [
    FK_Day_Registration_Days]
GO

ALTER TABLE [dbo].[Day_Registration]  WITH CHECK ADD
    CONSTRAINT [FK_Day_Registration_Registration] FOREIGN
    KEY([RegistrationID])
REFERENCES [dbo].[Registration] ([RegistrationID])
GO

ALTER TABLE [dbo].[Day_Registration] CHECK CONSTRAINT [
    FK_Day_Registration_Registration]
```

```
GO

ALTER TABLE [dbo].[Day_Registration] WITH CHECK ADD
    CONSTRAINT [CheckNumberOfPlaces_DR] CHECK (([
    NumberOfPlaces]>(0)))
GO

ALTER TABLE [dbo].[Day_Registration] CHECK CONSTRAINT [
    CheckNumberOfPlaces_DR]
GO
```

3.5 dbo.Days

```
USE [zakrzews_a]
GO

/****** Object:  Table [dbo].[Days]    Script Date:
    2017-01-25 23:35:00 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Days](
        [DayID] [int] IDENTITY(1,1) NOT NULL,
        [ConfID] [int] NOT NULL,
        [Date] [date] NOT NULL,
        [DayPrice] [money] NOT NULL,
 CONSTRAINT [PK_Days] PRIMARY KEY CLUSTERED
(
        [DayID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Days]  WITH CHECK ADD  CONSTRAINT [
    FK_Days_Conference] FOREIGN KEY([ConfID])
REFERENCES [dbo].[Conference] ([ConfID])
GO

ALTER TABLE [dbo].[Days] CHECK CONSTRAINT [
    FK_Days_Conference]
GO

ALTER TABLE [dbo].[Days]  WITH CHECK ADD  CONSTRAINT [
    CheckPrice] CHECK  (([DayPrice]>=(0)))
GO

ALTER TABLE [dbo].[Days] CHECK CONSTRAINT [CheckPrice]
GO
```

3.6 dbo.Discount

```
USE [zakrzews_a]
GO

/****** Object:  Table [dbo].[Discount]    Script Date:
    2017-01-25 23:35:13 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Discount](
        [DiscountID] [int] IDENTITY(1,1) NOT NULL,
        [ConfID] [int] NOT NULL,
        [Discount] [float] NOT NULL,
        [StartDate] [date] NOT NULL,
        [EndDate] [date] NOT NULL,
 CONSTRAINT [PK_Discount] PRIMARY KEY CLUSTERED
(
        [DiscountID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Discount]  WITH CHECK ADD  CONSTRAINT
    [FK_Discount_Conference1] FOREIGN KEY([ConfID])
REFERENCES [dbo].[Conference] ([ConfID])
GO

ALTER TABLE [dbo].[Discount] CHECK CONSTRAINT [
    FK_Discount_Conference1]
GO

ALTER TABLE [dbo].[Discount]  WITH CHECK ADD  CONSTRAINT
    [CheckDiscount] CHECK  (([Discount]>=(0) AND [Discount
    ]<=(1)))
GO

ALTER TABLE [dbo].[Discount] CHECK CONSTRAINT [
    CheckDiscount]
GO
```

```
ALTER TABLE [dbo].[Discount] WITH CHECK ADD CONSTRAINT
    [CheckStartEnd_Disc] CHECK (([StartDate]<=[EndDate]))
GO

ALTER TABLE [dbo].[Discount] CHECK CONSTRAINT [
    CheckStartEnd_Disc]
GO
```

3.7 dbo.Participant

```
USE [zakrzews_a]
GO

/****** Object:  Table [dbo].[Participant]    Script Date
    : 2017-01-25 23:36:07 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

SET ANSI_PADDING ON
GO

CREATE TABLE [dbo].[Participant](
        [ParticipantID] [int] IDENTITY(1,1) NOT NULL,
        [FirstName] [varchar](50) NULL,
        [LastName] [varchar](50) NULL,
 CONSTRAINT [PK_Participant] PRIMARY KEY CLUSTERED
(
        [ParticipantID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

SET ANSI_PADDING OFF
GO
```

3.8 dbo.Registration

```
USE [zakrzews_a]
GO

/****** Object:  Table [dbo].[Registration]    Script
    Date: 2017-01-25 23:36:22 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Registration](
        [RegistrationID] [int] IDENTITY(1,1) NOT NULL,
        [ConfID] [int] NOT NULL,
        [ClientID] [int] NOT NULL,
        [RegistrationDate] [date] NOT NULL,
        [PayDate] [date] NULL,
 CONSTRAINT [PK_Registration] PRIMARY KEY CLUSTERED
(
        [RegistrationID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Registration]  WITH CHECK ADD
    CONSTRAINT [FK_Registration_Client] FOREIGN KEY([
    ClientID])
REFERENCES [dbo].[Client] ([ClientID])
GO

ALTER TABLE [dbo].[Registration] CHECK CONSTRAINT [
    FK_Registration_Client]
GO

ALTER TABLE [dbo].[Registration]  WITH CHECK ADD
    CONSTRAINT [FK_Registration_Conference] FOREIGN KEY([
    ConfID])
REFERENCES [dbo].[Conference] ([ConfID])
GO
```

```
ALTER TABLE [dbo].[Registration] CHECK CONSTRAINT [
    FK_Registration_Conference]
GO
```

3.9 dbo.Student

```
USE [zakrzews_a]
GO

/****** Object:  Table [dbo].[Student]    Script Date:
    2017-01-25 23:36:40 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

SET ANSI_PADDING ON
GO

CREATE TABLE [dbo].[Student](
        [ParticipantID] [int] NOT NULL,
        [StudentCardNumber] [varchar](50) NOT NULL
) ON [PRIMARY]

GO

SET ANSI_PADDING OFF
GO

ALTER TABLE [dbo].[Student]  WITH CHECK ADD  CONSTRAINT [
    FK_Student_Participant] FOREIGN KEY([ParticipantID])
REFERENCES [dbo].[Participant] ([ParticipantID])
GO

ALTER TABLE [dbo].[Student] CHECK CONSTRAINT [
    FK_Student_Participant]
GO
```

3.10 dbo.WhichfDay

```
USE [zakrzews_a]
GO

/****** Object:  Table [dbo].[Which_Day]    Script Date:
    2017-01-25 23:36:52 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Which_Day](
        [ParticipantID] [int] NOT NULL,
        [DayRegID] [int] NOT NULL
) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Which_Day]  WITH CHECK ADD  CONSTRAINT
     [FK_Which_Day_Day_Registration] FOREIGN KEY([DayRegID
    ])
REFERENCES [dbo].[Day_Registration] ([DayRegID])
GO

ALTER TABLE [dbo].[Which_Day] CHECK CONSTRAINT [
    FK_Which_Day_Day_Registration]
GO

ALTER TABLE [dbo].[Which_Day]  WITH CHECK ADD  CONSTRAINT
     [FK_Which_Day_Participant] FOREIGN KEY([ParticipantID
    ])
REFERENCES [dbo].[Participant] ([ParticipantID])
GO

ALTER TABLE [dbo].[Which_Day] CHECK CONSTRAINT [
    FK_Which_Day_Participant]
GO
```

3.11 dbo.WhichWorkshop

```
USE [zakrzews_a]
GO

/****** Object:  Table [dbo].[Which_Workshop]    Script
    Date: 2017-01-25 23:37:16 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Which_Workshop](
        [ParticipantID] [int] NOT NULL,
        [WorkRegID] [int] NOT NULL
) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Which_Workshop]  WITH CHECK ADD
    CONSTRAINT [FK_Which_Workshop_Participant] FOREIGN KEY
    ([ParticipantID])
REFERENCES [dbo].[Participant] ([ParticipantID])
GO

ALTER TABLE [dbo].[Which_Workshop] CHECK CONSTRAINT [
    FK_Which_Workshop_Participant]
GO

ALTER TABLE [dbo].[Which_Workshop]  WITH CHECK ADD
    CONSTRAINT [FK_Which_Workshop_Workshop_Registration]
    FOREIGN KEY([WorkRegID])
REFERENCES [dbo].[Workshop_Registration] ([WorkRegID])
GO

ALTER TABLE [dbo].[Which_Workshop] CHECK CONSTRAINT [
    FK_Which_Workshop_Workshop_Registration]
GO
```

3.12 dbo.WorkshopLesson

```
USE [zakrzews_a]
GO

/****** Object:  Table [dbo].[Workshop_Lesson]    Script
    Date: 2017-01-25 23:37:47 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Workshop_Lesson](
        [LessonID] [int] IDENTITY(1,1) NOT NULL,
        [WorkshopID] [int] NOT NULL,
        [DayID] [int] NOT NULL,
        [NumberOfPlaces] [int] NOT NULL,
        [Price] [money] NOT NULL,
        [StartHour] [datetime] NOT NULL,
        [EndHour] [datetime] NOT NULL,
 CONSTRAINT [PK_Workshop_Lesson] PRIMARY KEY CLUSTERED
(
        [LessonID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Workshop_Lesson]  WITH CHECK ADD
    CONSTRAINT [FK_Workshop_Lesson_Days] FOREIGN KEY([
    DayID])
REFERENCES [dbo].[Days] ([DayID])
GO

ALTER TABLE [dbo].[Workshop_Lesson] CHECK CONSTRAINT [
    FK_Workshop_Lesson_Days]
GO

ALTER TABLE [dbo].[Workshop_Lesson]  WITH CHECK ADD
    CONSTRAINT [FK_Workshop_Lesson_Workshop_Name] FOREIGN
    KEY([WorkshopID])
REFERENCES [dbo].[Workshop_Name] ([WorkshopID])
GO
```

```
ALTER TABLE [dbo].[Workshop_Lesson] CHECK CONSTRAINT [
    FK_Workshop_Lesson_Workshop_Name]
GO

