Autor: Paweł Dziwisz

Tomasz Dębosz

Ihor-Severyn Dolhopolov

Piotr Kotarba

**2. Wypełnienie tabeli przykładowymi danymi**

insert into wycieczki (nazwa, kraj, data, opis, liczba\_miejsc)

values ('Wycieczka do Paryza','Francja','2012-01-01','Ciekawa wycieczka ...',3)

insert into wycieczki (nazwa, kraj, data, opis, liczba\_miejsc)

values ('Wycieczka Moskwa','Rosja','2022-07-01','Plac Czerwony tour',77)

insert into wycieczki (nazwa, kraj, data, opis, liczba\_miejsc)

values ('Na Wawelu','Polska','2025-01-10','Krolowie polscy',25)

insert into wycieczki (nazwa, kraj, data, opis, liczba\_miejsc)

values ('Piękny Kraków','Polska','2013-02-03','Najciekawa wycieczka ...',2)

insert into osoby (imie, nazwisko, pesel, kontakt)

values('Anna', 'Nowakowicz', '16795679', 'tel: 7541')

insert into osoby (imie, nazwisko, pesel, kontakt)

values('Ewa', 'Janowicz', '543200', 'tel: 2312, dzwonić po 22.00')

insert into osoby (imie, nazwisko, pesel, kontakt)

values('Marian', 'Piszczek', '12345678', 'tel: 2312, dzwonić po 18.00')

insert into osoby (imie, nazwisko, pesel, kontakt)

values('Adam', 'Kowalski', '87654321', 'tel: 6623')

insert into rezerwacje(id\_wycieczki, id\_osoby, status)

values (1,1,'P')

insert into rezerwacje(id\_wycieczki, id\_osoby, status)

values (2,2, 'N')

**3. Tworzenie widoków**

1. v\_wycieczki\_osoby(kraj,data, nazwa\_wycieczki, imie, nazwisko,status\_rezerwacji)

create view v\_wycieczki\_osoby

as

select r.id\_osoby, w.id\_wycieczki,

r.nr\_rezerwacji, liczba\_miejsc, kraj, data,

w.nazwa as nazwa\_wycieczki, imie,

nazwisko, status as status\_rezerwacji

from wycieczki w

join rezerwacje r

on w.id\_wycieczki = r.id\_wycieczki

join osoby o

on o.id\_osoby = r.id\_osoby

1. v\_wycieczki\_osoby\_potwierdzone (kraj,data, nazwa\_wycieczki, imie, nazwisko, status\_rezerwacji)

create view v\_wycieczki\_osoby\_potwierdzone

as

select kraj, data, nazwa\_wycieczki,

imie, nazwisko, status\_rezerwacji

from v\_wycieczki\_osoby

where status\_rezerwacji in ('P', 'Z')

1. v\_wycieczki\_przyszle (kraj,data, nazwa\_wycieczki, imie, nazwisko,status\_rezerwacji)

create view v\_wycieczki\_przyszle

as

select id\_wycieczki, kraj, data,

nazwa\_wycieczki, imie,

nazwisko, status\_rezerwacji

from v\_wycieczki\_osoby

where datediff(day, sysdatetime(), data) >= 1

1. v\_wycieczki\_miejsca(kraj, data, nazwa\_wycieczki, liczba\_miejsc, liczba\_wolnych\_miejsc)

create view v\_wycieczki\_miejsca

as

select w.id\_wycieczki, kraj, data,

w.nazwa as nazwa\_wycieczki, liczba\_miejsc,

liczba\_miejsc - count(status)

as liczba\_wolnych\_miejsc

from wycieczki as w

left join rezerwacje as r

on w.id\_wycieczki = r.id\_wycieczki

and status != 'A'

group by w.id\_wycieczki, kraj,

data, w.nazwa, liczba\_miejsc

1. v\_dostepne\_wycieczki(kraj, data, nazwa\_wycieczki, liczba\_miejsc, liczba\_wolnych\_miejsc)

create view [dbo].[v\_dostepne\_wycieczki]

as

select v\_w\_m.id\_wycieczki, v\_w\_m.kraj,

v\_w\_m.data, v\_w\_m.nazwa\_wycieczki,

liczba\_miejsc, liczba\_wolnych\_miejsc

from v\_wycieczki\_miejsca as v\_w\_m

where liczba\_wolnych\_miejsc >= 1

and v\_w\_m.data > sysdatetime()