ALTER TABLE [dbo].[Workshop_Lesson]  WITH CHECK ADD
    CONSTRAINT [CheckLessonHours] CHECK  (([StartHour]<[
    EndHour]))
GO

ALTER TABLE [dbo].[Workshop_Lesson] CHECK CONSTRAINT [
    CheckLessonHours]
GO

ALTER TABLE [dbo].[Workshop_Lesson]  WITH CHECK ADD
    CONSTRAINT [CheckNumberOfPlaces] CHECK  (([
    NumberOfPlaces]>(0)))
GO

ALTER TABLE [dbo].[Workshop_Lesson] CHECK CONSTRAINT [
    CheckNumberOfPlaces]
GO

ALTER TABLE [dbo].[Workshop_Lesson]  WITH CHECK ADD
    CONSTRAINT [CheckWorkshopPrice] CHECK  (([Price]>=(0))
    )
GO

ALTER TABLE [dbo].[Workshop_Lesson] CHECK CONSTRAINT [
    CheckWorkshopPrice]
GO
```

3.13 dbo.WorkshopName

```
USE [zakrzews_a]
GO

/****** Object:  Table [dbo].[Workshop_Name]    Script
    Date: 2017-01-25 23:38:04 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

SET ANSI_PADDING ON
GO

CREATE TABLE [dbo].[Workshop_Name](
        [WorkshopID] [int] IDENTITY(1,1) NOT NULL,
        [Title] [varchar](50) NOT NULL,
        [Description] [varchar](255) NOT NULL,
 CONSTRAINT [PK_Workshop_Name] PRIMARY KEY CLUSTERED
(
        [WorkshopID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

SET ANSI_PADDING OFF
GO
```

3.14 dbo.WorkshopRegistration

```
USE [zakrzews_a]
GO

/****** Object:  Table [dbo].[Workshop_Registration]
    Script Date: 2017-01-25 23:38:22 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Workshop_Registration](
        [WorkRegID] [int] IDENTITY(1,1) NOT NULL,
        [DayRegID] [int] NOT NULL,
        [LessonID] [int] NOT NULL,
        [Date] [date] NOT NULL,
        [NumberOfPlaces] [int] NOT NULL,
 CONSTRAINT [PK_Workshop_Registration] PRIMARY KEY
     CLUSTERED
(
        [WorkRegID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Workshop_Registration]  WITH CHECK ADD
      CONSTRAINT [
    FK_Workshop_Registration_Day_Registration] FOREIGN KEY
    ([DayRegID])
REFERENCES [dbo].[Day_Registration] ([DayRegID])
GO

ALTER TABLE [dbo].[Workshop_Registration] CHECK
    CONSTRAINT [FK_Workshop_Registration_Day_Registration]
GO

ALTER TABLE [dbo].[Workshop_Registration]  WITH CHECK ADD
      CONSTRAINT [FK_Workshop_Registration_Workshop_Lesson
    ] FOREIGN KEY([LessonID])
REFERENCES [dbo].[Workshop_Lesson] ([LessonID])
GO
```

```
ALTER TABLE [dbo].[Workshop_Registration] CHECK
    CONSTRAINT [FK_Workshop_Registration_Workshop_Lesson]
GO

ALTER TABLE [dbo].[Workshop_Registration]  WITH CHECK ADD
      CONSTRAINT [CheckNumberOfPlaces_WR] CHECK  (([
    NumberOfPlaces]>(0)))
GO

ALTER TABLE [dbo].[Workshop_Registration] CHECK
    CONSTRAINT [CheckNumberOfPlaces_WR]
GO
```

4 Triggery

4.1 dbo.TRIGGERblockAddUserDayRegistration

```
USE [zakrzews_a]
GO

/****** Object:  Trigger [dbo].[
    blockAddUserDayRegistration]    Script Date:
    2017-01-25 23:42:54 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO



--blockaddtoRegistrationDay

CREATE TRIGGER [dbo].[TRIGGER_blockAddUserDayRegistration
    ] ON [dbo].[Day_Registration]
AFTER INSERT
AS
BEGIN
                    IF(dbo.freePlacesForDayRegistration((
                        SELECT DayRegID FROM inserted)) = 0)
                    BEGIN

                            RAISERROR('All places for this
                                Registration Day are not
                                available!', 16, 1)
                            ROLLBACK TRANSACTION
                    END
END

GO
```

4.2 dbo.TRIGGERnumberOfPlacesDay

```
USE [zakrzews_a]
GO

/****** Object:  Trigger [dbo].[TRIGGER_numberOfPlacesDay
    ]    Script Date: 2017-01-25 23:44:45 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TRIGGER [dbo].[TRIGGER_numberOfPlacesDay] ON [dbo
    ].[Day_Registration]
AFTER INSERT
AS
BEGIN
                    SET NOCOUNT ON

                    DECLARE @taken int;
                    DECLARE @free int;

                    SET @taken = (SELECT numberOfPlaces FROM
                        inserted);
                    SET @free = dbo.freePlacesForDay((SELECT
                        DayID FROM inserted)) + @taken;

                    IF(@free < @taken)
                    BEGIN

                            RAISERROR('You tried to book %d
                                places, but only %d are
                                available!', 16, 1, @taken,
                                @free)
                            ROLLBACK TRANSACTION
                    END
END

GO
```

4.3 dbo.TRIGGERblockAddUserWorkshopRegistration

```
USE [zakrzews_a]
GO

/****** Object:  Trigger [dbo].[
    TRIGGER_blockAddUserWorkshopRegistration]    Script
    Date: 2017-01-25 23:47:54 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO




--blockaddtoRegistrationWorkshop

CREATE TRIGGER [dbo].[
    TRIGGER_blockAddUserWorkshopRegistration] ON [dbo].[
    Workshop_Registration]
AFTER INSERT
AS
BEGIN
                    IF(dbo.freePlacesForWorkshopRegistration
                        ((SELECT WorkRegID FROM inserted)) =
                        0)
                    BEGIN

                            RAISERROR('All places for this
                                Registration Workshop are not
                                available!', 16, 1)
                            ROLLBACK TRANSACTION
                    END
END


GO
```

5 Widoki

5.1 dbo.VIEWClientsRegistrations

```
CREATE VIEW [dbo].[VIEW_ClientsRegistrations]
AS
SELECT          C.ClientID, COUNT(*) AS NUMBER
FROM              dbo.Client AS C INNER JOIN
                                dbo.Registration AS R ON R.
                                    ClientID = C.ClientID
GROUP BY C.ClientID

GO
```

5.2 dbo.VIEWClientsToCall

```
CREATE VIEW [dbo].[VIEW_ClientsToCall]
AS
SELECT          C.ClientID, C.FirstName, C.LastName, C.
    phone, 'DAY' AS registrationType, R.RegistrationID,
    dbo.freePlacesForDayRegistration(DR.DayRegID) AS
    freePlaces
FROM            Client C INNER JOIN
                        Registration R ON R.ClientID = C
                            .ClientID INNER JOIN
                        Day_Registration DR ON DR.
                            RegistrationID = R.
                            RegistrationID INNER JOIN
                        Days D ON D.DayID = DR.DayID
                            INNER JOIN
                        Conference CO ON CO.ConfID = D.
                            ConfID
WHERE       dbo.freePlacesForDayRegistration(DR.DayID) >
     0 AND DATEDIFF(day, CONVERT(date, GETDATE()), CO.
    StartDate) BETWEEN 0 AND 14
UNION
SELECT          C.ClientID, C.FirstName, C.LastName, C.
    phone, 'WORKSHOP' AS registrationType, R.
    RegistrationID, dbo.freePlacesForWorkshopRegistration(
    WR.WorkRegID) AS freePlaces
FROM            Client C INNER JOIN
                        Registration R ON R.ClientID = C
                            .ClientID INNER JOIN
                        Day_Registration DR ON DR.
                            RegistrationID = R.
                            RegistrationID INNER JOIN
                        Days D ON D.DayID = DR.DayID
                            INNER JOIN
                        Conference CO ON CO.ConfID = D.
                            ConfID INNER JOIN
                        Workshop_Registration WR ON WR.
                            DayRegID = DR.DayRegID
WHERE       dbo.freePlacesForWorkshopRegistration(WR.
    WorkRegID) > 0 AND DATEDIFF(day, CONVERT(date, GETDATE
    ()), CO.StartDate) BETWEEN 0 AND 14

GO
```

5.3 dbo.VIEWCustomersThatShouldPayTomorrow

```
create view [dbo].[VIEW_CustomersThatShouldPayTomorrow]
    as
select C.ClientID, C.FirstName, C.LastName, C.[E-mail], C
    .Phone
        from Client as C
inner join Registration as R
        on C.ClientID = R.ClientID
where DATEDIFF(day, convert(date, getdate()), R.
    RegistrationDate) = 6

GO
```

5.4 dbo.VIEWOutdatedPayments

```
CREATE VIEW [dbo].[VIEW_OutdatedPayments]
AS
SELECT RegistrationID FROM Registration
WHERE DATEDIFF(day, CONVERT(date, GETDATE()),
    RegistrationDate) > 7
GO
```

5.5 dbo.VIEWParticipantStats

```
CREATE VIEW [dbo].[VIEW_ParticipantStats] AS

SELECT P.ParticipantID, COUNT(WD.DayRegID) AS number, '
    DAYS' AS activityType
FROM Participant P
INNER JOIN Which_Day WD ON WD.ParticipantID = P.
    ParticipantID
GROUP BY P.ParticipantID

UNION

SELECT P.ParticipantID, COUNT(WW.WorkRegID) AS number, '
    WORKSHOP' AS activityType
FROM Participant P
INNER JOIN Which_Workshop WW ON WW.ParticipantID = P.
    ParticipantID
GROUP BY P.ParticipantID