1. v\_rezerwacje\_do\_anulowania

create view [dbo].[v\_rezerwacje\_do\_anulowania]

as

select nr\_rezerwacji, nazwa\_wycieczki,

id\_osoby, imie, nazwisko, status\_rezerwacji

from v\_wycieczki\_osoby as v\_o\_s

where v\_o\_s.data > sysdatetime()

and datediff(day,v\_o\_s.data,sysdatetime()) < 7

and status\_rezerwacji = 'N'

**4. Tworzenie procedur pobierających danę**

1. p\_uczestnicy\_wycieczki (id\_wycieczki)

create procedure p\_uczestnicy\_wycieczki

@id\_wycieczki int

as

begin try

if @id\_wycieczki is null

raiserror('argument nie może przyjmować

wartości null', 16, 1)

if not exists

(

select \* from wycieczki

where wycieczki.id\_wycieczki =

@id\_wycieczki

)

raiserror('wycieczka o podanym id nie

istnieje', 16, 1)

select \* from v\_wycieczki\_osoby

where id\_wycieczki = @id\_wycieczki

end try

begin catch

print error\_message()

declare @err\_msg varchar(100)

set @err\_msg = error\_message()

raiserror(@err\_msg, 16, 1)

end catch

1. p\_rezerwacje\_osoby(id\_osoby)

ALTER procedure p\_rezerwacje\_osoby

@id\_osoby int

as

begin try

if @id\_osoby is null

raiserror('argument nie może przyjmować

wartości null', 16, 1)

if not exists

(

select \* from osoby

where id\_osoby = @id\_osoby

)

raiserror('osoba o podanym id nie istnieje',

16 ,1)

select \* from v\_wycieczki\_osoby

where id\_osoby = @id\_osoby

end try

begin catch

print error\_message()

declare @err\_msg varchar(100)

set @err\_msg = error\_message()

raiserror(@err\_msg, 16, 1)

end catch

1. p\_przyszle\_rezerwacje\_osoby(id\_osoby)

create procedure p\_przyszle\_rezerwacje\_osoby

@id\_osoby int

as

begin try

if @id\_osoby is null

raiserror('argument nie może przyjmować

wartości null', 16, 1)

select id\_osoby, kraj, data, nazwa\_wycieczki,

imie, nazwisko, status\_rezerwacji

from v\_wycieczki\_osoby

where datediff(day, sysdatetime(), data) >= 1

and id\_osoby = @id\_osoby

end try

begin catch

print error\_message()

declare @err\_msg varchar(100)

set @err\_msg = error\_message()

raiserror(@err\_msg, 16, 1)

end catch

1. p\_dostepne\_wycieczki(kraj, data\_od, data\_do)

create procedure p\_dostepne\_wycieczki

@kraj varchar(50),

@data\_od date,

@data\_do date

as

begin try

if @kraj is null

or @data\_od is null

or @data\_do is null

raiserror('argumenty nie mogą przyjmować

wartości null', 16, 1)

if not exists

(

select \* from v\_dostepne\_wycieczki

as v\_d\_w

where v\_d\_w.kraj = @kraj

)

raiserror('kraj o podanej nazwie nie

istnieje', 16, 1)

if sysdatetime() > @data\_od

or sysdatetime() > @data\_do

raiserror('argumenty data\_od oraz data\_do nie

mogą być mniejsze niż aktualna data', 16, 1)

if @data\_od > @data\_do

raiserror('argument data\_do nie może być

mniejszy niż data\_od', 16, 1)

select \*

from v\_dostepne\_wycieczki as v\_d\_w

where v\_d\_w.kraj = @kraj

and v\_d\_w.data < @data\_do

and v\_d\_w.data >= @data\_od

end try

begin catch

print error\_message()

declare @err\_msg varchar(100)

set @err\_msg = error\_message()

raiserror(@err\_msg, 16, 1)

end catch

5. Tworzenie procedur modyfikujących danę

1. p\_dodaj\_rezerwacje(id\_wycieczki, id\_osoby)