GO
```

5.6 dbo.VIEWStudentParticipants

```
CREATE VIEW [dbo].[VIEW_studentParticipants]
AS
SELECT ParticipantID FROM Student
GO
```

6 Procedury

6.1 dbo.PROCEDUREviewParticipantsForDay

```
CREATE PROCEDURE [dbo].[PROCEDURE_viewParticipantsForDay]
@DayID INT
AS
BEGIN
SET NOCOUNT ON;
        DECLARE @DayID2 INT = (SELECT DayID FROM Days
            WHERE DayID = @DayID)
        IF(@DayID2 IS NULL)
        BEGIN
        ;THROW 52000, 'There is no such day', 1
        END

        SELECT P.ParticipantID, P.FirstName, P.LastName
        FROM dbo.Participant P
        INNER JOIN dbo.Which_Day WD ON WD.ParticipantID =
            P.ParticipantID
        INNER JOIN dbo.Day_Registration DR ON DR.DayRegID
            = WD.DayRegID
        INNER JOIN dbo.Days D ON D.DayID = DR.DayID
        WHERE D.DayID = @DayID

END
```

6.2 dbo.PROCEDUREviewParticipantsForWorkshop

```
CREATE PROCEDURE [dbo].[
    PROCEDURE_viewParticipantsForWorkshop]
@LessonID INT
AS
BEGIN
SET NOCOUNT ON;
        DECLARE @LessonID2 INT = (SELECT LessonID FROM
            dbo.Workshop_Lesson WHERE LessonID = @LessonID
            )
        IF(@LessonID2 IS NULL)
        BEGIN
        ;THROW 52000, 'There is no such lesson', 1
        END

        SELECT P.ParticipantID, P.FirstName, P.LastName
        FROM dbo.Participant P
        INNER JOIN dbo.Which_Workshop WW ON WW.
            ParticipantID = P.ParticipantID
        INNER JOIN dbo.Workshop_Registration WR ON WR.
            WorkRegID = WW.WorkRegID
        INNER JOIN dbo.Workshop_Lesson WL ON WL.LessonID
            = WR.LessonID
        WHERE WL.LessonID = @LessonID2
END
```

6.3 dbo.PROCEDUREaddClient

```
USE [zakrzews_a]
GO

/****** Object:  StoredProcedure [dbo].[
    PROCEDURE_addClient]    Script Date: 2017-01-25
    23:53:43 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[PROCEDURE_addClient]
@firstName varchar(50), @lastName varchar(50), @Phone
    varchar(15), @mail varchar(255)
AS
BEGIN
SET NOCOUNT ON;

                    INSERT INTO Client(FirstName, LastName,
                        Phone, [E-mail])
                    VALUES(@firstName, @lastName, @Phone,
                        @mail)
END

GO
```

6.4 dbo.PROCEDUREaddCompany

```
USE [zakrzews_a]
GO

/****** Object:  StoredProcedure [dbo].[
    PROCEDURE_addCompany]    Script Date: 2017-01-25
    23:55:42 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[PROCEDURE_addCompany]
@ClientID int, @CompanyName varchar(50), @City varchar
    (50), @address varchar(50), @country varchar(50)
AS
BEGIN
SET NOCOUNT ON;

                    DECLARE @ClientID2 int = (SELECT ClientID
                        FROM CLient WHERE CLientID =
                        @ClientID)

                    IF(@ClientID2 IS NULL)
                    BEGIN
                    ;THROW 52000, 'There is no such client.'
                        ,1
                    END

                    INSERT INTO Company(ClientID, CompanyName
                        , City, Address, Country)
                    VALUES(@ClientID, @CompanyName, @City,
                        @address, @country)

END

GO
```

6.5 dbo.PROCEDUREaddConference

```
USE [zakrzews_a]
GO

/****** Object:  StoredProcedure [dbo].[
    PROCEDURE_addConference]    Script Date: 2017-01-25
    23:56:03 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[PROCEDURE_addConference]
@Name varchar(255), @StartDate date, @EndDate date, @City
     varchar(50), @Address varchar(50), @Country varchar
    (50), @NumberOfPlaces int, @StudentDiscount float
AS
BEGIN
SET NOCOUNT ON;

                    IF(@NumberOfPlaces < 0)
                    BEGIN
                    ;THROW 52000, 'The number of places must
                        not be negative.', 1
                    END
                    IF(@StartDate > @EndDate)
                    BEGIN
                    ;THROW 52000, 'EndDate should not be
                        earlier than StartDate.',1
                    END
                    IF(@StudentDiscount < 0 OR
                        @StudentDiscount > 1)
                    BEGIN
                    ;THROW 52000, 'The discount must be
                        between 0 and 1.',1
                    END

                    INSERT INTO Conference(Name, StartDate,
                        EndDate, City, Address, Country,
                        studentDiscount, NumberOfPlaces)
                    VALUES(@Name, @StartDate, @EndDate, @City
                        , @Address, @Country, @StudentDiscount
                        , @NumberOfPlaces)
END
```

GO

6.6 dbo.PROCEDUREaddDay

```
USE [zakrzews_a]
GO

/****** Object:  StoredProcedure [dbo].[PROCEDURE_addDay]
        Script Date: 2017-01-25 23:56:16 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[PROCEDURE_addDay]
@ConfID int, @Date date, @DayPrice money
AS
BEGIN
SET NOCOUNT ON;

                    DECLARE @Conference int = (SELECT ConfID
                        FROM Conference WHERE ConfID = @ConfID
                        )
                    IF(@conference IS NULL)
                    BEGIN
                    ;THROW 52000, 'There is no such
                        conference.', 1
                    END

                    IF(@DayPrice < 0)
                    BEGIN
                    ;THROW 52000, 'The price of day must not
                        be negative.', 1
                    END

                    DECLARE @startDate date = (SELECT
                        StartDate FROM Conference WHERE ConfID
                         = @ConfID)
                    DECLARE @EndDate date = (SELECT EndDate
                        FROM Conference WHERE ConfID = @ConfID
                        )
                    IF(@Date < @startDate OR  @Date >
                        @EndDate)
                    BEGIN
                    ;THROW 52000, 'Date must be between
                        startdate and enddate of conference.',
                         1
```

```
                    END

                    DECLARE @Date2 date = (SELECT date FROM
                        Days WHERE ConfID = @ConfID AND Date =
                         @Date)
                    IF(@Date2 IS NOT NULL)
                    BEGIN
                    ;THROW 52000, 'This day of conference
                        already exists.', 1
                    END

                    INSERT INTO Days(ConfID, date, DayPrice)
                    VALUES(@ConfID, @Date, @DayPrice)

END

GO
```

6.7 dbo.PROCEDUREaddDayRegistration

```
USE [zakrzews_a]
GO

/****** Object:  StoredProcedure [dbo].[
    PROCEDURE_addDayRegistration]    Script Date:
    2017-01-25 23:56:43 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[PROCEDURE_addDayRegistration]
@RegistrationID int, @NumberOfPlaces int, @DayID int
AS
BEGIN
SET NOCOUNT ON;

                    DECLARE @Day int = (SELECT DayID FROM
                        Days WHERE DayID = @DayID);
                    IF(@Day IS NULL)
                    BEGIN
                    ;THROW 52000, 'There is no such day.',1
                    END

                    DECLARE @Registration int = (SELECT
                        RegistrationID FROM Registration WHERE
                         RegistrationID = @RegistrationID);
                    IF(@Registration IS NULL)
                    BEGIN
                    ;THROW 52000, 'There is no such
                        Registration.', 1
                    END

                    DECLARE @ConfID int = (SELECT ConfID FROM
                         Registration WHERE RegistrationID =
                        @RegistrationID)
                    DECLARE @ConfID2 int = (SELECT COnfID
                       FROM DAys WHERE DayID = @DayID)

                    IF(@ConfID != @ConfID2)
                    BEGIN
                    ;THROW 52000, 'You picked a day from
                        wrong conference', 1
```

```
                    END

                    DECLARE @freePlacesforDay int = dbo.
                        freePlacesForDay(@DayID);

                    IF(@NumberOfPlaces > @freePlacesforDay)
                    BEGIN
                    PRINT 'Only ' + CAST(@freePlacesforDay AS
                        VARCHAR(10)) + ' available'
                    ;THROW 52000, 'Not enouch places
                        available', 1
                    END

                    INSERT INTO Day_Registration(
                        RegistrationID, DayID, NumberOfPlaces)
                    VALUES(@RegistrationID, @Day,
                        @NumberOfPlaces)
END
GO
```

6.8 dbo.PROCEDUREaddDiscount

```
USE [zakrzews_a]
GO

/****** Object:  StoredProcedure [dbo].[
    PROCEDURE_addDiscount]    Script Date: 2017-01-25
    23:57:02 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[PROCEDURE_addDiscount]
@ConfID int , @Discount float, @Startdate date, @EndDate
    date
AS
BEGIN

                    DECLARE @Conf int =(SELECT COnfID FROM
                        Conference WHERE ConfID = @ConfID)
                    IF(@Conf IS NULL)
                    BEGIN
                    ;THROW 52000, 'There is no such
                        conference.', 1
                    END

                    IF(@Discount NOT BETWEEN 0 AND 1)
                    BEGIN
                    ;THROW 52000, 'Discount should be between
                        0 and 1.', 1
                    END

                    IF(@StartDate > @EndDate)
                    BEGIN
                    ;THROW 52000, 'StartDate should be before
                        EndDate.', 1
                    END


                    IF(@ConfID  IN (        SELECT ConfID
                            FROM Discount
                            WHERE (@Startdate > StartDate AND
                                @StartDate < EndDate) OR (
                                @EndDate > StartDate AND
```