ALTER procedure p\_dodaj\_rezerwacje

@id\_wycieczki int,

@id\_osoby int

as

begin try

if @id\_wycieczki is null

or @id\_osoby is null

raiserror('argumenty nie mogą przyjmować

wartości null', 16, 1)

if not exists

(

select \* from wycieczki as w

where w.id\_wycieczki = @id\_wycieczki

)

raiserror('wycieczka o podanym id nie istnieje

na liscie wycieczek', 16 ,1)

if not exists

(

select \* from v\_dostepne\_wycieczki as

v\_d\_w

where v\_d\_w.id\_wycieczki = @id\_wycieczki

)

raiserror('wycieczka o podanym id nie istnieje

na liscie dostepnych wycieczek', 16 ,1)

if not exists

(

select \* from osoby

where osoby.id\_osoby = @id\_osoby

)

raiserror('osoba o podanym id nie istnieje',

16 ,1)

insert rezerwacje(id\_wycieczki,

id\_osoby,status)

values(@id\_wycieczki, @id\_osoby, 'N')

insert rez\_log(id\_rezerwacji, data, status)

values(@@identity, getdate(), 'N')

end try

begin catch

print error\_message()

declare @err\_msg varchar(100)

set @err\_msg = error\_message()

raiserror(@err\_msg, 16, 1)

end catch

1. p\_zmien\_status\_rezerwacji(id\_rezerwacji, status)

create procedure p\_zmien\_status\_rezerwacji

@id\_rezerwacji int,

@status char(1)

as

begin try

if @id\_rezerwacji is null

or @status is null

raiserror('argumenty nie mogą przyjmować

wartości null', 16, 1)

if not exists

(

select \* from rezerwacje

where nr\_rezerwacji = @id\_rezerwacji

)

raiserror('rezerwacja o podanym id nie

istnieje na liscie rezerwacji', 16 ,1)

if @status not in ('N', 'P', 'Z', 'A')

raiserror('podano status z poza zakresu

mozliwych wartosci', 16 ,1)

declare @id\_wycieczki int;

set @id\_wycieczki =

(

select id\_wycieczki

from rezerwacje

where nr\_rezerwacji = @id\_rezerwacji

)

declare @liczba\_wolnych\_miejsc int

set @liczba\_wolnych\_miejsc =

(

select liczba\_wolnych\_miejsc

from v\_wycieczki\_miejsca as l\_w\_m

where l\_w\_m.id\_wycieczki = @id\_wycieczki

)

if @status in ('N', 'P', 'Z')

and @liczba\_wolnych\_miejsc < 1

raiserror('zmiana tej rezerwacji jest nie

mozliwa, brak miejsca', 16 ,1)

update rezerwacje

set status = @status

where nr\_rezerwacji = @id\_rezerwacji

end try

begin catch

print error\_message()

declare @err\_msg varchar(100)

set @err\_msg = error\_message()

raiserror(@err\_msg, 16, 1)

end catch

1. p\_zmien\_liczbe\_miejsc(id\_wycieczki, liczba\_miejsc)

create procedure p\_zmien\_liczbe\_miejsc

@id\_wycieczki int,

@liczba\_miejsc int

as

begin try

if @id\_wycieczki is null

or @liczba\_miejsc is null

raiserror('argumenty nie mogą przyjmować

wartości null', 16, 1)

if @liczba\_miejsc < 0

raiserror('argument liczba\_miejsc nie moze byc

mniejszy od zera', 16, 1)

if not exists

(

select \* from wycieczki as w

where w.id\_wycieczki = @id\_wycieczki

)

raiserror('wycieczka o podanym id nie

istnieje', 16, 1)

declare @liczba\_miejsc\_zarezerwowanych int

set @liczba\_miejsc\_zarezerwowanych =

(

select liczba\_wolnych\_miejsc

from v\_wycieczki\_miejsca as v\_w\_m

where v\_w\_m.id\_wycieczki = @id\_wycieczki

)

if @liczba\_miejsc\_zarezerwowanych <

@liczba\_miejsc

raiserror('zmniejszenie liczby miejsc nie jest

mozliwe', 16, 1)

update wycieczki

set liczba\_miejsc = @liczba\_miejsc

where id\_wycieczki = @id\_wycieczki

end try

begin catch

print error\_message()

declare @err\_msg varchar(100)

set @err\_msg = error\_message()

raiserror(@err\_msg, 16, 1)

end catch

6. Tworzenie tabeli dziennikującej zmiany statusu rezerwacji

1. stworzenie tabeli rez\_log

create table rez\_log(

id int not null identity(1,1),

id\_rezerwacji int not null,

data datetime not null, status char(1) not null,

primary key(id),

foreign key(id\_rezerwacji) references Rezerwacje(nr\_rezerwacji),

check (status in('N','P','Z','A'))

)

1. modyfikacja procedury p\_dodaj\_rezerwacje o możliwość dodawania wpisów do tabeli rez\_log

alter procedure p\_dodaj\_rezerwacje

@id\_wycieczki int,

@id\_osoby int

as

begin try

if @id\_wycieczki is null

or @id\_osoby is null

raiserror('argumenty nie mogą przyjmować

wartości null', 16, 1)

if not exists

(

select \* from wycieczki as w

where w.id\_wycieczki = @id\_wycieczki

)

raiserror('wycieczka o podanym id nie istnieje

na liscie wycieczek', 16 ,1)

if not exists

(

select \* from v\_dostepne\_wycieczki as

v\_d\_w

where v\_d\_w.id\_wycieczki = @id\_wycieczki

)

raiserror('wycieczka o podanym id nie istnieje

na liscie dostepnych wycieczek', 16 ,1)

if not exists

(

select \* from osoby

where osoby.id\_osoby = @id\_osoby

)

raiserror('osoba o podanym id nie istnieje',

16 ,1)

insert rezerwacje(id\_wycieczki,

id\_osoby,status)

values(@id\_wycieczki, @id\_osoby, 'N')

insert rez\_log(id\_rezerwacji, data, status)

values(@@identity, getdate(), 'N')

end try

begin catch

print error\_message()

declare @err\_msg varchar(100)

set @err\_msg = error\_message()

raiserror(@err\_msg, 16, 1)

end catch

1. modyfikacja procedury p\_zmien\_status\_rezerwacji o możliwość dodawania wpisów do tabeli rez\_log

alter procedure p\_zmien\_status\_rezerwacji

@id\_rezerwacji int,

@status char(1)

as

begin try

if @id\_rezerwacji is null

or @status is null

raiserror('argumenty nie mogą przyjmować

wartości null', 16, 1)

if not exists

(

select \* from rezerwacje

where nr\_rezerwacji = @id\_rezerwacji

)

raiserror('rezerwacja o podanym id nie

istnieje na liscie rezerwacji', 16 ,1)

if @status not in ('N', 'P', 'Z', 'A')

raiserror('podano status z poza zakresu

mozliwych wartosci', 16 ,1)

declare @id\_wycieczki int;

set @id\_wycieczki =

(

select id\_wycieczki

from rezerwacje

where nr\_rezerwacji = @id\_rezerwacji

)

declare @liczba\_wolnych\_miejsc int

set @liczba\_wolnych\_miejsc =

(

select liczba\_wolnych\_miejsc

from v\_wycieczki\_miejsca as l\_w\_m

where l\_w\_m.id\_wycieczki = @id\_wycieczki

)

if @status in ('N', 'P', 'Z')