```
                    @EndDate < EndDate)))
                BEGIN
                ;THROW 52000, 'Collision with another
                    discount from this conference', 1
                END

                INSERT INTO Discount(ConfID, Discount,
                    StartDate, EndDate)
                VALUES(@ConfID, @Discount, @Startdate,
                    @EndDate)

END
GO
```

6.9 dbo.PROCEDUREaddLesson

```
USE [zakrzews_a]
GO

/****** Object:  StoredProcedure [dbo].[
    PROCEDURE_addLesson]    Script Date: 2017-01-25
    23:57:19 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[PROCEDURE_addLesson]
@WorkshopID int, @DayID int, @NumberOfPlaces int, @Price
    money, @StartHour datetime, @EndHour datetime
AS
BEGIN
SET NOCOUNT ON;

                    DECLARE @WorkshopID2 int = (SELECT
                        WorkshopID FROM Workshop_Name WHERE
                        WorkshopID = @WorkshopID)
                    IF(@WorkshopID2 IS NULL)
                    BEGIN
                    ;THROW 52000, 'There is no such workshop.
                        ', 1
                    END

                    DECLARE @DayID2 int = (SELECT DayID FROM
                        Days WHERE DayID = @DayID)
                    IF(@DayID2 IS NULL)
                    BEGIN
                    ;THROW 52000, 'There is no such day.', 1
                    END

                    IF(@NumberOfPlaces < 0)
                    BEGIN
                    ;THROW 52000, 'The number of places must
                        not be negative.', 1
                    END

                    DECLARE @conferencePlaces int = (
                        SELECT C.numberOfPlaces
                            FROM Conference C
```

```
                INNER JOIN Days D On D.ConfID = C
                    .ConfID
                WHERE D.DayID = @DayID)

            IF(@conferencePlaces < @NumberOfPlaces)
            BEGIN
            PRINT 'Only ' + CAST(@conferencePlaces AS
                VARCHAR(10)) + ' Available'
            ;THROW 52000, 'Not enough places
                available only', 1
            END

            IF(@StartHour > @EndHour)
            BEGIN
            ;THROW 52000, 'EndHour should not be
                earlier than StartHour.', 1
            END

            DECLARE @daydate date;
            SET @daydate = (SELECT date FROM Days
                WHERE DayID = @DayID)

            DECLARE @startDate date = CONVERT(date,
                @starthour)
            DECLARE @endDate date = CONVERT(date,
                @endhour)

            IF(@startDate != @daydate OR @endDate !=
                @daydate)
            BEGIN
            ;THROW 52000, 'You picked the wrong start
                hour or endhour - not from this day',
                 1
            END

            INSERT INTO Workshop_Lesson(WorkshopID,
                DayID, NumberOfPlaces, Price,
                StartHour, EndHour)
            VALUES(@WorkshopID, @DayID,
                @NumberOfPlaces, @Price, @StartHour,
                @EndHour)

END
GO
```

6.10 dbo.PROCEDUREaddParticipant

```
USE [zakrzews_a]
GO

/****** Object:  StoredProcedure [dbo].[
    PROCEDURE_addParticipant]    Script Date: 2017-01-25
    23:57:42 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[PROCEDURE_addParticipant]
@FirstName varchar(50), @LastName varchar(50)
AS
BEGIN
SET NOCOUNT ON;

                    INSERT INTO Participant(FirstName,
                        LastName)
                    VALUES(@FirstName, @LastName)

END

GO
```

6.11 dbo.PROCEDUREaddParticipantToDay

```
USE [zakrzews_a]
GO

/****** Object:  StoredProcedure [dbo].[
    PROCEDURE_addParticipantToDay]    Script Date:
    2017-01-25 23:58:01 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[PROCEDURE_addParticipantToDay]
@ParticipantID int, @DayRegID int
AS
BEGIN
SET NOCOUNT ON;

                    DECLARE @ParticipantID2 int = (SELECT
                        ParticipantID FROM Which_Day WHERE
                        ParticipantID = @ParticipantID AND
                        DayRegID = @DayRegID)
                    IF(@ParticipantID2 IS NOT NULL)
                    BEGIN
                    ;THROW 52000, 'This participant is
                        already in this Day Registration', 1
                    END

                    DECLARE @Participant int = (SELECT
                        ParticipantID FROM Participant WHERE
                        ParticipantID = @ParticipantID)
                    IF(@Participant IS NULL)
                    BEGIN
                    ;THROW 52000, 'There is no such
                        participant.', 1
                    END


                    DECLARE @DayRegID2 int = (SELECT DayRegID
                         FROM Day_Registration WHERE DayRegID
                        = @DayRegID)
                    IF(@DayRegID2 IS NULL)
                    BEGIN
```

```
                ;THROW 52000, 'There is no such
                    dayRegistration.', 1
                END

                INSERT INTO Which_Day(ParticipantID,
                    DayRegID)
                VALUES(@ParticipantID, @DayRegID)

END

GO
```

6.12 dbo.PROCEDUREaddParticipantToWorkshop

```
USE [zakrzews_a]
GO

/****** Object:  StoredProcedure [dbo].[
    PROCEDURE_addParticipantToWorkshop]    Script Date:
    2017-01-25 23:58:27 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[
    PROCEDURE_addParticipantToWorkshop]
@ParticipantID int, @WorkRegID int
AS
BEGIN
SET NOCOUNT ON;

                    DECLARE @DayRegID int = (SELECT DayRegID
                        FROM Workshop_Registration WHERE
                        WorkRegID = @WorkRegID)


                    IF(@ParticipantID NOT IN (SELECT
                        ParticipantID FROM Which_Day WHERE
                        DayRegID = @DayRegID))
                    BEGIN
                    ;THROW 52000, 'Participant is not
                        registered to day registration
                        connected to this workshop
                        registration', 1
                    END

                    DECLARE @ParticipantID2 int = (SELECT
                        ParticipantID FROM Participant WHERE
                        ParticipantID = @ParticipantID)
                    IF(@ParticipantID2 IS NULL)
                    BEGIN
                    ;THROW 52000, 'There is no such
                        participant.', 1
                    END
```

```
DECLARE @ParticipantID4 int = (SELECT
    ParticipantID FROM Which_Workshop
    WHERE ParticipantID = @ParticipantID
    AND WorkRegID = @WorkRegID)
IF(@ParticipantID4 IS NOT NULL)
BEGIN
;THROW 52000, 'Participant is already in
    this Workshop Registration', 1
END

DECLARE @WorkRegID2 int = (SELECT
    WorkRegID FROM Workshop_Registration
    WHERE WorkRegID = @WorkRegID)
IF(@WorkRegID2 IS NULL)
BEGIN
;THROW 52000, 'There is no such
    dayRegistration.', 1
END

DECLARE @LessonID int = (SELECT LessonID
    FROM Workshop_Registration WHERE
    WorkRegID = @WorkRegID)
DECLARE @DayID int = (SELECT DayID FROM
    Day_Registration WHERE DayRegID =
    @DayRegID)

IF(EXISTS (SELECT WR.LessonID
        FROM Workshop_Registration WR
        INNER JOIN Workshop_Lesson WL ON
            WL.LessonID = WR.LessonID
                WHERE WL.DayID = @DayID
                    AND dbo.
                     WorkshopCollision(
                     @LessonID, WR.LessonID
                     ) = 1

        INTERSECT

        SELECT WR.LessonID
        FROM Workshop_Registration WR
        INNER JOIN Which_Workshop WW ON
            WW.WorkRegID = WR.WorkRegID
        WHERE WW.ParticipantID =
            @ParticipantID))
BEGIN
```

```
                    ;THROW 52000, 'You cannot register this
                        participant to this registration (
                        workshop collision)', 1
                END
                INSERT INTO Which_Workshop(ParticipantID,
                    WorkRegID)
                VALUES(@ParticipantID, @WorkRegID)

END
GO
```

6.13 dbo.PROCEDUREaddRegistration

```
USE [zakrzews_a]
GO

/****** Object:  StoredProcedure [dbo].[
    PROCEDURE_addRegistration]    Script Date: 2017-01-25
    23:58:47 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[PROCEDURE_addRegistration]
@ClientID int, @ConfID int
AS
BEGIN
SET NOCOUNT ON;

                    DECLARE @Client int = (SELECT ClientID
                        FROM Client WHERE ClientID = @ClientID
                        )
                    IF(@Client IS NULL)
                    BEGIN
                    ;THROW 52000, 'There is no such client.'
                        ,1
                    END

                    DECLARE @Conf int = ( SELECT ConfID FROM
                        Conference WHERE ConfID = @ConfID)
                    IF(@Conf IS NULL)
                    BEGIN
                    ;THROW 52000, 'There is no such
                        conference.', 1
                    END

                    INSERT INTO Registration(ClientID,
                        RegistrationDate, ConfID)
                    VALUES(@ClientID, convert(date, getdate()
                        ), @ConfID)
END
GO
```

6.14 dbo.PROCEDUREaddStudent

```
USE [zakrzews_a]
GO

/****** Object:  StoredProcedure [dbo].[
    PROCEDURE_addStudent]    Script Date: 2017-01-25
    23:59:06 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO


CREATE PROCEDURE [dbo].[PROCEDURE_addStudent]
@ParticipantID int, @StudentCardNumber int
AS
BEGIN
SET NOCOUNT ON;
        DECLARE @Participant int = (
                SELECT ParticipantID FROM Participant
                WHERE ParticipantID = @ParticipantID
        )

        IF(@Participant IS NULL)
        BEGIN
        ;THROW 52000, 'There is no such participant.', 1
        END