and @liczba\_wolnych\_miejsc < 1

raiserror('zmiana tej rezerwacji jest nie

mozliwa, brak miejsca', 16 ,1)

update rezerwacje

set status = @status

where nr\_rezerwacji = @id\_rezerwacji

insert rez\_log(id\_rezerwacji, data, status)

values(@id\_rezerwacji, getdate(), @status)

end try

begin catch

print error\_message()

declare @err\_msg varchar(100)

set @err\_msg = error\_message()

raiserror(@err\_msg, 16, 1)

end catch

7. Zmiana struktury bazy danych, dodanie redundantnego pola liczba\_wolnych\_miejsc w tabeli wycieczki

1. Tworzenie redundantnego pola liczba\_wolnych\_miejsc w tabeli wycieczki

begin transaction

alter table dbo.wycieczki add

liczba\_wolnych\_miejsc int null

alter table dbo.wycieczki

set

(

lock\_escalation = table

)

commit

update wycieczki

set liczba\_wolnych\_miejsc = liczba\_miejsc -

(

select count(\*) from rezerwacje

where rezerwacje.id\_wycieczki =

wycieczki.id\_wycieczki

and status != 'A'

)

1. tworzenie procedury aktualizującej pole liczba\_wolnych\_miejsc w tabeli wycieczki

create procedure aktualizacja\_liczby\_wolnych\_miejsc

as

begin try

merge into wycieczki as w

using v\_wycieczki\_miejsca as v\_w\_m

on w.id\_wycieczki = v\_w\_m.id\_wycieczki

when matched then

update

set w.liczba\_wolnych\_miejsc =

v\_w\_m.liczba\_wolnych\_miejsc;

end try

begin catch

print error\_message()

declare @err\_msg varchar(100)

set @err\_msg = error\_message()

raiserror(@err\_msg, 16, 1)

end catch

1. tworzenie kopii widoku v\_dostepne\_wycieczki

create view v\_copy\_dostepne\_wycieczki

as

select id\_wycieczki, kraj,

data, nazwa as nazwa\_wycieczki,

liczba\_miejsc, liczba\_wolnych\_miejsc

from wycieczki

where liczba\_wolnych\_miejsc >= 1

and data > sysdatetime()

1. tworzenie kopii procedury p\_zmien\_status\_rezerwacji

create procedure p\_copy\_zmien\_status\_rezerwacji

@id\_rezerwacji int,

@status char(1)

as

begin try

if @id\_rezerwacji is null

or @status is null

raiserror('argumenty nie mogą przyjmować

wartości null', 16, 1)

if not exists

(

select \* from rezerwacje

where nr\_rezerwacji = @id\_rezerwacji

)

raiserror('rezerwacja o podanym id nie

istnieje na liscie rezerwacji', 16 ,1)

if @status not in ('N', 'P', 'Z', 'A')

raiserror('podano status z poza zakresu

mozliwych wartosci', 16 ,1)

declare @id\_wycieczki int;

set @id\_wycieczki =

(

select id\_wycieczki

from rezerwacje

where nr\_rezerwacji = @id\_rezerwacji

)

declare @liczba\_wolnych\_miejsc int

set @liczba\_wolnych\_miejsc =

(

select liczba\_wolnych\_miejsc

from wycieczki as w

where w.id\_wycieczki = @id\_wycieczki

)

if @status in ('N', 'P', 'Z')

and @liczba\_wolnych\_miejsc < 1

raiserror('zmiana tej rezerwacji jest nie

mozliwa, brak miejsca', 16 ,1)

update rezerwacje

set status = @status

where nr\_rezerwacji = @id\_rezerwacji

insert rez\_log(id\_rezerwacji, data, status)

values(@id\_rezerwacji, getdate(), @status)

end try

begin catch

print error\_message()

declare @err\_msg varchar(100)

set @err\_msg = error\_message()

raiserror(@err\_msg, 16, 1)