        INSERT INTO Student(ParticipantID,
            StudentCardNumber)
        VALUES (@ParticipantID, @StudentCardNumber)
END
GO
```

6.15 dbo.PROCEDUREaddWorkshop

```
USE [zakrzews_a]
GO

/****** Object:  StoredProcedure [dbo].[
    PROCEDURE_addWorkshop]    Script Date: 2017-01-25
    23:59:28 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[PROCEDURE_addWorkshop]
@Title varchar(50), @Description varchar(255)
AS
BEGIN
SET NOCOUNT ON;

                    INSERT INTO Workshop_Name(Title,
                        Description)
                    VALUES(@Title, @Description)

END

GO
```

6.16 dbo.PROCEDUREaddWorkshopRegistration

```
USE [zakrzews_a]
GO

/****** Object:  StoredProcedure [dbo].[
    PROCEDURE_addWorkshopRegistration]    Script Date:
    2017-01-25 23:59:45 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[PROCEDURE_addWorkshopRegistration
    ]
@RegistrationID int, @NumberOfPlaces int, @LessonID int
AS
BEGIN
SET NOCOUNT ON;

                    DECLARE @Registration int = (SELECT
                        RegistrationID FROM Registration WHERE
                         RegistrationID = @RegistrationID);
                    IF(@Registration IS NULL)
                    BEGIN
                    ;THROW 52000, 'There is no such
                        Registration', 1
                    END

                    DECLARE @Lesson int = (SELECT LessonID
                       FROM Workshop_Lesson WHERE LessonID =
                        @LessonID);
                    IF(@Lesson IS NULL)
                    BEGIN
                    ;THROW 52000, 'No such lesson.',1
                    END

                    DECLARE @DayID int = (SELECT DayID FROM
                        Workshop_Lesson WHERE LessonID =
                        @LessonID);
                    DECLARE @DayRegID int = (SELECT DayRegID
                       FROM Day_Registration WHERE DayID =
                        @DayID);

                    IF(@DayRegID IS NULL)
```

```
                    BEGIN
                    ;THROW 52000, 'There is no such day
                        registration', 1
                    END

                    DECLARE @freePlacesforLesson int = dbo.
                        freePlacesForWorkshop(@LessonID);
                    DECLARE @freePlacesforDayRegistration int
                         = dbo.freePlacesForDayRegistration(
                        @DayRegID);

                    IF(@NumberOfPlaces >
                        @freePlacesforDayRegistration)
                    BEGIN
                    PRINT 'Only ' + CAST(
                        @freePlacesforDayRegistration AS
                       VARCHAR(10)) + ' available'
                    ;THROW 52000, 'Not enough places for this
                        DayRegistration', 1
                    END

                    IF(@NumberOfPlaces > @freePlacesforLesson
                        )
                    BEGIN
                    PRINT 'Only ' + CAST(@freePlacesforLesson
                         AS VARCHAR(10)) + ' available'
                    ;THROW 52000, 'Not enough places for this
                        Lesson', 1
                    END

                    INSERT INTO Workshop_Registration(
                        DayRegID, LessonID, Date,
                        NumberOfPlaces)
                    VALUES(@DayRegID, @Lesson, convert(date,
                        getdate()), @NumberOfPlaces)
END
GO
```

6.17 dbo.PROCEDUREchangeNumberOfPlacesForLesson

```
USE [zakrzews_a]
GO

/****** Object:  StoredProcedure [dbo].[
    PROCEDURE_changeNumberOfPlacesForLesson]    Script
    Date: 2017-01-26 00:05:48 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[
    PROCEDURE_changeNumberOfPlacesForLesson]
@NumberOfPlaces int, @LessonID int
AS
BEGIN

                    DECLARE @Lesson int = (SELECT LessonID
                        FROM Workshop_Lesson WHERE LessonID =
                        @LessonID)
                    IF(@Lesson IS NULL)
                    BEGIN
                    ;THROW 52000, 'There is no such workshop
                        lesson', 1
                    END

                    DECLARE @allPlaces int = (SELECT
                        NumberOFPlaces FROM Workshop_Lesson
                        WHERE LessonID = @LessonID)
                    DECLARE @freePlaces int = dbo.
                        freePlacesForWorkshop(@LessonID)

                    DECLARE @takenPlaces int = (@allPlaces -
                        @freePlaces)

                    IF(@takenPlaces > @NumberOfPlaces)
                    BEGIN
                    ;THROW 52000, 'You cannot change number
                        of places to this number, already more
                        places taken.', 1
                    END
```

```
                    DECLARE @DayID int = (SELECT DayID FROM
                        Workshop_Lesson WHERE LessonID =
                        @LessonID)
                    DECLARE @freePlacesforDay int = dbo.
                        freePlacesForDay(@DayID)

                    IF(@NumberOfPlaces > (@allPlaces +
                        @freePlacesforDay))
                    BEGIN
                    ;THROW 52000, 'You cannot change number
                        of places to this number, it exeeds
                        number of places for this day', 1
                    END

                    UPDATE Workshop_Lesson
                    SET NumberOfPlaces = @NumberOfPlaces
                    WHERE LessonID = @LessonID

END
GO
```

6.18 dbo.PROCEDUREdeleteConference

```
USE [zakrzews_a]
GO

/****** Object:  StoredProcedure [dbo].[
    PROCEDURE_deleteConference]    Script Date: 2017-01-26
     00:07:05 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[PROCEDURE_deleteConference]
@ConfID int
AS
BEGIN

                    DECLARE @Conf int = (SELECT ConfID FROM
                        Conference WHERE ConfID = @ConfID)

                    IF(@Conf IS NULL)
                    BEGIN
                    ;THROW 52000, 'There is no such
                        conference', 1
                    END

                    DELETE
                    FROM Which_Workshop
                    WHERE WorkRegID IN (SELECT WorkRegID
                            FROM Workshop_Registration
                            WHERE LessonID IN (     SELECT
                                LessonID
                                    FROM Workshop_Lesson
                                    WHERE DayID IN (SELECT
                                        DayID
                                    FROM Days
                                    WHERE ConfID = @ConfID)))
                    DELETE
                    FROM Workshop_Registration
                    WHERE LessonID IN (     SELECT LessonID
                            FROM Workshop_Lesson
                            WHERE DayID IN (SELECT DayID
                            FROM Days
                            WHERE ConfID = @ConfID))
```

```
                    DELETE
                    FROM Workshop_Lesson
                    WHERE DayID IN (SELECT DayID
                    FROM Days
                    WHERE ConfID = @ConfID)

                    DELETE
                    FROM Which_Day
                    WHERE DayRegID IN (      SELECT DayRegID
                            FROM Day_Registration
                            WHERE DayID IN (SELECT DayID
                            FROM Days
                            WHERE ConfID = @ConfID))

                    DELETE
                    FROM Day_Registration
                    WHERE DayID IN (SELECT DayID
                    FROM Days
                    WHERE ConfID = @ConfID)

                    DELETE
                    FROM Days
                    WHERE ConfID = @ConfID

                    DELETE
                    FROM Registration
                    WHERE ConfID = @ConfID

                    DELETE
                    FROM Conference
                    WHERE ConfID = @ConfID

END
GO
```

6.19 dbo.PROCEDUREdeleteDayRegistration

```
USE [zakrzews_a]
GO

/****** Object:  StoredProcedure [dbo].[
    PROCEDURE_deleteDayRegistration]    Script Date:
    2017-01-26 00:07:25 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[PROCEDURE_deleteDayRegistration]
@DayRegID int
AS
BEGIN

                    DECLARE @DayReg int = (SELECT DayRegID
                        FROM Day_Registration WHERE DayRegID =
                          @DayRegID)
                    IF(@DayReg IS NULL)
                    BEGIN
                    ;THROW 52000, 'There is no such day
                        registration', 1
                    END

                    DELETE
                    FROM Which_Day
                    WHERE DayRegID = @DayRegID

                    DELETE
                    FROM Which_Workshop
                    WHERE WorkRegID IN (SELECT WorkRegID
                              FROM Workshop_Registration
                              WHERE DayRegID = @DayRegID)

                    DELETE
                    FROM Workshop_Registration
                    WHERE DayRegID = @DayRegID

                    DELETE
                    FROM Day_Registration
                    WHERE DayRegID = @DayRegID
```

```
END
GO
```

6.20 dbo.PROCEDUREdeleteLesson

```
USE [zakrzews_a]
GO

/****** Object:  StoredProcedure [dbo].[
    PROCEDURE_deleteLesson]    Script Date: 2017-01-26
    00:07:53 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[PROCEDURE_deleteLesson]
@LessonID int
AS
BEGIN

                    DECLARE @Lesson int = (SELECT LessonID
                        FROM Workshop_Lesson WHERE LessonID =
                        @LessonID)
                    IF(@Lesson IS NULL)
                    BEGIN
                    ;THROW 52000, 'There is no such lesson.',
                        1
                    END

                    DELETE
                    FROM Which_Workshop
                    WHERE WorkRegID IN (SELECT WorkRegID
                            FROM Workshop_Registration
                            WHERE LessonID = @LessonID)

                    DELETE
                    FROM Workshop_Registration
                    WHERE LessonID = @LessonID

                    DELETE
                    FROM Workshop_Lesson
                    WHERE LessonID = @LessonID

END
GO
```

6.21 dbo.PROCEDUREdeleteParticipantFromDayRegistration

```
USE [zakrzews_a]
GO

/****** Object:  StoredProcedure [dbo].[
    PROCEDURE_deleteParticipantFromDayRegistration]
    Script Date: 2017-01-26 00:08:28 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[
    PROCEDURE_deleteParticipantFromDayRegistration]
@ParticipantID int, @DayRegID int
AS
BEGIN
                    DECLARE @Participant int = (SELECT
                        ParticipantID FROM Participant WHERE
                        ParticipantID = @ParticipantID)
                    IF(@Participant IS NULL)
                    BEGIN
                    ;THROW 52000, 'There is no such
                        participant', 1
                    END

                    DECLARE @DayReg int = (SELECT DayRegID
                       FROM Day_Registration WHERE DayRegID =
                         @DayRegID)
                    IF(@DayReg IS NULL)
                    BEGIN
                    ;THROW 52000, 'There is no such day
                        registration', 1
                    END

                    DELETE
                    FROM Which_Day
                    WHERE ParticipantID = @ParticipantID AND
                        DayRegID = @DayRegID

                    DELETE
                    FROM Which_Workshop
                    WHERE ParticipantID = @ParticipantID AND
                        WorkRegID IN ( SELECT WorkRegID
```

```
                                    FROM
                                        Workshop_Registration
                                    WHERE DayRegID =
                                        @DayRegID)
END
GO
```

6.22 dbo.PROCEDUREdeleteParticipantFromWorkshopRegistration

```
USE [zakrzews_a]
GO

/****** Object:  StoredProcedure [dbo].[
    PROCEDURE_deleteParticipantFromWorkshopRegistration]
        Script Date: 2017-01-26 00:09:34 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[
    PROCEDURE_deleteParticipantFromWorkshopRegistration]
@ParticipantID int, @WorkRegID int
AS
BEGIN

                    DECLARE @Participant int = (SELECT
                        ParticipantID FROM Participant WHERE
                        ParticipantID = @ParticipantID)
                    IF(@Participant IS NULL)
                    BEGIN
                    ;THROW 52000, 'There is no such
                        participant', 1
                    END

                    DECLARE @WorkReg int = (SELECT WorkRegID
                        FROM Workshop_Registration WHERE
                        WorkRegID = @WorkRegID)
                    IF(@WorkReg IS NULL)
                    BEGIN
                    ;THROW 52000, 'There is no such day
                        registration', 1
                    END

                    DELETE
                    FROM Which_Workshop
                    WHERE ParticipantID = @ParticipantID AND
                        WorkRegID = @WorkRegID

END
GO
```

6.23 dbo.PROCEDUREdeleteRegistration

```
USE [zakrzews_a]
GO

/****** Object:  StoredProcedure [dbo].[
    PROCEDURE_deleteRegistration]    Script Date:
    2017-01-26 00:10:02 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[PROCEDURE_deleteRegistration]
@RegistrationID int
AS
BEGIN

                    DECLARE @Registration int = (SELECT
                        RegistrationID FROM Day_Registration
                       WHERE RegistrationID = @RegistrationID
                        )
                    IF(@Registration IS NULL)
                    BEGIN
                    ;THROW 52000, 'There is no such
                        registration', 1
                    END

                    DELETE
                    FROM Which_Workshop
                    WHERE WorkRegID IN (SELECT WorkRegID
                            FROM Workshop_Registration
                            WHERE DayRegID IN (     SELECT
                                DayRegID
                                    FROM Day_Registration
                                    WHERE RegistrationID =
                                        @RegistrationID))

                    DELETE
                    FROM Workshop_Registration
                    WHERE DayRegID IN (     SELECT DayRegID
                            FROM Day_Registration
                            WHERE RegistrationID =
                                @RegistrationID)
```

```
                    DELETE
                    FROM Which_Day
                    WHERE DayRegID IN (     SELECT DayRegID
                            FROM Day_Registration
                            WHERE RegistrationID =
                                @RegistrationID)

                    DELETE
                    FROM Day_Registration
                    WHERE RegistrationID = @RegistrationID

                    DELETE
                    FROM Registration
                    WHERE RegistrationID = @RegistrationID
END
GO
```

6.24 dbo.PROCEDUREdeleteWorkshopRegistration

```
USE [zakrzews_a]
GO

/****** Object:  StoredProcedure [dbo].[
    PROCEDURE_deleteWorkshopRegistration]    Script Date:
    2017-01-26 00:10:30 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[
    PROCEDURE_deleteWorkshopRegistration]
@WorkRegID int
AS
BEGIN

                    DECLARE @WorkReg int = (SELECT WorkRegID
                        FROM Workshop_Registration WHERE
                        WorkRegID = @WorkRegID)
                    IF(@WorkReg IS NULL)
                    BEGIN
                    ;THROW 52000, 'There is no such workshop
                        registration.', 1
                    END

                    DELETE FROM Which_Workshop
                    WHERE WorkRegID = @WorkRegID

                    DELETE FROM Workshop_Registration
                    WHERE WorkRegID = @WorkRegID

END
GO
```

6.25 dbo.PROCEDUREpayForRegistration

```
USE [zakrzews_a]
GO

/****** Object:  StoredProcedure [dbo].[
    PROCEDURE_payForRegistration]    Script Date:
    2017-01-26 00:11:03 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[PROCEDURE_payForRegistration]
@money money, @RegistrationID int
AS
BEGIN

        DECLARE @Registration int = (SELECT
            RegistrationID FROM Registration WHERE
            RegistrationID = @RegistrationID)
        IF(@Registration IS NULL)
        BEGIN
        ;THROW 52000, 'There is no such registration', 1
        END

        DECLARE @RegistrationMoney money = dbo.
            registrationCost(@RegistrationID)
        IF(@money < @RegistrationMoney)
        BEGIN
        ;THROW 52000, 'Not enough money paid.', 1
        END

        IF(@money > @RegistrationMoney)
        BEGIN
        ;THROW 52000, 'Too much money paid.', 1
        END

        UPDATE Registration
        SET PayDate = CONVERT(date, GETDATE())
        WHERE RegistrationID = @RegistrationID

END
GO
```

6.26 dbo.PROCEDUREviewParticipantDays

```
USE [zakrzews_a]
GO

/****** Object:  StoredProcedure [dbo].[
    PROCEDURE_viewParticipantDays]    Script Date:
    2017-01-26 00:11:32 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[PROCEDURE_viewParticipantDays]
@ParticipantID int
AS
BEGIN
SET NOCOUNT ON;
        DECLARE @Participant int = (
                SELECT ParticipantID FROM PARTICIPANT
                WHERE @ParticipantID = ParticipantID
        )

        IF (@Participant IS NULL)
        BEGIN
        ;THROW 52000, 'There is no such Participant.',1
        END

        SELECT D.DayID, D.Date FROM Days AS D
        INNER JOIN Day_Registration AS DR
        ON DR.DayID= D.DayID
        INNER JOIN Which_Day AS WD
        ON DR.DayRegID = WD.DayRegID
        WHERE WD.ParticipantID = @ParticipantID

END
GO
```

6.27 dbo.PROCEDUREviewParticipantWorkshop

```
USE [zakrzews_a]
GO

/****** Object:  StoredProcedure [dbo].[
    PROCEDURE_viewParticipantWorkshops]    Script Date:
    2017-01-26 00:12:04 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO


CREATE PROCEDURE [dbo].[
    PROCEDURE_viewParticipantWorkshops]
@ParticipantID int
AS
BEGIN
SET NOCOUNT ON;
        DECLARE @Participant int = (
                SELECT ParticipantID FROM PARTICIPANT
                WHERE @ParticipantID = ParticipantID
        )

        IF (@Participant IS NULL)
        BEGIN
        ;THROW 52000, 'There is no such Participant.',1
        END

        SELECT D.Date, WL.StartHour, WL.EndHour FROM Days
            AS D
        INNER JOIN Workshop_Lesson as WL
        ON WL.DayID = D.DayID
        INNER JOIN Workshop_Registration as WR
        ON WR.LessonID = WL.LessonID
        INNER JOIN Which_Workshop as WW
        ON WW.WorkRegID = WR.WorkRegID
        WHERE WW.ParticipantID = @ParticipantID

END
GO
```

7 Funkcje

7.1 dbo.freePlacesForDay

```
USE [zakrzews_a]
GO

/****** Object:  UserDefinedFunction [dbo].[
    freePlacesForDay]    Script Date: 2017-01-26 00:14:15
    ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE FUNCTION [dbo].[freePlacesForDay] (@DayID int)
RETURNS int
AS
BEGIN
                DECLARE @allPlaces int;
                SET @allPlaces = (SELECT NumberOfPlaces
                    FROM Conference WHERE ConfID = (SELECT
                      ConfID FROM Days WHERE DayID = @DayID
                    ))

                DECLARE @takenPlaces int;
                SET @takenPlaces = (SELECT SUM(
                    NumberOfPlaces) FROM Day_Registration
                    WHERE DayID = @DayID)

                RETURN (@allPlaces - @takenPlaces)
END


GO
```

7.2 dbo.freePlacesForDayRegistration

```
USE [zakrzews_a]
GO

/****** Object:  UserDefinedFunction [dbo].[
    freePlacesForDayRegistration]    Script Date:
    2017-01-26 00:14:29 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-- =============================================
-- Author:                <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =============================================
CREATE FUNCTION [dbo].[freePlacesForDayRegistration] (
    @DayRegID int)
RETURNS int
AS
BEGIN
                DECLARE @allPlaces int;
                SET @allPlaces = (SELECT NumberOfPlaces
                    FROM Day_Registration WHERE DayRegID =
                      @DayRegID)

                DECLARE @takenPlaces int;
                SET @takenPlaces = (SELECT COUNT(*) FROM
                    Which_Day WHERE DayRegID = @DayRegID)

                RETURN (@allPlaces - @takenPlaces)
END


GO
```

7.3 dbo.freePlacesForWorkshop

```
USE [zakrzews_a]
GO

/****** Object:  UserDefinedFunction [dbo].[
    freePlacesForWorkshop]    Script Date: 2017-01-26
    00:15:14 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-------------------------------------
-------------FUNKCJE-----------------
-------------------------------------