end catch

1. tworzenie kopii procedury p\_zmien\_liczbe\_miejsc

create procedure p\_copy\_zmien\_liczbe\_miejsc

@id\_wycieczki int,

@liczba\_miejsc int

as

begin try

if @id\_wycieczki is null

or @liczba\_miejsc is null

raiserror('argumenty nie mogą przyjmować

wartości null', 16, 1)

if @liczba\_miejsc < 0

raiserror('argument liczba\_miejsc nie moze byc

mniejszy od zera', 16, 1)

if not exists

(

select \* from wycieczki as w

where w.id\_wycieczki = @id\_wycieczki

)

raiserror('wycieczka o podanym id nie

istnieje', 16, 1)

declare @liczba\_miejsc\_zarezerwowanych int

set @liczba\_miejsc\_zarezerwowanych =

(

select liczba\_wolnych\_miejsc

from wycieczki as w

where w.id\_wycieczki = @id\_wycieczki

)

if @liczba\_miejsc\_zarezerwowanych <

@liczba\_miejsc

raiserror('zmniejszenie liczby miejsc nie jest

mozliwe', 16, 1)

update wycieczki

set liczba\_miejsc = @liczba\_miejsc

where id\_wycieczki = @id\_wycieczki

end try

begin catch

print error\_message()

declare @err\_msg varchar(100)

set @err\_msg = error\_message()

raiserror(@err\_msg, 16, 1)

end catch

8. Realizacja zapisywania danych do dziennika rezerwacji przy pomocy triggerów

1. Trigger obsługujący dodanie rezerwacji

create trigger dodanie\_rezerwacji

on rezerwacje

after insert

as

begin try

if (select count(\*) from inserted) > 1

raiserror('dopisywanie kilku wierszy jest

zabronione', 16, 1)

declare @id\_rezerwacji int

set @id\_rezerwacji =

(

select nr\_rezerwacji

from inserted

)

insert rez\_log(id\_rezerwacji, data, status)

values(@id\_rezerwacji, getdate(), 'N')

end try

begin catch

print error\_message()

declare @err\_msg varchar(100)

set @err\_msg = error\_message()

raiserror(@err\_msg, 16, 1)

end catch

1. Trigger obsługujący zmianę statusu

create trigger zmiana\_rezerwacji

on rezerwacje

after update

as

begin try

if (select count(\*) from inserted) > 1

raiserror('modyfikacja kilku wierszy jest

zabroniona', 16, 1)

declare @id\_rezerwacji int

declare @status char(1)

select @id\_rezerwacji = nr\_rezerwacji,

@status = status

from inserted

insert rez\_log(id\_rezerwacji, data, status)

values(@id\_rezerwacji, getdate(), @status)

end try

begin catch

print error\_message()

declare @err\_msg varchar(100)

set @err\_msg = error\_message()

raiserror(@err\_msg, 16, 1)

end catch

1. Trigger zabraniający usunięcia rezerwacji

create trigger usuniecie\_rezerwacji

on rezerwacje

instead of delete

as

begin try

raiserror('usuwanie wierszy jest zabronione',

16, 1)

end try

begin catch

print error\_message()

declare @err\_msg varchar(100)

set @err\_msg = error\_message()

raiserror(@err\_msg, 16, 1)

end catch

1. Wprowadzanie zmian w procedurze p\_dodaj\_rezerwacje

ALTER procedure p\_dodaj\_rezerwacje

@id\_wycieczki int,

@id\_osoby int

as

begin try

if @id\_wycieczki is null

or @id\_osoby is null

raiserror('argumenty nie mogą przyjmować

wartości null', 16, 1)

if not exists

(

select \* from wycieczki as w

where w.id\_wycieczki = @id\_wycieczki

)

raiserror('wycieczka o podanym id nie istnieje

na liscie wycieczek', 16 ,1)

if not exists

(

select \* from v\_dostepne\_wycieczki as

v\_d\_w

where v\_d\_w.id\_wycieczki = @id\_wycieczki

)

raiserror('wycieczka o podanym id nie istnieje

na liscie dostepnych wycieczek', 16 ,1)

if not exists

(

select \* from osoby

where osoby.id\_osoby = @id\_osoby

)

raiserror('osoba o podanym id nie istnieje',

16 ,1)

insert rezerwacje(id\_wycieczki, id\_osoby,status)

values(@id\_wycieczki, @id\_osoby, 'N')