--freePlacesForWorkshop
CREATE FUNCTION [dbo].[freePlacesForWorkshop] (@LessonID
    int)
RETURNS int
AS
BEGIN
                    DECLARE @allPlaces int;
                    SET @allPlaces = (SELECT NumberOfPlaces
                        FROM Workshop_Lesson WHERE LessonID =
                        @LessonID)

                    DECLARE @takenPlaces int;
                    SET @takenPlaces = (SELECT SUM(
                        NumberOfPlaces) FROM
                        Workshop_Registration WHERE LessonID =
                         @LessonID)

                    RETURN (@allPlaces - @takenPlaces)
END


GO
```

7.4 dbo.freePlacesForWorkshopRegistration

```
USE [zakrzews_a]
GO

/****** Object:  UserDefinedFunction [dbo].[
    freePlacesForWorkshopRegistration]    Script Date:
    2017-01-26 00:15:41 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO



--freePlacesForWorkshopRegistration
CREATE FUNCTION [dbo].[freePlacesForWorkshopRegistration]
     (@WorkRegID int)
RETURNS int
AS
BEGIN
                    DECLARE @allPlaces int;
                    SET @allPlaces = (SELECT NumberOfPlaces
                        FROM Workshop_Registration WHERE
                        WorkRegID = @WorkRegID)

                    DECLARE @takenPlaces int;
                    SET @takenPlaces = (SELECT COUNT(*) FROM
                        Which_Workshop WHERE WorkRegID =
                        @WorkRegID)

                    RETURN (@allPlaces - @takenPlaces)
END


GO
```

7.5 dbo.numberOfPlacesNotWorkshoped

```
USE [zakrzews_a]
GO

/****** Object:  UserDefinedFunction [dbo].[
    numberOfPlacesNotWorkshoped]    Script Date:
    2017-01-26 00:16:20 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE FUNCTION [dbo].[numberOfPlacesNotWorkshoped] (
    @DayID int)
RETURNS int
AS
BEGIN
                    DECLARE @allPlaces int;
                    SET @allPlaces = (      SELECT
                        NumberOfPlaces
                            FROM Conference
                            WHERE ConfID = (SELECT ConfID
                            FROM Days
                            WHERE DayID = @DayID));

                    DECLARE @takenPlaces int;
                    SET @takenPlaces = (SELECT SUM(
                        NumberOfPlaces)
                            FROM Workshop_Lesson WL
                            WHERE DayID = @DayID);

                    RETURN (@allPlaces - ISNULL(@takenPlaces,
                         0))
END

GO
```

7.6 dbo.numberOfStudentsInRegistration

```
USE [zakrzews_a]
GO

/****** Object:  UserDefinedFunction [dbo].[
    numberOfStudentsInRegistration]    Script Date:
    2017-01-26 00:16:53 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE FUNCTION [dbo].[numberOfStudentsInRegistration](
    @RegistrationID int)
RETURNS int
AS
BEGIN

                    DECLARE @StudentsNumber int = ( SELECT
                        COUNT(*)
                    FROM Student S
                    INNER JOIN Participant P ON P.
                        ParticipantID = S.ParticipantID
                    INNER JOIN Which_Day WD ON WD.
                        ParticipantID = P.ParticipantID
                    INNER JOIN Day_Registration DR ON DR.
                        DayRegID = WD.DayRegID
                    WHERE RegistrationID = @RegistrationID)

                    RETURN (@StudentsNumber)
END
GO
```

7.7 dbo.registrationCost

```
USE [zakrzews_a]
GO

/****** Object:  UserDefinedFunction [dbo].[
    registrationCost]    Script Date: 2017-01-26 00:17:19
    ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO


CREATE FUNCTION [dbo].[registrationCost] (@RegistrationID
    int)
RETURNS float
AS
BEGIN

        DECLARE @regDate date = (
                SELECT RegistrationDate from Registration
                WHERE RegistrationID = @RegistrationID
        )

        DECLARE @discount float = (
                SELECT D.Discount FROM Discount as D
                WHERE D.StartDate <= @regDate AND D.
                    EndDate >= @regDate AND D.ConfID IN (
                        SELECT Da.ConfID FROM Days as Da
                        WHERE Da.DayID IN (
                                SELECT DR.DayID FROM
                                    Day_Registration as DR
                                WHERE DR.RegistrationID =
                                    @RegistrationID
                        )
                )
        )
        IF(@discount IS NULL)
                BEGIN
                ;SET @discount = 0
                END

        DECLARE @dayCost float =(1-@discount)*(
                SELECT SUM(PRICE) FROM(
```

```
                SELECT /*DR.DayID as ID, */((DR.
                    NumberOfPlaces-dbo.
                    numberOfStudentsInRegistration
                    (@RegistrationID))*D.DayPrice
                +dbo.
                    numberOfStudentsInRegistration
                    (@RegistrationID)*(1-C.
                    StudentDiscount)*D.DayPrice)
                    as PRICE FROM Day_Registration
                     as DR
                INNER JOIN Days as D
                ON D.DayID = DR.DayID
                INNER JOIN Conference as C
                ON C.ConfID = D.ConfID
                WHERE DR.RegistrationID =
                    @RegistrationID
        ) AS ID
        --GROUP BY PRICE
)

DECLARE @workshopCost float = (
        SELECT SUM(PRICE) FROM(
                SELECT WR.NumberOfPlaces * WL.
                    Price as PRICE FROM
                    Workshop_Registration as WR
                INNER JOIN Workshop_Lesson as WL
                ON WL.LessonID = WR.LessonID
                INNER JOIN Day_Registration as DR
                ON DR.DayRegID = WR.DayRegID
                WHERE DR.RegistrationID =
                    @RegistrationID
        ) AS ID
        --GROUP BY PRICE
)

IF(@workshopCost IS NULL)
        BEGIN
        ;SET @workshopCost = 0
        END
IF(@dayCost IS NULL)
        BEGIN
        ;SET @dayCost = 0
        END

RETURN(@dayCost + @workshopCost)
```

```
END
GO
```

7.8 dbo.workshopCollision

```
USE [zakrzews_a]
GO

/****** Object:  UserDefinedFunction [dbo].[
    workshopCollision]    Script Date: 2017-01-26 00:17:37
     ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO


--workshop collision
CREATE FUNCTION [dbo].[workshopCollision] (@lessonID1 int
    , @lessonID2 int)
RETURNS bit
AS
BEGIN
                DECLARE @startTime1 datetime;
                DECLARE @endTime1 datetime;
                DECLARE @startTime2 datetime;
                DECLARE @endTime2 datetime;

                SET @startTime1 = (SELECT startHour FROM
                    Workshop_Lesson WHERE LessonID =
                    @lessonID1)
                SET @endTime1 = (SELECT EndHour FROM
                    Workshop_Lesson WHERE LessonID =
                    @lessonID1)
                SET @startTime2 = (SELECT startHour FROM
                    Workshop_Lesson WHERE LessonID =
                    @lessonID2)
                SET @endTime2 = (SELECT EndHour FROM
                    Workshop_Lesson WHERE LessonID =
                    @lessonID2)

                DECLARE @collision bit;
                IF((@startTime1 > @startTime2 AND
                    @startTime1 < @endTime2) OR (@endTime1
                     > @startTime2 AND @endTime1 <
                    @endTime2))
                        SET @collision = 1
                ELSE
```

```
                    SET @collision = 0

                RETURN @collision

END

GO
```

8 Generowanie danych

Przy uzyciu RedGate SQL Data Generator wygenerowalismy
recordy w tablicach Workshop_Name,
Conference, Client, Company, Participant oraz Student.
W pozostalych uzywalismy nastepujacych fragmentow kodu.

8.1 Dni

```
DECLARE @ILEKONF INT = 72
DECLARE @ITERATORKONFERENCJEID INT = 1
DECLARE @IDDNIAKONFERENCJI INT =1
DECLARE @ITERATORDNIKONFERENCJI INT =1
DECLARE @ILEDNIWKONFERENCJI INT;
DECLARE @DATADNIA DATE ;

WHILE(@ITERATORKONFERENCJEID <= @ILEKONF)
BEGIN
        SET @ILEDNIWKONFERENCJI  = ISNULL((SELECT
            DATEDIFF(DAY, StartDate , EndDate)+1
        FROM dbo.Conference
        WHERE ConfID = @ITERATORKONFERENCJEID) ,0)
        SET @DATADNIA  = (SELECT StartDate
          FROM dbo.Conference
          WHERE ConfID = @ITERATORKONFERENCJEID)

        WHILE(@ITERATORDNIKONFERENCJI <=
            @ILEDNIWKONFERENCJI)
        BEGIN

                DECLARE @CENA int =  ABS(Checksum(NewID()
                    ) % 50) + 50
                INSERT INTO Days(ConfID, Date, DayPrice)
                VALUES(@ITERATORKONFERENCJEID ,@DATADNIA
                    , @CENA)


                SET @IDDNIAKONFERENCJI =
                    @IDDNIAKONFERENCJI + 1
                SET @ITERATORDNIKONFERENCJI =
                    @ITERATORDNIKONFERENCJI +1
                SET @DATADNIA = DATEADD(DAY ,1 ,@DATADNIA
                    )
        END
        SET @ITERATORDNIKONFERENCJI = 1
        SET @ITERATORKONFERENCJEID =
            @ITERATORKONFERENCJEID + 1
END
```

8.2 Discount

```
SET @ILEKONF INT = 72
DECLARE @ITERATORPOKONFID INT = 1
WHILE(@ITERATORPOKONFID <= @ILEKONF)
BEGIN
        DECLARE @DATE DATE  = ( SELECT StartDate
                        FROM dbo.Conference
                        WHERE ConfID = @ITERATORPOKONFID)

        DECLARE @DISCOUNT FLOAT = 0.3
        DECLARE @TROJKA INT = 0
        DECLARE @STARTDATE DATE = DATEADD(dd, -93, @DATE)
        WHILE(@TROJKA < 3)
        BEGIN

                DECLARE @ENDDATE DATE = DATEADD(dd, 30,
                    @STARTDATE)
                INSERT dbo.Discount
                        ( ConfID ,
                          Discount ,
                          StartDate ,
                          EndDate
                        )
                VALUES  ( @ITERATORPOKONFID,
                          @DISCOUNT,
                          @STARTDATE,
                          @ENDDATE
                        )
                SET @DISCOUNT = @DISCOUNT + 0.2
                SET @STARTDATE = DATEADD(dd, 31,
                    @STARTDATE)
                SET @TROJKA = @TROJKA + 1
        END
        SET @ITERATORPOKONFID = @ITERATORPOKONFID + 1
END
```

8.3 WorkshopLesson

```
DECLARE @ILOSCDNI INT = 0
DECLARE @IDDNIA INT = 1254
DECLARE @ILOSCWARSZTATOWWDNIU INT = 0
DECLARE @IDWARSZTATU INT
DECLARE @STARTDATE DATETIME
DECLARE @ENDDATE DATETIME
DECLARE @DATE DATE
DECLARE @CENA INT
WHILE(@ILOSCDNI < 186)
BEGIN
        SET @ILOSCWARSZTATOWWDNIU = 0
        SET @DATE = (SELECT Date FROM Days WHERE DayID =
            @IDDNIA)
        SET @STARTDATE = CONVERT(DATETIME, @DATE)
        SET @ENDDATE = CONVERT(DATETIME, @DATE)
        SET @STARTDATE = DATEADD(hh, 9, @STARTDATE)
        SET @ENDDATE = DATEADD(hh, 10, @STARTDATE)

        WHILE(@ILOSCWARSZTATOWWDNIU < 4)
        BEGIN
                SET @IDWARSZTATU = ABS(Checksum(NewID())
                    % 20)+1
                SET @CENA = ABS(Checksum(NewID()) % 20)
                    +20
                INSERT dbo.Workshop_Lesson
                        ( WorkshopID ,
                          DayID ,
                          NumberOfPlaces ,
                          Price ,
                          StartHour ,
                          EndHour
                        )
                VALUES  ( @IDWARSZTATU , -- WorkshopID -
                    int
                          @IDDNIA , -- DayID - int
                          40 , -- NumberOfPlaces - int
                          @CENA , -- Price - money
                          @STARTDATE , -- StartHour -
                              datetime
                          @ENDDATE -- EndHour - datetime
                        )

                SET @STARTDATE = DATEADD(hh, 2,
                    @STARTDATE)
```

```
                    SET @ENDDATE = DATEADD(hh, 3, @STARTDATE)
                    SET @ILOSCWARSZTATOWWDNIU =
                        @ILOSCWARSZTATOWWDNIU + 1
            END
            SET @ILOSCDNI = @ILOSCDNI + 1
            SET @IDDNIA = @IDDNIA + 1
END
```

8.4 Registration

```
DECLARE @ILOSCKONF INT = 72
DECLARE @ILOSCKLIENTOW INT = 600
DECLARE @DATE DATE
DECLARE @PAYDATE DATE
DECLARE @PAIED INT
DECLARE @INCREMENTATOR INT
WHILE(@ILOSCKONF > 0)
BEGIN
        SET @INCREMENTATOR = 0
        WHILE(@INCREMENTATOR < 8)
        BEGIN
                SET @DATE = DATEADD(dd, -20, (SELECT
                    StartDate FROM dbo.Conference WHERE
                    ConfID = @ILOSCKONF))
                SET @PAYDATE = DATEADD(dd, -10, (SELECT
                    StartDate FROM dbo.Conference WHERE
                    ConfID = @ILOSCKONF))
                SET @PAIED = ABS(Checksum(NewID()) % 10)
                    +1
                IF(@PAIED <= 1)
                BEGIN
                        INSERT dbo.Registration
                                ( ConfID ,
                                  ClientID ,
                                  RegistrationDate ,
                                  PayDate
                                )
                        VALUES  ( @ILOSCKONF, -- ConfID -
                            int
                                  @ILOSCKLIENTOW, --
                                      ClientID - int
                                  @Date, --
                                      RegistrationDate -
                                      date
                                  NULL -- PayDate - date
                                )
                END
                IF(@PAIED > 1)
                BEGIN
                        INSERT dbo.Registration
                                ( ConfID ,
                                  ClientID ,
                                  RegistrationDate ,
                                  PayDate
```

```
                                    )
                                VALUES ( @ILOSCKONF , -- ConfID
                                    - int
                                        @ILOSCKLIENTOW , --
                                            ClientID - int
                                        @DATE , --
                                            RegistrationDate -
                                            date
                                        @PAYDATE -- PayDate -
                                            date
                                    )
                    END
                    SET @INCREMENTATOR = @INCREMENTATOR + 1
                    SET @ILOSCKLIENTOW = @ILOSCKLIENTOW - 1
            END
            SET @ILOSCKONF = @ILOSCKONF - 1
END
```

8.5 DayRegistration

```
DECLARE @REGISTRATIONID INT = 1576
DECLARE @ILOSCDNI INT
DECLARE @DNIILOSC INT
DECLARE @CONFID INT
DECLARE @DAYID INT = 1254
DECLARE @DATE DATE
WHILE(@REGISTRATIONID > 1000)
BEGIN
        SET @CONFID = (SELECT ConfID FROM dbo.
            Registration WHERE RegistrationID =
            @REGISTRATIONID)
        SET @ILOSCDNI = (SELECT COUNT(*) FROM DAYS WHERE
            ConfID = @CONFID)
        SET @ILOSCDNI = @ILOSCDNI
        SET @DNIILOSC = 0

        WHILE(@ILOSCDNI > 0)
        BEGIN
                SET @DATE = (SELECT StartDate FROM
                    Conference WHERE ConfID = @CONFID)
                SET @DATE = DATEADD(dd, @DNIILOSC, @DATE)
                SET @DAYID = (SELECT DayID FROM Days
                    WHERE Date = @Date AND ConfID =
                    @CONFID)
                INSERT dbo.Day_Registration
                        ( RegistrationID ,
                          DayID ,
                          NumberOfPlaces
                        )
                VALUES  ( @REGISTRATIONID , --
                    RegistrationID - int
                          @DAYID , -- DayID - int
                          20  -- NumberOfPlaces - int
                        )
                SET @ILOSCDNI = @ILOSCDNI - 1
                SET @DNIILOSC = @DNIILOSC + 1
        END

        SET @REGISTRATIONID = @REGISTRATIONID - 1

END
```

8.6 WorkshopRegistration

```
DECLARE @LESSONID INT = 2489
DECLARE @DAYREGID INT
DECLARE @DAYID INT
DECLARE @DAYID2 INT
DECLARE @DATE DATE
WHILE(@LESSONID < 3233)
BEGIN

        SET @DAYID = (SELECT DayID FROM dbo.
            Workshop_Lesson WHERE LessonID = @LESSONID)
        SET @DAYREGID = 3977
        WHILE(@DAYREGID < 5465)
        BEGIN

                SET @DATE = CONVERT(DATE,(SELECT
                    StartHour FROM dbo.Workshop_Lesson
                    WHERE LessonID = @LESSONID))
                SET @DATE = DATEADD(dd, -10, @DATE)
                SET @DAYID2 = (SELECT DAYID FROM dbo.
                    Day_Registration WHERE DayRegID =
                    @DAYREGID)
                IF(@DAYID = @DAYID2)
                BEGIN
                        INSERT dbo.Workshop_Registration
                                ( DayRegID ,
                                  LessonID ,
                                  Date ,
                                  NumberOfPlaces
                                )
                        VALUES  ( @DAYREGID , -- DayRegID
                            - int
                                  @LESSONID , -- LessonID
                                      - int
                                  @DATE, -- Date - date
                                  5 -- NumberOfPlaces -
                                      int
                                )
                END
                SET @DAYREGID = @DAYREGID + 1
        END
        SET @LESSONID = @LESSONID + 1
END
```

8.7 WhichWorkshop

```
DECLARE @PARTICIPANTID INT = 1
DECLARE @WORKREGID INT = 1001
DECLARE @ITERATOR INT
WHILE(@PARTICIPANTID < 11905)
BEGIN
        SET @ITERATOR = 0
        WHILE(@ITERATOR < 2)
        BEGIN

                INSERT dbo.Which_Workshop
                        ( ParticipantID , WorkRegID )
                VALUES  ( @PARTICIPANTID, --
                    ParticipantID - int
                        @WORKREGID -- WorkRegID - int
                        )

                SET @ITERATOR = @ITERATOR + 1
                SET @PARTICIPANTID = @PARTICIPANTID + 1
        END
        SET @WORKREGID = @WORKREGID + 1
END
```

8.8 WhichDay

```
DECLARE @PARTICIPANTID INT = 1
DECLARE @DAYREGID INT
WHILE(@PARTICIPANTID < 11905)
BEGIN
        SET @DAYREGID = (SELECT DayRegID FROM dbo.
            Workshop_Registration WHERE WorkRegID = (
            SELECT WorkRegID FROM dbo.Which_Workshop WHERE
              ParticipantID = @PARTICIPANTID))

        INSERT dbo.Which_Day
                ( ParticipantID , DayRegID )
        VALUES  ( @PARTICIPANTID, -- ParticipantID - int
                  @DAYREGID  -- DayRegID - int
                  )
        SET @PARTICIPANTID = @PARTICIPANTID + 1
END
```