*-- insert rez\_log(id\_rezerwacji, data, status)*

*-- values(@@identity, getdate(), 'N')*

end try

begin catch

print error\_message()

declare @err\_msg varchar(100)

set @err\_msg = error\_message()

raiserror(@err\_msg, 16, 1)

end catch

1. Wprowadzanie zmian w procedurze p\_copy\_zmien\_status\_rezerwacji

ALTER procedure p\_copy\_zmien\_status\_rezerwacji

@id\_rezerwacji int,

@status char(1)

as

begin try

if @id\_rezerwacji is null

or @status is null

raiserror('argumenty nie mogą przyjmować

wartości null', 16, 1)

if not exists

(

select \* from rezerwacje

where nr\_rezerwacji = @id\_rezerwacji

)

raiserror('rezerwacja o podanym id nie

istnieje na liscie rezerwacji', 16 ,1)

if @status not in ('N', 'P', 'Z', 'A')

raiserror('podano status z poza zakresu

mozliwych wartosci', 16 ,1)

declare @id\_wycieczki int;

set @id\_wycieczki =

(

select id\_wycieczki

from rezerwacje

where nr\_rezerwacji = @id\_rezerwacji

)

declare @liczba\_wolnych\_miejsc int

set @liczba\_wolnych\_miejsc =

(

select liczba\_wolnych\_miejsc

from wycieczki as w

where w.id\_wycieczki = @id\_wycieczki

)

if @status in ('N', 'P', 'Z')

and @liczba\_wolnych\_miejsc < 1

raiserror('zmiana tej rezerwacji jest nie

mozliwa, brak miejsca', 16 ,1)

update rezerwacje

set status = @status

where nr\_rezerwacji = @id\_rezerwacji

*-- insert rez\_log(id\_rezerwacji, data, status)*

*-- values(@id\_rezerwacji, getdate(), @status)*

end try

begin catch

print error\_message()

declare @err\_msg varchar(100)

set @err\_msg = error\_message()

raiserror(@err\_msg, 16, 1)

end catch

9. Realizacja przy pomocy triggerów obsługi redundantnego pola liczba\_wolnych\_miejsc

1. Trigger obsługujący dodanie rezerwacji

alter trigger dodanie\_rezerwacji

on rezerwacje

instead of insert

as

begin try

if (select count(\*) from inserted) > 1

raiserror('dopisywanie kilku wierszy jest

zabronione', 16, 1)

declare @id\_rezerwacji int

declare @id\_wycieczki int

declare @id\_osoby int

select @id\_rezerwacji = nr\_rezerwacji,

@id\_wycieczki = id\_wycieczki,

@id\_osoby = id\_osoby

from inserted

if @id\_wycieczki is null

or @id\_osoby is null

raiserror('argumenty nie mogą przyjmować

wartości null', 16, 1)

if not exists

(

select \* from wycieczki as w

where w.id\_wycieczki = @id\_wycieczki

)

raiserror('wycieczka o podanym id nie istnieje

na liscie wycieczek', 16 ,1)

if not exists

(

select \* from v\_dostepne\_wycieczki as

v\_d\_w

where v\_d\_w.id\_wycieczki = @id\_wycieczki

)

raiserror('wycieczka o podanym id nie istnieje

na liscie dostepnych wycieczek', 16 ,1)

if not exists

(

select \* from osoby

where osoby.id\_osoby = @id\_osoby

)

raiserror('osoba o podanym id nie istnieje',

16 ,1)

insert rezerwacje(id\_wycieczki,

id\_osoby,status)

values(@id\_wycieczki, @id\_osoby, 'N')

insert rez\_log(id\_rezerwacji, data, status)

values(@id\_rezerwacji, getdate(), 'N')

end try

begin catch

print error\_message()

declare @err\_msg varchar(100)

set @err\_msg = error\_message()

raiserror(@err\_msg, 16, 1)

end catch

1. Trigger obsługujący zmianę statusu

alter trigger zmiana\_rezerwacji

on rezerwacje

instead of update

as

begin try

if (select count(\*) from inserted) > 1

raiserror('modyfikacja kilku wierszy jest

zabroniona', 16, 1)

declare @id\_rezerwacji int

declare @status char(1)

select @id\_rezerwacji = nr\_rezerwacji,

@status = status

from inserted

if @id\_rezerwacji is null

or @status is null

raiserror('argumenty nie mogą przyjmować

wartości null', 16, 1)

if not exists

(

select \* from rezerwacje

where nr\_rezerwacji = @id\_rezerwacji

)

raiserror('rezerwacja o podanym id nie

istnieje na liscie rezerwacji', 16 ,1)

if @status not in ('N', 'P', 'Z', 'A')

raiserror('podano status z poza zakresu

mozliwych wartosci', 16 ,1)

declare @id\_wycieczki int;

set @id\_wycieczki =

(

select id\_wycieczki

from rezerwacje

where nr\_rezerwacji = @id\_rezerwacji

)

declare @liczba\_wolnych\_miejsc int

set @liczba\_wolnych\_miejsc =

(

select liczba\_wolnych\_miejsc

from wycieczki as w

where w.id\_wycieczki = @id\_wycieczki

)

if @status in ('N', 'P', 'Z')

and @liczba\_wolnych\_miejsc < 1

raiserror('zmiana tej rezerwacji jest nie

mozliwa, brak miejsca', 16 ,1)

update rezerwacje

set status = @status

where nr\_rezerwacji = @id\_rezerwacji

insert rez\_log(id\_rezerwacji, data, status)

values(@id\_rezerwacji, getdate(), @status)

end try

begin catch

print error\_message()

declare @err\_msg varchar(100)

set @err\_msg = error\_message()

raiserror(@err\_msg, 16, 1)

end catch

1. Trigger obsługujący zamianę liczby miejsc na poziomie wycieczki

alter trigger zmiana\_liczby\_miejsc

on wycieczki

instead of update

as

begin try

if (select count(\*) from inserted) > 1

raiserror('modyfikacja kilku wierszy jest

zabroniona', 16, 1)

declare @id\_wycieczki int

declare @liczba\_miejsc int

select @id\_wycieczki = id\_wycieczki,

@liczba\_miejsc = liczba\_miejsc

from inserted

if @id\_wycieczki is null

or @liczba\_miejsc is null

raiserror('argumenty nie mogą przyjmować

wartości null', 16, 1)

if @liczba\_miejsc < 0

raiserror('argument liczba\_miejsc nie moze byc

mniejszy od zera', 16, 1)

if not exists

(

select \* from wycieczki as w

where w.id\_wycieczki = @id\_wycieczki

)

raiserror('wycieczka o podanym id nie

istnieje', 16, 1)

declare @liczba\_miejsc\_zarezerwowanych int

set @liczba\_miejsc\_zarezerwowanych =

(

select liczba\_wolnych\_miejsc

from wycieczki as w

where w.id\_wycieczki = @id\_wycieczki

)

if @liczba\_miejsc\_zarezerwowanych <

@liczba\_miejsc

raiserror('zmniejszenie liczby miejsc nie jest

mozliwe', 16, 1)

update wycieczki

set liczba\_miejsc = @liczba\_miejsc

where id\_wycieczki = @id\_wycieczki

end try

begin catch

print error\_message()

declare @err\_msg varchar(100)

set @err\_msg = error\_message()

raiserror(@err\_msg, 16, 1)

end catch

1. procedury dodaj\_rezerwacje, zmien\_status\_rezerwacji i zmien\_liczbe\_miejsc zostały całkowicie zastąpione poprzez odpowiednie triggery.